Chapter 13
IODDTs and Device DDTs for Analog Modules

Subject of this Chapter

This chapter presents the various language objects, IODDTs and Device DDTs associated with analog input/output modules.

In order to avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before to call any EF using this channel.

What Is in This Chapter?

This chapter contains the following topics:

Detailed Description of T_ANA_IN_BMX-type IODDT Objects

At a Glance

The following tables describe the T_ANA_IN_BMX-type IODDT objects applicable to **BME AHI 0812**, **BMX AMI 0410**, **BMX AMI 0800** and **BMX AMI 0810**, and to **the inputs of the BMX AMM 600 mixed module**.

Input Measurement

The analog input measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog input measurement.	%IWr.m.c.0

%Ir.m.c.ERR error bit

The %Ir.m.c.ERR error bit is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit for analog channel.	%Ir.m.c.ERR

MEASURE_STS Measurement Status Word

The meaning of the MEASURE_STS (%IWr.m.c.1) measurement status word bits is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ALIGNED	BOOL	R	Aligned channel.	%IWr.m.c.1.0
CH_FORCED	BOOL	R	Forced channel.	%IWr.m.c.1.1
LOWER_LIMIT	BOOL	R	Measurement within lower tolerance area.	%IWr.m.c.1.5
UPPER_LIMIT	BOOL	R	Measurement within upper tolerance area.	%IWr.m.c.1.6
INT_OFFSET_ERROR	BOOL	R	Internal offset detected error.	%IWr.m.c.1.8
INT_REF_ERROR	BOOL	R	Internal reference detected error.	%IWr.m.c.1.10
POWER_SUP_ERROR	BOOL	R	Power supply detected error.	%IWr.m.c.1.11
SPI_COM_ERROR	BOOL	R	SPI communication detected error.	%IWr.m.c.1.12

Explicit Exchange Execution Flag: EXCH_STS

The meaning of the exchange control bits of the channel EXCH_STS (%MWr.m.c.0) is as follows.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Read channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment parameter exchange in progress.	%MWr.m.c.0.2

Explicit Exchange Report: EXCH_RPT

The meaning of the EXCH_RPT (%MWr.m.c.1) report bits is as follows.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Read error detected for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error detected during command parameter exchange.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error detected while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error detected while reconfiguring the channel.	%MWr.m.c.1.15

Standard Channel Status: CH_FLT

The following table explains the meaning of the CH_FLT (%MWr.m.c.2) status word bits. Reading is performed by a READ_STS (IODDT_VAR1).

Standard symbol	Type	Access	Meaning	Address
SENSOR_FLT	BOOL	R	Sensor connection detected error.	%MWr.m.c.2.0
RANGE_FLT	BOOL	R	Range under/overflow detected error.	%MWr.m.c.2.1
CH_ERR_RPT	BOOL	R	Channel detected error report.	%MWr.m.c.2.2
INTERNAL_FLT	BOOL	R	Inoperative channel.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem detected communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error detected (adjustment or configuration error).	%MWr.m.c.2.7
NOT_READY	BOOL	R	Channel not ready.	%MWr.m.c.3.0
CALIB_FLT	BOOL	R	Calibration detected error.	%MWr.m.c.3.2
INT_OFFS_FLT	BOOL	R	Internal calibration offset detected error.	%MWr.m.c.3.3
INT_REF_FLT	BOOL	R	Internal calibration reference detected error.	%MWr.m.c.3.4
INT_SPI_PS_FLT	BOOL	R	Internal serial link or power supply detected error.	%MWr.m.c.3.5
RANGE_UNF	BOOL	R	Recalibrated channel or range underflow.	%MWr.m.c.3.6
RANGE_OVF	BOOL	R	Aligned channel or range overflow.	%MWr.m.c.3.7

Command Controls

The following table explains the meaning of the COMMAND_ORDER (%MWr.m.c.4) status word bit. Reading is performed by a READ_STS;

Standard symbol	Type	Access	Meaning	Address
FORCING_ORDER	BOOL	R/W	Forcing/unforcing command.	%MWr.m.c.4.13

Parameters

The table below presents the meaning of the %MWr.m.c.5, %MWr.m.c.8 and %MWr.m.c.9 words. Queries used are those associated with parameters (READ_PARAM, WRITE_PARAM):

Standard symbol	Type	Access	Meaning	Address
CMD_FORCING_VALUE	INT	R/W	Forcing value to be applied.	%MWr.m.c.5
FILTER_COEFF	INT	R/W	Value of filter coefficient.	%MWr.m.c.8
ALIGNMENT_OFFSET	INT	R/W	Alignment offset value. **NOTE:** Offset=Target value - Mesured value, for instance, if you want to see a value of 3000 when the measured value is 2400 you have to set an offset of 600.	%MWr.m.c.9

NOTE: In order to force a channel, you have to use the WRITE_CMD (%MWr.m.c.5) instruction and set the %MWr.m.c.4.13 bit to 1.

NOTE: To unforce a channel and use it normally, you have to set the %MWr.m.c.4.13 bit to 0.

Detailed Description of T_ANA_IN_T_BMX-type IODDT Objects

At a Glance

The following tables describe the T_ANA_IN_T_BMX-type IODDT objects applicable to **BMX ART 0414/0814** analog input modules.

Input Measurement

The analog input measurement object is as follows:

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog input measurement.	%IWr.m.c.0

%Ir.m.c.ERR error bit

The %Ir.m.c.ERR error bit is as follows:

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Error bit for analog channel.	%Ir.m.c.ERR

MEASURE_STS Measurement Status Word

The various meanings of the MEASURE_STS (%IWr.m.c.1) measurement status word bits are as follows:

Standard symbol	Type	Access	Meaning	Address
CH_ALIGNED	BOOL	R	Aligned channel.	%IWr.m.c.1.0
CH_FORCED	BOOL	R	Forced channel.	%IWr.m.c.1.1
LOWER_LIMIT	BOOL	R	Measurement within lower tolerance area.	%IWr.m.c.1.5
UPPER_LIMIT	BOOL	R	Measurement within upper tolerance area.	%IWr.m.c.1.6
INT_OFFSET_ERROR	BOOL	R	Internal offset error.	%IWr.m.c.1.8
INT_REF_ERROR	BOOL	R	Internal reference error.	%IWr.m.c.1.10
POWER_SUP_ERROR	BOOL	R	Power supply error.	%IWr.m.c.1.11
SPI_COM_ERROR	BOOL	R	SPI communication error.	%IWr.m.c.1.12

Cold Junction Compensation

The value of the cold junction compensation is as follows:

Standard symbol	Type	Access	Meaning	Address
CJC_VALUE	INT	R	Cold junction compensation value (1/10° C).	%IWr.m.c.2

Explicit Exchange Execution Flag: EXCH_STS

The meaning of the exchange control bits of the channel EXCH_STS (%MWr.m.c.0) is as follows:

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Read channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment parameter exchange in progress.	%MWr.m.c.0.2

Explicit Exchange Report: EXCH_RPT

The meaning of the EXCH_RPT (%MWr.m.c.1) report bits is as follows:

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Read error for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during command parameter exchange.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error while reconfiguring the channel.	%MWr.m.c.1.15

Standard Channel Status: CH_FLT

The following table explains the meaning of the CH_FLT (%MWr.m.c.2) status word bits. Reading is performed by a READ_STS (IODDT_VAR1).

Standard symbol	Type	Access	Meaning	Address
SENSOR_FLT	BOOL	R	Sensor connection error.	%MWr.m.c.2.0
RANGE_FLT	BOOL	R	Range under/overflow error.	%MWr.m.c.2.1
CH_ERR_RPT	BOOL	R	Channel error report.	%MWr.m.c.2.2
INTERNAL_FLT	BOOL	R	Inoperative channel.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application error (adjustment or configuration error).	%MWr.m.c.2.7
NOT_READY	BOOL	R	Channel not ready.	%MWr.m.c.3.0
COLD_JUNCTION_FLT	BOOL	R	Cold junction compensation error.	%MWr.m.c.3.1
CALIB_FLT	BOOL	R	Calibration error.	%MWr.m.c.3.2
INT_OFFS_FLT	BOOL	R	Internal calibration offset error.	%MWr.m.c.3.3
INT_REF_FLT	BOOL	R	Internal calibration reference error.	%MWr.m.c.3.4
INT_SPI_PS_FLT	BOOL	R	Internal serial link or power supply error.	%MWr.m.c.3.5
RANGE_UNF	BOOL	R	Range underflow.	%MWr.m.c.3.6
RANGE_OVF	BOOL	R	Range overflow.	%MWr.m.c.3.7

Command Controls

The following table explains the meaning of the COMMMAND_ORDER (%MWr.m.c.4) status word bit. Reading is performed by a READ_STS:

Standard symbol	Type	Access	Meaning	Address
FORCING__UNFORCING_ORDER	BOOL	R/W	Forcing/unforcing command.	%MWr.m.c.4.13

Parameters

The table below presents the meaning of the %MWr.m.c.5, %MWr.m.c.8 and %MWr.m.c.9 status words. Queries used are those associated with parameters (READ_PARAM, WRITE_PARAM).

Standard symbol	Type	Access	Meaning	Address
CMD_FORCING_VALUE	INT	R/W	Forcing value to be applied.	%MWr.m.c.5
FILTER_COEFF	INT	R/W	Value of filter coefficient.	%MWr.m.c.8
ALIGNMENT_OFFSET	INT	R/W	Alignment offset value. **NOTE:** Offset=Target value - Mesured value, for instance, if you want to see a value of 3000 when the measured value is 2400 you have to set an offset of 600.	%MWr.m.c.9

NOTE: In order to force a channel, you have to use the WRITE_CMD (%MWr.m.c.5) instruction and set the %MWr.m.c.4.13 bit to 1.

NOTE: To unforce a channel and use it normally, you have to set the %MWr.m.c.4.13 bit to 0.

Detailed Description of T_ANA_OUT_BMX-type IODDT Objects

At a Glance

The following tables describe the T_ANA_OUT_BMX-type IODDT objects applicable to the **BME AHO 0412**, **BMX AMO 0210**, **BMX AMO 0410** and **BMX AMO 0802** analog output modules and the outputs of the **BMX AMM 600** mixed module.

Value of the Output

The analog output measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog output measurement.	%QWr.m.c.0

%Ir.m.c.ERR error bit

The %Ir.m.c.ERR error bit is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Error bit for analog channel.	%Ir.m.c.ERR

Value Forcing

The value forcing bit is as follows.

Standard symbol	Type	Access	Meaning	Address
FORCING_VALUE	INT	R	Forcing of the value.	%IWr.m.c.0

Channel forcing indicator.

The meaning of the forcing control bits of the channel (%IWr.m.c.1) is as follows.

Standard symbol	Type	Access	Meaning	Address
CHANNEL_FORCED	BOOL	R	Forcing of the channel.	%MWr.m.c.1.1

Explicit Exchange Execution Flag: EXCH_STS

The meaning of the exchange control bits of the channel EXCH_STS (%MWr.m.c.0) is as follows:

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Read channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment parameter exchange in progress.	%MWr.m.c.0.2

Explicit Exchange Report: EXCH_RPT

The meaning of the EXCH_RPT (%MWr.m.c.1) report bits is as follows:

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Read error detected for channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error detected during command parameter exchange.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error detected while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error detected while reconfiguring the channel.	%MWr.m.c.1.15

Standard Channel Status: CH_FLT

The following table explains the meaning of the CH_FLT (%MWr.m.c.2) status word bits. Reading is performed by a READ_STS (IODDT_VAR1).

Standard symbol	Type	Access	Meaning	Address
ACT_WIRE_FLT	BOOL	R	Actuator wire open or short.	%MWr.m.c.2.0
RANGE_FLT	BOOL	R	Range under/overflow detected error.	%MWr.m.c.2.1
SHORT_CIRCUIT	BOOL	R	Short-circuit.	%MWr.m.c.2.2
CAL_PRM_FLT	BOOL	R	Calibration parameters not configured.	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Inoperative channel.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Different hardware and software configurations.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Problem detected communicating with the PLC.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application detected error (adjustment or configuration detected error).	%MWr.m.c.2.7
ALIGNED_CH	BOOL	R	Aligned channels.	%MWr.m.c.3.0
INT_CAL_FLT	BOOL	R	Calibration parameters not defined.	%MWr.m.c.3.2
INT_PS_FLT	BOOL	R	Internal power supply detected error.	%MWr.m.c.3.3
INT_SPI_FLT	BOOL	R	Serial link detected error.	%MWr.m.c.3.4
RANGE_UNF	BOOL	R	Range underflow.	%MWr.m.c.3.6
RANGE_OVF	BOOL	R	Range overflow.	%MWr.m.c.3.7

Command Control

The following table explains the meaning of the COMMAND_ORDER (%MWr.m.c.4) status word bit. Reading is performed by a READ_STS:

Standard symbol	Type	Access	Meaning	Address
FORCING_UNFORCING_ORDER	BOOL	R/W	Forcing/unforcing command.	%MWr.m.c.4.13

Parameters

The following table shows the meaning of the words %MWr.m.c.5 to %MWr.m.c.8. The requests used are those associated with the parameters (READ_PARAM and WRITE_PARAM).

Standard symbol	Type	Access	Meaning	Address
CMD_FORCING_VALUE	INT	R/W	Forcing value to be applied.	%MWr.m.c.5
FALLBACK	INT	R/W	Fallback value.	%MWr.m.c.7
ALIGNMENT	INT	R/W	Alignment value.	%MWr.m.c.8

NOTE: In order to force a channel, you have to use the WRITE_CMD (%MWr.m.c.5) instruction and set the %MWr.m.c.4.13 bit to 1.

NOTE: To unforce a channel and use it normally, you have to set the %MWr.m.c.4.13 bit to 0.

Detailed Description of T_ANA_IN_GEN-type IODDT Objects

At a Glance

The tables below present the T_ANA_IN_GEN-type IODDT objects that are applicable to the **BME AHI 0812**, **BMX AMI 0410**, **BMX AMI 0800** and **BMX AMI 0810** input modules, to the inputs of the **BMX AMM 600** mixed module and to the **BMX ART 0414/0814** analog input module.

Input Measurement

The analog input measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog input measurement.	%IWr.m.c.0

%Ir.m.c.ERR Error Bit

The %Ir.m.c.ERR error bit is as follows:

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit for analog channel.	%Ir.m.c.ERR

Detailed Description of T_ANA_OUT_GEN-type IODDT Objects

At a Glance

The following tables describe the `T_ANA_OUT_GEN`-type IODDT objects applicable to the **BME AHO 0412**, **BMX AMO 0210**, **BMX AMO 0410** and **BMX AMO 0802** analog output modules and to the output of the **BMX AMM 600** mixed module.

Input Measurement

The analog output measurement object is as follows.

Standard symbol	Type	Access	Meaning	Address
VALUE	INT	R	Analog output measurement.	%IWr.m.c.0

%Ir.m.c.ERR Error Bit

The `%Ir.m.c.ERR` error bit is as follows.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Detected error bit for analog channel.	%Ir.m.c.ERR

Details of the Language Objects of the IODDT of Type T_GEN_MOD

Introduction

The modules of Modicon M340 and X80 PLCs have an associated IODDT of type T_GEN_MOD.

Observations

In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.

Some bits are not used.

List of Objects

The table below presents the objects of the IODDT.

Standard Symbol	Type	Access	Meaning	Address
MOD_ERROR	BOOL	R	Module detected error bit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Module exchange control word	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Reading of status words of the module in progress	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Exchange report word	%MWr.m.MOD.1
STS_ERR	BOOL	R	Event when reading module status words	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal detected errors word of the module	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	module inoperable	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Inoperative channel(s)	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block incorrectly wired	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration anomaly	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or inoperative	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internal detected errors word of the module (Fipio extension only)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Internal detected error, module unserviceable (Fipio extension only)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Inoperative channel(s) (Fipio extension only)	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Terminal block incorrectly wired (Fipio extension only)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware or software configuration anomaly (Fipio extension only)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Module missing or inoperative (Fipio extension only)	%MWr.m.MOD.2.14

Analog Device DDT

Introduction

This topic describes the Unity Pro **Analog Device DDT**.

The default device DDT name contains the following information:
- module input and or output (***X*** symbol)
- module insertion number (**#** symbol).

Example: `MOD_ANA_`***x***`_#`

The default device DDT type contains the following information:
- platform with:
  - U for unified structure between Modicon M340 and X80 or Quantum
- device type (ANA for analog)
- function (STD for standard)
  - STD for standard
  - TEMP for temperature
- direction:
  - IN
  - OUT
- max channel (2, 4, 8)

Example: For a Modicon M340 with 4 standard inputs and 2 outputs the Device Derived Data Type is T_U_ANA_STD_IN_4_OUT_2

Adjustment Parameter limitation

In Quantum EIO and M580 RIO, adjustment parameters cannot be changed from the PLC application during operation (no support of READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM).

The concerned analog input parameters are:
- `FILTER_COEFF`
  Value of filter coefficient
- `ALIGNMENT_OFFSET`
  Alignment offset value
- `THRESHOLD0`
  Low threshold value
- `THRESHOLD1`
  High threshold value

The concerned analog output parameters are:
- `FALLBACK`
  Fallback value
- `ALIGNMENT`
  Alignment value

List of Device DDT

The following table shows the list of Modicon M340 and X80 devices and their corresponding device DDT name and type:

Device DDT Name	Device DDT Type	Modicon M340/X80 Devices
MOD_ANA_4_#	T_U_ANA_STD_IN_4	BMX AMI 0410
MOD_ANA_8_#	T_U_ANA_STD_IN_8	BME AHI 0812 BMX AMI 0800 BMX AMI 0810
MOD_ANA_2_#	T_U_ANA_STD_OUT_2	BMX AMO 0210
MOD_ANA_4_#	T_U_ANA_STD_OUT_4	BME AHO 0412 BMX AMO 0410
MOD_ANA_8_#	T_U_ANA_STD_OUT_8	BMX AMO 0802
MOD_ANA_6_#	T_U_ANA_STD_IN_4_OUT_2	BMX AMM 0600
MOD_ANA_4_#	T_U_ANA_TEMP_IN_4	BMX ART 0414
MOD_ANA_8_#	T_U_ANA_TEMP_IN_8	BMX ART 0814

Device DDT Instances Description

The following table shows the T_U_ANA_STD_IN_x and the T_U_ANA_STD_OUT_y status word bits:

Standard Symbol	Type	Meaning	Access
MOD_HEALTH	BOOL	0 = the module has a detected error	read
		1 = the module is operating correctly	
MOD_FLT	BYTE	internal detected errors byte of the module	read
ANA_CH_IN	ARRAY [0...x-1] of T_U_ANA_STD_CH_IN	array of structure	–
ANA_CH_OUT	ARRAY [0..y-1] of T_U_ANA_STD_CH_OUT	array of structure	–

The following table shows the T_U_ANA_STD_IN_x_OUT_y status word bits:

Standard Symbol	Type	Meaning	Access
MOD_HEALTH	BOOL	0 = the module has a detected error	read
		1 = the module is operating correctly	
MOD_FLT	BYTE	internal detected errors byte of the module	read
ANA_CH_IN	ARRAY [0..x-1] of T_U_ANA_STD_CH_IN	array of structure	–
ANA_CH_OUT	ARRAY [x..x+y-1] of T_U_ANA_STD_CH_OUT	array of structure	–

The following table shows the T_U_ANA_TEMP_IN_x status word bits:

Standard Symbol	Type	Meaning	Access
MOD_HEALTH	BOOL	0 = the module has a detected error	read
		1 = the module is operating correctly	
MOD_FLT	BYTE	internal detected errors byte of the module	read
ANA_CH_IN	ARRAY [[0..x-1] of T_U_ANA_TEMP_CH_IN	array of structure	–

The following table shows the T_U_ANA_STD_CH_IN[0..x-1] structure status word bits:

Standard Symbol		Type	Bit	Meaning	Access
FCT_TYPE		WORD	–	0 = channel is not used	read
				1 = channel is used	
CH_HEALTH		BOOL	–	0 = the channel has a detected error	read
				1 = the channel is operating correctly	
CH_WARNING		BOOL	–	not used	–
ANA		STRUCT	–	T_U_ANA_VALUE_IN	read
MEASURE_STS [INT]	CH_ALIGNED	BOOL	0	aligned channel	read
	LOWER_LIMIT	BOOL	5	measurement within lower tolerance area	read
	UPPER_LIMIT	BOOL	6	measurement within upper tolerance area	read
	INT_OFFSET_ERROR	BOOL	8	internal offset detected error	read
	IN_REF_ERROR	BOOL	10	internal reference detected error	read
	POWER_SUP_ERROR	BOOL	11	power supply detected error	read
	SPI_COM_ERROR	BOOL	12	SPI communication detected error	read

The following table shows the T_U_ANA_STD_CH_OUT[0..y-1] status word bits:

Standard Symbol	Type	Meaning	Access
FCT_TYPE	WORD	0 = channel is not used	read
		1 = channel is used	
CH_HEALTH	BOOL	0 = the channel has a detected error	read
		1 = the channel is operating correctly	
ANA	STRUCT	T_U_ANA_VALUE_OUT	read

The following table shows the T_U_ANA_VALUE_IN[0..x-1] and T_U_ANA_VALUE_OUT[0..y-1] structure status word bits:

Standard Symbol	Type	Bit	Meaning	Access
VALUE	INT	–	if FORCE_CMD = 1 then VALUE = FORCED_VALUE	read(1)
			if FORCE_CMD = 0 then VALUE = TRUE_VALUE	
FORCED_VALUE	INT	–	forced value of the channel	read / write
FORCE_CMD	BOOL	–	0 = Un-force command	read / write
			1 = force command	
FORCE_STATE	BOOL	–	0 = value is not forced	read
			1 = value is forced	
TRUE_VALUE(2)	INT	–	True value of the channel (from the sensor)	read

1 VALUE of the T_U_ANA_VALUE_OUT structure word can be accessed in read / write
2 TRUE_VALUE of the T_U_ANA_VALUE_OUT is the value calculated from the application.

The following table shows the T_U_ANA_TEMP_CH_IN[0..x-1] structure status word bits:

Standard Symbol	Type	Bit	Meaning	Access
FCT_TYPE	WORD	–	0 = channel is not used	read
			1 = channel is used	
CH_HEALTH	BOOL	–	0 = the channel has a detected error	read
			1 = the channel is operating correctly	
CH_WARNING	BOOL	–	not used	–
ANA	STRUCT	–	T_U_ANA_VALUE_IN	read
MEASURE_STS	INT	–	measurement status	read
CJC_VALUE	INT	–	Cold junction compensation value (1/10 °C)	read

Use and Description of DDT for Explicit Exchange

The following table shows the DDT type used for the variables connected to dedicated EFB parameter to perform an explicit exchange:

DDT	Description	
T_M_ANA_STD_CH_STS	Structure to read the channel status of an analog module.	Depending on the I/O module location, the DDT can be connected to the STS output parameter of the EFB: ● READ_STS_QX *(see Unity Pro, I/O Management, Block Library)* when the module is located in Quantum EIO. ● READ_STS_MX *(see Unity Pro, I/O Management, Block Library)* when the module is located in a M580 local rack or in M580 RIO drops.
T_M_ANA_STD_CH_IN_STS	Structure to read the channel status of an analog output module.	
T_M_ANA_STD_CH_OUT_STS	Structure to read the channel status of an analog output module.	
T_M_ANA_TEMP_CH_STS	Structure to read the channel status of an analog temperature input module.	
T_M_ANA_STD_CH_IN_PRM	Structure for adjustment parameters of a channel of an analog input module in a M580 local rack.	The DDT can be connected to the PARAM output parameter of the EFB: ● READ_PARAM_MX *(see Unity Pro, I/O Management, Block Library)* to read module parameters. ● WRITE_PARAM_MX *(see Unity Pro, I/O Management, Block Library)* to write module parameters. ● SAVE_PARAM_MX *(see Unity Pro, I/O Management, Block Library)* to save module parameters. ● RESTORE_PARAM_MX *(see Unity Pro, I/O Management, Block Library)* to restore the new parameters of the module.
T_M_ANA_STD_CH_OUT_PRM	Structure for adjustment parameters of a channel of an analog output module in a M580 local rack.	
NOTE: Targeted channel address (ADDR) can be managed with ADDMX *(see Unity Pro, Communication, Block Library)* EF (connect the output parameter OUT to the input parameter ADDR of the communication functions).		

The following table shows the DDT structure for T_M_ANA_STD_CH_STS, T_M_ANA_STD_CH_IN_STS, T_M_ANA_STD_CH_OUT_STS and T_M_ANA_TEMP_CH_STS:

Standard Symbol		Type	Bit	Meaning	Access
CH_FLT [INT]	SENSOR_FLT	BOOL	0	detected sensor faults	read
	RANGE_FLT	BOOL	1	detected range fault	read
	CH_ERR_RPT	BOOL	2	channel detected error report	read
	INTERNAL_FLT	BOOL	4	internal detected error: module out of order	read
	CONF_FLT	BOOL	5	detected configuration fault: different hardware and software configurations	read
	COM_FLT	BOOL	6	problem communicating with the PLC	read
	APPLI_FLT	BOOL	7	detected application fault	read
	COM_FLT_ON_EVT(1)	BOOL	8	detected communication fault on event	read
	OVR_ON_CH_EVT(1)	BOOL	9	detected overrun fault on CPU event	read
	OVR_ON_CH_EVT(1)	BOOL	10	detected overrun fault on channel event	read
CH_FLT_2 [INT]	NOT_READY	BOOL	0	Channel not ready	read
	COLD_JUNCTION_FLT(2)	BOOL	1	Cold junction compensation detected error	read
	CALIB_FLT	BOOL	2	detected calibration fault	read
	INT_OFFS_FLT	BOOL	3	detected internal offset error	read
	IN_REF_FLT	BOOL	4	detected internal reference fault	read
	INT_SPI_PS_FLT	BOOL	5	detected internal serial link or power supply error	read
	RANGE_UNF	BOOL	6	recalibrated channel or range underflow	read
	RANGE_OVF	BOOL	7	aligned channel or range overflow	read

(1) only available with T_M_ANA_STD_CH_IN_STS and T_M_ANA_STD_CH_OUT_STS
(2) only available with T_M_ANA_TEMP_CH_STS

The following table shows the T_M_ANA_STD_CH_IN_PRM DDT structure:

Standard Symbol	Type	Bit	Meaning	Access
FILTERCOEFF	INT	–	Value of filter coefficient	read/write
ALIGNMENT_OFFSET	INT	–	Alignment offset value	read/write
THRESHOLD0	INT	–	Threshold 0 value **NOTE:** Fast input only	read/write
THRESHOLD1	INT	–	Threshold 1 value **NOTE:** Fast input only	read/write

The following table shows the T_M_ANA_STD_CH_OUT_PRM DDT structure:

Standard Symbol	Type	Bit	Meaning	Access
FALLBACK	INT	–	fallback value	read/write
ALIGNMENT	INT	–	alignment value	read/write

Analog Device Ethernet Remote I/O Forcing Mode

Introduction

Input and output values of Modicon M340 and X80 analog modules can be forced through the device DDT value.

NOTE: Modicon M340 and X80 discrete modules values are forced using the `EBOOL` mechanism, refer to chapter ***Force Mode*** *(see Unity Pro, Operating Modes)*.

Forcing input and output values in a running controller can have serious consequences to the operation of a machine or process. Only those who understand the implications in the controlling logic, and who understand the consequences of forced I/O on the machine or process, should attempt to use this function.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

You must have prior knowledge of the process, the controlled equipment and the modified behavior in Unity Pro before attempting to force analog inputs or outputs.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Modicon M340 and X80 Analog Device T_U_ANA_VALUE_•• Structure

The following table shows the content of analog devices DDT *(see page 248)* type used to force a value:

Standard Symbol	Type	Meaning
VALUE	INT	Channel value. It represents the value used in the application and is either the FORCED_VALUE or the TRUE_VALUE depending on the FORCED_STATE.
FORCED_VALUE	INT	Value applied to an output or interpreted as an input during forcing. If FORCED_STATE = 1 then VALUE = FORCED_VALUE
FORCE_CMD	BOOL	Parameter used to force or unforce an analog output or input value
FORCED_STATE	BOOL	Forcing status: • 0: value is not forced • 1: value is forced
TRUE_VALUE	INT	Represents the true value of the analog output or input whatever the state of the forcing command

Forcing a Value with the Animation Tables

To force a DDT value in an animation table proceed as follows:

Step	Action
1	Select the chosen analog channel.
2	Set the FORCED_VALUE parameter value of the selected channel to the chosen value, for details on how to set a value, refer to chapter ***Modification Mode*** *(see Unity Pro, Operating Modes)*.
3	Set the FORCE_CMD parameter to 1.
4	**Result**: ● Check that forcing is applied: FORCED_STATE needs to be equal to 1 ● VALUE = FORCED_VALUE

Unforcing a Value with the Animation Tables

To unforce a DDT value in an animation table proceed as follows:

Step	Action
1	Select the chosen analog channel.
2	Set the FORCE_CMD parameter to 0.
3	**Result**: ● Check that forcing is released: FORCED_STATE needs to be equal to 0 ● VALUE = TRUE_VALUE

Chapter 14
Analog Module Debugging

Subject of this Chapter

This chapter describes the debugging aspect of the analog modules.

What Is in This Chapter?

This chapter contains the following topics:

Introducing the Debug Function of an Analog Module

Introduction

This function is only accessible in online mode. For each input/output module of the project, it can be used to:

- display measurements
- display the parameters of each channel (channel state, filtering value, etc.)
- access the diagnostics and adjustment of the selected channel (masking the channel, etc.)

The function also gives access to the module diagnostics in the case of an event.

Procedure

The procedure to access the **Debugging** function is as follows.

Step	Action
1	configure the module
2	transfer the application to the PLC
3	change to online mode
4	in the rack configuration screen, double-click on the module
5	select the **Debugging** tab

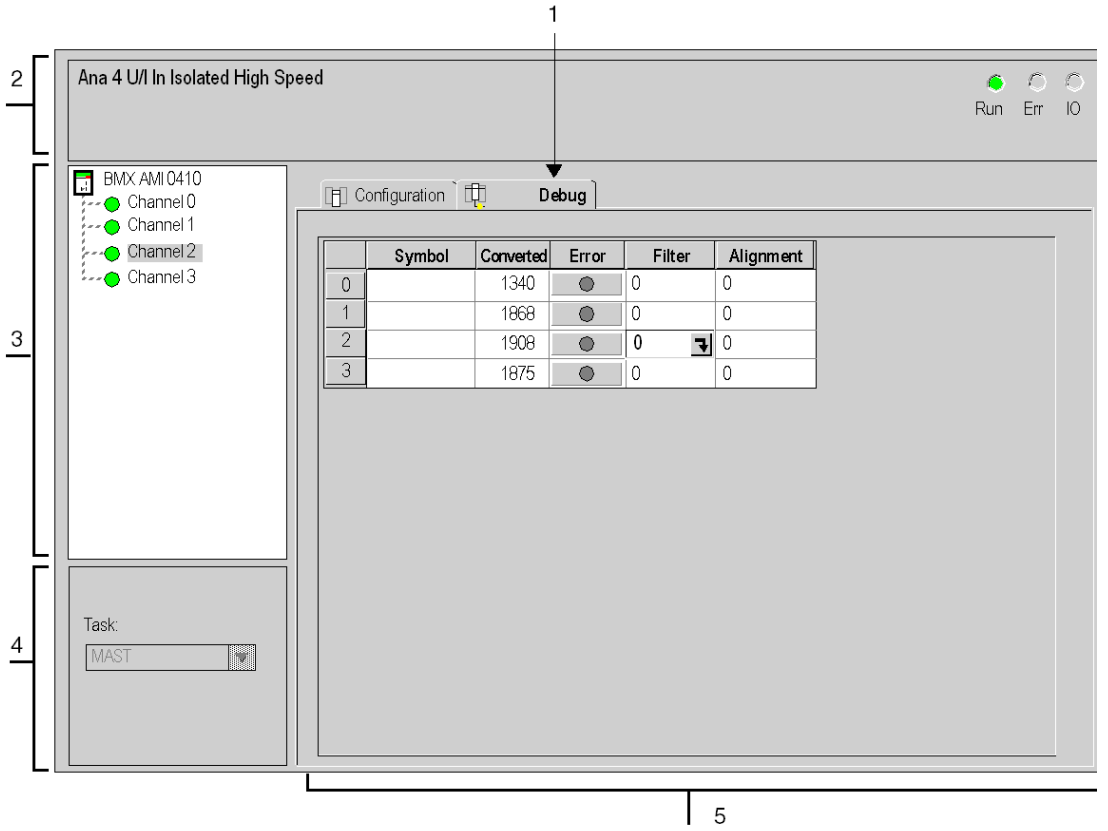
Description of the Analog Module Debug Screen

At a Glance

The Debug Screen displays, in real time, the current value and status for each of the selected module's channels.

Illustration

The figure below shows a sample debugging screen.



Description

The table below shows the different elements of the debug screen and their functions.

Address	Element	Function
1	Tabs	The tab in the foreground indicates the mode in progress (**Debug** in this example). Each mode can be selected by the corresponding tab. The available modes are: ● **Debug** which can be accessed only in online mode. ● **Configuration**.
2	**Module** area	Specifies the shortened name of the module. In the same area there are 3 LEDs which indicate the status of the module in online mode: ● **RUN** indicates the operating status of the module, ● **ERR** indicates an internal detected error in the module, ● **I/O** indicates an event from outside the module or an application error.
3	**Channel** area	Is used: ● To select a channel. ● To display the **Symbol**, name of the channel defined by the user (using the variable editor).
4	**General parameters** area	Specifies the **MAST** or **FAST** task configured. This information cannot be modified.
5	**Viewing and control** area	Displays the value and status for each channel in the module in real-time. The symbol column displays the symbol associated with the channel when the user has defined this (from the variable editor). This area provides direct access to channel by channel diagnostics when these are inoperative (indicated by error column LED ,which turns red). ● Access to the settings of the filtering, alignment and fallback values of the outputs, ● To channel-by-channel diagnostics when channels have an error (indicated by the LED built into the diagnostics access button, which turns red).

NOTE: LEDs and commands not available appear grayed out.

Selecting the Adjustment Values for the Input Channels and Measurement Forcing

At a Glance

This function is used to modify the filter, alignment and forcing value of one or more channels of an analog module.

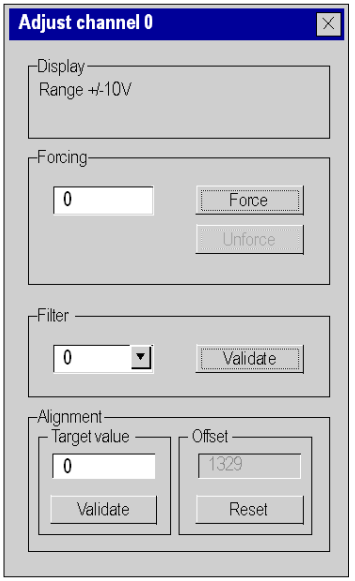
The available commands are:

- forcing
- filter
- alignment

To align several analog channels on the BMX AMO/AMI/AMM/ART modules, we recommand proceeding channel by channel. Test each channel after alignment before moving to the next channel, in order toapply the parameters correctly.

Procedure

The table below summarizes the procedure for modifying the filter, forcing and alignment values.

Step	Action for a channel
1	Access the debug screen.
2	Select the channel to be modified in the **Display** zone and double-click in the corresponding box. **Result**: The **Adjust channel** dialog box appears. 
3	Click on the text field in the **Forcing** field. Enter the forcing value. Send the forcing order by clicking on the **Forcing** button.
4	Click on the drop-down menu in the **Filter** field, and define the new selected filter value. Confirm this selection by clicking **OK**.
5	In the **Alignment** field click on the text field and define the target value. Confirm this selection by clicking **OK**.
6	Close the **Adjust channel** dialog box. **Results**: The new filter, forcing or alignment value then appears in the box corresponding to the selected channel in the **Filter**, **Forcing** or **Alignment** column of the **Display** area.

Modification of Output Channels Adjustment Values

At a Glance

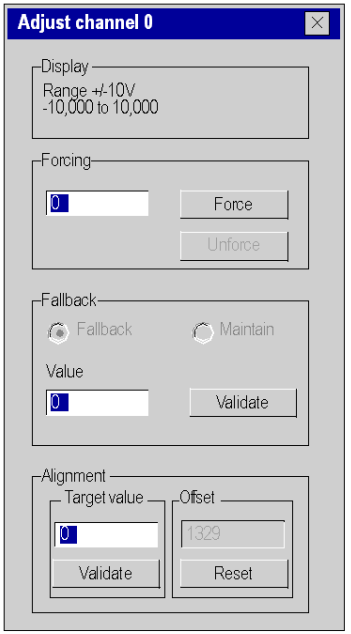
This function is used to modify the forcing, fallback and alignment values for one or several output channels of an analog module.

The available commands are:

- forcing
- fallback
- alignment

Procedure

The table below summarizes the procedure for modifying the values to be applied at the output channels:

Step	Action for a channel
1	Access the debug screen.
2	Select the channel in the **Display** zone and double-click in the corresponding box. **Result**: The **Adjust channel** dialog box appears. 

Step	Action for a channel
3	Click on the text field in the **Forcing** field of the **Adjust channel** dialog box. Enter the forcing value. Send the forcing order by clicking on the **Forcing** button.
4	Click on the box in the **Value** field of the **Fallback** dialog box and enter the new fallback value. Confirm this new value by clicking **OK**.
5	Click on the text field in the **Alignment** field of the **Adjust channel** dialog box and define the target value. Confirm this selection by clicking **OK**.
6	Close the **Adjust channel** dialog box.

Chapter 15
Analog Module Diagnostics

Subject of this Chapter

This chapter describes the diagnostics aspect in the implementation of analog modules.

What Is in This Chapter?

This chapter contains the following topics:

Diagnostics of an Analog Module

At a Glance

The Module diagnostics function displays errors when they occur, classified according to category:

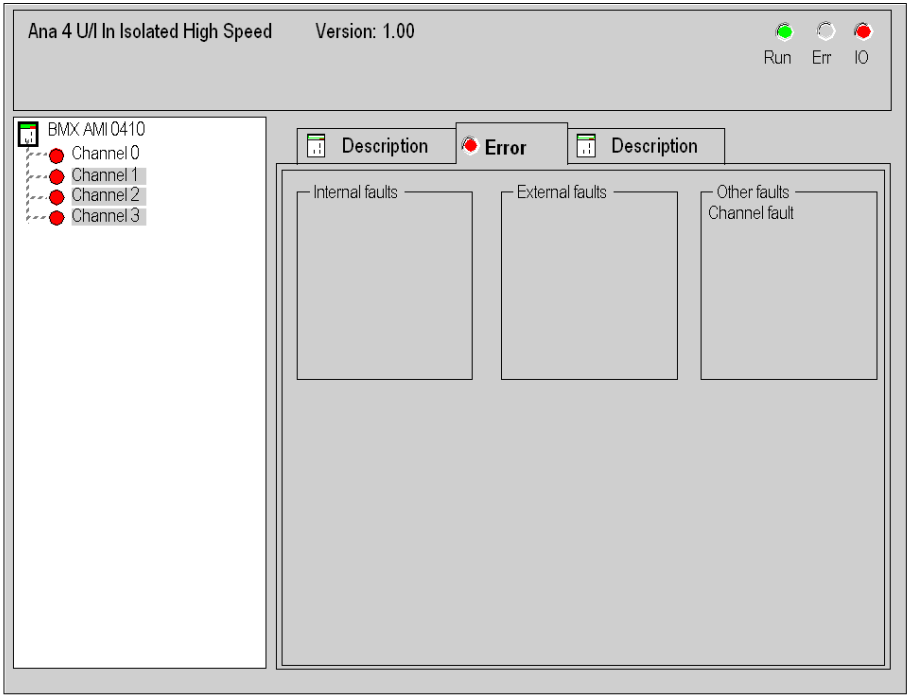
- **Internal detected error**:
  - module malfunction
  - self-testing error
- **External events**:
  - Wiring control (broken-wire, overload or short-circuit)
  - Under range/over range
- **Other errors**:
  - configuration error
  - module missing or off
  - inoperative channel

A module error is indicated by a number of LEDs changing to red, such as:

- in the rack-level configuration editor:
  - the LED of the rack number
  - the LED of the slot number of the module on the rack
- in the module-level configuration editor:
  - the **Err** and **I/O** LEDs, depending on the type of error
  - the **Channel** LED in the **Channel** field

Procedure

The table below shows the procedure for accessing the module Fault screen.

Step	Action
1	Open the module debugging screen.
2	Click on the module reference in the channel zone and select the **Fault** tab. **Result**: The list of module errors appears.  **Note:** It is not possible to access the module diagnostics screen if a configuration error, major breakdown error, or module missing error occurs. The following message then appears on the screen: "`The module is missing or different from that configured for this position.`"

Detailed Diagnostics by Analog Channel

At a Glance

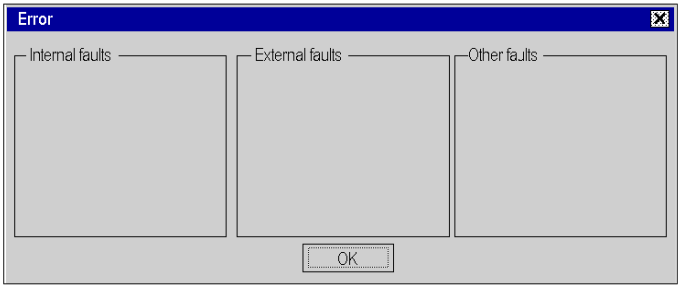
The channel Diagnostics function displays errors when they occur, classified according to category:

- **Internal errors**
  - inoperative channel
  - calibration error
- **External events**
  - sensor link event
  - range overflow/underflow
  - cold junction compensation error
- **Other errors**
  - configuration error
  - communication loss
  - application error
  - value outside range (output channel)
  - channel not ready

A channel error is indicated in the **Debug** tab when the  LED, located in the **Error** column, turns red.

Procedure

The table below shows the procedure for accessing the channel Fault screen.

Step	Action
1	Open the module debugging screen.
2	For the inoperative channel, click on the button  situated in the **Error** column. **Result**: The list of channel errors appears.  **Note:** Channel diagnostics information can also be accessed by program (instruction READ_STS).

Chapter 16
Operating Modules from the Application

Subject of this Chapter

This chapter explains how to operate the analog input/output modules from an application.

What Is in This Chapter?

This chapter contains the following sections:

Section 16.1
Access to the Measurements and Statuses

Subject of this Section

This section indicates how to configure an analog module in order to be able to access the input/outputs measurements and the various statuses.

What Is in This Section?

This section contains the following topics:

Addressing of the Analog Module Objects

At a Glance

The addressing of the main bit and word objects of the analog input/output modules depends upon:

- the rack address
- the physical position of the module in the rack
- the module channel number

NOTE: With Unity Pro 6.1 or later and Modicon M340 firmware 2.4 or later, you can access the modules either via topological or State RAM addresses.
Please refer to *Memory Tab (see Unity Pro, Operating Modes)* and *Topological/State RAM Addressing of Modicon M340 Analog Modules* *(see page 355)*.

Description

Addressing is defined in the following way.

%	I, Q, M, K	X, W, D, F	r	.	m	.	c	.	i	.	j
Symbol	Object type	Format	Rack		Module position		Channel no.		Rank		Word bit

The table below describes the different elements that make up addressing.

Family	Element	Meaning
Symbol	%	-
Object type	I Q	Image of the physical input of the module. Image of the physical output of the module. This information is exchanged automatically for each cycle of the task to which they are attached.
	M	Internal variable. This read or write information is exchanged at the request of the application.
	K	Internal constant. This configuration information is available as read only.
Format (size)	X	Boolean. For Boolean objects the X can be omitted.
	W	Single length.
	D	Double length.
	F	Floating point.
Rack address	r	Rack address.
Module position	m	Module position number in the rack.

Family	Element	Meaning
Channel no.	c	Channel no. 0 to 127 or MOD (MOD: channel reserved for managing the module and parameters common to all the channels).
Rank	i	Word rank. 0 to 127 or ERR (ERR: indicates an error in the word).
Word bit	j	Position of the bit in the word.

Examples

The table below shows some examples of analog object addressing.

Object	Description
%I1.3.MOD.ERR	Error information for the analog input module located in position 3 on rack 1.
%I1.4.1.ERR	Channel 1 error information for the analog input module located in position 4 on rack 1.
%IW1.2.2	Image word for the analog input 2 of the module located in position 2 on rack 1.
%QW2.4.1	Image word for the analog output 1 of the module located in position 4 on rack 2.

Module Configuration

At a Glance

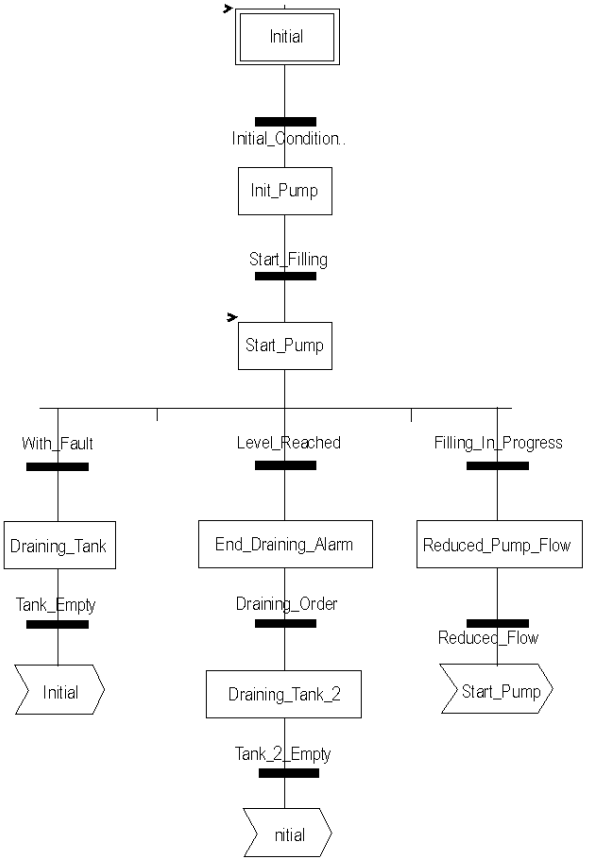
The application used here as an example manages liquid levels in a tank. The tank is filled by a pump and drained using a valve. The different levels of the tank are measured with sensors placed above the tank. The tank should not be filled with more than 100 liters of liquid.

Once the tank is full, the pump stops, and the operator drains the tank manually.

This application requires the use of a BMX AMI 0410 analog input module and a BMX AMO 0210 analog output module. This application may also require a BMX AMM 0600 input/output module.

Tank Management Grafcet

The application's grafcet is as follows:



Using the Measurements

We are going to configure the BMX_AMI_0410 analog input module so that we can retrieve the level of the liquid in the tank.

Step	Action
1	In the `Project browser` and in `Variables & FB instances`, double-click on `Elementary variables`.
2	Create the `INT`-type variable, `Level`.
3	In the `Address` column, enter the address associated with this variable. In our example, we consider that the sensor is connected to channel 0 of the BMX AMI 0410 module. This module is in turn connected to slot 1 of rack 0. We therefore have the following address: %IW0.1.0. Illustration: 

This variable can be used to check whether the level of liquid in the tank has reached maximum level.

To do this, the following line of code can be associated with the `Level_Reached` transition of the grafcet.

If the level of liquid in the tank reaches or exceeds the maximum level, the `Level_Reached` transition is enabled.

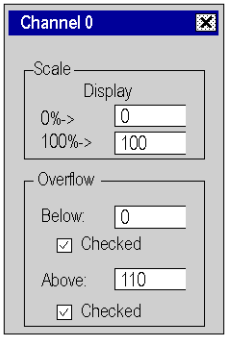
Using the Statuses

We will need to program the `With_fault` transition so that we can stop the pump in three cases:

- the maximum liquid level has been reached
- the pump has been stopped manually
- the measurement falls beyond the upper tolerance area

Before we can use the bit, which will indicate whether the measure still falls within the upper tolerance area (%IWr.m.c.1.6), we need to define the display format and scale of the channel used.

Step	Action
1	Access the hardware configuration screen for the appropriate module.
2	Select the range 0...10 V *(see page 223)* for channel 0.

Step	Action
4	Access the Parameters dialog box *(see page 226)* for the channel in order to input the following parameters:  The upper tolerance area will be between 100 and 110 liters.
5	Confirm your changes by closing the dialog box.
6	Validate the change with **Edit->Validate**.

The code associated with the fault control transition looks like this:

Section 16.2
Additional Programming Features

Subject of this Section

This section presents some useful additional features for the programming of applications that use analog input/output modules.

What Is in This Section?

This section contains the following topics:

Presentation of Language Objects Associated with the Analog Modules

General

Analog modules are associated with different IODDTs.

The IODDTs are predefined by the manufacturer. They contain input/output language objects belonging to a channel of an analog module.

There are several distinct IODDT types for the analog module:

- `T_ANA_IN_BMX` specific to analog input modules such as the BME AHI 0812 and BMX AMI 0410, and specific to the inputs of the BMX AMM 600 mixed module
- `T_ANA_IN_T_BMX` specific to analog input modules such as the BMX ART 0414/0814
- `T_ANA_OUT_BMX` specific to analog output modules such as the BME AHO 0412 and BMX AMO 0210, and specific the outputs of the BMX AMM 600 mixed module
- `T_ANA_IN_GEN` specific to all analog input modules such as the BME AHI 0812, BMX AMI 0410, BMX ART 0414/0814, and the inputs of the BMX AMM 600 mixed module

NOTE: IODDT variables may be created in 2 ways:
- by using the **I/O Objects** tab,
- by using the data editor.

Types of Language Objects

In each IODDT, there exists a set of language objects you can use to control the modules and check their correct operation.

There are 2 types of language objects:

- **Implicit Exchange Objects**, which are automatically exchanged at each cycle of the task assigned to the module. They concern the inputs/outputs of the module (measurement results, information, commands, and so forth).
- **Explicit Exchange Objects**, which are exchanged at the application request, using explicit exchange instructions. They are used to set the module and perform diagnostics.

Implicit Exchange Language Objects Associated with Analog Modules

At a Glance

An integrated interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated interface.

Reminders

The module inputs (`%I` and `%IW`) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

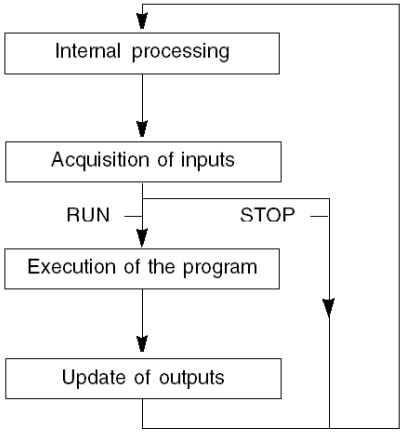
The outputs (`%Q` and `%QW`) are updated at the end of the task, only when the PLC is in RUN mode.

NOTE: When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:

- Outputs are set to fallback position (fallback mode).
- Outputs are maintained at their last value (maintain mode).

Illustration

The operating cycle of a PLC task (cyclical execution) looks like this:



Explicit Exchange Language Objects Associated with Analog Modules

Introduction

Explicit exchanges are performed at the user program's request, using the following instructions:

- READ_STS: read status words
- WRITE_CMD: write command words
- WRITE_PARAM: write adjustment parameters
- READ_PARAM: read adjustment parameters
- SAVE_PARAM: save adjustment parameters
- RESTORE_PARAM: restore adjustment parameters

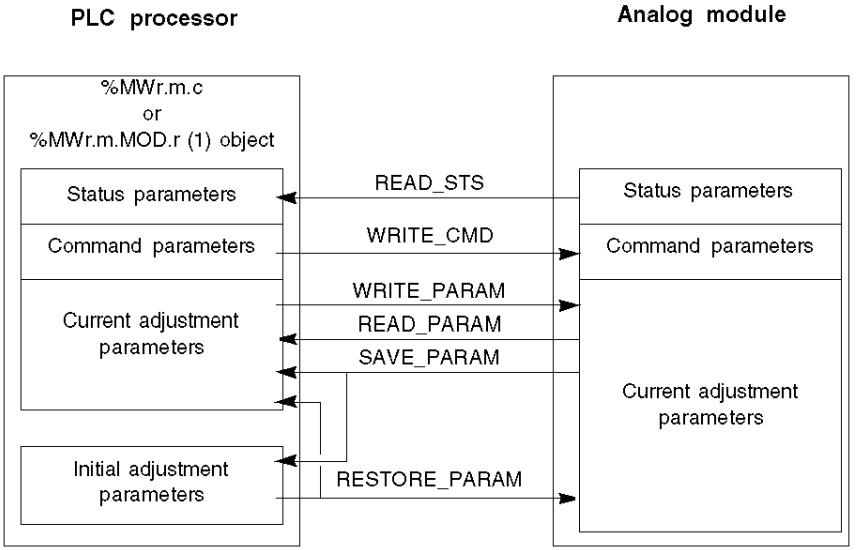
These exchanges apply to a set of %MW objects of the same type (status, commands, or parameters) that belong to a channel.

NOTE: These objects provide information about the module (e.g.: error type for a channel, etc.) and can be used to command them (e.g.: switch command) and to define their operating modes (save and restore currently applied adjustment parameters).

NOTE: You can not send the WRITE_PARAM and RESTORE_PARAM requests at the same time to the channels managed by the same logical nodes, The logical node can only process one request, the other request will generate an error.To avoid this kind of errors you have to manage the exchange for each channel with %MWr.m.c.0.x and %MWr.m.c.1.x.

General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the processor and module.



(1) Only with READ_STS and WRITE_CMD instructions.

Example of Using Instructions

READ_STS instruction:

The `READ_STS` instruction is used to read `SENSOR_FLT` (`%MWr.m.c.2`) and `NOT_READY` (`%MWr.m.c.3`) words. It is therefore possible to determine with greater precision the errors which may have occurred during operation.

Performing a `READ_STS` of all the channels would result in overloading of the PLC. A less burdensome method would be to test the error bit of all the modules in each cycle, and then the channels of the modules in question. You would then only need to use the `READ_STS` instruction on the address obtained.

The algorithm could look like this:

```
WHILE (%I0.m.ERR <> 1) OR (m <= Number of modules) THEN
  m=m+1
  Loop
END WHILE
```

```
WHILE (%I0.m.c.ERR <> 1) OR (c <= Number of channels) THEN
  c=c+1
  Loop
END WHILE

READ_STS (%I0.m.c)
```

WRITE_PARAM instruction:

The WRITE_PARAM instruction is used to modify certain configuration parameters for the modules during operation.

All you need to do is to assign the new values to the relevant objects and use the WRITE_PARAM instruction on the required channel.

For example, you can use this instruction to modify the fallback value by program (only for output analog modules). Assign the required value to the Fallback(%MWr.m.c.7) word and then use the WRITE_PARAM instruction.

Management of Exchanges and Reports with Explicit Objects

At a Glance

When data is exchanged between the PLC memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:

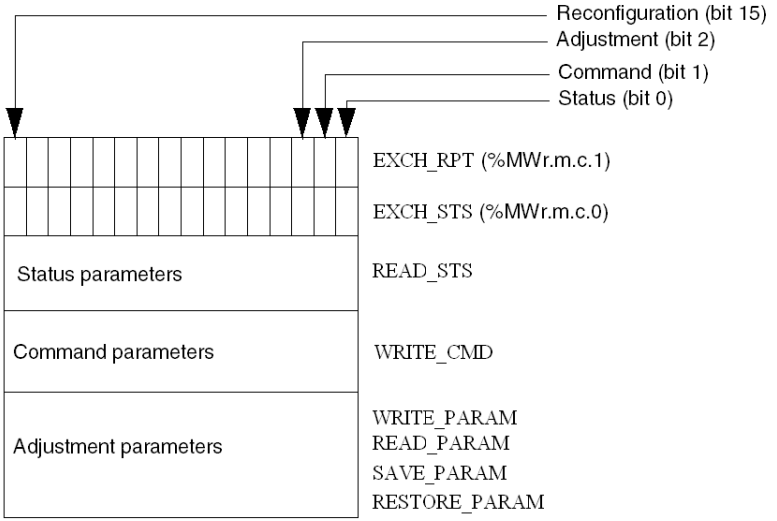
- `EXCH_STS` (`%MWr.m.c.0`) : exchange in progress
- `EXCH_RPT` (`%MWr.m.c.1`) : report

NOTE: Depending on the localization of the module, the management of the explicit exchanges (`%MW0.0.MOD.0.0 for example) will not be detected by the application`:

- for in-rack modules, explicit exchanges are doneimmediately on the local PLC Bus and are finished before the end of the executon task, so the `READ_STS, for example, is always finished when the %MW0.0.mod.0.0 bit is checked by the application.`
- for remote bus (Fipio for example), explicit exchanges are not synchronous with the execution task, so the detection is possible by the application.

Illustration

The illustration below shows the different significant bits for managing exchanges.



Description of Significant Bits

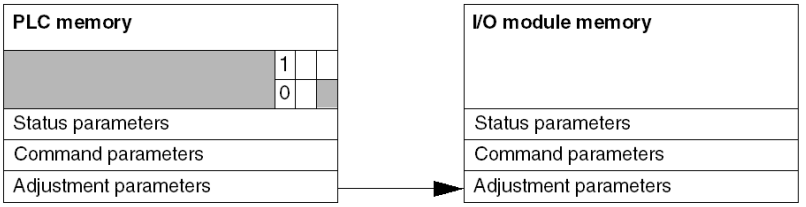
Each bit of the EXCH_STS (%MWr.m.c.0) and EXCH_RPT (%MWr.m.c.1) words is associated with a type of parameter:

- Rank 0 bits are associated with the status parameters:
  - The STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress.
  - The STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - The CMD_IN_PROGR bit (%MWr.m.c.0.1) indicates whether command parameters are being sent to the module channel.
  - The CMD_ERR bit (%MWr.m.c.1.1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - The ADJ_IN_PROGR bit (%MWr.m.c.0.2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM).
  - The ADJ_ERR bit (%MWr.m.c.1.2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- Rank 15 bits indicate a reconfiguration on channel c of the module from the console (modification of the configuration parameters and cold start-up of the channel).
- Bits r, m, and c indicate the following slots:
  - Bit r represents the rack number.
  - Bit m represents the position of the module in the rack.
  - Bit c represents the channel number in the module.

NOTE: Exchange and report words also exist at the level of EXCH_STS (%MWr.m.MOD.0) and EXCH_RPT (%MWr.m.MOD.1) modules, as per T_ANA_IN_BMX, T_ANA_IN_T_BMX and T_ANA_OUT_BMX-type IODDTs.

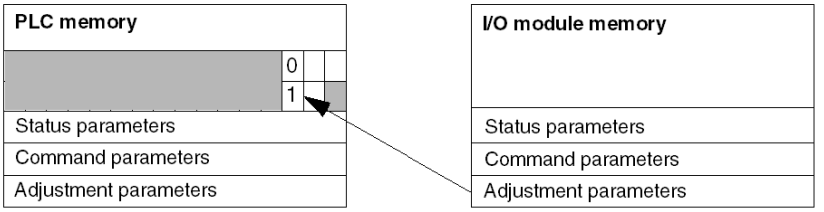
Example

Phase 1: Sending data by using the WRITE_PARAM instruction:



When the instruction is scanned by the PLC processor, the Exchange in progress bit is set to 1 in %MWr.m.c.

Phase 2: Analysis of the data by the input/output module and report:



When data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the ADJ_ERR (%MWr.m.c.1.2) bit which, depending on its value, gives the following report:

- **0:** correct exchange.
- **1:** error in exchange.

NOTE: There is no adjustment parameter at module level.

Explicit Exchange Execution Flag: EXCH_STS

The table below shows the EXCH_STS (%MWr.m.c.0) explicit exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of channel status words in progress	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjust parameters exchange in progress	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguration of the module in progress	%MWr.m.c.0.15

NOTE: If the module is not present or is disconnected, explicit exchange objects (READ_STS, for example) are not sent to the module (STS_IN_PROG (%MWr.m.c.0.0) = 0), but the words are refreshed.

Explicit Exchange Report: EXCH_RPT

The table below presents the `EXCH_RPT` (`%MWr.m.c.1`) report bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error reading channel status words (1 = error)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during a command parameter exchange (1 = error)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters (1 = error)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error during reconfiguration of the channel (1 = error)	%MWr.m.c.1.15

Language Objects Associated with Configuration

At a Glance

The configuration of an analog module is stored in the configuration constants (%KW).

The parameters r, m, and c shown in the following tables represent the topologic addressing of the module. Each parameter had the following signification:

- **r:** represents the rack number
- **m:** represents the position of the module on the rack
- **c:** represents the channel number

BME AHI 0812, BMX AMI 0410, BMX AMI 0800, and BMX AMI 0810 Configuration Objects and Inputs of BMX AMM 0600

The process control language objects associated to the configuration of the BME AHI 0812, BMX AMI 0410, BMX AMI 0800, and BMX AMI 0810 modules include the following:

Addresses	Description	Bits Meaning
%KWr.m.c.0	Channel range configuration	**Bit 0 to 5:** electric range (hexadecimal value) **Bit 7:** 0=electrical range (always 0)
%KWr.m.c.1	Scale/User scaling min value	-
%KWr.m.c.2	Scale/User scaling max value	-
%KWr.m.c.3	Over range below value	-
%KWr.m.c.4	Over range above value	-
%KWr.m.c.5	Channel treatment configuration	**Bit 0:** 0=Mast mode, 1=Fast mode **Bit 1:** 0=channel disabled, 1=channel enabled **Bit 2:** 0=sensor monitor off, 1=sensor monitor on **Bit 7:** 0=Manufacturer scale, 1=user scale **Bit 8:** over range lower threshold enabled **Bit 9:** over range upper threshold enabled

BMX ART 0414/0814 Configuration Objects

The process control language objects associated to the configuration of the BMX ART 0414/0814 modules include the following:

Addresses	Description	Bits Meaning
%KWr.m.c.0	Channel range configuration	**Bit 0 to 5:** Temperature range (hexadecimal value) **Bit 6:** Temperature range (0=° C, 1=F°) **Bit 7:** 1=Temperature range **Bit 8:** 0=rejection 50 Hz, 1=rejection 60 Hz
%KWr.m.c.1	Scale/User scaling min value	-
%KWr.m.c.2	Scale/User scaling max value	-
%KWr.m.c.3	Over range below value	-
%KWr.m.c.4	Over range above value	-
%KWr.m.c.5	Channel treatment configuration	**Bit 0:** 0=Standard mode (always 0) **Bit 1:** 0=channel disabled (only in Fast mode), 1=channel enabled **Bit 2:** 0=sensor monitor off, 1=sensor monitor on **Bits 3 to 6:** CJC Configuration Mode for channels 0/3: ● Bit 3=0 and Bit 4=0: Int. Telefast, ● Bit 3=1 and Bit 4=0: External RTD, ● Bit 3=0 and Bit 4=1: CJC on channels 4/7. **Bits 3 to 6:** CJC Configuration Mode for channels 4/7: ● Bit 5=0 and Bit 6=0: Int. Telefast, ● Bit 5=1 and Bit 6=0: External RTD. **Bit 7:** 0=Manufacturer scale, 1=user scale **Bit 8:** Over range lower threshold enabled **Bit 9:** Over range upper threshold enabled

BME AHO 0412, BMX AMO 0210, BMX AMO 0410, and BMX AMO 0802 Configuration Objects and Outputs of BMX AMM 0600

The process control language objects associated to the configuration of the BME AHO 0412, BMX AMO 0210, BMX AMO 0410, and BMX AMO 0802 modules include the following:

Addresses	Description	Bits Meaning
%KWr.m.c.0	Channel range configuration	**Bit 0 to 5:** Electric range (hexadecimal value) **Bit 8:** Fallback mode (0=Fallback, 1=Maintain) **Bit 11:** Actuator wiring control (0=disabled, 1=enabled) **Bit 14:** Output lower overshoot below range valid (0=disabled, 1=enabled) **Bit 15:** Output upper overshoot above range valid (0=disabled, 1=enabled)
%KWr.m.c.1	Scale/User scaling min value	-
%KWr.m.c.2	Scale/User scaling max value	-
%KWr.m.c.3	Overshoot below value	-
%KWr.m.c.4	Overshoot above value	-

Part III
Quick Start: Example of Analog I/O Module Implementation

In this Part

This part presents an example of implementation of the analog input/output modules.

What Is in This Part?

This part contains the following chapters: