ModBus

IAN FELLOWS LTD

Industrial Weighing Instruments CSW, CSW-20, CPI

Program versions PO6.xxx

INDUSTRIAL WEIGHING INSTRUMENTS

CSW, CSW-20, CPI

MODBUS SPECIFICATION

Lower Keyford • Frome
Somerset BA11 4AR • England
Phone 01373 473161 • Fax 01373.451609

MODBUS Introduction

The common language used by all AEG Modicon controllers, and subsequently many of other manufacture, is the ModBus protocol. This protocol defines a message structure that controllers will recognise and use, regardless of the type of network over which they communicate. It describes the process a controller uses to request access to another device, how it will respond to requests from other devices, and how errors are detected and reported. It establishes a common format for the layout and contents of the message fields.

For this implementation controllers communicate using a master- slave technique, in which only one device (the master) can initiate transactions (called 'queries'). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query.

Typical master devices include host processors and programming panels.

Typical slaves include programmable controllers and intelligent instruments.

The CSW, CPI SERIES INDICATORS may be interfaced to the ModBus master via an RS232 or RS485 interface. Alternatively (it may be interfaced) to the ModBus master and to any further slave devices via an RS485 interface
Each CSW, CPI SERIES INDICATOR on the interface appears to the ModBus master as a Modicon controller with a limited register, coil and function code set.

Each instrument will have its own slave address.

The CSW, CPI SERIES INDICATORS will respond as per MODICON specification whenever indicator is in (default) weighing MODE. It will NOT provide any response when in MENU MODE or until POWER-ON sequence has been successfully completed.

The ModBus ALARM status condition is a latched function.
It will be set ON by a wide range of diagnostic error functions
It may be reset OFF via the Front panel TEST button or via reset alarm command.
The source of the ALARM condition may be traced by reference to the Error Log

Implementation

Implementation is essentially as defined in MODICON, Modbus Protocol Reference Guide PI-MBUS-300 Rev. E

Restricted to:

- Slave Mode
- RTU Mode only
- No Broadcast facility
- Slave Device addresses 1-99
- ModBus function Codes 4, 8, 16 only, as defined below. (Else EXCEPTION CODE 1)
- Exception codes 1,2,3 only implemented
- Implemented for RS232 or RS485.

ModBus uses the standard CSW, CPI SERIES INDICATOR communications interface. Standard communication protocols will be disabled once ModBus is enabled.

For a (typical) ModBus poll rate of 100ms. there should be no discernable degradation of weighing response.

(Note that for ModBus commands input to CSW, CPI SERIES INDICATORS standard response times will continue to apply i.e. commands incur internal eeprom write delay).

Setup / Operation

- The facility to select Modbus communication is a factory configured option. Units ordered with Modbus capability can be set to run Modbus or standard (ascii) IFL communications (default).
- Suggested procedure for ModBus configuration;
  - Perform all calibration /configuration before setting up ModBus
  - Connect required (RS232 or RS422/ 485) interface as detailed in the Operations Manual
  - Set required Communications parameters SERIAL_/bAUd (EV)
    & SERIAL_/PrtY (CP)
  - Set SERIAL_/Addr (AE) to desired slave address (dec 01-99)
  - Set SERIAL_/ΞbUS(MD) =1.
  - The indicator will now respond to ModBus commands as detailed below.

  NOTES
  - Note that once the ModBus facility is enabled the only facilities available through the serial interface are as described below.

ModBus RTU Mode

Format for each byte is-

Coding 8 bit binary, hex 0-9,A-F
2 hex characters in each 8 bit field

Bits per byte 1 start, 8 data, LSB first. Selectable parity O, E, N.
1 stop bit if parity, 2 if none.

Error Check field Cyclical Redundancy Check (CRC)

Framing

- Allowable characters for all fields are 0-9,A-F
- All devices monitor the network bus continuously, including during the 'silent' intervals. When the first field (the address field) is received, each device decodes it to establish whether it is the addressed device.
- All messages end with a further silent interval of at least 3.5 character periods.
- The entire frame must be transmitted as a continuous stream.
- Flush on silent interval of > 1.5 character times. Assumes next character will be new message.
- If message begins less than 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will generate an error, as the value in the final CRC field will not be valid for the combined messages.

Typical Message Frame

Start	Address	Function	Data	CRC Check	End
T1-T2-T3-T4	8 bits H-L	8 bits H-L	n x 8 bits H-L	16 bits L-H	T1-T2-T3-T4

Address Field

- Valid codes will be 1-99.

Function Field

- The following will be the only valid function codes-
  4 Read Input Registers
  8 Diagnostics
  16 Preset Multiple Registers
- Sent by master and returned by slave if a normal response.
- The slave response will have the most significant bit marked for an exception (error) response.

Data Field

- Contains the data sent or received according to the function.
- For an exception response this field will contain the exception code.

Exception Response Codes

- When the instrument receives a message that is logically incorrect (other than a CRC error) it marks the MS bit of the returned address and returns a error string in the response data field;

Code	Error
01	Incorrect function code
02	Incorrect address
03	Incorrect data

Weight Representation

- All weight related parameters (* in tables below) use scaled binary format for weight data, setpoint entry and transmission.
  - All weight data is represented as 8 BCD 'nibbles' (4 bytes, 2 registers)
  - 1st register: sddd bbbb bbbb bbbb 2nd register: bbbb bbbb bbbb tttt
  - s=sign bit (1=neg), ddd=DP position (000=0dp, 001=1dp, 010=2dp, etc)
  - bbbb/tttt=bcd digit (7 available)
  - Always right justified – tttt is zero unless in test (ET=1) mode.

Protocol

- The following functions will be provided in order to achieve the requirements.

AA= Address, HH,LL= CRC, XX= Data, QQ =Response is Echo of similar QUERY field

ModBus function codes

Allocated Read Registers.

Read Registers	Byte Count	Bit	Function
30001 -30002	4	-	GROSS Weight *
30003	2		Low byte – Weight Status ; High Byte I/O Status
30003	Low	0	ZERO i.e. AZ successful
		1	NET @ 0 i.e. AT successful ('frozen' @ <+/-0.5e)
		2	TRUE GROSS i.e. CT successful (no tares)
		3	Filling status ST=1
		4	Gross Display mode
		5	NET display Mode
		6	Not used
		7	MOTION
30003	High	0	Input 1 (current state)
		1	Input 2 (current state)
		2	Output 1
		3	Output 2
		4	Output 3
		5	Input 1 latched (ie has been asserted since last reset)
		6	Input 2 latched (reset by FN16 register 40003 bits 5/6)
		7	Alarm (latched)
30004 –30005	4	-	NET Weight*
30006 -30007	4	-	TARE Weight (Semi-auto)*
30008-30100			*Reserved for future extensions*
30101-30102	4	-	Set-point 1*
30103-30104	4	-	Set-point 2*
30105-30106	4	-	Set-point 3*
30107-30108	4	-	In-flight weight value*
30109-30200			*Reserved for future extensions*
30201-30202	4	-	Calibration Date (00ddmmyy) as a 4 byte bcd
30203-30204	4	-	Serial Number (0-999999) as a 4 byte bcd
30205-30208	8	-	Calibrated by (aaaaaaaa) as 8 ASCII bytes/charas
30210-30300			*Reserved for future extensions*
30301	2	-	Product code 00-99 – 2 hex (bcd) low byte – high byte=00h
30302-30205	8	-	ST1A/SA – text as 8 ASCII bytes/charas (1st chara is nul – following 7 relevant)
30306-30309	8	-	ST1B/SB
30310-30313	8	-	ST2A/SC
30314-30317	8	-	ST2B/SD
30318 – 30400			*Reserved for future extensions*
30401	2		Alarm Error Code
30401	Low	0	Power On Reset
		1	Menu entry has occurred
		2	Uncalibrated
		3	Defaults reloaded
		4	Millivolt mode
		5	Invalid command
		6	Overrange
		7	Underrange
30401	High	0	Keyboard fail
		1	Clock/calendar battery low
		2	Cell Error
		3	Voltage Fault
		4	Watchdog trip
		5	Eeprom Error
		6	Ram Error
		7	ADC Error
30402-30500			*Reserved for future extensions*
30501	2	-	Tare Look Up (TLU) register XX (bcd) currently selected
30502-30503	4	-	Read Preset Tare weight value for current TLU*

Bit 7(high byte) of 30003 is latched on by assertion of any of the alarm condition bits in 30401 and is reset by setting bit7 (high byte) in register 40003 (FN 16). The Alarm bit will be set again as soon as another fault is detected.

FN4 Read Input Registers

NB: The formats below are shown, but are not now specific. Data is returned until, at least, the 'No. of Points' is satisfied, or the end of a register group is reached (defined by the registers between green bands in the table above – i.e. registers in the same '100' number group). The byte count in the response will be correct, but may not correspond with twice the register count ('No. of Points') in the query. (The 'No. of Points' will not correspond if, for example, '0001' point was requested, but the register was '30001' (a 32-bit data field). Previously, such a request would have been given an 'EXCEPTION 3' response. This is not thought necessary. In the example, a full 32-bit response – 2 registers, 4 bytes – will result, even though only one register was requested.)

Query **Obtain Weight data (fast/ full +status +alarm)**

Field Name	Byte	Byte	Byte	Response
Slave address	***AA***	***AA***	***AA***	Match else NO Response
Function	04	04	04	
Starting address Hi	00	00	01	else EXCEPTION CODE 2
Starting address Lo	00	00	90(h)	else EXCEPTION CODE 2
No. of points Hi	00	00	00	else EXCEPTION CODE 3 (see above)
No. of points Lo	03	07	01	else EXCEPTION CODE 3
Error Check CRC Hi	***LL***	***LL***	***LL***	Match else NO Response
Error Check CRC Lo	***HH***	***HH***	***HH***	Match else NO Response

Response

Field Name	Byte	Byte	Byte	Function
Slave address	***AA***	***AA***	***AA***	Match else invalid
Function	04	04	04	Match else invalid
Byte Count	06	OD(=14d)	02	Match else invalid
30001	***XXXX***	***XXXX***		GROSS Hi*
30002	***XXXX***	***XXXX***		GROSS Lo*
30003	***XXXX***	***XXXX***		Status*
30004		***XXXX***		NET Hi*
30005		***XXXX***		NET Lo*
30006		***XXXX***		TARE Hi*
30007		***XXXX***		TARE Lo*
30401			***XXXX***	Alarm flags
Error Check CRC Hi	***LL***	***LL***	***LL***	Match else invalid
Error Check CRC Lo	***HH***	***HH***	***HH***	Match else invalid

FN4 Read Input Registers

Query **Obtain set-point, cal data, text string, alarm info.**

Field Name	Byte	Byte	Byte	Response
Slave address	***AA***	***AA***	***AA***	Match else NO Response
Function	04	04	04	
Starting address Hi	00	00	01	else EXCEPTION CODE 2
Starting address Lo	64(h)	C8(h)	2C(h)	else EXCEPTION CODE 2
No. of points Hi	00	00	00	else EXCEPTION CODE 3
No. of points Lo	08	08	11(h) 17(d)	else EXCEPTION CODE 3
Error Check CRC Hi	***LL***	***LL***	***LL***	Match else NO Response
Error Check CRC Lo	***HH***	***HH***	***HH***	Match else NO Response

Response

Field Name	Byte	Byte	Byte	Function
Slave address	***AA***	***AA***	***AA***	Match else invalid
Function	04	04	04	Match else invalid
Byte Count	0F(=16d)	0F(=16d)	22(=34d)	Match else invalid
30101	***XXXX***			Set point 1 HI
30102	***XXXX***			Set point 1 Lo
30103	***XXXX***			Set point 2 Hi
30104	***XXXX***			Set point 2 Lo
30105	***XXXX***			Set point 3 Hi
30106	***XXXX***			Set point 3 Lo
30107	***XXXX***			In-flight Hi
30108	***XXXX***			In-flight Lo
30201 – 30202		***00XXXXXX***		Cal date as 4 byte hex
30203 – 30204		***XXXXXXXX***		Serial no as 4 byte hex
30205 – 30208		***AAAAAAAA***		Cal by – aaaaaaaa
30301			***00XX***	Product code low byte
30302 – 30305			***0AAAAAAA***	Text string 1A – 7 Alpha bytes
30306 – 30309			***0AAAAAAA***	Text string 1B – 7 Alpha bytes
30310 – 30313			***0AAAAAAA***	Text string 2A – 7 Alpha bytes
30314 – 30317			***0AAAAAAA***	Text string 2B – 7 Alpha bytes
Error Check CRC Hi	***LL***	***LL***	***LL***	Match else invalid
Error Check CRC Lo	***HH***	***HH***	***HH***	Match else invalid

FN4 Read Input Registers

Query **Obtain tare look up.**

Field Name	Byte	Response
Slave address	***AA***	Match else NO Response
Function	04	
Starting address Hi	01	Else EXCEPTION CODE 2
Starting address Lo	F4(h)	Else EXCEPTION CODE 2
No. of points Hi	00	Else EXCEPTION CODE 3
No. of points Lo	03	Else EXCEPTION CODE 3
Error Check CRC Hi	***LL***	Match else NO Response
Error Check CRC Lo	***HH***	Match else NO Response

Response

Field Name	Byte	Function
Slave address	***AA***	Match else invalid
Function	04	Match else invalid
Byte Count	06	Match else invalid
30501	***00XX***	Tare look up
30502	***XXXX***	Preset tare value low*
30503	***XXXX***	Preset tare value high*
Error Check CRC Hi	***LL***	Match else invalid
Error Check CRC Lo	***HH***	Match else invalid

FN8 Diagnostics

Query

Field Name	Byte	Response
Slave address	***AA***	Match else NO Response
Function	08	
Subfunction Hi	00	else EXCEPTION CODE 2
Subfunction Lo	00	else EXCEPTION CODE 2
Data Hi	***XX***	
Data Lo	***XX***	
Error Check CRC Hi	***LL***	Match else NO Response
Error Check CRC Lo	***HH***	Match else NO Response

Response

Field Name	Byte	Function
Slave address	***AA***	Match else invalid
Function	04	Match else invalid
Register Address Hi	***QQ***	Match else invalid
Register Address Lo	***QQ***	Match else invalid
Preset Data Hi	***QQ***	Match else invalid
Preset Data Lo	***QQ***	Match else invalid
Error Check CRC Hi	***LL***	Match else invalid
Error Check CRC Lo	***HH***	Match else invalid

NOTE Valid only for sub-function 0,0 –

Just for Communication TEST

Some applications use as "Are you there" function

Allocated Write Registers.

Write Registers	Byte Count	Bit	Function
40001 -40002	4	-	*Read only*
40003	2		Low byte – Weight OPS ; High Byte I/O OPS (see below)
40003	Low	0	Perform Set Zero (1)
		1	Perform Auto Tare operation (1)
		2	Perform Cancel Tare operation (1)
		3	Start Batch (1)
		4	Switch indicator to display Gross Mode
		5	Switch indicator display to Net Mode (1)
		6	Switch indicator display to Tare Mode (1)
		7	Stop Batch (1)
40003	High	0-4	*Read Only – values ignored – remote setting of outputs not permitted.*
		5	If =1 clear latched input 1
		6	If =1 clear latched input 2
		7	If =1 attempt to clear Alarm status
40004 - 40005	4	-	*Read only*
40006 - 40007	4	-	*Read only*
40008-40100			*Reserved for future extensions*
40101-40102	4	-	Setpoint 1*
40103-40104	4	-	Setpoint 2*
40105-40106	4	-	Setpoint 3*
40107-40108	4	-	Inflight* (set into current Product code as set by register 40301)
40109-40200			*Reserved for future extensions*
40201-40202	4	-	*Read only*
40203-40204	4	-	*Read only*
40205-40208	8	-	Calibrated by (aaaaaaaa) as 8 ASCII bytes/charas
40209-40300			*Reserved for future extensions*
40301	2	-	Product code 00-99 * – 2 bcd low byte – high byte=00h
40302-40205	8	-	ST1A/SA – text as 8 ASCII bytes/charas (1st chara is ignored – following 7 stored)
40306-40309	8	-	ST1B/SB
40310-40313	8	-	ST2A/SC
40314-40317	8	-	ST2B/SD
40318-40400			*Reserved for future extensions*
40401	2	-	*Read only*
40501	2	-	Select Tare Look up register 00XX – activate stored preset tare
40502-40503	4	-	Set Preset tare value to TLU*

Bit register 40003 may have multiple bits set, but only one in each of the 3 groups below as they are mutually exclusive (EXCEPTION 3 if tried).
The groups are handled in the order:
A) Zero/ Auto Tare/ Cancel Tare
B) Display Gross/ Net/ Tare
C) Start/ Stop

Apart from the above, only single parameters may be written (unlike the FN4 read operations).

FN16 Preset Multiple registers

Query

	Single Register	Double Register	X4 Register*	
Field Name	**Byte**	**Byte**	**Byte**	**Response**
Slave address	***AA***	***AA***	***AA***	Match else NO Response
Function	10 (=16d)	10 (=16d)	10 (=16d)	
Starting Address Hi	XX	XX	XX	else EXCEPTION CODE 2
Starting Address Lo	XX	XX	XX	(See table below)
No. of registers Hi	00	00	00	else EXCEPTION CODE 3
No. of registers Lo	01	02	04	else EXCEPTION CODE 3
Byte count	02	04	08	else EXCEPTION CODE 3
Preset Data Hi	***XX***	***XXXX***	***XXXXXXXX***	See table below
Preset Data Lo	***XX***	***XXXX***	***XXXXXXXX***	
Error Check CRC Hi	***HH***	***HH***	***HH***	Match else NO Response
Error Check CRC Lo	***LL***	***LL***	***LL***	Match else NO Response

Response

Field Name	**Byte**	**Byte**	**Byte**	**Function**
Slave address	***AA***	***AA***	***AA***	Match else invalid
Function	10 (=16d)	10 (=16d)	10 (=16d)	Match else invalid
Starting Address Hi	***QQ***	***QQ***	***QQ***	Match else invalid
Starting Address Lo	***QQ***	***QQ***	***QQ***	Match else invalid
No. of registers Hi	***QQ***	***QQ***	***QQ***	Match else invalid
No. of registers Lo	***QQ***	***QQ***	***QQ***	Match else invalid
Error Check CRC Hi	***HH***	***HH***	***HH***	Match else invalid
Error Check CRC Lo	***LL***	***LL***	***LL***	Match else invalid

References:

- AEG Modicon, ModBus Protocol Reference Guide.
- IAC 4.2 ModBus RTU Protocol