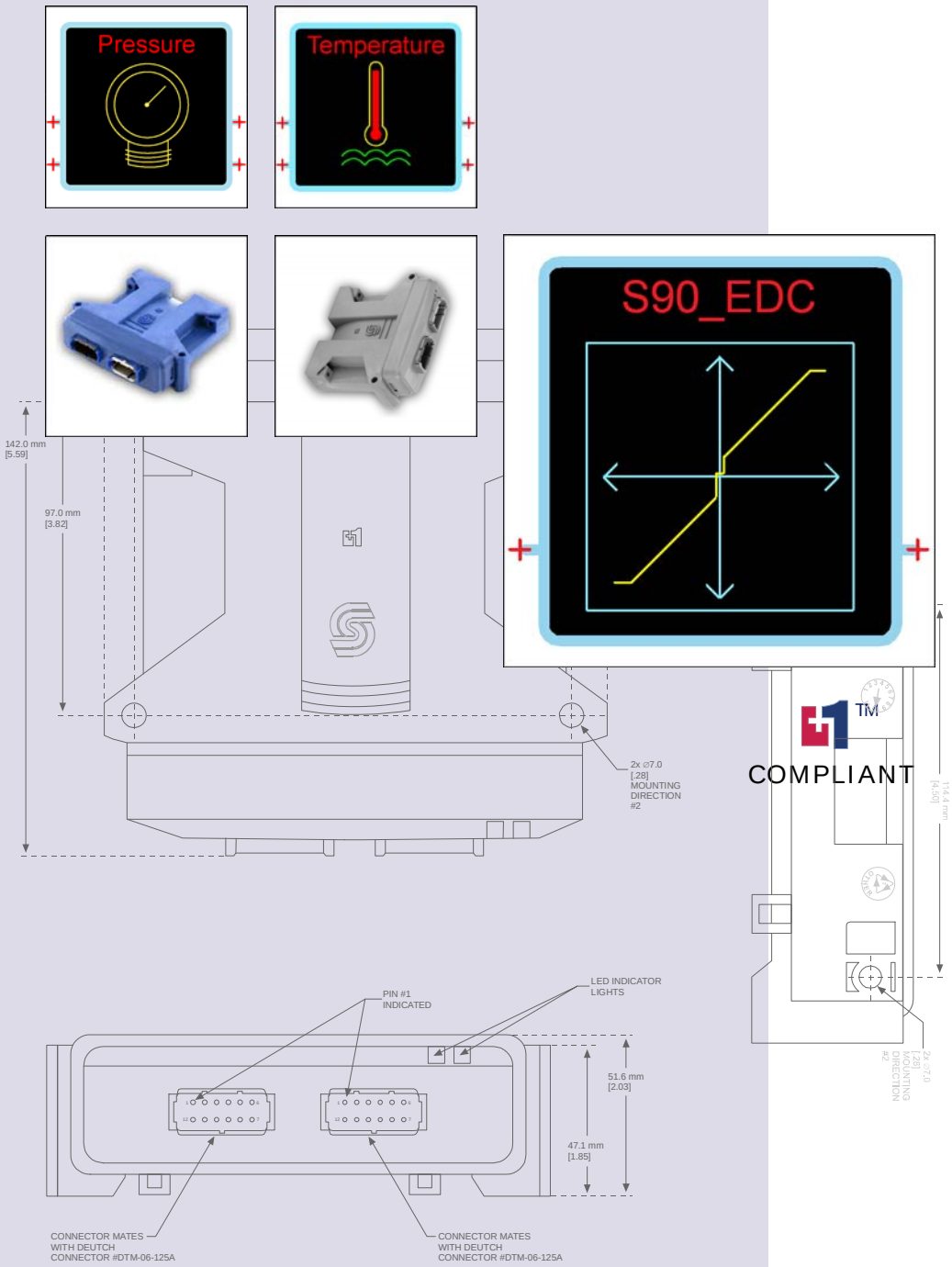
PLUS+1™ GUIDE
Software

PLUS+1 Compliant S90 EDC Control Function Block User Manual



About this Manual

Organization and Headings

To help you quickly find information in this manual, the material is divided into sections, topics, subtopics, and details, with descriptive headings set in red type. Section titles appear at the top of every page in large red type.

In the PDF version of this document, clicking an item *underlined in blue italic type* jumps you to the referenced page in the document.

Special Text Formatting

Controls and indicators are set in **bold black type**.

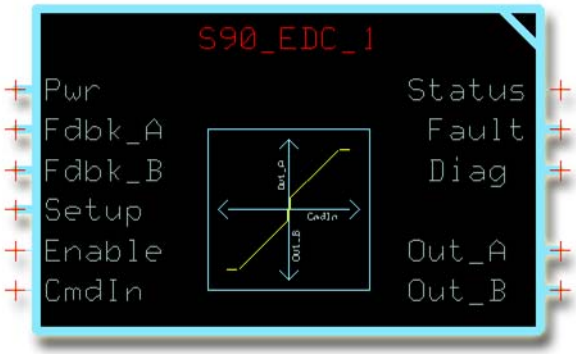
Table of Contents

A Table of Contents (TOC) appears on the next page. In the PDF version of this document, the TOC entries are hyperlinked.

Revision History

Revision	Date	Comment
Rev CA	October 2011	

Contents



Overview

The output of an **S90_EDC** function block interfaces between your application and a Sauer-Danfoss S90 EDC (Electronic Displacement Control). The output of this function block drives the coils that control flow direction and magnitude through a pump's A and B ports.

PLUS+1 I/O modules do not have the fault-checking feedback signals that are needed by this function block. An application that uses this function block to control an I/O module will fail to compile.

See:

- *About Function Block Connections* on page *9* for an overview of this function block's connections and signals.
- *About the Name Space Feature* on page *25* if you are using more than one of these function blocks in your application.

Inputs

S90_EDC *Function Block Inputs*

Input	Type	Range	Description
Pwr	——	——	Reports controller power supply voltage. The function block uses this voltage when it calculates the measured resistance of the control circuit.
Fdbk_A	——	——	Reports the status of the **MFOut** (Multifunction Output) block that receives the function block's **Out_A** (Output A) bus. Each **MFOut** block in the **Outputs** page has a corresponding **MFOut** block in the **Inputs** page that reports on its status through a **Status** bus. The **Inputs** bus on the **Application** page contains these **Status** buses.
Fdbk_B	——	——	Reports the status of the **MFOut** (Multifunction Output) block that receives the function block's **Out_B** (Output B) bus. Each **MFOut** block in the **Outputs** page has a corresponding **MFOut** block in the **Inputs** page that reports on its status through a **Status** bus. The **Inputs** bus on the **Application** page contains these **Status** buses.

S90_EDC *Function Block Inputs*

Input	Type	Range	Description
Setup	——	——	Allows common configuration to be applied to multiple function blocks. See *About Modifying the Config_Data Page* on page *15* for more information.
Enable	BOOL	——	Enables the **OutputValue** signals in the **Out_A** and **Out_B** buses. – T = **OutputValue** signals follow the **CmdIn** signal. – F = Holds both **OutputValue** signals at zero. – F/T = Clears latched faults if **CmdIn** is zero.
CmdIn	S16	±10000	Specifies the requested speed and direction. – +10000 = Requests maximum **Out_A** speed. – 0 = Requests neutral (stop). – –10000 = Requests maximum **Out_B** speed.

Function Block Internal Constants

The following table lists constant values that are provided in the function blocks for setting limits. These constants cannot be edited.

S90_EDC *Internal Constants*

Constant	Value	Description
MaxCrnt	100 mA	The maximum current for displacement or proportional control.
Nominal Ω	18.0 Ω	The resistance of the load normally connected to **Out_A** and **Out_B**.

Function Block Parameters

Warning

Using the PLUS+1 Service Tool program download new parameter values to an application can result in unexpected and sudden machine movements.

Unexpected and sudden machine movements can result in personal injury and equipment damage.

Always secure your machine against unexpected and sudden movements before you use the Service Tool program download new parameter values.

The following table lists function block parameters that are stored in the controller's non-volatile memory.

You can change these values by:

- Execution of the calibration process.
- Direct access to serial EE memory using the PLUS+1 Service Tool.
- Recalling default values.

***S90_EDC** Function Block Parameters*

Parameter	Type	Range	Description
A.EE_Thld, **B.EE_Thld**	U16	**CalThldMin** to **CalThldMax**	Calibrated threshold values for **Out_A** and **Out_B**.
A.EE_EndCrnt **B.EE_EndCrnt**	U16	**Thld** + 1 to **MaxCurrent**	Calibrated end current values for **Out_A** and **Out_B**.
A.EE_CalFlg, **B.EE_CalFlg**	U8	——	Calibration flags for **Out_A** and **Out_B**. – Bit 1 = 1 (0x0001)—Threshold is not calibrated. – Bit 2 = 1 (0x0002)—End current is not calibrated. The **CalFlg** signal in the **Diag** bus contains both pairs of flags, with the **Out_B** bits shifted to positions 3 and 4.

These parameters determine the values used as threshold and end current for each direction during normal operation. However, the values actually applied might be different because:

- **Thld** and **EndCrnt** are both subject to range limits.
- **Thld** is modified by the **ThldMult** value.

The following intermediate values represent the values that are actually applied:

- **EndCrntApplied** = MIN(**EE_EndCrnt**, **MaxCrnt**).
- **ThldApplied** = MIN((**EE_Thld** × **ThldMult**), **EndCrntApplied**).

This assures that:

- **EndCrntApplied** never exceeds **MaxCrnt**.
- **ThldApplied** never exceeds **EndCrntApplied**.

Also note that, while the **CalTask** value is set to select a calibration task:

- **EndCrntApplied** = **MaxCrnt**.
- **ThldApplied** = 1.

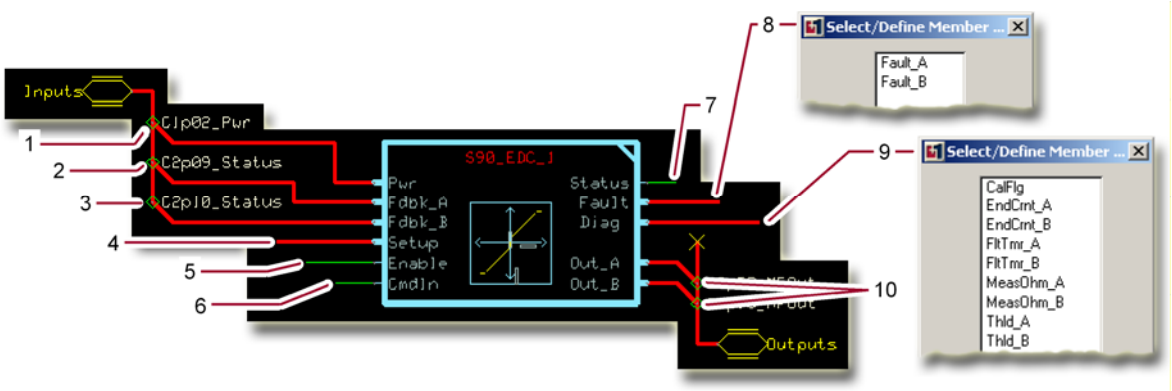
The applied values are available in the **Diag** bus as **Thld_A**, **Thld_B**, **EndCrnt_A**, and **EndCrnt_B**.

Outputs

S90_EDC Function Block Outputs

Output	Type	Range	Description
Status	U16	——	Reports the function block's status conditions. This output uses the standard bitwise scheme described in the *Basic Function Blocks Library User's Manual*. For more information about status logic, see *Status and Fault Logic* on page *10*.
Fault	——	——	Reports the function block's fault conditions. This output uses the standard bitwise scheme described in the *Basic Function Blocks Library User's Manual*. For more information about fault logic, see *Status and Fault Logic* on page *10*.
Fault_A	U16	——	Reports fault conditions for the A output.
Fault_B	U16	——	Reports fault conditions for the B output.
Diag	——	——	Outputs a bus with the following signals: – **CalFlg** (Calibration Flag). – **EndCrnt_A** (End Current A). – **FltTmr_A** (Fault Timer A). – **MeasOhm** (Ohm A). – **Thld_A** (Threshold A). – **EndCrnt_B** (End Current B). – **FltTmr_B** (Fault Timer B). – **MeasOhm_B** (Ohm B). – **Thld_B** (Threshold B). The **Diag** bus also has **Config** and **Define** sub-buses that report the configuration and setup values used by the function block. Use these signals for troubleshooting.
CalFlg	U8	——	The **CalFlg** signal is a bitmask value that indicates the state of calibration values. – Bit 1 = 1 (0x0001)—Threshold A is not calibrated. – Bit 2 = 1 (0x0002)—End current A is not calibrated. – Bit 3 = 1 (0x0004)—Threshold B is not calibrated. – Bit 4 = 1 (0x0008)—End current B is not calibrated. Use these flags to manage the calibration process.
EndCrnt_A, **EndCrnt_B**	U16	——	Report the active end current value for the A and B outputs. During: – Normal operation, end current values equal **A.EE_EndCrnt** and **B.EE_EndCrnt**. – Calibration, the active end current values equal **MaxCrnt**.
FltTmr_A, **FltTmr_B**	U16	——	Reports the value of the forward fault delay timer.
MeasOhm_A, **MeasOhm_B**	U16	——	Indicate the measured resistance of the control circuits that drive the pump's coils.
Out_A	U16	0-8500	Has an **OutputValue** signal that drives the A coil.
Out_B	U16	0-8500	Has an **OutputValue** signal that drives the B coil.

About Function Block Connections



Function Block Connections

Item	Description
1	Controller power supply voltage.
2	Status of **Out_A**.
3	Status of **Out_B**.
4	Input for configuration values from an external source.
5	T enables **Out_A** and **Out_B**.
6	Commands direction and speed.
7	Reports the status of the function block.
8	Reports the faults of the function block.
9	Diagnostic signals.
10	Drives the A and B coils.

Status and Fault Logic

The following table lists function block status codes. These codes indicate the calibration state of the function block.

Status Logic

Status	Bit*	Reported While
Block is not calibrated.	1	Any **CalFlg** bit is set to 1.
Calibration active.	2	**Enable** is T and the **CalTask** = –2, –1, +1, or +2.
Parameters are corrupt.	3	(**EE_EndCrnt** > **MaxCrnt**) or (**EE_Thld** × **ThldMult**) > **EndCrntApplied**) for either direction A or direction B. The actual tests in the software are: – (**EndCrntApplied** ≠ **EE_EndCrnt**)—happens when (**EE_EndCrnt** > **MaxCrnt**). It means that **EndCrntApplied** = **MaxCrnt**. – (**ThldApplied** = **EndCrntApplied**)—happens when ((**EE_Thld** × **ThldMult**) ≥ **EndCrntApplied**).
Invalid setup/calibration.	4	Any of the following setup and configuration values are not within their valid ranges: – **CalThldMin**. – **CalThldMax**. – **ThldMult**. – **FltDelay**. – **CalTask**. Also reported while **PinStatus** reports a configuration error (value =1).
Value too low.	7	Threshold calibration is active and the **OutputValue** < **CalThldMin** for the calibration direction.
Value too high.	8	Threshold calibration is active and the **OutputValue** > **CalThldMax** for the calibration direction.

*Position of set bit in a 16 bit fault code or status code. Bit 1 is the least significant bit. Bit 16 set to 1 identifies a standard Sauer-Danfoss status code or fault code.

The function block's **Out_A** bus contains an **OutputValue** signal that drives a forward EDC coil, while its **Out_B** bus contains an **OutputValue** signal that drives a reverse EDC coil. The following table lists function block faults and how they affect the two **OutputValue** signals.

Fault Logic

Fault	Cause	Bit*	Response	Delay†	Latch‡	Correction
Input value too low.	**CmdIn** < –100000.	1	**OutputValue** held at 0.	No.	No.	Fix **CmdIn.**
Input value too high.	**CmdIn** > +10000.	2				
Open circuit.	Measured Ω > (Nominal Ω × 3).§	3		Yes.	Yes.	Check for open circuit or high resistance between output pin and ground.
Short circuit.	Measured Ω < (Nominal Ω ÷ 3).§	4				Check for short circuit or low resistance between output pin and ground.
Hardware.	Current exceeds continuous or peak rating.	6				Correct cause of overload.
General.	Current flow between **Out_A** and **Out_B**. Possible cause is loss of a common ground connection.	7				Correct open ground connection.

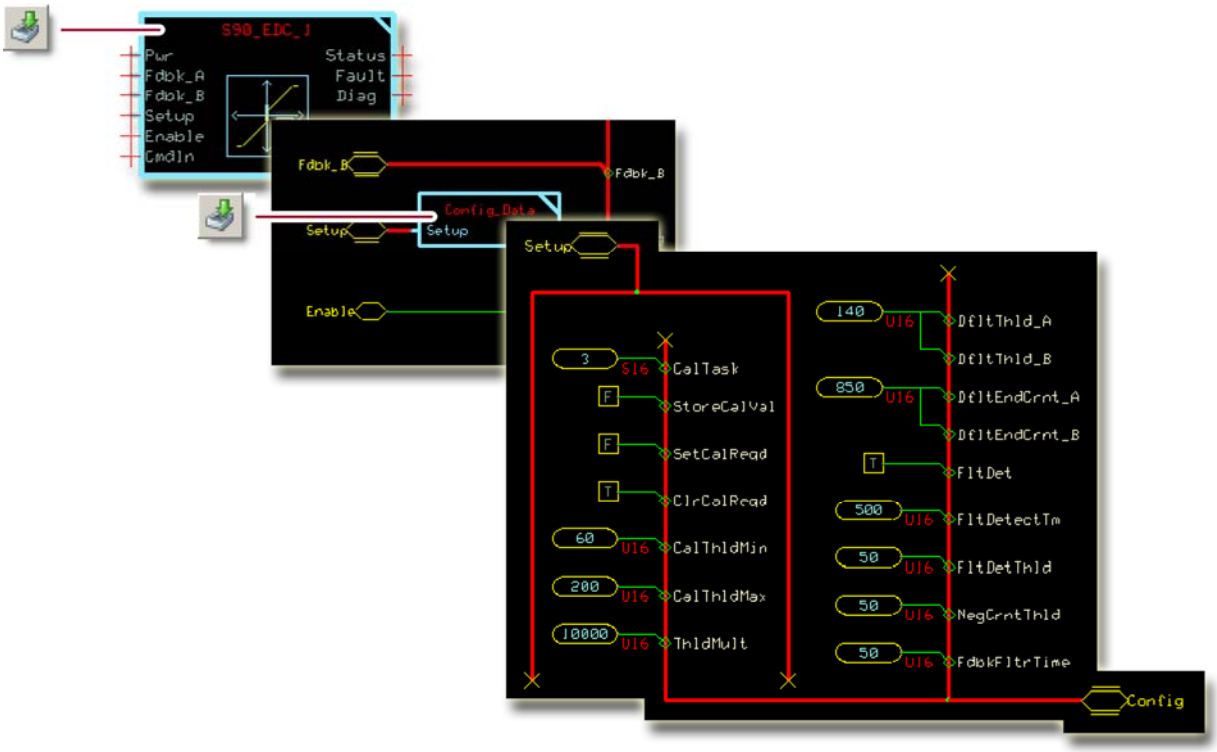
*Position of set bit in a 16 bit status or fault code. Bit 1 is the least significant bit. Bit 16 set to 1 identifies a standard Sauer-Danfoss status code or fault code.

†A delayed fault gets reported if the detected fault condition persists for a specified delay time. A delayed fault cannot be cleared until the fault condition remains undetected for the delay time.

‡A latched fault report maintains until the latch is released. A release is attempted each time that **Enable** becomes **T** while **CmdIn** is 0.

§The nominal Ω value is an internal constant of each function block. It can be viewed, with other defined constants, in the **Define** sub-bus of the **Diag** bus. The other constants specify the maximum current and the range limits for configuration values.

Configuration Values



The default **Config_Data** page contains constant values that set the function block's operating characteristics. Typically, you do not have to change any of these values for the correct operation of the function block.

Optionally, you can:

- Modify the constant values on this page.
- Replace some or all the constant values on this page with signals brought in on the **Setup** bus, which is connected to the function block's **Setup** input. See *About Modifying the Config_Data Page* on page *15* for more information.

Config_Data *Function Block Configuration Values*

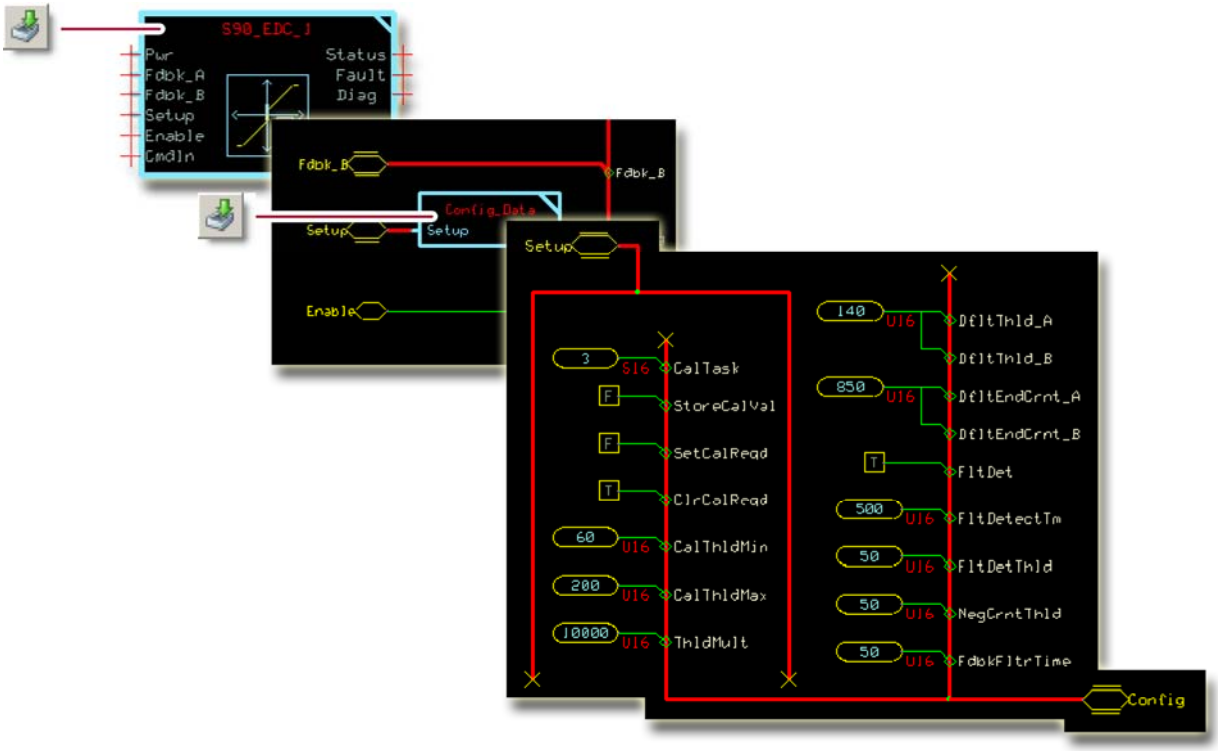
Input	Type	Range	Description
CalTask	S16	–2 to +3	Selects a calibration process. – –2 = Calibrate the B end current. – –1 = Calibrate the B threshold. – 0 = Not in the calibration mode. – +1 = Calibrate the A threshold. – +2 = Calibrate the A end current. – +3 = Applies default calibration values for both outputs on a transition to +3. The default value of +3 applies default values each time the controller powers up and allows normal operation.
StoreCalVal	BOOL	——	While calibration is active, a **StoreCalVal** (Store Calibration Value) F/T transition writes the selected calibration value to memory and sets the **CalFlg** to indicate this value as calibrated. The **CalFlg** in the **Diag** bus reports the status of each calibration value. – T = Not calibrated. – F = Calibrated.
SetCalReqd	BOOL	——	A **SetCalReqd** (Set Calibration Required) F to T transition sets all **CalFlg** bits to 1 to mark all calibration values as uncalibrated.
ClrCalReqd	BOOL	——	A **ClrCalReqd** (Clear Calibration Required) F to T transition clears all **CalFlg** bits to 0 to mark all calibration values as calibrated.
CalThldMin	U16	0–89	Sets the lower limit of the valid range for the threshold parameter. 10000 = 1000 mA.
CalThldMax	U16	90–300	Sets the upper limit for of the valid range for the threshold parameter. 10000 = 1000 mA.
ThldMult	U16	0–10000	The function block multiplies the **Out_A** and **Out_B** thresholds by the **ThldMult** (Threshold Multiplier) to calculate the applied threshold. Enter a value of less than 10000 to reduce the applied threshold. For example, a value of 7500 reduces both the **Out_A** and **Out_B** thresholds by 25%. 10000 = 100.00%.
DfltThld_A	U16	**CalThldMin** to **CalThldMax**	When the **CalTask** becomes +3, the **Out_A** threshold parameter resets to equal the **DfltThld_A** (Default Threshold A). 10000 = 1000 mA.
DfltThld_B	U16	**CalThldMin** to **CalThldMax**	When the **CalTask** becomes +3, the **Out_B** threshold parameter resets to equal the **DfltThld_B** (Default Threshold B). 10000 = 1000 mA.
DfltEndCrnt_A	U16	0–100	When the **CalTask** becomes +3, the **Out_A** end current parameter resets to equal the **DfltEndCrnt_A** (Default End Current A). 10000 = 1000 mA.
DfltEndCrnt_B	U16	0–100	When the **CalTask** becomes +3, the **Out_B** end current parameter resets to equal the **DfltEndCrnt_B** (Default End Current B). 10000 = 1000 mA.

Config_Data *Function Block Configuration Values*

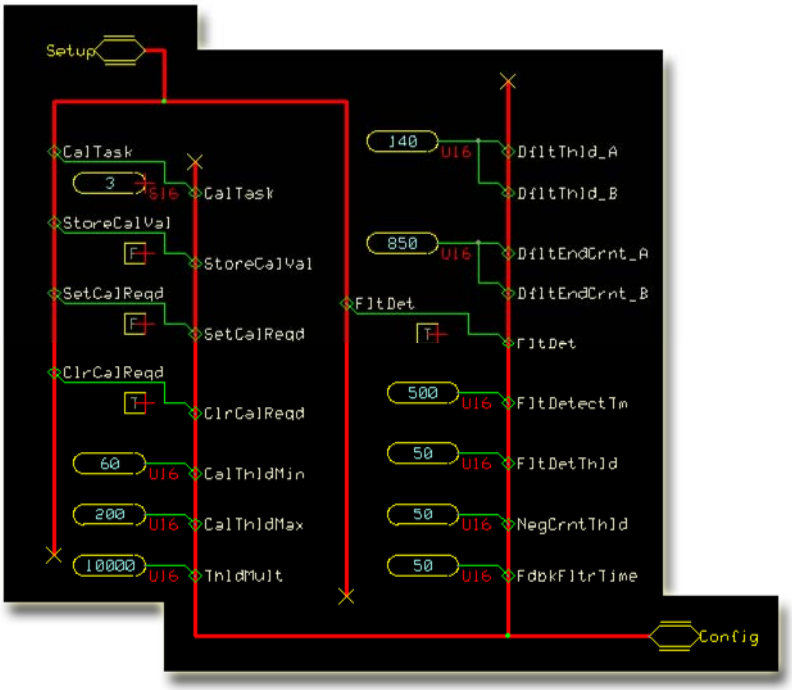
Input	Type	Range	Description
FltDet	BOOL	——	The **FltDet** (Fault Detection) signal enables the detection of open and short conditions, based on the measured resistance of the EDC control circuit. – T = Enable fault detection. – F = Disable fault detection.
FltDetectTm	U16	100–2000	Sets the time before the fault detection logic reports or clears fault conditions. This value specifies how long a fault condition must be detected before it is reported. It also specifies how long the fault condition must remain undetected before the report can be cleared. 1000 = 1000 ms.
FltDetThld	U16	——	For each direction, the open and short faults are only detected while the **OutputValue** exceeds the **FltDetThld** (Fault Detect Threshold). Typically, set this value below the **Out_A** and **Out_B** threshold values. If you set the value too: – Low, you get nuisance faults. – High, you turn off fault detection for some or all of the output range.
NegCrntThld	U16	——	Negative feedback current in an uncommanded direction must be greater than the **NegCrntThld** (Negative Current Threshold) value to set a fault. Typically, this fault results when the A and B outputs drive coils that share a common ground and their connection to the controller ground is lost.
FdbkFltrTime	U16	——	Sets the time constant for the exponential filtering applied to the current measurement used to detect a fault caused by negative feedback in an uncommanded direction.

About Modifying the Config_Data Page

Modify the **Config_Data** page to control the configuration process with signals routed into the function block from an application.



The preceding figure shows an unmodified **Config_Data** page. This **Config_Data** page has all the values that are needed to configure the function block.

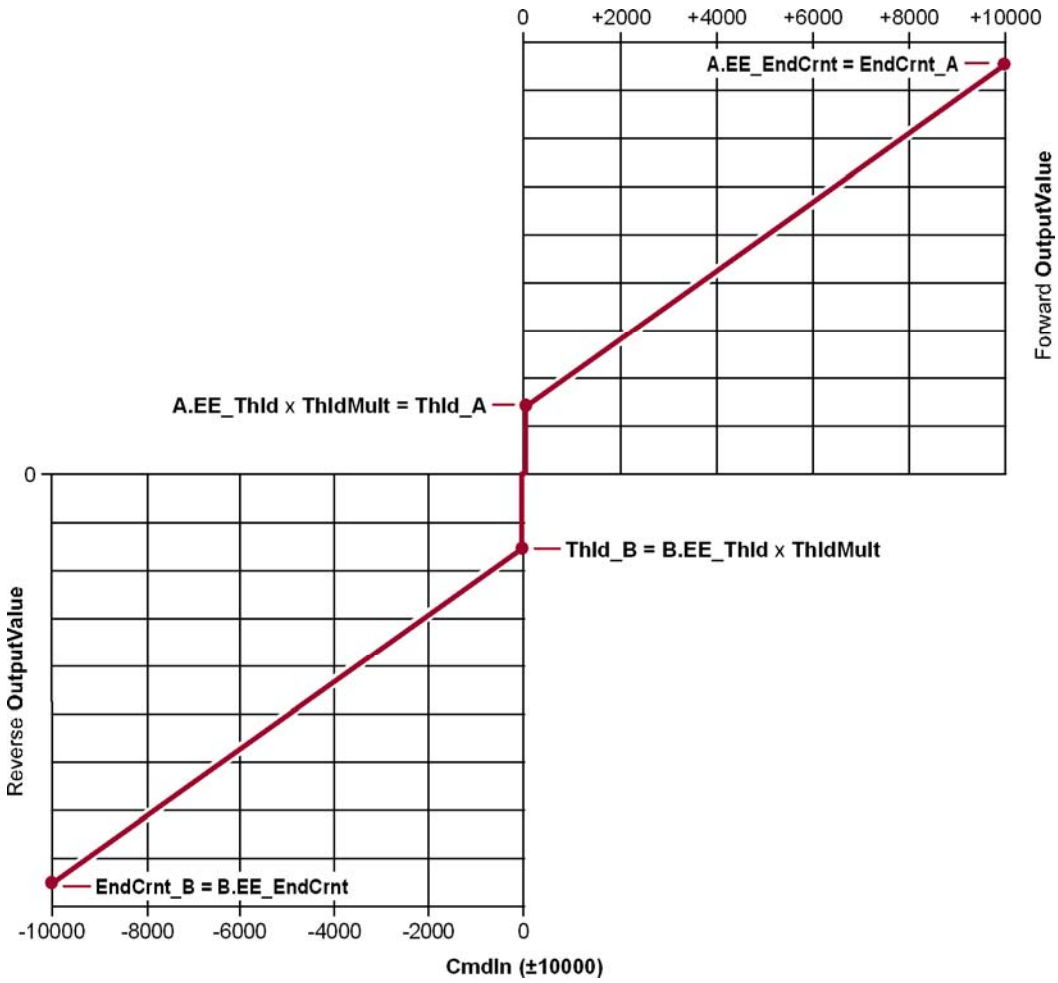


The preceding figure shows the changes made to a **Config_Data** page to allow an application to control the configuration process using signals routed via the **Setup** bus.

About the Relationship between the Function Block Input and Output Signals

The function block's **Out_A** and **Out_B** buses each contain an **OutputValue** signal.

The following figure plots the relationship between the function block's **CmdIn** and **OutputValue** signals.



- The **Thld_A** value sets the **Out_A OutputValue** of the block when it receives a **CmdIn** of **+1**.
- The **Thld_B** value sets the **Out_B OutputValue** of the block when it receives a **CmdIn** of **–1**.
- The **EndCrnt_A** value sets the **Out_A OutputValue** of the block when it receives a **CmdIn** of **+10000**.
- The **EndCrnt_B** value sets the **Out_B OutputValue** of the block when it receives a **CmdIn** of **–10000**.

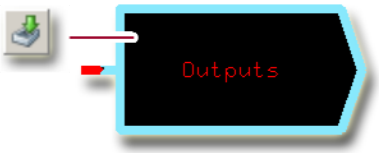
MC Controller—Output Configuration

If you have an SC controller, see *SC Controller—Output Configuration* on page *20*.

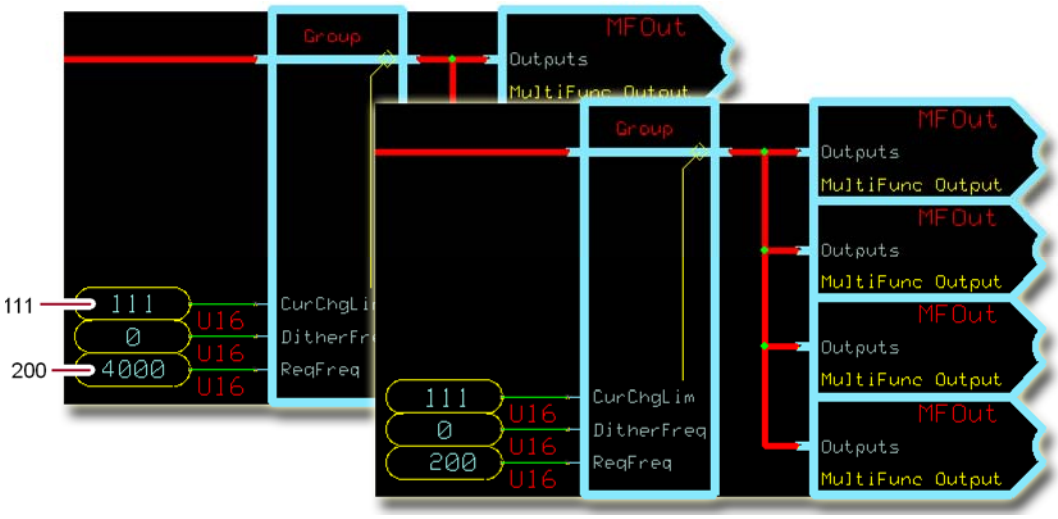
You route the function block's **Output** bus to an **MFOut**.

How to Configure an MFOut

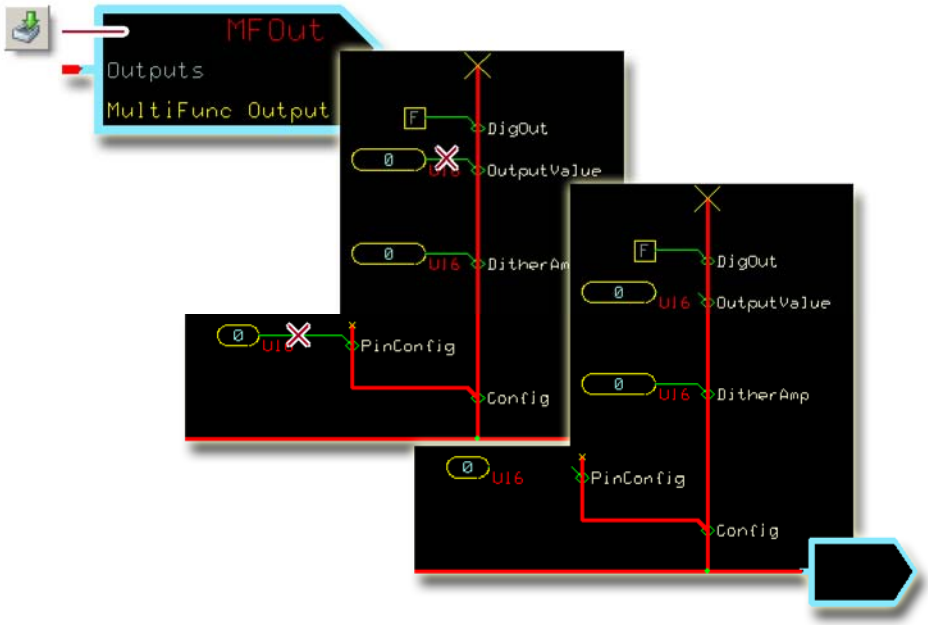
You must configure an **MFOut** to receive the signals in an **Output** bus.



1. In the GUIDE template, enter the **Outputs** block.



2. In the **Group** that receives the signals in an **Output** bus, make the changes that are shown in the preceding figure.



3. In the individual **MFOut** that receives the signals in an **Output** bus, make the changes that are shown in the preceding figure.

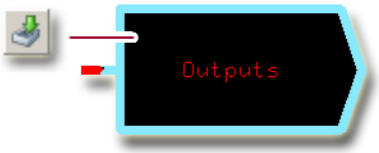
SC Controller—Output Configuration

If you have an MC controller, see *MC Controller—Output Configuration* on page *18*.

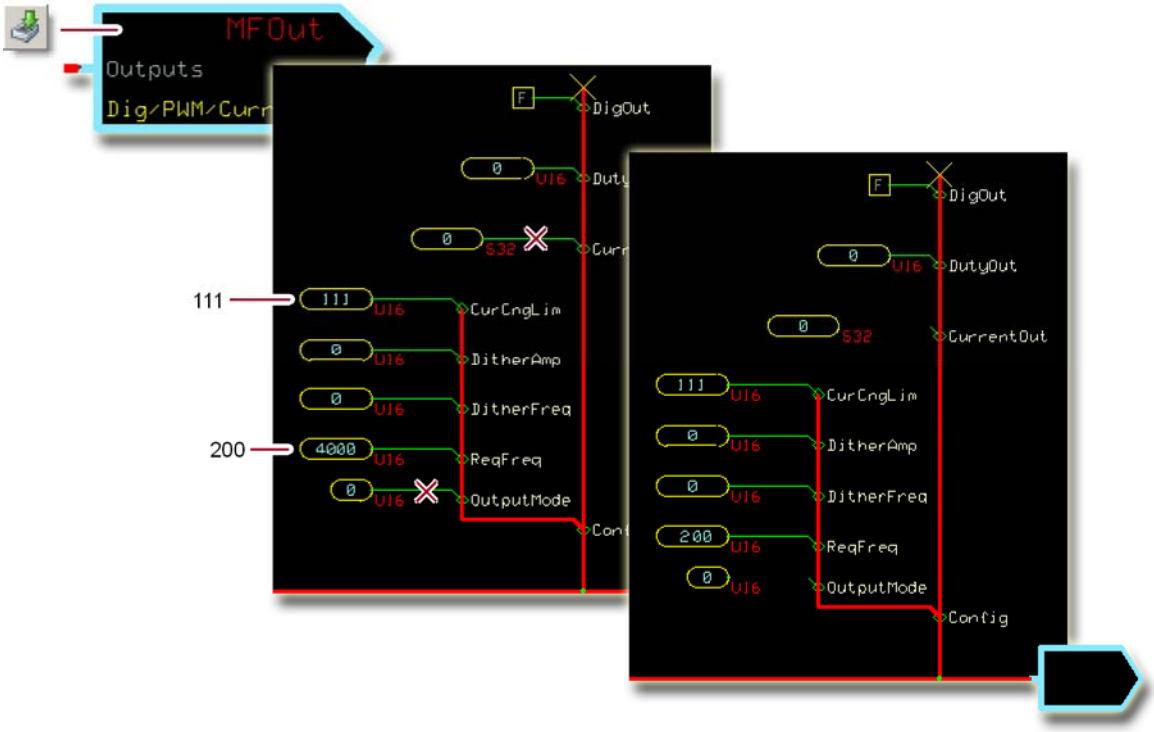
You route the function block's **Output** bus to an **MFOut**.

How to Configure an MFOut

You must configure the **MFOut** to receive the signals in an **Output** bus.



1. In the GUIDE template, enter the **Outputs** block.



2. In the **MFOut** that receives the signals in an **Output** bus, make the changes that are shown in the preceding figure.

How to Calibrate the Function Block

See *About Modifying the Config_Data Page* on page *15* for an example of a **Config_Data** page that has been modified to allow an application program to control the calibration process.

1. Prepare to calibrate.
   A. Set the **SetCalReq** signal to F.
   B. Set the **ClrCalReq** signal to F.
   C. Set the **StoreCalVal** signal to F.
   D. Toggle the **SetCalReq** signal from F to T.

      Toggling sets the "not calibrated" bits in the **CalFlag** signal to 1.
2. Set the **Enable** signal to T.
3. Calibrate the **Out_A** threshold parameter.
   A. Set the **CalTask** signal to +1.
   B. Gradually modify the **CmdIn** signal in a positive (0 to +10000) direction to find the minimum command that causes motion.

      In the **Status** signal, monitor bit 7—**Value too low** and bit 8—**Value too high** to make sure that the **OutputValue** signal is within the valid threshold range.
   C. Toggle the **StoreCalVal** signal from F to T to write the **OutputValue** to memory.

      In the **CalFlag** signal, check that bit 1 clears to 0, to verify that the controller has written the **Out_A** threshold parameter to memory.
4. Calibrate the reverse threshold parameter.
   A. Set the **CalTask** signal to –1.
   B. Gradually modify the **CmdIn** signal in a negative (0 to –10000) direction to find the minimum command that causes motion.

      In the **Status** signal, monitor bit 7—**Value too low** and bit 8—**Value too high** to make sure that the **OutputValue** signal is within the valid threshold range.
   C. Toggle the **StoreCalVal** signal from F to T to write the **OutputValue** to memory.

      In the **CalFlag** signal, check that bit 3 clears to 0, to verify that the controller has written the **Out_B** threshold parameter to memory.

5. Calibrate the **Out_A** end current parameter.
   A. Set the **CalTask** signal to +2.
   B. Gradually modify the **CmdIn** signal in a positive direction to find the command that produces the desired maximum pump flow.
   C. Toggle the **StoreCalVal** signal from F to T to write the **OutputValue** to memory.

      In the **CalFlag** signal, check that bit 2 clears to 0, to verify that the controller has written the **Out_A** end current parameter to memory.
6. Calibrate the **Out_B** end current parameter.
   A. Set the **CalTask** signal to –2.
   B. Gradually modify the **CmdIn** signal in a negative direction to find the command that produces the desired maximum pump flow.
   C. Toggle the **StoreCalVal** signal from F to T to write the **OutputValue** to memory.

      In the **CalFlag** signal, check that bit 4 clears to 0, to verify that the controller has written the **Out_B** end current parameter to memory.
7. End the calibration process.
   A. Set the **CalTask** signal to 0.
   B. In the **CalFlag** signal, verify that all bits are now 0.
   C. Verify that no **Status** or **Fault** conditions are reported.
   D. Set the **ThldMult** signal to the desired value.

How to Calibrate an Individual Parameter

Partial calibration allows you to calibrate an individual parameter while leaving the values of other parameters unchanged.

1. Prepare to calibrate.
   A. Set the **SetCalReq** signal to F.
   B. Set the **ClrCalReq** signal to F.
   C. Set the **StoreCalVal** signal to F.
   D. Toggle the **SetCalReq** signal from F to T.

      Toggling sets all the “not calibrated” bits in the **CalFlag** signal to 1.
2. Set the **Enable** signal to T.
3. Use the **CalTask** signal to select the parameter to be calibrated.
4. Calibrate the parameter.
   A. Toggle the **StoreCalVal** signal from F to T to write the **OutputValue** to memory.
   B. In the **CalFlag** signal, check that the calibration bit for the selected parameter clears to 0, to verify that the controller has written the parameter to memory.
5. End the calibration process.
   A. Set the **CalTask** signal to 0.
   B. Toggle the **ClrCalReqd** signal from F to T.
   C. In the **CalFlag** signal, verify that all bits are now 0.
   D. Verify that no **Status** or **Fault** conditions are reported.

About Manual Calibration

Warning

Using the PLUS+1 Service Tool program download new parameter values to an application can result in unexpected and sudden machine movements.

Unexpected and sudden machine movements can result in personal injury and equipment damage.

Always secure your machine against unexpected and sudden movements before you use the Service Tool program download new parameter values.

You can manually calibrate the function block using the PLUS+1 Service Tool program to download calibration parameters.

When you manually calibrate, make sure that the:

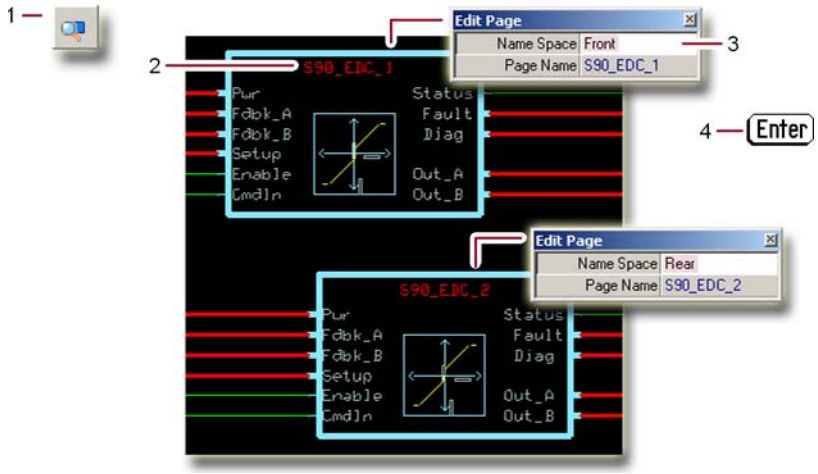
- Calibration parameters are valid. If the values are out of range for a given direction, the block limits the values of **ThldApplied**, **EndCrntApplied** or both for that direction. This condition is indicated by the “Parameters are corrupt” status and can produce unintended output commands.
- **Status** signal indicates normal status conditions.

About the Name Space Feature

If you use this function block more than once in an application, you must change each function block's **Name Space** value to avoid compiler errors.

These function blocks allocate memory using memory names ("aliases"). Identical function blocks have identical memory names. Identical memory names will cause a compiler error.

The **Name Space** value adds a unique prefix to each memory name to avoid memory allocation errors. Keep **Name Space** values short to save controller memory.



How to Enter a Name Space Value

1. In the PLUS+1 GUIDE menu bar, click the Query/Change button.
2. Click the function block's page name to display the **Edit Page** window.
3. In the **Edit Page** window, enter a meaningful **Name Space** value.
4. Press Enter↵.
5. Repeat these steps to enter unique **Name Space** values for other identical function blocks.

Products we offer:

- Bent Axis Motors
- Closed Circuit Axial Piston Pumps and Motors
- Displays
- Electrohydraulic Power Steering
- Electrohydraulics
- Hydraulic Power Steering
- Integrated Systems
- Joysticks and Control Handles
- Microcontrollers and Software
- Open Circuit Axial Piston Pumps
- Orbital Motors
- PLUS+1™ GUIDE
- Proportional Valves
- Sensors
- Steering
- Transit Mixer Drives

Members of the Sauer-Danfoss Group:

Comatrol
www.comatrol.com

Schwarzmüller-Inverter
www.schwarzmueller-inverter.com

Turolla
www.turollaocg.com

Hydro-Gear
www.hydro-gear.com

Sauer-Danfoss-Daikin
www.sauer-danfoss-daikin.com

11022912 • Rev CA • October 2011

Sauer-Danfoss is a global manufacturer and supplier of high-quality hydraulic and electronic components. We specialize in providing state-of-the-art technology and solutions that excel in the harsh operating conditions of the mobile off-highway market. Building on our extensive applications expertise, we work closely with our customers to ensure exceptional performance for a broad range of off-highway vehicles.

We help OEMs around the world speed up system development, reduce costs and bring vehicles to market faster.
Sauer-Danfoss—Your Strongest Partner in Mobile Hydraulics.

Go to www.sauer-danfoss.com for further product information.

Wherever off-highway vehicles are at work, so is Sauer-Danfoss.

We offer expert worldwide support for our customers, ensuring the best possible solutions for outstanding performance. And with an extensive network of Global Service Partners, we also provide comprehensive global service for all of our components.

Please contact the Sauer-Danfoss representative nearest you.

Local address:
Sauer-Danfoss Inc.
3500 Annapolis Lane North
Minneapolis, MN 55447, USA
Phone: +1 763 509-2000
Fax: +1 763 559-5769

Sauer-Danfoss (US) Company
2800 East 13th Street
Ames, IA 50010, USA
Phone: +1 515 239-6000
Fax: +1 515 239-6618

Sauer-Danfoss ApS
DK-6430 Nordborg, Denmark
Phone: +45 7488 4444
Fax: +45 7488 4400

Sauer-Danfoss GmbH & Co. OHG
Postfach 2460, D-24531 Neumünster
Krokamp 35, D-24539 Neumünster, Germany
Phone: +49 4321 871-0
Fax: +49 4321 871 122

Sauer-Danfoss-Daikin LTD
Shin-Osaka TERASAKI 3rd Bldg. 6F
1-5-28 Nishimiyahara, Yodogawa-ku
Osaka 532-0004, Japan
Phone: +81 6 6395 6066
Fax: +81 6 6395 8585

www.sauer-danfoss.com