

Rexroth IndraMotion MLC 14VRS

RegisterControl

Bibliothek
R911341508

Ausgabe 01



Titel Rexroth IndraMotion
MLC 14VRS
RegisterControl

Art der Dokumentation Bibliothek

Dokumentations-Type DOK-MLC***-REGI*CO*V14-LI01-DE-P

Interner Ablagevermerk RS-a51833f0e93de9c00a6846a5017fafb6-1-de-DE-16

Änderungsverlauf

Ausgabe	Stand	Bemerkung
01	2014-11	Erstausgabe

Schutzvermerk © Bosch Rexroth AG 2014
Alle Rechte bei Bosch Rexroth AG, auch für den Fall von Schutzrechtsanmeldungen. Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht, bei uns.

Verbindlichkeit Die angegebenen Daten dienen allein der Produktbeschreibung und sind nicht als zugesicherte Eigenschaften im Rechtssinne zu verstehen. Änderungen im Inhalt der Dokumentation und Liefermöglichkeiten der Produkte sind vorbehalten.

Redaktion Systementwicklung Automation Motion Logic Control, HoSc (FrWe)

Inhaltsverzeichnis

	Seite
1 Zu dieser Dokumentation.....	3
1.1 Gültigkeit der Dokumentation.....	3
1.2 Erforderliche und ergänzende Dokumentationen.....	3
1.3 Darstellung von Informationen.....	9
1.3.1 Gebrauch der Sicherheitshinweise.....	9
1.3.2 Bezeichnungen.....	11
1.3.3 Abkürzungen.....	11
1.4 Kundenfeedback.....	11
2 Wichtige Gebrauchshinweise.....	13
2.1 Bestimmungsgemäßer Gebrauch.....	13
2.1.1 Einführung.....	13
2.1.2 Einsatz- und Anwendungsbereiche.....	13
2.2 Nicht-bestimmungsgemäßer Gebrauch.....	14
3 RMB_TechRegiExtended.....	15
3.1 Lizenzpflichtige Bibliothek RMB_TechRegiExtended.....	15
3.2 Allgemeines.....	17
3.3 Funktionsbausteine zur Kommunikation mit dem Markenstromsensor.....	18
3.3.1 Einführung und Übersicht.....	18
3.3.2 MB_RegiSensor01CommType01.....	22
3.3.3 MB_RegiSensor01ConfigType01.....	40
3.3.4 MB_RegiSensor01SendRecvDataType01.....	56
3.3.5 MB_RegiSensor01CycComType01.....	60
3.3.6 MB_RegiSensor01CycComType02.....	64
3.3.7 MB_RegiSensor01CommType02.....	68
3.3.8 MB_RegiSensor01ConfigType02.....	87
3.3.9 MB_RegiSensor01SendRecvDataType02.....	105
3.3.10 MB_RegiSensor01CycComType04.....	108
3.3.11 MB_RegiSensor01ReadDiagType01.....	112
3.3.12 MB_RegiSensor01SendSpeedAccType01.....	115
3.3.13 MB_RegiSensor01TeachDataType01.....	119
3.3.14 MB_RegiSensor01SetIPAddressType01.....	122
3.4 Funktionsbaustein zum Erzeugen bestimmter Geschwindigkeitsverläufe.....	124
3.4.1 Einführung und Übersicht.....	124
3.4.2 MB_ReachVelocityType01.....	124
4 RMB_TechRegiAdvanced.....	129
4.1 Lizenzpflichtige Bibliothek RMB_TechRegiAdvanced.....	129
4.2 Allgemeines.....	131
4.3 Funktionsbausteine zur Regelung und Optimierung des Druckprozesses.....	131
4.3.1 Einführung und Übersicht.....	131
4.3.2 MB_RegisterControllerAdvType01.....	132

Inhaltsverzeichnis

	Seite
4.3.3 MB_RegiAccelerationCompensationType01.....	142
4.3.4 MB_RegiGravureRegulateType01.....	148
4.3.5 MB_RegiGravureRegulateType02.....	155
 5 Service und Support.....	 161
Index.....	163

1 Zu dieser Dokumentation

1.1 Gültigkeit der Dokumentation

Diese Dokumentation richtet sich an Anwender, die mit einer Auswerteeinheit (Markenstromsensor) die Position von Registermarken ermitteln möchten. Weiterhin werden die Funktionsbausteine zur Registerregelung (automatische Reglerparametrierung, Beschleunigungskompensation und Entkopplungsnetzwerk) beschrieben.

Zielgruppe In der folgenden Grafik beziehen sich die umrandeten Aktivitäten, Produktphasen und Zielgruppen auf die vorliegende Dokumentation.

Beispiel: In der Produktphase "Engineering" kann die Zielgruppe "Programmierer" mit Hilfe dieser Dokumentation die Aktivität "Parametrieren" ausführen.

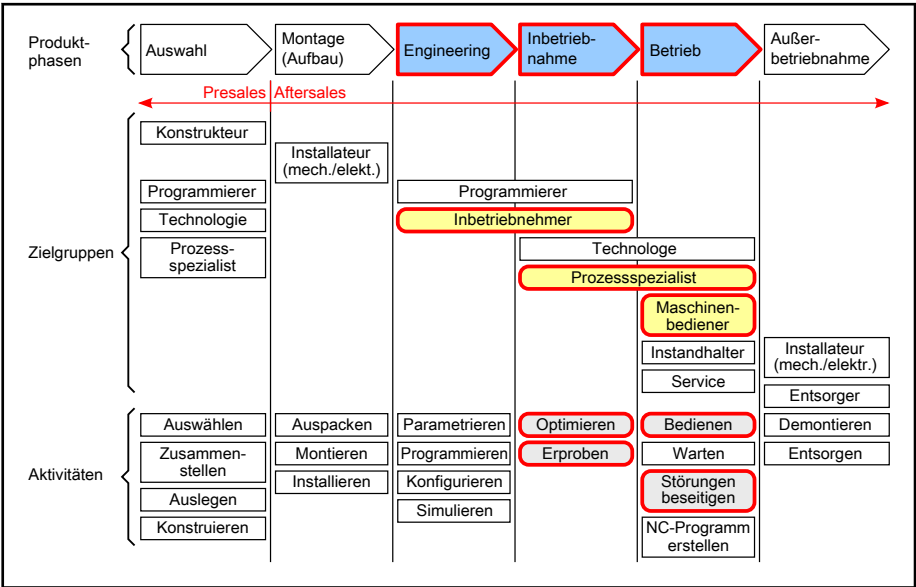


Abb. 1-1: Zuordnung der vorliegenden Dokumentation zu den Zielgruppen, Produktphasen und den Aktivitäten der Zielgruppe

1.2 Erforderliche und ergänzende Dokumentationen

Dokumentationstitel mit Typenschlüssel und Materialnummer

Systemübersicht
Rexroth IndraLogic XLC 14VRS Systemübersicht DOK-XLC***-SYSTEM**V14-PRRS-DE-P, R911341482 Diese Dokumentation stellt eine Übersicht der möglichen Hardware-/Softwarekomponenten des Automationssystems IndraLogic XLC in der genannten Version dar. Die Dokumentation dient als Auswahlhilfe zur Zusammenstellung eines Systems.
Rexroth IndraMotion MLC 14VRS Systemübersicht DOK-MLC***-SYSTEM**V14-PRRS-DE-P, R911341503 Diese Dokumentation stellt eine Übersicht der möglichen Hardware-/Softwarekomponenten des Automationssystems IndraMotion MLC in der genannten Version dar. Die Dokumentation dient als Auswahlhilfe zur Zusammenstellung eines Systems.

Tab. 1-1: XLC/MLC-Dokumentationsübersicht - Systemübersicht

Zu dieser Dokumentation

Erste Schritte
Rexroth IndraMotion MLC 14VRS Erste Schritte DOK-MLC***-F*STEP**V14-CORS-DE-P, R911341516 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraMotion MLC und RobotControl. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.
Rexroth IndraLogic XLC 14VRS Erste Schritte DOK-XLC***-F*STEP**V14-CORS-DE-P, R911341488 Diese Dokumentation beschreibt die "ersten Schritte" mit der IndraLogic XLC. Beschrieben werden die Hard- und Softwarevoraussetzungen und das Erstellen eines Projekts.
Rexroth IndraMotion MLC 14VRS Projektkonvertierung DOK-XLCMLC-PROCONV*V14-APRS-DE-P, R911341493 Diese Dokumentation beschreibt die Projektkonvertierung der IndraMotion MLC.

Tab. 1-2: XLC/MLC-Dokumentationsübersicht - Erste Schritte

Engineering
Rexroth IndraWorks 14VRS Software-Installation DOK-IWORKS-SOFTINS*V14-CORS-DE-P, R911344285 Die Dokumentation beschreibt den Installationsprozess von IndraWorks.
Rexroth IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-DE-P, R911343564 Diese Dokumentation beschreibt die Verwendung von IndraWorks, in das sich die Rexroth Engineering-Tools integrieren. Beschrieben werden das Arbeiten mit IndraWorks und die Bedienung der Oszilloskop-Funktion.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Funktionsbeschreibung DOK-XLCMLC-FUNC****V14-APRS-DE-P, R911341699 Die Dokumentation dient der Beschreibung der Wizards, Kontextmenüs, Dialoge, Steuerungsinbetriebnahme, Gerätekonfiguration, Funktionalitäten der IndraMotion MLC.

Tab. 1-3: XLC/MLC-Dokumentationsübersicht - Engineering

Diagnose und Service
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Diagnosen DOK-XLCMLC-DIAG****V14-RERS-DE-P, R911341480 Diese Dokumentation enthält alle in den Steuerungssystemen IndraLogic XLC und IndraMotion MLC implementierten Steuerungsparameter.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Parameter DOK-XLCMLC-PARAM***V14-RERS-DE-P, R911341478 Diese Dokumentation beschreibt die Parameter der Systeme XLC/MLC, sowie das Wechselspiel zwischen Parametrieren und Programmieren. Beschrieben werden die Achs-Parameter, die Steuerungs-Parameter, die Kinematik-Parameter, die Messtaster- und die Nockenschaltwerks-Parameter.

Zu dieser Dokumentation

Diagnose und Service Rexroth IndraLogic XLC IndraMotion MLC 14VRS Inbetriebnahme DOK-XLCMLC-STARTUP*V14-CORS-DE-P, R911341500 Das vorliegende Handbuch beschreibt die Inbetriebnahme- und Servicearbeitsschritte für die Systeme IndraLogic XLC und IndraMotion MLC. Es beinhaltet Checklisten für häufig durchzuführende Arbeiten und eine ausführliche Beschreibung der Arbeitsschritte. Rexroth IndraWorks IndraMotion Service Tool DOK-IWORKS-IMST*****-APRS-DE-P, R911341382 Diese Dokumentation beschreibt das IndraMotion Service Tool (IMST). Das IMST ist ein webbasiertes Diagnosewerkzeug, das zum Zugriff auf ein Steuerungssystem über eine Ethernet-Hochgeschwindigkeitsverbindung dient. Das IMST ermöglicht OEMs, Endanwendern und Kundenbetreuern von überall auf ein System zuzugreifen und Ferndiagnosen zu stellen. Voraussetzung ist ein PC mit Internet Explorer Version 8 oder Firefox 3.5 oder eine höhere Version. Die folgenden Steuerungs-Varianten werden unterstützt: • IndraMotion MLC L25/L45/L65 • IndraLogic XLC L25/L45/L65/VEP

Tab. 1-4: XLC/MLC-Dokumentationsübersicht - Diagnose und Service

Antriebsregelgeräte Rexroth IndraDrive Firmware für Antriebsregelgeräte MPH-, MPB-, MPD-, MPC-08 DOK-INDRV*-MP*-08VRS**-APxx-DE-P-DE-P, R911332642 Diese Dokumentation beschreibt alle funktionellen Eigenschaften der IndraDrive-Firmware in den Varianten MPH-08, MPB-08, MPD-08 und MPC-08. Rexroth IndraDrive Antriebsregelgeräte MPx-02 bis MPx-08 DOK-INDRV*-GEN-**VRS**-PAxx-DE-P, R911297316 Diese Dokumentation enthält die Beschreibung aller, in der Firmware für Antriebsregelgeräte der IndraDrive-Familie implementierten Parameter. Sie unterstützt die Parametrierung der Antriebsregelgeräte. Rexroth IndraDrive MPx-02 bis MPx-08 und HMV DOK-INDRV*-GEN-**VRS**-WAxx-DE-P, R911297318 Diese Dokumentation enthält die Beschreibung aller in den folgenden Firmwares implementierten Diagnosen: • Antriebsregelgerät-Firmwares MPx-02 bis MPx-08 und • Firmwares der Versorgungsgeräte des Typs "HMV". Sie unterstützt den Maschinenbediener und den Anlagenprogrammierer bei der Fehlerbehebung. Rexroth IndraDrive Steuerteile CSB02, CSE02, CSH02, CDB02 DOK-INDRV*-CXX02*****-PRRS-DE-P, R911338961 Rexroth IndraDrive MPx-18 Funktionen DOK-INDRV*-MP*-18VRS**-APRS-DE-P, R911338674 Diese Dokumentation beschreibt alle funktionellen Eigenschaften der IndraDrive-Firmware in den Varianten MPB-18, MPM-18, MPC-18 und MPE-18.

xx entsprechender Ausgabestand

Tab. 1-5: XLC/MLC-Dokumentationsübersicht - Antriebsregelgeräte

Zu dieser Dokumentation

SPS
Rexroth IndraWorks 14VRS IndraLogic 2G SPS-Programmiersystem DOK-IWORKS-IL2GPRO*V14-APRS-DE-P, R911343570 Diese Dokumentation beschreibt das SPS-Programmierzug IndraLogic 2G und seine Verwendung. Beschrieben werden die Basisverwendung, Erste Schritte, die Visualisierung, Menüpunkte und die Editoren.
Rexroth IndraWorks 14VRS Basis-Bibliotheken IndraLogic 2G DOK-IL*2G*-BASLIB**V14-LIRS-DE-P, R911343918 Das vorliegende Handbuch beschreibt die systemübergreifenden SPS-Bibliotheken.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS PLCopen-Bibliotheken DOK-XLCMLC-FUNLIB**V14-LIRS-DE-P, R911341490 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken RIL_CommonTypes, ML_Base und ML_PLCopen für die IndraLogic XLC/IndraMotion MLC. Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.

Tab. 1-6: XLC/MLC-Dokumentationsübersicht - SPS

Feldbusse
Rexroth IndraWorks 14VRS Feldbusse DOK-IWORKS-FB*****V14-APRS-DE-P, R911341484 Die Dokumentation dient der Beschreibung der von den Systemen IndraLogic XLC, IndraMotion MLC und IndraMotion MTX unterstützten Feldbus- und lokalen Peripherieanschlüssen. Der Schwerpunkt dieser Dokumentation liegt in der Konfiguration, Parametrierung, Inbetriebnahme und Diagnose der verschiedenen Peripherieanschlüssen. Sie ist die Grundlage für die Online-Hilfe.
Rexroth IndraWorks 14VRS Feldbus-Bibliotheken DOK-IWORKS-FB*LIB**V14-LIRS-DE-P, R911343574 Das vorliegende Handbuch beschreibt die Bibliotheken der Feldbusse: RIL_ProfibusDP_02, RIL_ProfibusDP Slave, RIL_ProfinetIO, RIL_ProfinetIODevice, RIL_EtherNetIPAdapter, RIL_MappingList, RIL_SERCOSIII, RIL_Inline, einschließlich deren Diagnosen, sowie die Fehlerreaktion der Bausteine.
Rexroth IndraWorks 12VRS FDT-Container DOK-IWORKS-FDT*CON*V12-APRS-DE-P, R911334397 Diese Dokumentation beschreibt die FDT-Container-Funktionalität von IndraWorks. Beschrieben wird das Aktivieren der Funktionalität im Projekt und das Arbeiten mit DTMs.
sercos Systemhandbuch für I/O-Geräte DOK-CONTRL-ILS3*****-APxx-DE-P, R911333511 Diese Dokumentation beschreibt die Projektierung, Parametrierung, Inbetriebnahme und Diagnose von I/O-Geräten mit sercos Schnittstelle.

xx entsprechender Ausgabestand

Tab. 1-7: XLC/MLC-Dokumentationsübersicht - Feldbusse

Zu dieser Dokumentation

HMI
Rexroth IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-DE-P, R911343567 Diese Dokumentation beschreibt die Funktionen, die Konfiguration und die Bedienung der Bedienoberflächen IndraWorks HMI Engineering und IndraWorks HMI Operation.
Rexroth IndraWorks 14VRS WinStudio DOK-IWORKS-WINSTUD*V14-APRS-DE-P, R911341584 Dieses "Benutzerhandbuch und Technische Referenz" soll Sie dabei unterstützen, mit Ihrer Software "Rexroth WinStudio" optimale Ergebnisse zu erzielen. Das Dokument bietet technische Informationen und Schritt-für-Schritt-Anleitungen zum Anlegen von webfähigen HMI/SCADA-Programmen.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS HMI-Anbindung DOK-XLCMLC-HMI*****V14-APRS-DE-P, R911341496 Dieses Dokument beschreibt die von der IndraLogic XLC und IndraMotion MLC unterstützten Visualisierungssysteme sowie deren Anbindung.

Tab. 1-8: XLC/MLC-Dokumentationsübersicht - HMI

Technologie
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Generic Application Template DOK-XLCMLC-TF*GAT**V14-APRS-DE-P, R911341486 Dieses Handbuch stellt dem IndraLogic SPS-Programmierer ein strukturiertes Template, welches zum Hinzufügen und Bearbeiten des SPS-Programmiercodes verwendet werden kann, zur Verfügung. Beschrieben werden das Template selbst, der Template Wizard und Beispielapplikationen.
Rexroth IndraMotion MLC 14VRS Technologie-Bibliotheken DOK-MLC***-TF*LIB**V14-LIRS-DE-P, R911341510 Diese Dokumentation beschreibt die Funktionsbausteine, Funktionen und Datentypen der Bibliotheken "ML_TechInterface.library", "ML_TechMotion.library", "RMB_TechCam.library" und "ML_TechBase.library" sowie Bibliotheken zur Wicklerfunktionalität, Registerreglerfunktionalität und Querschneiderfunktionalität.
Rexroth IndraWorks Energy Efficiency Management 14VRS DOK-IWORKS-4EE*****V14-APRS-DE-P, R911339228 Diese Dokumentation beschreibt die HMI-Anbindung des Energiemanagement-Systems und die Funktionsbausteine, Funktionen und Datentypen der Bibliothek "RIL_4EE ". Die Dokumentation enthält ferner die Fehlerreaktion der Bausteine.
Rexroth IndraMotion MLC 14VRS RegisterControl (Bibliothek) DOK-MLC***-REGI*CO*V14-LIRS-DE-P, R911341508 Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.
Rexroth IndraMotion MLC 14VRS RegisterControl (Anwendungsbeschreibung) DOK-MLC***-REGI*CO*V14-APRS-DE-P, R911341506 Dieses Dokument beschreibt die Anwendung der integrierten Registerregelung für eine Tiefdruckmaschine. Beschrieben werden die Komponenten des Markenstromsensors, die HMI-Anwendung und die Fehlerbehebung. Diese Anleitung dient dazu, die Registerregelung zu bedienen und auf Fehler zu reagieren sowie Diagnosen abzufragen. Die Zielgruppe für diese Dokumentation sind Maschineneinrichter und Maschinenbediener.
Rexroth IndraMotion MLC 14VRS Anwendung Wicklerfunktionen DOK-MLC***-TF*WIND*V14-APRS-DE-P, R911341512 Diese anwendungsbezogene Systemdokumentation beschreibt die Anwendung der Wickler-Technologiefunktionen.

Zu dieser Dokumentation

Technologie
Rexroth IndraWorks 13VRS CamBuilder DOK-IWORKS-CAMBUIL*V13-APRS-DE-P, R911336290 Diese Dokumentation dient zur Beschreibung der Grundlagen und der Bedienung des CamBuilder, des Werkzeugs zur Kurvenscheibenbearbeitung.
Rexroth IndraMotion MLC 14VRS Robot-Control V2 DOK-MLC***-ROCO****V14-RERS-DE-P, R911341586 Die Dokumentation informiert über die Robot-Control V2. Die Programmierung über PLCopen steht hierbei im Vordergrund. Beschrieben werden die Programmstruktur, Variablen, Funktionen, Bewegungsanweisungen sowie die notwendigen Parameter des Systems. Die Dokumentation enthält ferner die Beschreibung der Bibliothek "ML_Robot". Robot-Control V2 wird unterstützt durch ein Kinematik-Interface. Die Komponenten dieses Interfaces werden in der Bibliothek "ML_KinTech" beschrieben.

Tab. 1-9: XLC/MLC-Dokumentationsübersicht - Technologie

Open Core Engineering - OCE
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Erste Schritte MLPI DOK-XLCMLC-MLPI****V14-CORS-DE-P, R911341032 Dieses Dokument beschreibt die Installation und Inbetriebnahme der MLPI-Schnittstelle.
Rexroth IndraLogic XLC IndraMotion MLC 14VRS Automation Interface DOK-XLCMLC-AUT*INT*V14-APRS-DE-P, R911341498 Diese Dokumentation beschreibt den script-basierten Zugriff auf IndraWorks-Projektdateien über die Schnittstelle des Automation Interfaces.

Tab. 1-10: XLC/MLC-Dokumentationsübersicht - OCE

SafeLogic
Rexroth IndraWorks SafeLogic 14VRS Die ersten Schritte DOK-IWORKS-SL*STEP*V14-CORS-DE-P, R911341519 Dieses Dokument beschreibt die Erstinbetriebnahme des Safety-Funktionsmoduls und dessen externer Safety-Peripherie. Am Beispiel eines Projektes werden alle notwendigen Schritte zur Inbetriebnahme einer einfachen Safety-Anwendung durchgeführt.
Rexroth IndraMotion MLC IndraLogic XLC 14VRS SafeLogic Systemübersicht DOK-XLCMLC-SL**SYS*V14-PRRS-DE-P, R911341695 Dieses Dokument beschreibt die Projektierung, die Erstellung und die Inbetriebnahme von sicherheitstechnischen Vorrichtungen.
Rexroth IndraWorks 14VRS SafeLogic Projektkonfiguration DOK-IWORKS-SL**PRJ*V14-APRS-DE-P, R911341693 Dieses Dokument beschreibt die Projektierung von Safety-Anwendungen.
Rexroth IndraControl SafeLogic Funktionsmodul DOK-CTRL-SL**FM*****-ITRS-DE-P, R911336575 Dieses Dokument liegt der Hardware bei und beschreibt die Inbetriebnahme, die Installation, die Außerbetriebnahme und Entsorgung der Hardware.

Tab. 1-11: XLC/MLC-Dokumentationsübersicht - SafeLogic

Hardware
Rexroth IndraControl L45/L65/L85 Steuerung DOK-CONTRL-ICL45L65L85-PRxx-DE-P, R911332115 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl L45/L65/L85.
Rexroth IndraControl L25 DOK-CONTRL-IC*L25****-PRxx-DE-P, R911328473 Diese Dokumentation dient der Beschreibung der Steuerungen IndraControl L25.
Rexroth IndraControl Lxx 14VRS Funktionsmodule DOK-CONTRL-FM*LXX**V14-APRS-DE-P, R911341582 Diese Dokumentation dient der Beschreibung aller Funktionsmodule der Lxx Steuerungen inkl. Engineering und Diagnose.

xx entsprechender Ausgabestand
Tab. 1-12: XLC/MLC-Dokumentationsübersicht - Hardware

1.3 Darstellung von Informationen

1.3.1 Gebrauch der Sicherheitshinweise

Aufbau der Sicherheitshinweise

Die Sicherheitshinweise sind wie folgt aufgebaut:

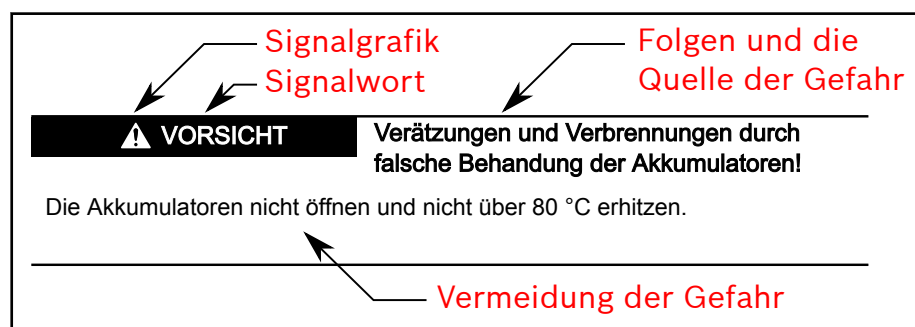


Abb. 1-2: Aufbau der Sicherheitshinweise

Erläuterung der Signalwörter und der Signalgrafik

Die Sicherheitshinweise in der vorliegenden Dokumentation beinhalten bestimmte Signalwörter (Gefahr, Warnung, Vorsicht, Hinweis) und gegebenenfalls eine Signalgrafik (nach ANSI Z535.6-2006).

Das Signalwort soll die Aufmerksamkeit auf den Sicherheitshinweis lenken und bezeichnet die Schwere der Gefährdung.

Die Signalgrafik (Warndreieck mit Ausrufezeichen), welche den Signalwörtern Gefahr, Warnung und Vorsicht vorangestellt wird, weist auf Gefährdungen für Personen hin.

⚠ GEFAHR

Bei Nichtbeachtung dieses Sicherheitshinweises **werden** Tod oder schwere Körperverletzung eintreten.

Zu dieser Dokumentation

** WARNUNG**

Bei Nichtbeachtung dieses Sicherheitshinweises **können** Tod oder schwere Körperverletzung eintreten.

** VORSICHT**

Bei Nichtbeachtung dieses Sicherheitshinweises können mittelschwere oder leichte Körperverletzung eintreten.

HINWEIS

Bei Nichtbeachtung dieses Sicherheitshinweises können Sachschäden eintreten.

Verwendete Symbole

Fingerzeige werden wie folgt dargestellt:



Dies ist ein Hinweis.

Tipps werden wie folgt dargestellt:



Dies ist ein Tipp.

1.3.2 Bezeichnungen

Bezeichnung	Erklärung
Runtime-Lizenz	Zur Laufzeit wird auf Freischaltung geprüft. Pro Steuerung, auf der die Funktionsbausteine verwendet werden, ist jeweils eine Lizenz erforderlich.
Markenstromsensor	Auswerteinheit, die einen Markenstrom abtastet und Markenpositionen vermisst.
Teach-In	Prozess, in dem sich der Markenstromsensor die Schwellwerte für die Markenmessung ermittelt.
Messung	Es werden die Markenpositionen der definierten Marken ermittelt und an die Steuerung gesendet.
Barcode	Sequenz von mindestens 3 Blöcken, die durch den Sensor automatisch gefunden wird.
S-Kurve	Ruckbegrenzte Beschleunigungsphase

Tab. 1-13: Verwendete Bezeichnungen

1.3.3 Abkürzungen

Abkürzung	Bedeutung
LWL	Lichtwellenleiter
D.h.	Das heißt
mm	Millimeter
z.B.	zum Beispiel
n.def.	nicht definiert
vgl.	vergleiche
MLC	Motion logic control
s	Sekunde
m	Meter
%	Prozent

Tab. 1-14: Verwendete Abkürzungen

1.4 Kundenfeedback

Anregungen, Wünsche oder Verbesserungen von unseren Kunden haben bei uns einen hohen Stellenwert. Senden Sie uns Ihre Anmerkungen zu den Do-

Zu dieser Dokumentation

kumentationen per E-Mail an Feedback.Documentation@boschrexroth.de.
Sie können direkt im elektronischen PDF-Dokument Kommentare einfügen
und uns die PDF-Datei zusenden.

2 Wichtige Gebrauchshinweise

2.1 Bestimmungsgemäßer Gebrauch

2.1.1 Einführung

Die Produkte von Bosch Rexroth werden nach dem jeweiligen Stand der Technik entwickelt und gefertigt. Vor ihrer Auslieferung werden sie auf ihren betriebssicheren Zustand hin überprüft.

Die Produkte dürfen nur bestimmungsgemäß eingesetzt werden. Wenn sie nicht bestimmungsgemäß eingesetzt werden, dann können Situationen entstehen, die Sach- und Personenbeschädigung nach sich ziehen.



Für Schäden bei nicht-bestimmungsgemäßigem Gebrauch der Produkte leistet Bosch Rexroth als Hersteller keinerlei Gewährleistung, Haftung oder Schadensersatz; die Risiken bei nicht-bestimmungsgemäßigem Gebrauch der Produkte liegen allein beim Anwender.

Bevor Sie die Produkte der Firma Bosch Rexroth einsetzen, müssen die folgenden Voraussetzungen erfüllt sein, um einen bestimmungsgemäßen Gebrauch der Produkte zu gewährleisten:

- Jeder, der in irgendeiner Weise mit einem unserer Produkte umgeht, muss die entsprechenden Sicherheitsvorschriften und den bestimmungsgemäßen Gebrauch lesen und verstehen.
- Sofern es sich bei den Produkten um Hardware handelt, müssen sie in ihrem Originalzustand belassen werden; d. h. es dürfen keine baulichen Veränderungen an ihnen vorgenommen werden. Softwareprodukte dürfen nicht dekompiert werden und ihre Quellcodes dürfen nicht verändert werden.
- Beschädigte oder fehlerhafte Produkte dürfen nicht eingebaut oder in Betrieb genommen werden.
- Es muss gewährleistet sein, dass die Produkte entsprechend den in der Dokumentation genannten Vorschriften installiert sind.

2.1.2 Einsatz- und Anwendungsbereiche

**Rexroth IndraLogic XLC /
IndraMotion MLC**

Die IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule sind für Motion-/Logic-Anwendungen geeignet.



Die IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule dürfen nur mit den in dieser Dokumentation angegebenen Zubehör- und Anbauteilen benutzt werden. Nicht ausdrücklich genannte Komponenten dürfen weder angebaut noch angeschlossen werden. Gleiches gilt für Kabel und Leitungen.

Der Betrieb darf nur in den ausdrücklich angegebenen Konfigurationen und Kombinationen der Hardware-Komponenten und mit der in den jeweiligen Dokumentationen und den Funktionsbeschreibungen angegebenen und spezifizierten Soft- und Firmware erfolgen.

Die IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule sind für den Einsatz in ein- und mehrachsigen Antriebs- und Steuerungsaufgaben entwickelt worden.

Wichtige Gebrauchshinweise

Für den applikationsspezifischen Einsatz des Maschinenbedien- und Visualisierungsterminals stehen Gerätetypen mit unterschiedlicher Ausstattung und unterschiedlichen Schnittstellen zur Verfügung.

Typische Anwendungsbereiche für IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule sind:

- Handhabungs- und Montagesysteme
- Nahrungsmittel- und Verpackungsmaschinen
- Druck- und Verarbeitungsmaschinen

Rexroth IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule dürfen nur unter den in den weiterführenden Dokumentationen angegebenen Montage- und Installationsbedingungen, in der angegebenen Gebrauchslage und unter den angegebenen Umweltbedingungen (Temperatur, Schutzart, Feuchtigkeit, EMV u. a.) betrieben werden.

2.2 Nicht-bestimmungsgemäßer Gebrauch

Die Verwendung der Rexroth IndraLogic XLC / IndraMotion MLC und ihrer Funktionsmodule außerhalb der vorgenannten Anwendungsgebiete oder unter anderen als den in der Dokumentation beschriebenen Betriebsbedingungen und angegebenen technischen Daten gilt als "nicht-bestimmungsgemäß".

Die Rexroth IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule dürfen nicht eingesetzt werden, wenn sie

- Betriebsbedingungen ausgesetzt wird, die die vorgeschriebenen Umgebungsbedingungen nicht erfüllen. Untersagt sind z. B. der Betrieb unter Wasser, unter extremen Temperaturschwankungen oder extremen Maximaltemperaturen.
- Außerdem dürfen die IndraLogic XLC / IndraMotion MLC und ihre Funktionsmodule nicht bei Anwendungen eingesetzt werden, die von Bosch Rexroth nicht ausdrücklich freigegeben sind. Beachten Sie hierzu bitte unbedingt die Aussagen in den allgemeinen Sicherheitshinweisen!

3 RMB_TechRegiExtended

3.1 Lizenzpflichtige Bibliothek RMB_TechRegiExtended



Die Verwendung der Funktionsbausteine der Bibliothek RMB_TechRegiExtended unterliegt einer Lizenzpflicht. Je nach Anwendung muss eine der 3 Lizenzen FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT, FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1 oder FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2 zur Nutzung der Funktionsbausteine erworben werden. Pro Steuerung, auf der die Funktionsbausteine verwendet werden, ist jeweils eine Lizenz erforderlich (Runtime-Lizenz).

Einzellizenz Technologiebibliothek

Berechtigt zur Verwendung mit einer einzigen Steuerungsfirmware.

Single licence for technology library

Authorize the use with exactly one controller firmware.

TechLib RegisterControl Extended TechLib RegisterControl Extended		
Licence key (MNR): R911xxxxxx	Serial number: 1234567	Type code: FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT
Registration key WAHOF2-FBJYZF-8CSB0F-9DQO0D-9E8ZER		

Abb. 3-1: Lizenzblatt FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT

Einzellizenz Technologiebibliothek

Berechtigt zur Verwendung mit einer einzigen Steuerungsfirmware.

Single licence for technology library

Authorize the use with exactly one controller firmware.

TechLib RegisterControl Advanced 1 TechLib RegisterControl Advanced 1		
Licence key (MNR): R911xxxxxx	Serial number: 1234567	Type code: FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1
Registration key WAHOF2-FBJYZF-8CSB0F-9DQO0D-9E8ZER		

Abb. 3-2: Lizenzblatt FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1

RMB_TechRegiExtended

Einzellizenz Technologiebibliothek Berechtigt zur Verwendung mit einer einzigen Steuerungsfirmware.		
Single licence for technology library Authorize the use with exactly one controller firmware.		
TechLib RegisterControl Advanced 2 TechLib RegisterControl Advanced 2		
Licence key (MNR): R911xxxxxx	Serial number: 1234567	Type code: FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2
		
Registration key WAHOF2-FBJYZF-8CSB0F-9DQO0D-9E8ZER		

Abb. 3-3: Lizenzblatt FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2

Die Abbildungen zeigen jeweils die dazugehörigen Lizenzblätter. Pro Steuerung muss ein Lizenzblatt erworben werden, das zur Verwendung der Funktionsbausteine der Bibliothek RMB_TechRegiExtended oder RMB_TechRegiAdvanced auf diesem einen Steuerungssystem berechtigt.

Die Bibliotheken RMB_TechRegiExtended und RMB_TechRegiAdvanced werden bei den 3 Lizenzen FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT, FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1 oder FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2 mitgeliefert. Der Inhalt der jeweiligen Lizenzen ist nachfolgend aufgeführt:

FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT

- HMI für Schmalbahnanwendung (bis zu 12 Druckwerke, keine Entkopplung und Beschleunigungskompensation)
- Firmware Update Tool
- Dokumentationen: Anwenderbeschreibung und Bibliotheksbeschreibung
- Bibliotheken: RMB_TechRegiExtended und RMB_TechRegiAdvanced

FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1

- HMI für Anwendungen mit Klemmstellen (z. B. Tiefdruck) (bis zu 12 Druckwerke, mit Entkopplung und Beschleunigungskompensation)
- Firmware Update Tool
- Dokumentationen: Anwenderbeschreibung und Bibliotheksbeschreibung
- Bibliotheken: RMB_TechRegiExtended und RMB_TechRegiAdvanced

FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2

- HMI für Anwendungen mit Klemmstellen (z. B. Tiefdruck) (bis zu 24 Druckwerke, mit Entkopplung und Beschleunigungskompensation)
- Firmware Update Tool
- Dokumentationen: Anwenderbeschreibung und Bibliotheksbeschreibung
- Bibliotheken: RMB_TechRegiExtended und RMB_TechRegiAdvanced

Wird eine der Bibliotheken RMB_TechRegiExtended oder RMB_TechRegiAdvanced genutzt, so muss pro Steuerung, auf der einer der Funktionsbausteine genutzt wird, eine der Lizenzen FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT, FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1 oder FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2 vorliegen.

3.2 Allgemeines

Die Bibliothek RMB_TechRegiExtended stellt Funktionsbausteine zur Verfügung, die zur Kommunikation mit dem Markenstromsensor benötigt werden. Der Markenstromsensor besteht aus einer Auswerteeinheit, die über Lichtwellenleiter mit einer Optik verbunden sind.



Abb. 3-4: Markenstromsensor

Der Markenstromsensor ermittelt Markenpositionen von bis zu 12 Registermarken gleichzeitig und gibt deren Messwerte zurück. Neben der Hardware des Markenstromsensors handelt es sich um eine kostenpflichtige Bibliothek. Sie benötigen pro verwendetes Steuerungssystem eine Lizenz.

Um die Funktionalität des Sensors nutzen zu können, benötigen Sie:

- 1 Auswerteeinheit des Markenstromsensors
- 1 Sensoroptik
- 1 Kabelsatz (2 LWL, 1 Ethernet-, 1 Inkrementalgeber- und ein Stromkabel)
- 1 Antriebssystem mit MEM-Karte

Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.

Nachfolgende Übersicht zeigt alle Funktionsbausteine der lizenzpflichtigen Bibliothek RMB_TechRegiExtended:

Funktionsbaustein	Beschreibung
FBs für Sensorfirmwareversion <= 11	
MB_RegiSensor01CommType01	Hüllbaustein, der die komplette Kommunikation mit dem Markenstromsensor übernimmt bis Sensorfirmware 11
MB_RegiSensor01ConfigType01	Konfiguration des Markenstromsensors bis Sensorfirmware 11
MB_RegiSensor01SendRecvDataType01	Führt die azyklische Kommunikation mit dem Markenstromsensor durch bis Sensorfirmware 11
MB_RegiSensor01CycComType01	Wertet die Markenabstände und Breiten zweier Marken zueinander aus bis Sensorfirmware 11
MB_RegiSensor01CycComType02	Wertet alle Markenabstände zum Nullimpuls sowie Markenbreiten aus bis Sensorfirmware 11
FBs ab Sensorfirmwareversion >= 12	
MB_RegiSensor01CommType02	Hüllbaustein, der die komplette Kommunikation mit dem Markenstromsensor übernimmt ab Sensorfirmware 12
MB_RegiSensor01ConfigType02	Konfiguration des Markenstromsensors ab Sensorfirmware 12

RMB_TechRegiExtended

Funktionsbaustein	Beschreibung
MB_RegiSensor01SendRecvDataType02	Führt die azyklische Kommunikation mit dem Markenstromsensor durch ab Sensorfirmware 12
MB_RegiSensor01CycComType04	Wertet alle Markenabstände zum Nullimpuls sowie Markenbreiten aus ab Sensorfirmware 12
FBs unabhängig von der Sensorfirmwareversion	
MB_RegiSensor01SendSpeedAccType01	Sendet Geschwindigkeitsinformationen an den Markenstromsensor
MB_RegiSensor01ReadDiagType01	Liest Diagnosedaten des Markenstromsensors
MB_RegiSensor01TeachDataType01	Liest/Schreibt die zuletzt gültigen Messdaten des Markenstromsensors
MB_RegiSensor01SetIPAddressType01	Setzt die IP-Adresse des Markenstromsensors
MB_ReachVelocityType01	Berechnet unterschiedliche Geschwindigkeitsverläufe

Tab. 3-1: Funktionsbausteine der lizenzpflichtigen Bibliothek RMB_TechRegiExtended



Die Funktionsbausteine MB_RegiSensor01SendSpeedAccType01, MB_RegiSensor01CommType01 und MB_RegiSensor01CommType02 müssen in einer Motion synchronen Task aufgerufen werden, damit die Beschleunigung korrekt berechnet wird. Ist die aus Laufzeitgründen nicht möglich, so muss die Funktion mMotionCyclic() der beiden Funktionsbausteine aus der Motion synchronen Task aufgerufen werden und der Funktionsbaustein MB_RegiSensor01SendSpeedAccType01, MB_RegiSensor01CommType01 oder MB_RegiSensor01CommType02 kann in einer zyklischen Task aufgerufen werden.

3.3 Funktionsbausteine zur Kommunikation mit dem Markenstromsensor

3.3.1 Einführung und Übersicht

Es wird empfohlen ausschließlich die Funktionsbausteine MB_RegiSensor01CommType01 und MB_RegiSensor01CommType02 zu verwenden, da die interne Verschaltung hierbei schon vollständig implementiert ist. Der Funktionsbaustein MB_RegiSensor01CommType01 übernimmt die Kommunikation zu einem Markenstromsensor mit einer Sensorfirmwareversion ≤ 11 . Der Funktionsbaustein MB_RegiSensor01CommType02 übernimmt die Kommunikation zu einem Markenstromsensor mit einer Sensorfirmwareversion > 12 .

MB_RegiSensor01CommType01 Sensorfirmwareversion ≤ 11

Er enthält die bisherigen einzelnen Funktionsbausteine MB_RegiSensor01ConfigType01, MB_RegiSensor01CycComType02, MB_RegiSensor01ReadDiagType01, MB_RegiSensor01SendRecvDataType01, MB_RegiSensor01SendSpeedAccType01 und MB_RegiSensor01TeachDataType01 zusammengefasst. D. h. die einzelnen Funktionsbausteine werden in dem Funktionsbaustein MB_RegiSensor01CommType01 intern fest verschaltet, um es dem Anwender zu ermöglichen, den Sensor auf einfache Art und Weise zu konfigurieren und die Messung zu starten.

Der Funktionsbaustein hat intern eine feste Verschaltung der Funktionsbausteine und bietet eine einfache Bedienerschnittstelle zum Ausführen der einzelnen Funktionen. Mit dem Funktionsbaustein ist die volle Steuerung des Markenstromsensors möglich, einzig der Funktionsbaustein zum Setzen der IP-Adresse MB_RegiSensor01SetIPAddressType01 ist nicht integriert, da er

RMB_TechRegiExtended

nur einmal pro Maschine benötigt wird. Somit ergibt sich für die Kommunikation mit dem Markenstromsensor die nachfolgend gezeigte Vereinfachung.

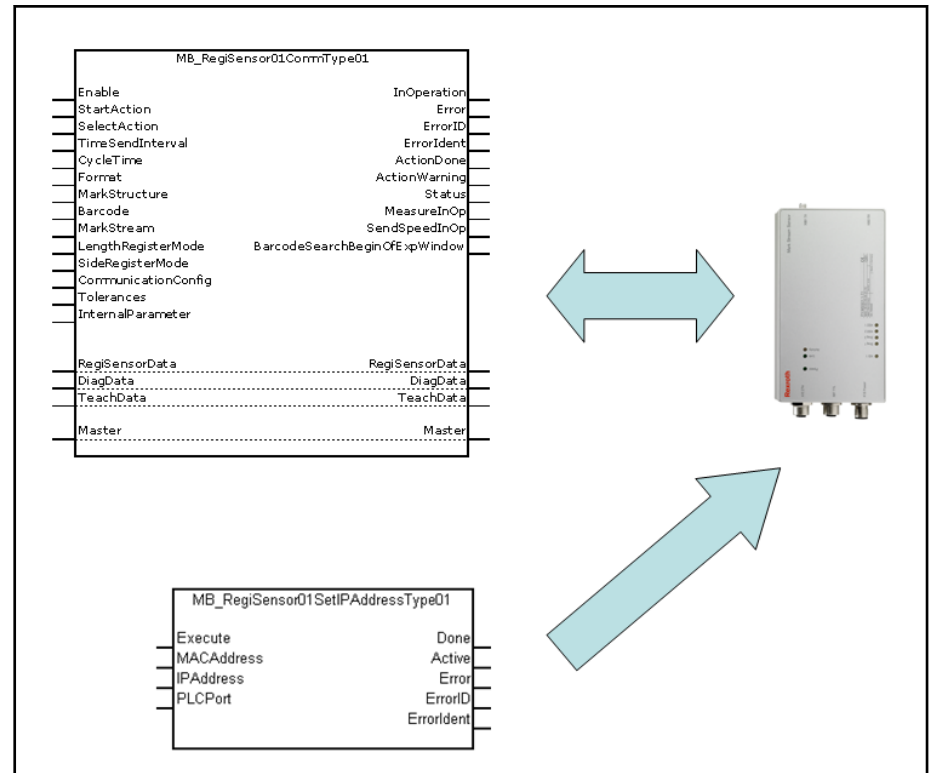


Abb. 3-5: Kommunikationsstruktur mit dem Funktionsbaustein MB_RegiSensor01CommType01

Es wird nur noch ein Funktionsbaustein MB_RegiSensor01CommType01 benötigt, sowie der Funktionsbaustein MB_RegiSensor01SetIPAddressType01. Die bisher vorgegebene Verschaltung sowie die Einschaltreihenfolge können entfallen.



Es wird aufgrund der fest vorgegebenen internen Verschaltung und der einfachen Bedienung empfohlen nur den Funktionsbaustein MB_RegiSensor01CommType01 für die Kommunikation mit dem Markenstromsensor bis zur Sensorfirmwareversion 11 zu verwenden.

MB_RegiSensor01CommType02 Sensorfirmwareversion >= 12

Er enthält die bisherigen einzelnen Funktionsbausteine MB_RegiSensor01ConfigType02, MB_RegiSensor01CycComType04, MB_RegiSensor01ReadDiagType01, MB_RegiSensor01SendRecvDataType02, MB_RegiSensor01SendSpeedAccType01 und MB_RegiSensor01TeachDataType01 zusammengefasst. D. h. die einzelnen Funktionsbausteine werden in dem Funktionsbaustein MB_RegiSensor01CommType02 intern fest verschaltet, um es dem Anwender zu ermöglichen, den Sensor auf einfache Art und Weise zu konfigurieren und die Messung zu starten.

Der Funktionsbaustein hat intern eine feste Verschaltung der Funktionsbausteine und bietet eine einfache Bedienerschnittstelle zum Ausführen der einzelnen Funktionen. Mit dem Funktionsbaustein ist die volle Steuerung des Markenstromsensors möglich, einzig der Funktionsbaustein zum Setzen der IP-Adresse MB_RegiSensor01SetIPAddressType01 ist nicht integriert, da er nur einmal pro Maschine benötigt wird. Somit ergibt sich für die Kommunikation mit dem Markenstromsensor die nachfolgend gezeigte Vereinfachung.

RMB_TechRegiExtended

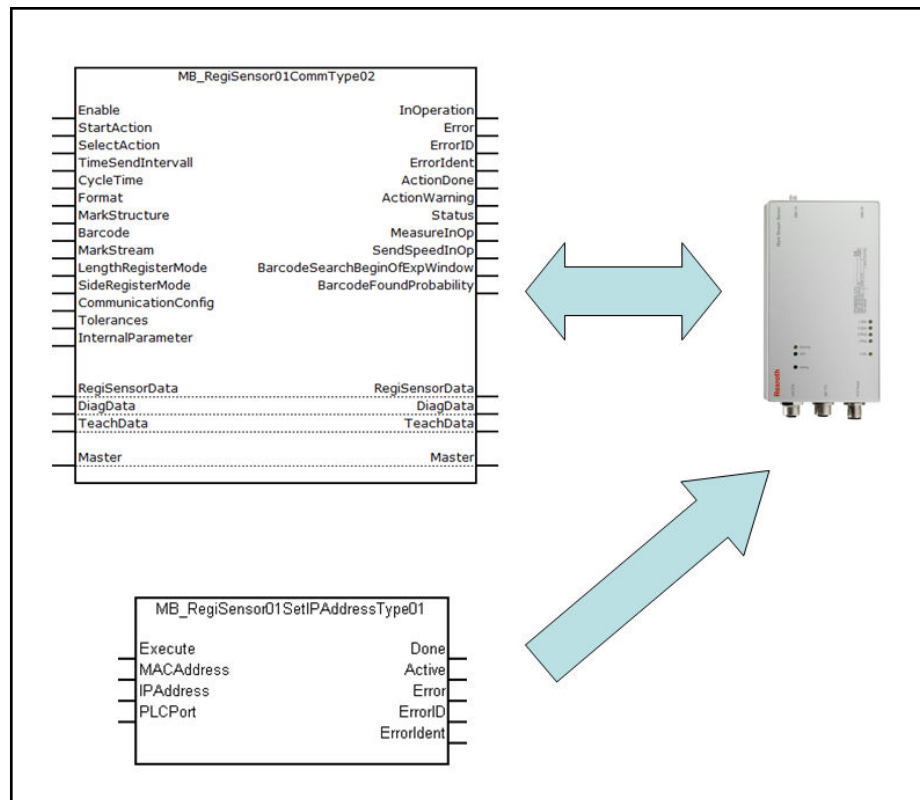


Abb. 3-6: Kommunikationsstruktur mit dem Funktionsbaustein MB_RegiSensor01CommType02

Es wird nur noch ein Funktionsbaustein MB_RegiSensor01CommType02 benötigt, sowie der Funktionsbaustein MB_RegiSensor01SetIPAddrType01. Die bisher vorgegebene Verschaltung sowie die Einschaltreihenfolge können entfallen.



Es wird aufgrund der fest vorgegebenen internen Verschaltung und der einfachen Bedienung empfohlen nur den Funktionsbaustein MB_RegiSensor01CommType02 für die Kommunikation mit dem Markenstromsensor ab Sensorfirmwareversion 12 zu verwenden.

Beispielhaft wird im nachfolgenden auch das manuelle Verschalten der Funktionsbausteine für die Sensorfirmware ≤ 11 beschrieben. Dieses Vorgehen wird jedoch nicht empfohlen. Hierbei wird zwischen zyklischer und azyklischer Kommunikation unterschieden. Azyklische Kommunikation bedeutet, dass die Steuerung der Client ist und der Sensor der Server. D. h. die Steuerung fragt Daten an und der Sensor antwortet. Hierzu zählen Konfigurations-, Prozess- als auch Diagnosedaten.

Folgende Bausteine sind an der azyklischen Kommunikation beteiligt:

- MB_RegiSensor01ConfigType01: Konfiguration des Sensors, Starten der Messung, Teach-In, usw.
- MB_RegiSensor01SendSpeedAccType01: Sendet Geschwindigkeits- und Beschleunigungsinformationen
- MB_RegiSensor01SendRecvDataType01: Übernimmt die azyklische Kommunikation mit dem Sensor
- MB_RegiSensor01ReadDiagType01: Liest Diagnosedaten des Sensors aus

- MB_RegiSensor01TeachDataType01: Liest/Schreibt die zuletzt gültigen Messdaten

Jeder Sensor benötigt 2 Ports auf der Steuerung. Einen Port für die azyklische Kommunikation und einen Port für die zyklische Kommunikation. In der zyklischen Kommunikation sendet der Sensor nach jeder Zylinderumdrehung ein Datenpaket, das die in der Konfiguration definierten Messdaten enthält. Die Steuerung hört auf diesem Port mit und wertet die Messergebnisse aus.

Hierfür werden die folgenden Funktionsbaustein verwendet:

- MB_RegiSensor01CycComType01: Wertet die Markenabstände und Breiten zweier Marken zueinander aus.
- MB_RegiSensor01CycComType02: Wertet die Abstände aller Marken zum Nullimpuls sowie alle Markenbreiten aus.

Der Funktionsbaustein MB_RegiSensor01CycComType01 liefert genau den Markenabstand zwischen 2 angegebenen Marken und deren Differenz in der Markenbreite. Der Funktionsbaustein MB_RegiSensor01CycComType02 liefert alle Abstände konfigurierter Marken zum Nullimpuls sowie deren Markenbreiten. Die Funktionsbausteine können je nach Anwendungsfall gegeneinander ausgetauscht werden. Pro Sensor ergibt sich die nachfolgende Kommunikationsstruktur:

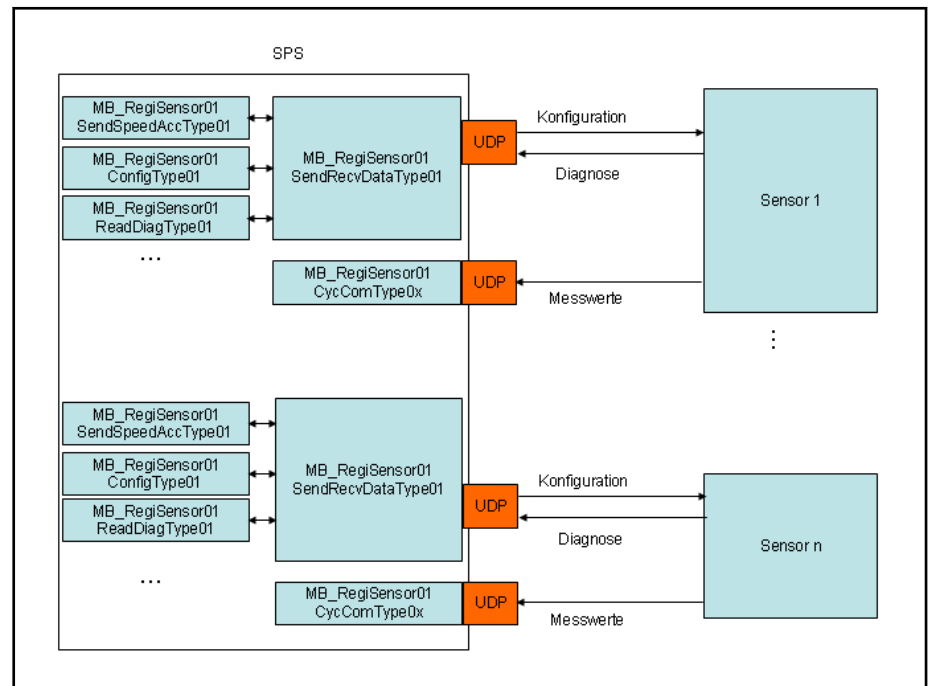


Abb. 3-7: Kommunikationsstruktur

Man erkennt somit, dass ein Funktionsbaustein die komplette azyklische Kommunikation übernimmt und alle Datenpakete der anderen Funktionsbausteine an den Sensor sendet. D. h. der Funktionsbaustein MB_RegiSensor01ConfigType01 benötigt zwingend den Funktionsbaustein MB_RegiSensor01SendRecvDataType01 zum Senden der Datenpakete.

Ein weiterer Funktionsbaustein in der Bibliothek RMB_TechRegiExtended, zum Setzen der IP-Adresse, ist der Funktionsbaustein:

- MB_RegiSensor01SetIPAddressType01

Hier wird nur ein Funktionsbaustein pro Maschine benötigt.

Nachfolgende Abbildung zeigt die Minimalverschaltung der Funktionsbausteine und deren Einschaltreihenfolge.

RMB_TechRegiExtended

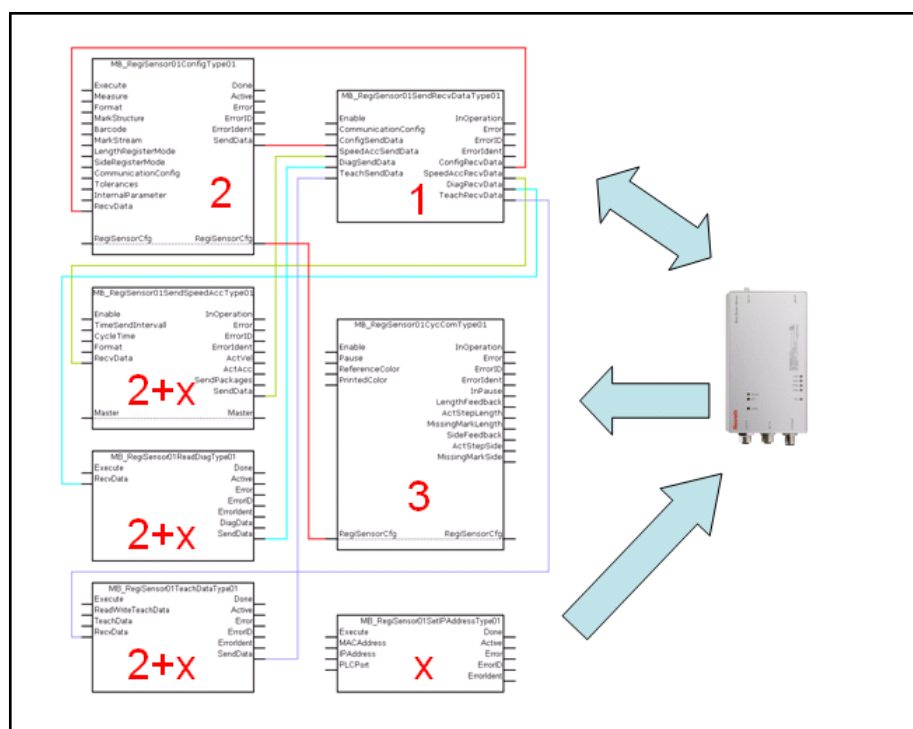


Abb. 3-8: Minimalverschaltung und Einschaltreihenfolge

Um mit dem Markenstromsensor eine Messung durchzuführen, muss als erstes der Funktionsbaustein MB_RegiSensor01SendRecvDataType01 aktiviert werden. Anschließend kann mit dem Funktionsbaustein MB_RegiSensor01ConfigType01 eine Konfiguration zum Sensor übertragen werden und das Teach-In und die Messung gestartet werden. Die Funktionsbausteine MB_RegiSensor01SendSpeedAccType01, MB_RegiSensor01ReadDiagType01 und MB_RegiSensor01TeachDataType01 sind nicht zwingend für eine Messung nötig. Sie können beliebig aktiviert werden, wenn sie benötigt werden, unter der Voraussetzung, dass der Funktionsbaustein MB_RegiSensor01SendRecvDataType01 bereits aktiviert wurde. Wurde die Messung beim Sensor gestartet, muss zum Empfangen der Messwerte anschließend einer der Funktionsbausteine MB_RegiSensor01CycComType01 oder MB_RegiSensor01CycComType02 aktiviert werden. Der Funktionsbaustein MB_RegiSensor01SetIPAddressType01 zum Setzen der IP-Adresse wird nur einmal pro Maschine benötigt und besitzt deshalb einen eigenen UDP-Port und kann zu jedem beliebigen Zeitpunkt aktiviert werden, ohne jede Abhängigkeit der anderen Funktionsbausteine. Die farbigen Linien kennzeichnen die minimale Verschaltung zwischen den Funktionsbausteinen, damit eine korrekte Kommunikation mit dem Sensor möglich ist.

3.3.2 MB_RegiSensor01CommType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01CommType01 enthält die bisher einzeln implementierten Funktionsbausteine:

- MB_RegiSensor01ConfigType01
- MB_RegiSensor01CycComType02
- MB_RegiSensor01ReadDiagType01
- MB_RegiSensor01SendRecvDataType01
- MB_RegiSensor01SendSpeedAccType01 und

- MB_RegiSensor01TeachDataType01

Der Funktionsbaustein hat intern eine feste Verschaltung der Funktionsbausteine und bietet eine einfache Bedienerchnittstelle zum Ausführen der einzelnen Funktionen. Mit dem Funktionsbaustein ist die volle Steuerung des Markenstromsensors möglich.



Der folgende Funktionsbaustein zum Setzen der IP-Adresse ist nicht integriert, da er nur einmal pro Maschine benötigt wird:

- MB_RegiSensor01SetIPAddressType01



Der Funktionsbaustein MB_RegiSensor01CommType01 kann nur zusammen mit der Sensorfirmware 11 VRS und kleiner verwendet werden.

Ab Sensorfirmware 12 muss der Funktionsbaustein MB_RegiSensor01CommType02 verwendet werden.



Der Funktionsbaustein MB_RegiSensor01CommType01 muss in einer Motion synchronen Task aufgerufen werden, damit die Beschleunigung korrekt berechnet wird. Ist dies aus Laufzeitgründen nicht möglich, so kann die Funktion mMotionCyclic() des Funktionsbausteins aus der Motion synchronen Task aufgerufen werden und der Funktionsbaustein MB_RegiSensor01CommType01 kann in einer zyklischen Task aufgerufen werden. Der Aufruf in der MotionTask muss folgende Struktur besitzen: {CyclicTask}.fbRegiSensor01CommType01.{fbInstanz}.mMotionCyclic(). Die Elemente in der geschweiften Klammer müssen durch den Namen des Programms in der zyklischen Task und durch den Instanznamen des Funktionsbausteins ersetzt werden. Somit könnte der Methodenaufruf aus einem Programm das in der Motion Task läuft wie folgt aussehen: PlcProg.fbRegiSensor01CommType01.fbRegiSensor01CommType01.mMotionCyclic(). Wird dieser Aufruf nicht durchgeführt, kann eine korrekte Berechnung der Beschleunigung nicht gewährleistet werden.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V12	RMB_TechRegiExtended

Tab. 3-2: Referenztabelle Funktionsbaustein MB RegiSensor01CommType01

Schnittstellenbeschreibung

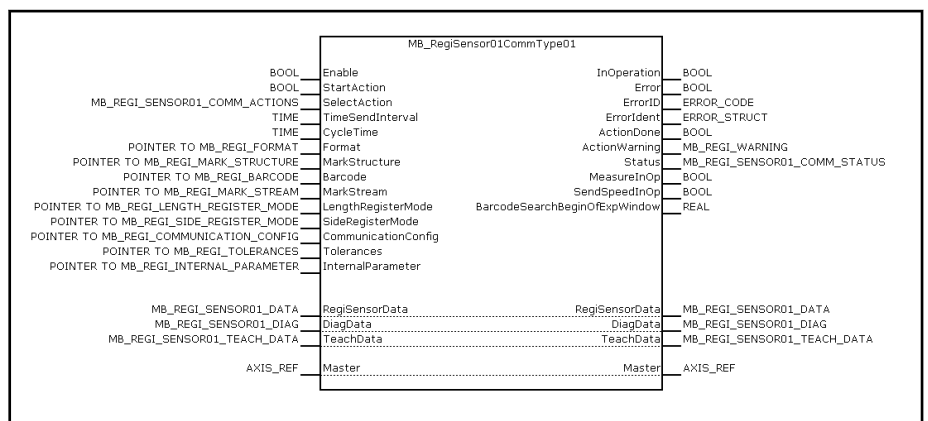


Abb. 3-9: Funktionsbaustein MB RegiSensor01CommType01

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT CONST	Master	AXIS_REF	Adressierung der Leitachse
VAR_IN_OUT	RegiSensor-Data	MB_REGI_SENSOR01_DATA	Enthält die entsprechenden Messwerte des Sensors
	DiagData	MB_REGI_SENSOR01_DIAG	Diagnosedaten des Sensors
	TeachData	MB_REGI_SENSOR01_TEACH_DATA	Zu schreibende oder gelesene Daten des Sensors
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	StartAction	BOOL	Startet die ausgewählte Aktion
	SelectAction	MB_REGI_SENSOR01_COMM_ACTIONS	Auswahl der durchführbaren Aktionen
	TimeSendInterval	TIME	In diesem Intervall werden die Daten der Geschwindigkeits- und Beschleunigungsinformation berechnet und versendet
	CycleTime	TIME	Zykluszeit in der der Funktionsbaustein aufgerufen wird.
	Format	POINTER TO MB_REGI_FORMAT	Definiert die Formateigenschaften
	MarkStructure	POINTER TO MB_REGI_MARK_STRUCTURE	Definiert die Markenstruktur
	Barcode	POINTER TO MB_REGI_BARCODE	Definiert den Barcode
	MarkStream	POINTER TO MB_REGI_MARK_STREAM	Definiert den Markenstrom
	LengthRegisterMode	POINTER TO MB_REGI_LENGTH_REGISTER_MODE	Definiert das Bahnmessverfahren für das Längsregister
	SideRegisterMode	POINTER TO MB_REGI_SIDE_REGISTER_MODE	Definiert das Bahnmessverfahren für das Seitenregister
	CommunicationConfig	POINTER TO MB_REGI_COMMUNICATION_CONFIG	Definiert die Kommunikationseigenschaften der UDP-Verbindung
	Tolerances	POINTER TO MB_REGI_TOLERANCES	Definiert die zulässigen Toleranzen für das Teach und das Messen
	InternalParameter	POINTER TO MB_REGI_INTERNAL_PARAMETER	Interne Parameter des Sensors, die nur auf ausdrücklichen Hinweis geändert werden sollen.

I/O-Typ	Name	Datentyp	Erläuterung
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	ActionDone	BOOL	Aktion wurde erfolgreich abgeschlossen
	ActionWarning	MB_REGI_WARNING	Enthält Warnungen von nicht erfolgreich durchgeführten Aktionen bzw. vom Sensorsystem, die nicht zu einem Fehler des Funktionsbausteins führen
	Status	MB_REGI_SENSOR01_COMM_STATUS	Gibt den aktuellen Status des Funktionsbausteins MB_RegiSensor01CommType01 an
	MeasureInOp	BOOL	Messung läuft
	SendSpeedInOp	BOOL	Geschwindigkeits- und Beschleunigungswerte werden übertragen.
	BarcodeSearchBeginOfExpWindow	REAL	Dieser Wert enthält die Position des Endes des Barcodes

Tab. 3-3: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01CommType01

Datenstrukturen

Die folgenden, verwendeten Datenstrukturen wurden bei den vorherigen Funktionsbausteinen der Bibliothek RMB_TechRegiExtended beschrieben:

- [Tab. 3-15 "Datenstruktur MB_REGI_FORMAT" auf Seite 42](#)
- [Tab. 3-16 "Datenstruktur MB_REGI_MARK_STRUCTURE" auf Seite 42](#)
- [Tab. 3-17 "Datenstruktur MB_REGI_BARCODE" auf Seite 42](#)
- [Tab. 3-18 "Datenstruktur MB_REGI_MARK_STREAM" auf Seite 42](#)
- [Tab. 3-19 "Datenstruktur MB_REGI_LENGTH_REGISTER_MODE" auf Seite 42](#)
- [Tab. 3-20 "Datenstruktur MB_REGI_SIDE_REGISTER_MODE" auf Seite 43](#)
- [Tab. 3-21 "Datenstruktur MB_REGI_COMMUNICATION_CONFIG" auf Seite 43](#)
- [Tab. 3-22 "Datenstruktur MB_REGI_TOLERANCES" auf Seite 43](#)
- [Tab. 3-23 "Datenstruktur MB_REGI_INTERNAL_PARAMETER" auf Seite 43](#)
- [Tab. 3-41 "Datenstruktur MB_REGI_SENSOR01_DATA" auf Seite 66](#)
- [Tab. 3-89 "Datenstruktur MB_REGI_SENSOR01_DIAG" auf Seite 114](#)
- [Tab. 3-98 "Datenstruktur MB_REGI_SENSOR01_TEACH_DATA" auf Seite 121](#)

RMB_TechRegiExtended

MB_REGI_WARNING		
Warning	BOOL	Zeigt an, das eine Warnung vorliegt
WarningAdditional1	DWORD	Zeigt den Additional 1 an.
WarningAdditional2	DWORD	Zeigt den Additional 2 an.

Tab. 3-4: Datenstruktur MB_REGI_WARNING

Aufzählungstyp

Der Aufzählungstyp MB_REGI_SENSOR01_COMM_STATUS kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
WAIT_FOR_NEXT_ACTION	0	Funktionsbaustein wartet auf nachfolgende Aktion
CONFIGURATION_IN_WORK	1	Konfiguration wird übertragen
BARCODE_SEARCH_IN_WORK	2	Barcode Suche läuft
MARKSTREAM_SEARCH_IN_WORK	3	Markenstrom Suche läuft
TEACH_IN_IN_WORK	4	TeachIn läuft
EASY_TEACH_IN_IN_WORK	5	EasyTeachIn läuft
READ_DIAG_IN_WORK	6	ReadDiagData läuft
READ_TEACH_DATA_IN_WORK	7	ReadTeachData läuft
WRITE_TEACH_DATA_IN_WORK	8	WriteTeachData läuft
MEASUREMENT_STARTING	9	Messung wird gestartet
MEASUREMENT_STOPPING	10	Messung wird gestoppt
SEND_SPEED_STARTING	11	SendSpeed wird gestartet
SEND_SPEED_STOPPING	12	SendSpeed wird gestoppt
SENSOR_OFF	13	Der Funktionsbaustein ist nicht im Zustand InOperation.
SENSOR_ERROR	14	Sensor im Error Zustand

Tab. 3-5: Aufzählungstyp MB_REGI_SENSOR01_COMM_STATUS

Der Aufzählungstyp MB_REGI_SENSOR01_COMM_ACTIONS kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
SET_CONFIG	1	Übertragen der Konfiguration starten
SEARCH_BARCODE	2	Barcode Suche starten
SEARCH_MARK_STREAM	3	Markenstrom Suche starten
TEACH_IN	4	TeachIn starten
EASY_TEACH_IN	5	Easy TeachIn starten
READ_DIAG	6	Diagnosedaten lesen
READ_TEACHING_DATA	7	Teach Daten lesen
WRITE_TEACHING_DATA	8	Teach Daten schreiben
START_MEASURING	9	Messung starten

Element	Wert	Beschreibung
STOP_MEASURING	10	Messung stoppen
START_SEND_SPEED	11	Geschwindigkeitsinformationen senden starten
STOP_SEND_SPEED	12	Geschwindigkeitsinformationen senden stoppen

Tab. 3-6: Aufzählungstyp MB_REGI_SENSOR01_COMM_ACTIONS

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
StartAction	BOOL			FALSE	Kontinuierlich
SelectAction	MB_REGI_SENSOR01_COMM_ACTIONS	1	12	1 (SET_CONFIG)	Bei steigender Flanke an StartAction und Status = WAIT_FOR_NEXT_ACTION
TimeSendInterval	TIME	$\geq 2 \cdot \text{CycleTime}$	n. def.	t#32ms	Aktion START_SEND_SPEED wird gestartet
CycleTime	TIME	> 0	n. def.	t#8ms	Aktion START_SEND_SPEED wird gestartet
Format	POINTER TO MB_REGI_FORMAT	> 0	n. def.	0	Steigende Flanke an Enable
MarkStructure	POINTER TO MB_REGI_MARK_STRUCTURE	> 0	n. def.	0	Steigende Flanke an Enable
Barcode	POINTER TO MB_REGI_BARCODE	> 0	n. def.	0	Steigende Flanke an Enable
MarkStream	POINTER TO MB_REGI_MARK_STREAM	> 0	n. def.	0	Steigende Flanke an Enable
LengthRegisterMode	POINTER TO MB_REGI_LENGTH_REGISTER_MODE	> 0	n. def.	0	Steigende Flanke an Enable
SideRegisterMode	POINTER TO MB_REGI_SIDE_REGISTER_MODE	> 0	n. def.	0	Steigende Flanke an Enable
Communication-Config	POINTER TO MB_REGI_COMMUNICATION_CONFIG	> 0	n. def.	0	Steigende Flanke an Enable
Tolerances	POINTER TO MB_REGI_TOLERANCES	> 0	n. def.	0	Steigende Flanke an Enable
InternalParameter	POINTER TO MB_REGI_INTERNAL_PARAMETER	> 0	n. def.	0	Steigende Flanke an Enable

Tab. 3-7: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Es wird bei einer steigenden Flanke an Enable überprüft, ob alle Pointer > 0 sind und somit die entsprechenden Adressen, unter denen die Datenstrukturen liegen, angelegt sind. Die Daten der angelegten Pointer werden bei den Aufrufen der einzelnen Funktionsbausteine überprüft. Die Datenstrukturen besitzen die folgenden Standardwerte und werden bei den folgenden Aktionen jeweils übernommen bzw. geprüft:

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
MB_REGI_FORMAT					
FormatLength	REAL	> 0	n. def.	0	Steigende Flanke an Enable
Increments	UDINT	> 0	n. def.	0	Steigende Flanke an Enable
MB_REGI_MARK_STRUCTURE					
MarkType	UDINT	1	7	1 (Es sind nur Marken 1,3,5 implementiert)	Aktion SET_CONFIG wird gestartet
SingleMark	ARRAY [0...3] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_BARCODE					
NumberOfBlocks	UDINT	0 (1 und 2 Blocks sind ungültige Eingaben)	10	0	Aktion SET_CONFIG wird gestartet
BeginExpectationWindow	REAL	0	n. def.	0	Aktion SET_CONFIG wird gestartet
SizeOfExpectationWindow	REAL	0	n. def.	1	Aktion SET_CONFIG wird gestartet
FreeSpaceBeforeBarcode	REAL	0	n. def.	0	Aktion SET_CONFIG wird gestartet
BarcodeBlocks	ARRAY [0...19] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_MARK_STREAM					
NumberOfMarks	UDINT	1	12	1	Aktion SET_CONFIG wird gestartet
MarkStream	Array [1...12] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Aktion SET_CONFIG wird gestartet
MarkDirection	BOOL			FALSE	Aktion SET_CONFIG wird gestartet
MB_REGI_LENGTH_REGISTER_MODE					
ModeSelect	UDINT	1	1	1	Aktion SET_CONFIG wird gestartet
LengthProperties	ARRAY [0...4] OF REAL	n. def.	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_SIDE_REGISTER_MODE					
ModeSelect	UDINT	0	1	1	Aktion SET_CONFIG wird gestartet
SelectMarks	ARRAY [0...1] OF UINT	n. def.	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_COMMUNICATION_CONFIG					
IpSensor	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
PortConfigurati-on	UINT	1	n. def.	0	Steigende Flanke an Enable
IpPLC	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortMeasure-ment	UINT	1	n. def.	0	Steigende Flanke an Enable
MB_REGI_TOLERANCES					
TeachZeroPul-seToMark	REAL	1	n. def.	5	Aktion SET_CONFIG wird ge-startet
TeachMarkTo-Mark	REAL	1	n. def.	2	Aktion SET_CONFIG wird ge-startet
MeasureMark-Tolerance	REAL	1	n. def.	10	Aktion SET_CONFIG wird ge-startet
DeviationOf-MarkWidth	REAL	1	100	20	Aktion SET_CONFIG wird ge-startet
DeviationOf-MarkHight	REAL	1	100	100	Aktion SET_CONFIG wird ge-startet
DeviationOfLast-Position	REAL	1	n. def.	10	Aktion SET_CONFIG wird ge-startet
ShiftSyncPulse	REAL	1	n. def.	5	Aktion SET_CONFIG wird ge-startet
MB_REGI_INTERNAL_PARAMETER					
ColorWeighting	REAL	0	< 1	0.9999	Aktion SET_CONFIG wird ge-startet
IntensityWeigh-ting	REAL	0	< 1	0.15	Aktion SET_CONFIG wird ge-startet
Gain	UINT	0	7	0	Aktion SET_CONFIG wird ge-startet

Tab. 3-8: Übersicht Min-/Max und Default-Werte der referenzierten Strukturen

Die VAR_IN_OUTs RegiSensorData und DiagData besitzen keine Min-/Max und Default-Werte, da sie nur von den internen Funktionsbausteinen beschrieben werden und nicht vom Anwender beschrieben werden dürfen. In TeachData werden beim Lesen der Teach Daten die entsprechenden Daten abgelegt. Beim Schreiben der Teach Daten werden die, z. B. aus einer vorherig abgespeicherten Produktion, angelegten Teach Daten zum Sensor geschrieben. Diese Datenstruktur ist nicht interpretierbar.

Funktionsbeschreibung

Für die Funktionsbeschreibung wird ebenfalls empfohlen, die einzelnen Beschreibungen der folgenden Funktionsbausteine zu beachten:

- [Kap. 3.3.3 "MB_RegiSensor01ConfigType01" auf Seite 40](#)
- [Kap. 3.3.6 "MB_RegiSensor01CycComType02" auf Seite 64](#)
- [Kap. 3.3.11 "MB_RegiSensor01ReadDiagType01" auf Seite 112](#)
- [Kap. 3.3.4 "MB_RegiSensor01SendRecvDataType01" auf Seite 56](#)
- [Kap. 3.3.12 "MB_RegiSensor01SendSpeedAccType01" auf Seite 115](#)
- [Kap. 3.3.13 "MB_RegiSensor01TeachDataType01" auf Seite 119](#)

RMB_TechRegiExtended

Initialisierung Bei einer steigenden Flanke an Enable werden zunächst die Pointer geprüft, ob sie angelegt wurden. Die Inhalte der Pointer der jeweiligen Datenstruktur werden direkt an die jeweiligen Funktionsbausteine angelegt und erst dann geprüft, wenn der zugehörige Funktionsbaustein aufgerufen wird. Anschließend wird der Funktionsbaustein MB_RegiSensor01SendRecvDataType gestartet, der die Kommunikation mit dem Markenstromsensor durchführt. Hier wird bereits durch das Kommando "Ping" überprüft ob der Markenstromsensor unter der vorgegebenen IP-Adresse zu erreichen ist. Im Anschluss wird die aktuell am Funktionsbaustein angelegte Konfiguration mittels des Funktionsbausteins MB_RegiSensor01ConfigType01 übertragen und der Funktionsbaustein MB_RegiSensor01CycComType02 zum Empfangen der Messwerte gestartet. Anschließend ist die Initialisierungsphase abgeschlossen und InOperation wird TRUE. Jeder in der Initialisierungsphase aufgetretene Fehler führt sofort zu einem Übergang in den Fehlerzustand und Error = TRUE gesetzt. Im zyklischen Betrieb wird zwischen Fehlern und Warnungen unterschieden. Während der Initialisierung befindet sich der Sensor im Zustand SENSOR_OFF.

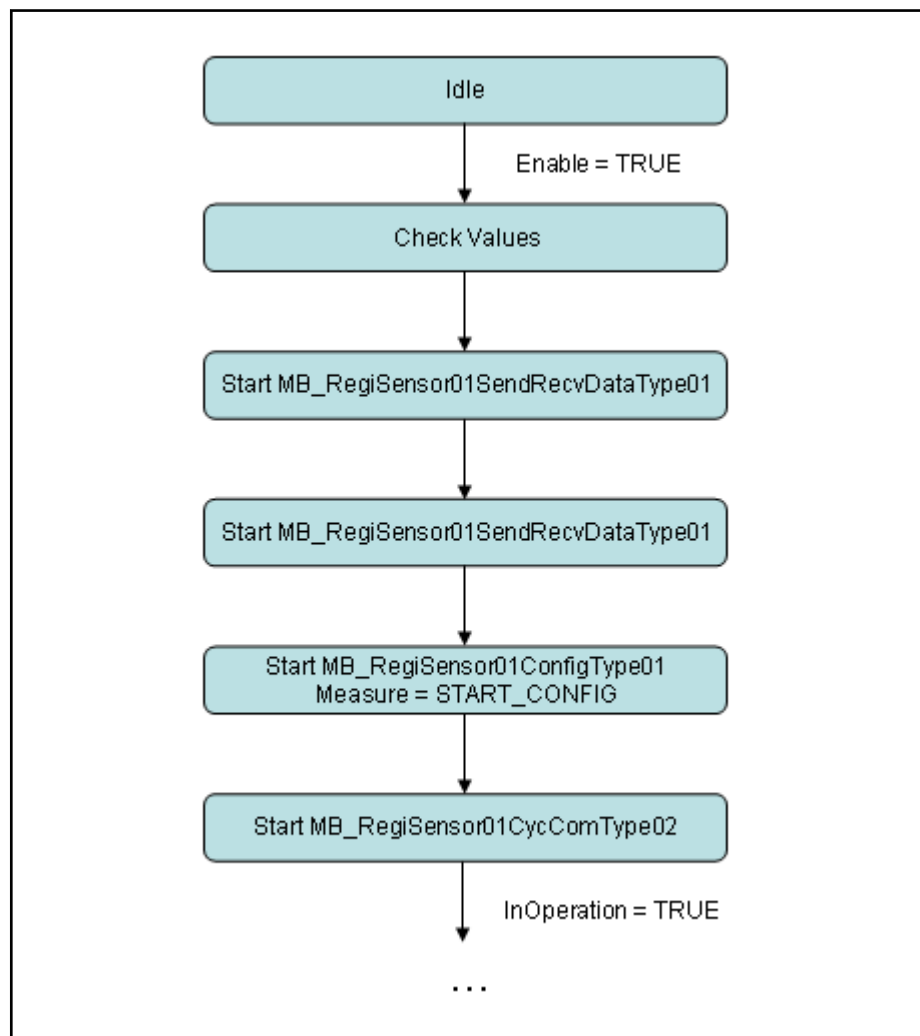


Abb. 3-10: Ablauf Initialisierungsphase



Es wird während der Initialisierung geprüft, ob auf dem Markenstromsensor noch eine Messung läuft. Ist dies der Fall wird diese Messung automatisch gestoppt.

Zyklischer Betrieb

Folgende Zustände können im zyklischen Betrieb erreicht werden:

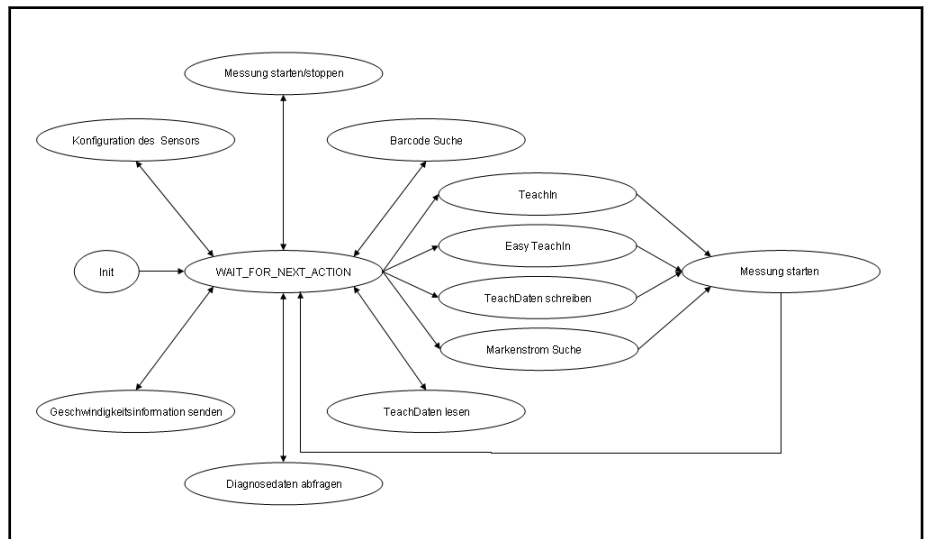


Abb. 3-11: Zustandsübergänge im zyklischen Betrieb

Im zyklischen Betrieb (d. h. InOperation = TRUE) befindet sich der Funktionsbaustein in einem Zustand "WAIT_FOR_NEXT_ACTION". In diesem Zustand können alle im Aufzählungstyp "MB_REGI_SENSOR01_COMM_ACTIONS" definierten Aktionen ausgeführt werden. Die entsprechende Aktion muss in SelectAction ausgewählt und anschließend mit einer steigenden Flanke an StartAction gestartet werden. Während eine Aktion läuft, kann über Status der entsprechende Zustand des Funktionsbausteins ausgelesen werden. Wird eine Aktion erfolgreich beendet, so wird dies über ActionDone angezeigt. Läuft bereits eine Aktion, so kann keine neue Aktion gestartet werden. Eine Aktion kann durch 3 unterschiedliche Ereignisse beendet werden. Wurde die Aktion erfolgreich ausgeführt, so wird dies mit ActionDone angezeigt. Ist ein schwerwiegender Kommunikationsfehler aufgetreten, so geht der Funktionsbaustein in den Error Zustand (Error = TRUE). Wurde eine Aktion nicht erfolgreich ausgeführt (z. B. wurden beim TeachIn nicht alle Marken gefunden), so gibt der Funktionsbaustein eine Warning (ActionWarning = TRUE) aus und kehrt in den Zustand "WAIT_FOR_NEXT_ACTION" zurück. Beim Starten einer erneuten Aktion wird die Warnung automatisch zurückgesetzt.

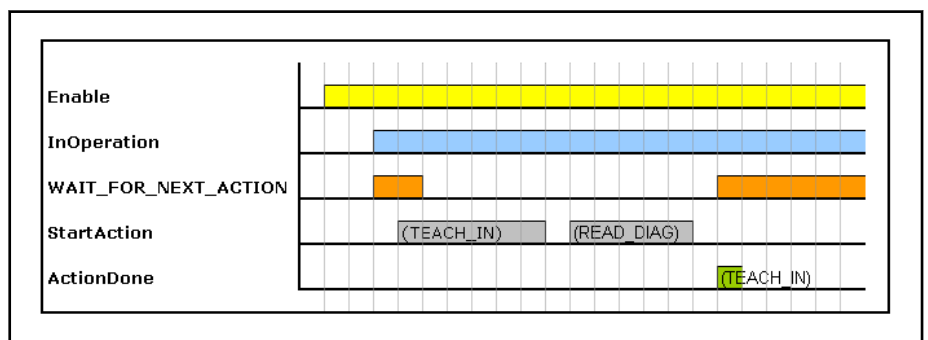


Abb. 3-12: Zeitdiagramm im zyklischen Betrieb

Das Zeitdiagramm zeigt, dass nach einer steigenden Flanke an Enable nach einigen Zyklen, das InOperation gesetzt und in den Zustand "WAIT_FOR_NEXT_ACTION" übergegangen wird. Eine steigende Flanke an StartAction mit SelectAction := TEACH_IN führt dazu, dass der Zustand "WAIT_FOR_NEXT_ACTION" verlassen und das Teachen durchgeführt wird. Eine erneute steigende Flanke an StartAction mit SelectAction := READ DI-

AG wird dagegen ignoriert. Der Funktionsbaustein befindet sich zu diesem Zeitpunkt nicht im Status "WAIT_FOR_NEXT_ACTION", da die Aktion des Teachens bereits ausgeführt wird. Nach dem erfolgreichen Abschluss des Teachens wird ActionDone = TRUE für einen Zyklus gesetzt (da die steigende Flanke an StartAction bereits weggenommen wurde). Der Funktionsbaustein kehrt zurück in den Zustand "WAIT_FOR_NEXT_ACTION".

Eine ausgeführte Aktion wird genau dann erfolgreich beendet, wenn das dazugehörige Acknowledgement ActionDone = TRUE kommt. Es ist nicht möglich, mehrere Operationen zeitgleich durchzuführen. Wurde die steigende Flanke am Eingang bereits weggenommen, so steht ActionDone am Ausgang genau 1 Zyklus an (siehe folgendes Zeitdiagramm).

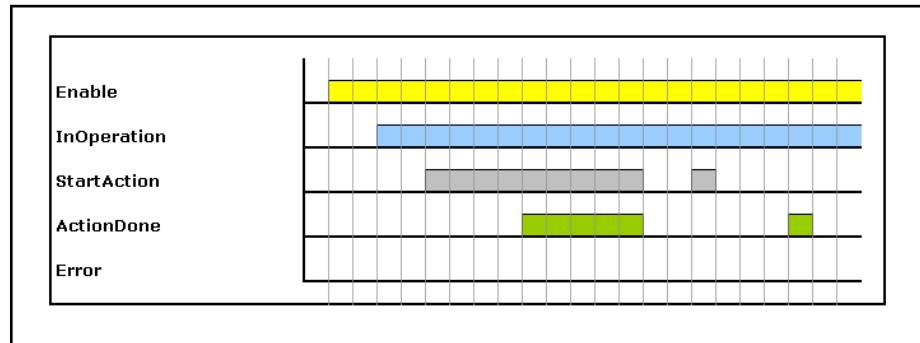


Abb. 3-13: Zeitdiagramm des Verhaltens von ActionDone

Da einige Aktionen (z. B. TeachIn) einige Zeit in Anspruch nehmen, besitzt der Funktionsbaustein den Ausgang "Status". Ist Status = 0 und InOperation = TRUE, so befindet sich der Funktionsbaustein im Zustand "WAIT_FOR_NEXT_ACTION". Hier kann die nächste Aktion ausgeführt werden. Der Status kann die Werte des Aufzählungstyps "MB_REGI_SENSOR01_COMM_STATUS" annehmen. Ist der Funktionsbaustein nicht in InOperation = TRUE, so ist der Status "SENSOR_OFF" außer er befindet sich im Fehlerzustand, dann ist der Status "SENSOR_ERROR".

Nach dem Starten oder Stoppen der Messung oder nach dem Starten oder Stoppen des Sendens der Geschwindigkeitsinformation wird der aktuelle Status an den Ausgängen MeasureInOp und SendSpeedInOp angezeigt. Somit wird kontinuierlich angezeigt, dass die Messung und das Übertragen der Geschwindigkeitsinformation laufen.

Vor dem Ausführen der folgenden Aktionen wird zunächst überprüft, ob eine Messung läuft:

- SET_CONFIG
- SEARCH_BARCODE
- SEARCH_MARK_STREAM
- TEACH_IN
- EASY_TEACH_IN

Ist dies der Fall wird die Messung zunächst gestoppt und anschließend die Aktion ausgeführt.

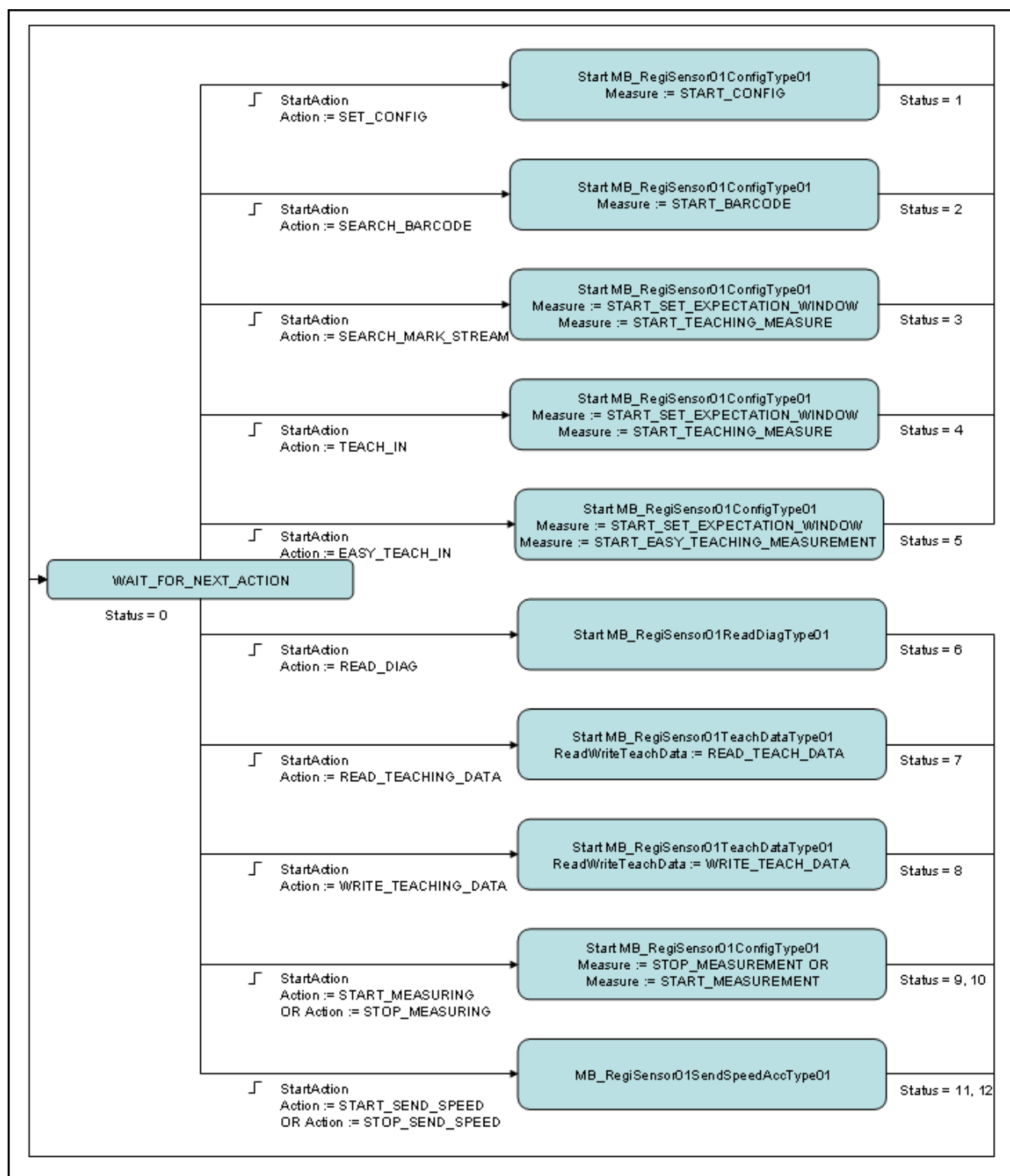


Abb. 3-14: Ablauf des zyklischen Betriebs

Die Abbildung zeigt den angedeuteten Ablauf des zyklischen Betriebs. Hier ist zu erkennen, dass ausgehend vom Zustand "WAIT_FOR_NEXT_ACTION" in die jeweiligen Aktionen gesprungen wird. Nach Abschluss der Aktion folgt wiederum die Rückkehr in den Ausgangszustand. Es kann natürlich (nicht eingezeichnet) auch in den Fehlerzustand gesprungen werden. Es wurden

RMB_TechRegiExtended

jeweils die für diese Aktionen grundlegenden Eingänge angezeichnet. In den Zuständen, können mehrere Funktionsbausteine aufgerufen werden. Beispielsweise wird bei einer steigenden Flanke an StartAction (SelectAction := TEACH_IN) zunächst der Funktionsbaustein MB_RegiSensor01Config-Type01 zum Setzen des Erwartungsfensters aufgerufen, um ihn anschließend nochmals zum Teachen und Starten der Messung aufzurufen.

SEARCH_MARK_STREAM

Diese Funktion sucht sich selbständig den Markenstrom ohne Barcode Suche und ohne Angabe des Erwartungsfensters.

Folgende Voraussetzungen müssen für diese Funktion erfüllt sein:

- Auf der 360° Umdrehung die der Markenstromsensor abtastet befinden sich ausschließlich die Registermarken. D. h. es dürfen sich weder weitere Passerkreuze oder sonstige Kennzeichnungen in der Abtastspur des Sensors befinden.
- Der Registermarkenstrom darf maximal 2/3 des Druckzylinderformats ausmachen. Mindestens 1/3 des Druckzylinderformats müssen somit druckfreier Bereich sein. Beispiel: Befinden sich auf dem zu vermessen- den Format 12 Registermarken im Abstand von 20 mm (Markenstrom der Größe 220 mm), so muss das Druckzylinderformat mindestens 330 mm betragen.

Ist eine der genannten Voraussetzungen verletzt, so kann die Funktion nicht ausgeführt werden oder wird mit einem Fehler beendet.



Die Funktion sucht sich mit maximal 3 Teach-Vorgängen selbständig den Markenstrom. Es muss nur eine gültige Konfiguration übertragen worden sein. Es ist keine Angabe des Erwartungsfensters nötig.

Verwendete Daten/Datenstrukturen

Nachdem nicht alle Eingänge bei jeder Aktion benötigt werden, führt die nachfolgende Tabelle die Daten auf, die bei den einzelnen Aktionen übernommen werden:

SelectAction	Verwendete Daten	Durch die Aktion veränderte Daten
SET_CONFIG	Format MarkStructure Barcode MarkStream LengthRegisterMode SideRegisterMode CommunicationConfig Tolerances InternalParameter	ActionDone Status
SEARCH_BARCODE	Barcode.NumberOfBlocks Barcode.FreeSpaceBeforeBarcode Barcode.BarcodeBlocks InternalParameter.Gain	ActionDone Status BarcodeSearchBeginOfExpWindow
SEARCH_MARK_STREAM	-	ActionDone Status

SelectAction	Verwendete Daten	Durch die Aktion veränderte Daten
TEACH_IN	Barcode.BeginExpectationWindow Barcode.SizeOfExpectationWindow	ActionDone Status RegiSensorData
EASY_TEACH_IN	Barcode.BeginExpectationWindow Barcode.SizeOfExpectationWindow InternalParameter.Gain	ActionDone Status RegiSensorData
READ_DIAG	-	ActionDone Status DiagData
READ_TEACHING_DATA	-	ActionDone Status TeachData
WRITE_TEACHING_DATA	TeachData	ActionDone Status
START_MEASURING	-	ActionDone MeasureInOp Status RegiSensorData
STOP_MEASURING	-	ActionDone MeasureInOp Status RegiSensorData
START_SEND_SPEED	TimeSendIntervall CycleTime Master	ActionDone SendSpeedInOp Status
STOP_SEND_SPEED	-	ActionDone SendSpeedInOp Status

Tab. 3-9: Datenabhängigkeiten

Beispielhafte Abläufe

RMB_TechRegiExtended

In nachfolgender Abbildung wird der beispielhafte Ablauf von der Konfiguration zur Messung dargestellt für die Instanz fbRegiSensor01Comm des Funktionsbausteins MB_RegiSensor01CommType01:

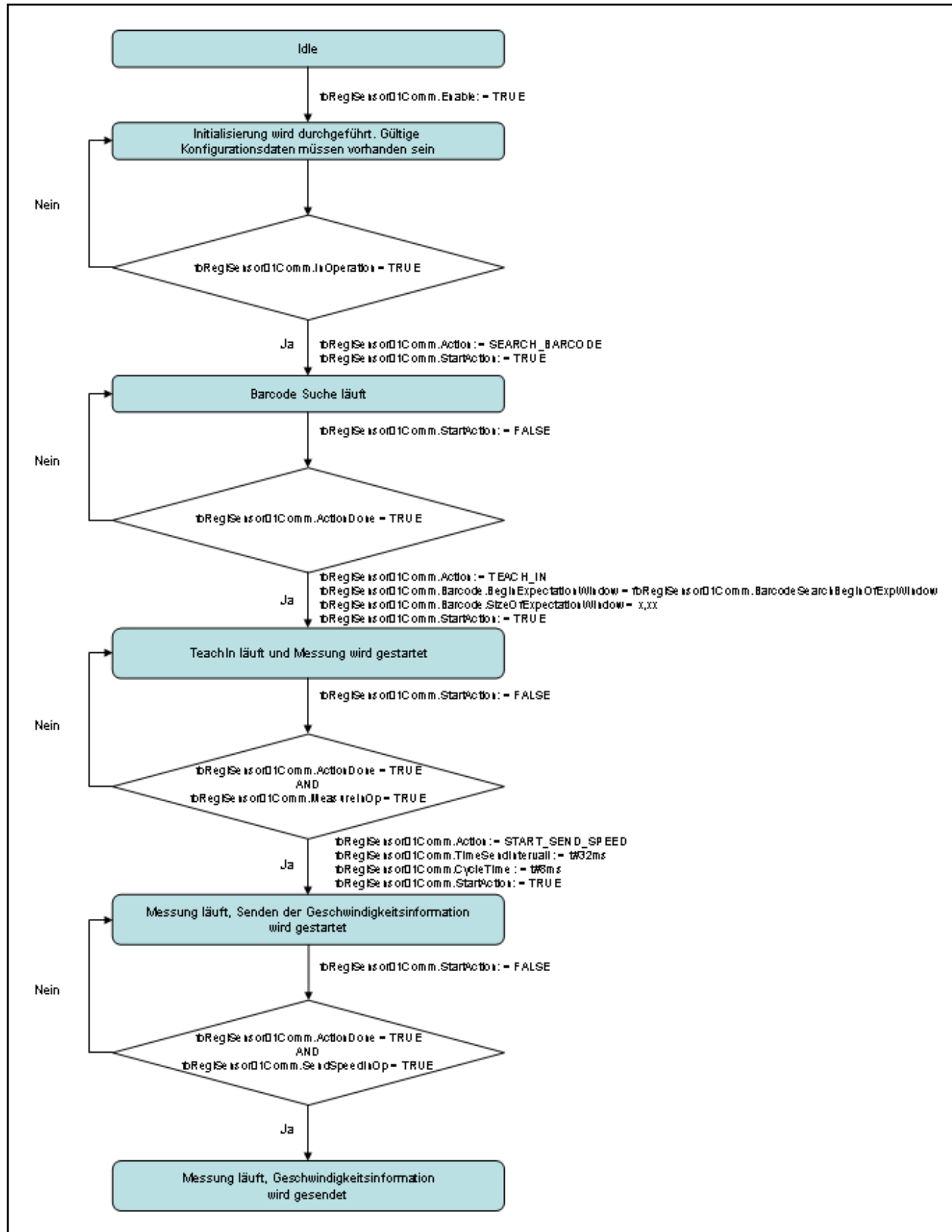


Abb. 3-15: Beispielhafter Ablauf von der Konfiguration zur Messung

Zunächst wird eine steigende Flanke an Enable der Instanz fbRegiSensor01Comm gesetzt. Hier stellt zunächst der Funktionsbaustein MB_RegiSensor01SendRecvDataType01 die Kommunikationsverbindung zum Sensor her. Anschließend wird die Konfiguration mit dem Funktionsbaustein MB_RegiSensor01ConfigType01 übertragen. Dies bedeutet es müssen gültige Daten anliegen. Weiterhin wird der Messbaustein MB_RegiSensor01CycComType02 gestartet und InOperation wird "TRUE". Daraufhin befindet sich der Funktionsbaustein im zyklischen Betrieb. Es wird die Suche des Barcodes gestartet. Nach dem Abschluss der Suche muss der zurückgelieferte Wert als Beginn des Erwartungsfensters zugewiesen werden und ein gültiger Wert für die Größe des Erwartungsfensters vorliegen. Es wird nun der TeachIn gestartet, bei dem zunächst das Erwartungsfenster gesetzt, dann das Teachen gestartet wird um anschließend die Messung zu beginnen. Um eine höhere Genauigkeit in der Beschleunigungsphase zu erhalten wird anschließend noch das Übertragen der Geschwindigkeitsinformation mit dem Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 aktiviert. Daraufhin läuft die Messung und die Messdaten befinden sich in der Datenstruktur RegiSensorData.

Besonderheiten Warning und Error

Der Funktionsbaustein besitzt neben dem standardisierten Fehlerhandling eine zweite Diagnoseschnittstelle "MB_REGI_WARNING" mit dem Warnungen ausgegeben werden können. Die Notwendigkeit dieser zweiten Diagnosestruktur ergibt sich durch die Verschaltung zahlreicher Funktionsbausteine und dem angebunden externen Rexroth Mark Stream Sensor. Es findet somit eine Unterscheidung zwischen schwerwiegenden Fehlern (Error = TRUE und Errorzustand) und Fehlern die zwar von einem internen Funktionsbaustein ausgegeben werden, diese jedoch nicht den gesamten Funktionsbaustein MB_RegiSensor01CommType01 zurücksetzen sollen. Beispielsweise können bei einem TeachIn nicht alle Marken gefunden werden. Dies wird als Warnung ausgegeben, da dies kein schwerwiegender Fehler ist und der Anwender eine weitere Aktion daraufhin ausführen kann.

Eine Warnung wird wie folgt definiert:

- Warnungen werden nur im zyklischen Betrieb erzeugt.
- Eine Warnung wird gesetzt nachdem eine Aktion nicht erfolgreich ausgeführt wird.
- Es erfolgt keine Bestätigung durch ActionDone.
- Der Funktionsbaustein kehrt in den Zustand "WAIT_FOR_NEXT_ACTION" zurück.
- Die Warnung wird automatisch zurückgesetzt wenn eine erneute Aktion ausgeführt wird.
- In WarningAdditional1 steht das Additional 1 der Fehlerstruktur "ERROR_STRUCT" und in WarningAdditional2 steht das Additional 2 der Fehlerstruktur "ERROR_STRUCT" des internen Funktionsbausteins.

RMB_TechRegiExtended

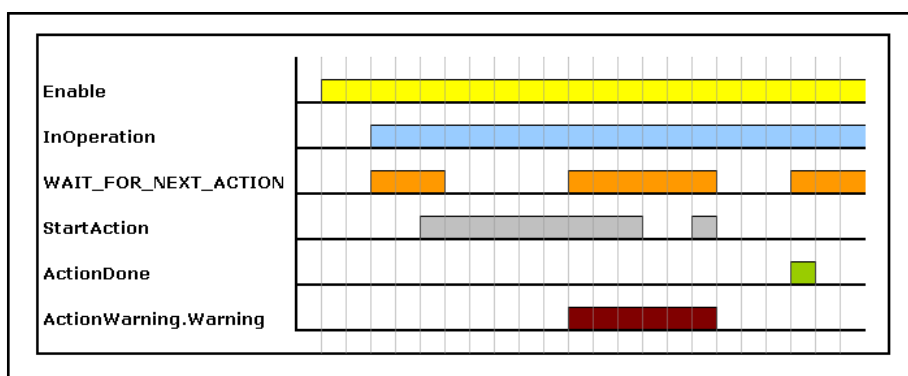


Abb. 3-16: Zeitverhalten der Warnung

Das Diagramm zeigt das zeitliche Verhalten beim Auftreten einer Warnung. Es wird eine Aktion ausgeführt im zyklischen Betrieb (InOperation=TRUE). Diese wird nicht erfolgreich beendet. Deshalb steht eine Warnung an. Der Funktionsbaustein kehrt zeitgleich mit der Warnung in den Zustand "WAIT_FOR_NEXT_ACTION" zurück. Nach dem Starten einer neuen Aktion wird die Warnung automatisch zurückgesetzt. Diese Aktion endet nun erfolgreich und wird mit ActionDone bestätigt.

Warning und Error bei der Initialisierung

Während der Initialisierung führt jeder von einem internen Funktionsbaustein erzeugte Fehler zu einem Fehler des Funktionsbausteins MB_RegiSensor01CommType01. D. h. es muss sowohl die Verbindung mit dem Sensor erneut hergestellt werden, als auch eine gültige Konfiguration übertragen und der Messbaustein gestartet werden.

Warning und Error im zyklischen Betrieb

Im zyklischen Betrieb führen ausschließlich Verbindungsfehler zu einem Error, da dies schwerwiegende Fehler sind. Hierzu zählen alle Fehler der Funktionsbausteine MB_RegiSensor01SendRecvDataType01 und MB_RegiSensor01CycComType02.

Weiterhin führen die folgenden Fehlernummern der anderen intern verwendeten Funktionsbausteine zu einem Übergang in den Error-Zustand:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000000	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvDataType01 innerhalb von 10 Sekunden
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000005	Timeout beim Empfangen eines Pakets.
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000006	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvDataType01 innerhalb von 10 Sekunden

Tab. 3-10: Fehler der internen Funktionsbausteine, die in den Fehlerzustand führen

Alle anderen internen Fehler der Funktionsbausteine werden als Warnungen interpretiert. Tritt eine Warnung auf, wird der Additional1 und Additional2 der "ERROR_STRUCT" des internen Funktionsbausteins auf WarningAdditional1 und WarningAdditional2 kopiert sowie das Warning Bit gesetzt. Intern wird daraufhin der Funktionsbaustein bei dem der Fehler auftritt zurückgesetzt. Bei einem erneuten Ausführen einer Aktion erlischt die Warnung automatisch und WarningAdditional1 und WarningAdditional2 werden auf 0 zurückgesetzt.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000001	Suche des Markenstroms fehlgeschlagen
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000002	Messung ist schon gestartet
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000003	Messung ist schon gestoppt
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000004	Geschwindigkeitsinformationen senden bereits gestartet
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000005	Geschwindigkeitsinformationen senden bereits gestoppt
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003C	Nullpointer Format
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003D	Nullpointer MarkStructure
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003E	Nullpointer Barcode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003F	Nullpointer MarkStream
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000040	Nullpointer LengthRegisterMode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000041	Nullpointer SideRegisterMode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000042	Nullpointer CommunicationConfig
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000043	Nullpointer Tolerances
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000044	Nullpointer InternalParameter

Tab. 3-11: Fehlercodes des Funktionsbausteins MB_RegiSensor01Comm-Type01

Weiterhin liefert der Funktionsbaustein Fehlermeldungen der integrierten Funktionsbausteine:

- MB_RegiSensor01ConfigType01 (Kap. "Fehlerbehandlung" auf Seite 55)
- MB_RegiSensor01CycComType02 (Kap. "Fehlerbehandlung" auf Seite 68)
- MB_RegiSensor01ReadDiagType01 (Kap. "Fehlerbehandlung" auf Seite 115)
- MB_RegiSensor01SendRecvDataType01 (Kap. "Fehlerbehandlung" auf Seite 60)
- MB_RegiSensor01SendSpeedAccType01 (Kap. "Fehlerbehandlung" auf Seite 119)
- MB_RegiSensor01TeachDataType01 (Kap. "Fehlerbehandlung" auf Seite 121)

Die entsprechenden Fehlermeldungen werden erst bei einer Aktivierung des jeweiligen Funktionsbausteins erzeugt. Hierzu muss die Übernahme der jeweiligen Werte beachtet werden. Es können somit Fehlermeldungen erst bei aktivem InOperation erzeugt werden. Weiterhin können diese Fehler, wie zuvor beschrieben, ebenfalls als Warnungen ausgegeben werden.

RMB_TechRegiExtended

3.3.3 MB_RegiSensor01ConfigType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01ConfigType01 wird für die Konfiguration des Markenstromsensors benötigt. Der Funktionsbaustein bereitet die Konfigurationsdaten auf und übergibt die Daten an den Funktionsbaustein MB_RegiSensor01SendRecvDataType01, der die Versendung des Pakets und den Empfang des Antwortpakets durchführt. Neben der Konfiguration kann das Erwartungsfenster gesetzt, der Teach-In und die Messung gestartet werden oder die Messung gestoppt werden. Er führt die Erstkonfiguration des Sensors durch und konfiguriert den Sensor während der Messung um.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-12: Referenztabelle Funktionsbaustein MB_RegiSensor01ConfigType01

Schnittstellenbeschreibung

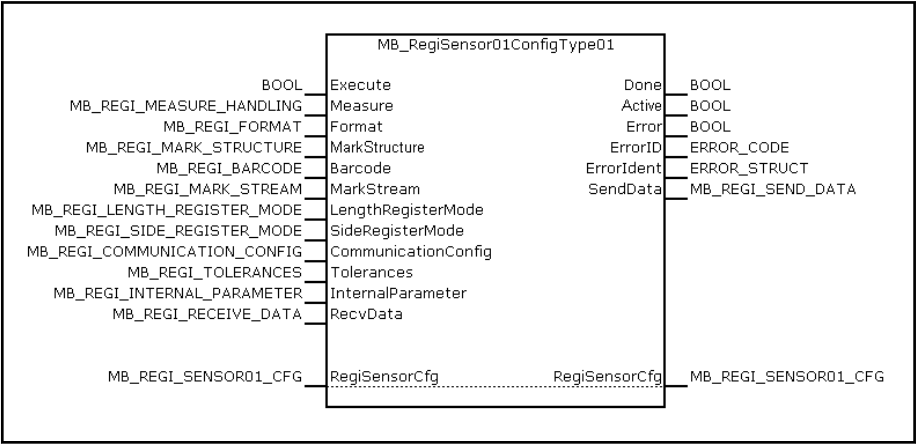


Abb. 3-17: Funktionsbaustein MB_RegiSensor01ConfigType01

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT	RegiSensorCfg	MB_REGI_SENSOR01_CFG	Enthält die aktuellen Konfigurationsdaten des Sensors
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	Measure	MB_REGI_MEASURE_HANDLING	Überträgt die Konfiguration, setzt das Erwartungsfenster, startet das Teach-In und die Messung oder stoppt die Messung.
	Format	MB_REGI_FORMAT	Definiert die Formateigenschaften
	MarkStructure	MB_REGI_MARK_STRUCTURE	Definiert die Markenstruktur
	Barcode	MB_REGI_BARCODE	Definiert den Barcode
	MarkStream	MB_REGI_MARK_STREAM	Definiert den Markenstrom
	LengthRegister-Mode	MB_REGI_LENGTH_REGISTER_MODE	Definiert das Bahnmessverfahren für das Längsregister
	SideRegisterMode	MB_REGI_SIDE_REGISTER_MODE	Definiert das verwendete Seitenregistermessverfahren
	Communication-Config	MB_REGI_COMMUNICATION_CONFIG	Definiert die Kommunikationseigenschaften der UDP Verbindung
	Tolerances	MB_REGI_TOLERANCES	Definiert die zulässigen Toleranzen für das Teach-In und das Messen
	InternalParameter	MB_REGI_INTERNAL_PARAMETER	Interne Parameter des Sensors, die nur auf ausdrücklichen Hinweis geändert werden sollen
	RecvData	MB_REGI_RECEIVE_DATA	Empfange Daten vom Funktionsbaustein MB_RegiSensor01SendRecvDataType01
VAR_OUTPUT	Done	BOOL	Datenausgänge sind gültig
	Active	BOOL	Funktionsbaustein arbeitet und befindet sich in einem Zwischenzustand
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	SendData	MB_REGI_SEND_DATA	Daten die über den Funktionsbaustein MB_RegiSensor01SendRecvData gesendet werden sollen

Tab. 3-13: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01ConfigType01

Datenstrukturen

Folgende Datenstrukturen werden vom Funktionsbaustein MB_RegiSensor01ConfigType01 verwendet:

MB_REGI_SENSOR01_CFG		
CurrentConfig	USINT	Aktuelle Konfiguration
Format	MB_REGI_FORMAT	Beschreibt das Druckzylinderformat
MarkStructure	MB_REGI_MARK_STRUCTURE	Beschreibt die Markenstruktur
Barcode	MB_REGI_BARCODE	Beschreibt den Barcode

RMB_TechRegiExtended

MB_REGI_SENSOR01_CFG		
MarkStream	MB_REGI_MARK_STREAM	Beschreibt den Markenstrom
LengthRegisterMode	MB_REGI_LENGTH_REGISTER_MODE	Längsregistermodus
SideRegisterMode	MB_REGI_SIDE_REGISTER_MODE	Seitenregistermodus
CommunicationData	MB_REGI_COMMUNICATION_CONFIG	Die Kommunikationsports der UDP Verbindung
Tolerances	MB_REGI_TOLERANCES	Teach- und Messtoleranzen
InternalParameter	MB_REGI_INTERNAL_PARAMETER	Interne Sensor Parameter
BarcodeSearchBeginOfExpWindow	REAL	Gibt den Beginn des Erwartungsfensters aus der Barcode-Suche zurück

Tab. 3-14: Datenstruktur MB_REGI_SENSOR01_CFG

MB_REGI_FORMAT		
FormatLength	REAL	Druckzylinderformat in [mm]
Increments	UDINT	Anzahl der Inkremente pro Umdrehung

Tab. 3-15: Datenstruktur MB_REGI_FORMAT

MB_REGI_MARK_STRUCTURE		
MarkType	UDINT	Markentyp
SingleMark	ARRAY [0...3] OF REAL	Beschreibt die Druckmarke

Tab. 3-16: Datenstruktur MB_REGI_MARK_STRUCTURE

MB_REGI_BARCODE		
NumberOfBlocks	UDINT	Anzahl der Barcode Blöcke (0 bedeutet, dass kein Barcode vorhanden ist)
BeginExpectationWindow	REAL	Anfang des Erwartungsfensters (Entfernung des Beginns des Erwartungsfensters zum Nullimpuls) [mm]
SizeOfExpectationWindow	REAL	Größe des Erwartungsfensters [mm]
FreeSpaceBeforeBarcode	REAL	Druckfreier Raum vor dem Barcode [mm]
BarcodeBlocks	ARRAY [0...19] OF REAL	Barcode Beschreibung in [mm]

Tab. 3-17: Datenstruktur MB_REGI_BARCODE

MB_REGI_MARK_STREAM		
NumberOfMarks	UINT	Anzahl der Marken
MarkStream	ARRAY [1...12] OF REAL	Abstand der Marken [mm]
MarkDirection	BOOL	Erste oder zweite Markenflanke für das Längsregister

Tab. 3-18: Datenstruktur MB_REGI_MARK_STREAM

MB_REGI_LENGTH_REGISTER_MODE		
ModeSelect	UDINT	Ausgewählter Längsregistermodus
LengthProperties	ARRAY [0...4] OF REAL	Eigenschaften für den Längsregistermodus [mm]

Tab. 3-19: Datenstruktur MB_REGI_LENGTH_REGISTER_MODE

MB_REGI_SIDE_REGISTER_MODE		
ModeSelect	UDINT	Ausgewählter Seitenregistermodus
MarkSelect	ARRAY [0...1] OF UINT	Marken für das Seitenregister

Tab. 3-20: Datenstruktur MB_REGI_SIDE_REGISTER_MODE

MB_REGI_COMMUNICATION_CONFIG		
IpSensor	STRING	IP Adresse des Sensors
PortConfiguration	UINT	Port für die azyklische Kommunikation (Konfiguration, Diagnose, Geschwindigkeitsdaten)
IpPLC	STRING	IP Adresse der SPS
PortMeasurement	UINT	Port für die Messergebnisse

Tab. 3-21: Datenstruktur MB_REGI_COMMUNICATION_CONFIG

MB_REGI_TOLERANCES		
TeachZeroPulseToMark	REAL	Teach Toleranz der ersten Marke zum Nullimpuls [mm]
TeachMarkToMark	REAL	Teach Toleranz zwischen den einzelnen Marken [mm]
MeasureMarkTolerance	REAL	Messtoleranz im Vergleich zum nächsten Messwert [mm]
DeviationOfMarkWidth	REAL	Abweichung der Markenbreite zum letzten Messwert [%]
DeviationOfMarkHeight	REAL	Abweichung der Markenhöhe zum letzten Messwert [%]
DeviationOfLastPosition	REAL	Absolute Abweichung der Messtoleranz vom letzten Messwert [mm]
ShiftSyncPulse	REAL	Der Nullimpuls wird genau um ShiftSyncPulse [mm] vor das Erwartungsfenster verschoben

Tab. 3-22: Datenstruktur MB_REGI_TOLERANCES

MB_REGI_INTERNAL_PARAMETER		
ColorWeighting	REAL	Farbgewichtung des Sensors
IntensityWeighting	REAL	Intensitätsgewichtung des Sensors
Gain	UINT	Verstärkung des Sensors

Tab. 3-23: Datenstruktur MB_REGI_INTERNAL_PARAMETER

MB_REGI_SEND_DATA		
NoOfBytes	UDINT	Anzahl der Bytes, die versendet werden sollen
ValueAddr	POINTER TO DWORD	Pointer auf die Daten
NewData	UINT	Zeigt neue Daten zum Senden an
SensorPort	UINT	Sensor Port, auf den die Daten gesendet werden sollen

Tab. 3-24: Datenstruktur MB_REGI_SEND_DATA

MB_REGI_RECEIVE_DATA		
NoOfBytes	UDINT	Anzahl der Bytes, die empfangen wurden
ValueAddr	POINTER TO DWORD	Pointer auf die Daten
NewData	UINT	Zeigt neue empfangene Daten an

Tab. 3-25: Datenstruktur MB_REGI_RECEIVE_DATA

RMB_TechRegiExtended

Aufzählungstyp

Der Aufzählungstyp **MB_REGI_MEASURE_HANDLING** wird für das Konfigurieren, das Starten der Messung, des Teach-Ins, sowie zum Setzen des Erwartungsfensters verwendet und kann folgende Werte annehmen:

Element	Wert	Beschreibung
START_CONFIG	0	Startet die Konfiguration
START_TEACHING	1	Startet das Teachen
START_MEASUREMENT	2	Startet die Messung
STOP_MEASUREMENT	3	Stoppt die Messung
START_CONFIG_TEACHING	4	Startet die Konfiguration und das Teachen
START_TEACHING_MEASUREMENT	5	Startet das Teachen und die Messung
START_CONFIG_TEACHING_MEASUREMENT	6	Startet die Konfiguration, das Teachen und die Messung
START_SET_EXPECTATION_WINDOW	7	Setzt das Erwartungsfenster
START_EASY_TEACHING	8	Startet einen Easy-Teach-In. Es müssen die Verstärkung sowie Beginn und Größe des Erwartungsfensters übertragen werden.
START_EASY_TEACHING_MEASUREMENT	9	Startet den Easy-Teach-In und anschließend die Messung (vgl. 8)
START_BARCODE	10	Startet die Barcode-Suche

Tab. 3-26: Aufzählungstyp **MB_REGI_MEASURE_HANDLING**

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
MB_REGI_SENSOR01_CFG					
CurrentConfig	USINT	0	255	0	Steigende Flanke an Execute
MB_REGI_FORMAT					
FormatLength	REAL	> 0	n. def.	0	Steigende Flanke an Execute
Increments	UDINT	> 0	n. def.	0	Steigende Flanke an Execute
MB_REGI_MARK_STRUCTURE					
MarkType	UDINT	1	7	1 (Es sind nur Marken 1,3,5 implementiert)	Steigende Flanke an Execute
SingleMark	ARRAY [0...3] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Steigende Flanke an Execute
MB_REGI_BARCODE					
NumberOfBlocks	UDINT	0 (1 und 2 Blocks sind ungültige Eingaben)	10	0	Steigende Flanke an Execute
BeginExpectationWindow	REAL	0	n. def.	0	Steigende Flanke an Execute
SizeOfExpectationWindow	REAL	0	n. def.	1	Steigende Flanke an Execute

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
FreeSpaceBeforeBarcode	REAL	0	n. def.	0	Steigende Flanke an Execute
BarcodeBlocks	ARRAY [0...19] OF REAL	> 0 (für benötigte Eingaben)	n. def.		Steigende Flanke an Execute
MB_REGI_MARK_STREAM					
NumberOfMarks	UDINT	1	12	1	Steigende Flanke an Execute
MarkStream	Array [1...12] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Steigende Flanke an Execute
MarkDirection	BOOL			FALSE	Steigende Flanke an Execute
MB_REGI_LENGTH_REGISTER_MODE					
ModeSelect	UDINT	1	1	1	Steigende Flanke an Execute
LengthProperties	ARRAY [0...4] OF REAL	n. def.	n. def.	0	Steigende Flanke an Execute
MB_REGI_SIDE_REGISTER_MODE					
ModeSelect	UDINT	0	1	1	Steigende Flanke an Execute
SelectMarks	ARRAY [0...1] OF UINT	n. def.	n. def.	0	Steigende Flanke an Execute
MB_REGI_COMMUNICATION_CONFIG					
IpSensor	STRING	LEN > 0	n. def.	""	Steigende Flanke an Execute
PortConfiguration	UINT	1	n. def.	0	Steigende Flanke an Execute
IpPLC	STRING	LEN > 0	n. def.	""	Steigende Flanke an Execute
PortMeasurement	UINT	1	n. def.	0	Steigende Flanke an Execute
MB_REGI_TOLERANCES					
TeachZeroPulseToMark	REAL	1	n. def.	5	Steigende Flanke an Execute
TeachMarkToMark	REAL	1	n. def.	2	Steigende Flanke an Execute
MeasureMarkTolerance	REAL	1	n. def.	10	Steigende Flanke an Execute
DeviationOfMarkWidth	REAL	1	100	20	Steigende Flanke an Execute
DeviationOfMarkHeight	REAL	1	100	20	Steigende Flanke an Execute
DeviationOfLastPosition	REAL	1	n. def.	10	Steigende Flanke an Execute
ShiftSyncPulse	REAL	1	n. def.	5	Steigende Flanke an Execute
MB_REGI_INTERNAL_PARAMETER					
ColorWeighting	REAL	0	< 1	0.9999	Steigende Flanke an Execute

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
IntensityWeighting	REAL	0	< 1	0.15	Steigende Flanke an Execute
Gain	UINT	0	7	0	Steigende Flanke an Execute
MB_REGI_RECEIVE_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAddr	POINTER TO BYTE	n. def.	n. def.	0	Änderung an NewData
NewData	UINT	0	65535	0	Kontinuierlich

Tab. 3-27: Übersicht Min/Max-Werte und Initialwerte der Eingänge



Als gültige Eingaben werden nur die bisher im Sensor implementierten Funktionalitäten zugelassen.

Funktionsbeschreibung

Bei der azyklischen Kommunikation arbeitet der Sensor als Server, d. h. die Steuerung sendet Daten an den Sensor und der Sensor antwortet. Mit der azyklischen Kommunikation wird der Sensor konfiguriert.

Die Struktur RegiSensorCfg enthält die aktuelle Konfiguration des Sensors. Sollen Konfigurationsänderungen vorgenommen werden, so können die geänderten Konfigurationsdaten direkt an den Baustein angelegt werden. Sie werden erst bei einer steigenden Flanke von Execute übernommen. Im Folgenden werden die einzelnen Strukturen beschrieben:

RegiSensorCfg: Enthalten die aktuellen Konfigurationsdaten des Sensors und die aktuelle Konfigurations-ID, die in den zyklischen Messdaten übertragen werden. Außerdem wird nach einer Barcode-Suche der ermittelte Beginn des Erwartungsfensters angegeben.

Measure: Bei einer steigenden Flanke an Execute wird zwischen 11 unterschiedlichen Abläufen unterschieden, die je nach gewählter Funktion ausgeführt werden:

- **START_CONFIG:**

Hier wird eine neue Konfiguration an den Sensor übertragen. Das Erwartungsfenster wird hier selbständig durch den Sensor über die Markenabstände und die Toleranzen, sowie den Beginn des Erwartungsfensters berechnet.



Hierzu sind sehr exakte Eingaben erforderlich. Es wird empfohlen, den Beginn und die Größe des Erwartungsfensters über eine grafische Oberfläche (HMI) zu ermitteln und diese Werte mit der Option START_SET_EXPECTATION_WINDOW zu übertragen bzw. die Barcode-Suche zu verwenden.

- **START_TEACHING:**

Startet das Teachen des Sensors. Hierzu werden die zuletzt an den Sensor übertragenen Konfigurationsdaten und das Erwartungsfenster übernommen und die Marken mit ihren Schwellwerten im Erwartungsfenster ermittelt. In diesem Modus ermittelt der Sensor zusätzlich die ideale Verstärkung.

- **START_MEASUREMENT:**

Startet eine Messung, ohne eine Konfiguration zu übertragen oder ein Teach-In zu starten.

- **STOP_MEASUREMENT:**
Beendet eine laufende Messung.
- **START_CONFIG_TEACHING:**
Überträgt zunächst die Konfiguration und startet im Anschluss daran einen Teach-In. Wurde in `SizeOfExpectationWindow` eine Zahl größer Null angegeben, so wird dieser Wert für die Berechnung des Erwartungsfensters übernommen. Ist der Wert gleich Null, so wird das Erwartungsfenster selbständig durch den Sensor über die Markenabstände und die Toleranzen, sowie den Beginn des Erwartungsfensters berechnet.
- **START_TEACHING_MEASUREMENT:**
Startet das Teachen des Sensors und direkt im Anschluss daran die Messung. Für das Teachen werden die zuletzt an den Sensor übertragenen Konfigurationsdaten und das Erwartungsfenster übernommen.
- **START_CONFIG_TEACHING_MEASUREMENT:**
Überträgt zunächst die Konfiguration und startet im Anschluss daran einen Teach-In und danach die Messung. Wurde in `SizeOfExpectationWindow` eine Zahl größer Null angegeben, so wird dieser Wert für die Berechnung des Erwartungsfensters übernommen. Ist der Wert gleich Null, so wird das Erwartungsfenster selbständig durch den Sensor über die Markenabstände und die Toleranzen, sowie den Beginn des Erwartungsfensters berechnet.
- **START_SET_EXPECTATION_WINDOW:**
Setzt nur das Erwartungsfenster, das beispielsweise aus der grafischen Oberfläche übermittelt wurde. Hierzu muss in `SizeOfExpectationWindow` ein Wert größer Null angegeben werden, sowie der Beginn des Erwartungsfensters (`BeginExpectationWindow`).



Wird nach dem Übertragen des Erwartungsfensters eine neue Konfiguration gestartet, so wird das übertragene Erwartungsfenster überschrieben. Aus diesem Grund wird empfohlen, zunächst die Konfiguration zu übertragen und anschließend das Erwartungsfenster.

- **START_EASY_TEACHING:**
Es wird ein Teach-In gestartet, ohne dass die Verstärkung ermittelt wird. Es wird die Verstärkung, die unter `InternalParameters` in `Gain` steht, übernommen. Diese Verstärkung ist idealerweise nach einem erstmaligen normalen Teach-In mit dem Funktionsbaustein `MB_RegiSensor01ReadDiagType01` ausgelesen worden. Weiterhin müssen die Größe und der Beginn des Erwartungsfensters angegeben werden.
- **START_EASY_TEACHING_MEASUREMENT:**
Wie `START_EASY_TEACHING`, jedoch wird nach dem Teachen sofort die Messung gestartet.
- **START_BARCODE:**
Wird dieser Modus ausgeführt, so wird die Barcode-Suche gestartet. Hierzu müssen vorab 3-10 Barcodes definiert sein. Nach erfolgreicher Barcode-Suche steht die Position direkt nach dem Barcode, d. h. der Beginn des Erwartungsfensters in `RegiSensorCfg.BarcodeSearchBegin`.

RMB_TechRegiExtended

nOfExpWindow. Dieser Wert kann zur Definition des Erwartungsfensters verwendet werden.

In nachfolgender Abbildung wird der beispielhafte Ablauf von der Konfiguration des Markenstromsensors bis zur Messung dargestellt.

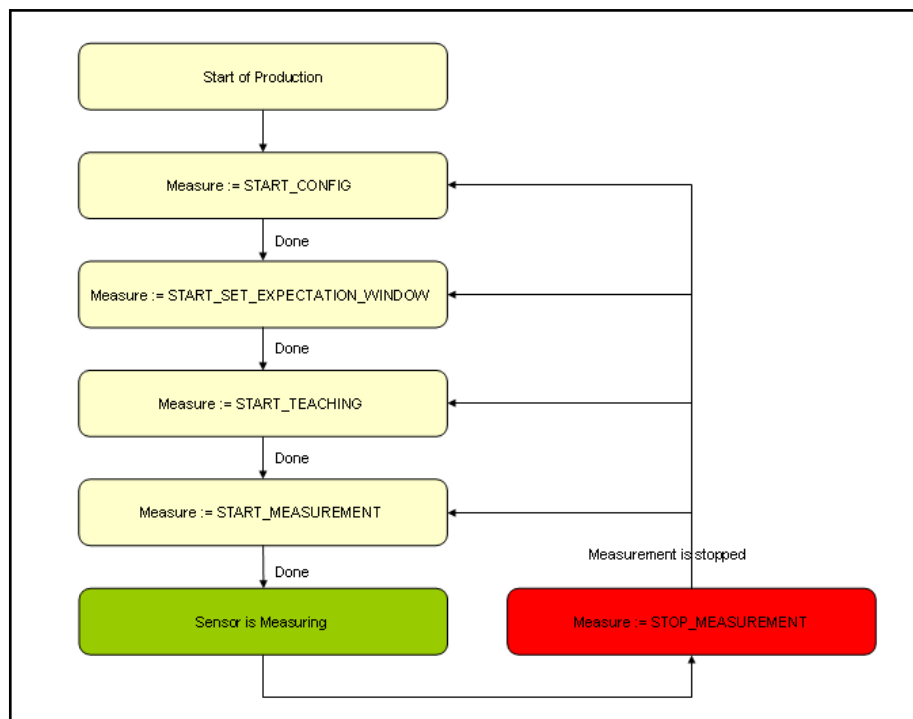


Abb. 3-18: Ablauf von der Konfiguration bis zur Messung

Der Sensor muss zunächst konfiguriert werden. Anschließend muss das Erwartungsfenster gesetzt werden, was vorzugsweise über eine Visualisierung bzw. die hierüber ermittelten Werte geschehen soll. Danach kann das Teach und das Messen gestartet werden. Nachdem die Messung gestartet wurde, kann sie beliebig gestoppt werden und bei dem gewünschten Sensor-Zustand fortgefahren werden. Die Measure-Funktionalitäten können in den beliebigen Kombinationen aufgerufen werden.

MarkStructure: Enthält eine vordefinierte Markenstruktur und die Abmessung der Marken.

Folgende Markentypen werden unterstützt:

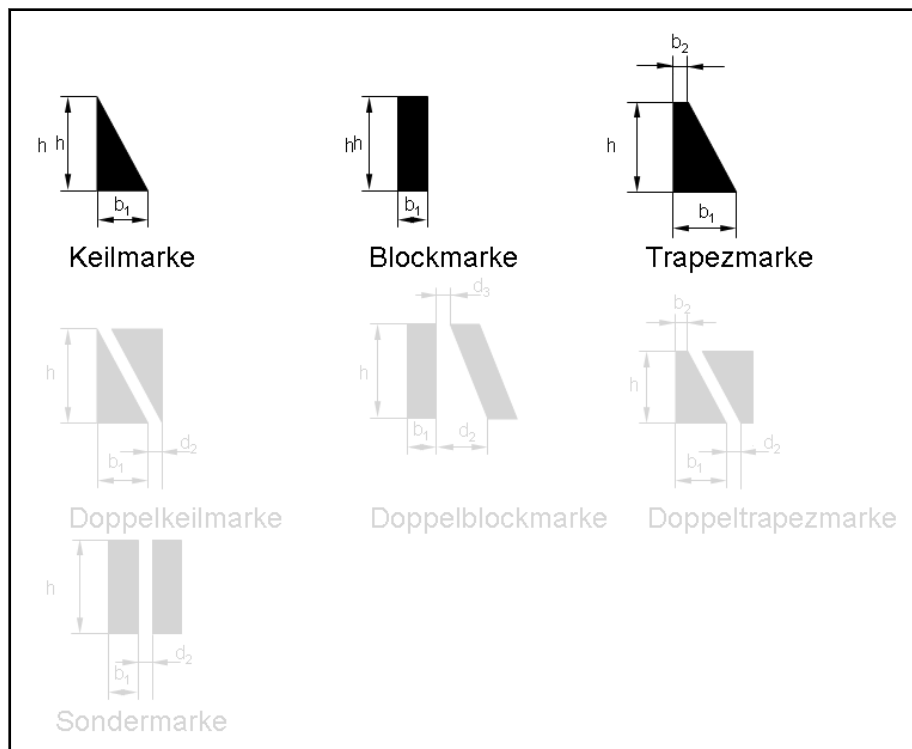


Abb. 3-19: Unterstützte Markentypen

Der Parameter MarkType kann wie folgt gesetzt werden und die Parameter müssen nachfolgend in SingleMark je nach Markentyp angegeben werden:

	MarkType	SingleMark			
Keilmarke	1	Breite [mm]	Höhe [mm]		
Doppelkeilmarke	2	Breite [mm]	Höhe [mm]	Abstand [mm]	
Blockmarken	3	Breite [mm]	Höhe [mm]		
Doppelblockmarken	4	Breite [mm]	Höhe [mm]	Abstand [mm]	Abstand2 [mm]
Trapezmarken	5	Breite [mm]	Höhe [mm]	Breite2 [mm]	
Doppeltrapezmarken	6	Breite [mm]	Höhe [mm]	Abstand [mm]	Breite2 [mm]
Sondermarke	7	Breite [mm]	Höhe [mm]	Abstand [mm]	

Tab. 3-28: Parameter der Markentypen

Nachfolgend wird ein Beispiel der Konfiguration einer Dreiecksmarke angegeben:

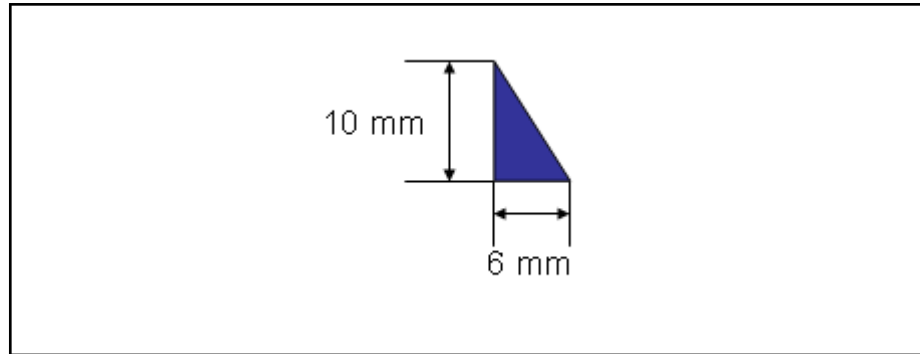


Abb. 3-20: Beispiel Markenstruktur

Code-Beispiel:

```
MarkStruct: MB_REGI_MARK_STRUCTURE := (MarkType:=1, SingleMark:= 6, 10, 0, 0);
```

Es handelt sich hier um eine Keilmärke (Markentyp 1). Diese wird durch ihre Höhe und die Breite definiert.



In der bisherigen Implementierung sind nur die Markentypen 1, 3 und 5 zugelassen.

Barcode: Enthält die Anzahl der Barcode-Blöcke und die Beschreibung des Erwartungsfensters. Es sind bis zu 10 Barcode-Blöcke möglich. Ist NumberOfBlocks = 0, so ist kein Barcode vorhanden. Hier muss der Beginn und die Größe des Erwartungsfensters durch den Anwender gesetzt werden. Dies kann entweder über die Ermittlung dieser Daten über die Oszilloskopdarstellung erfolgen oder über die selbständige Berechnung der Größe des Erwartungsfensters durch den Sensor (aus Anzahl der Marken und deren Abstand, sowie den Toleranzen und dem Beginn des Erwartungsfensters). Hierzu muss Barcode.SizeOfExpectationWindow = 0 gesetzt werden.

Ist ein Barcode vorhanden, so kann sich der Sensor selbständig den Beginn des Erwartungsfensters ermitteln, indem er die Position des Barcode-Endes zurückliefert. Hierzu muss START_BARCODE gestartet werden. Liefert der Funktionsbaustein ein Done, so steht in RegiSensorCfg.BarcodeSearchBeginOfExpWindow die Position des Barcodeendes. Der Anwender muss jetzt noch die Größe des Erwartungsfensters angeben und kann nachfolgend mit START_SET_EXPECTATION_WINDOW das Erwartungsfenster setzen. Anschließend kann der normale Ablauf mit Teach-In und Messung starten fortgesetzt werden.

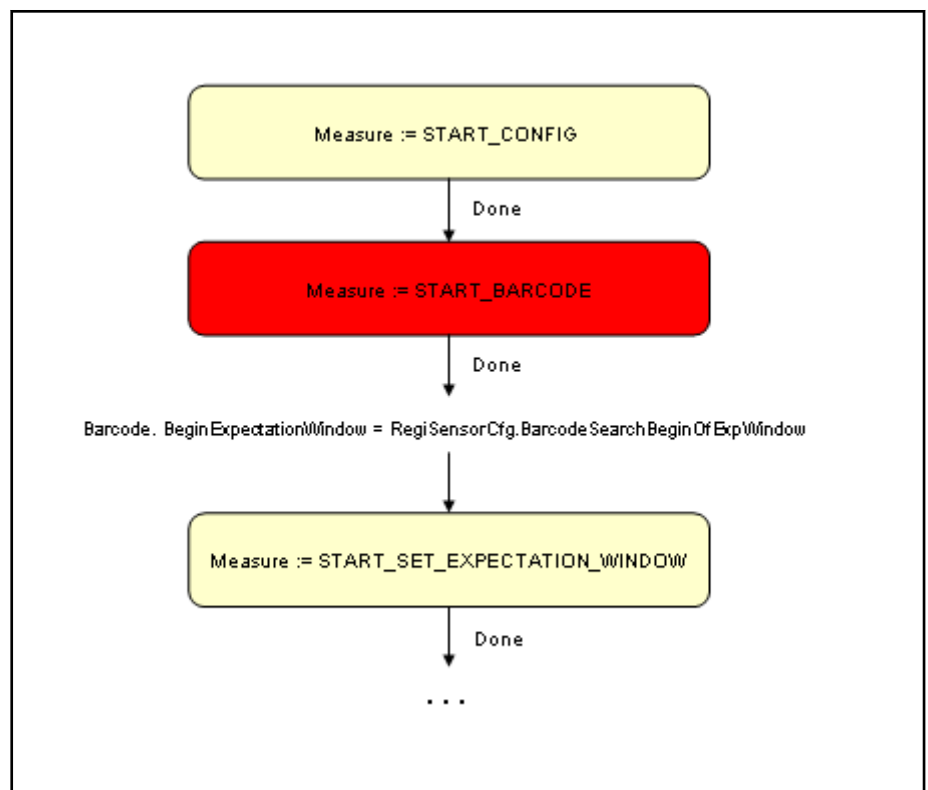


Abb. 3-21: Ablauf der Barcode-Suche

Ist ein Barcode vorhanden, so sucht sich der Sensor selbständig den Beginn des Erwartungsfensters. `FreeSpaceBeforeBarcode` gibt den druckfreien Raum vor dem Barcode an und `BarcodeBlocks` definiert die einzelnen Barcode Blöcke. Weiterhin müssen die Abmessungen des Barcodes wie folgt angegeben werden:

Beispiel für einen Barcode:

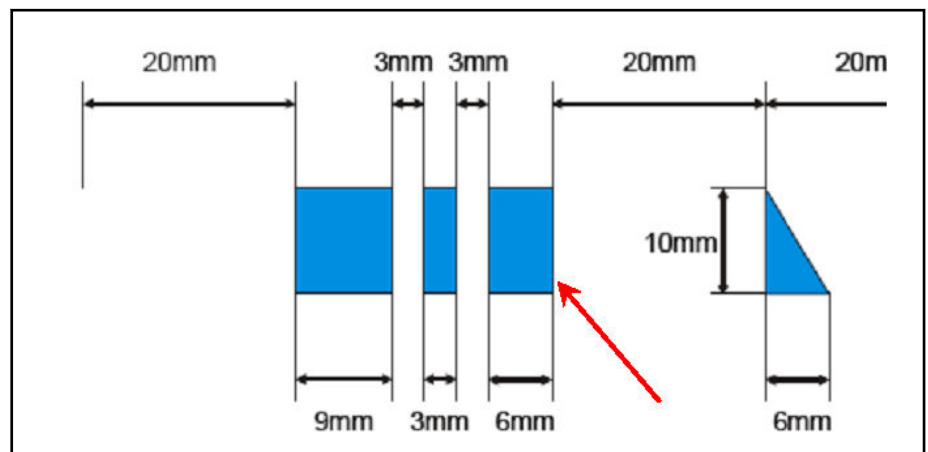


Abb. 3-22: Beispiel für einen Barcode

Die Abmessungen stehen im Array BarcodeBlocks in [mm]:

BarcodeBlocks[0]: Breite des Barcode Blocks 1 in [mm]

BarcodeBlocks[1]: Abstand des Barcode Blocks 1 zum Block 2 in [mm]

BarcodeBlocks[2]: Breite des Barcode Blocks 2 in [mm]

BarcodeBlocks[3]: Abstand des Barcode Blocks 2 zum Block 3 in [mm]

.....

Der letzte angegebene Abstand ist der Abstand vom letzten Barcode Block zur ersten Marke (in diesem Beispiel beträgt der Abstand zwischen Barcode Block und erster Registermarke 20 mm). Mit dem roten Pfeil ist die Position gekennzeichnet, die der Sensor nach erfolgreicher Barcode-Suche in Regi-SensorCfg.BarcodeSearchBeginOfExpWindow zurückliefert, also direkt die Position nach dem Barcode.

MarkStream: Enthält die Anzahl der Marken und den jeweiligen Abstand zwischen den Marken. Sind 5 Marken vorhanden, so gibt es 4 Markenabstände und der erste definierte Abstand ist der Abstand zwischen Marke 1 und Marke 2.

Beispiel für einen Markenstrom:

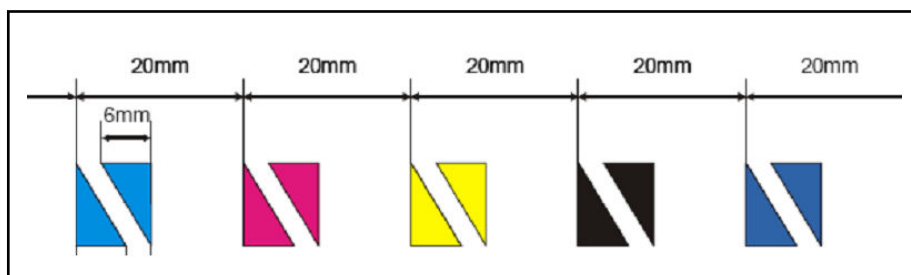


Abb. 3-23: Beispiel für einen Markenstrom

Beispiel für den dargestellten Markenstrom:

```
MarkStream: MB_REGI_MARK_STREAM := (NumberOfMarks:=5,
MarkStream:=20,20,20,20);
```

Der Markenstrom besteht aus 5 Marken. Diese besitzen jeweils den Abstand von 20 mm.

Weiterhin kann bei der Definition des Markenstroms angegeben werden, ob die senkrechte Markenflanke an erster oder zweiter Stelle kommt.

Ist MarkDirection = FALSE wird der Längsregisterwert in Bezug auf die jeweils erste Markenflanke zurückgegeben. Also der Abstand vom Nullimpuls zur ersten Markenflanke der Marke.

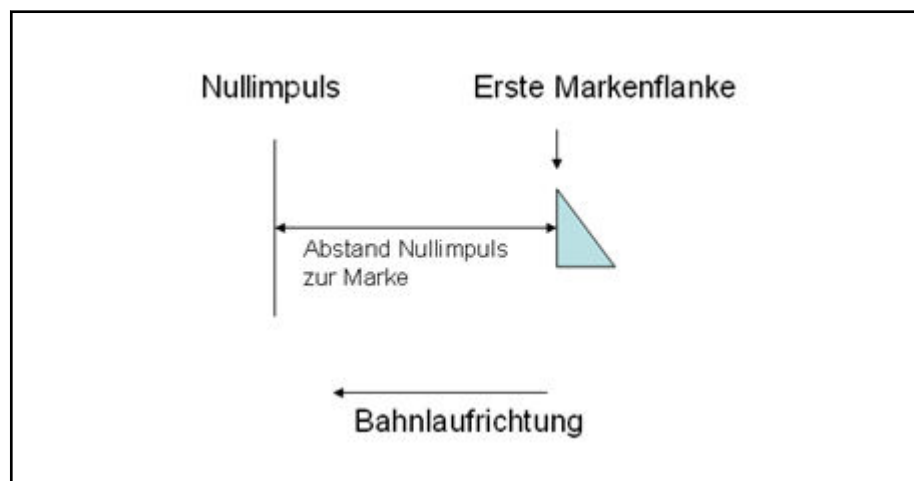


Abb. 3-24: Erste Markenflanke

Ist MarkDirection = TRUE wird der Längsregisterwert in Bezug auf die jeweils zweite Markenflanke zurückgegeben.

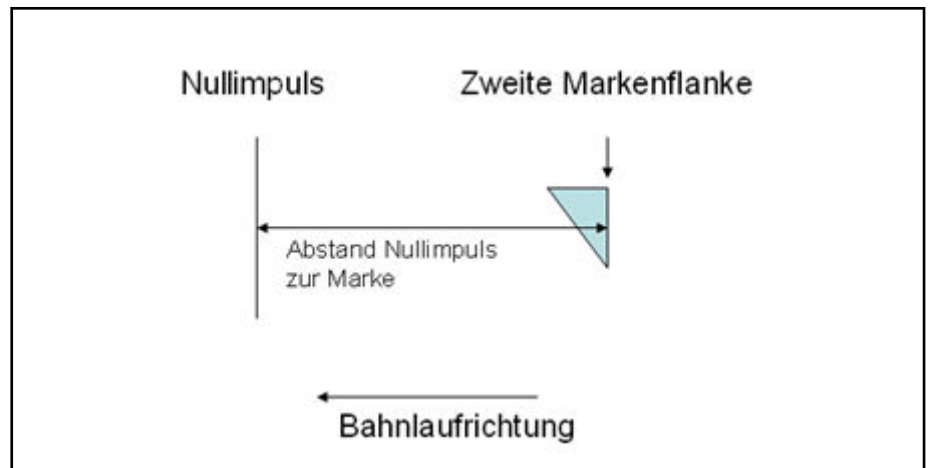


Abb. 3-25: Zweite Markenflanke



Es wird empfohlen, **immer** die senkrechte Flanke für die Längsregisterregelung zu verwenden. Wird die schräge Flanke der Marke verwendet, so würde eine Verstellung im Seitenregister immer eine Verschiebung im Längsregister bedeuten.

LengthRegisterMode: Gibt das entsprechende Bahnmessverfahren für das Längsregister an. Weiterhin sind alle benötigten Abmessungen für das entsprechende Bahnmessverfahren gespeichert. Im Moment ist im Markenstromsensor nur der Modus 1 der Rückgabe aller Markenabstände implementiert.

Beispiel:

```
LengthRegisterMode: MB_REGI_LENGTH_REGISTER_MODE := (ModeSelect:=1);
```

ModeSelect := 1 bedeutet, dass alle Markenabstände aller Marken zurückgegeben werden.

SideRegisterMode: Gibt an, welche Markenbreiten welcher Marken für das Seitenregister vermessen werden sollen. Ist ModeSelect := 0, werden keine Markenbreiten zurückgegeben. Ist ModeSelect := 1, werden alle Markenbreiten zurückgegeben.

Beispiel 1:

```
SideRegisterMode: MB_REGI_SIDE_REGISTER_MODE := (ModeSelect:=0);
```

Es werden keine Markenbreiten zurückgegeben.

Beispiel 2:

```
SideRegisterMode: MB_REGI_SIDE_REGISTER_MODE := (ModeSelect:=1);
```

Es werden alle Markenbreiten zurückgegeben.

Es steht im Moment noch keine Funktion zur Verfügung, zwei Marken, deren Markenbreiten ausgegeben werden sollen, anzugeben.

CommunicationConfig: Enthält die Kommunikationseinstellungen für die azyklische und die zyklische Kommunikation. Hierbei werden der Port auf der SPS für die Messwerte und der Port für Konfigurations- und Diagnosedaten angegeben, sowie die IP-Adresse des Sensors und der SPS.

Beispiel:

RMB_TechRegiExtended

```
CommunicationCfg: MB_REGI_COMMUNICATION_CONFIG := (IpSensor:='192.168.2.5', PortConfiguration:=6001,
IpPLC:='192.168.2.3', PortMeasurement:=6000);
```

Im obigen Beispiel wird eine Konfiguration der Verbindungsdaten aufgezeigt. Der Sensor hat die IP Adresse 192.168.2.5 und die azyklische Kommunikation erfolgt über den UDP-Port 6001 der Steuerung. Die SPS hat die IP-Adresse 192.168.2.3. Somit weiß der Sensor, dass er die zyklischen Messergebnisse an diese IP 192.168.2.3 und den Udp-Port 6000 der Steuerung senden soll.

Tolerances:

Bei den Toleranzen sollten die entsprechenden Default-Werte verwendet und nur im Ausnahmefall geändert werden.

Tolerances gibt Toleranzen für das Teachen, das Messen und für die Robustheit der Messung an:

- TeachZeroPulseToMark:
Gibt die Toleranz an, um wie viel die Angabe BeginExpectationWindow abweichen darf. Der Sensor setzt sein Erwartungsfenster genau an die Position:
BeginExpectationWindow - TeachZeroPulseToMark.
- TeachMarkToMark:
Gibt für das Teachen eine Toleranz an, wie viel mm an das Erwartungsfenster als Toleranz angehängt werden. Dieser Wert sollte gleich MeasureMarkTolerance gewählt werden, um zu verhindern, dass beim Start der Messung eine Marke verloren wird.
- MeasureMarkTolerance:
Während einer Messung wird für jede Marke ein eigenes Markenerwartungsfenster verwendet. D. h. die Marke wird in Bezug auf Ihre letzte Position genau in diesem Erwartungsfenster gesucht. Hat sich die Position der Marke seit der letzten Umdrehung weiter verändert, so wird die Marke nicht gefunden.
Die nachfolgenden 3 Parameter dienen der Robustheit der Messung. D. h. eine Marke wurde bei der letzten Messung mit einer bestimmten Breite, Höhe und Position erfasst. Weicht bei der nächsten Messung die gefundene Marke um den entsprechenden Prozentsatz von der absoluten Position ab, so wird diese Messung ignoriert und erneut mit den bestehenden Daten der letzten gültigen Messung gesucht. Dies soll verhindern, dass eine einmalig auftretende Störung (z. B. Falte) eine Marke bei der nächsten Messung unauffindbar macht.
- DeviationOfMarkWidth:
Maximale prozentuale zugelassene Abweichung der Markenbreite von der letzten Messung. Wird dieser Wert auf 100 % gesetzt, so wird die Überwachung der Markenbreite abgeschaltet.
- DeviationOfMarkHeight:
Maximale prozentuale zugelassene Abweichung der Markenhöhe von der letzten Messung. Wird dieser Wert auf 100 % gesetzt, so wird die Überwachung der Markenhöhe abgeschaltet.
- DeviationOfLastPosition:
Maximale absolute zugelassene Abweichung der Markenposition von der letzten Messung
- ShiftSyncPulse:

RMB_TechRegiExtended

Gibt die Entfernung vom verschobenen Nullimpuls bis zum Beginn des Erwartungsfensters an. Um eine hohe Messgenauigkeit während der Beschleunigungsphase zu gewährleisten, wird der Nullimpuls hardwaremäßig bis zu dem Beginn des Erwartungsfensters verschoben. Dieser Wert gibt die Toleranz zwischen verschobenem Nullimpuls und Beginn des Erwartungsfensters an. Ist zu erwarten, dass der Markenstrom nicht weiter nach vorne wandert, wird ein Default-Wert von 5 mm empfohlen.

InternalParameter: Diese Datenstruktur enthält interne Parameter des Sensors, die nur auf Anweisung von Bosch Rexroth geändert werden sollten. Hier kann die Gewichtung des Farb- und des Intensitätsanteils vorgenommen werden. Die Default-Werte sind gut geeignete Erfahrungswerte. Weiterhin enthält die Struktur die Verstärkung, die beim Easy-Teach-In übertragen wird.

- **ColorWeighting:**
Gibt die Farbgewichtung im Analogsignal an.
- **IntensityWeighting:**
Gibt die Intensitätsgewichtung im Analogsignal an.
- **Gain:**
Gibt die aktuelle Verstärkung des Sensors an, die beim Easy-Teach-In geschrieben wird. Diese Verstärkung wird in der Regel über den Diagnose Funktionsbaustein MB_RegiSensor01ReadDiagType01 ausgelesen.

Senden und Empfangen der Datenpakete

Der Funktionsbaustein zeigt durch die Inkrementierung von NewData in SendData an, dass er neue Daten versenden will. Das Versenden geschieht durch den Funktionsbaustein MB_RegiSensor01SendRecvDataType01. Empfängt dieser neue Daten, so setzt er den Wert von NewData in RecvData gleich. Somit weiß der Funktionsbaustein, dass es sich um das erwartete Antwortpaket handelt. Ein Antwortpaket wird bei jedem übertragenen Datenpaket erwartet.

Timeout:

Wird ein neues Datenpaket gesendet, so läuft ein interner Timer im Funktionsbaustein mit. Bekommt er innerhalb von 10s keine Rückantwort, so bricht er aufgrund einer Zeitüberschreitung mit einem Fehler ab.

Ein möglicher Fehlerfall könnte sein, dass der Funktionsbaustein MB_RegiSensor01SendRecvDataType01 nicht aktiviert ist bzw. fehlerhaft verbunden ist.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170. Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000000	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvDataType01 innerhalb von 10 Sekunden
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000001	Timeout beim Teachen. Das Teachen konnte nicht in der angegebenen Zeit abgeschlossen werden.
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Formzylinderformat

RMB_TechRegiExtended

ErrorID	Additional 1	Additional 2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000001	Ungültiger Markentyp
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000002	Ungültige Markengeometrie
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000003	Ungültige Anzahl der Barcodeblöcke
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000004	Ungültige Anzahl an Marken im Markenstrom
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000005	Ungültiger Markenabstand
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000006	Ungültiger Längsregistermodus
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000007	Ungültige Längsregisterparameter
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000008	Ungültiger Seitenregistermodus
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000009	Ungültige Sensor IP
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000A	Ungültige Steuerungs IP
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000B	Ungültiger UDP-Port für die Messung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000C	Ungültiger Konfigurationsport
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000D	Ungültige Anzahl an Inkrementen
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000E	Ungültige Messtoleranz
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000F	Ungültige Teachtoleranz einer Marke
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000010	Ungültige Teachtoleranz Erwartungsfensterbeginn
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000011	Ungültiger Beginn des Erwartungsfensters
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000012	Ungültige Größe des Erwartungsfenster
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000013	Ungültige Toleranz für die Markenbreite
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000014	Ungültige Toleranz für die Markenhöhe
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000015	Ungültige Toleranz für die Markenposition
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000016	Ungültige ShiftSyncPulse Toleranz
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000017	Ungültige Farbgewichtung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000018	Ungültige Intensitätsgewichtung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000019	Ungültiger Measurement Modus
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001A	Ungültige Verstärkung
COMMUNICATION_ERROR (16#0002)	16#0C001D10	16#00000002	Timeout bei der Barcode-Suche
OTHER_ERROR (16#7FFE)	16#00001D12	16#xxxxxxx	Fehlerbehandlung für den Markenstromsensor (siehe Tabelle Markenstromsensor) nicht in „F_RELATED_TABLE“ integriert

Tab. 3-29: Fehlercodes des Funktionsbausteins MB_RegiSensor01ConfigType01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors ausgeben.

3.3.4 MB_RegiSensor01SendRecvDataType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01SendRecvDataType01 führt die Kommunikation mit dem Sensor durch. Hierzu übergeben die Funktionsbau-

RMB_TechRegiExtended

steine für die Konfiguration des Sensors, für das Senden von Geschwindigkeits- und Beschleunigungsinformationen und der Diagnosebaustein während einer Beschleunigungsphase ihre zu sendenden Daten und bekommen die entsprechenden Datenpakete zurück. Der Funktionsbaustein verhindert eine Kollision von gleichzeitig auf ein und denselben UDP-Port zugreifenden Funktionsbausteinen. Weiterhin ist eine Priorisierung implementiert, da das Senden der Beschleunigungsinformationen zeitkritisch ist. Aus diesem Grund werden Datenpakete des Funktionsbausteins MB_RegiSensor01SendSpeedAccType01 sofort verschickt, sobald das Senden möglich ist. Konfigurationsdaten des Funktionsbausteins MB_RegiSensor01ConfigType01, Diagnosedaten des Funktionsbausteins MB_RegiSensor01ReadDiagType01 oder Teach-Daten des Funktionsbausteins MB_RegiSensor01TeachDataType01 sind in der Priorisierung dem Funktionsbausteins MB_RegiSensor01SendSpeedAccType01 untergeordnet.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-30: Referenztabelle Funktionsbaustein MB_RegiSensor01SendRecvDataType01

Schnittstellenbeschreibung

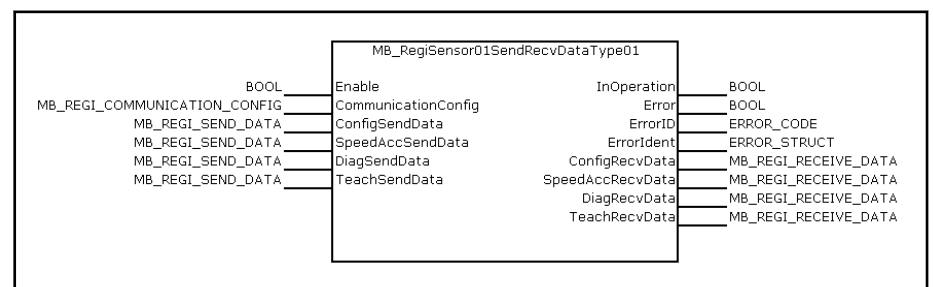


Abb. 3-26: Funktionsbaustein MB_RegiSensor01SendRecvDataType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins.
	CommunicationConfig	MB_REGI_COMMUNICATION_CFG	Enthält den UDP Port der Steuerung auf den sich der Sensor die Messdaten schickt.
	ConfigSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01ConfigType01 versenden will.
	SpeedAccSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 versenden will.
	DiagSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01ReadDiagType01 versenden will.
	TeachSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01TeachDataType01 versenden will.

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detailldiagnose
	ConfigRecv-Data	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01ConfigType01 empfangen wurden.
	SpeedAcc-RecvData	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 empfangen wurden.
	DiagRecvData	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01ReadDiagType01 empfangen wurden.
	TeachRecv-Data	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01TeachDataType01 empfangen wurden.

Tab. 3-31: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01SendRecvDataType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
MB_REGI_COMMUNICATION_CONFIG					
IpSensor	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortConfiguration	UINT	1	n. def.	0	Steigende Flanke an Enable
IpPLC	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortMeasurement	UINT	1	n. def.	0	Steigende Flanke an Enable
MB_REGI_SEND_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAddr	POINTER TO BYTE	n. def.	n. def.	0	Änderung an NewData
NewData	UINT	0	65535	0	Kontinuierlich
SensorPort	UINT	1	65535	0	Änderung an NewData

Tab. 3-32: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

Die verwendeten Datenstrukturen [Tab. 3-24 "Datenstruktur MB_REGI_SEND_DATA" auf Seite 43](#) und [Tab. 3-21 "Datenstruktur MB_REGI_COMMUNICATION_CONFIG" auf Seite 43](#) wurden beim Funktionsbaustein [Kap. 3.3.3 "MB_RegiSensor01ConfigType01" auf Seite 40](#) beschrieben.

Funktionsbeschreibung

Soll eine Kommunikation mit einem Sensor aufgebaut werden, so muss dieser Funktionsbaustein vor oder zeitgleich mit den angehängten Funktions-

bausteinen MB_RegiSensor01SendSpeedAccType01, MB_RegiSensor01ConfigType01, MB_RegiSensor01TeachDataType01 und MB_RegiSensor01ReadDiagType01 mit einer steigenden Flanke an Enable gestartet werden. Bei einer steigenden Flanke an Enable, werden nach der Prüfung der Eingänge der entsprechende UDP-Port PortConfiguration auf der Steuerung gebunden, um auf diesem Port entsprechende Daten zu empfangen. Der Funktionsbaustein wartet nun auf Daten. D. h. er überprüft bei jedem Aufruf, ob sich der Eingang NewData bei der Struktur ConfigSendData, AccelerationSendData, DiagSendData und TeachSendData ändert. Ist dies der Fall, so werden die entsprechenden Daten (NoOfBytes) versendet. Der Funktionsbaustein wartet anschließend auf die entsprechenden Daten, falls ein Antwortpaket vom Sensor erwartet wird.

Ausnahme: Beim Übertragen der Beschleunigung und der Geschwindigkeit wird kein Antwortpaket erwartet.

Es findet auch ein Timeouthandling statt. Ist nach 2 Sekunden kein Antwortpaket empfangen worden, so wird die Sendung des Datenpakets wiederholt. Nach dem dritten Fehlversuch wird ein Fehler erzeugt und die Übertragung abgebrochen. Wird ein Paket empfangen, so wird die Anzahl der Bytes und die Adresse (ValueAddr) angegeben, an der die Daten liegen. NewData am Ausgang wird auf den gleichen Wert gesetzt, den NewData am Eingang beim Senden der Daten hatte. Somit kann exakt zugeordnet werden zu welchem gesendeten Paket die Antwort gehört.

Datenpuffer: Es ist kein **Datenpuffer** intern implementiert, da jeder angelegte Funktionsbaustein (falls ein Handshake stattfindet) das nächste Paket erst dann versendet, wenn er die entsprechende Antwort erhalten hat.

Ausnahme: Liegt ein neues Paket mit Geschwindigkeits- und Beschleunigungsinformation vor und kann dieses nicht gesendet werden bis ein neues Paket erzeugt wird, so wird das neue Paket versendet und das alte verworfen.

Priorisierung: Die übertragenen Konfigurationsdaten sind nicht zeitkritisch. Einzig die Werte der Beschleunigung und der Geschwindigkeit müssen schnellstmöglich übertragen werden. Aus diesem Grund ist eine Priorisierung implementiert. Das Datenpaket des Funktionsbausteins MB_RegiSensor01SendSpeedAccType01 hat absoluten Vorrang und wird sofort gesendet, sobald der "Senden"-Funktionsbaustein frei wird. Wird in dieser Zeit auf ein Antwortpaket der Konfiguration oder Diagnose gewartet, so wird dies zunächst unterbrochen, das zeitkritische Paket gesendet, um anschließend das Konfigurations- bzw. Diagnoseantwortpaket zu empfangen. Die Priorisierung ist in nachfolgender Abbildung dargestellt.

RMB_TechRegiExtended

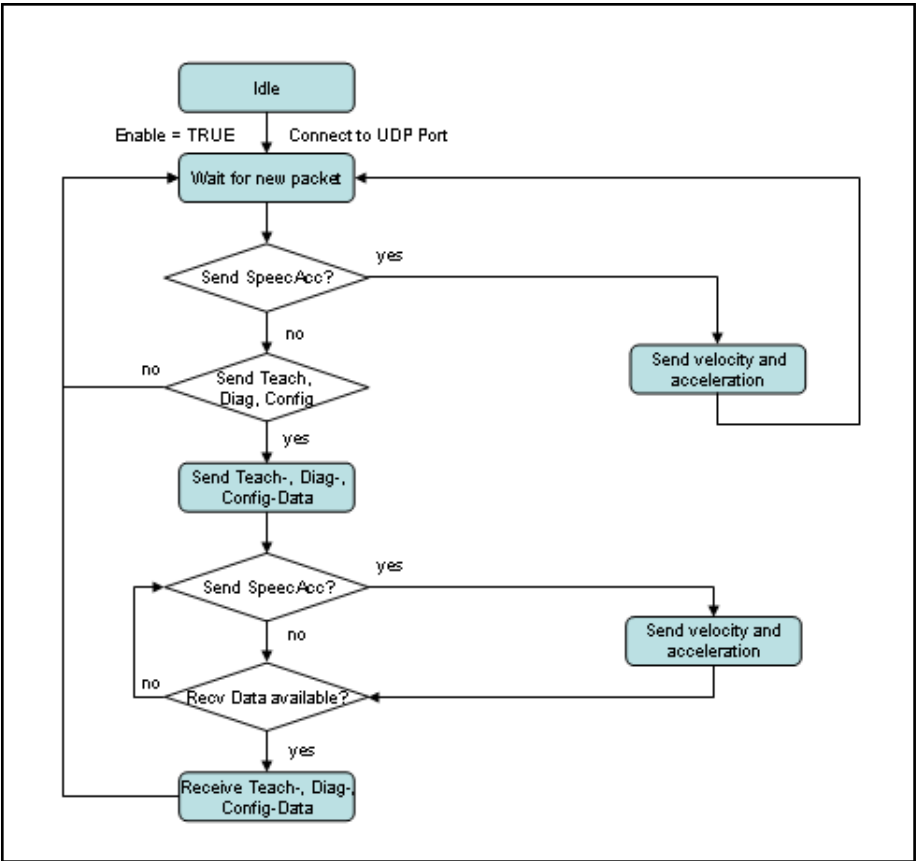


Abb. 3-27: Schema des internen Ablaufs des Funktionsbausteins MB_RegiSensor01SendRecvDataType01

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000003	Timeout beim Verbinden auf den UDP-Port
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001B	Ungültiger UDP-Port PortConfiguration
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000004	Timeout beim Senden
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000005	Timeout beim Empfangen
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001C	Ungültige Sensor IP

Tab. 3-33: Fehlercodes des Funktionsbausteins MB_RegiSensor01SendRecvDataType01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors und der Bibliothek RIL_SocketComm.lib ausgeben.

3.3.5 MB_RegiSensor01CycComType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01CycComType01 führt die zyklische Kommunikation mit dem Sensor über die Ethernet-Schnittstelle durch. Er

RMB_TechRegiExtended

empfängt die Messwerte des Sensors, wertet diese aus und gibt den Markenabstand bzw. die Differenz der Markenbreiten von 2 Registermarken zurück. Es wird jeweils ein Messwert für das Längs- bzw. Seitenregister zurückgegeben. Die Kommunikation mit dem Sensor findet intern statt. Werden Messwerte mehrerer Registermarken benötigt, so ist der Baustein MB_RegiSensor01CycComType02 zu verwenden.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-34: Referenztabelle Funktionsbaustein MB_RegiSensor01CycComType01

Schnittstellenbeschreibung

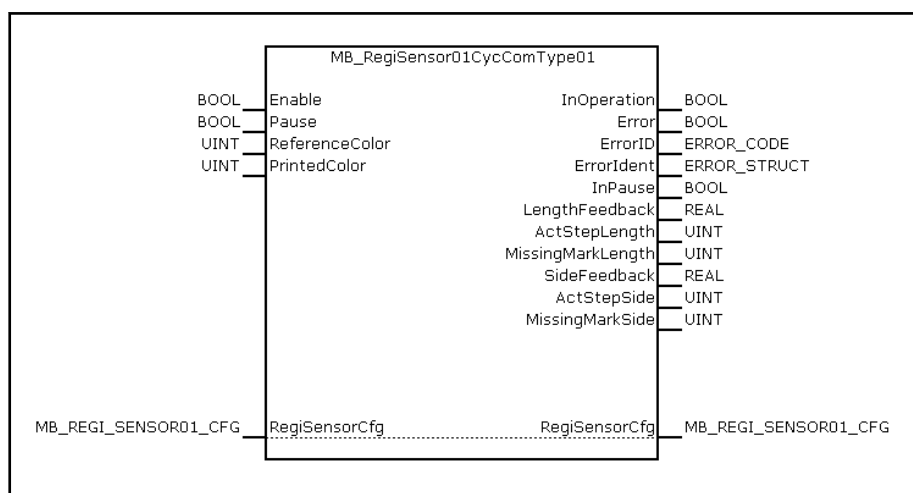


Abb. 3-28: Funktionsbaustein MB_RegiSensor01CycComType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	RegiSensorCfg	MB_REGI_SENSOR01_CFG	Enthält die aktuellen Konfigurationsdaten des Sensors.
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert).
	Pause	BOOL	Unterbricht die Abarbeitung des FBs. Der Wert an "LengthFeedback" und "SideFeedback" bleibt konstant.
	ReferenceColor	UINT	Gibt die Referenzfarbe an, zu der die aktuell gedruckte Marke verglichen wird. ReferenceColor = 1 bedeutet, dass die aktuell gedruckte Marke mit der ersten Marke des Markenstroms verglichen wird.
	PrintedColor	UINT	Gibt die Position der vom zu regelnden Druckwerk gedruckten Marke im Markenstrom an.

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPause	BOOL	Zeigt an, dass der Eingang "Pause" aktiv ist und sich der Wert in "LengthFeedback" und "SideFeedback" nicht ändert.
	LengthFeed-back	REAL	Der berechnete Markenabstand für das Längsregister.
	ActStepLength	UINT	Enthält die Sequenznummer des letzten Messpakets, wenn ein neuer gültiger Messwert für das Längsregister vorliegt.
	MissingMarkLength	UINT	Wird inkrementiert, wenn ein ungültiger Messwert für das Längsregister vorliegt.
	SideFeedback	REAL	Der Istwert für das Seitenregister.
	ActStepSide	UINT	Enthält die Sequenznummer des letzten Messpakets, wenn ein neuer gültiger Messwert für das Seitenregister vorliegt.
	MissingMark-Side	UINT	Wird inkrementiert, wenn ein ungültiger Messwert für das Seitenregister vorliegt.

Tab. 3-35: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01CycComType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
ReferenceColor	UINT	0	16	0	Kontinuierlich
PrintedColor	UINT	0	16	0	Kontinuierlich

Tab. 3-36: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

Die verwendete Datenstruktur [Tab. 3-14 "Datenstruktur MB_REGI_SENSOR01_CFG" auf Seite 41](#) wurde beim Funktionsbaustein [Kap. 3.3.3 "MB_RegiSensor01ConfigType01" auf Seite 40](#) beschrieben.

Funktionsbeschreibung

Der Funktionsbaustein darf erst nach einem Aufruf des Funktionsbausteins MB_RegiSensor01ConfigType01 zum Durchführen einer Konfiguration mit einer steigenden Flanke an Enable aktiviert werden, da der Funktionsbaustein MB_RegiSensor01CycComType01 auf den UDP-Port der Struktur RegiSensorCfg zugreift und sich darauf verbindet. Wurde noch keine Konfiguration durchgeführt, so liegt ein fehlerhafter UDP-Port vor, da dieser noch nicht in die Struktur geschrieben wurde. In diesem Fall wird der entsprechende Error Ausgang des Funktionsbausteins gesetzt.

Bei der zyklischen Kommunikation arbeitet die SPS als UDP-Server. Das bedeutet, die Steuerung empfängt, an dem über die azyklische Kommunikation definierten UDP-Port, die Pakete des Sensors mit seinen Messwerten. Der Sensor sendet zyklisch nach jeder Zylinderumdrehung (d. h. nach jeder er-

folgten Messauswertung) die Messdaten. Diese Messdaten werden von dem Funktionsblock ausgewertet. Beim Setzen von Enable wird ein bind() auf dem in RegiSensorCfg definierten UDP-Port gemacht. Somit wird auf dem UDP-Port mitgehört und die Messwerte empfangen und ausgewertet.

ReferenceColor und PrintedColor:

Die Eingänge ReferenceColor und PrintedColor geben an, welche Marken zueinander vermessen werden.

Ist beispielsweise ReferenceColor = 1 und PrintedColor = 4 so wird der Markenabstand zwischen den Marken 1 und 4 des Markenstroms zurückgegeben. Dies ermöglicht, dass während des Betriebs, von Referenzfarbe auf Vorgängerfarbe umgeschaltet werden kann, indem man ReferenceColor = 3 wählt. PrintedColor ist die Marke, die vom zu regelnden Druckwerk gedruckt wird.

Wurde ein neuer Istwert berechnet, so wird dieser ausgegeben und ActStepLength bzw. ActStepSide enthalten die Sequenznummer des letzten Messpakets, damit der Registerregler weiß, dass ein neuer Messwert vorliegt.

Ist ein ungültiger Messwert vorhanden, der die Auswertung der Marken ReferenceColor und PrintedColor nicht möglich macht, so wird MissingMarkLength/Side inkrementiert und ActStepLength/Side bleibt unverändert. Kommt ein neuer gültiger Wert, so wird MissingMarkLength/Side wieder zurückgesetzt.

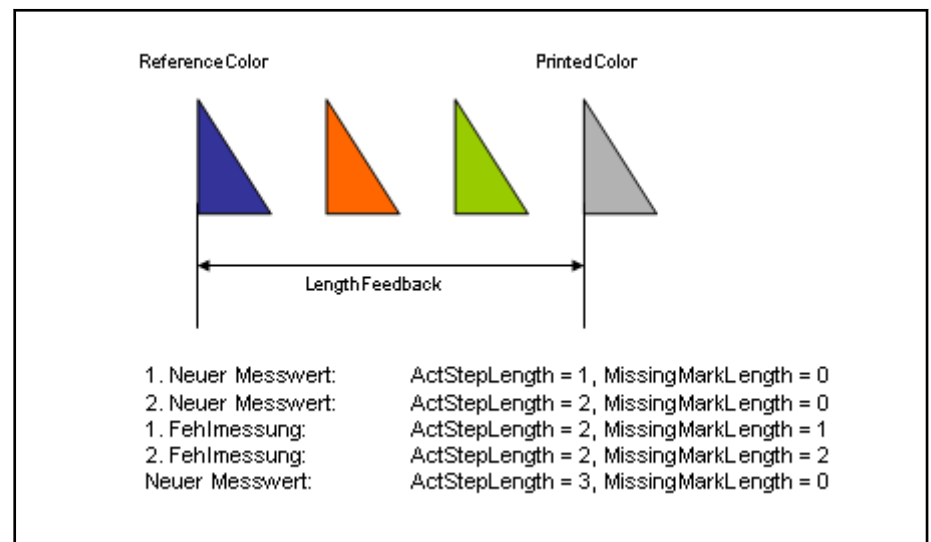


Abb. 3-29: Verhalten der Ausgänge bei Fehlmessungen

In der obigen Abbildung wird beispielhaft das Verhalten der Ausgänge für die Messung des Längsregisters gezeigt. Für das Seitenregister verhält es sich äquivalent. Als Referenzfarbe wurde Registermarke 1 gewählt, als gedruckte Marke Registermarke 4. LengthFeedback ist somit der Abstand zwischen Marke 4 und 1. Bei einem neuen Messwert wird LengthFeedback aktualisiert, ActStepLength inkrementiert. Bei der ersten Fehlmessung wird ActStepLength und LengthFeedback eingefroren und es wird MissingMarkLength inkrementiert. Bei einer erneuten gültigen Messung wird MissingMarkLength auf 0 gesetzt und ActStepLength inkrementiert und der aktuelle Messwert LengthFeedback ausgegeben.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

RMB_TechRegiExtended

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000003	Timeout beim Verbinden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001D	ReferenceColor ist außerhalb der definierten Grenze
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001E	PrintedColor ist außerhalb der definierten Grenze
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001F	PrintedColor und ReferenceColor sind gleich
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000020	Keine gültige Konfiguration gefunden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000021	PrintedColor oder ReferenceColor liegen nicht im Bereich der definierten Marken
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000B	Ungültiger UDP-Port für die Messung

Tab. 3-37: Fehlercodes des Funktionsbausteins MB_RegiSensor01CycComType01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors und der Bibliothek RIL_SocketComm.lib ausgeben.

3.3.6 MB_RegiSensor01CycComType02

Kurzbeschreibung

Der Baustein MB_RegiSensor01CycComType02 wird für die Kommunikation mit dem Markenstromsensor benötigt. Er führt jedoch nur die zyklische Kommunikation durch. Die Konfiguration des Sensors wird von dem Baustein MB_RegiSensor01ConfigType01 übernommen. Erst wenn der Sensor korrekt konfiguriert wurde, wird mit der Messung begonnen und der Baustein kann mit der Messauswertung beginnen. Es werden alle gemessenen Werte für das Längs- und Seitenregister am Ausgang angegeben. Diese Daten können vom Benutzer entsprechend der Konfiguration ausgewertet werden. Dieser Baustein MB_RegiSensor01CycComType02 eignet sich besonders für Anwendungen, bei denen mehrere gedruckte Marken zueinander vermessen werden. Beispielsweise ist dies der Fall, wenn sich nur ein Markenstromsensor am Ende der Maschine befindet.

Eine beispielhafte Anwendung ist in nachfolgender Abbildung zu sehen:

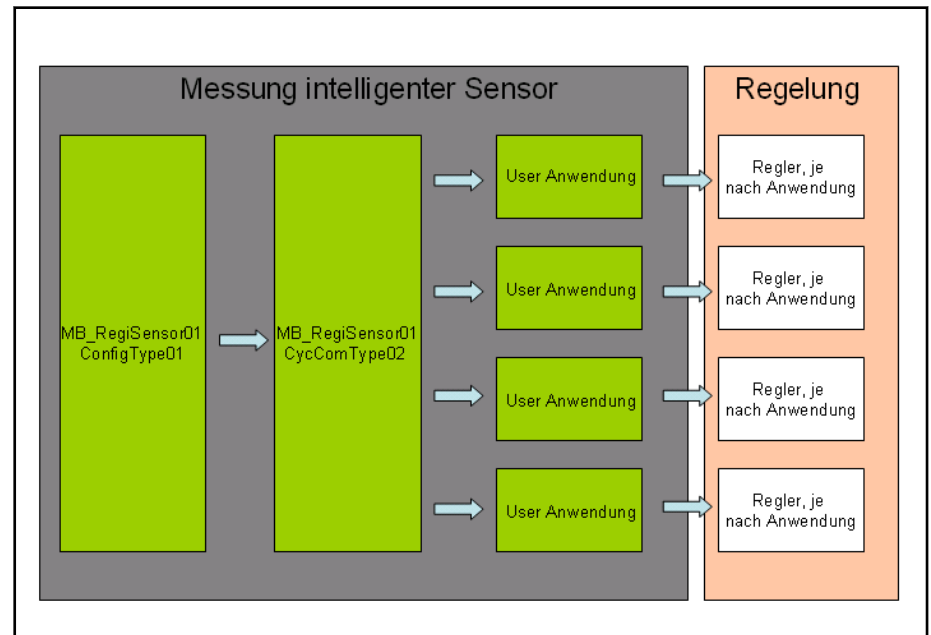


Abb. 3-30: Beispielhafte Anwendung des Funktionsbausteins MB_RegiSensor01CycComType02

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-38: Referenztabelle Funktionsbaustein MB_RegiSensor01CycComType02

Schnittstellenbeschreibung

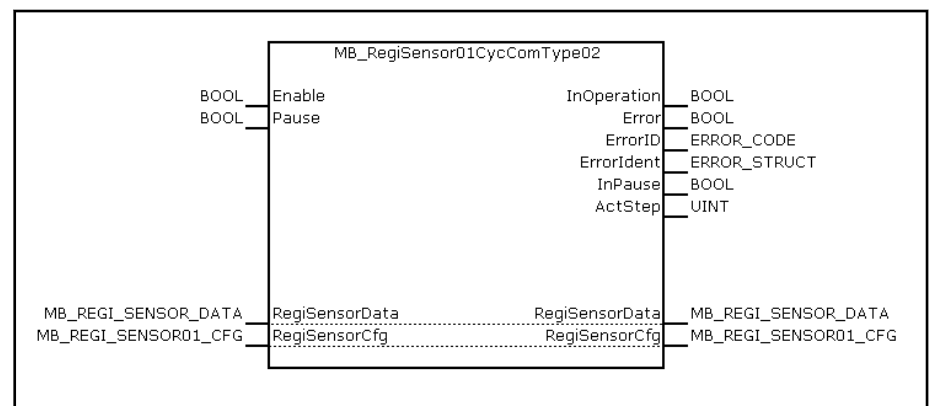


Abb. 3-31: Funktionsbaustein MB_RegiSensor01CycComType02

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT	RegiSensorCfg	MB_REGI_SENSOR01_CFG	Enthält die aktuellen Konfigurationsdaten des Sensors.
	RegiSensorData	MB_REGI_SENSOR01_DATA	Enthält die entsprechenden Messwerte des Sensors.

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert).
	Pause	BOOL	Unterbricht die Abarbeitung des FBs. Es werden keine neuen Messwerte ausgewertet. Die letzten Messdaten werden eingefroren.
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	InPause	BOOL	Zeigt an, dass der Eingang "Pause" aktiv ist und sich der Wert in RegiSensorData nicht ändert.
	ActStep	UINT	Zeigt an, dass ein neues Datenpaket mit Messwerten erhalten und ausgewertet wurde.

Tab. 3-39: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01CycCom-Type02

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich

Tab. 3-40: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

MB_REGI_SENSOR01_DATA		
ActStep	UINT	Aktueller Schritt
LengthFeedback	ARRAY [0..16] OF REAL	Enthält den Markenabstand vom Nullimpuls zu den Marken 1..16. In [0] steht der Markenabstand für Bahn/Bahn2, falls eine Marke auf der Oberseite mit einer auf der Unterseite vermessen wurde.
SideFeedback	ARRAY [1..16] OF REAL	Enthält die entsprechenden Markenbreiten.
MarkState	ARRAY [0..16] OF MB_REGI_MARK_STATE	Status der Registermarken.
SensorState	MB_REGI_SENSOR_STATE	Gibt Warnmeldungen und Diagnosen des Sensors während der Messung an.

Tab. 3-41: Datenstruktur MB_REGI_SENSOR01_DATA

Die verwendete Datenstruktur [Tab. 3-14 "Datenstruktur MB_REGI_SENSOR01_CFG"](#) auf Seite 41 wurde beim Funktionsbaustein [Kap. 3.3.3 "MB_RegiSensor01ConfigType01"](#) auf Seite 40 beschrieben.

Aufzählungstyp

Der Aufzählungstyp MB_REGI_MARK_STATE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
FOUND	0	Marke gefunden
ROTATED	1	Marke verdreht

Element	Wert	Beschreibung
LARGE	2	Marke zu groß
SMALL	3	Marke zu klein
MISSING	4	Marke nicht gefunden
NOT_CONFIGURED	5	Marke nicht konfiguriert

Tab. 3-42: Aufzählungstyp MB_REGI_MARK_STATE

Der Aufzählungstyp MB_REGI_SENSOR_STATE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
START	0	Startzustand
SEARCHING	1	Suche Marken
INOP	2	Marken gefunden (Betriebsmodus)
STOP	3	Bahngeschwindigkeit unter 0,4m/s – keine Auswertung
LAMP	4	Lampe defekt
SENSOR	5	Sensor verschmutzt
TEACHING	6	Sensor in Teach-Phase

Tab. 3-43: Aufzählungstyp MB_REGI_SENSOR_STATE

Funktionsbeschreibung

Der Funktionsbaustein darf erst nach einem Aufruf des Funktionsbausteins MB_RegiSensor01ConfigType01 zum Durchführen einer Konfiguration mit einer steigenden Flanke an Enable aktiviert werden, da der Funktionsbaustein MB_RegiSensor01CycComType02 auf den UDP-Port der Struktur RegiSensorCfg zugreift und sich darauf verbindet. Wurde noch keine Konfiguration durchgeführt, so liegt ein fehlerhafter UDP-Port vor, da dieser noch nicht in die Struktur geschrieben wurde. In diesem Fall wird der entsprechende Error Ausgang des Funktionsbausteins gesetzt.

Bei der zyklischen Kommunikation arbeitet die SPS als UDP-Server. Das bedeutet, die Steuerung empfängt an dem über die azyklische Kommunikation definierten UDP-Port die Pakete des Sensors mit seinen Messwerten. Der Sensor sendet zyklisch nach jeder Zylinderumdrehung (d. h. nach jeder erfolgten Messauswertung) die Messdaten. Diese Messdaten werden von dem Funktionsblock ausgewertet. Beim Setzen von Enable wird ein bind() auf dem in RegiSensorCfg definierten UDP-Port gemacht. Somit wird auf dem UDP-Port mitgehört und die Messwerte empfangen und ausgewertet.

Jeder Sensor benötigt somit 2 UDP-Ports auf der Steuerung: Einen für die Konfigurations- und Diagnosedaten und einer für die Messwerterfassung. Die Kommunikation mit dem Sensor für das Empfangen der Messwerte erfolgt intern im Funktionsbaustein.

Die Struktur **RegiSensorData** enthält alle gemessenen Werte.

LengthFeedback[i]: Das Array LengthFeedback enthält alle gemessenen Markenabstände in Bezug auf den Nullimpuls sowie einen Messwert des Bahn/Bahn2 Verfahrens, die wie folgt in das Array eingetragen werden:

LengthFeedback[0]: Markenabstand definierte Marke unten zu definierter Marke oben.

LengthFeedback[1]: Markenabstand zwischen Nullimpuls und Marke 1.

RMB_TechRegiExtended

LengthFeedback[2]: Markenabstand zwischen Nullimpuls und Marke 2.
...
LengthFeedback[15]: Markenabstand zwischen Nullimpuls und Marke 15.
LengthFeedback[16]: Markenabstand zwischen Nullimpuls und Marke 16.
Ist ein Messwert nicht vorhanden, da er nicht konfiguriert wurde oder weniger Marken vorhanden sind, so ist der Rückgabewert 0.0 und entsprechend im MarkState als Missing oder Not_Configured definiert.

SideFeedback[i]: Das Array SideFeedback enthält alle gemessenen Markenbreiten, die wie folgt im Array eingetragen sind:
SideFeedback[1]: Markenbreite Marke 1
SideFeedback[2]: Markenbreite Marke 2
...
SideFeedback[16]: Markenbreite Marke 16
Ist ein Messwert nicht vorhanden, da er nicht konfiguriert wurde oder weniger Marken vorhanden sind, so ist der Rückgabewert 0.0 und entsprechend im MarkState als Missing oder Not_Configured definiert.

SensorState: Wird nicht gesetzt.

MarkState[i]: Gibt die Gültigkeit der Werte in den Arrays LengthFeedback und SideFeedback an. Ist der Messwert z. B. der Marke 1 gültig, so steht in MarkState[1] = FOUND. Somit können die Werte LengthFeedback[1] und SideFeedback[1] ausgewertet werden. Ansonsten steht im MarkState[1] = Missing oder Not_Configured, je nachdem, ob die Marke ausgefallen ist oder ob die Marke nicht konfiguriert wurde.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000003	Timeout beim Verbinden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000020	Keine gültige Konfiguration gefunden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000B	Ungültiger UDP Port

Tab. 3-44: Fehlercodes des Funktionsbausteins MB_RegiSensor01CycCom-Type02

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors und der Bibliothek RIL_SocketComm.lib ausgeben.

3.3.7 MB_RegiSensor01CommType02

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01CommType02 enthält die bisher einzeln implementierten Funktionsbausteine:

- MB_RegiSensor01ConfigType02
- MB_RegiSensor01CycComType04
- MB_RegiSensor01ReadDiagType01
- MB_RegiSensor01SendRecvDataType02

- MB_RegiSensor01SendSpeedAccType01 und
- MB_RegiSensor01TeachDataType01

Der Funktionsbaustein hat intern eine feste Verschaltung der Funktionsbausteine und bietet eine einfache Bedienerschnittstelle zum Ausführen der einzelnen Funktionen. Mit dem Funktionsbaustein ist die volle Steuerung des Markenstromsensors möglich.



Der folgende Funktionsbaustein zum Setzen der IP-Adresse ist nicht integriert, da er nur einmal pro Maschine benötigt wird:

- MB_RegiSensor01SetIPAddressType01



Der Funktionsbaustein MB_RegiSensor01CommType02 muss in einer Motion synchronen Task aufgerufen werden, damit die Beschleunigung korrekt berechnet wird. Ist dies aus Laufzeitgründen nicht möglich, so kann die Funktion mMotionCyclic() des Funktionsbausteins aus der Motion synchronen Task aufgerufen werden und der Funktionsbaustein MB_RegiSensor01CommType02 kann in einer zyklischen Task aufgerufen werden. Der Aufruf in der MotionTask muss folgende Struktur besitzen: {CyclicTask}.fbRegiSensor01CommType02.{fbInstanz}.mMotionCyclic(). Die Elemente in der geschweiften Klammer müssen durch den Namen des Programms in der zyklischen Task und durch den Instanznamen des Funktionsbausteins ersetzt werden. Somit könnte der Methodenaufruf aus einem Programm das in der Motion Task läuft wie folgt aussehen: PlcProg.fbRegiSensor01CommType02.fbRegiSensor01CommType02.mMotionCyclic(). Wird dieser Aufruf nicht durchgeführt, kann eine korrekte Berechnung der Beschleunigung nicht gewährleistet werden.



Folgende Erweiterungen wurden im Funktionsbaustein MB_RegiSensor01CommType02 vorgenommen:

- Die Datenstruktur MB_REGI_INTERNAL_PARAMETER_TYPE02 wurde modifiziert
- Die Farbgewichtung definiert nun gleichzeitig die Intensitätsgewichtung
- Der MarkAnalogLevel gibt den Analoglevel an, ab welchem eine Marke als gültige Marke erkannt wird
- Der BarcodeSearchLevel gibt die Wahrscheinlichkeitsschwelle an, ab welcher ein Barcode als gültige Barcode erkannt wird
- Der Ausgang BarcodeFoundProbability gibt die Wahrscheinlichkeit an, mit der ein Barcode bei der letzten Barcodesuche ermittelt wurde
- Die Verstärkung, die Farbgewichtung, der MarkAnalogLevel und der BarcodeSearchLevel können nun mittels einer eigenen Aktion MB_REGI_SENSOR01_COMM_ACTIONS alleine gesetzt werden
- Doppelmarken sind zugelassen
- Es wird überprüft, dass eine Sensorfirmware 12VRS verwendet wird, ansonsten wird ein Fehler zurück gegeben



Der Funktionsbaustein MB_RegiSensor01CommType02 muss zusammen mit der Sensorfirmware 12 VRS verwendet werden.
Mit ihm können zusätzlich benötigte Einstellungen für die Sensorfirmware 12 vorgenommen werden, die mit dem Funktionsbaustein MB_RegiSensor01CommType01 nicht möglich sind.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V14	RMB_TechRegiExtended

Tab. 3-45: Referenztabelle Funktionsbaustein MB_RegiSensor01CommType02

Schnittstellenbeschreibung

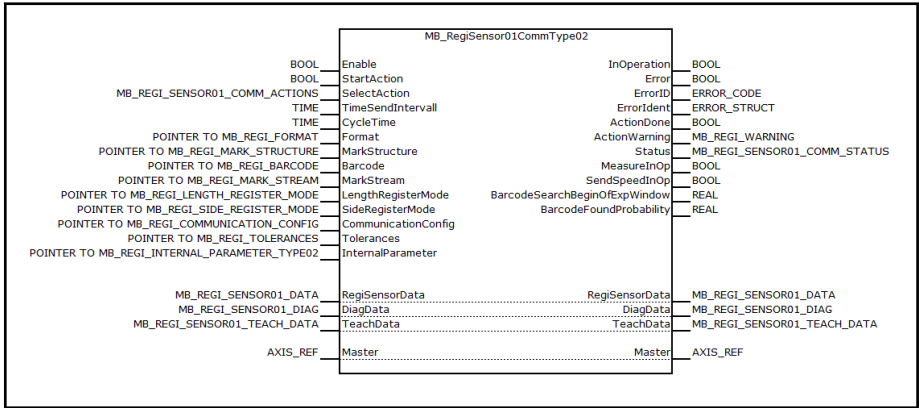


Abb. 3-32: Funktionsbaustein MB_RegiSensor01CommType02

I/O-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT CONST	Master	AXIS_REF	Adressierung der Leitachse
VAR_IN_OUT	RegiSensor-Data	MB_REGI_SENSOR01_DATA	Enthält die entsprechenden Messwerte des Sensors
	DiagData	MB_REGI_SENSOR01_DIAG	Diagnosedaten des Sensors
	TeachData	MB_REGI_SENSOR01_TEACH_DATA	Zu schreibende oder gelesene Daten des Sensors

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Erläuterung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert)
	StartAction	BOOL	Startet die ausgewählte Aktion
	SelectAction	MB_REGI_SENSOR01_COMM_ACTIONS	Auswahl der durchführbaren Aktionen
	TimeSendInterval	TIME	In diesem Intervall werden die Daten der Geschwindigkeits- und Beschleunigungsinformation berechnet und versendet
	CycleTime	TIME	Zykluszeit in der der Funktionsbaustein aufgerufen wird.
	Format	POINTER TO MB_REGI_FORMAT	Definiert die Formateigenschaften
	MarkStructure	POINTER TO MB_REGI_MARK_STRUCTURE	Definiert die Markenstruktur
	Barcode	POINTER TO MB_REGI_BARCODE	Definiert den Barcode
	MarkStream	POINTER TO MB_REGI_MARK_STREAM	Definiert den Markenstrom
	LengthRegisterMode	POINTER TO MB_REGI_LENGTH_REGISTER_MODE	Definiert das Bahnmessverfahren für das Längsregister
	SideRegisterMode	POINTER TO MB_REGI_SIDE_REGISTER_MODE	Definiert das Bahnmessverfahren für das Seitenregister
	CommunicationConfig	POINTER TO MB_REGI_COMMUNICATION_CONFIG	Definiert die Kommunikationseigenschaften der UDP-Verbindung
	Tolerances	POINTER TO MB_REGI_TOLERANCES	Definiert die zulässigen Toleranzen für das Teachin und das Messen
	InternalParameter	POINTER TO MB_REGI_INTERNAL_PARAMETER_TYPE02	Interne Parameter des Sensors, die nur auf ausdrücklichen Hinweis geändert werden sollen.

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Erläuterung
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Beschreibung der Diagnose im Fehlerfall
	ErrorIdent	ERROR_STRUCT	Detailldiagnose
	ActionDone	BOOL	Aktion wurde erfolgreich abgeschlossen
	ActionWarning	MB_REGI_WARNING	Enthält Warnungen von nicht erfolgreich durchgeführten Aktionen bzw. vom Sensorsystem, die nicht zu einem Fehler des Funktionsbausteins führen
	Status	MB_REGI_SENSOR01_COMM_STATUS	Gibt den aktuellen Status des Funktionsbausteins MB_RegiSensor01CommType02 an
	MeasureInOp	BOOL	Messung läuft
	SendSpeedInOp	BOOL	Geschwindigkeits- und Beschleunigungswerte werden übertragen.
	BarcodeSearchBeginOfExpWindow	REAL	Dieser Wert enthält die Position des Endes des Barcodes
	BarcodeFoundProbability	REAL	Wahrscheinlichkeit mit der der Barcode gefunden wurde

Tab. 3-46: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01CommType02

Datenstrukturen

Die folgenden, verwendeten Datenstrukturen wurden bei den vorherigen Funktionsbausteinen der Bibliothek RMB_TechRegiExtended beschrieben:

- [Tab. 3-59 "Datenstruktur MB_REGI_FORMAT" auf Seite 90](#)
- [Tab. 3-60 "Datenstruktur MB_REGI_MARK_STRUCTURE" auf Seite 90](#)
- [Tab. 3-61 "Datenstruktur MB_REGI_BARCODE" auf Seite 90](#)
- [Tab. 3-62 "Datenstruktur MB_REGI_MARK_STREAM" auf Seite 90](#)
- [Tab. 3-63 "Datenstruktur MB_REGI_LENGTH_REGISTER_MODE" auf Seite 91](#)
- [Tab. 3-64 "Datenstruktur MB_REGI_SIDE_REGISTER_MODE" auf Seite 91](#)
- [Tab. 3-65 "Datenstruktur MB_REGI_COMMUNICATION_CONFIG" auf Seite 91](#)
- [Tab. 3-66 "Datenstruktur MB_REGI_TOLERANCES" auf Seite 91](#)
- [Tab. 3-81 "Datenstruktur MB_REGI_SENSOR01_DATA" auf Seite 110](#)
- [Tab. 3-89 "Datenstruktur MB_REGI_SENSOR01_DIAG" auf Seite 114](#)
- [Tab. 3-98 "Datenstruktur MB_REGI_SENSOR01_TEACH_DATA" auf Seite 121](#)

MB_REGI_INTERNAL_PARAMETER_TYPE02		
ColorWeighting	REAL	Farbgewichtung des Sensors. Definiert ebenfalls die Intensitätsgewichtung mit 1-Color-Weighting.
Gain	UINT	Verstärkung des Sensors.
MarkAnalogLevel	UINT	Analoglevel ab diesem eine Marke als gültige Marke erkannt wird.
BarcodeSearchLevel	REAL	Wahrscheinlichkeitsschwelle ab welcher ein Barcode als gültige Barcode erkannt wird.

Tab. 3-47: Datenstruktur MB_REGI_INTERNAL_PARAMETER_TYPE02

MB_REGI_WARNING		
Warning	BOOL	Zeigt an, dass eine Warnung vorliegt
WarningAdditional1	DWORD	Zeigt den Additional 1 an.
WarningAdditional2	DWORD	Zeigt den Additional 2 an.

Tab. 3-48: Datenstruktur MB_REGI_WARNING

Aufzählungstyp

Der Aufzählungstyp MB_REGI_SENSOR01_COMM_STATUS kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
WAIT_FOR_NEXT_ACTION	0	Funktionsbaustein wartet auf nachfolgende Aktion
CONFIGURATION_IN_WORK	1	Konfiguration wird übertragen
BARCODE_SEARCH_IN_WORK	2	Barcode Suche läuft
MARKSTREAM_SEARCH_IN_WORK	3	Markenstrom Suche läuft
TEACH_IN_IN_WORK	4	TeachIn läuft
EASY_TEACH_IN_IN_WORK	5	EasyTeachIn läuft
READ_DIAG_IN_WORK	6	ReadDiagData läuft
READ_TEACH_DATA_IN_WORK	7	ReadTeachData läuft
WRITE_TEACH_DATA_IN_WORK	8	WriteTeachData läuft
MEASUREMENT_STARTING	9	Messung wird gestartet
MEASUREMENT_STOPPING	10	Messung wird gestoppt
SEND_SPEED_STARTING	11	SendSpeed wird gestartet
SEND_SPEED_STOPPING	12	SendSpeed wird gestoppt
SENSOR_OFF	13	Der Funktionsbaustein ist nicht im Zustand InOperation.
SENSOR_ERROR	14	Sensor im Error Zustand
SET_GAIN_IN_WORK	15	Die Verstärkung wird gesetzt
SET_MARK_ANALOG_LEVEL_IN_WORK	16	Der Marken Analoglevel wird gesetzt
SET_BARCODE_SEARCH_LEVEL_IN_WORK	17	Die Wahrscheinlichkeit der Barcodeschwelle wird gesetzt
SET_COLOR_WEIGHTING_IN_WORK	18	Die Farbgewichtung wird gesetzt

Tab. 3-49: Aufzählungstyp MB_REGI_SENSOR01_COMM_STATUS

RMB_TechRegiExtended

Der Aufzählungstyp MB_REGI_SENSOR01_COMM_ACTIONS kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
SET_CONFIG	1	Übertragen der Konfiguration starten
SEARCH_BARCODE	2	Barcode Suche starten
SEARCH_MARK_STREAM	3	Markenstrom Suche starten
TEACH_IN	4	TeachIn starten
EASY_TEACH_IN	5	Easy TeachIn starten
READ_DIAG	6	Diagnosedaten lesen
READ_TEACHING_DATA	7	Teach Daten lesen
WRITE_TEACHING_DATA	8	Teach Daten schreiben
START_MEASURING	9	Messung starten
STOP_MEASURING	10	Messung stoppen
START_SEND_SPEED	11	Geschwindigkeitsinformationen senden starten
STOP_SEND_SPEED	12	Geschwindigkeitsinformationen senden stoppen
SET_GAIN	13	Gain übertragen
SET_MARK_ANALOG_LEVEL	14	Marken Analoglevel setzen
SET_BARCODE_SEARCH_LEVEL	15	Wahrscheinlichkeit der Barcodeschwelle setzen
SET_COLOR_WEIGHTING	16	Farbgewichtung setzen

Tab. 3-50: Aufzählungstyp MB_REGI_SENSOR01_COMM_ACTIONS

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
StartAction	BOOL			FALSE	Kontinuierlich
SelectAction	MB_REGI_SENSOR01_COMM_ACTIONS	1	16	1 (SET_CONFIG)	Bei steigender Flanke an StartAction und Status = WAIT_FOR_NEXT_ACTION
TimeSendInterval	TIME	$\geq 2 \cdot \text{CycleTime}$	n. def.	t#32ms	Aktion START_SEND_SPEED wird gestartet
CycleTime	TIME	> 0	n. def.	t#8ms	Aktion START_SEND_SPEED wird gestartet
Format	POINTER TO MB_REGI_FORMAT	> 0	n. def.	0	Steigende Flanke an Enable
MarkStructure	POINTER TO MB_REGI_MARK_STRUCTURE	> 0	n. def.	0	Steigende Flanke an Enable
Barcode	POINTER TO MB_REGI_BARCODE	> 0	n. def.	0	Steigende Flanke an Enable
MarkStream	POINTER TO MB_REGI_MARK_STREAM	> 0	n. def.	0	Steigende Flanke an Enable

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
LengthRegister-Mode	POINTER TO MB_REGI_LENGTH_REGISTER_MODE	> 0	n. def.	0	Steigende Flanke an Enable
SideRegisterMode	POINTER TO MB_REGI_SIDE_REGISTER_MODE	> 0	n. def.	0	Steigende Flanke an Enable
Communication-Config	POINTER TO MB_REGI_COMMUNICATION_CONFIG	> 0	n. def.	0	Steigende Flanke an Enable
Tolerances	POINTER TO MB_REGI_TOLERANCES	> 0	n. def.	0	Steigende Flanke an Enable
InternalParameter	POINTER TO MB_REGI_INTERNAL_PARAMETER_TYPE02	> 0	n. def.	0	Steigende Flanke an Enable

Tab. 3-51: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Es wird bei einer steigenden Flanke an Enable überprüft, ob alle Pointer > 0 sind und somit die entsprechenden Adressen, unter denen die Datenstrukturen liegen, angelegt sind. Die Daten der angelegten Pointer werden bei den Aufrufen der einzelnen Funktionsbausteine überprüft. Die Datenstrukturen besitzen die folgenden Standardwerte und werden bei den folgenden Aktionen jeweils übernommen bzw. geprüft:

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
MB_REGI_FORMAT					
FormatLength	REAL	> 0	n. def.	0	Steigende Flanke an Enable
Increments	UDINT	> 0	n. def.	0	Steigende Flanke an Enable
MB_REGI_MARK_STRUCTURE					
MarkType	UDINT	1	7	1 (Es sind die Markentypen 1,2,3,5,6,7 implementiert)	Aktion SET_CONFIG wird gestartet
SingleMark	ARRAY [0...3] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_BARCODE					
NumberOfBlocks	UDINT	0 (1 und 2 Blocks sind ungültige Eingaben)	10	0	Aktion SET_CONFIG wird gestartet
BeginExpectationWindow	REAL	0	n. def.	0	Aktion SET_CONFIG wird gestartet
SizeOfExpectationWindow	REAL	0	n. def.	1	Aktion SET_CONFIG wird gestartet
FreeSpaceBeforeBarcode	REAL	0	n. def.	0	Aktion SET_CONFIG wird gestartet
BarcodeBlocks	ARRAY [0...19] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Aktion SET_CONFIG wird gestartet

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
MB_REGI_MARK_STREAM					
NumberOfMarks	UDINT	1	12	1	Aktion SET_CONFIG wird gestartet
MarkStream	Array [1...12] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Aktion SET_CONFIG wird gestartet
MarkDirection	BOOL			FALSE	Aktion SET_CONFIG wird gestartet
MB_REGI_LENGTH_REGISTER_MODE					
ModeSelect	UDINT	1	1	1	Aktion SET_CONFIG wird gestartet
LengthProperties	ARRAY [0...4] OF REAL	n. def.	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_SIDE_REGISTER_MODE					
ModeSelect	UDINT	0	1	1	Aktion SET_CONFIG wird gestartet
SelectMarks	ARRAY [0...1] OF UINT	n. def.	n. def.	0	Aktion SET_CONFIG wird gestartet
MB_REGI_COMMUNICATION_CONFIG					
IpSensor	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortConfiguration	UINT	1	n. def.	0	Steigende Flanke an Enable
IpPLC	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortMeasurement	UINT	1	n. def.	0	Steigende Flanke an Enable
MB_REGI_TOLERANCES					
TeachZeroPulseToMark	REAL	1	n. def.	5	Aktion SET_CONFIG wird gestartet
TeachMarkToMark	REAL	1	n. def.	2	Aktion SET_CONFIG wird gestartet
MeasureMarkTolerance	REAL	1	n. def.	10	Aktion SET_CONFIG wird gestartet
DeviationOfMarkWidth	REAL	1	100	20	Aktion SET_CONFIG wird gestartet
DeviationOfMarkHight	REAL	1	100	100	Aktion SET_CONFIG wird gestartet
DeviationOfLastPosition	REAL	1	n. def.	10	Aktion SET_CONFIG wird gestartet
ShiftSyncPulse	REAL	1	n. def.	5	Aktion SET_CONFIG wird gestartet
MB_REGI_INTERNAL_PARAMETER_TYPE02					

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ColorWeighting	REAL	0	< 1	0.5	Aktion SET_CONFIG oder SET_COLOR_WEIGHTING wird gestartet
Gain	UINT	0	7	0	Aktion SET_CONFIG oder SET_GAIN wird gestartet
MarkAnalogLevel	UINT	0	n. def.	100	Aktion SET_CONFIG oder SET_MARK_ANALOG_LEVEL wird gestartet
BarcodeSearchLevel	REAL	0	< 100	50.0	Aktion SET_CONFIG oder SET_BARCODE_SEARCH_LEVEL wird gestartet

Tab. 3-52: Übersicht Min-/Max und Default-Werte der referenzierten Strukturen

Die VAR_IN_OUTs RegiSensorData und DiagData besitzen keine Min-/Max und Default-Werte, da sie nur von den internen Funktionsbausteinen beschrieben werden und nicht vom Anwender beschrieben werden dürfen. In TeachData werden beim Lesen der Teach Daten die entsprechenden Daten abgelegt. Beim Schreiben der Teach Daten werden die, z. B. aus einer vorherig abgespeicherten Produktion, angelegten Teach Daten zum Sensor geschrieben. Diese Datenstruktur ist nicht interpretierbar.

Funktionsbeschreibung



Für die Funktionsbeschreibung wird ebenfalls empfohlen, die einzelnen Beschreibungen der Funktionsbausteine zu beachten:

- [Kap. 3.3.8 "MB_RegiSensor01ConfigType02" auf Seite 87](#)
- [Kap. 3.3.10 "MB_RegiSensor01CycComType04" auf Seite 108](#)
- [Kap. 3.3.11 "MB_RegiSensor01ReadDiagType01" auf Seite 112](#)
- [Kap. 3.3.9 "MB_RegiSensor01SendRecvDataType02" auf Seite 105](#)
- [Kap. 3.3.12 "MB_RegiSensor01SendSpeedAccType01" auf Seite 115](#)
- [Kap. 3.3.13 "MB_RegiSensor01TeachDataType01" auf Seite 119](#)

Initialisierung

Bei einer steigenden Flanke an Enable werden zunächst die Pointer geprüft, ob sie angelegt wurden. Die Inhalte der Pointer der jeweiligen Datenstruktur werden direkt an die jeweiligen Funktionsbausteine angelegt und erst dann geprüft, wenn der zugehörige Funktionsbaustein aufgerufen wird. Anschließend wird der Funktionsbaustein MB_RegiSensor01SendRecvDataType gestartet, der die Kommunikation mit dem Markenstromsensor durchführt. Hier wird bereits durch das Kommando "Ping" überprüft ob der Markenstromsensor unter der vorgegebenen IP-Adresse zu erreichen ist. Im Anschluss wird die aktuell am Funktionsbaustein angelegte Konfiguration mittels des Funktionsbausteins MB_RegiSensor01ConfigType02 übertragen und der Funktionsbaustein MB_RegiSensor01CycComType04 zum Empfangen der Messwerte gestartet. Anschließend ist die Initialisierungsphase abgeschlossen und InOperation wird TRUE. Jeder in der Initialisierungsphase aufgetretene Fehler führt sofort zu einem Übergang in den Fehlerzustand und Error = TRUE gesetzt. Im zyklischen Betrieb wird zwischen Fehlern und Warnungen

RMB_TechRegiExtended

unterschieden. Während der Initialisierung befindet sich der Sensor im Zustand SENSOR_OFF.

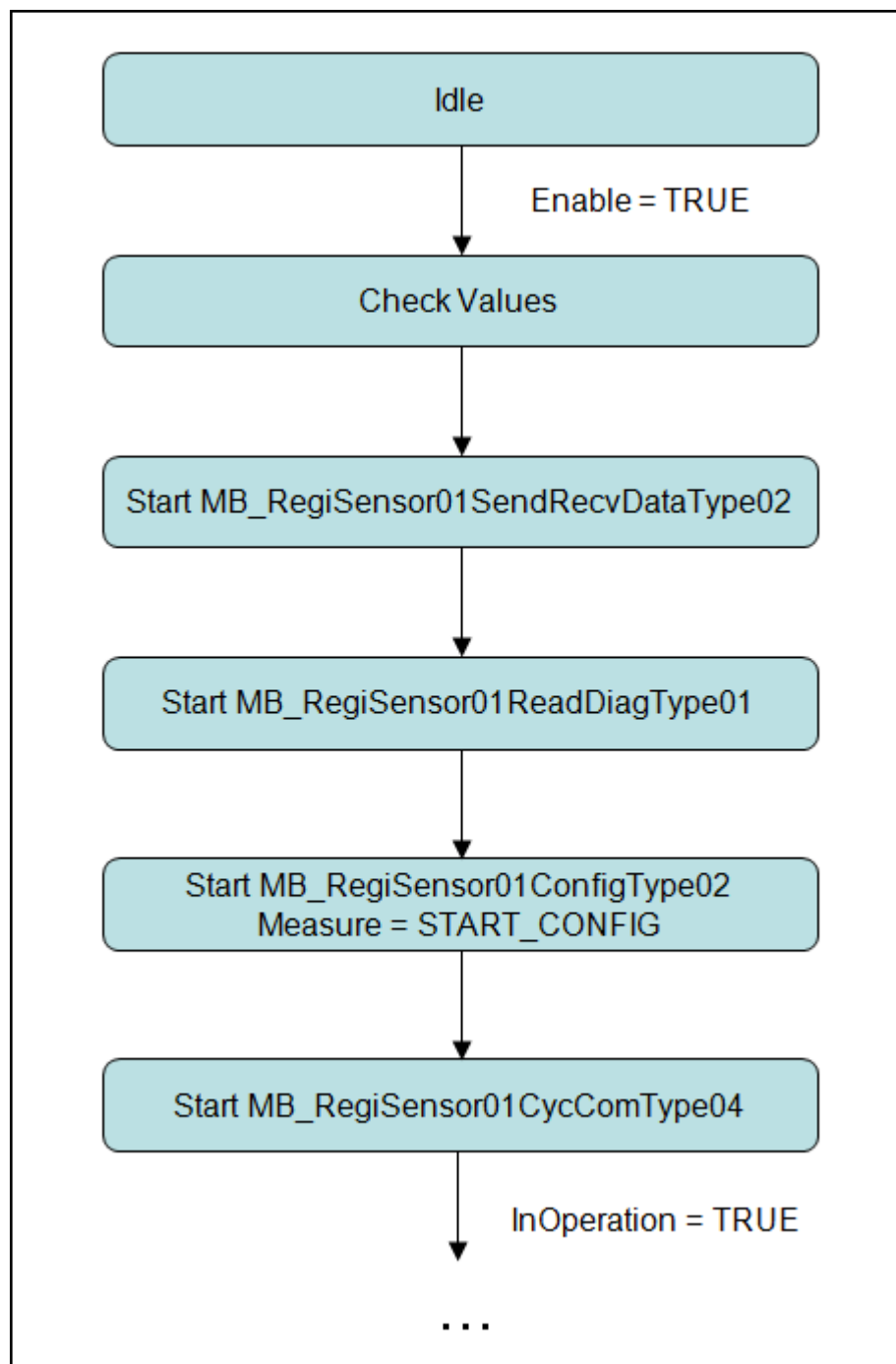


Abb. 3-33: Ablauf Initialisierungsphase



Es wird während der Initialisierung geprüft, ob auf dem Markenstromsensor noch eine Messung läuft. Ist dies der Fall wird diese Messung automatisch gestoppt.

Zyklischer Betrieb

Folgende Zustände können im zyklischen Betrieb erreicht werden:

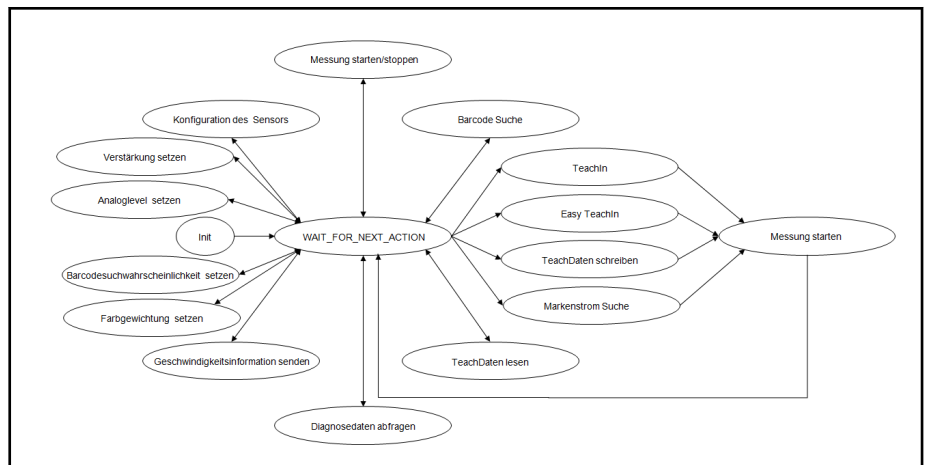


Abb. 3-34: Zustandsübergänge im zyklischen Betrieb

Im zyklischen Betrieb (d. h. InOperation = TRUE) befindet sich der Funktionsbaustein in einem Zustand "WAIT_FOR_NEXT_ACTION". In diesem Zustand können alle im Aufzählungstyp "MB_REGI_SENSOR01_COMM_ACTIONS" definierten Aktionen ausgeführt werden. Die entsprechende Aktion muss in SelectAction ausgewählt und anschließend mit einer steigenden Flanke an StartAction gestartet werden. Während eine Aktion läuft, kann über Status der entsprechende Zustand des Funktionsbausteins ausgelesen werden. Wird eine Aktion erfolgreich beendet, so wird dies über ActionDone angezeigt. Läuft bereits eine Aktion, so kann keine neue Aktion gestartet werden. Eine Aktion kann durch 3 unterschiedliche Ereignisse beendet werden. Wurde die Aktion erfolgreich ausgeführt, so wird dies mit ActionDone angezeigt. Ist ein schwerwiegender Kommunikationsfehler aufgetreten, so geht der Funktionsbaustein in den Error Zustand (Error = TRUE). Wurde eine Aktion nicht erfolgreich ausgeführt (z. B. wurden beim TeachIn nicht alle Marken gefunden), so gibt der Funktionsbaustein eine Warning (ActionWarning = TRUE) aus und kehrt in den Zustand "WAIT_FOR_NEXT_ACTION" zurück. Beim Starten einer erneuten Aktion wird die Warnung automatisch zurückgesetzt.

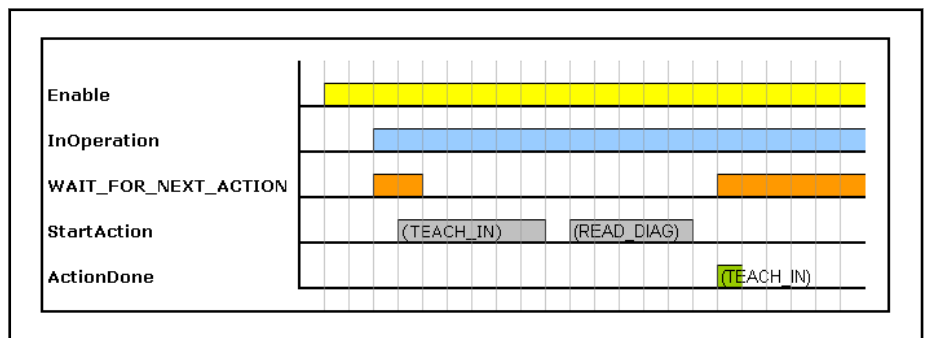


Abb. 3-35: Zeitdiagramm im zyklischen Betrieb

Das Zeitdiagramm zeigt, dass nach einer steigenden Flanke an Enable nach einigen Zyklen, das InOperation gesetzt und in den Zustand "WAIT_FOR_NEXT_ACTION" übergegangen wird. Eine steigende Flanke an StartAction mit SelectAction := TEACH_IN führt dazu, dass der Zustand "WAIT_FOR_NEXT_ACTION" verlassen wird und das Teachen durchgeführt wird. Eine erneute steigende Flanke an StartAction mit SelectAction := READ_DIAG wird dagegen ignoriert. Der Funktionsbaustein befindet sich zu diesem Zeitpunkt nicht im Status "WAIT_FOR_NEXT_ACTION", da die Aktion des Teachens bereits ausgeführt wird. Nach dem erfolgreichen Abschluss des Teachens wird ActionDone = TRUE für einen Zyklus gesetzt (da die stei-

gende Flanke an StartAction bereits weggenommen wurde) und der Funktionsbaustein kehrt zurück in den Zustand "WAIT_FOR_NEXT_ACTION".

Eine ausgeführte Aktion wird genau dann erfolgreich beendet, wenn das dazugehörige Acknowledgement ActionDone = TRUE kommt. Es ist nicht möglich, mehrere Operationen zeitgleich durchzuführen. Wurde die steigende Flanke am Eingang bereits weggenommen, so steht ActionDone am Ausgang genau 1 Zyklus an (siehe folgendes Zeitdiagramm).

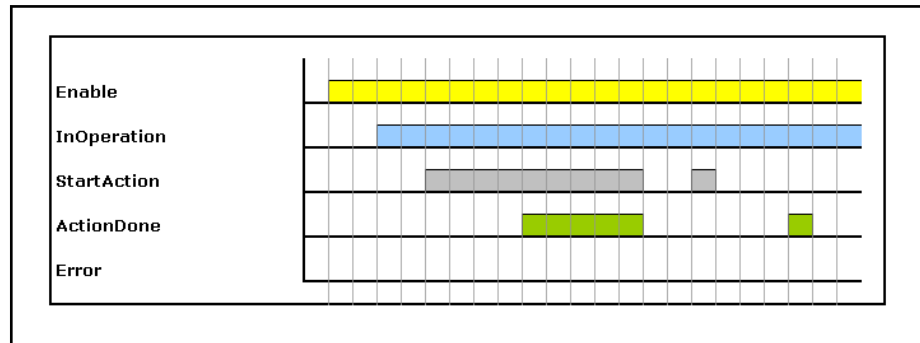


Abb. 3-36: Zeitdiagramm des Verhaltens von ActionDone

Da einige Aktionen (z. B. TeachIn) einige Zeit in Anspruch nehmen, besitzt der Funktionsbaustein den Ausgang "Status". Ist Status = 0 und InOperation = TRUE, so befindet sich der Funktionsbaustein im Zustand "WAIT_FOR_NEXT_ACTION". Hier kann die nächste Aktion ausgeführt werden. Der Status kann die Werte des Aufzählungstyps "MB_REGI_SENSOR01_COMM_STATUS" annehmen. Ist der Funktionsbaustein nicht in InOperation = TRUE, so ist der Status "SENSOR_OFF" außer er befindet sich im Fehlerzustand, dann ist der Status "SENSOR_ERROR".

Nach dem Starten oder Stoppen der Messung oder nach dem Starten oder Stoppen des Sendens der Geschwindigkeitsinformation wird der aktuelle Status an den Ausgängen MeasureInOp und SendSpeedInOp angezeigt. Somit wird kontinuierlich angezeigt, dass die Messung und das Übertragen der Geschwindigkeitsinformation laufen.

Vor dem Ausführen der Aktionen "SET_CONFIG", "SEARCH_BARCODE", "SEARCH_MARK_STREAM", "TEACH_IN" und "EASY_TEACH_IN" wird zunächst überprüft, ob eine Messung läuft. Ist dies der Fall wird die Messung zunächst gestoppt und anschließend die Aktion ausgeführt.

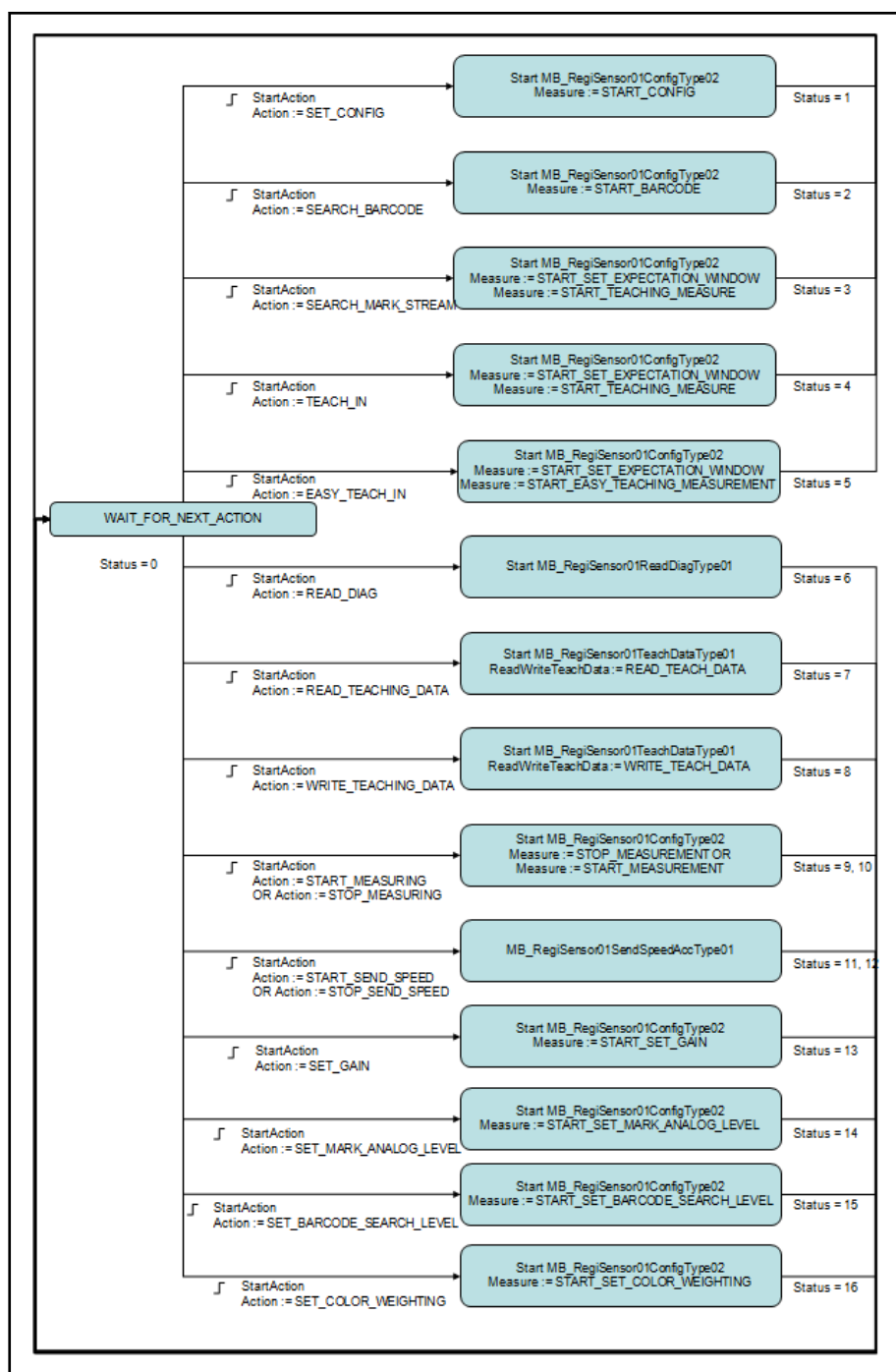


Abb. 3-37: Ablauf des zyklischen Betriebs

Die Abbildung zeigt den angedeuteten Ablauf des zyklischen Betriebs. Hier ist zu erkennen, dass ausgehend vom Zustand "WAIT_FOR_NEXT_ACTION" in die jeweiligen Aktionen gesprungen wird. Nach Abschluss der Aktion folgt wiederum die Rückkehr in den Ausgangszustand. Es kann natürlich (nicht eingezeichnet) auch in den Fehlerzustand gesprungen werden. Es wurden jeweils die für diese Aktionen grundlegenden Eingänge angezeichnet. In den Zuständen, können mehrere Funktionsbausteine aufgerufen werden. Beispielsweise wird bei einer steigenden Flanke an StartAction (SelectAction := TEACH_IN) zunächst der Funktionsbaustein MB_RegiSensor01ConfigType02 zum Setzen des Erwartungsfensters aufgerufen, um ihn anschließend nochmals zum Teachen und Starten der Messung aufzurufen.

RMB_TechRegiExtended

SEARCH_MARK_STREAM

Diese Funktion sucht sich selbständig den Markenstrom ohne Barcode Suche und ohne Angabe des Erwartungsfensters.

Folgende Voraussetzungen müssen für diese Funktion erfüllt sein:

- Auf der 360° Umdrehung die der Markenstromsensor abtastet befinden sich ausschließlich die Registermarken. D. h. es dürfen weder weitere Passerkreuze oder sonstige Kennzeichnungen in der Abtastspur des Sensors befinden.
- Der Registermarkenstrom darf maximal 2/3 des Druckzylinderformats ausmachen. Mindestens 1/3 des Druckzylinderformats müssen somit druckfreier Bereich sein. Beispiel: Befinden sich auf dem zu vermessenden Format 12 Registermarken im Abstand von 20 mm (Markenstrom der Größe 220 mm), so muss das Druckzylinderformat mindestens 330 mm betragen.

Ist eine der genannten Voraussetzungen verletzt, so kann die Funktion nicht ausgeführt werden oder wird mit einem Fehler beendet.



Die Funktion sucht sich mit maximal 3 Teach-Vorgängen selbständig den Markenstrom. Es muss nur eine gültige Konfiguration übertragen worden sein. Es ist keine Angabe des Erwartungsfensters nötig.

Verwendete Daten/Datenstrukturen

Nachdem nicht alle Eingänge bei jeder Aktion benötigt werden, führt die nachfolgende Tabelle die Daten auf, die bei den einzelnen Aktionen übernommen werden:

SelectAction	Verwendete Daten	Durch die Aktion veränderte Daten
SET_CONFIG	Format MarkStructure Barcode MarkStream LengthRegisterMode SideRegisterMode CommunicationConfig Tolerances InternalParameter	ActionDone Status
SEARCH_BARCODE	Barcode.NumberOfBlocks Barcode.FreeSpaceBeforeBarcode Barcode.BarcodeBlocks InternalParameter.Gain	ActionDone Status BarcodeSearchBeginOfExpWindow BarcodeFoundProbability
SEARCH_MARK_STREAM	-	ActionDone Status
TEACH_IN	Barcode.BeginExpectationWindow Barcode.SizeOfExpectationWindow	ActionDone Status RegiSensorData

RMB_TechRegiExtended

SelectAction	Verwendete Daten	Durch die Aktion veränderte Daten
EASY_TEACH_IN	Barcode.BeginExpectationWindow Barcode.SizeOfExpectationWindow InternalParameter.Gain	ActionDone Status RegiSensorData
READ_DIAG	-	ActionDone Status DiagData
READ_TEACHING_DATA	-	ActionDone Status TeachData
WRITE_TEACHING_DATA	TeachData	ActionDone Status
START_MEASURING	-	ActionDone MeasureInOp Status RegiSensorData
STOP_MEASURING	-	ActionDone MeasureInOp Status RegiSensorData
START_SEND_SPEED	TimeSendIntervall CycleTime Master	ActionDone SendSpeedInOp Status
STOP_SEND_SPEED	-	ActionDone SendSpeedInOp Status
SET_GAIN	InternalParameter.Gain	ActionDone Status
SET_MARK_ANALOG_LEVEL	InternalParameter.MarkAnalogLevel	ActionDone Status
SET_BARCODE_SEARCH_LEVEL	InternalParameter.BarcodeSearchLevel	ActionDone Status
SET_COLOR_WEIGHTING	InternalParameter.ColorWeighting	ActionDone Status

Tab. 3-53: Datenabhängigkeiten

Beispielhafte Abläufe

RMB_TechRegiExtended

In nachfolgender Abbildung wird der beispielhafte Ablauf von der Konfiguration zur Messung dargestellt für die Instanz fbRegiSensor01Comm des Funktionsbausteins MB_RegiSensor01CommType02:

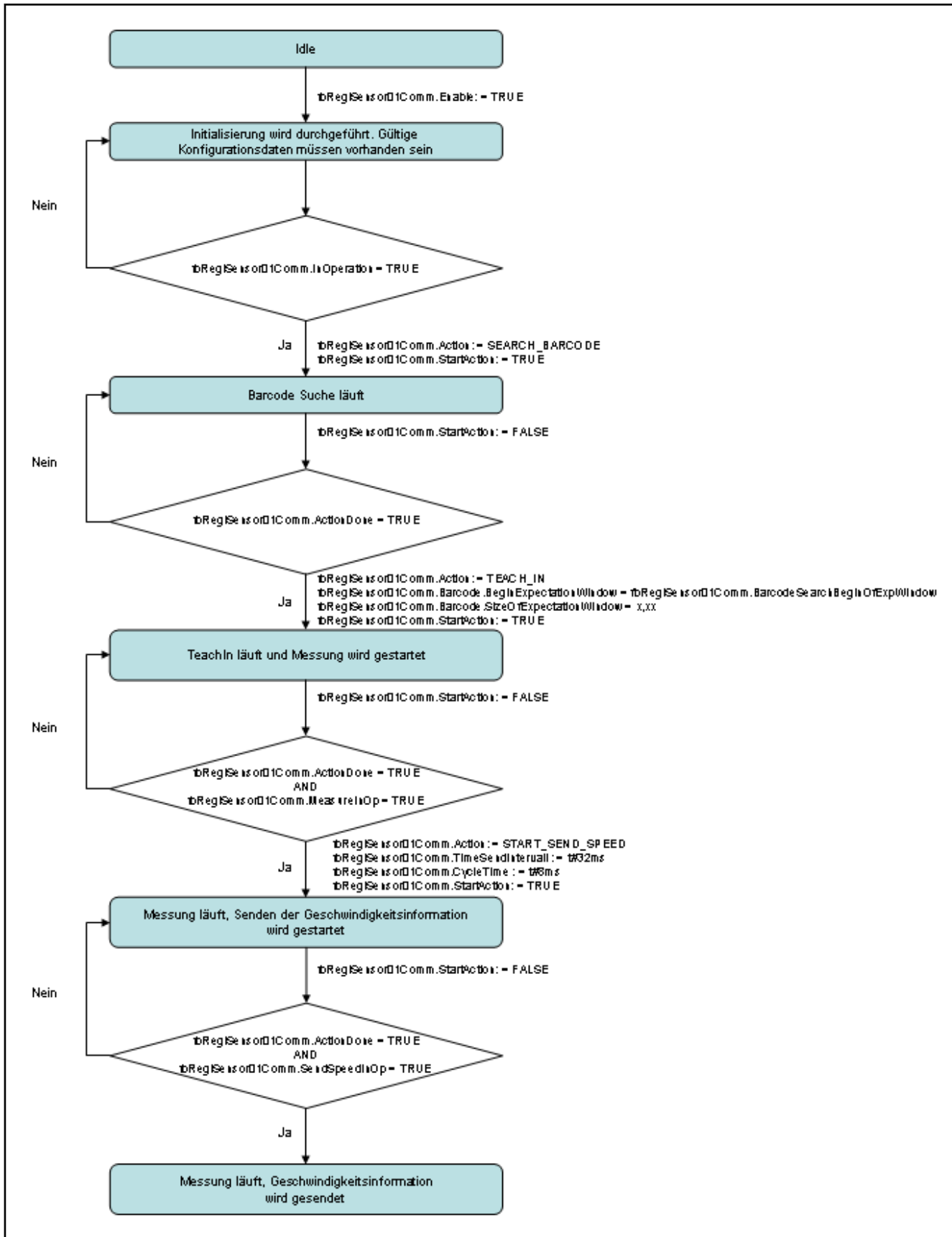


Abb. 3-38: Beispielhafter Ablauf von der Konfiguration zur Messung

Zunächst wird eine steigende Flanke an Enable der Instanz fbRegiSensor01Comm gesetzt. Hier stellt zunächst der Funktionsbaustein MB_RegiSensor01SendRecvDataType02 die Kommunikationsverbindung zum Sensor her. Anschließend wird die Konfiguration mit dem Funktionsbaustein MB_RegiSensor01ConfigType02 übertragen. Dies bedeutet es müssen gültige Daten anliegen. Weiterhin wird der Messbaustein MB_RegiSensor01CycComType04 gestartet und InOperation wird "TRUE". Daraufhin befindet sich der Funktionsbaustein im zyklischen Betrieb. Es wird die Suche des Barcodes gestartet. Nach dem Abschluss der Suche muss der zurückgelieferte Wert als Beginn des Erwartungsfensters zugewiesen werden und ein gültiger Wert für die Größe des Erwartungsfensters vorliegen. Es wird nun der TeachIn gestartet, bei dem zunächst das Erwartungsfenster gesetzt wird, dann das TeachIn gestartet wird um anschließend die Messung zu beginnen. Um eine höhere Genauigkeit in der Beschleunigungsphase zu erhalten wird anschließend noch das Übertragen der Geschwindigkeitsinformation mit dem Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 aktiviert. Daraufhin läuft die Messung und die Messdaten befinden sich in der Datenstruktur RegiSensorData.

Besonderheiten Warning und Error

Der Funktionsbaustein besitzt neben dem standardisierten Fehlerhandling eine zweite Diagnoseschnittstelle "MB_REGI_WARNING" mit dem Warnungen ausgegeben werden können. Die Notwendigkeit dieser zweiten Diagnosestruktur ergibt sich durch die Verschaltung zahlreicher Funktionsbausteine und dem angebunden externen Rexroth Mark Stream Sensor. Es findet somit eine Unterscheidung zwischen schwerwiegenden Fehlern (Error = TRUE und Errorzustand) und Fehlern die zwar von einem internen Funktionsbaustein ausgegeben werden, diese jedoch nicht den gesamten Funktionsbaustein MB_RegiSensor01CommType02 zurücksetzen sollen. Beispielsweise können bei einem TeachIn nicht alle Marken gefunden werden. Dies wird als Warnung ausgegeben, da dies kein schwerwiegender Fehler ist und der Anwender eine weitere Aktion daraufhin ausführen kann.

Eine Warnung wird wie folgt definiert:

- Warnungen werden nur im zyklischen Betrieb erzeugt.
- Eine Warnung wird gesetzt nachdem eine Aktion nicht erfolgreich ausgeführt wird.
- Es erfolgt keine Bestätigung durch ActionDone.
- Der Funktionsbaustein kehrt in den Zustand "WAIT_FOR_NEXT_ACTION" zurück.
- Die Warnung wird automatisch zurückgesetzt wenn eine erneute Aktion ausgeführt wird.
- In WarningAdditional1 steht das Additional 1 der Fehlerstruktur "ERROR_STRUCT" und WarningAdditional2 steht das Additional 2 der Fehlerstruktur "ERROR_STRUCT" des internen Funktionsbausteins.

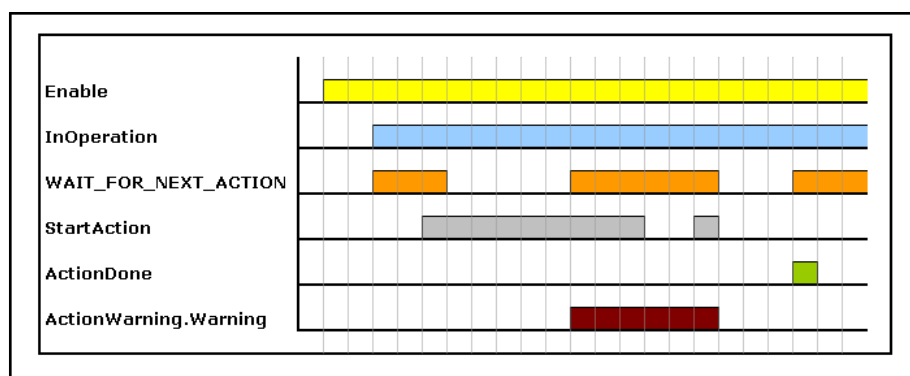


Abb. 3-39: Zeitverhalten der Warnung

Das Diagramm zeigt das zeitliche Verhalten beim Auftreten einer Warnung. Es wird eine Aktion ausgeführt im zyklischen Betrieb (InOperation=TRUE). Diese wird nicht erfolgreich beendet. Deshalb steht eine Warnung an. Der Funktionsbaustein kehrt zeitgleich mit der Warnung in den Zustand "WAIT_FOR_NEXT_ACTION" zurück. Nach dem Starten einer neuen Aktion wird die Warnung automatisch zurückgesetzt. Diese Aktion endet nun erfolgreich und wird mit ActionDone bestätigt.

Warning und Error bei der Initialisierung

Während der Initialisierung führt jeder von einem internen Funktionsbaustein erzeugte Fehler zu einem Fehler des Funktionsbausteins MB_RegiSensor01CommType02. D. h. es muss sowohl die Verbindung mit dem Sensor erneut hergestellt werden, als auch eine gültige Konfiguration übertragen und der Messbaustein gestartet werden.

Warning und Error im zyklischen Betrieb

Im zyklischen Betrieb führen ausschließlich Verbindungsfehler zu einem Error, da dies schwerwiegende Fehler sind. Hierzu zählen alle Fehler der Funktionsbausteine MB_RegiSensor01SendRecvDataType02 und MB_RegiSensor01CycComType04.

Weiterhin führen die folgenden Fehlernummern der anderen intern verwendeten Funktionsbausteine zu einem Übergang in den Error-Zustand:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000000	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvDataType02 innerhalb von 10 Sekunden
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000005	Timeout beim Empfangen eines Pakets.
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000006	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvDataType02 innerhalb von 10 Sekunden.

Tab. 3-54: Fehler der internen Funktionsbausteine, die in den Fehlerzustand führen

Alle anderen internen Fehler der Funktionsbausteine werden als Warnungen interpretiert. Tritt eine Warnung auf, wird der Additional1 und Additional2 der "ERROR_STRUCT" des internen Funktionsbausteins auf WarningAdditional1 und WarningAdditional2 kopiert sowie das Warning Bit gesetzt. Intern wird daraufhin der Funktionsbaustein bei dem der Fehler auftritt zurückgesetzt. Bei einem erneuten Ausführen einer Aktion erlischt die Warnung automatisch und WarningAdditional1 und WarningAdditional2 werden auf 0 zurückgesetzt.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000001	Suche des Markenstroms fehlgeschlagen
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000002	Messung ist schon gestartet
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000003	Messung ist schon gestoppt
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000004	Geschwindigkeitsinformationen senden bereits gestartet
OTHER_ERROR (16#7FFE)	16#00001D13	16#00000005	Geschwindigkeitsinformationen senden bereits gestoppt
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003C	Nullpointer Format
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003D	Nullpointer MarkStructure
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003E	Nullpointer Barcode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003F	Nullpointer MarkStream
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000040	Nullpointer LengthRegisterMode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000041	Nullpointer SideRegisterMode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000042	Nullpointer CommunicationConfig
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000043	Nullpointer Tolerances
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000044	Nullpointer InternalParameter

Tab. 3-55: Fehlercodes des Funktionsbausteins MB_RegiSensor01Comm-Type02

Weiterhin liefert der Funktionsbaustein Fehlermeldungen der integrierten Funktionsbausteine:

- [Kap. 3.3.8 "MB_RegiSensor01ConfigType02" auf Seite 87](#)
- [Kap. 3.3.10 "MB_RegiSensor01CycComType04" auf Seite 108](#)
- [Kap. 3.3.11 "MB_RegiSensor01ReadDiagType01" auf Seite 112](#)
- [Kap. 3.3.9 "MB_RegiSensor01SendRecvDataTypes02" auf Seite 105](#)
- [Kap. 3.3.12 "MB_RegiSensor01SendSpeedAccType01" auf Seite 115](#)
- [Kap. 3.3.13 "MB_RegiSensor01TeachDataType01" auf Seite 119](#)

Die entsprechenden Fehlermeldungen werden erst bei einer Aktivierung des jeweiligen Funktionsbausteins erzeugt. Hierzu muss die Übernahme der jeweiligen Werte beachtet werden. Es können somit Fehlermeldungen erst bei aktivem InOperation erzeugt werden. Weiterhin können diese Fehler, wie zuvor beschrieben, ebenfalls als Warnungen ausgegeben werden.

3.3.8 MB_RegiSensor01ConfigType02

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01ConfigType02 wird für die Konfiguration des Markenstromsensors ab Sensorfirmware 12 benötigt. Der Funktionsbaustein bereitet die Konfigurationsdaten auf und übergibt die Daten an

RMB_TechRegiExtended

den Funktionsbaustein MB_RegiSensor01SendRecvDataType02, der die Versendung des Pakets und den Empfang des Antwortpakets durchföhrt. Neben der Konfiguration kann das Erwartungsfenster gesetzt, der Teach-In und die Messung gestartet werden oder die Messung gestoppt werden. Er föhrt die Erstkonfiguration des Sensors durch und konfiguriert den Sensor während der Messung um.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V14	RMB_TechRegiExtended

Tab. 3-56: Referenztabelle Funktionsbaustein MB_RegiSensor01ConfigType02

Schnittstellenbeschreibung

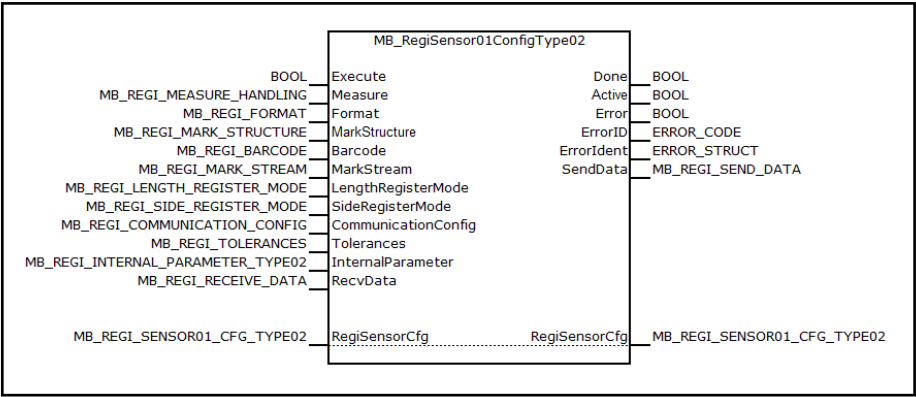


Abb. 3-40: Funktionsbaustein MB_RegiSensor01ConfigType02

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT	RegiSensorCfg	MB_REGI_SENSOR01_CFG_TYPE02	Enthält die aktuellen Konfigurationsdaten des Sensors
VAR_INPUT	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (flankengesteuert)
	Measure	MB_REGI_MEASURE_HANDLING	Überträgt die Konfiguration, setzt das Erwartungsfenster, startet das Teach-In und die Messung oder stoppt die Messung.
	Format	MB_REGI_FORMAT	Definiert die Formateigenschaften
	MarkStructure	MB_REGI_MARK_STRUCTURE	Definiert die Markenstruktur
	Barcode	MB_REGI_BARCODE	Definiert den Barcode
	MarkStream	MB_REGI_MARK_STREAM	Definiert den Markenstrom
	LengthRegister-Mode	MB_REGI_LENGTH_REGISTER_MODE	Definiert das Bahnmessverfahren für das Längsregister
	SideRegisterMode	MB_REGI_SIDE_REGISTER_MODE	Definiert das verwendete Seitenregistermessverfahren
	Communication-Config	MB_REGI_COMMUNICATION_CONFIG	Definiert die Kommunikationseigenschaften der UDP Verbindung
	Tolerances	MB_REGI_TOLERANCES	Definiert die zulässigen Toleranzen für das Teach-In und das Messen
	InternalParameter	MB_REGI_INTERNAL_PARAMETER_TYPE02	Interne Parameter des Sensors, die nur auf ausdrücklichen Hinweis geändert werden sollen.
	RecvData	MB_REGI_RECEIVE_DATA	Empfange Daten vom Funktionsbaustein MB_RegiSensor01SendRecvDataType02
VAR_OUTPUT	Done	BOOL	Datenausgänge sind gültig
	Active	BOOL	Funktionsbaustein arbeitet und befindet sich in einem Zwischenzustand
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	SendData	MB_REGI_SEND_DATA	Daten die über den Funktionsbaustein MB_RegiSensor01SendRecvData gesendet werden sollen

Tab. 3-57: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01ConfigType02

Datenstrukturen

Folgende Datenstrukturen werden vom Funktionsbaustein MB_RegiSensor01ConfigType02 verwendet:

MB_REGI_SENSOR01_CFG_TYPE02		
CurrentConfig	USINT	Aktuelle Konfiguration
Format	MB_REGI_FORMAT	Beschreibt das Druckzylinderformat
MarkStructure	MB_REGI_MARK_STRUCTURE	Beschreibt die Markenstruktur
Barcode	MB_REGI_BARCODE	Beschreibt den Barcode

RMB_TechRegiExtended

MB_REGI_SENSOR01_CFG_TYPE02		
MarkStream	MB_REGI_MARK_STREAM	Beschreibt den Markenstrom
LengthRegisterMode	MB_REGI_LENGTH_REGISTER_MODE	Längsregistermodus
SideRegisterMode	MB_REGI_SIDE_REGISTER_MODE	Seitenregistermodus
CommunicationData	MB_REGI_COMMUNICATION_CONFIG	Die Kommunikationsports der UDP Verbindung
Tolerances	MB_REGI_TOLERANCES	Teach- und Messtoleranzen
InternalParameter	MB_REGI_INTERNAL_PARAMETER_TYPE02	Interne Sensor Parameter
BarcodeSearchBeginOfExpWindow	REAL	Gibt den Beginn des Erwartungsfensters aus der Barcode-Suche zurück
BarcodeFoundProbability	REAL	Gibt den Wahrscheinlichkeitswert des Barcodes aus der Barcode-Suche zurück

Tab. 3-58: Datenstruktur MB_REGI_SENSOR01_CFG_TYPE02

MB_REGI_FORMAT		
FormatLength	REAL	Druckzylinderformat in [mm]
Increments	UDINT	Anzahl der Inkremente pro Umdrehung

Tab. 3-59: Datenstruktur MB_REGI_FORMAT

MB_REGI_MARK_STRUCTURE		
MarkType	UDINT	Markentyp
SingleMark	ARRAY [0...3] OF REAL	Beschreibt die Druckmarke

Tab. 3-60: Datenstruktur MB_REGI_MARK_STRUCTURE

MB_REGI_BARCODE		
NumberOfBlocks	UDINT	Anzahl der Barcode Blöcke (0 bedeutet, dass kein Barcode vorhanden ist)
BeginExpectationWindow	REAL	Anfang des Erwartungsfensters (Entfernung des Beginns des Erwartungsfensters zum Nullimpuls) [mm]
SizeOfExpectationWindow	REAL	Größe des Erwartungsfensters [mm]
FreeSpaceBeforeBarcode	REAL	Druckfreier Raum vor dem Barcode [mm]
BarcodeBlocks	ARRAY [0...19] OF REAL	Barcode Beschreibung in [mm]

Tab. 3-61: Datenstruktur MB_REGI_BARCODE

MB_REGI_MARK_STREAM		
NumberOfMarks	UINT	Anzahl der Marken
MarkStream	ARRAY [1...12] OF REAL	Abstand der Marken [mm]
MarkDirection	BOOL	Erste oder zweite Markenflanke für das Längsregister

Tab. 3-62: Datenstruktur MB_REGI_MARK_STREAM

MB_REGI_LENGTH_REGISTER_MODE		
ModeSelect	UDINT	Ausgewählter Längsregistermodus
LengthProperties	ARRAY [0...4] OF REAL	Eigenschaften für den Längsregistermodus [mm]

Tab. 3-63: Datenstruktur MB_REGI_LENGTH_REGISTER_MODE

MB_REGI_SIDE_REGISTER_MODE		
ModeSelect	UDINT	Ausgewählter Seitenregistermodus
MarkSelect	ARRAY [0...1] OF UINT	Marken für das Seitenregister

Tab. 3-64: Datenstruktur MB_REGI_SIDE_REGISTER_MODE

MB_REGI_COMMUNICATION_CONFIG		
IpSensor	STRING	IP Adresse des Sensors
PortConfiguration	UINT	Port für die azyklische Kommunikation (Konfiguration, Diagnose, Geschwindigkeitsdaten)
IpPLC	STRING	IP Adresse der SPS
PortMeasurement	UINT	Port für die Messergebnisse

Tab. 3-65: Datenstruktur MB_REGI_COMMUNICATION_CONFIG

MB_REGI_TOLERANCES		
TeachZeroPulseToMark	REAL	Teach Toleranz der ersten Marke zum Nullimpuls [mm]
TeachMarkToMark	REAL	Teach Toleranz zwischen den einzelnen Marken [mm]
MeasureMarkTolerance	REAL	Messtoleranz im Vergleich zum nächsten Messwert [mm]
DeviationOfMarkWidth	REAL	Abweichung der Markenbreite zum letzten Messwert [%]
DeviationOfMarkHight	REAL	Abweichung der Markenhöhe zum letzten Messwert [%]
DeviationOfLastPosition	REAL	Absolute Abweichung der Messtoleranz vom letzten Messwert [mm]
ShiftSyncPulse	REAL	Der Nullimpuls wird genau um ShiftSyncPulse [mm] vor das Erwartungsfenster verschoben

Tab. 3-66: Datenstruktur MB_REGI_TOLERANCES

MB_REGI_INTERNAL_PARAMETER_TYPE02		
ColorWeighting	REAL	Farbgewichtung des Sensors. Die Intensitätsgewichtung berechnet sich anschließend automatisch aus (1-ColorWeighting)
Gain	UINT	Verstärkung des Sensors
MarkAnalogLevel	UINT	Der Analoglevel der Marke muss beim TeachIn über dieser Schwelle liegen, damit die Marke als gültige Marke erkannt wird.
BarcodeSearchLevel	REAL	Barcodewahrscheinlichkeitsschwelle. Liegt die Barcodewahrscheinlichkeit über dieser Schwelle wurde der Barcode erfolgreich gefunden. Ansonsten schlägt die Barcodesuche fehl.

Tab. 3-67: Datenstruktur MB_REGI_INTERNAL_PARAMETER_TYPE02

MB_REGI_SEND_DATA		
NoOfBytes	UDINT	Anzahl der Bytes, die versendet werden sollen
ValueAddr	POINTER TO DWORD	Pointer auf die Daten

RMB_TechRegiExtended

MB_REGI_SEND_DATA		
NewData	UINT	Zeigt neue Daten zum Senden an
SensorPort	UINT	Sensor Port, auf den die Daten gesendet werden sollen

Tab. 3-68: Datenstruktur MB_REGI_SEND_DATA

MB_REGI_RECEIVE_DATA		
NoOfBytes	UDINT	Anzahl der Bytes, die empfangen wurden
ValueAddr	POINTER TO DWORD	Pointer auf die Daten
NewData	UINT	Zeigt neue empfangene Daten an

Tab. 3-69: Datenstruktur MB_REGI_RECEIVE_DATA

Aufzählungstyp

Der Aufzählungstyp MB_REGI_MEASURE_HANDLING wird für das Konfigurieren, das Starten der Messung, des Teach-Ins, sowie zum Setzen des Erwartungsfensters verwendet und kann folgende Werte annehmen:

Element	Wert	Beschreibung
START_CONFIG	0	Startet die Konfiguration
START_TEACHING	1	Startet das Teachen
START_MEASUREMENT	2	Startet die Messung
STOP_MEASUREMENT	3	Stoppt die Messung
START_CONFIG_TEACHING	4	Startet die Konfiguration und das Teachen
START_TEACHING_MEASUREMENT	5	Startet das Teachen und die Messung
START_CONFIG_TEACHING_MEASUREMENT	6	Startet die Konfiguration, das Teachen und die Messung
START_SET_EXPECTATION_WINDOW	7	Setzt das Erwartungsfenster
START_EASY_TEACHING	8	Startet einen Easy-Teach-In. Es müssen die Verstärkung sowie Beginn und Größe des Erwartungsfensters übertragen werden.
START_EASY_TEACHING_MEASUREMENT	9	Startet den Easy-Teach-In und anschließend die Messung (vgl. 8)
START_BARCODE	10	Startet die Barcode-Suche
START_SET_GAIN	11	Startet das Setzen des Gains
START_SET_MARK_ANALOG_LEVEL	12	Startet das Setzen des Markenanaloglevels
START_SET_BARCODE_SEARCH_LEVEL	13	Startet das Setzen des Barcodesuchlevels
START_SET_COLOR_WEIGHTING	14	Startet das Setzen der Farbgewichtung
START_SET_DATE	15	Startet das Setzen des Datums

Tab. 3-70: Aufzählungstyp MB_REGI_MEASURE_HANDLING

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
MB_REGI_SENSOR01_CFG_TYPE02					
CurrentConfig	USINT	0	255	0	Steigende Flanke an Execute

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
MB_REGI_FORMAT					
FormatLength	REAL	> 0	n. def.	0	Steigende Flanke an Execute
Increments	UDINT	> 0	n. def.	0	Steigende Flanke an Execute
MB_REGI_MARK_STRUCTURE					
MarkType	UDINT	1	7	1 (Es sind die Marken 1,2,3,5,6,7 implementiert)	Steigende Flanke an Execute
SingleMark	ARRAY [0...3] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Steigende Flanke an Execute
MB_REGI_BARCODE					
NumberOfBlocks	UDINT	0 (1 und 2 Blocks sind ungültige Eingaben)	10	0	Steigende Flanke an Execute
BeginExpectationWindow	REAL	0	n. def.	0	Steigende Flanke an Execute
SizeOfExpectationWindow	REAL	0	n. def.	1	Steigende Flanke an Execute
FreeSpaceBeforeBarcode	REAL	0	n. def.	0	Steigende Flanke an Execute
BarcodeBlocks	ARRAY [0...19] OF REAL	> 0 (für benötigte Eingaben)	n. def.		Steigende Flanke an Execute
MB_REGI_MARK_STREAM					
NumberOfMarks	UDINT	1	12	1	Steigende Flanke an Execute
MarkStream	Array [1...12] OF REAL	> 0 (für benötigte Eingaben)	n. def.	0	Steigende Flanke an Execute
MarkDirection	BOOL			FALSE	Steigende Flanke an Execute
MB_REGI_LENGTH_REGISTER_MODE					
ModeSelect	UDINT	1	1	1	Steigende Flanke an Execute
LengthProperties	ARRAY [0...4] OF REAL	n. def.	n. def.	0	Steigende Flanke an Execute
MB_REGI_SIDE_REGISTER_MODE					
ModeSelect	UDINT	0	1	1	Steigende Flanke an Execute
SelectMarks	ARRAY [0...1] OF UINT	n. def.	n. def.	0	Steigende Flanke an Execute
MB_REGI_COMMUNICATION_CONFIG					
IpSensor	STRING	LEN > 0	n. def.	""	Steigende Flanke an Execute
PortConfiguration	UINT	1	n. def.	0	Steigende Flanke an Execute
IpPLC	STRING	LEN > 0	n. def.	""	Steigende Flanke an Execute
PortMeasurement	UINT	1	n. def.	0	Steigende Flanke an Execute

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
MB_REGI_TOLERANCES					
TeachZeroPulseToMark	REAL	1	n. def.	5	Steigende Flanke an Execute
TeachMarkToMark	REAL	1	n. def.	2	Steigende Flanke an Execute
MeasureMarkTolerance	REAL	1	n. def.	10	Steigende Flanke an Execute
DeviationOfMarkWidth	REAL	1	100	20	Steigende Flanke an Execute
DeviationOfMarkHight	REAL	1	100	20	Steigende Flanke an Execute
DeviationOfLastPosition	REAL	1	n. def.	10	Steigende Flanke an Execute
ShiftSyncPulse	REAL	1	n. def.	5	Steigende Flanke an Execute
MB_REGI_INTERNAL_PARAMETER_TYPE02					
ColorWeighting	REAL	0	< 1	0.5	Steigende Flanke an Execute
Gain	UINT	0	7	0	Steigende Flanke an Execute
MarkAnalogLevel	UINT	0	n. def.	100	Steigende Flanke an Execute
BarcodeSearchLevel	REAL	0	<100	50.0	Steigende Flanke an Execute
MB_REGI_RECEIVE_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAddr	POINTER TO BYTE	n. def.	n. def.	0	Änderung an NewData
NewData	UINT	0	65535	0	Kontinuierlich

Tab. 3-71: Übersicht Min/Max-Werte und Initialwerte der Eingänge



Als gültige Eingaben werden nur bisher im Sensor implementierte Funktionalitäten zugelassen.

Funktionsbeschreibung

Bei der azyklischen Kommunikation arbeitet der Sensor als Server, d. h. die Steuerung sendet Daten an den Sensor und der Sensor antwortet. Mit der azyklischen Kommunikation wird der Sensor konfiguriert.

Die Struktur RegiSensorCfg enthält die aktuelle Konfiguration des Sensors. Sollen Konfigurationsänderungen vorgenommen werden, so können die geänderten Konfigurationsdaten direkt an den Baustein angelegt werden. Sie werden erst bei einer steigenden Flanke von Execute übernommen. Im Folgenden werden die einzelnen Strukturen beschrieben:

RegiSensorCfg:

Enthalten die aktuellen Konfigurationsdaten des Sensors und die aktuelle Konfigurations-ID, die in den zyklischen Messdaten übertragen werden. Außerdem wird nach einer Barcode-Suche der ermittelte Beginn des Erwartungsfensters angegeben.

Measure: *Bei einer steigenden Flanke an Execute wird zwischen 11 unterschiedlichen Abläufen unterschieden, die je nach gewählter Funktion ausgeführt werden:*

- **START_CONFIG:**

Hier wird eine neue Konfiguration an den Sensor übertragen. Das Erwartungsfenster wird hier selbständig durch den Sensor über die Markenabstände und die Toleranzen, sowie den Beginn des Erwartungsfensters berechnet.



Hierzu sind sehr exakte Eingaben erforderlich. Es wird empfohlen, den Beginn und die Größe des Erwartungsfensters über eine grafische Oberfläche (HMI) zu ermitteln und diese Werte mit der Option **START_SET_EXPECTATION_WINDOW** zu übertragen bzw. die Barcode-Suche zu verwenden.

- **START_TEACHING:**

Startet das Teachen des Sensors. Hierzu werden die zuletzt an den Sensor übertragenen Konfigurationsdaten und das Erwartungsfenster übernommen und die Marken mit ihren Schwellwerten im Erwartungsfenster ermittelt. In diesem Modus ermittelt der Sensor zusätzlich die ideale Verstärkung.

- **START_MEASUREMENT:**

Startet eine Messung, ohne eine Konfiguration zu übertragen oder ein Teach-In zu starten.

- **STOP_MEASUREMENT:**

Beendet eine laufende Messung.

- **START_CONFIG_TEACHING:**

Überträgt zunächst die Konfiguration und startet im Anschluss daran einen Teach-In. Wurde in `SizeOfExpectationWindow` eine Zahl größer Null angegeben, so wird dieser Wert für die Berechnung des Erwartungsfensters übernommen. Ist der Wert gleich Null, so wird das Erwartungsfenster selbständig durch den Sensor über die Markenabstände und die Toleranzen, sowie den Beginn des Erwartungsfensters berechnet.

- **START_TEACHING_MEASUREMENT:**

Startet das Teachen des Sensors und direkt im Anschluss daran die Messung. Für das Teachen werden die zuletzt an den Sensor übertragenen Konfigurationsdaten und das Erwartungsfenster übernommen.

- **START_CONFIG_TEACHING_MEASUREMENT:**

Überträgt zunächst die Konfiguration und startet im Anschluss daran einen Teach-In und danach die Messung. Wurde in `SizeOfExpectationWindow` eine Zahl größer Null angegeben, so wird dieser Wert für die Berechnung des Erwartungsfensters übernommen. Ist der Wert gleich Null, so wird das Erwartungsfenster selbständig durch den Sensor über die Markenabstände und die Toleranzen, sowie den Beginn des Erwartungsfensters berechnet.

- **START_SET_EXPECTATION_WINDOW:**

Setzt nur das Erwartungsfenster, das beispielsweise aus der grafischen Oberfläche übermittelt wurde. Hierzu muss in `SizeOfExpectationWindow` ein Wert größer Null angegeben werden, sowie der Beginn des Erwartungsfensters (`BeginExpectationWindow`).



Wird nach dem Übertragen des Erwartungsfensters eine neue Konfiguration gestartet, so wird das übertragene Erwartungsfenster überschrieben. Aus diesem Grund wird empfohlen, zunächst die Konfiguration und anschließend das Erwartungsfenster zu übertragen.

- **START_EASY_TEACHING:**

Es wird ein Teach-In gestartet, ohne dass die Verstärkung ermittelt wird. Es wird die Verstärkung, die unter InternalParameters in Gain steht, übernommen. Diese Verstärkung ist idealerweise nach einem erstmaligen normalen Teach-In mit dem Funktionsbaustein MB_RegiSensor01ReadDiagType01 ausgelesen worden. Weiterhin müssen die Größe und der Beginn des Erwartungsfensters angegeben werden.

- **START_EASY_TEACHING_MEASUREMENT:**

Wie START_EASY_TEACHING, jedoch wird nach dem Teachen sofort die Messung gestartet.

- **START_BARCODE:**

Wird dieser Modus ausgeführt, so wird die Barcode-Suche gestartet. Hierzu müssen vorab 3-10 Barcodes definiert sein. Nach erfolgreicher Barcode-Suche steht die Position direkt nach dem Barcode, d. h. der Beginn des Erwartungsfensters in RegiSensorCfg.BarcodeSearchBeginOfExpWindow. Dieser Wert kann zur Definition des Erwartungsfensters verwendet werden.

- **START_SET_GAIN**

Wird dieser Modus ausgeführt, so kann der Gain gesetzt werden. Der Gain kann somit einzeln übertragen werden ohne dass eine komplette Konfiguration übergeben werden muss.

- **START_SET_MARK_ANALOG_LEVEL**

Wird dieser Modus ausgeführt, so kann der Markenanaloglevel gesetzt werden. Der Parameter kann somit einzeln übertragen werden ohne dass eine komplette Konfiguration übergeben werden muss.

- **START_SET_BARCODE_SEARCH_LEVEL**

Wird dieser Modus ausgeführt, so kann der Barcodesuchlevel gesetzt werden. Der Parameter kann somit einzeln übertragen werden ohne dass eine komplette Konfiguration übergeben werden muss.

- **START_SET_COLOR_WEIGHTING**

Wird dieser Modus ausgeführt, so kann die Farbgewichtung gesetzt werden. Der Parameter kann somit einzeln übertragen werden ohne dass eine komplette Konfiguration übergeben werden muss.

- **START_SET_DATE**

Wird dieser Modus ausgeführt, so kann das Datum gesetzt werden. Der Parameter kann somit einzeln übertragen werden ohne dass eine komplette Konfiguration übergeben werden muss. Das Datum ist ein Basisparameter des Sensors und muss beispielsweise vor dem Verbinden des ActiveXControls an den Sensor übertragen worden sein.

In nachfolgender Abbildung wird der beispielhafte Ablauf von der Konfiguration des Markenstromsensors bis zur Messung dargestellt.

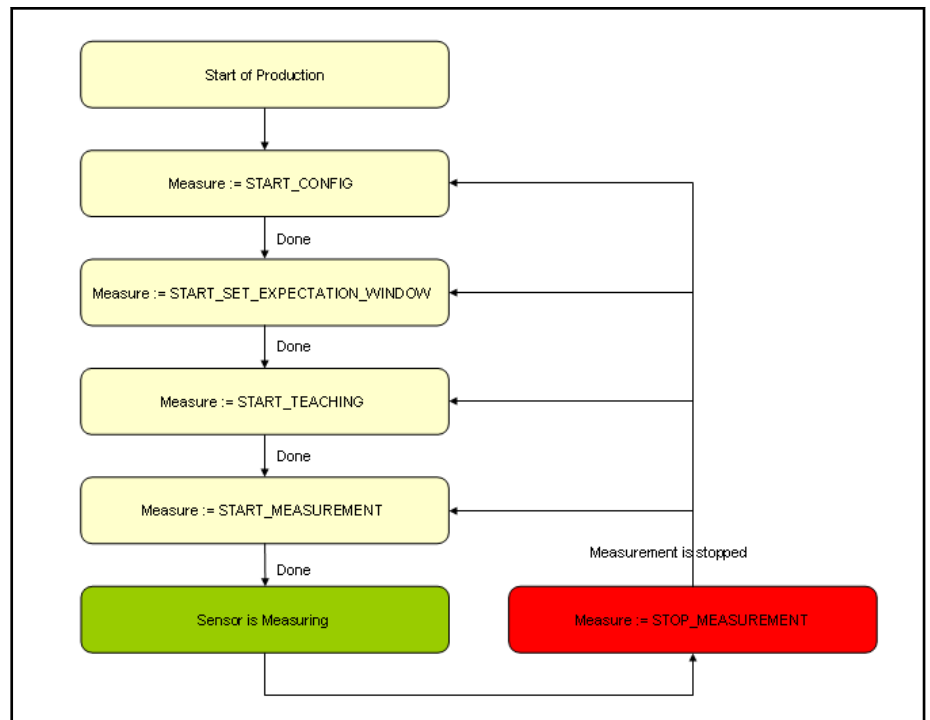


Abb. 3-41: Ablauf von der Konfiguration bis zur Messung

Der Sensor muss zunächst konfiguriert werden. Anschließend muss das Erwartungsfenster gesetzt werden, was vorzugsweise über eine Visualisierung bzw. die hierüber ermittelten Werte geschehen soll. Danach kann das Teachen und das Messen gestartet werden. Nachdem die Messung gestartet wurde, kann sie beliebig gestoppt werden und bei dem gewünschten Sensor-Zustand fortgefahren werden. Die Measure-Funktionalitäten können in den beliebigen Kombinationen aufgerufen werden.

MarkStructure: Enthält eine vordefinierte Markenstruktur und die Abmessung der Marken.

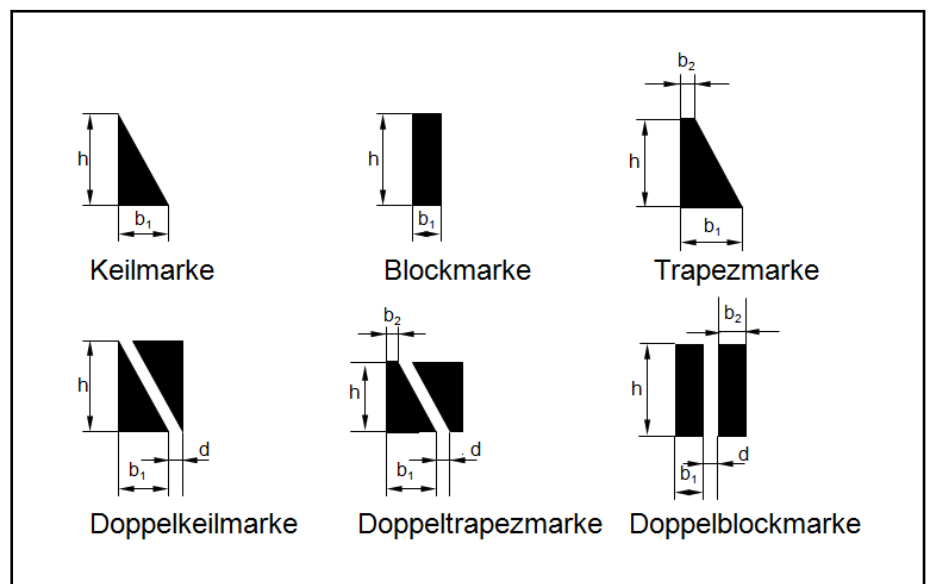


Abb. 3-42: Unterstützte Markentypen

RMB_TechRegiExtended

Der Parameter MarkType kann wie folgt gesetzt werden und die Parameter müssen nachfolgend in SingleMark je nach Markentyp angegeben werden:

	MarkType	SingleMark			
Keilmärke	1	Breite b_1 [mm]	Höhe h [mm]		
Doppelkeilmärke	2	Breite b_1 [mm]	Höhe h [mm]	Abstand d [mm]	
Blockmärke	3	Breite b_1 [mm]	Höhe h [mm]		
Trapezmärke	5	Breite b_1 [mm]	Höhe h [mm]	Breite b_2 [mm]	
Doppeltrapezmärke	6	Breite b_1 [mm]	Höhe h [mm]	Abstand d [mm]	Breite b_2 [mm]
Doppelblockmärke	7	Breite b_1 [mm]	Höhe h [mm]	Abstand d [mm]	Breite b_2 [mm]

Tab. 3-72: Parameter der Markentypen

Nachfolgend wird ein Beispiel der Konfiguration einer Dreiecksmärke angegeben:

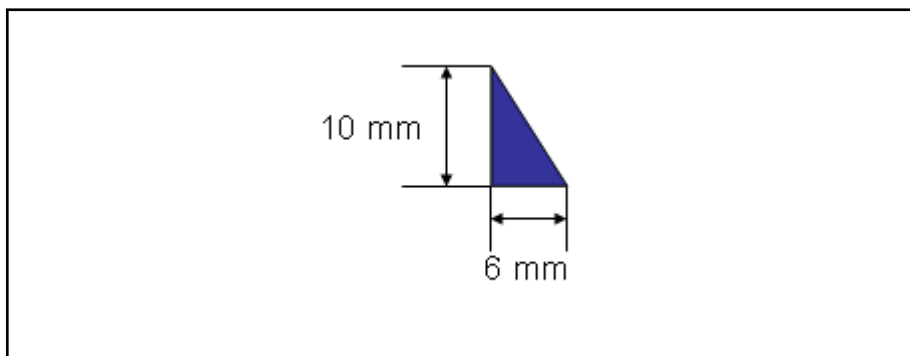


Abb. 3-43: Beispiel Markenstruktur

Code-Beispiel:

```
MarkStruct: MB_REGI_MARK_STRUCTURE := (MarkType:=1, SingleMark:= 6, 10, 0, 0);
```

Es handelt sich hier um eine Keilmärke (Markentyp 1). Diese wird durch ihre Höhe und die Breite definiert.

Barcode: Enthält die Anzahl der Barcode-Blöcke und die Beschreibung des Erwartungsfensters. Es sind bis zu 10 Barcode-Blöcke möglich. Ist NumberOfBlocks = 0, so ist kein Barcode vorhanden. Hier muss der Beginn und die Größe des Erwartungsfensters durch den Anwender gesetzt werden. Dies kann entweder über die Ermittlung dieser Daten über die Oszilloskopdarstellung erfolgen oder über die selbständige Berechnung der Größe des Erwartungsfensters durch den Sensor (aus Anzahl der Marken und deren Abstand, sowie den Toleranzen und dem Beginn des Erwartungsfensters). Hierzu muss Barcode.SizeOfExpectationWindow = 0 gesetzt werden.

Ist ein Barcode vorhanden, so kann sich der Sensor selbständig den Beginn des Erwartungsfensters ermitteln, indem er die Position des Barcode-Endes zurückliefert. Hierzu muss START_BARCODE gestartet werden. Liefert der Funktionsbaustein ein Done, so steht in RegiSensorCfg.BarcodeSearchBeginOfExpWindow die Position des Barcodeendes. Der Anwender muss jetzt noch die Größe des Erwartungsfensters angeben und kann nachfolgend mit START_SET_EXPECTATION_WINDOW das Erwartungsfenster setzen. Anschließend kann der normale Ablauf mit Teach-In und Messung starten fortgesetzt werden.

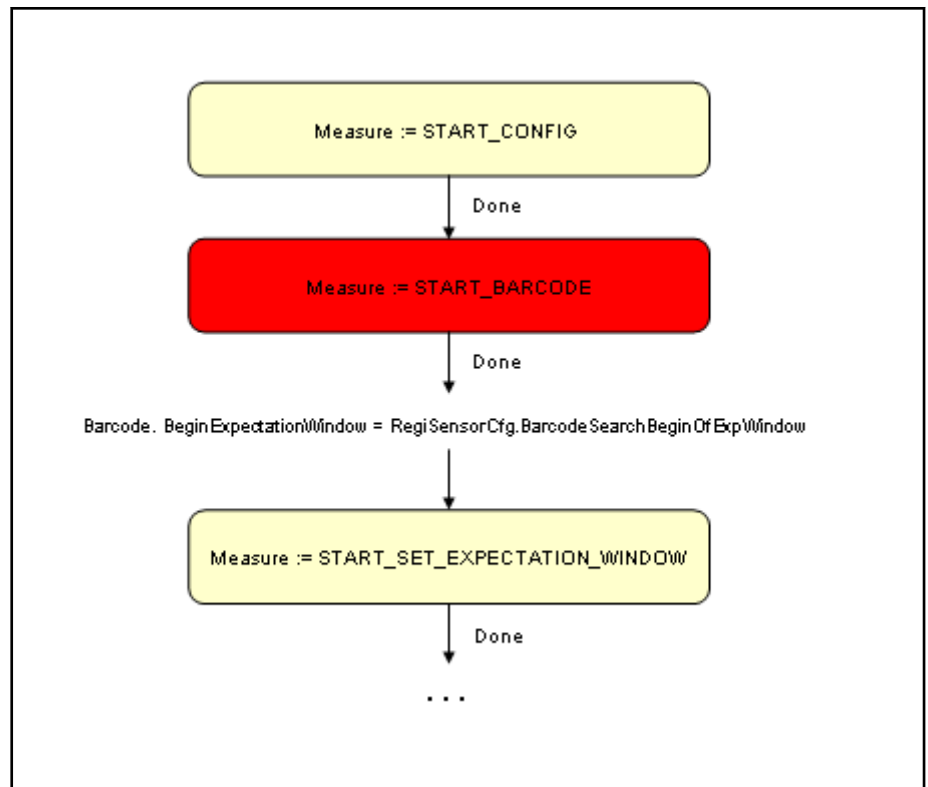


Abb. 3-44: Ablauf der Barcode-Suche

Ist ein Barcode vorhanden, so sucht sich der Sensor selbständig den Beginn des Erwartungsfensters. `FreeSpaceBeforeBarcode` gibt den druckfreien Raum vor dem Barcode an und `BarcodeBlocks` definiert die einzelnen Barcode Blöcke. Weiterhin müssen die Abmessungen des Barcodes wie folgt angegeben werden:

Beispiel für einen Barcode:

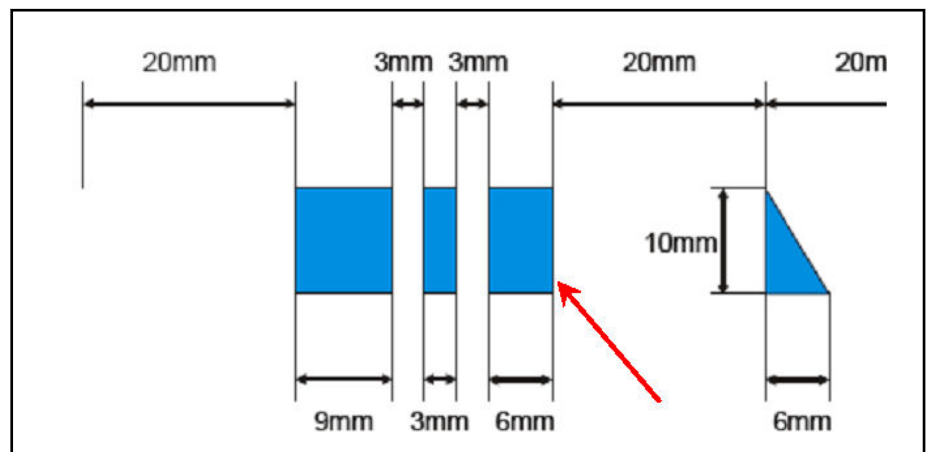


Abb. 3-45: Beispiel für einen Barcode

Die Abmessungen stehen im Array `BarcodeBlocks` in [mm]:

`BarcodeBlocks[0]`: Breite des Barcode Blocks 1 in [mm]

`BarcodeBlocks[1]`: Abstand des Barcode Blocks 1 zum Block 2 in [mm]

`BarcodeBlocks[2]`: Breite des Barcode Blocks 2 in [mm]

`BarcodeBlocks[3]`: Abstand des Barcode Blocks 2 zum Block 3 in [mm]

.....

Der letzte angegebene Abstand ist der Abstand vom letzten Barcode Block zur ersten Marke (in diesem Beispiel beträgt der Abstand zwischen Barcode Block und erster Registermarke 20 mm). Mit dem roten Pfeil ist die Position gekennzeichnet, die der Sensor nach erfolgreicher Barcode-Suche in Regi-SensorCfg.BarcodeSearchBeginOfExpWindow zurückliefert, also direkt die Position nach dem Barcode.

MarkStream: Enthält die Anzahl der Marken und den jeweiligen Abstand zwischen den Marken. Sind 5 Marken vorhanden, so gibt es 4 Markenabstände und der erste definierte Abstand ist der Abstand zwischen Marke 1 und Marke 2.

Beispiel für einen Markenstrom:

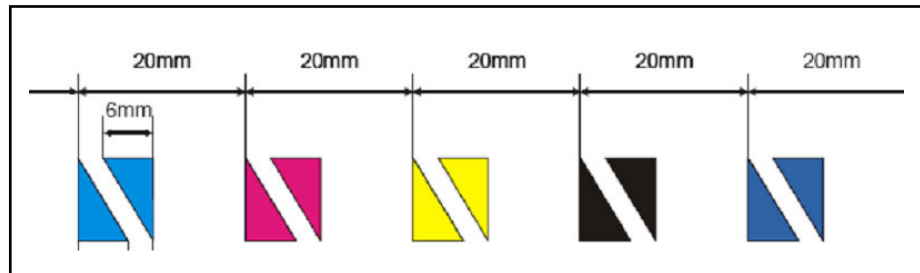


Abb. 3-46: Beispiel für einen Markenstrom

Beispiel für den dargestellten Markenstrom:

```
MarkStream: MB_REGI_MARK_STREAM := (NumberOfMarks:=5,
MarkStream:=20,20,20,20);
```

Der Markenstrom besteht aus 5 Marken. Diese besitzen jeweils den Abstand von 20 mm.

Weiterhin kann bei der Definition des Markenstroms angegeben werden, ob die senkrechte Markenflanke an erster oder zweiter Stelle kommt.

Ist MarkDirection = FALSE wird der Längsregisterwert in Bezug auf die jeweils erste Markenflanke zurückgegeben. Also der Abstand vom Nullimpuls zur ersten Markenflanke der Marke.

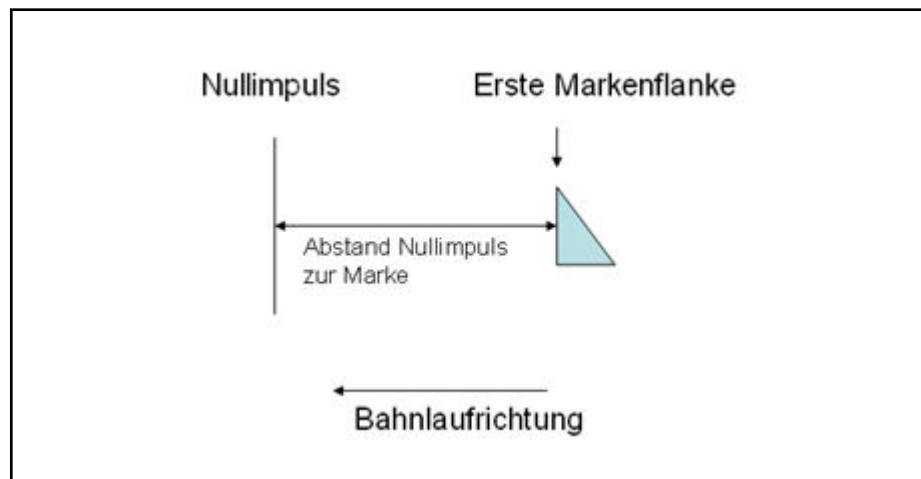


Abb. 3-47: Erste Markenflanke

Ist MarkDirection = TRUE wird der Längsregisterwert in Bezug auf die jeweils zweite Markenflanke zurückgegeben.

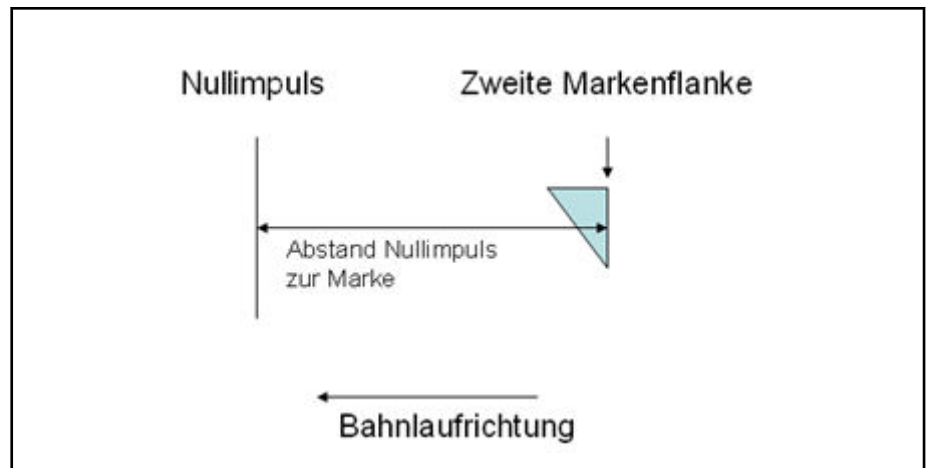


Abb. 3-48: Zweite Markenflanke



Es wird empfohlen, **immer** die senkrechte Flanke für die Längsregisterregelung zu verwenden. Wird die schräge Flanke der Marke verwendet, so würde eine Verstellung im Seitenregister immer eine Verschiebung im Längsregister bedeuten.

LengthRegisterMode: Gibt das entsprechende Bahnmessverfahren für das Längsregister an. Weiterhin sind alle benötigten Abmessungen für das entsprechende Bahnmessverfahren gespeichert. Im Moment ist im Markenstromsensor nur der Modus 1 der Rückgabe aller Markenabstände implementiert.

Beispiel:

```
LengthRegisterMode: MB_REGI_LENGTH_REGISTER_MODE := (ModeSelect:=1);
```

ModeSelect := 1 bedeutet, dass alle Markenabstände aller Marken zurückgegeben werden.

SideRegisterMode: Gibt an, welche Markenbreiten welcher Marken für das Seitenregister vermessen werden sollen. Ist ModeSelect := 0, werden keine Markenbreiten zurückgegeben. Ist ModeSelect := 1, werden alle Markenbreiten zurückgegeben.

Beispiel 1:

```
SideRegisterMode: MB_REGI_SIDE_REGISTER_MODE := (ModeSelect:=0);
```

Es werden keine Markenbreiten zurückgegeben.

Beispiel 2:

```
SideRegisterMode: MB_REGI_SIDE_REGISTER_MODE := (ModeSelect:=1);
```

Es werden alle Markenbreiten zurückgegeben.

Es steht im Moment noch keine Funktion zur Verfügung, zwei Marken, deren Markenbreiten ausgegeben werden sollen, anzugeben.

CommunicationConfig: Enthält die Kommunikationseinstellungen für die azyklische und die zyklische Kommunikation. Hierbei werden der Port auf der SPS für die Messwerte und der Port für Konfigurations- und Diagnosedaten angegeben, sowie die IP-Adresse des Sensors und der SPS.

Beispiel:

RMB_TechRegiExtended

```
CommunicationCfg: MB_REGI_COMMUNICATION_CONFIG := (IpSensor:='192.168.2.5', PortConfiguration:=6001,
IpPLC:='192.168.2.3', PortMeasurement:=6000);
```

Im obigen Beispiel wird eine Konfiguration der Verbindungsdaten aufgezeigt. Der Sensor hat die IP Adresse 192.168.2.5 und die azyklische Kommunikation erfolgt über den UDP-Port 6001 der Steuerung. Die SPS hat die IP-Adresse 192.168.2.3. Somit weiß der Sensor, dass er die zyklischen Messergebnisse an diese IP 192.168.2.3 und den Udp-Port 6000 der Steuerung senden soll.

Tolerances:

Bei den Toleranzen sollten die entsprechenden Default-Werte verwendet und nur im Ausnahmefall geändert werden.

Tolerances gibt Toleranzen für das Teachen, das Messen und für die Robustheit der Messung an:

- TeachZeroPulseToMark:
Diese Toleranz wird nicht mehr berücksichtigt. Immer Default-Wert verwenden.
- TeachMarkToMark:
Diese Toleranz soll gleich der MeasureMarkTolerance gesetzt werden.
- MeasureMarkTolerance:
Während einer Messung wird für jede Marke ein eigenes Markenerwartungsfenster verwendet. D. h. die Marke wird in Bezug auf Ihre letzte Position genau in diesem Erwartungsfenster gesucht. Hat sich die Position der Marke seit der letzten Umdrehung weiter verändert, so wird die Marke nicht gefunden.

Die nachfolgenden 3 Parameter dienen der Robustheit der Messung. D. h. eine Marke wurde bei der letzten Messung mit einer bestimmten Breite, Höhe und Position erfasst. Weicht bei der nächsten Messung die gefundene Marke um den entsprechenden Prozentsatz von der absoluten Position ab, so wird diese Messung ignoriert und erneut mit den bestehenden Daten der letzten gültigen Messung gesucht. Dies soll verhindern, dass eine einmalig auftretende Störung (z. B. Falte) eine Marke bei der nächsten Messung unauffindbar macht.
- DeviationOfMarkWidth:
Maximale prozentuale zugelassene Abweichung der Markenbreite von der letzten Messung. Wird dieser Wert auf 100 % gesetzt, so wird die Überwachung der Markenbreite abgeschaltet.
- DeviationOfMarkHight:
Maximale prozentuale zugelassene Abweichung der Markenhöhe (Analogwert) von der Markenhöhe (Analogwert) des Teachergebnisses. Wird dieser Wert auf 100 % gesetzt, so wird die Überwachung der Markenhöhe abgeschaltet.
- DeviationOfLastPosition:
Maximale absolute zugelassene Abweichung der Markenposition von der letzten Messung
- ShiftSyncPulse:
Gibt die Entfernung vom verschobenen Nullimpuls bis zum Beginn des Erwartungsfensters an. Um eine hohe Messgenauigkeit während der Beschleunigungsphase zu gewährleisten, wird der Nullimpuls hardwaremäßig bis zu dem Beginn des Erwartungsfensters verschoben. Dieser

Wert gibt die Toleranz zwischen verschobenem Nullimpuls und Beginn des Erwartungsfensters an. Ist zu erwarten, dass der Markenstrom nicht weiter nach vorne wandert, wird ein Default-Wert von 5 mm empfohlen.

InternalParameter: Diese Datenstruktur enthält interne Parameter des Sensors, die nur auf Anweisung von Bosch Rexroth geändert werden sollten. Hier kann die Gewichtung des Farb- und des Intensitätsanteils vorgenommen werden. Die Default-Werte sind gut geeignete Erfahrungswerte. Weiterhin enthält die Struktur die Verstärkung, die beim Easy-Teach-In übertragen wird.

- **ColorWeighting:**

Gibt die Farbgewichtung im Analogsignal an. Die Intensitätsgewichtung ergibt sich anschließend aus (1-ColorWeighting). D. h. soll der Sensor als reiner Farbsensor verwendet werden, ist der Parameter ColorWeighting auf 0,99 zu setzen.

- **Gain:**

Die Verstärkung des Sensors kann hier mit 0-7 manuell gesetzt werden. Der Sensor ermittelt bei einem TeachIn selbständig den optimalen Verstärkungswert. Deshalb sollte ein manuelles Setzen nur verwendet werden, wenn die Oszilloskopdaten nicht korrekt dargestellt werden bei der ersten Anzeige der 360° Daten.

- **MarkAnalogLevel:**

Der MarkAnalogLevel gibt die Schwelle an, ab der eine Registermarke beim TeachIn als Registermarke erkannt wird. Hiermit soll vermieden werden, dass eine Registermarke im Rauschen gefunden wird. Wird der MarkAnaloglevel auf 500 gesetzt, so muss beim Teachen das Dach der Registermarke im Mittel einen Analogwert von > 500 haben. In diesem Fall wird die Registermarke erkannt, ansonsten wird die Registermarke ignoriert. Weiterhin bezieht sich der MarkAnalogLevel ebenfalls auf den Barcode. D. h. die definierten Barcodeblöcke müssen im Mittel den MarkAnalogLevel überschreiten.

- **BarcodeSearchLevel:**

Der BarcodeSearchLevel gibt an ab welcher Wahrscheinlichkeit ein Barcode erkannt wird. Ein Barcode wird nach einem definierten Muster gesucht. Als Endergebnis erhält man die Barcodeposition sowie eine Wahrscheinlichkeit mit der der Barcode mit dem definierten Muster übereinstimmt. Ist die Wahrscheinlichkeit geringer als der BarcodeSearchLevel, so wird der Barcode nicht erkannt. Der Parameter BarcodeFoundProbability gibt im Anschluss an eine Barcodesuche die Wahrscheinlichkeit des gefundenen Barcodes an. Von daher kann dieser Parameter für die Festlegung einer neuen Barcodesuchschwelle verwendet werden.

Senden und Empfangen der Datenpakete

Der Funktionsbaustein zeigt durch die Inkrementierung von NewData in SendData an, dass er neue Daten versenden will. Das Versenden geschieht durch den Funktionsbaustein MB_RegiSensor01SendRecvDataType02. Empfängt dieser neue Daten, so setzt er den Wert von NewData in RecvData gleich. Somit weiß der Funktionsbaustein, dass es sich um das erwartete Antwortpaket handelt. Ein Antwortpaket wird bei jedem übertragenen Datenpaket erwartet.

Timeout:

Wird ein neues Datenpaket gesendet, so läuft ein interner Timer im Funktionsbaustein mit. Bekommt er innerhalb von 10s keine Rückantwort, so bricht er aufgrund einer Zeitüberschreitung mit einem Fehler ab.

RMB_TechRegiExtended

Ein möglicher Fehlerfall könnte sein, dass der Funktionsbaustein MB_Regi-Sensor01SendRecvDataType02 nicht aktiviert ist bzw. fehlerhaft verbunden ist.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in **Additional 1/Additional 2** folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000000	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvDataType02 innerhalb von 10 Sekunden
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000001	Timeout beim Teachen. Das Teachen konnte nicht in der angegebenen Zeit abgeschlossen werden.
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Formzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000001	Ungültiger Markentyp
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000002	Ungültige Markengeometrie
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000003	Ungültige Anzahl der Barcodeblöcke
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000004	Ungültige Anzahl an Marken im Markenstrom
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000005	Ungültiger Markenabstand
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000006	Ungültiger Längsregistermodus
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000007	Ungültige Längsregisterparameter
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000008	Ungültiger Seitenregistermodus
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000009	Ungültige Sensor IP
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000A	Ungültige Steuerungs IP
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000B	Ungültiger UDP-Port für die Messung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000C	Ungültiger Konfigurationsport
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000D	Ungültige Anzahl an Inkrementen
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000E	Ungültige Messtoleranz
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000F	Ungültige Teachtoleranz einer Marke
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000010	Ungültige Teachtoleranz Erwartungsfensterbeginn
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000011	Ungültiger Beginn des Erwartungsfensters
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000012	Ungültige Größe des Erwartungsfenster
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000013	Ungültige Toleranz für die Markenbreite
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000014	Ungültige Toleranz für die Markenhöhe
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000015	Ungültige Toleranz für die Markenposition
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000016	Ungültige ShiftSyncPulse Toleranz
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000017	Ungültige Farbgewichtung

ErrorID	Additional 1	Additional 2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000018	Ungültige Intensitätsgewichtung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000019	Ungültiger Measurement Modus
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001A	Ungültige Verstärkung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003B	Ungültiger Freiraum vor dem Barcode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000004D	Ungültiger BarcodeSearchLevel
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000004E	Ungültiger MarkAnalogLevel
COMMUNICATION_ERROR (16#0002)	16#0C001D10	16#00000002	Timeout bei der Barcode-Suche
OTHER_ERROR (16#7FFE)	16#00001D12	16#xxxxxxx	Fehlerbehandlung für den Markenstromsensor (siehe Tabelle Markenstromsensor) nicht in „F_RELATED_TABLE“ integriert

Tab. 3-73: Fehlercodes des Funktionsbausteins MB_RegiSensor01ConfigType02

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors ausgeben.

3.3.9 MB_RegiSensor01SendRecvDataType02

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01SendRecvDataType02 führt die Kommunikation mit dem Sensor durch, falls eine Sensorfirmware ≥ 12 verwendet wird. Hierzu übergeben die Funktionsbausteine für die Konfiguration des Sensors, für das Senden von Geschwindigkeits- und Beschleunigungsinformationen und der Diagnosebaustein während einer Beschleunigungsphase ihre zu sendenden Daten und bekommen die entsprechenden Datenpakete zurück. Der Funktionsbaustein verhindert eine Kollision von gleichzeitig auf ein und denselben UDP-Port zugreifenden Funktionsbausteinen. Weiterhin ist eine Priorisierung implementiert. Das Senden der Beschleunigungsinformationen ist zeitkritisch. Aus diesem Grund werden Datenpakete des Funktionsbausteins MB_RegiSensor01SendSpeedAccType01 bevorzugt gegenüber Konfigurationsdaten des Funktionsbausteins MB_RegiSensor01ConfigType02, Diagnosedaten des Funktionsbausteins MB_RegiSensor01ReadDiagType01 oder Teach-Daten des Funktionsbausteins MB_RegiSensor01TeachDataType01 behandelt und sofort verschickt, sobald das Senden möglich ist.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V14	RMB_TechRegiExtended

Tab. 3-74: Referenztabelle Funktionsbaustein MB_RegiSensor01SendRecvDataType02

Schnittstellenbeschreibung

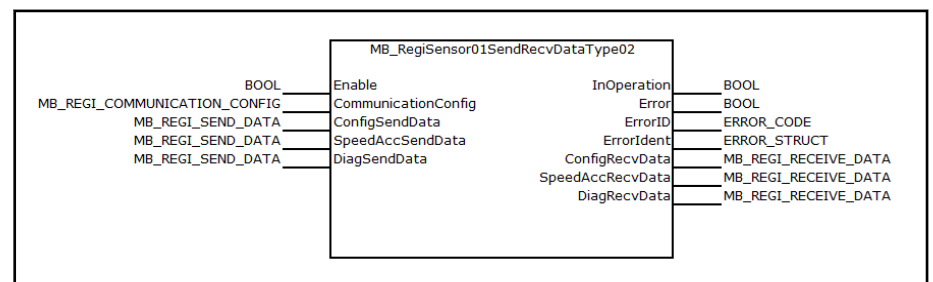


Abb. 3-49: Funktionsbaustein MB_RegiSensor01SendRecvDataType02

RMB_TechRegiExtended

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins.
	CommunicationConfig	MB_REGI_COMMUNICATION_CFG	Enthält den UDP Port der Steuerung auf den sich der Sensor verbindet und an den der Sensor die Messdaten schickt.
	ConfigSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01ConfigType02 versenden will.
	SpeedAccSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 versenden will.
	DiagSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01ReadDiagType01 versenden will.
	TeachSendData	MB_REGI_SEND_DATA	Enthält Daten, die der Funktionsbaustein MB_RegiSensor01TeachDataType01 versenden will.
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detailldiagnose
	ConfigRecvData	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01ConfigType02 empfangen wurden.
	SpeedAccRecvData	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 empfangen wurden.
	DiagRecvData	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01ReadDiagType01 empfangen wurden.
	TeachRecvData	MB_REGI_RECEIVE_DATA	Enthält Daten, die für den Funktionsbaustein MB_RegiSensor01TeachDataType01 empfangen wurden.

Tab. 3-75: E/A-Schnittstelle des Funktionsbausteins MB_RegiSensor01SendRecvDataType02

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
MB_REGI_COMMUNICATION_CONFIG					
IpSensor	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortConfiguration	UINT	1	n. def.	0	Steigende Flanke an Enable
IpPLC	STRING	LEN > 0	n. def.	""	Steigende Flanke an Enable
PortMeasurement	UINT	1	n. def.	0	Steigende Flanke an Enable
MB_REGI_SEND_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAddr	POINTER TO BYTE	n. def.	n. def.	0	Änderung an NewData

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
NewData	UINT	0	65535	0	Kontinuierlich
SensorPort	UINT	1	65535	0	Änderung an NewData

Tab. 3-76: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

Die verwendeten Datenstrukturen [Tab. 3-68 "Datenstruktur MB_REGI_SEND_DATA" auf Seite 91](#) und [Tab. 3-65 "Datenstruktur MB_REGI_COMMUNICATION_CONFIG" auf Seite 91](#) wurden beim Funktionsbaustein [Kap. 3.3.8 "MB_RegiSensor01ConfigType02" auf Seite 87](#) beschrieben.

Funktionsbeschreibung

Soll eine Kommunikation mit einem Sensor aufgebaut werden, so muss dieser Funktionsbaustein vor oder zeitgleich mit den angehängten Funktionsbausteinen MB_RegiSensor01SendSpeedAccType01, MB_RegiSensor01ConfigType02, MB_RegiSensor01TeachDataType01 und MB_RegiSensor01ReadDiagType01 mit einer steigenden Flanke an Enable gestartet werden. Bei einer steigenden Flanke an Enable, werden nach der Prüfung der Eingänge der entsprechende UDP-Port PortConfiguration auf der Steuerung gebunden und somit können auf diesem Port entsprechende Daten empfangen werden. Der Funktionsbaustein wartet nun auf Daten. D. h. er überprüft bei jedem Aufruf, ob sich der Eingang NewData bei der Struktur ConfigSendData, AccelerationSendData, DiagSendData und TeachSendData ändert. Ist dies der Fall, so werden die entsprechenden Daten (NoOfBytes) versendet. Der Funktionsbaustein wartet anschließend auf die entsprechenden Daten, falls ein Antwortpaket vom Sensor erwartet wird (Ausnahme: Bei dem Übertragen der Beschleunigung und der Geschwindigkeit wird kein Antwortpaket erwartet). Es findet auch ein Timeout Handling statt. Ist nach 2 Sekunden kein Antwortpaket empfangen worden, so wird die Sendung des Datenpakets wiederholt. Nach dem dritten Fehlversuch wird ein Fehler erzeugt und die Übertragung abgebrochen. Wird ein Paket empfangen, so wird die Anzahl der Bytes und die Adresse (ValueAddr) angegeben, an der die Daten liegen und NewData am Ausgang wird auf den gleichen Wert gesetzt, den NewData am Eingang beim Senden der Daten hatte. Somit kann exakt zugeordnet werden zu welchem gesendeten Paket die Antwort gehört.

Datenpuffer: Es ist kein **Datenpuffer** intern implementiert, da jeder angelegte Funktionsbaustein (falls ein Handshake stattfindet) das nächste Paket erst dann versendet, wenn er die entsprechende Antwort erhalten hat.

Ausnahme: Liegt ein neues Paket mit Geschwindigkeits- und Beschleunigungsinformation vor und kann dieses nicht gesendet werden bis ein neues Paket erzeugt wird, so wird das neue Paket versendet und das alte verworfen.

Priorisierung:

Die übertragenen Konfigurationsdaten sind nicht zeitkritisch. Einzig die Werte der Beschleunigung und der Geschwindigkeit müssen schnellstmöglich übertragen werden. Aus diesem Grund ist eine Priorisierung implementiert. Das Datenpaket des Funktionsbausteins MB_RegiSensor01SendSpeedAccType01 hat absoluten Vorrang und wird sofort gesendet, sobald der "Senden"-Funktionsbaustein frei wird. Wird in dieser Zeit auf ein Antwortpaket der Konfiguration oder Diagnose gewartet, so wird dies zunächst unterbrochen, das zeitkritische Paket gesendet, um anschließend das Konfigurations- bzw. Diagnoseantwortpaket zu empfangen. Die Priorisierung ist in nachfolgender Abbildung dargestellt.

RMB_TechRegiExtended

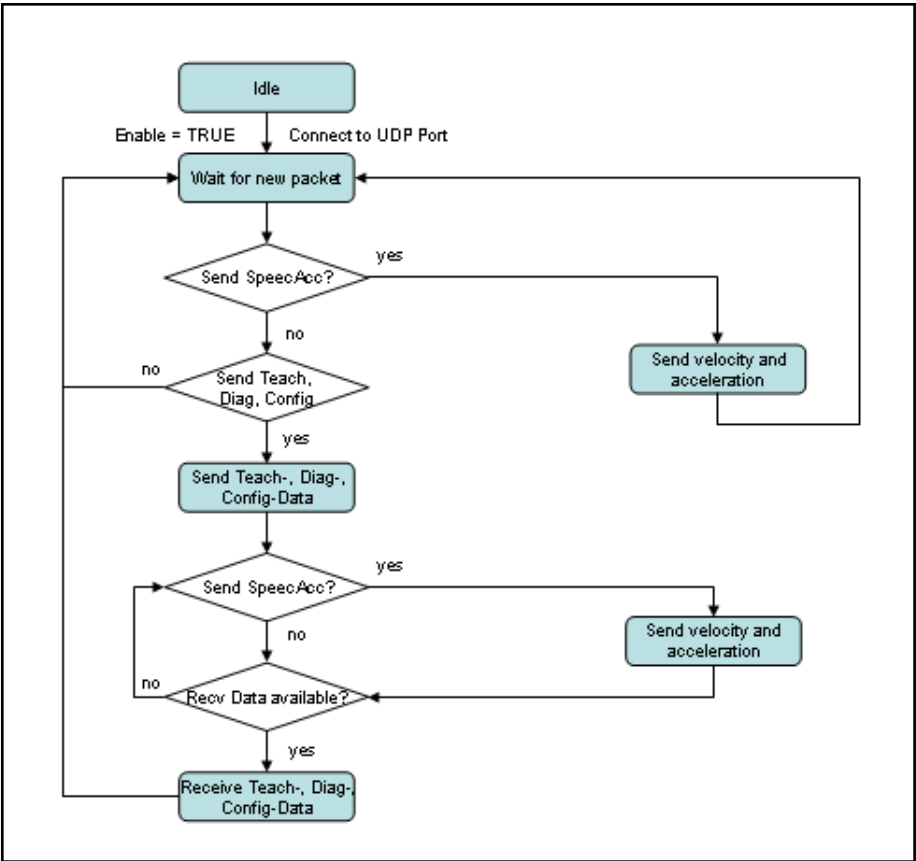


Abb. 3-50: Schema des internen Ablaufs des Funktionsbausteins MB_RegiSensor01SendRecvDataType02

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000003	Timeout beim Verbinden auf den UDP-Port
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001B	Ungültiger UDP-Port PortConfiguration
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000004	Timeout beim Senden
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000005	Timeout beim Empfangen
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000001C	Ungültige Sensor IP

Tab. 3-77: Fehlercodes des Funktionsbausteins MB_RegiSensor01SendRecvDataType02

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors und der Bibliothek RIL_SocketComm.lib ausgeben.

3.3.10 MB_RegiSensor01CycComType04

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01CycComType04 wird für die Kommunikation mit dem Markenstromsensor benötigt, falls die Sensorfirmware

RMB_TechRegiExtended

≥ 12 verwendet wird. Er führt jedoch nur die zyklische Kommunikation durch. Die Konfiguration des Sensors wird von dem Baustein MB_RegiSensor01ConfigType02 übernommen. Erst wenn der Sensor korrekt konfiguriert wurde, wird mit der Messung begonnen und der Baustein kann mit der Messauswertung beginnen. Es werden alle gemessenen Werte für das Längs- und Seitenregister am Ausgang angegeben. Diese Daten können vom Benutzer entsprechend der Konfiguration ausgewertet werden. Dieser Baustein MB_RegiSensor01CycComType04 eignet sich besonders für Anwendungen, bei denen mehrere gedruckte Marken zueinander vermessen werden. Beispielsweise ist dies der Fall, wenn sich nur ein Markenstromsensor am Ende der Maschine befindet.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V14	RMB_TechRegiExtended

Tab. 3-78: Referenztabelle Funktionsbaustein MB_RegiSensor01CycComType04

Schnittstellenbeschreibung

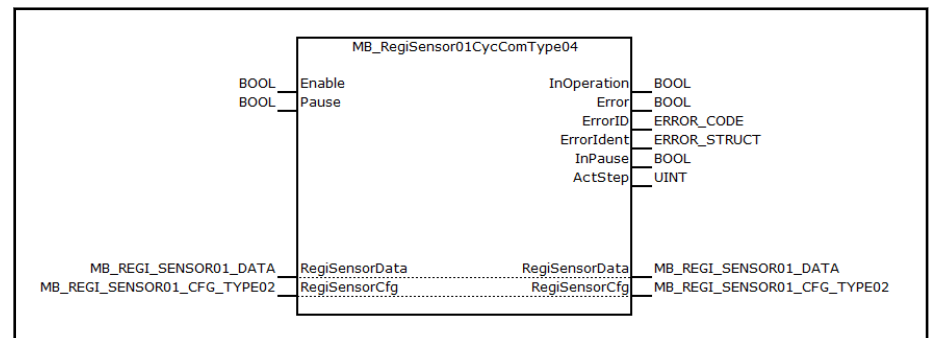


Abb. 3-51: Funktionsbaustein MB_RegiSensor01CycComType04

I/O-TYP	Name	Datentyp	Beschreibung
VAR_IN_OUT	RegiSensorCfg	MB_REGI_SENSOR01_CFG_TYPE02	Enthält die aktuellen Konfigurationsdaten des Sensors.
	RegiSensorData	MB_REGI_SENSOR01_DATA	Enthält die entsprechenden Messwerte des Sensors.
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert).
	Pause	BOOL	Unterbricht die Abarbeitung des FBs. Es werden keine neuen Messwerte ausgewertet. Die letzten Messdaten werden eingefroren.
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detailldiagnose
	InPause	BOOL	Zeigt an, dass der Eingang "Pause" aktiv ist und sich der Wert in RegiSensorData nicht ändert.
	ActStep	UINT	Zeigt an, dass ein neues Datenpaket mit Messwerten erhalten und ausgewertet wurde.

Tab. 3-79: E/A-Schnittstelle des Funktionsbausteins MB_RegiSensor01CycComType04

RMB_TechRegiExtended

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich

Tab. 3-80: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

MB_REGI_SENSOR01_DATA		
ActStep	UINT	Aktueller Schritt
LengthFeedback	ARRAY [0..16] OF REAL	Enthält den Markenabstand vom Nullimpuls zu den Marken 1..16. In [0] steht der Markenabstand für Bahn/Bahn2, falls eine Marke auf der Oberseite mit einer auf der Unterseite vermessen wurde.
SideFeedback	ARRAY [1..16] OF REAL	Enthält die entsprechenden Markenbreiten.
MarkState	ARRAY [0..16] OF MB_REGI_MARK_STATE	Status der Registermarken.
SensorState	MB_REGI_SENSOR_STATE	Gibt Warnmeldungen und Diagnosen des Sensors während der Messung an.

Tab. 3-81: Datenstruktur MB_REGI_SENSOR01_DATA

Die verwendete Datenstruktur MB_REGI_SENSOR01_CFG_TYPE02 [Tab. 3-58 "Datenstruktur MB_REGI_SENSOR01_CFG_TYPE02" auf Seite 89](#) wurde beim Funktionsbaustein [Kap. "Schnittstellenbeschreibung" auf Seite 88](#) beschrieben.

Aufzählungstyp

Der Aufzählungstyp MB_REGI_MARK_STATE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
FOUND	0	Marke gefunden
ROTATED	1	Marke verdreht
LARGE	2	Marke zu groß
SMALL	3	Marke zu klein
MISSING	4	Marke nicht gefunden
NOT_CONFIGURED	5	Marke nicht konfiguriert

Tab. 3-82: Aufzählungstyp MB_REGI_MARK_STATE

Der Aufzählungstyp MB_REGI_SENSOR_STATE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
START	0	Startzustand
SEARCHING	1	Suche Marken
INOP	2	Marken gefunden (Betriebsmodus)
STOP	3	Bahngeschwindigkeit unter 0,4m/s – keine Auswertung

Element	Wert	Beschreibung
LAMP	4	Lampe defekt
SENSOR	5	Sensor verschmutzt
TEACHING	6	Sensor in Teach-Phase

Tab. 3-83: Aufzählungstyp MB_REGI_SENSOR_STATE

Funktionsbeschreibung

Der Funktionsbaustein darf erst nach einem Aufruf des Funktionsbausteins MB_RegiSensor01ConfigType02 zum Durchführen einer Konfiguration mit einer steigenden Flanke an Enable aktiviert werden, da der Funktionsbaustein MB_RegiSensor01CycComType04 auf den UDP-Port der Struktur RegiSensorCfg zugreift und darauf verbindet. Wurde noch keine Konfiguration durchgeführt, so liegt ein fehlerhafter UDP-Port vor, da dieser noch nicht in die Struktur geschrieben wurde. In diesem Fall wird der entsprechende Error Ausgang des Funktionsbausteins gesetzt.

Bei der zyklischen Kommunikation arbeitet die SPS als UDP-Server. Das bedeutet, die Steuerung empfängt an dem über die azyklische Kommunikation definierten UDP-Port die Pakete des Sensors mit seinen Messwerten. Der Sensor sendet zyklisch nach jeder Zylinderumdrehung (d. h. nach jeder erfolgten Messauswertung) die Messdaten. Diese Messdaten werden von dem Funktionsblock ausgewertet. Beim Setzen von Enable wird ein bind() auf dem in RegiSensorCfg definierten UDP-Port gemacht. Somit wird auf dem UDP-Port mitgehört und die Messwerte empfangen und ausgewertet.

Jeder Sensor benötigt somit 2 UDP-Ports auf der Steuerung: Einen für die Konfigurations- und Diagnosedaten und einer für die Messwerterfassung. Die Kommunikation mit dem Sensor für das Empfangen der Messwerte erfolgt intern im Funktionsbaustein.

Die Struktur **RegiSensorData** enthält alle gemessenen Werte.

LengthFeedback[i]: Das Array LengthFeedback enthält alle gemessenen Markenabstände in Bezug auf den Nullimpuls sowie einen Messwert des Bahn/Bahn2 Verfahrens, die wie folgt in das Array eingetragen werden:

LengthFeedback[0]: Markenabstand definierte Marke unten zu definierter Marke oben.

LengthFeedback[1]: Markenabstand zwischen Nullimpuls und Marke 1.

LengthFeedback[2]: Markenabstand zwischen Nullimpuls und Marke 2.

...

LengthFeedback[15]: Markenabstand zwischen Nullimpuls und Marke 15.

LengthFeedback[16]: Markenabstand zwischen Nullimpuls und Marke 16.

Ist ein Messwert nicht vorhanden, da er nicht konfiguriert wurde oder weniger Marken vorhanden sind, so ist der Rückgabewert 0.0 und entsprechend im MarkState als Missing oder Not_Configured definiert.

SideFeedback[i]: Das Array SideFeedback enthält alle gemessenen Markenbreiten, die wie folgt im Array eingetragen sind:

SideFeedback[1]: Markenbreite Marke 1

SideFeedback[2]: Markenbreite Marke 2

...

SideFeedback[16]: Markenbreite Marke 16

RMB_TechRegiExtended

Ist ein Messwert nicht vorhanden, da die Marke nicht konfiguriert wurde, so ist der Rückgabewert 0.0 und entsprechend im MarkState als Not_Configured definiert. Wird eine Marke nicht gefunden und im MarkState mit "Missing" ausgegeben, so befindet sich im SideFeedback die Information über die Ursache des Markenverlusts.

Die Diagnosenummern sind in nachfolgender Tabelle aufgeführt:

Diagnosenummer	Bedeutung
11	Positionstoleranz überschritten
24	Höhtoleranz nicht erfüllt
26	Breitentoleranz verletzt
30	ShiftSyncPulse wurde wegen Überlauf nach links verschoben
31	ShiftSyncPulse wurde wegen Überlauf nach rechts verschoben
39	Zwischenraum der Doppelmarke wurde nicht gefunden
alle anderen Diagnosenummern	Interne Fehlernummer

Tab. 3-84: Diagnosenummern bei Mark "Missing"

SensorState: Wird nicht gesetzt.

MarkState[i]: Gibt die Gültigkeit der Werte in den Arrays LengthFeedback und SideFeedback an. Ist der Messwert z. B. der Marke 1 gültig, so steht in MarkState[1] = FOUND. Somit können die Werte LengthFeedback[1] und SideFeedback[1] ausgewertet werden. Ansonsten steht im MarkState[1] = Missing oder Not_Configured, je nachdem, ob die Marke ausgefallen ist oder ob die Marke nicht konfiguriert wurde.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000003	Timeout beim Verbinden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000020	Keine gültige Konfiguration gefunden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000B	Ungültiger UDP Port

Tab. 3-85: Fehlercodes des Funktionsbausteins MB_RegiSensor01CycCom-Type04

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors und der Bibliothek RIL_SocketComm.lib ausgeben.

3.3.11 MB_RegiSensor01ReadDiagType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01ReadDiagType01 liest Diagnose-daten, sowie das elektronische Typenschild des Sensors und legt sie in einer Datenstruktur ab. Dieser Funktionsbaustein benötigt zum Senden und Emp-

RMB_TechRegiExtended

fangen der Daten den Funktionsbaustein MB_RegiSensor01SendRecvData, der die UDP Kommunikation mit dem Sensor abwickelt.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-86: Referenztabelle Funktionsbaustein MB_RegiSensor01ReadDiagType01

Schnittstellenbeschreibung

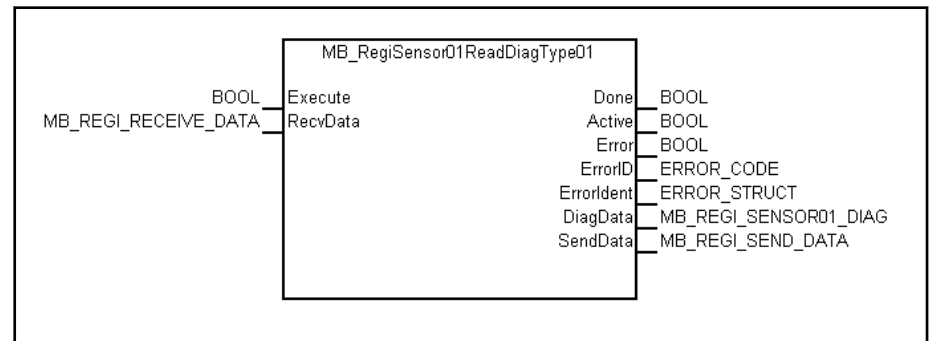


Abb. 3-52: Funktionsbaustein MB_RegiSensor01ReadDiagData Type01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Startet das Auslesen der Diagnosedaten.
	RecvData	MB_REGI_RECEIVE_DATA	Empfangene Daten vom Funktionsbaustein MB_RegiSensor01SendRecvData Type01.
VAR_OUTPUT	Done	BOOL	Das Auslesen der Diagnosedaten war erfolgreich.
	Active	BOOL	Das Auslesen der Diagnosedaten läuft gerade.
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerursache.
	DiagData	MB_REGI_SENSOR01_DIAG	Diagnosedaten des Sensors.
	SendData	MB_REGI_SEND_DATA	Daten die über den Funktionsbaustein MB_RegiSensor01SendRecvData gesendet werden sollen.

Tab. 3-87: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01ReadDiagData Type01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
MB_REGI_RECEIVE_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAdr	POINTER TO BYTE	n. def.	n. def.	0	Änderung an NewData
NewData	UINT	0	65535	0	Kontinuierlich

Tab. 3-88: Übersicht Min/Max-Werte und Initialwerte der Eingänge

RMB_TechRegiExtended

Datenstrukturen

MB_REGI_SENSOR01_DIAG		
ComponentName	STRING(52)	Zeigt den Namen der Komponente an
VendorName	STRING(28)	Lieferantenname
VendorCode	DWORD	Lieferantencode
DeviceName	STRING(40)	Typenschlüssel
VendorDeviceID	STRING(24)	BR-Materialnummer
FunctionRevision	WORD	Funktionsversion
HardwareRevision	STRING(16)	Hardwareversion
SoftwareRevision	STRING(16)	Softwareversion
SerialNumber	STRING(24)	Seriennummer
OperationalHours	DWORD	Betriebsstundenzähler in Std
Mac-Address	STRING(17)	Mac-Adresse
ChromaticityX	REAL	Aktuelle Farbintensität x (Auflösung 1/32768)
ChromaticityY	REAL	Aktuelle Farbintensität y (Auflösung 1/32768)
Intensity	UDINT	Aktuelle Intensität
Gain	UINT	Aktuelle Verstärkung (0..7)
Velocity	UDINT	Aktuelle Geschwindigkeit (Inc/s)
Status	WORD	Status (bitcodiert)
MeasurementState	WORD	Status Messung (Start/Stop)
SmoothFactorVel	UINT	Glättungsfaktor Mittelwertbildung Geschwindigkeit (0..8)
Acceleration	DINT	Aktuelle Beschleunigung (Inc/s ²)
SmoothFactorSensor	UINT	Smooth Faktor Sensor (0..4)
SensorSignals	UINT	Sensorsignale

Tab. 3-89: Datenstruktur MB_REGI_SENSOR01_DIAG

Die verwendete Datenstruktur [Tab. 3-24 "Datenstruktur MB_REGI_SEND_DATA"](#) auf Seite 43 wurde beim Funktionsbaustein [Kap. 3.3.3 "MB_RegiSensor01ConfigType01"](#) auf Seite 40 beschrieben.

Funktionsbeschreibung

Der Funktionsbaustein liest die Diagnosedaten des Sensors und legt sie in einer Datenstruktur ab. Die ersten 11 Datensätze stellen das elektronische Typenschild dar. Die nachfolgenden Parameter sind Einstellungen und aktuelle Zustände des Sensors die hierüber entsprechend ausgelesen und dargestellt werden können.

Im Folgenden wird die Auswertung der nicht selbsterklärenden Fehler beschrieben:

Status

Hier findet eine Unterscheidung in Low und HighByte statt. Low und HighByte können unabhängig voneinander geändert werden:

0x0800: Kanal Blau Sensor ist übersteuert

0x1000: Kanal Grün Sensor ist übersteuert

0x2000: Kanal Rot Sensor ist übersteuert

0x0001: Idle-Mode

0x0002: Configuration-Mode

0x0003: Measurement-Mode

0x0004: Teach-Mode

MeasurementState

0x0000: Keine Messung gestartet

0x0001: Messung gestartet

SensorSignals

Reserviert für weitere Diagnosen

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000000	Timeout beim Empfangen eines Pakets.

Tab. 3-90: Fehlercodes des Funktionsbausteins MB_RegiSensor01ReadDiagData Type01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors ausgeben.

3.3.12 MB_RegiSensor01SendSpeedAccType01

Kurzbeschreibung

Dieser Funktionsbaustein sendet in einem definierten Intervall dem Markenstromsensor entsprechende Geschwindigkeits- und Beschleunigungsinformationen. In Abhängigkeit eines Zeitintervalls werden die Daten in den entsprechenden Abständen berechnet und versendet. Desto geringer das Zeitintervall, desto mehr Werte werden versendet, jedoch wird auch deutlich mehr Datenverkehr erzeugt. Deshalb sollte das Zeitintervall mindestens doppelt so groß als die aktuelle Zykluszeit der Task in der der Funktionsbaustein aufgerufen wird sein. Der Sensor kann somit eine hochgenaue Messung auch in Beschleunigungsphasen sicherstellen. Der Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 berechnet die aktuellen Geschwindigkeits- und Beschleunigungswerte und erstellt das entsprechende Datenpaket. Dieses Datenpaket wird am Ausgang angelegt und über den Funktionsbaustein MB_RegiSensor01SendRecvDataType01 versendet. D. h. dieser Funktionsbaustein muss zeitgleich oder vorher gestartet werden. Da es sich um zeitkritische Daten handelt, werden die Datenpakete mit höchster Priorität berücksichtigt und sofort verschickt, sobald der "Senden"-Funktionsbaustein frei ist.

Es wird empfohlen, den Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 aufgrund des Datenverkehrs nur während Beschleunigungsphasen zu nutzen und ihn im Konstantlauf zu deaktivieren. Soll der Funktionsbaustein auch im Konstantlauf arbeiten, so wird ein Zeitintervall von > 250 ms empfohlen. Sendet der Funktionsbaustein Daten, so ermittelt der Sensor keine Geschwindigkeitsdaten mehr aus den Inkrementalgeberinformationen. Erst nach einer fallenden Flanke an Enable ermittelt der Sensor selbständig Geschwindigkeitsinformationen aus den Inkrementalgeberinformationen.



Der Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 muss in einer motion synchronen Task aufgerufen werden, damit die Beschleunigung korrekt berechnet wird. Ist dies aus Laufzeitgründen nicht möglich, so kann die Funktion mMotionCyclic() des Funktionsbausteins aus der motion synchronen Task aufgerufen werden und der Funktionsbaustein MB_RegiSensor01SendSpeedAccType01 kann in einer zyklischen Task aufgerufen werden. Der Aufruf in der MotionTask muss folgende Struktur besitzen: {CyclicTask}.fbInstanz}.mMotionCyclic(). Die Elemente in der geschweiften Klammer müssen durch den Namen des Programms in der zyklischen Task und durch den Instanznamen des Funktionsbausteins ersetzt werden. Somit könnte der Methodenaufwurf aus einem Programm das in der Motion Task läuft wie folgt aussehen: PLCProg.fbRegiSensor01SendSpeedAccType01.mMotionCyclic(). Wird dieser Aufruf nicht durchgeführt, kann eine korrekte Berechnung der Beschleunigung nicht gewährleistet werden.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-91: Referenztabelle Funktionsbaustein MB_RegiSensor01SendSpeedAccType01

Schnittstellenbeschreibung

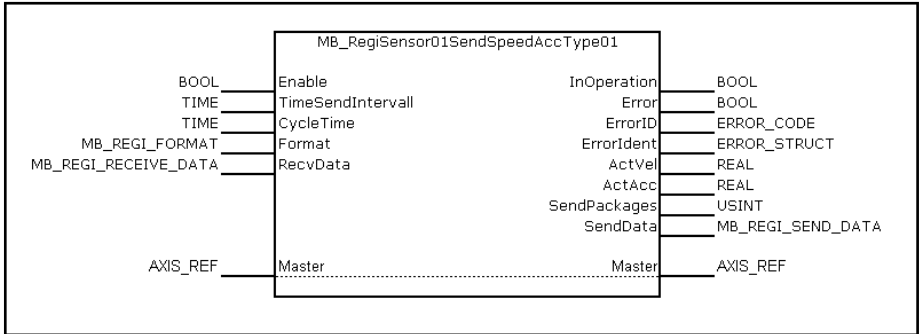


Abb. 3-53: Funktionsbaustein MB_RegiSensor01SendSpeedAccType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT CONST	Master	AXIS_REF	Achse, deren Geschwindigkeits- und Beschleunigungsinformation versendet werden soll. Aufgrund der Genauigkeit wird empfohlen eine virtuelle Leitachse anzulegen.
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert).
	TimeSendIntervall	TIME	In diesem Intervall werden die Daten berechnet und versendet.
	CycleTime	TIME	Zykluszeit der Task der SPS in der der Funktionsbaustein aufgerufen wird.
	Format	MB_REGI_FORMAT	Enthält den Druckzylinderumfang und die Anzahl der Inkremente pro Umdrehung.
	RecvData	MB_REGI_RECEIVE_DATA	Empfangene Daten vom Funktionsbaustein MB_RegiSensor01SendRecvDataType01.

I/O-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	ActVel	REAL	Aktuelle Geschwindigkeit in [Inc/s] die an den Sensor übertragen wird.
	ActAcc	REAL	Aktuelle Beschleunigung in [Inc/s^2] die an den Sensor übertragen wird.
	SendPackages	USINT	Paketzähler, wenn neue Daten berechnet werden und zum Senden übergeben werden.
	SendData	MB_REGI_SEND_DATA	Daten die über den Funktionsbaustein MB_RegiSensor01SendRecvData gesendet werden sollen.

Tab. 3-92: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01SendSpeedAccType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
TimeSendIntervall	TIME	$\geq 2 \cdot \text{CycleTime}$	n. def.	t#100ms	Steigende Flanke an Enable
CycleTime	TIME	> 0	n. def.	t#8ms	Steigende Flanke an Enable
MB_REGI_FORMAT					
FormatLength	REAL	>0	n. def.	0	Steigende Flanke an Enable
Increments	UDINT	>0	n. def.	0	Steigende Flanke an Enable
MB_REGI_RECEIVE_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAddr	POINTER TO MB_REGI_SENSOR01_ACK_BUFFER	n. def.	n. def.	0	Änderung an NewData
NewData	UINT	0	65535	0	Kontinuierlich

Tab. 3-93: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

Die verwendeten Datenstrukturen [Tab. 3-15 "Datenstruktur MB_REGI_FORMAT" auf Seite 42](#) und [Tab. 3-25 "Datenstruktur MB_REGI_RECEIVE_DATA" auf Seite 43](#) wurden beim Funktionsbaustein [Kap. 3.3.3 "MB_RegiSensor01ConfigType01" auf Seite 40](#) beschrieben.

Funktionsbeschreibung

Bei steigender Flanke an Enable werden die Eingänge überprüft und übernommen. Es wird zunächst ein Timeoutwert an den Sensor übertragen (vgl. Init packet). D. h. wird kein neuer Beschleunigungs- bzw. Geschwindigkeitswert mehr gesendet, so wird nach dieser Zeit die Geschwindigkeitsermittlung wieder im Sensor übernommen. Dieser Wert wird fest definiert mit 10 s vorgegeben. Der Sensor wartet auf das Antwortpaket, um anschließend eine

RMB_TechRegiExtended

erste aktuelle Beschleunigung zu ermitteln und an den Funktionsbaustein MB_RegiSensor01SendRecvDataType01 zu liefern, damit der Wert an den Sensor übertragen wird. Ab diesem Zeitpunkt wird nach Ablauf des Zeitintervalls eine neue Geschwindigkeit und Beschleunigung übertragen. Das Zeitintervall muss mindestens 2 mal größer sein als die Zykluszeit der Task, in der der Funktionsbaustein aufgerufen wird, damit der Datenverkehr beherrscht werden kann und auch weitere Funktionsbausteine entsprechende Daten senden und empfangen können. Bei einer fallenden Flanke an Enable wird ein Abschlusspaket (Send End packet) an den Sensor geschickt, durch das dem Sensor mitgeteilt wird, dass er seine Geschwindigkeit selbständig aus den Inkrementalgeber Signalen ermitteln soll. Es wird hierbei auf das Bestätigungspaket des Sensors gewartet. Somit kann es passieren, dass nach fallender Flanke von Enable, InOperation noch x Zykluszeiten anstehen bleibt.

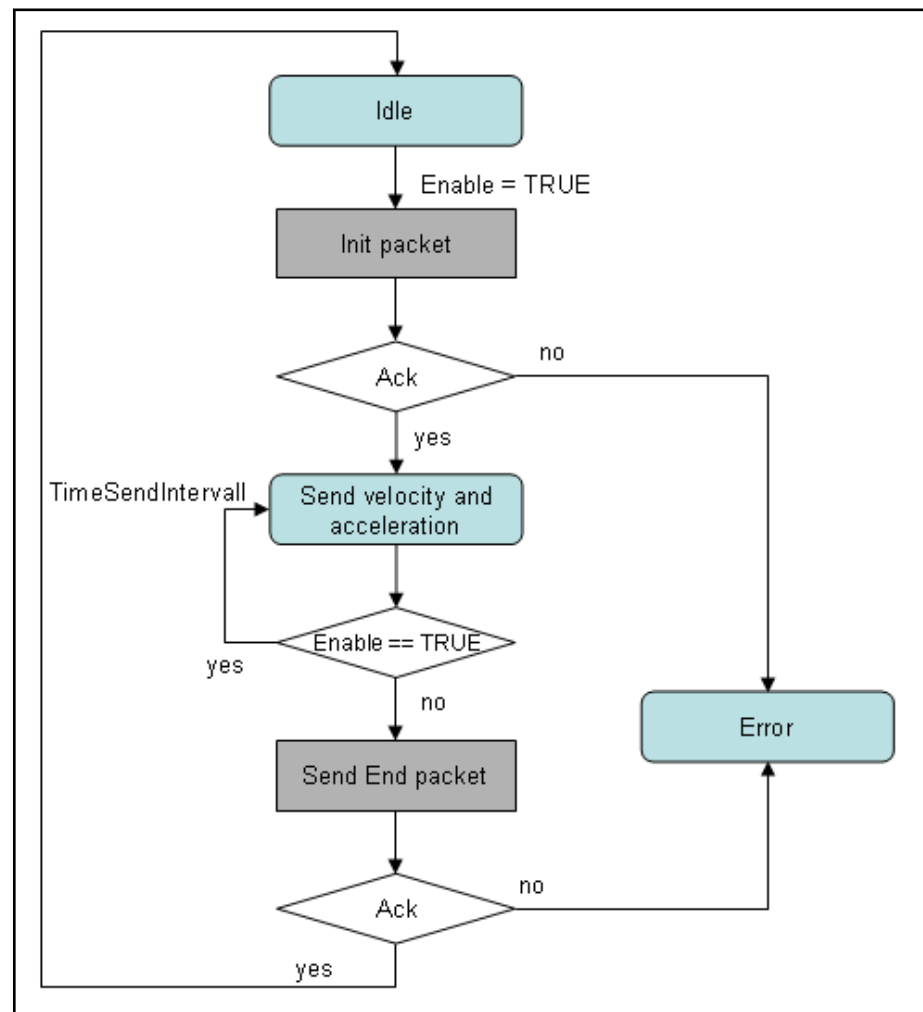


Abb. 3-54: Verhalten des MB_RegiSensor01SendSpeedAccType01

Handshake: Der Funktionsbaustein zeigt durch die Inkrementierung von NewData in SendData an, dass er neue Daten versenden will. Dies geschieht durch den Funktionsbaustein MB_RegiSensor01SendRecvDataType01. Empfängt dieser neue Daten, so setzt er den Wert von NewData in RecvData gleich. Somit weiß der Funktionsbaustein, dass es sich um das erwartete Antwortpaket handelt. Ein Handshake findet nur bei der Übertragung des Timeouts und des Abschlussdiagramms statt. Beim Senden der Beschleunigung und der Geschwindigkeit wird kein Antwortpaket des Sensors generiert, da es sich hierbei um zeitkritische Daten handelt und somit veraltete Daten in der Sendungswiederholung übertragen werden würden.

Timeout: Wird ein neues Datenpaket gesendet, so läuft ein interner Timer im Funktionsbaustein mit. Bekommt er innerhalb von 10 s keine Rückantwort, so wirft er einen Timeout und bricht mit einem Fehler ab. Ein möglicher Fehlerfall könnte sein, dass der Funktionsbaustein MB_RegiSensor01SendRecvData-Type01 nicht aktiviert bzw. fehlerhaft verbunden ist.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00001D11	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000022	Ungültiges Zeitintervall
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000023	Ungültige Zykluszeit
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Formzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000000D	Ungültige Anzahl der Inkremente
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000006	Timeout, keine Antwort des Funktionsbausteins MB_RegiSensor01SendRecvData-Type01 innerhalb von 10 Sekunden
INPUT_RANGE_ERROR (16#0006)	16#0000000D	16#00000000	AxisNo von "Master" ist außerhalb des gültigen Bereichs

Tab. 3-94: Fehlercodes des Funktionsbausteins MB_RegiSensor01SendSpeedAccType01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors ausgeben.

3.3.13 MB_RegiSensor01TeachData-Type01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01TeachData-Type01 liest und schreibt Messdaten. Aktuell gültige Messdaten können aus dem Markenstromsensor ausgelesen werden und beispielsweise nach Maschinenstopp oder Stromausfall direkt wieder in den Sensor geladen werden. Somit kann sofort mit der Messung begonnen werden ohne erneut eine Konfiguration zu übertragen bzw. ein Teachen durchzuführen.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-95: Referenztablette Funktionsbaustein MB_RegiSensor01TeachData-Type01

RMB_TechRegiExtended

Schnittstellenbeschreibung

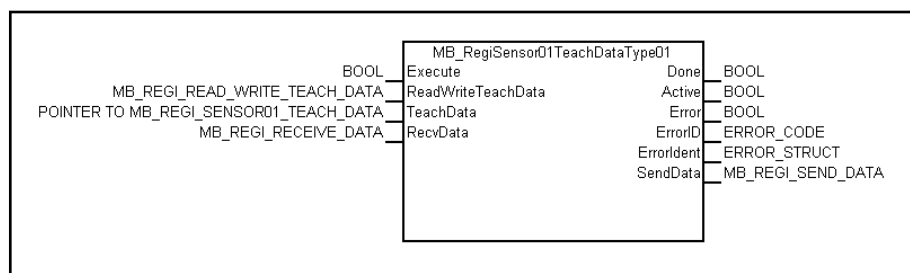


Abb. 3-55: Funktionsbaustein MB_RegiSensor01TeachDataType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Startet das Laden oder Speichern der Messdaten.
	ReadWriteTeachData	MB_REGI_READ_WRITE_TEACH_DATA	Auswahl, ob die Daten zum Sensor geladen oder vom Sensor gespeichert werden sollen.
	TeachData	POINTER TO MB_REGI_SENSOR01_TEACH_DATA	Adresse des Datentyps MB_REGI_SENSOR01_TEACH_DATA, in welchem die Daten abgelegt werden bzw. aus welchem die Daten geladen werden.
	RecvData	MB_REGI_RECEIVE_DATA	Empfangene Daten vom Funktionsbaustein MB_RegiSensor01SendRecvDataType01.
VAR_OUTPUT	Done	BOOL	Das Laden bzw. Speichern der Messdaten war erfolgreich.
	Active	BOOL	Das Laden bzw. Speichern der Messdaten läuft gerade.
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerursache.
	SendData	MB_REGI_SEND_DATA	Daten die über den Funktionsbaustein MB_RegiSensor01SendRecvData gesendet werden sollen.

Tab. 3-96: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01TeachDataType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
ReadWriteTeachData	MB_REGI_READ_WRITE_TEACH_DATA	0	1	0	Bei steigender Flanke an Execute
TeachData	POINTER TO MB_REGI_SENSOR01_TEACH_DATA	> 0	n. def.	0	Bei steigender Flanke an Execute
MB_REGI_RECEIVE_DATA					
NoOfBytes	UDINT	0	n. def.	0	Änderung an NewData
ValueAdr	POINTER TO BYTE	0	n. def.	0	Änderung an NewData
NewData	UINT	0	65535	0	Kontinuierlich

Tab. 3-97: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

MB_REGI_SENSOR01_TEACH_DATA		
TeachData	ARRAY [0..1310] OF BYTE	Messdaten (Teach-Daten)

Tab. 3-98: Datenstruktur MB_REGI_SENSOR01_TEACH_DATA

Die verwendeten Datenstrukturen Tab. 3-24 "Datenstruktur MB_REGI_SEND_DATA" auf Seite 43 und Tab. 3-25 "Datenstruktur MB_REGI_RECEIVE_DATA" auf Seite 43 wurden beim Funktionsbaustein Kap. 3.3.3 "MB_RegiSensor01ConfigType01" auf Seite 40 beschrieben.

Aufzählungstyp

Der Aufzählungstyp MB_REGI_READ_WRITE_TEACH_DATA kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
READ_TEACH_DATA	0	Liest die Daten aus dem Sensor aus
WRITE_TEACH_DATA	1	Schreibt die Daten zu dem Sensor

Tab. 3-99: Aufzählungstyp MB_REGI_READ_WRITE_TEACH_DATA

Funktionsbeschreibung

Der Funktionsbaustein liest/schreibt bei einer steigenden Flanke die aktuellen Messdaten vom/zum Markenstromsensor.

MB_READ_WRITE_TEACH_DATA:

Ist READ_TEACH_DATA ausgewählt, so werden die zuletzt gültigen Messdaten vom Markenstromsensor ausgelesen. Es werden die Daten an der angegebenen Adresse TeachData abgelegt.

Ist WRITE_TEACH_DATA ausgewählt, so werden die Messdaten in den Markenstromsensor geschrieben. Hierzu werden die Daten, die an der angegebenen Adresse TeachData liegen an den Markenstromsensor gesendet. Nach dem Senden der Daten kann sofort mit der Messung begonnen werden, ohne die Konfigurationsdaten zu übertragen oder ein neues Teachen zu starten.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle F_RELATED_TABLE, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000029	Aufzählungstyp MB_REGI_READ_WRITE_TEACH_DATA ungültig
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000028	TeachData zeigt auf 0 (Nullpointer)
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000005	Timeout beim Empfangen eines Pakets.

Tab. 3-100: Fehlercodes des Funktionsbausteins MB_RegiSensor01TeachData-Type01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors ausgeben.

RMB_TechRegiExtended

3.3.14 MB_RegiSensor01SetIPAddressType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiSensor01SetIPAddressType01 legt die IP-Adresse des Sensors in den permanenten Speicher des Sensors ab. Die MAC-Adresse wird für die Auswahl des Sensors benötigt und ist auf dem Gehäuse der Auswerteelektronik aufgedruckt.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-101: Referenztabelle Funktionsbaustein MB_RegiSensor01SetIPAddressType01

Schnittstellenbeschreibung

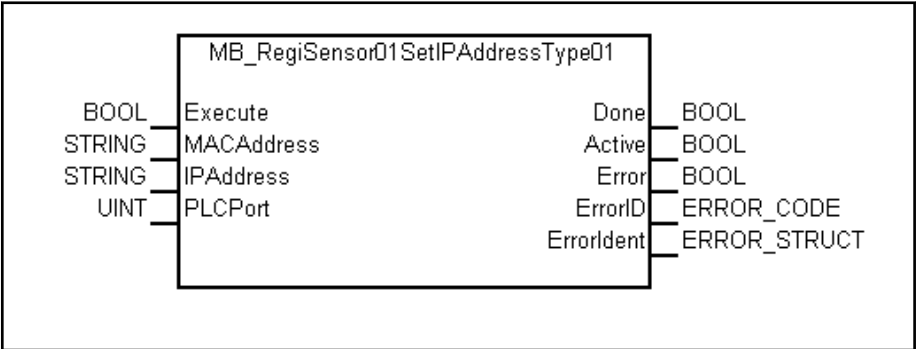


Abb. 3-56: Funktionsbaustein MB_RegiSensor01SetIPAddressType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Execute	BOOL	Startet das Setzen der IP-Adresse.
	MACAddress	STRING	MAC-Adresse als HEX-String. Doppelpunkte zur Trennung der 6 Hex-Werte sind erlaubt.
	IPAddress	STRING	Durch Punkte getrennte vierstellige IP-Adresse als String.
	PLCPort	UINT	Port von dem aus die IP-Adresse von der Steuerung an den Sensor gesendet wird.
VAR_OUTPUT	Done	BOOL	Das Setzen der IP-Adresse war erfolgreich.
	Active	BOOL	Das Setzen der IP-Adresse läuft gerade.
	Error	BOOL	Zeigt an, dass ein Fehler aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaillierte Fehlerursache.

Tab. 3-102: I/O-Schnittstelle des Funktionsbausteins MB_RegiSensor01SetIPAddressType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
MACAddress	STRING	LEN = 0	LEN = 17 (nur Länge 0, 12 und 17 gültig)	""	Steigende Flanke an Execute

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
IPAddress	STRING	LEN > 0	n. def.	""	Steigende Flanke an Execute
PLCPort	UINT	> 0	n. def.	6000	Steigende Flanke an Execute

Tab. 3-103: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Funktionsbeschreibung

Der Funktionsbaustein setzt die IP-Adresse des Markenstromsensors. Der Sensor wird über die MAC-Adresse ausgewählt. Die MAC-Adresse besteht aus 6 Hexadezimalwerten, die durch Doppelpunkte getrennt werden können. Führende Nullen dürfen nicht weggelassen werden. Wird keine MAC-Adresse angegeben, so darf nur ein Sensor im Ethernet vorhanden sein.

Die IP-Adresse besteht aus vier, durch Punkte getrennte, Dezimalwerte zwischen 0 und 255.

Die Funktion startet bei steigender Flanke an "Execute". Der Sensor sendet **keine** Rückantwort auf das Setzen der IP-Adresse. Nach erfolgreicher Ausführung ist "Done" auf TRUE. Die IP-Adresse ist dann dauerhaft im Sensor gespeichert. Es ist möglich, eine neue IP-Adresse durch erneutes Aufrufen des Funktionsbausteins zu setzen.

Tritt ein Fehler auf, so ist "Error" TRUE und "ErrorIdent" enthält genauere Informationen über den Fehler.



Der Funktionsbaustein sendet die neue IP-Adresse an den Sensor. Der Sensor sendet daraufhin keine Rückantwort. Der Sensor muss nach dem Übertragen der IP-Adresse zurückgesetzt werden. Der Ausgang "Done" sagt nur aus, dass das Paket erfolgreich gesendet wurde. Der Anwender muss nach dem Reset selbstständig feststellen, ob der Sensor die zugewiesene neue IP besitzt.

Die MAC-Adresse darf eine Länge von 12 oder 17 Zeichen haben. Alles andere wird nicht zugelassen. Das Trennzeichen kann beliebig gewählt werden. Es muss aber immer nach 2 Zeichen ein Trennzeichen kommen.

Beispiele für MAC-Adressen im Eingang "MACAddress":

- "00:60:35:08:35:9E"
- "00603508359E"

Es wird ausdrücklich darauf hingewiesen, dass immer die entsprechende MAC-Adresse mit angegeben werden soll, um sicherzustellen, dass nicht mehrere Sensoren dieselbe IP-Adresse zugewiesen bekommen. Die MAC-Adresse kann nur bei einer direkten Punkt-zu-Punkt Verbindung weggelassen werden.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000000	Timeout beim Lesen
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000009	Ungültige IP-Adresse angegeben

RMB_TechRegiExtended

ErrorID	Additional 1	Additional 2	Beschreibung
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000026	Ungültiger PLC-Port
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000003	Verbindungsfehler
COMMUNICATION_ERROR (16#0002)	16#00001D10	16#00000004	Timeout beim Senden
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000027	Format der MAC-Adresse fehlerhaft

Tab. 3-104: Fehlercodes des Funktionsbausteins MB_RegiSensor01SetIPAddressType01

Zusätzlich kann der Funktionsbaustein Fehler des Markenstromsensors und der Bibliothek RIL_SocketComm.lib ausgeben.

3.4 Funktionsbaustein zum Erzeugen bestimmter Geschwindigkeitsverläufe

3.4.1 Einführung und Übersicht

Zur Abbildung unterschiedlicher Geschwindigkeitsverläufe (Linear, S-Kurve) wird der folgende Funktionsbaustein in der Bibliothek RMB_TechRegiExtended zur Verfügung gestellt:

- MB_ReachVelocityType01

Hiermit werden in Abhängigkeit der Eingangsgrößen Werte für die Beschleunigung, Geschwindigkeit, sowie den Ruck berechnet.

3.4.2 MB_ReachVelocityType01

Kurzbeschreibung

Der Funktionsbaustein MB_ReachVelocityType01 berechnet den entsprechenden Ruck, die Beschleunigung und die Zielgeschwindigkeit zum Abfahren einer S-Kurve. Hierzu kann eine Ruckbegrenzung von 0-100 % ausgewählt werden. Ist der Eingabewert 0, so ist der Verlauf der Geschwindigkeit linear, bei 100 % S-kurvenförmig. Aus der Beschleunigungsdauer, der aktuellen Geschwindigkeit, der Zielgeschwindigkeit und dem JerkFactor werden die entsprechenden Sollwerte für Geschwindigkeit, Beschleunigung und Ruck berechnet. Die Beschleunigungsphase kann über StopAcc entsprechend gestoppt werden.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiExtended

Tab. 3-105: Referenztablette Funktionsbaustein MB_ReachVelocityType01

Schnittstellenbeschreibung

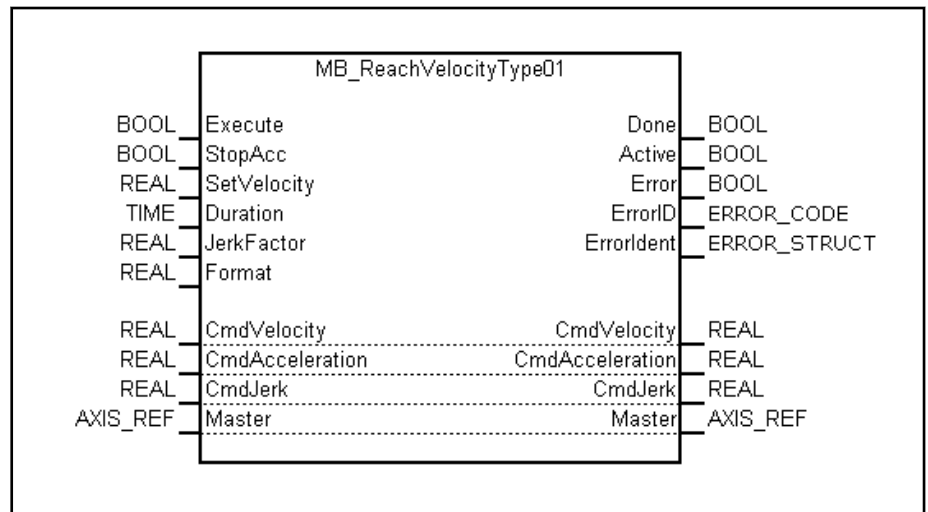


Abb. 3-57: Funktionsbaustein MB_ReachVelocityType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT CONST	Master	AXIS_REF	Enthält Informationen zur Achse, deren Geschwindigkeit entsprechend verändert werden soll. Es wird empfohlen eine virtuelle Leitachse anzulegen.
	CmdVelocity	REAL	Sollgeschwindigkeit [U/min]
	CmdAcceleration	REAL	Sollbeschleunigung [rad/s ²]
VAR_INPUT	CmdJerk	REAL	Sollruck [rad/s ³]
	Execute	BOOL	Bearbeitungsfreigabe des Funktionsbausteins.
	StopAcc	BOOL	Stoppt weitere Beschleunigung.
	SetVelocity	REAL	Zielgeschwindigkeit [m/min]
	Duration	TIME	Beschleunigungszeit
	JerkFactor	REAL	Ruckbegrenzung des Beschleunigungsprofils in %.
VAR_OUTPUT	Format	REAL	Format [mm]
	Done	BOOL	Beschleunigungsphase erfolgreich beendet.
	Active	BOOL	Funktionsbaustein arbeitet und befindet sich in einem Zwischenzustand.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Beschreibung der Diagnose im Fehlerfall.

Tab. 3-106: I/O-Schnittstelle des Funktionsbausteins MB_ReachVelocityType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Execute	BOOL			FALSE	Kontinuierlich
StopAcc	BOOL			FALSE	Kontinuierlich

RMB_TechRegiExtended

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
SetVelocity	REAL	n. def.	n. def.	0	Steigende Flanke an Execute
Duration	TIME	> t#0s	n. def.	t#30s	Steigende Flanke an Execute
JerkFactor	REAL	0	100	50	Steigende Flanke an Execute
Format	REAL	> 0	n. def.	1000	Steigende Flanke an Execute

Tab. 3-107: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Funktionsbeschreibung

Die Beschleunigungsphase startet mit einer steigenden Execute-Flanke. Beschleunigungsdauer (Duration), die anzufahrende Bahngeschwindigkeit (SetVelocity), sowie das Beschleunigungsprofil (JerkFactor) können vorher angegeben und übernommen werden. Eine Besonderheit ist, dass Active bei steigender Flanke an Execute gesetzt wird, die Beschleunigungsphase aber erst nach 500 ms beginnt. D. h. nachfolgende Funktionsbausteine (beispielsweise der Registerregelung) können Active verwenden, um sich den Beginn einer Beschleunigungsphase vorab anzeigen zu lassen. Nachdem die Zielgeschwindigkeit erreicht wurde, steht Active ebenfalls weitere 500ms an, bis schließlich die Bearbeitung mit Done abgeschlossen wird.

Die Auswirkung des JerkFactors wird in nachfolgender Abbildung veranschaulicht:

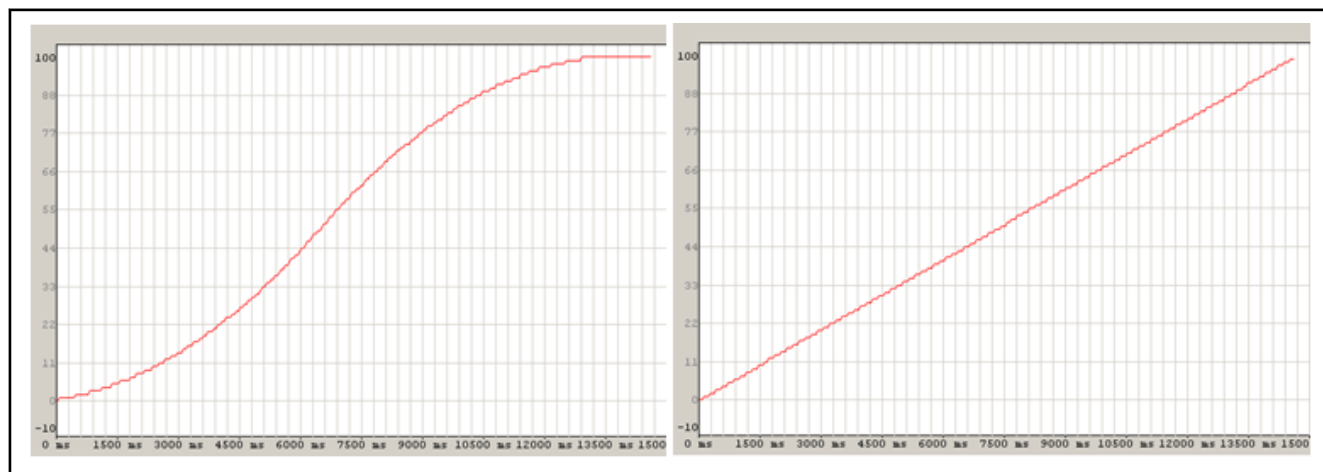


Abb. 3-58: Beispielhafte Beschleunigungsphasen

Abbildung links zeigt eine S-Kurve im Geschwindigkeitsverlauf bei einer Beschleunigungsphase von 0 auf 100 m/min (SetVelocity) in 15 Sekunden (Duration) mit JerkFactor = 100 % und Format von 1 m.

Abbildung rechts zeigt einen linearen Geschwindigkeitsverlauf bei einer Beschleunigungsphase von 0 auf 100 m/min (SetVelocity) in 15 Sekunden (Duration) mit JerkFactor = 0 % und Format von 1 m.

Die mit Execute angestoßene Beschleunigungsphase kann nur noch mit StopAcc angehalten werden. Der Funktionsblock beendet augenblicklich die weitere Beschleunigung. Done wird aktiv. Soll der gesetzte Geschwindigkeitswert SetVelocity dennoch angefahren werden, muss StopAcc deaktiviert und eine weitere steigende Flanke von Execute gesetzt werden.

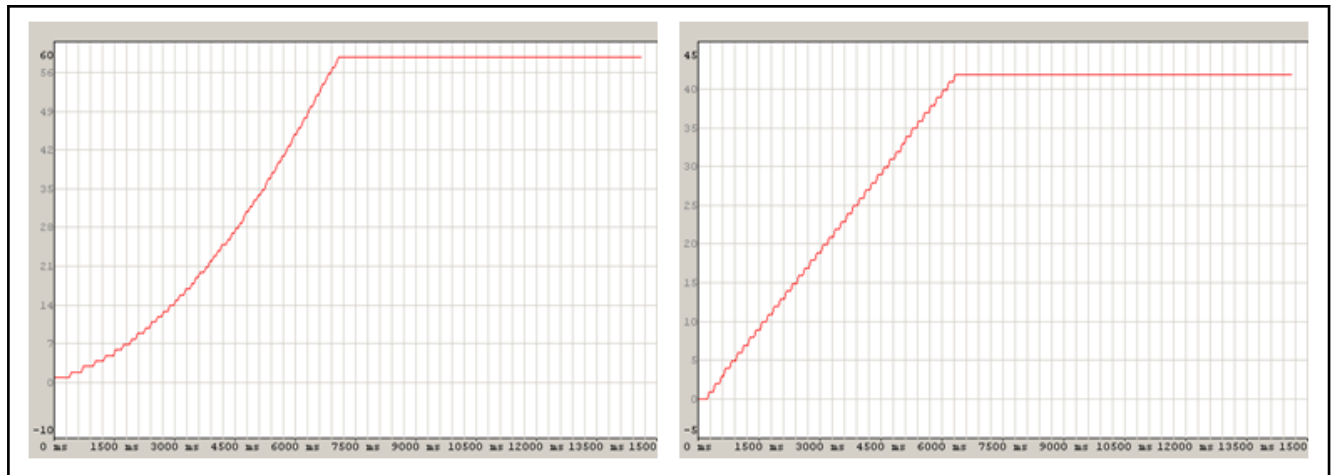


Abb. 3-59: Abbruch der Beschleunigungsphasen (links Abbruch bei 7000 ms, rechts Abbruch bei 6300 ms)

Die Abbildung zeigt den Abbruch der Beschleunigungsphase bei einem S-kurvenförmigen und linearen Geschwindigkeitsverlauf.

Die Bedeutung des Rucks wird im Folgenden erklärt: Der Ruck ist die zeitliche Ableitung der Beschleunigung. Der JerkFactor gibt den prozentualen zeitlichen Anteil der Beschleunigungsphase an, in der der Ruck ungleich Null ist. Nachfolgende Abbildung verdeutlicht die Bedeutung des Rucks. Findet keine Beschleunigungsänderung mehr statt, so ist der Ruck = 0.

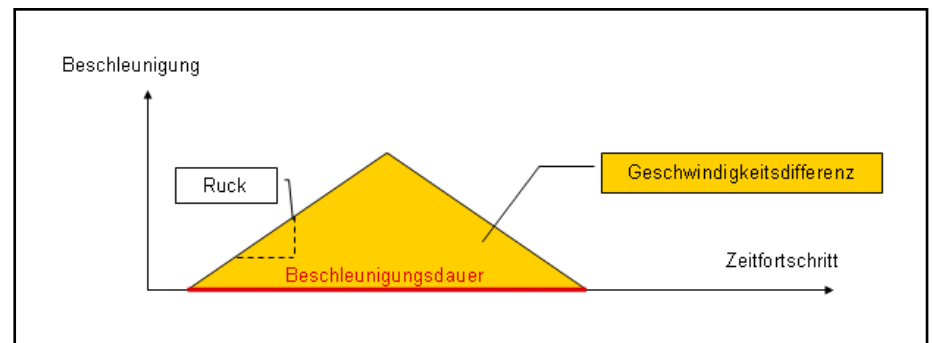


Abb. 3-60: Bedeutung des Rucks in der Beschleunigungsphase

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Druckzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000024	Ungültige Zeitdauer
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000025	Ungültiger JerkFactor
INPUT_RANGE_ERROR (16#0006)	16#0000000D	16#00000000	AxisNo von "Master" ist außerhalb des gültigen Bereichs

Tab. 3-108: Fehlercodes des Funktionsbausteins MB_ReachVelocityType01

4 RMB_TechRegiAdvanced

4.1 Lizenzpflichtige Bibliothek RMB_TechRegiAdvanced



Die Verwendung der Funktionsbausteine der Bibliothek RMB_TechRegiAdvanced unterliegt einer Lizenzpflicht. Je nach Anwendung muss eine der 3 Lizenzen FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT, FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1 oder FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2 zur Nutzung der Funktionsbausteine erworben werden. Pro Steuerung, auf der die Funktionsbausteine verwendet werden, ist jeweils eine Lizenz erforderlich (Runtime-Lizenz).

Einzellizenz Technologiebibliothek

Berechtigt zur Verwendung mit einer einzigen Steuerungsfirmware.

Single licence for technology library

Authorize the use with exactly one controller firmware.

TechLib RegisterControl Extended		
TechLib RegisterControl Extended		
Licence key (MNR): R911xxxxxx	Serial number: 1234567	Type code: FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT
Registration key WAHOF2-FBJYZF-8CSB0F-9DQO0D-9E8ZER		

Abb. 4-1: Lizenzblatt FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT

Einzellizenz Technologiebibliothek

Berechtigt zur Verwendung mit einer einzigen Steuerungsfirmware.

Single licence for technology library

Authorize the use with exactly one controller firmware.

TechLib RegisterControl Advanced 1		
TechLib RegisterControl Advanced 1		
Licence key (MNR): R911xxxxxx	Serial number: 1234567	Type code: FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1
Registration key WAHOF2-FBJYZF-8CSB0F-9DQO0D-9E8ZER		

Abb. 4-2: Lizenzblatt FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1

Einzellizenz Technologiebibliothek Berechtigt zur Verwendung mit einer einzigen Steuerungsfirmware.		
Single licence for technology library Authorize the use with exactly one controller firmware.		
TechLib RegisterControl Advanced 2 TechLib RegisterControl Advanced 2		
Licence key (MNR): R911xxxxxx	Serial number: 1234567	Type code: FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2
		
Registration key WAHOF2-FBJYZF-8CSB0F-9DQO0D-9E8ZER		

Abb. 4-3: Lizenzblatt FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2

Die Abbildungen zeigen jeweils die dazugehörigen Lizenzblätter. Pro Steuerung muss ein Lizenzblatt erworben werden, das zur Verwendung der Funktionsbausteine der Bibliothek RMB_TechRegiExtended oder RMB_TechRegiAdvanced auf diesem einen Steuerungssystem berechtigt.

Die Bibliotheken RMB_TechRegiExtended und RMB_TechRegiAdvanced werden bei den 3 Lizenzen FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT, FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1 oder FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2 mitgeliefert. Der Inhalt der jeweiligen Lizenzen ist nachfolgend aufgeführt:

FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT

- HMI für Schmalbahnanwendung (bis zu 12 Druckwerke, keine Entkopplung und Beschleunigungskompensation)
- Firmware Update Tool
- Dokumentationen: Anwenderbeschreibung und Bibliotheksbeschreibung
- Bibliotheken: RMB_TechRegiExtended und RMB_TechRegiAdvanced

FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1

- HMI für Anwendungen mit Klemmstellen (z. B. Tiefdruck) (bis zu 12 Druckwerke, mit Entkopplung und Beschleunigungskompensation)
- Firmware Update Tool
- Dokumentationen: Anwenderbeschreibung und Bibliotheksbeschreibung
- Bibliotheken: RMB_TechRegiExtended und RMB_TechRegiAdvanced

FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2

- HMI für Anwendungen mit Klemmstellen (z. B. Tiefdruck) (bis zu 24 Druckwerke, mit Entkopplung und Beschleunigungskompensation)
- Firmware Update Tool
- Dokumentationen: Anwenderbeschreibung und Bibliotheksbeschreibung
- Bibliotheken: RMB_TechRegiExtended und RMB_TechRegiAdvanced

Wird eine der Bibliotheken RMB_TechRegiExtended oder RMB_TechRegiAdvanced genutzt, so muss pro Steuerung, auf der einer der Funktionsbausteine genutzt wird, eine der Lizenzen FWS-IM*MLC-TEC-NNVRS-NN-REGI*EXT, FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV1 oder FWS-IM*MLC-TEC-NNVRS-NN-REGI*ADV2 vorliegen.

4.2 Allgemeines

Die Bibliothek RMB_TechRegiAdvanced stellt Funktionsbausteine zur Verfügung, die zur Regelung und zur Optimierung des Druckprozesses eingesetzt werden können.

Es handelt sich um eine kostenpflichtige Bibliothek, die die Bibliothek RMB_TechRegiExtended erfordert und beinhaltet. Sie benötigen pro verwendetes Steuerungssystem eine Lizenz.

Diese Dokumentation beschreibt die Ein- und Ausgänge der einzelnen Funktionsbausteine und gibt Hinweise für deren Verwendung.

Nachfolgende Übersicht zeigt alle Funktionsbausteine der lizenzpflichtigen Bibliothek RMB_TechRegiAdvanced:

Funktionsbaustein	Beschreibung
MB_RegisterControllerAdvType01	Registerregler für Anwendungen mit und ohne Klemmstellen
MB_RegiAccelerationCompensationType01	Modellbasierte und selbsteinstellende Beschleunigungskompensation
MB_RegiGravureRegulateType01	Dynamisches Entkopplungsnetzwerk
MB_RegiGravureRegulateType02	Dynamisches Entkopplungsnetzwerk mit Preset eines einzelnen Druckwerks und abschaltbarer Entkopplungswerte davorliegender Druckwerke

Tab. 4-1: Funktionsbausteine der lizenzpflichtigen Bibliothek RMB_TechRegiAdvanced



Der Funktionsbaustein MB_RegiAccelerationCompensationType01 muss in einer motion synchronen Task aufgerufen werden, damit die Beschleunigung korrekt berechnet wird. Ist dies aus Laufzeitgründen nicht möglich, so muss die Funktion mMotionCyclic() des Funktionsbausteins aus der motion synchronen Task aufgerufen werden und der Funktionsbaustein MB_RegiAccelerationCompensationType01 kann in einer zyklischen Task aufgerufen werden.

4.3 Funktionsbausteine zur Regelung und Optimierung des Druckprozesses

4.3.1 Einführung und Übersicht

Der Funktionsbaustein MB_RegisterControllerAdvType01 übernimmt die Regelung des Druckprozesses. Der Anwender muss hierzu keine Reglerparametrierungen mehr vornehmen, da die Reglerparameter selbständig in Abhängigkeit von Prozessdaten (Materialbahnlänge, Materialbahngeschwindigkeit, Totzeiten,...) berechnet werden. Dies stellt sicher, dass zu jedem Zeitpunkt des Druckprozesses optimale Reglerparameter verwendet werden. Der Anwender hat mittels eines Dämpfungsfaktors die Möglichkeit die Reglerparameter zu gewichten und kann über unterschiedliche Reglermodi die Auswahl zwischen unterschiedlichen Regelungsalgorithmen vornehmen.

Der Funktionsbaustein MB_RegiAccelerationCompensationType01 führt eine modellbasierte und adaptiv lernende Kompensation während der Beschleunigungsphase durch. Somit kann der auftretende dynamische Registerfehler während einer Beschleunigungsphase kompensiert werden und die Makulatur während einer Beschleunigungsphase drastisch reduziert werden. Aufgrund der selbstlernenden Funktionalität passt sich die Kompensationsgröße nach jeder Beschleunigungsphase selbständig an die Produktionsbedingungen an.

RMB_TechRegiAdvanced

Die Funktionsbausteine MB_RegiGravureRegulateType01 und MB_RegiGravureRegulateType02 entkoppeln Stellbewegungen von Registerreglern in verkoppelten Teilsystemen. Die Funktionsbausteine unterstützen sowohl die Regelung auf die Referenzfarbe wie auch auf die Vorgängerfarbe. Verstellbewegungen von davorliegenden Registerreglern werden auf nachfolgende Druckwerke vorgesteuert, um diese Druckwerke von der Verstellung zu entkoppeln.

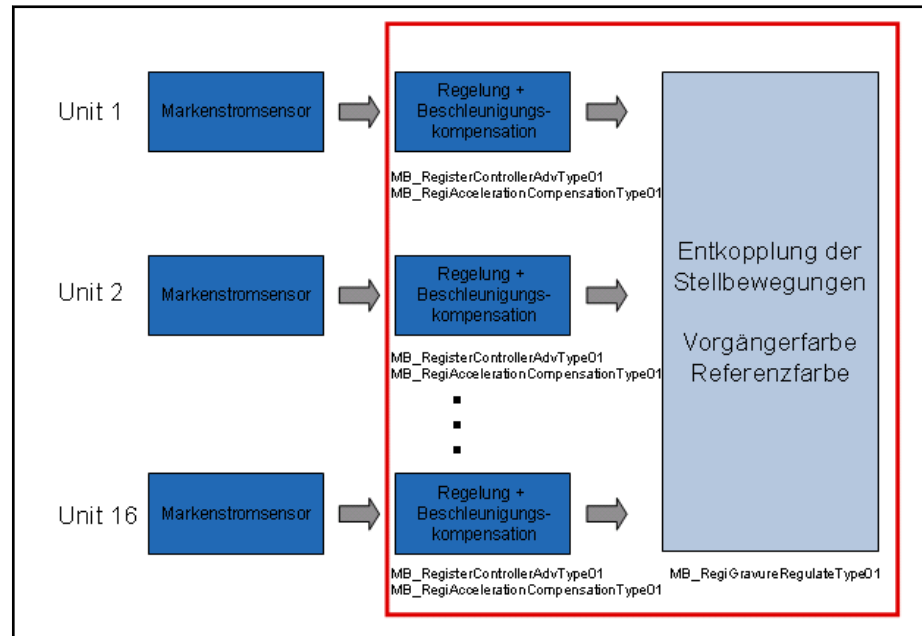


Abb. 4-4: Verwendung der Regelungsfunktionsbausteine

Die Abbildung zeigt die Verwendung der Regelungsfunktionsbausteine. Die Messwerte des Markenstromsensors, die durch die Funktionsbausteine der Bibliothek RMB_TechRegiExtended ermittelt werden, werden der Regelung zur Verfügung gestellt. Die berechneten Vorsteuerwerte der Beschleunigungskompensation und die Stellgröße der Regelung werden an das Entkopplungsnetzwerk weitergegeben, welches die Verstellungen für die Antriebssysteme berechnet und die Verstellungen für nachfolgende Druckwerke entkoppelt.

4.3.2 MB_RegisterControllerAdvType01

Kurzbeschreibung

Der Registerregler ist für Anwendungen mit Klemmstellen (z. B. beim Tiefdruck) und ohne Klemmstellen konzipiert. Es stehen unterschiedliche Reglerparametrierungen zur Auswahl. Diese werden geschwindigkeitsabhängig adaptiert. Somit liegen außen am Funktionsbaustein keine Reglerparameter an. Wird die Reglerparametrierung des symmetrischen Optimums verwendet, so wird ein Vorfilter für den Sollwert eingesetzt. Weiterhin erfolgt die Verstellung der Stellgröße des Registerreglers über eine Zylinderumdrehung rampenförmig. Dies bedeutet, dass eine Verstellung der berechneten Reglerstellgröße nicht ruckartig erfolgt, sondern über die Zeitdauer einer Zylinderumdrehung. Der Registerregler teilt sich in folgende Blöcke auf:

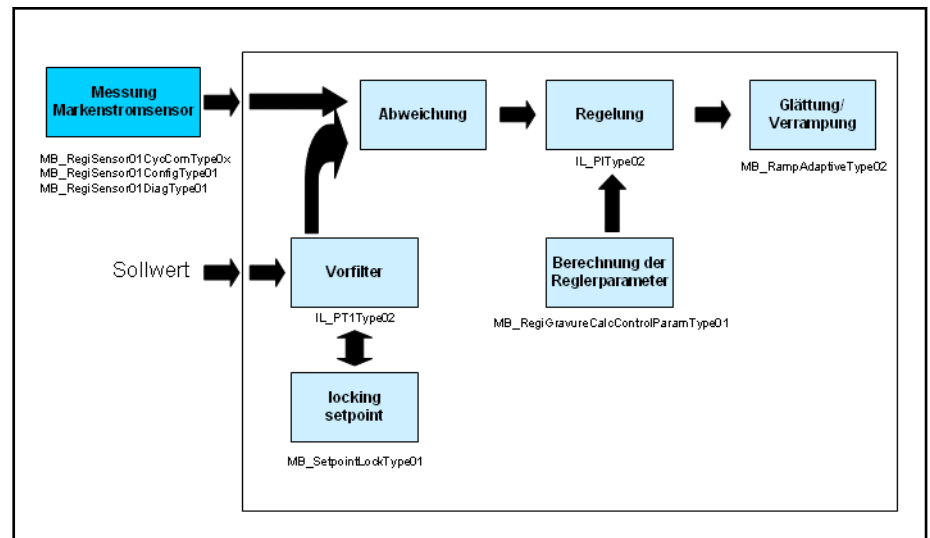


Abb. 4-5: Struktur des Funktionsbausteins *MB_RegisterControllerAdvType01*
MB_RegiGravureCalcControlParamType01

- Berechnung der Reglerparameter in Abhängigkeit der physikalischen Parameter.

IL_PT1Type02

- Berechnung des Sollwertfilters bei Reglerparametrierung nach dem symmetrischen Optimum

IL_PIType02

- PI Regler
- Tote Zone
- Begrenzung

MB_RampAdaptiveType02

- Rampenförmige Veränderung der Stellgröße des Registerreglers über eine Zylinderumdrehung.

MB_RegiSetpointLockType01

- Übernimmt den Istwert als neuen Sollwert

Der Registerregler FB "*MB_RegisterControllerAdvType01*" unterstützt die folgende Funktionalität:

- Berechnung einer Reglerstellgröße durch eine PI-Regelung
- Preset Feature
- Pause Feature
- Lock setpoint Feature
- Min/Max Begrenzung der Reglerstellgröße
- Lineare rampenförmige Veränderung der Reglerstellgröße
- Tote Zone
- Unterschiedliche Reglerparametrierungen
- Automatische Adaption der Reglerparameter an Prozessdaten
- Berechnung eines Vorfilters für eine Sollwertverstellung
- Dämpfung der optimalen Reglerparameter
- Umschaltung auf Führungsverhalten

RMB_TechRegiAdvanced

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiAdvanced

Tab. 4-2: Referenztabelle Funktionsbaustein MB_RegisterControllerAdvType01

Schnittstellenbeschreibung

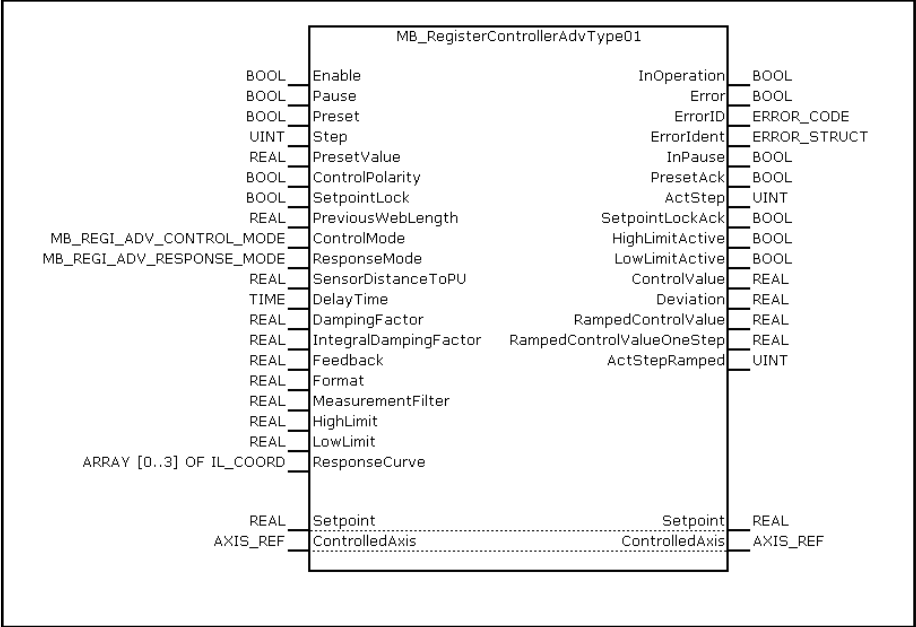


Abb. 4-6: Funktionsbaustein MB_RegisterControllerAdvType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_ OUT CONST	ControlledAxis	AXIS_REF	Referenz zur geregelten Achse
VAR_IN_ OUT	Setpoint	REAL	Sollwert Wird mit Feedback (Istwert) verglichen und kann auf den nächsten Wert von Feedback gesetzt werden (lock setpoint Feature).

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN-PUT	Enable	BOOL	Wenn TRUE, werden zunächst die Reglerparameter berechnet und "ControlValue" wird entweder auf den voreingestellten Wert von "PresetValue" (wenn an "Preset" eine steigende Flanke vorhanden ist) oder auf den berechneten Wert gesetzt. RampedControlValue wird auf den Wert von ControlValue gesetzt.
	Pause	BOOL	Der Eingang "Pause" wird nur ausgewertet, wenn der Eingang "Enable" aktiv ist. In diesem Fall werden keine neuen Reglerparameter mehr berechnet, der Wert "ControlValue" wird nicht verändert und auch die rampenförmige Veränderung dieses Wertes wird angehalten, d. h. RampedControlValue bleibt unverändert. ActStep wird verändert, ActStepRamped jedoch nicht.
	Preset	BOOL	Der Eingang "Preset" wird nur ausgewertet, wenn der Eingang "Enable" aktiv ist. Eine positive Flanke dieses Eingangs setzt den I-Anteil des Reglers auf "PresetValue", d. h. der Reglerausgang wird auf den in "PresetValue" parametrierbaren Wert gesetzt.
	Step	UINT	Taktschritt des Reglers. Wird erhöht, falls ein neuer Messwert des Sensors vorliegt. In diesem Fall werden die Reglerparameter neu berechnet und ein neuer "ControlValue" am Reglerausgang gesetzt, der über das neue Zeitintervall rampenförmig verändert wird.
	PresetValue	REAL	Der Ausgang "ControlValue" wird auf diesen Wert gesetzt bei steigender Flanke von "Preset".
	ControlPolarity	BOOL	Die Polarität von "ControlValue" wird invertiert, so lange dieser Wert "TRUE" ist.
	SetpointLock	BOOL	Ist der Eingang TRUE, so wird der nächste Istwert (Änderung von Step) als neuer Sollwert für den Registerregler verwendet.
	PreviousWebLength	REAL	Gibt die Materialbahnlänge zum davor liegenden Druckwerk in [mm] an.
	ControlMode	MB_REGI_ADV_CONTROL_MODE	Gibt den entsprechenden Algorithmus für die Berechnung der Regelungsparameter an.
	ResponseMode	MB_REGI_ADV_RESPONSE_MODE	Schaltet zwischen Führungs- und Störverhalten um
	SensorDistance-ToPU	REAL	Entfernung des Sensors zum Druckwerk [mm]
	DelayTime	REAL	Konstante im Regelkreis auftretende Totzeit
	DampingFactor	REAL	Gibt an, wie die optimal berechneten Reglerparameter gedämpft werden sollen
	IntegralDamping-Factor	REAL	Gibt an, wie die Nachstellzeit (T_N) gedämpft werden soll
	Feedback	REAL	Der gemessene Istwert
	Format	REAL	Druckzylinderformat in [mm]
	MeasurementFilter	REAL	Anzahl der Messwerte die gefiltert werden
	HighLimit	REAL	Die obere Grenze für "ControlValue". Wenn "ControlValue" diesen Wert überschreitet, wird er auf diesen Wert begrenzt.
	LowLimit	REAL	Die untere Grenze für "ControlValue". Wenn "ControlValue" diesen Wert unterschreitet, wird er auf diesen Wert begrenzt.
	ResponseCurve	ARRAY[0..3] OF IL_COORD	Sollwertkurve für die Proportionalverstärkung

RMB_TechRegiAdvanced

I/O-Typ	Name	Datentyp	Beschreibung
VAR_OUTPUT	InOperation	BOOL	Wenn TRUE, arbeitet der PI-Regler und die Reglerausgänge "ControlValue" und "RampedControlValue" sind gültig.
	Error	BOOL	Zeigt an, dass ein Fehler innerhalb des Funktionsbausteins aufgetreten ist.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Fehlerstruktur mit weiterer Unterteilung des Fehlers.
	InPause	BOOL	Zeigt an, dass der Eingang "Pause" aktiv ist und sich der Wert in "ControlValue" und "RampedControlValue" nicht ändert.
	PresetAck	BOOL	Zeigt an, dass "PresetValue" übernommen wurde.
	ActStep	UINT	Step Wert, wenn ein neuer "ControlValue" berechnet wurde, bezieht sich auf den Eingang von Step.
	SetpointLockAck	BOOL	Neuer Sollwert wurde übernommen
	HighLimitActive	BOOL	"ControlValue" hat obere Grenze erreicht
	LowLimitActive	BOOL	"ControlValue" hat untere Grenze erreicht
	ControlValue	REAL	Der berechnete Reglerausgangswert des PI-Reglers
	Deviation	REAL	Registerabweichung = Sollwert – Istwert
	RampedControlValue	REAL	Der aktuelle rampenförmig verstellte Wert des Registerreglers.
	RampedControlValueOneStep	REAL	Gibt nur die Änderung von RampedControlValue (differenziert) seit dem letzten SPS Zyklus wieder. Dies ist die Schnittstelle zum Entkopplungsnetzwerk.
	ActStepRamped	UINT	Step Wert, wenn ein neuer "RampedControlValueOneStep" berechnet wurde. Dieser ist unabhängig von Step, da sich der rampenförmige Wert in jedem Zyklus ändern kann.

Tab. 4-3: I/O-Schnittstelle des Funktionsbausteins MB_RegisterControllerAdv-Type01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Pause	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
Step	UINT	0	65335	0	Kontinuierlich
PresetValue	REAL	n. def.	n. def.	0.0	Steigende Flanke an "Preset"
ControlPolarity	BOOL			FALSE	Änderung von Step
SetpointLock	BOOL			FALSE	Änderung von Step
PreviousWebLength	REAL	> 0.0	n. def.	0.0	Steigender Flanke von "Enable"
ControlMode	MB_REGI_ADV_CONTROL_MODE	1	4	1	Steigender Flanke von "Enable"

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ResponseMode	MB_REGI_ADV_RESPONSE_MODE	1	2	1	Kontinuierlich
SensorDistance-ToPU	REAL	0	n. def.	400	Steigende Flanke von Enable
DelayTime:	TIME	0	n. def.	T#60ms	Steigende Flanke von Enable
DampingFactor	REAL	0	200	100	Änderung von Step
IntegralDampingFactor	REAL	0	1000	100	Änderung von Step
Feedback	REAL	n. def.	n. def.	0.0	Änderung von Step
Format	REAL	> 0		0.0	Steigende Flanke von Enable
MeasurementFilter	REAL	0	n. def.	0	Änderung von Step
HighLimit	REAL	LowLimit	n. def.	20000000	Änderung von Step
LowLimit	REAL	n. def.	HighLimit	-20000000	Änderung von Step
ResponseCurve	ARRAY[0..3] OF IL_COORD	n. def.	n. def.	0.0	Steigende Flanke von "Enable"
Setpoint	REAL	n. def.	n. def.	0.0	Änderung von Step

Tab. 4-4: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Aufzählungstyp

Der Aufzählungstyp MB_REGI_ADV_CONTROL_MODE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
CONTROL_MODE_CHR	1	CHR-Auslegungsverfahren
CONTROL_MODE_AMPLITUDE_OPTIMUM	2	Betragsoptimum
CONTROL_MODE_SYMMETRICAL_OPTIMUM	3	Symmetrisches Optimum
CONTROL_MODE_WOK_APERIODIC_FAIL	4	Wurzelortskurvenverfahren
CONTROL_MODE_LOOP_NO_LAG	5	Regelung ohne Klemmstelle
CONTROL_MODE_CHR_WITH_ADD_INTEGRAL_ADAPTION	6	CHR mit automatischer Anpassung des P- und I-Anteils in Abhängigkeit des DampingFactors
CONTROL_MODE_LOOP_NO_LAG_WITH_ADD_INTEGRAL_ADAPTION	7	Regelung ohne Klemmstelle mit automatischer Anpassung des P- und I-Anteils in Abhängigkeit des DampingFactors

Tab. 4-5: Aufzählungstyp MB_REGI_ADV_CONTROL_MODE

Der Aufzählungstyp MB_REGI_ADV_RESPONSE_MODE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
RESPONSE_MODE_COMMAND	1	Führungsverhalten beim Einrichten
RESPONSE_MODE_PRODUCTION	2	Störverhalten in der Produktion

Tab. 4-6: Aufzählungstyp MB_REGI_ADV_RESPONSE_MODE

RMB_TechRegiAdvanced

Funktionsbeschreibung

Regelalgorithmus

Der Regelalgorithmus kann mit folgendem Diagramm dargestellt werden:

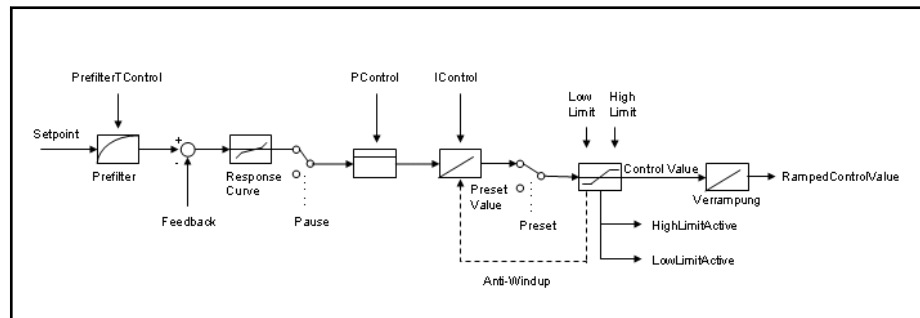


Abb. 4-7: Regelalgorithmus MB_RegisterControllerAdvType01

Die Eingangswerte des Vorfilters und die P- und I- Eingangsgrößen des Registerreglers werden im Funktionsbaustein MB_RegiGravureCalcControlParamType01 berechnet. Hierfür stehen unterschiedliche Parametriermodi zur Auswahl. Die Reglerparameter werden bei jeder Änderung des Step Eingangs in Abhängigkeit von physikalischen Parametern neu berechnet. Der Registerregler und der Vorfilter berechnen ebenfalls nur bei einer Änderung des Step Eingangs die Ausgangsgrößen neu. Zu beachten ist, dass der Funktionsbaustein MB_RampAdaptiveType02 in jedem Zyklus einen neuen "RampedControlValue" berechnet. Er verändert rampenförmig die Stellgröße des Registerreglers über eine Zylinderumdrehung. Somit ändert sich der Eingang Step und der Ausgang ActStepRamped in unterschiedlichen Zyklen. Der Vorfilter wird nur verwendet, wenn als Regelmodus das symmetrische Optimum gewählt wurde, um das Überspringen bei einer Sollwertänderung zu vermeiden bzw. zu minimieren.

Das Differenzsignal aus "Feedback"-Eingang und "Setpoint" wird über die in "ResponseCurve" angegebene Kennlinie gewichtet. Die Kennlinie ist punktsymmetrisch zum Ursprung. Die X-Werte der Kennlinie entsprechen dem Differenzsignal, die Y-Werte dem gewichteten Differenzsignal. Es können bis zu vier Punkte angegeben werden. Die Teilstücke zwischen den Punkten werden linear interpoliert. Wird "ResponseCurve" nicht angelegt, so entfällt die Wichtung.

Initialisierung

Während der Initialisierung müssen die Eingänge "Setpoint" und "Feedback" gültige Daten haben. Der ControlValue am Ausgang des PI-Reglers wird auf den RampedControlValue geschrieben. Der RampedControlValue wird erst nach der zweiten Änderung am Step Eingang rampenförmig verändert, da erst zu diesem Zeitpunkt die Dauer einer Zylinderumdrehung bekannt ist. Dies bedeutet wiederum, dass der erste ControlValue bei der ersten Änderung am Step-Eingang nicht rampenförmig verstellt wird.

PresetValue

Wenn der Anwender einen störungsfreien Start des Reglerausgangs "ControlValue" wünscht, dann muss er den "Preset"-Eingang setzen, wenn "Enable" aktiviert wird.

Während dieser Zeit müssen die Eingänge "Setpoint" und "Feedback" gültige Daten haben.

Zusätzlich ist die Überprüfung der Ober-/Untergrenzen noch aktiv, um eine Begrenzung des "ControlValue" zu gewährleisten, wenn der vorgegebene Wert ("PresetValue") außerhalb der Grenzen von "HighLimit" oder "LowLimit" ist.

PresetAck wird aktiv, wenn der Wert an "PresetValue" übernommen wurde. "Preset" hat Priorität über "Pause". Der Wert in "PresetValue" wird bei steigender Flanke von "Preset" übernommen, wenn "Enable" gesetzt ist. Eine erfolgte Übernahme wird am Ausgang "PresetAck" angezeigt.

Pause

Der Eingang "Pause" wird nur ausgewertet, wenn der "Enable"-Eingang aktiv ist. In diesem Fall wird intern die Regelabweichung "Setpoint" - "Feedback" auf 0 gesetzt. Das bewirkt, dass "ControlValue" fixiert ist. Dies kann dazu benutzt werden, um den Ausgang konstant in einem stabilen Zustand zu halten. Der Ausgang ActStep wird bei aktiviertem "Pause"-Eingang aktualisiert, der Ausgang ActStepRamped ändert sich nicht. Solange "Pause" aktiv ist, ist der Ausgang "InPause" gesetzt. Weiterhin werden keine neuen Reglerparameter berechnet und auch der Wert am Ausgang "RampedControlValue" wird eingefroren.

Lock Setpoint

Der aktuelle Feedback-Wert kann mit dem Feature SetpointLock als Sollwert übernommen werden. Wird der Eingang "TRUE", so wird der Wert von Setpoint mit dem Wert von Feedback überschrieben, der bei der nächsten Änderung von "Step" anliegt.

Dieses Feature bietet dem Benutzer z. B. die Möglichkeit, den Regler in den Pause-Mode zu schalten, das System manuell ins Register zu vertippen und den nächsten Feedback-Wert als neuen Setpoint zu übernehmen.

Tote Zone

Die Abweichung, die aus der Differenz des Setpoints mit dem Feedback berechnet wird, wird mittels der "ResponseCurve" gewichtet, d. h. der Registerregler wird bei einer geringen Registerabweichung "eingefroren" bzw. er wird die Regeldynamik zurücknehmen. Letzteres ist hilfreich, um im eingeschwungenen Zustand einen möglichst "ruhigen" Regelverlauf zu erhalten und größere Abweichungen relativ schnell auszuregeln.

Normalbetrieb

Der Regler berechnet nur einen neuen "ControlValue", wenn sich der Wert am "Step"-Eingang ändert. Dieser Wert wird über die Dauer der letzten Zylinderumdrehung rampenförmig verstellt, d. h. in jedem Zyklus ändert sich der Wert "RampedControlValue".

Wenn "ControlValue" den Wert von "HighLimit" oder "LowLimit" über-, bzw. unterschreitet, dann wird "ControlValue" auf einen dieser Werte begrenzt. Wenn der Regler einen Integrationsanteil hat, wird zusätzlich ein Anti-Windup aktiv. D. h. der integrale Anteil integriert nicht weiter auf. In diesem Fall werden die Ausgänge "HighLimitActive" und "LowLimitActive" gesetzt.

ResponseMode

Der Standard-Wert ist auf Führungsverhalten (RESPONSE_MODE_COMMAND) eingestellt. Dieser Modus sollte beim Zuschalten des Reglers und beim Vertippen des Sollwerts verwendet werden. Am Ende der Einrichtung sollte auf RESPONSE_MODE_PRODUCTION umgeschaltet werden, um eine bessere Regeldynamik zu erreichen. In RESPONSE_MODE_PRODUCTION wird der Modus für das Störverhalten ausgewählt der in ControlMode gesetzt wurde. Für RESPONSE_MODE_COMMAND wird bei der Berechnung der Reglerparameter nur zwischen den Modi CONTROL_MODE_CHR, CONTROL_MODE_LOOP_NO_LAG, CONTROL_MODE_CHR_WITH_ADD_INTEGRAL_ADAPTION und CONTROL_MODE_LOOP_NO_LAG_WITH_ADD_INTEGRAL_ADAPTION unterschieden. Wird ein anderer Modus gewählt so werden die Reglerparameter nach dem CONTROL_MODE_CHR berechnet.

Control Mode / Berechnung der Reglerparameter

Für die Reglerparameterberechnung werden der ausgewählte Mode, die Geschwindigkeit, die Materialbahnlänge, der Formzylinderumfang, der Abstand des Sensors zum Druckwerk, der Messwertfilter und eine konstante Totzeit aufgrund Übertragungs- und Rechenzeiten berücksichtigt. Der Standardwert für die konstanten Totzeiten von 60 ms ist ein Erfahrungswert und sollte zunächst nicht geändert werden. Der Abstand des Sensors zum Druckwerk bringt eine zusätzliche Verzögerung ein und sollte deshalb eingegeben werden, ebenso wie die Anzahl der Messwerte über die gefiltert wird.

Control Mode	Druckprozess	Beeinflussung der Reglerparameter
CONTROL_MODE_CHR	Mit Klemmstelle	DampingFactor, beeinflusst P-Anteil IntegralDampingFactor, beeinflusst I-Anteil
CONTROL_MODE_AMPLITUDE_OPTIMUM	Mit Klemmstelle	DampingFactor, beeinflusst P-Anteil IntegralDampingFactor, beeinflusst I-Anteil
CONTROL_MODE_SYMMETRICAL_OPTIMUM	Mit Klemmstelle	DampingFactor, beeinflusst P-Anteil IntegralDampingFactor, beeinflusst I-Anteil
CONTROL_MODE_WOK_APERIODIC_FAIL	Mit Klemmstelle	DampingFactor, beeinflusst P-Anteil IntegralDampingFactor, beeinflusst I-Anteil
CONTROL_MODE_LOOP_NO_LAG	Ohne Klemmstelle	DampingFactor, beeinflusst P-Anteil IntegralDampingFactor, beeinflusst I-Anteil
CONTROL_MODE_CHR_WITH_ADD_INTEGRAL_ADAPTION	Mit Klemmstelle	DampingFactor beeinflusst P- und I-Anteil IntegralDampingFactor hat keine Wirkung
CONTROL_MODE_LOOP_NO_LAG_WITH_ADD_INTEGRAL_ADAPTION	Ohne Klemmstelle	DampingFactor beeinflusst P- und I-Anteil IntegralDampingFactor hat keine Wirkung

Tab. 4-7: Verwendung der Control Modes

Bei den Regelungsmodi wird zunächst zwischen dem Druckprozess unterschieden. Besitzt, der Druckprozess eine Klemmstelle (z. B. Tiefdruck) wird eine Verstellung nicht sofort wirksam und es werden spezielle Verfahren für die Berechnung der Reglerparameter verwendet (Modi 1,2,3,4,6). Besitzt der Druckprozess keine Klemmstelle, so wird eine Verstellung nahezu sofort wirksam. Auch dies wird bei der Berechnung der Reglerparameter berücksichtigt (Modi 5,7).

Die Reglerparameter werden in Abhängigkeit des Control Modes automatisch berechnet. Der Anwender hat die Möglichkeit diese Parameter über den DampingFactor bzw. IntegralDampingFactor zu beeinflussen. Bei den ersten fünf Modi wirkt der DampingFactor auf den P-Anteil und der IntegralDampingFactor auf den I-Anteil. Beispiel: Beträgt der DampingFactor 50% so wird der automatisch berechnete P-Anteil halbiert.

Ausnahmen bilden "CONTROL_MODE_CHR_WITH_ADD_INTEGRAL_ADAPTION" und "CONTROL_MODE_LOOP_NO_LAG_WITH_ADD_INTEGRAL_ADAPTION". Hier hat der IntegralDampingFactor keine Wirkung. Wird der DampingFactor verändert, so wirkt dies automatisch auf den P- und I-Anteil. Beispiel: Ist der DampingFactor < 100 % so wird der P-Anteil ent-

sprechend reduziert und der I-Anteil entsprechend vergrößert, da der Anwender eine geringere Regelkreisdynamik wünscht.

Wird der DampingFactor bei den Modi 1-5 auf 0 gesetzt, so wird der P-Regler deaktiviert. Wird der IntegralDampingFactor bei den Modi 1-5 auf 0 gesetzt, so wird der I-Regler deaktiviert. Wird der DampingFactor bei den Modi 6 oder 7 auf 0 gesetzt, so wird der P-Regler sowie der I-Regler deaktiviert, der IntegralDampingFactor hat in diesen Modi keine Bedeutung.



Es wird empfohlen die automatisch berechneten Reglerparameter zu verwenden. D. h. der DampingFactor und der IntegralDampingFactor sollten auf 100 % gesetzt werden. Muss aufgrund des Druckprozesses eine manuelle Anpassung der Reglerparameter vorgenommen werden, so wird empfohlen einen der Modi "CONTROL_MODE_CHR_WITH_ADD_INTEGRAL_ADAPTION" oder "CONTROL_MODE_LOOP_NO_LAG_WITH_ADD_INTEGRAL_ADAPTION" zu verwenden, da hier automatisch eine Anpassung des P- und I-Anteils im optimierten Verhältnis erfolgt.

Verwendung der Ausgangsgrößen

Nachfolgende Abbildung zeigt die vom Regler erzeugten Ausgangsgrößen. Der ControlValue ist der Ausgangswert des Reglers, der nach jedem neuen Messwert berechnet wird. Es ist bereits der aufintegrierte Wert und muss im Folgenden nicht mehr aufaddiert werden. Bei jedem neuen Messwert wird der ActStep um eins hochgezählt. Die Stellgröße des ControlValue wird über die Dauer der Zylinderumdrehung rampenförmig verstellt. Dieser Wert wird mit dem RampedControlValue ausgegeben. Der RampedControlValue wird in jedem Rechenzyklus aktualisiert und der dazugehörige ActStepRamped hochgezählt. Weiterhin wird der RampedControlValueOneStep ausgegeben, der jeweils die Änderung des RampedControlValue angibt. Dieser Wert muss mit dem Entkopplungsnetzwerk verbunden werden, da dieser Wert im Entkopplungsnetzwerk aufintegriert wird. Er wird ebenfalls in jedem Rechenzyklus aktualisiert und bezieht sich auf den ActStepRamped.

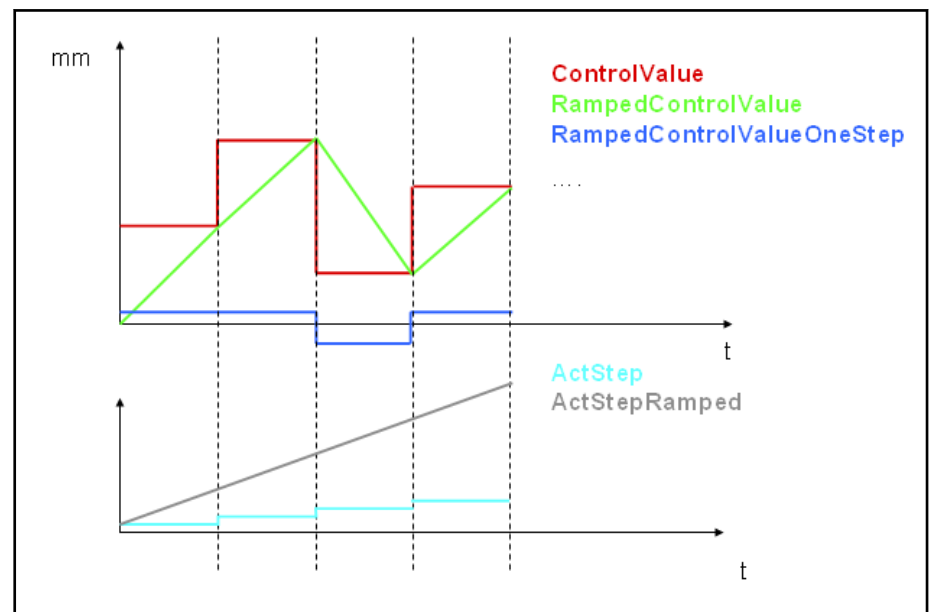


Abb. 4-8: Verwendung der Reglerausgangsgrößen

HighLimit und LowLimit

Dieser Wert begrenzt die Reglerstellgröße von einer Änderung des Step Eingangs zur nächsten Änderung. D. h. wird beispielsweise ein Wert von 1 mm

RMB_TechRegiAdvanced

für das HighLimit ausgewählt, so wird die Änderung der Reglerstellgröße auf maximal 1 mm pro Rechenschritt (ein Rechenschritt entspricht dem Erhalt eines neuen Messwerts) begrenzt.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in **Additional 1/Additional 2** folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002A	Ungültiger Dämpfungsfaktor
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002B	Ungültige Materialbahnlänge
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002C	Ungültiger Control Mode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002D	Ungültige Geschwindigkeit
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Formzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002E	Ungültiger Response Mode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002F	Ungültige Sensordistanz
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000030	Ungültige Verzögerungszeit
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000045	Ungültiger integraler Dämpfungsfaktor
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000046	Ungültiger Messwertfilter
INPUT_RANGE_ERROR (16#0006)	16#0000000D	16#00000000	AxisNo von "Master" ist außerhalb des gültigen Bereichs

Tab. 4-8: Fehlercodes des Funktionsbausteins MB_RegisterControllerAdv-Type01

4.3.3 MB_RegiAccelerationCompensationType01

Kurzbeschreibung

Der Funktionsbaustein MB_RegiAccelerationCompensationType01 berechnet eine Vorsteuerung, die den Registerfehler in der Beschleunigungsphase und Produktionsphase kompensiert. Dabei kann eine modellbasierte Kompensation durchgeführt werden, wenn die entsprechenden Maschinen- und Materialdaten vorhanden sind. Weiterhin ist eine adaptive und selbsteinzelnde Kompensation implementiert, die sich aus der Reglerstellgröße und der Geschwindigkeit die Kompensationswerte selbständig ermittelt und in jeder Beschleunigungsphase optimiert.

Die Vorsteuerung MB_RegiAccelerationCompensationType01 ist einem Regelkreis zur Minimierung des Registerfehlers vorgelagert. Die Stellsignale des Reglers und der Vorsteuerung werden addiert und auf die Strecke gegeben.

Die Verwendung einer Vorsteuerung ist aus zwei Gründen sinnvoll:

1. Während der Beschleunigungsphase tritt ein großer instationärer Registerfehler auf, der danach wieder restlos verschwindet. Der Regler ist für den stationären Bahnlauf ausgelegt.
2. Die Dynamik des Regelkreises wird in jeder Betriebsphase erhöht.



Der Funktionsbaustein MB_RegiAccelerationCompensationType01 muss in einer motion synchronen Task aufgerufen werden, damit die Beschleunigung korrekt berechnet wird. Ist dies aus Laufzeitgründen nicht möglich, so kann die Funktion mMotionCyclic() des Funktionsbausteins aus der motion synchronen Task aufgerufen werden und der Funktionsbaustein MB_RegiAccelerationCompensationType01 kann in einer zyklischen Task aufgerufen werden. Der Aufruf in der MotionTask muss folgende Struktur besitzen: {CyclicTask}.fbInstanz.mMotionCyclic(). Die Elemente in der geschweiften Klammer müssen durch den Namen des Programms in der zyklischen Task und durch den Instanznamen des Funktionsbausteins ersetzt werden. Somit könnte der Methodenaufruf aus einem Programm das in der Motion Task läuft wie folgt aussehen: PLCProg.fbRegiAccelerationCompensationType01.mMotionCyclic(). Wird dieser Aufruf nicht durchgeführt, kann eine korrekte Berechnung der Beschleunigung und somit eine fehlerfreie Beschleunigungskompensation nicht gewährleistet werden.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiAdvanced

Tab. 4-9: Referenztabelle Funktionsbaustein MB_RegiAccelerationCompensationType01

Schnittstellenbeschreibung

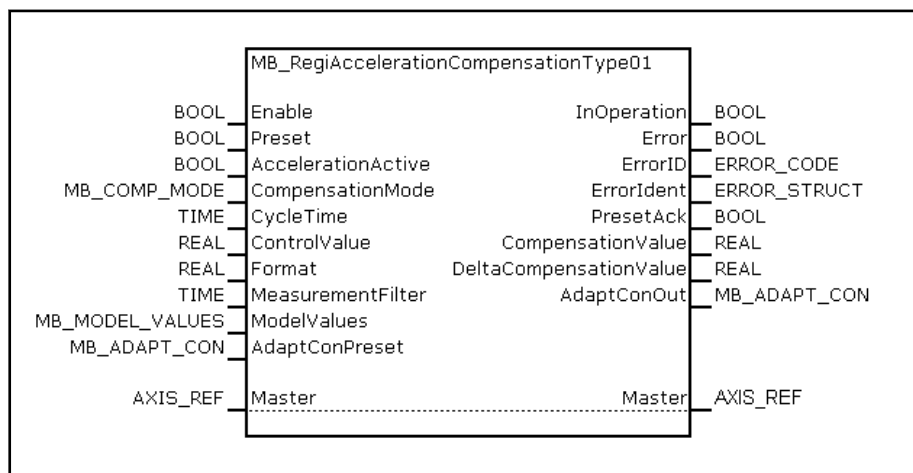


Abb. 4-9: Funktionsbaustein MB_RegiAccelerationCompensationType01

RMB_TechRegiAdvanced

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT CONST	Master	AXIS_REF	Referenz zur Achse zur Ermittlung der Geschwindigkeits- und Beschleunigungskompensation. Es wird aufgrund der Genauigkeit empfohlen eine virtuelle Leitachse anzulegen.
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins (pegelgesteuert).
	Preset	BOOL	Preset der adaptiven Stellwerte.
	AccelerationActive	BOOL	Zeigt dem Funktionsbaustein an, dass eine Beschleunigungsphase beginnt oder beendet wird.
	CompensationMode	MB_COMP_MODE	Auswahl zwischen adaptiver und/oder modellbasierter Kompensation.
	CycleTime	TIME	Zykluszeit der Task der SPS, in der der Funktionsbaustein aufgerufen wird, zur Berechnung der Beschleunigung.
	ControlValue	REAL	Reglerstellsignal in [mm]
	Format	REAL	Druckzylinderumfang in [mm]
	MeasurementFilter	TIME	Filter für den ControlValue.
	ModelValues	MB_MODEL_VALUES	Physikalische Parameter für die modellbasierte Kompensation.
	AdaptConPreset	MB_ADAPT_CON	Adaptive Stellsignale, die bei Preset gesetzt werden.
VAR_OUTPUT	InOperation	BOOL	Baustein arbeitet.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detaildiagnose
	PresetAck	BOOL	Eingang Preset aktiv
	CompensationValue	DINT	Winkeloffset (Modulo 360°) [Grad]
	DeltaCompensationValue	REAL	Schnittstelle zum Entkopplungsnetzwerk in [mm]. Zeigt die differenzierte Änderung an.
	AdaptConOut	STRUCT	Adaptiv verbesserte Stellsignale, die aktuell verwendet werden.

Tab. 4-10: I/O-Schnittstelle des Funktionsbausteins MB_RegiAccelerationCompensationType01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
AccelerationActive	BOOL			FALSE	Kontinuierlich
CompensationMode	MB_COMP_MODE	0	3	MODEL_AND_ADAPTIVE=0	Enable steigende Flanke
CycleTime	TIME	> 0 ms	n. def.	8 ms	Enable steigende Flanke
ControlValue	REAL	n. def.	n. def.	0	Kontinuierlich

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Format	REAL	>0	n. def.	0	Enable steigende Flanke
MeasurementValues	TIME	n. def.	n. def.	0ms	Kontinuierlich (Bei Eingabe 0 ist der Filter deaktiviert)
MB_MODEL_VALUES					
L_DK	REAL	>=0	n. def.	0	Enable steigende Flanke
R_kuehl	REAL	>0	n. def.	1	Enable steigende Flanke
J_kuehl	REAL	>=0	n. def.	0	Enable steigende Flanke
E	REAL	>0	n. def.	1	Enable steigende Flanke
A	REAL	>0	n. def.	1	Enable steigende Flanke
MB_ADAPT_CON					
CA	REAL	n. def.	n. def.	0	Preset steigende Flanke
CP	REAL	n. def.	n. def.	0	Preset steigende Flanke

Tab. 4-11: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

MB_MODEL_VALUES		
L_DK	REAL	Länge zwischen DW_links und Kühlwalze [m]
R_kuehl	REAL	Radius Kühlwalze [mm]
J_kuehl	REAL	Drehträgheit Kühlwalze [kg*m^2]
E	REAL	Elastizitätsmodul Druckbahn [N/m^2]
A	REAL	Querschnittsfläche Druckbahn [m^2]

Tab. 4-12: Datenstruktur MB_MODEL_VALUES

MB_ADAPT_CON		
CA	REAL	Kompensationswert für die Beschleunigungskompensation
CP	REAL	Kompensationswert für die Geschwindigkeitskompensation

Tab. 4-13: Datenstruktur MB_ADAPT_CON

Aufzählungstyp

Der Aufzählungstyp MB_COMP_MODE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
MODEL_AND_ADAPTIVE	0	Modellbasierte und adaptive Kompensation
ADAPTIVE	1	Nur adaptive Kompensation
MODELBASED	2	Nur modellbasierte Kompensation
ADAPTIVE_WITHOUT_LEARNING	3	Es werden die Werte CA und CP verwendet aber nicht nach jeder Beschleunigungsphase aktualisiert.

Tab. 4-14: Aufzählungstyp MB_COMP_MODE

RMB_TechRegiAdvanced

Funktionsbeschreibung

Der Funktionsblock enthält folgende Funktionalitäten:

- Eine **modellbasierte Beschleunigungskompensation** und eine
- **adaptive Geschwindigkeits- und Beschleunigungskompensation**.

Diese ermittelt sich iterativ durch Auswertung des Reglerstellsignals geeignete Kompensationswerte um die Dynamik des Regelkreises zu erhöhen. Die Vorsteuerung erfolgt in linearer Abhängigkeit von der Geschwindigkeit und der Beschleunigung.

Bei steigender Flanke von Preset oder Enable werden die physikalischen Parameter der physikalischen Beschleunigungskompensation MB_MODEL_VALUES übernommen und auf Gültigkeit geprüft. Sollte es sich um einen Druckstoff mit zeitabhängigen Kriech- und Relaxationsverhalten handeln, so ist ein dynamischer E-Modul aus einem rasch durchgeführten Zugversuch zu verwenden. Existiert kein derartiger Wert, so muss er geschätzt werden. Ohne die Vorgabe der physikalischen Parameter ist der Eingang MB_MODEL_VALUES so initialisiert, dass keine modellbasierte Kompensation stattfindet.

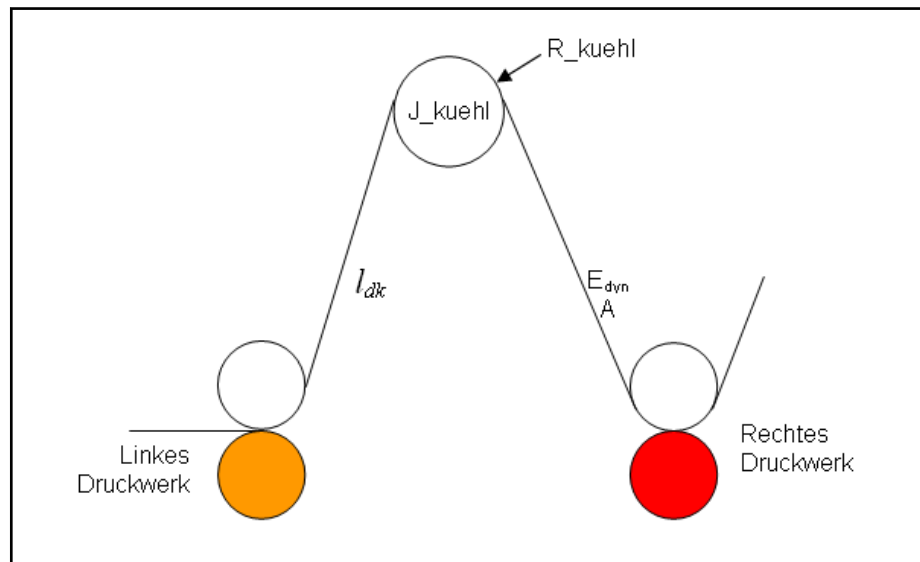


Abb. 4-10: Physikalische Parameter der modellbasierten Kompensation

Preset

Die adaptiven Stellparameter AdaptConPreset werden nur bei Preset neu übernommen. Dies könnte bei einer Auswechslung der Warenbahn sinnvoll sein.

AccelerationActive

Das BOOL AccelerationActive muss gesetzt werden um anzuzeigen, dass eine neue Beschleunigungsphase beginnt. Dies sollte vor Beginn der Beschleunigungsphase geschehen, da hier die erste Reglerstellgröße übernommen wird. Nach Beendigung der Beschleunigungsphase (idealerweise auch eine kurze Zeitspanne danach ~ 1 Sekunde) wird dieses BOOL wieder auf FALSE gesetzt. Mit Inaktivierung des Eingangs AccelerationActive werden die Kompensationswerte (für weitere Beschleunigungsvorgänge) aktualisiert. Solange AccelerationActive gesetzt ist, werden die entsprechenden Kompensationswerte berechnet und ausgegeben.

Die adaptiven Stellwerte werden als AdaptConOut ausgegeben. Für häufige Wechsel der Druckstoffe wird empfohlen, AdaptConOut in einer Rezeptur abzuspeichern. Bei wiederholter Nutzung des Materials können so die Kompensationsparameter erneut verwendet werden.

MB_REGI_ADV_CONTROL_MODE

Es wird empfohlen, sowohl die Funktionalität der modellbasierten als auch der adaptiven Kompensation zu nutzen. Hierzu wird der Eingang CompensationMode auf MODEL_AND_ADAPTIVE initialisiert. Möchte man nur die modellbasierte Kompensation nutzen, setzt man den Eingang auf MODELBASED. Für eine rein adaptive Kompensation wählt man ADAPTIVE. Sollen die ermittelten oder eingetragenen Werte für CA und CP genutzt werden, das Lernen jedoch ausgeschaltet werden, so muss der Modus ADAPTIVE_WITHOUT_LEARNING gewählt werden.

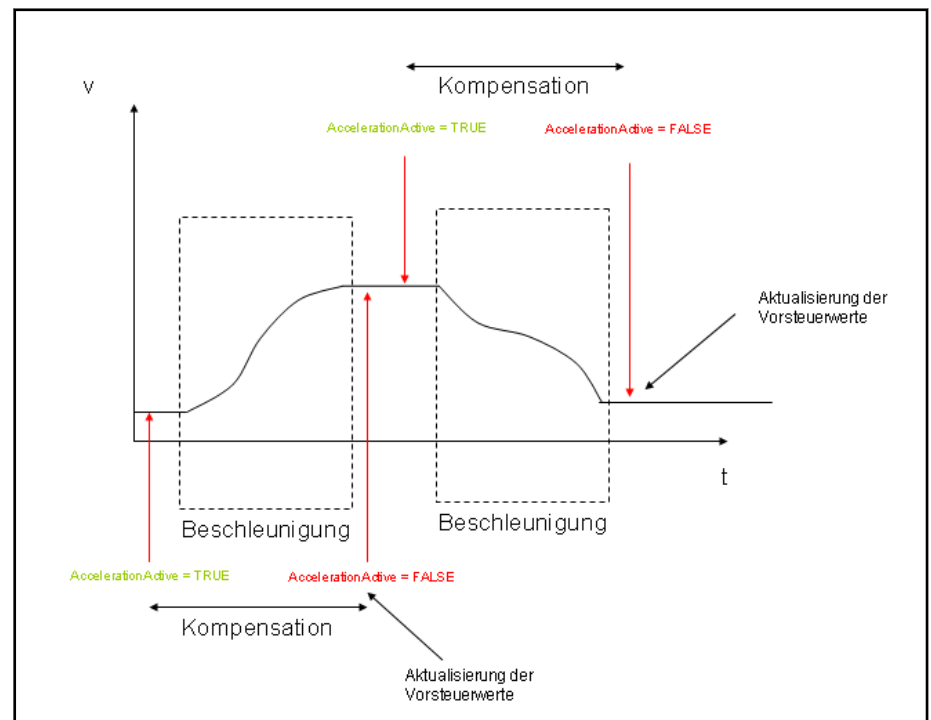


Abb. 4-11: Funktionsweise des MB_RegiAccelerationCompensationType01

In obiger Abbildung ist die Funktionsweise des MB_RegiAccelerationCompensationType01 dargestellt. Solange AccelerationActive = FALSE ist befindet sich der Baustein in der Ruhephase (Enable = TRUE). Wird eine steigende Flanke an AccelerationActive gesetzt, so wird die aktuelle Reglerstellgröße übernommen und bei einer Änderung der Beschleunigung und Geschwindigkeit wird die entsprechende Kompensation vorgenommen. Hierzu wird am Ausgang DeltaCompensationValue der Wert in mm angegeben der an das Entkopplungsnetzwerk übergeben werden muss, in dem dieser Wert aufintegriert und entkoppelt wird. Am Ende der Beschleunigungsphase wird AccelerationActive = FALSE gesetzt. Somit wird der Kompensationsmodus verlassen und es wird die Beschleunigungsphase ausgewertet. Es werden die Werte für CA und CP aktualisiert und es wird wieder in die Ruhephase übergegangen bis eine steigende Flanke an AccelerationActive auftritt.



Idealerweise wird für die Beschleunigung der Funktionsbaustein MB_ReachVelocityType01 verwendet. Hier kann der Ausgang Active mit AccelerationActive verbunden werden. Der Ausgang Active wird 500 ms vor Beginn der Beschleunigungsphase gesetzt und 500 ms nach dem Ende der Beschleunigungsphase zurückgenommen.

RMB_TechRegiAdvanced

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in **Additional 1/Additional 2** folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000031	Ungültige Länge zwischen Druckwerk und Kühlwalze
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000032	Ungültiger Radius der Kühlwalze
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000033	Ungültige Trägheit der Kühlwalze
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000034	Ungültiges E-Modul
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000035	Ungültige Fläche der Materialbahn
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Formzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000036	Ungültige Zykluszeit
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000037	Ungültiger Kompensationsmodus
INPUT_RANGE_ERROR (16#0006)	16#0000000D	16#00000000	AxisNo von "Master" ist außerhalb des gültigen Bereichs

Tab. 4-15: Fehlercodes des Funktionsbausteins MB_RegiAccelerationCompensationType01

4.3.4 MB_RegiGravureRegulateType01

Kurzbeschreibung

Das Entkopplungsnetzwerk ist speziell für Anwendungen mit Klemmstellen an den Druckwerken konzipiert. Es dient dazu Verstellbewegungen in verkoppelten Teilsystemen zu entkoppeln. Es kann hierzu auch in nicht Tiefdruckanwendungen verwendet werden, in denen Klemmstellen vorliegen, die zu verkoppelten Teilsystemen führen. Es stehen unterschiedliche Algorithmen zur Verfügung, wie die Stellgrößen des Registerreglers kaskadiert werden.

Das Entkopplungsnetzwerk wird wie folgt verwendet:

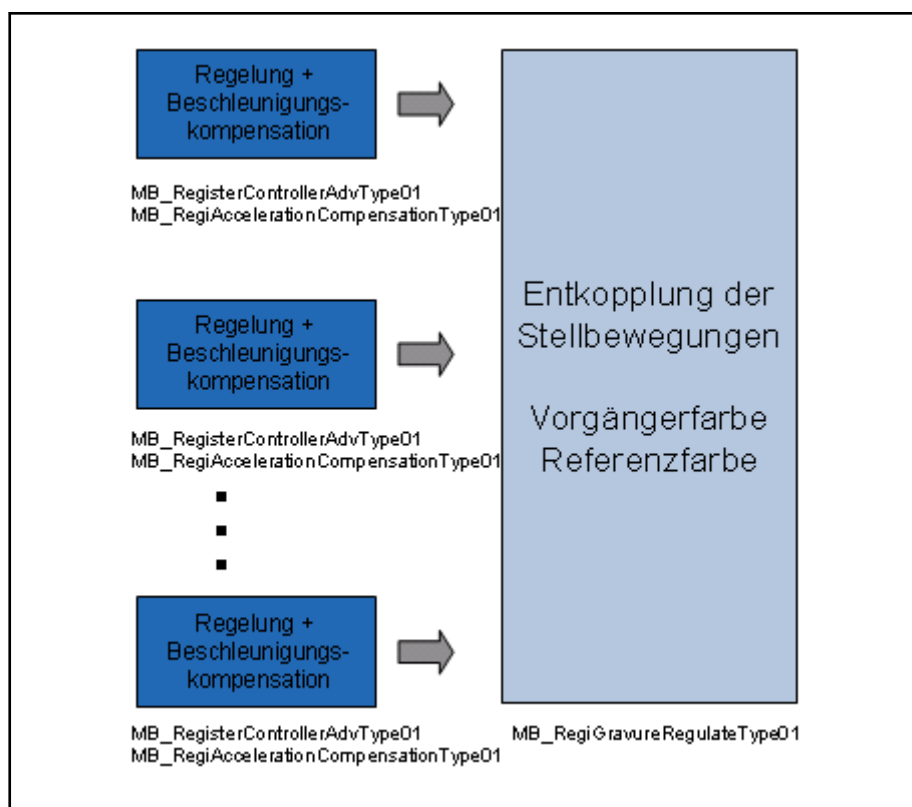


Abb. 4-12: Verwendung des Entkopplungsnetzwerks

Die Stellgrößen der Registerregler werden in dem Entkopplungsnetzwerk entsprechend dem gewählten Algorithmus kaskadiert und für jedes Druckwerk wird ein Winkeloffset berechnet, der ausgegeben wird.

Das Entkopplungsnetzwerk *MB_RegiGravureRegulateType01* unterstützt die folgende Funktionalität:

- Dynamische Kaskadierung der Stellgrößen eines Registerreglers oder der Beschleunigungskompensation
- Es stehen die dynamischen Downstream Varianten für die Vorgängerfarbe und Referenzfarbe zur Verfügung
- Preset Feature zur Vorregistrierung
- Unterstützung eines manuellen Modus pro Druckeinheit
- Unterschiedliche Verstellgeschwindigkeiten je Modus und je Druckeinheit, bei manueller Verstellung oder Verwendung eines externen Registerreglers
- Verwendung des Rexroth RegisterControl Reglers sowie eines externen Registerregler möglich
- Bis zu 16 Druckwerke (Achsen) möglich

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V11	RMB_TechRegiAdvanced

Tab. 4-16: Referenztabelle Funktionsbaustein *MB_RegiGravureRegulateType01*

RMB_TechRegiAdvanced

Schnittstellenbeschreibung

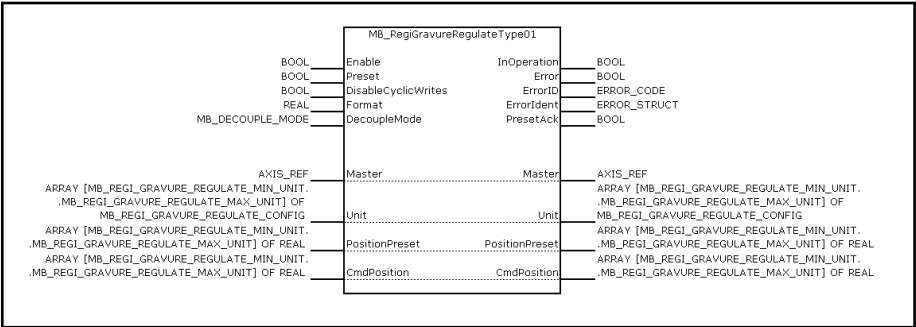


Abb. 4-13: Funktionsbaustein MB_RegiGravureRegulateType01

I/O-Typ	Name	Datentyp	Beschreibung
VAR_IN_OUT CONST	Master	AXIS_REF	Referenz zur Leitachse
VAR_IN_OUT	Unit	ARRAY [MB_REGI_GRAVURE_RE- GULATE_MIN_UNIT .. MB_REGI_GRAVURE_RE- GULATE_MAX_UNIT] OF MB_REGI_GRAVURE_RE- GULATE_CONFIG	Das Array enthält für jedes Druckwerk eine Datenstruktur. In dieser Datenstruktur wird die Achse des jeweiligen Druckwerks angelegt, sowie alle weiteren benötigten Parameter eines Druckwerks.
	PositionPreset	ARRAY [MB_REGI_GRAVURE_RE- GULATE_MIN_UNIT .. MB_REGI_GRAVURE_RE- GULATE_MAX_UNIT] OF REAL	Array mit den Presetwerten für alle Druckwerke in [°].
	CmdPosition	ARRAY [MB_REGI_GRAVURE_RE- GULATE_MIN_UNIT .. MB_REGI_GRAVURE_RE- GULATE_MAX_UNIT] OF REAL	Array mit den Offsetwerten für alle Druckwerke in [°]. Ist DisableCyclicWrites = TRUE muss dieser Wert manuell vom Anwender verschaltet werden.

I/O-Typ	Name	Datentyp	Beschreibung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins
	Preset	BOOL	Das Array CmdPosition wird zurückgesetzt und die Werte aus PositionPreset werden übernommen. Weiterhin wird dieser Wert auch in den Antrieb geschrieben, wenn DisableCyclicWrites = FALSE.
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht an die Achsen geschrieben, sondern nur in CmdPosition ausgegeben.
	Format	REAL	Druckzylinderformat [mm]
	DecoupleMode	MB_DECOUPLE_MODE	Gibt den entsprechenden Algorithmus für die Kaskadierung der Reglerstellgrößen an.
VAR_OUT-PUT	InOperation	BOOL	Der Entkopplungsbaustein arbeitet und die Ausgänge sind gültig.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detailldiagnose
	PresetAck	BOOL	Werte aus PositionPreset wurden in CmdPosition geschrieben bzw. auch auf die Achse.

Tab. 4-17: I/O-Schnittstelle des Funktionsbausteins MB_RegiGravureRegulate-Type01

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Bei steigender Flanke an Enable
Format	REAL	> 0	n. def.	1.0	Bei steigender Flanke an Enable
DecoupleMode	MB_DECOUPLE_MODE	0	1	COLOR_TO_COLOR := 0	Bei steigender Flanke an Enable
MB_REGI_GRAVURE_REGULATE_CONFIG					
Slave	AXIS_REF	NoAxis			Bei steigender Flanke an Enable Steigende Flanke Enable
Active	BOOL			FALSE	Kontinuierlich
Inc	BOOL			FALSE	Kontinuierlich
Dec	BOOL			FALSE	Kontinuierlich
RegiGravure-Control	BOOL			FALSE	Kontinuierlich
RampedControl-ValueOneStep	REAL	n. def	n. def	0.0	Kontinuierlich

RMB_TechRegiAdvanced

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ManualMode	BOOL			FALSE	Kontinuierlich
RedSpeed	BOOL			FALSE	Kontinuierlich
NormalSpeed	REAL	0.0	n. def	1	Kontinuierlich
ReducedSpeed	REAL	0.0	n. def	0.5	Kontinuierlich
WebLength	REAL	1.0	n. def	10000	Bei steigender Flanke an Enable

Tab. 4-18: Übersicht Min/Max-Werte und Initialwerte der Eingänge

Datenstrukturen

MB_REGI_GRAVURE_REGULATE_CONFIG		
Slave	AXIS_REF	Achse des Druckwerks
Active	BOOL	Druckwerk ist aktiv (druckt).
Inc	BOOL	Vertippen, inkrementieren der Position (Manueller Modus oder externer Registerregler).
Dec	BOOL	Vertippen, dekrementieren der Position (Manueller Modus oder externer Registerregler).
RegiGravureControl	BOOL	Der Registerregler MB_RegisterControllerAdvType01 wird verwendet, wenn RegiGravureControl aktiviert ist. Im Automatikmodus und im manuellen Modus wird somit der Eingang RampedControlValueOneStep übernommen, der die Registerverstellung in [mm] angibt.
RampedControlValueOneStep	REAL	Die differenzierte Stellgröße des Registerreglers in [mm].
ManualMode	BOOL	Aktiviert den manuellen Modus. Es findet somit keine Entkopplung der Stellgröße statt.
RedSpeed	BOOL	Auswahl der Verstellgeschwindigkeit
NormalSpeed	REAL	Verstellgeschwindigkeit [1/sec] im Modus RedSpeed = FALSE.
ReducedSpeed	REAL	Verstellgeschwindigkeit [1/sec] im Modus RedSpeed = TRUE.
WebLength	REAL	Länge der Materialbahn vom davorliegenden aktiven Druckwerk bis zum eigenen Druckwerk [mm].

Tab. 4-19: Datenstruktur MB_REGI_GRAVURE_REGULATE_CONFIG

Aufzählungstyp

Der Aufzählungstyp MB_DECOUPLE_MODE kann die folgenden Werte annehmen:

Element	Wert	Beschreibung
COLOR_TO_COLOR	0	Vorgängerfarbenregelung
REFERENCE_COLOR	1	Referenzfarbenregelung

Tab. 4-20: Aufzählungstyp MB_DECOUPLE_MODE

Definition der Defines:

MB_REGI_GRAVURE_REGULATE_MIN_UNIT := 1;

MB_REGI_GRAVURE_REGULATE_MAX_UNIT := 16;

Funktionsbeschreibung

Der Funktionsbaustein MB_RegiGravureRegulateType01 dient als Schnittstelle zwischen den Registerreglern und den Antrieben. Der verwendete Registerregler kann ein externer Registerregler sein, ist aber idealerweise der für den Tiefdruck optimierte Registerregler MB_RegisterControllerAdv-Type01. Die Stellgrößen werden je nach Stellstrategie mit unterschiedlichen Algorithmen an nachfolgende Druckwerke als Vorsteuerung kaskadiert.

Initialisierung

Mit einer steigenden Flanke von Enable startet der FB mit den Werten in CmdPosition. Dieser Wert entspricht für jedes Druckwerk dem Winkeloffset.

Preset

Um die Werte für die CmdPosition zurückzusetzen, muss Preset auf TRUE gesetzt werden. Somit werden die Werte von PositionPreset in das Array CmdPosition kopiert.

DisableCyclicWrites

Der Eingang "DisableCyclicWrites" bestimmt, ob die vom Funktionsbaustein berechneten Werte direkt auf die Achsen geschrieben werden (FALSE) oder nur am Baustein Ausgang in CmdPosition verfügbar sind (TRUE). Wird vom Anwender das AxisInterface oder weitere Funktionalitäten verwendet, die auf diese Achse schreiben, so muss DisableCyclicWrites aktiviert werden. Dann kann der Anwender selbständig den in CmdPosition berechneten Wert an die Achse schreiben (beispielsweise mit dem AxisInterface über PhaseOffset in SyncMode). Hierdurch wird ermöglicht, den Winkeloffset selbständig auf den entsprechenden Parameter zu kopieren, indem man „DisableCyclicWrites“ auf TRUE setzt. Dann ist nur das Ausgangsarray "CmdPosition" zu verwenden und weiter zu modifizieren.

DecoupleMode

Es besteht die Möglichkeit zwischen 2 verschiedenen Kaskadierungsalgorithmen zu wählen. Es wird empfohlen, die Vorgängerfarbenregelung zu verwenden.

- **COLOR_TO_COLOR**

In diesem Modus werden die Korrekturwerte nicht mehr statisch kaskadiert, sondern dynamisch. Das heißt, dass die Korrekturwerte eines Registerreglers über mehrere dynamische Zeitglieder an nachfolgende Druckwerke kaskadiert werden. Somit kann eine ideale Entkopplung der Stellgröße eines vorangehenden Registerreglers erfolgen. Hierzu muss die Länge der Materialbahn zwischen den jeweiligen Druckwerken angegeben werden, damit die Zeitkonstanten für die dynamischen Zeitglieder berechnet werden können.

Die nachfolgende Abbildung zeigt die Kaskadierung der Reglerstellgröße:

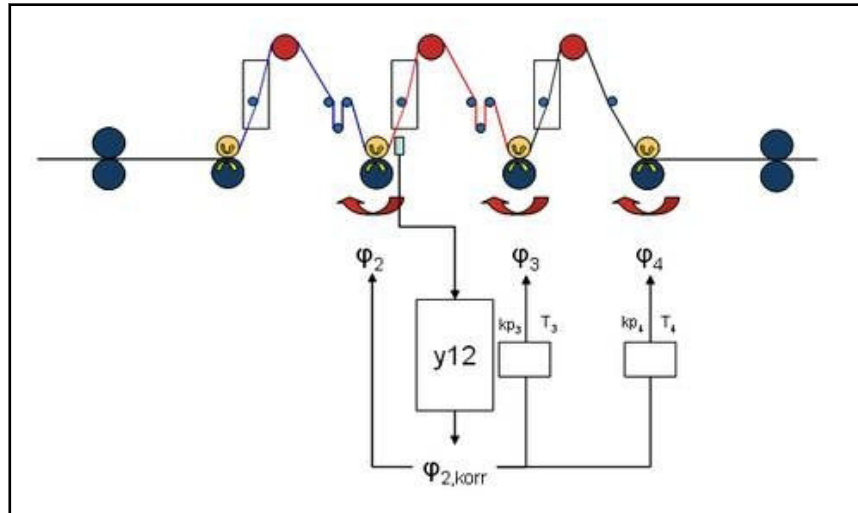


Abb. 4-14: Dynamische Kaskadierung

Die weißen Kästchen stehen für mehrere dynamische Zeitglieder, die die Verstellung des Registerreglers 2 für die Registerregler 3 und 4 vollständig entkoppeln.

Dieser Modus kann nur verwendet werden, wenn der Sensor jeweils die Marken des davorliegenden Druckwerks mit dem aktuellen Druckwerk vergleicht und daraufhin das Register geregelt wird.

- **REFERENCE_COLOR**

Dieser Modus kann nur verwendet werden, wenn der Sensor jeweils die Marken des eigenen Druckwerks mit einer Referenzfarbe (meist Druckwerk 1) vergleicht.

Dieser Modus verhält sich grundsätzlich so wie COLOR_TO_COLOR, besitzt jedoch eine andere Kombination an Dynamikgliedern. Die Beschleunigungskompensation ist nur für den Modus COLOR_TO_COLOR einsetzbar. Aus diesem Grund sollte dieser Modus nur verwendet werden, wenn keine hohe Genauigkeit in der Beschleunigungsphase gefordert ist.

Funktionalitäten, die für das jeweilige Druckwerk eingestellt werden können:

- **RegiGravureControl:**

Ist RegiGravureControl = TRUE, so wird im automatischen und manuellen Modus der Eingang RampedControlValueOneStep als Reglerstellgröße verwendet. Dies ist die Schnittstelle zu dem internen Tiefdruckregler MB_RegisterControllerAdvType01.

Ist RegiGravureControl = FALSE, so werden sowohl im manuellen Modus als auch im automatischen Modus nur die Inc und Dec Eingänge als Reglerstellgröße verwendet (Schnittstelle zu einem externen Registerregler).

- **ManualMode**

Ist ManualMode = TRUE, so werden die Eingangsgrößen nicht für nachfolgende Druckwerke entkoppelt.

Ist ManualMode = FALSE, so werden die Reglerstellgrößen entsprechend der gewählten Stellstrategie für nachfolgende Druckwerke entkoppelt.

- **RedSpeed**

Es können 2 unterschiedliche Verstellgeschwindigkeiten je Druckwerk eingegeben werden. Ist RedSpeed = TRUE, so wird ReducedSpeed als Verstellgeschwindigkeit verwendet, ansonsten NormalSpeed. Dies gilt jedoch nur bei der Verwendung der Vertippeingänge Inc und Dec, also genau dann, wenn RegiGravureControl = FALSE ist.

- Active

Jedes am Druckprozess beteiligte Druckwerk muss mit Active aktiviert werden.

- WebLength

Materialbahnlänge zwischen dem davorliegenden druckenden Druckwerk und dem zu regelnden Druckwerk. Diese Materialbahnlänge wird für die Ermittlung der dynamischen Zeitkonstanten verwendet.

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Druckzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003A	Ungültiger Decouple Mode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000038	Ungültiger NormalSpeed
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000039	Ungültiger ReducedSpeed
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002B	Ungültige Materialbahnlänge
RESOURCE_ERROR (16#0003)	16#00000001	16#00000000	Bei dem in "Slave" angegebenen Antrieb ist keine Leistung zugeschaltet. Eine Registerregelung ist daher nicht möglich.
INPUT_RANGE_ERROR (16#0006)	16#0000000D	16#00000000	AxisNo von "Axis" ist außerhalb des gültigen Bereichs

Tab. 4-21: Fehlercodes des Funktionsbausteins MB_RegiGravureRegulateType01

Weiterhin können Fehler der Bibliotheken RIL_LoopControl und ML_TechBase auftreten.

4.3.5 MB_RegiGravureRegulateType02

Kurzbeschreibung

Der Funktionsbaustein besitzt die komplette Funktionalität des Funktionsbausteins MB_RegiGravureRegulateType01. Aus diesem Grund wird empfohlen die Dokumentation des Funktionsbausteins MB_RegiGravureRegulateType01 zu lesen.

Es werden nur die zusätzlichen Funktionen des Funktionsbausteins MB_RegiGravureRegulateType02 erläutert:

- Durchführung eines Presets für jedes einzelne Druckwerk (Anwendungsfall Vorregistrierung).
- Ignorieren der Vorsteuerwerte davor liegender Druckwerke (Anwendungsfall Bearbeitungseinheit mit davor liegender Entkopplung zum Druckprozess).

Die Erweiterungen beziehen sich auf die Datenstruktur MB_REGI_GRAVURE_REGULATE_CONFIG. Hier werden zusätzliche Variablen, die die entsprechenden Aktionen auslösen in die davon abgeleitete Datenstruktur MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02 eingefügt.

Zielsystem	Bibliothek
Rexroth RegisterControl MLC L65/L45 ab V13	RMB_TechRegiAdvanced

Tab. 4-22: Referenztabelle Funktionsbaustein MB_RegiGravureRegulateType02

Schnittstellenbeschreibung

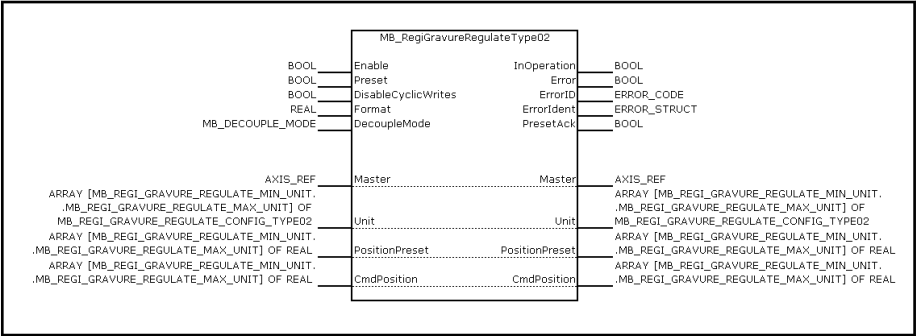


Abb. 4-15: Funktionsbaustein MB_RegiGravureRegulateType02

I/O-Typ	Name	Datentyp	Erläuterung
VAR_IN_OUT CONST	Master	AXIS_REF	Referenz zur Leitachse
VAR_IN_OUT	Unit	ARRAY [MB_REGI_GRAVURE_REGULATE_MIN_UNIT .. MB_REGI_GRAVURE_REGULATE_MAX_UNIT] OF MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02	Das Array enthält für jedes Druckwerk eine Datenstruktur. In dieser Datenstruktur wird die Achse des jeweiligen Druckwerks angelegt, sowie alle weiteren benötigten Parameter eines Druckwerks.
	PositionPreset	ARRAY [MB_REGI_GRAVURE_REGULATE_MIN_UNIT .. MB_REGI_GRAVURE_REGULATE_MAX_UNIT] OF REAL	Array mit den Presetwerten für alle Druckwerke in [°].
	CmdPosition	ARRAY [MB_REGI_GRAVURE_REGULATE_MIN_UNIT .. MB_REGI_GRAVURE_REGULATE_MAX_UNIT] OF REAL	Array mit den Offsetwerten für alle Druckwerke in [°]. Ist DisableCyclicWrites = TRUE muss dieser Wert manuell vom Anwender verschaltet werden.

I/O-Typ	Name	Datentyp	Erläuterung
VAR_INPUT	Enable	BOOL	Bearbeitungsfreigabe des Funktionsbausteins
	Preset	BOOL	Das Array CmdPosition wird zurückgesetzt und die Werte aus PositionPreset werden übernommen. Weiterhin wird dieser Wert auch zum Antrieb geschrieben, wenn DisableCyclicWrites = FALSE.
	DisableCyclicWrites	BOOL	Bei gesetztem Eingang werden die zyklischen Daten nicht an die Achsen geschrieben, sondern nur in CmdPosition ausgegeben.
	Format	REAL	Druckzylinderformat [mm]
	Decouple-Mode	MB_DECOUPLE_MODE	Gibt den entsprechenden Algorithmus für die Kaskadierung der Reglerstellgrößen an.
VAR_OUTPUT	InOperation	BOOL	Der Entkopplungsbaustein arbeitet und die Ausgänge sind gültig.
	Error	BOOL	Bearbeitung mit Fehler abgeschlossen.
	ErrorID	ERROR_CODE	Fehlergruppe
	ErrorIdent	ERROR_STRUCT	Detailldiagnose
	PresetAck	BOOL	Werte aus PositionPreset wurden in CmdPosition geschrieben bzw. auch auf die Achse.

Tab. 4-23: I/O-Schnittstelle des Funktionsbausteins MB_RegiGravureRegulateType02

Datenstrukturen

Die Datenstruktur MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02 erweitert die Datenstruktur MB_REGI_GRAVURE_REGULATE_CONFIG:

MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02		
Slave	AXIS_REF	Achse des Druckwerks.
Active	BOOL	Druckwerk ist aktiv (druckt).
Inc	BOOL	Vertippen, inkrementieren der Position (Manueller Modus oder externer Registerregler).
Dec	BOOL	Vertippen, dekrementieren der Position (Manueller Modus oder externer Registerregler).
RegiGravureControl	BOOL	Der Registerregler MB_RegisterControllerAdvType01 wird verwendet, wenn RegiGravureControl aktiviert ist. Im Automatikmodus und im manuellen Modus wird somit der Eingang RampedControlValueOneStep übernommen, der die Registerverstellung in [mm] angibt.
RampedControlValueOneStep	REAL	Die differenzierte Stellgröße des Registerreglers in [mm].
ManualMode	BOOL	Aktiviert den manuellen Modus. Es findet somit keine Entkopplung der Stellgröße statt.

RMB_TechRegiAdvanced

MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02		
RedSpeed	BOOL	Auswahl der Verstellgeschwindigkeit.
NormalSpeed	REAL	Verstellgeschwindigkeit [1/sec] im Modus RedSpeed = FALSE.
ReducedSpeed	REAL	Verstellgeschwindigkeit [1/sec] im Modus RedSpeed = TRUE.
WebLength	REAL	Länge der Materialbahn vom davor liegenden aktiven Druckwerk bis zum eigenen Druckwerk [mm].
UnitPreset	BOOL	Es wird ein Preset der einzelnen Units mit dem Wert in UnitPresetValue durchgeführt.
UnitPresetAck	BOOL	Bestätigung, dass das UnitPreset erfolgreich ausgeführt wurde.
UnitPresetValue	REAL	Wert der bei steigender Flanke an UnitPreset in CmdPosition der zugehörigen Unit geschrieben wird.
NoDecPreviousUnits	BOOL	Ist der Eingang aktiviert, so werden keine Entkopplungswerte davor liegender Druckwerke berücksichtigt.

Tab. 4-24: Datenstruktur MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02

Aufzählungstyp

Der folgende Aufzählungstyp wurde bei den vorherigen Funktionsbausteinen der Bibliothek RMB_TechRegiAdvanced beschrieben:

- MB_DECOUPLE_MODE (Tab. 4-20 "Aufzählungstyp MB_DECOUPLE_MODE" auf Seite 152)

Definition der Standardeinstellungen:

MB_REGI_GRAVURE_REGULATE_MIN_UNIT := 1;

MB_REGI_GRAVURE_REGULATE_MAX_UNIT := 16;

Min/Max-Werte und Standardwerte der Eingänge

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
Enable	BOOL			FALSE	Kontinuierlich
Preset	BOOL			FALSE	Kontinuierlich
DisableCyclicWrites	BOOL			FALSE	Bei steigender Flanke an Enable
Format	REAL	> 0	n. def.	1.0	Bei steigender Flanke an Enable
DecoupleMode	MB_DECOUPLE_MODE	0	1	COLOR_TO_COLOR := 0	Bei steigender Flanke an Enable
MB_REGI_GRAVURE_REGULATE_CONFIG_TYPE02					
Slave	AXIS_REF	NoAxis			Bei steigender Flanke an Enable
Active	BOOL			FALSE	Kontinuierlich
Inc	BOOL			FALSE	Kontinuierlich
Dec	BOOL			FALSE	Kontinuierlich
RegiGravureControl	BOOL			FALSE	Kontinuierlich
RampedControlValueOneStep	REAL	n. def.	n. def.	0.0	Kontinuierlich

Name	Typ	Min.-Wert	Max.-Wert	Default-Wert	Übernahme
ManualMode	BOOL			FALSE	Kontinuierlich
RedSpeed	BOOL			FALSE	Kontinuierlich
NormalSpeed	REAL	0.0	n. def.	1	Kontinuierlich
ReducedSpeed	REAL	0.0	n. def.	0.5	Kontinuierlich
WebLength	REAL	1.0	n. def.	10000	Bei steigender Flanke an Enable
UnitPreset	BOOL			FALSE	Kontinuierlich
UnitPresetAck	BOOL			FALSE	Kontinuierlich
UnitPresetValue	REAL	n. def.	n. def.	0.0	Steigende Flanke an UnitPreset
NoDecPreviousUnits	BOOL			FALSE	Kontinuierlich

Tab. 4-25: Übersicht Min/Max-Werte und Initialwerte der Eingänge MB_RegiGravureRegulateType02

Funktionsbeschreibung



Der Funktionsbaustein MB_RegiGravureRegulateType02 besitzt die komplette Funktionalität des Funktionsbausteins MB_RegiGravureRegulateType01.

Für die Funktionsbeschreibung empfehlen wir folgende Dokumentation zu lesen:

- MB_RegiGravureRegulateType01 (Kap. 4.3.4 "MB_RegiGravureRegulateType01" auf Seite 148)

Zusätzlich besitzt der Funktionsbaustein 2 weitere Funktionalitäten:

Preset für einzelnes Druckwerk durchführen

Bisher kann nur ein globaler Preset aller Druckwerke durchgeführt werden. Dies kann anwendungsspezifisch nicht ausreichend sein, da häufig nur Presetwerte eines Druckwerks bzw. einer Unit geschrieben werden. Wird Unit.UnitPreset := TRUE gesetzt (steigende Flanke), so wird der Wert aus Unit.UnitPresetValue in CmdPosition für diese eine Unit geschrieben. Das erfolgreiche Schreiben des Wertes wird über Unit.UnitPresetAck angezeigt. Ist DisableCyclicWrites = FALSE, dann wird dieser Wert zusätzlich noch an die entsprechende Slave Achse geschrieben.

Keine Entkopplungswerte davorliegender Druckwerke berücksichtigen

Eine Unit, die davor liegende Units besitzt, die aktiv sind und nicht im manuellen Modus arbeiten, hat immer Vorsteuerwerte von davor liegenden Units erhalten. Aus prozesstechnischen Gründen kann es möglich sein, dass eine Unit von Verstellungen davorliegender Einheiten entkoppelt ist. Über den Eingang NoDecPreviousUnits = TRUE können nun alle Vorsteuerwerte davorliegender Druckwerke ignoriert werden. Ist NoDecPreviousUnits = FALSE, so werden die Vorsteuerwerte davorliegender Druckwerke, die sich nicht im manuellen Modus befinden, berücksichtigt. In der Regel sollten bei Druckwerken immer die Vorsteuerwerte berücksichtigt werden. Die Deaktivierung dieser Vorsteuerwerte sollte nur im Ausnahmefall vorgenommen werden.

RMB_TechRegiAdvanced

Fehlerbehandlung

Der Funktionsbaustein nutzt die Fehlertabelle **F_RELATED_TABLE**, 16#0170.

Er kann in Additional 1/Additional 2 folgende Fehlermeldungen erzeugen:

ErrorID	Additional 1	Additional 2	Beschreibung
STATE_MACHINE_ERROR (16#0005)	16#00000006	16#00000000	Fehler in der State Machine
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000000	Ungültiges Druckzylinderformat
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000003A	Ungültiger Decouple Mode
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000038	Ungültiger NormalSpeed
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#00000039	Ungültiger ReducedSpeed
INPUT_RANGE_ERROR (16#0006)	16#00001D11	16#0000002B	Ungültige Materialbahnlänge
INPUT_RANGE_ERROR (16#0006)	16#0000000D	16#00000000	AxisNo von "Axis" ist außerhalb des gültigen Bereichs
RESOURCE_ERROR (16#0003)	16#00000001	16#00000000	Bei dem in "Slave" angegebenen Antrieb ist keine Leistung zugeschaltet. Eine Registerregelung ist daher nicht möglich.

Tab. 4-26: Fehlercodes des Funktionsbausteins MB_RegiGravureRegulate-Type02

Weiterhin können Fehler der Bibliotheken RIL_LoopControl und ML_TechBase auftreten.

5 Service und Support

Für Ihre schnelle und optimale Unterstützung verfügen wir über ein dichtes weltweites Servicenetz. Unsere Experten stehen Ihnen mit Rat und Tat zur Seite. Sie erreichen uns täglich **rund um die Uhr – auch an Wochenenden und Feiertagen**.

Service Deutschland

Unser technologieorientiertes Competence Center in Lohr deckt alle Belange rund um den Service für elektrische Antriebe und Steuerungen ab.

Sie erreichen unsere **Service-Hotline** und unseren **Service-Helpdesk** unter:

Telefon: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-Mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Auf unseren Internetseiten finden Sie ergänzende Hinweise zu Service, Reparatur (z. B. Anlieferadressen) und Training.

Service weltweit

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem Ansprechpartner auf. Die Hotline-Rufnummern entnehmen Sie bitte den Vertriebsadressen im Internet.

Vorbereitung der Informationen

Wir können Ihnen schnell und effizient helfen, wenn Sie folgende Informationen bereithalten:

- Eine detaillierte Beschreibung der Störung und der Umstände
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern
- Ihre Kontaktdaten (Telefon-, Faxnummer und E-Mail-Adresse)

Index

A

Anregungen.....	11
ANSI Z535.6-2006.....	9
Aufzählungstyp	
MB_COMP_MODE.....	145
MB_DECOUPLE_MODE.....	152
MB_REGI_ADV_CONTROL_MODE.....	137
MB_REGI_ADV_RESPONSE_MODE.....	137
MB_REGI_MARK_STATE.....	67, 110
MB_REGI_MEASURE_HANDLING.....	44, 92
MB_REGI_READ_WRITE_TEACH_DATA..	121
MB_REGI_SENSOR_STATE.....	67, 111
MB_REGI_SENSOR01_COMM_AC-	
TIONS.....	27, 74
MB_REGI_SENSOR01_COMM_STA-	
TUS	26, 73

B

Beschwerde.....	11
Bestimmungsgemäßer Gebrauch	
Anwendungsbereiche.....	13
Einleitung.....	13
Einsatzfälle.....	13

D

Datenstruktur	
MB_ADAPT_CON.....	145
MB_MODEL_VALUES.....	145
MB_REGI_BARCODE.....	42, 90
MB_REGI_COMMUNICATION_CONFIG	43, 91
MB_REGI_FORMAT.....	42, 90
MB_REGI_GRAVURE_REGULA-	
TE_CONFIG.....	152
MB_REGI_GRAVURE_REGULA-	
TE_CONFIG_TYPE02.....	158
MB_REGI_INTERNAL_PARAMETER.....	43
MB_REGI_INTERNAL_PARAME-	
TER_TYPE02.....	73, 91
MB_REGI_LENGTH_REGISTER_MODE	42, 91
MB_REGI_MARK_STREAM.....	42, 90
MB_REGI_MARK_STRUCTURE.....	42, 90
MB_REGI_RECEIVE_DATA.....	43, 92
MB_REGI_SEND_DATA.....	43, 92
MB_REGI_SENSOR01_CFG.....	42
MB_REGI_SENSOR01_CFG_TYPE02.....	90
MB_REGI_SENSOR01_DATA.....	66, 110
MB_REGI_SENSOR01_DIAG.....	114
MB_REGI_SENSOR01_TEACH_DATA.....	121
MB_REGI_SIDE_REGISTER_MODE.....	43, 91
MB_REGI_TOLERANCES.....	43, 91
MB_REGI_WARNING.....	26, 73

E

Erzeugen bestimmter Geschwindigkeitsverläufe	
Einführung und Übersicht.....	124

F

Feedback.....	11
Fehlercodes	
MB_ReachVelocityType01.....	127
MB_RegiAccelerationCompensation-	
Type01.....	148
MB_RegiGravureRegulateType01.....	155
MB_RegiGravureRegulateType02.....	160
MB_RegiSensor01CommType01.....	39
MB_RegiSensor01CommType02.....	87
MB_RegiSensor01ConfigType01.....	56
MB_RegiSensor01ConfigType02.....	105
MB_RegiSensor01CycComType01.....	64
MB_RegiSensor01CycComType02.....	68
MB_RegiSensor01CycComType04.....	112
MB_RegiSensor01ReadDiagDataType01...	115
MB_RegiSensor01SendRecvDataType01.....	60
MB_RegiSensor01SendRecvDataType02...	108
MB_RegiSensor01SendSpeedAccType01..	119
MB_RegiSensor01SetIPAddressType01.....	124
MB_RegiSensor01TeachDataType01.....	121
MB_RegisterControllerAdvType01.....	142

G

Gefahrenhinweise.....	9
-----------------------	---

H

Helpdesk.....	161
Hotline.....	161

K

Kritik.....	11
Kundenfeedback.....	11

M

MB_ReachVelocityType01	
Funktionsbeschreibung.....	126, 127
Kurzbeschreibung.....	124
Min/Max-Werte und Standardwerte der	
Eingänge.....	125
Schnittstellenbeschreibung.....	125
MB_RegiAccelerationCompensationType01	
Aufzählungstyp.....	145
Datenstrukturen.....	145
Fehlerbehandlung.....	148
Funktionsbeschreibung.....	146
Kurzbeschreibung.....	142
Min/Max-Werte und Standardwerte der	
Eingänge.....	144
Schnittstellenbeschreibung.....	143
MB_RegiGravureRegulateType01	
Aufzählungstyp.....	152
Datenstrukturen.....	152
Fehlerbehandlung.....	155
Funktionsbeschreibung.....	153

Index

Kurzbeschreibung.....	148	Datenstrukturen.....	66
Min/Max-Werte und Standardwerte der		Fehlerbehandlung.....	68
Eingänge.....	151	Funktionsbeschreibung.....	67
Schnittstellenbeschreibung.....	150	Kurzbeschreibung.....	64
MB_RegiGravureRegulateType02		Min/Max-Werte und Standardwerte der	
Aufzählungstyp.....	158	Eingänge.....	66
Datenstrukturen.....	157	Schnittstellenbeschreibung.....	65
Fehlerbehandlung.....	160	MB_RegiSensor01CycComType04	
Funktionsbeschreibung.....	159	Aufzählungstyp.....	110
Kurzbeschreibung.....	155	Datenstrukturen.....	110
Min/Max-Werte und Standardwerte der		Fehlerbehandlung.....	112
Eingänge.....	158	Funktionsbeschreibung.....	111
Schnittstellenbeschreibung.....	156	Kurzbeschreibung.....	108
MB_RegiSensor01CommType01		Min/Max-Werte und Standardwerte der	
Aufzählungstyp.....	26	Eingänge.....	110
Datenstrukturen.....	25	Schnittstellenbeschreibung.....	109
Fehlerbehandlung.....	39	MB_RegiSensor01ReadDiagType01	
Funktionsbeschreibung.....	29	Datenstrukturen.....	114
Kurzbeschreibung.....	22	Funktionsbeschreibung.....	114
Min/Max-Werte und Standardwerte der		Kurzbeschreibung.....	112
Eingänge.....	27	Min/Max-Werte und Standardwerte der	
Schnittstellenbeschreibung.....	23	Eingänge.....	113
MB_RegiSensor01CommType02		Schnittstellenbeschreibung.....	113
Aufzählungstyp.....	73	MB_RegiSensor01SendRecvDataType01	
Datenstrukturen.....	72	Datenstrukturen.....	58
Fehlerbehandlung.....	87	Fehlerbehandlung.....	60
Funktionsbeschreibung.....	77	Funktionsbeschreibung.....	58
Kurzbeschreibung.....	68	Kurzbeschreibung.....	56
Min/Max-Werte und Standardwerte der		Min/Max-Werte und Standardwerte der	
Eingänge.....	74	Eingänge.....	58
Schnittstellenbeschreibung.....	70	Schnittstellenbeschreibung.....	57
MB_RegiSensor01ConfigType01		MB_RegiSensor01SendRecvDataType02	
Aufzählungstyp.....	44	Datenstrukturen.....	107
Fehlerbehandlung.....	55	Fehlerbehandlung.....	108
Funktionsbeschreibung.....	46	Funktionsbeschreibung.....	107
Kurzbeschreibung.....	40, 41	Kurzbeschreibung.....	105
Min/Max-Werte und Standardwerte der		SchnittsMin/Max-Werte und Standard-	
Eingänge.....	44	werte der Eingängetellenbeschreibung.....	106
Schnittstellenbeschreibung.....	40	Schnittstellenbeschreibung.....	105
MB_RegiSensor01ConfigType02		MB_RegiSensor01SendSpeedAccType01	
Aufzählungstyp.....	92	Datenstrukturen.....	117
Datenstrukturen.....	89	Fehlerbehandlung.....	119
Fehlerbehandlung.....	104	Funktionsbeschreibung.....	117
Funktionsbeschreibung.....	94	Kurzbeschreibung.....	115
Kurzbeschreibung.....	87	Min/Max-Werte und Standardwerte der	
Min/Max-Werte und Standardwerte der		Eingänge.....	117
Eingänge.....	92	Schnittstellenbeschreibung.....	116
Schnittstellenbeschreibung.....	88	MB_RegiSensor01SetIPAddressType01	
MB_RegiSensor01CycComType01		Fehlerbehandlung.....	123
Datenstrukturen.....	62	Funktionsbeschreibung.....	123
Fehlerbehandlung.....	63	Kurzbeschreibung.....	122
Funktionsbeschreibung.....	62	Min/Max-Werte und Standardwerte der	
Kurzbeschreibung.....	60	Eingänge.....	122
Min/Max-Werte und Standardwerte der		Schnittstellenbeschreibung.....	122
Eingänge.....	62	MB_RegiSensor01TeachDataType01	
Schnittstellenbeschreibung.....	61	Aufzählungstyp.....	121
MB_RegiSensor01CycComType02		Datenstrukturen.....	121
Aufzählungstyp.....	66	Fehlerbehandlung.....	121

Index

Funktionsbeschreibung.....	121
Kurzbeschreibung.....	119
Min/Max-Werte und Standardwerte der Eingänge.....	120
Schnittstellenbeschreibung.....	120
MB_RegisterControllerAdvType01	
Aufzählungstyp.....	137
Funktionsbeschreibung.....	138
Kurzbeschreibung.....	132
Min/Max-Werte und Standardwerte der Eingänge.....	136
Schnittstellenbeschreibung.....	134

N

Nicht-bestimmungsgemäßer Gebrauch.....	14
Folgen, Haftungsausschluss.....	13

R

Regelung und Optimierung des Druckprozesses	
Einführung und Übersicht.....	131
MB_RegiAccelerationCompensation- Type01.....	142
MB_RegiGravureRegulateType01.....	148
MB_RegisterControllerAdvType01.....	132
RMB_TechRegiAdvanced.....	129
Allgemeines.....	131
Lizenzpflichtige Bibliothek.....	129
MB_RegiGravureRegulateType02.....	155
Regelung und Optimierung des Druck- prozesses.....	131
RMB_TechRegiExtended.....	15
Allgemeines.....	17
Einführung und Übersicht.....	18
Erzeugen bestimmter Geschwindigkeits- verläufe.....	124
Kommunikation mit dem Markenstromsensor	18
Lizenzpflichtige Bibliothek.....	15
MB_ReachVelocityType01.....	124
MB_RegiSensor01CommType01.....	22
MB_RegiSensor01CommType02.....	68
MB_RegiSensor01ConfigType01.....	40
MB_RegiSensor01ConfigType02.....	87
MB_RegiSensor01CycComType01.....	60
MB_RegiSensor01CycComType02.....	64
MB_RegiSensor01CycComType04.....	108
MB_RegiSensor01ReadDiagType01.....	112
MB_RegiSensor01SendRecvDataType01.....	56
MB_RegiSensor01SendRecvDataType02...	105
MB_RegiSensor01SendSpeedAccType01..	115
MB_RegiSensor01SetIPAddressType01.....	122
MB_RegiSensor01TeachDataType01.....	119

S

Service-Hotline.....	161
Sicherheitshinweise.....	9
Signalgrafik.....	9
Signalwörter.....	9

Support.....	161
Symbole.....	11

W

Warnhinweise.....	9
-------------------	---

Z

Zu dieser Dokumentation.....	3
Darstellung von Informationen.....	9
Gültigkeit der Dokumentation.....	3

Notizen

Notizen

Bosch Rexroth AG

Electric Drives and Controls

Postfach 13 57

97803 Lohr, Deutschland

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Deutschland

Tel. +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911341508