ScandiCAT ModbusTCP specification

Revision history

Rev no	Author	Reviewer	Approver	Description
00	FH Sign	Sign	Sign	Initial
Rev no	Author Sign	Reviewer Sign	Approver Sign	Description

1. Introduction

1.1. The internal Modbus TCP server enables communication over an Ethernet connection (TCP/IP, 100Mbit or 1000Mbit, IPv4 or IPv6) using the Modbus protocol.

1.2. Modbus is an open standard in industrial communication which is maintained by the independent Modbus Organization. The official specification can be found here:
http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

Here you find a good tutorial:
http://www.rtaautomation.com/modbustcp/

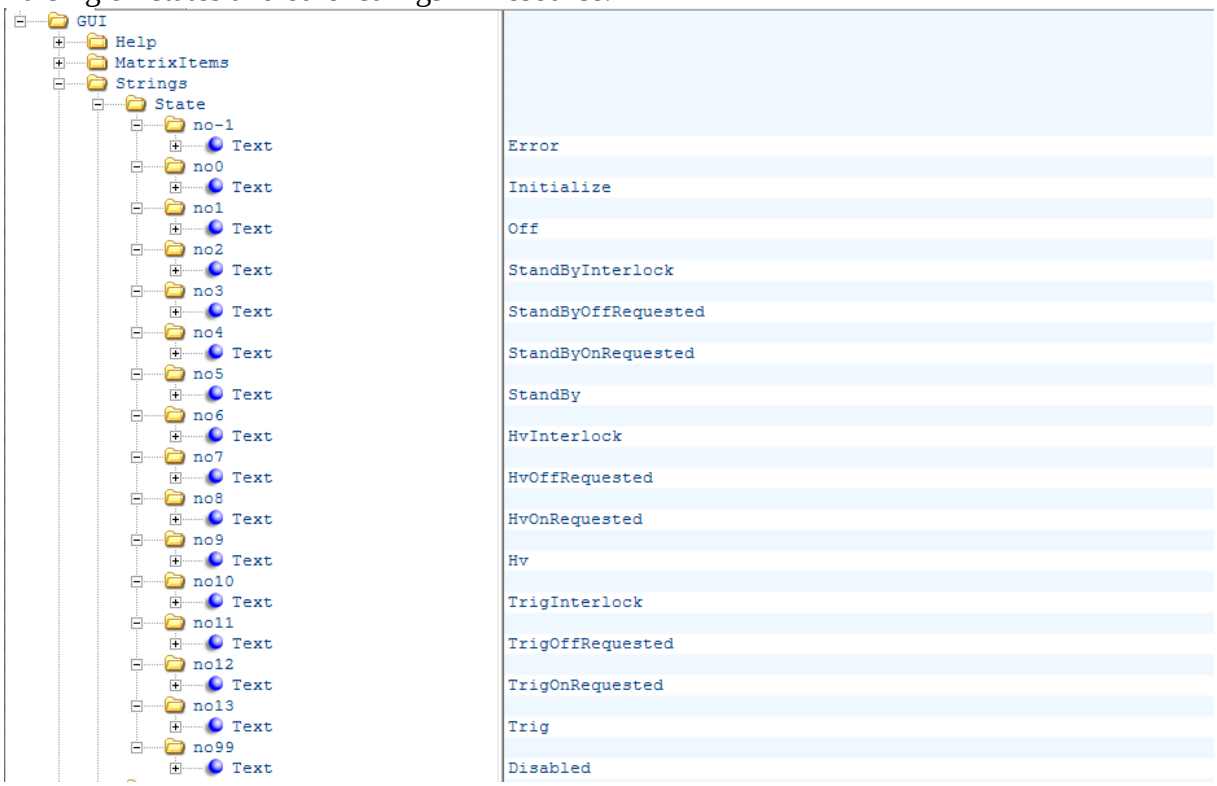
1.3. Refresh rate depends on how big CPU is used in the modulator and also the size/complexity of the modulator, large data such as log-files or waveforms will be slower.

1.4. This protocol is not intended for synchronous streaming of data, such communication can be handled via the optional EtherCAT interface.

1.5. The file Recources.xml should be used by your system as a look-up table for strings, integer values for states etc and simplifies the understanding of the state machine related registers:

StateTarget
StateRead
EventLogg
MainEvent
FirstInterlockEvent

1.6. Parsing of states and other strings in Resource.xml



To interpret the state number in registers that contain state you should use the file *C:\ScandiCAT GUI\Contents\Resource.xml* .
The states are found in GUI/Strings/State. See the image above.
Example: If you read the Modbus input register with address 3 (the actual state of the modulator) and get the value 9, you look in GUI/Strings/State/no9 to get the name of the state (Hv).

Other strings are interpreted the same way. A number X read from a Modbus register corresponds to the string in GUI/Strings/<stringtype>/noX .

1.7. All registers are little-endian (Intel order)

1.8. The Modbus-TCP server uses the following configuration file “C:\Mbus\ScandiCAT ModbusTCP.xml” (the register list below is generated from this file).

1.9. Event log (Address 999-1649)

Reading the eventLogg is optional (you can control the modulator without this)
Here the fifty last events of the modulator are made available in eight Circular Buffer arrays
Usage: Poll the Event logg index (Adress 999) every second, if Index<>IndexPrev (your local variable) then irncrement IndexPrev by one and copy the data at IndexPrev of each array to your own event storage and continue to evaluate Index, if Index<> IndexPrev continue to copy the next index of each array and so on until Index= IndexPrev. Optionally you could also add a check that new events must be newer than the previous by comparing TimeStamp[IndexPrev]<> TimeStampPrev (your local variable)

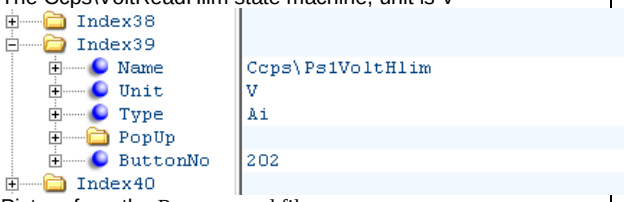
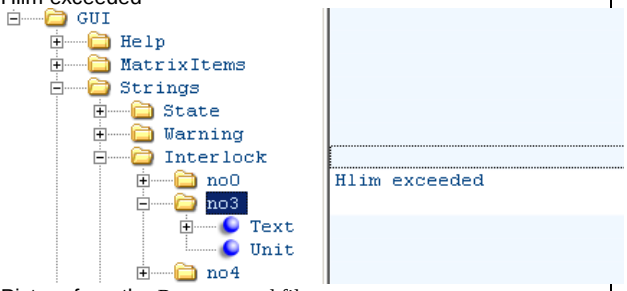
vb.net example:

```
While Me.IndexPrev <> NewIndex
    Me. IndexPrev += 1
    If Me. IndexPrev = 50 Then Me. IndexPrev = 0

    dataStream1.Position = Me. IndexPrev * 2 'This stream starts at Modbus input adress 1000
    dataStream2.Position = Me. IndexPrev * 2 'This stream starts at Modbus input adress 1050
    dataStream3.Position = Me. IndexPrev * 8 'This stream starts at Modbus input adress 1100
    dataStream4.Position = Me. IndexPrev * 4 'This stream starts at Modbus input adress 1300
    dataStream5.Position = Me. IndexPrev * 2 'This stream starts at Modbus input adress 1400
    dataStream6.Position = Me. IndexPrev * 2 'This stream starts at Modbus input adress 1450
    dataStream7.Position = Me. IndexPrev * 2 'This stream starts at Modbus input adress 1500
    dataStream8.Position = Me. IndexPrev * 4 'This stream starts at Modbus input adress 1550

    Me.Incr = binRead1.ReadUInt16
    Me.Type = binRead2.ReadUInt16
    Me.Time = binRead3.ReadUInt64
    Me.TrigId = binRead4.ReadInt32
    Me.Index = binRead5.ReadUInt16
    Me.TextNo = binRead6.ReadUInt16
    Me.DataType = binRead7.ReadUInt16
    Me.Data = binRead8.ReadBytes(4)

    'If new event...
    If Me.Time > Me.NewestLoggedEvent Or _
        (Me.Time = Me.NewestLoggedEvent And Me.Index <> Me.NewestLoggedIndex) Then

        '...add to EventList

    End If

End While
dataStream1.Close()
dataStream2.Close()
dataStream3.Close()
dataStream4.Close()
dataStream5.Close()
dataStream6.Close()
dataStream7.Close()
dataStream8.Close()
```

1.10. Parsing an event

Each event consists of a number of data (from Address 999…)
It's a good idea to fetch names/strings from the Resource.xml file

Example of an interlock event:

Event logg index	7	The newest event is at index 7 in each array
Event logg increment array	7667	Just increases for each event
Event logg type array	2	0=State, 1=Warning, **2=Interlock,** 3=Error, 4=Parameter, 5=Message
Event logg time-stamp array	0	This event either occurred 1601-01-01 or 2000-01-01 00:00:00:000:000:0 depending on if the modulator uses UTC FileTime or EpicsTime
Event logg trig id array	98089	The current trig number at the time of the event
Event logg index array	39	The Ccps\VoltReadHlim state machine, unit is V  Picture from the Resource.xml file
Event logg text number array	3	Hlim exceeded  Picture from the Resource.xml file
Event logg data type array	1	**1=Real, (PLC name for single precision floating point)** 2=Bool, 3=Int, 4=Uint, 5=Word, 6=Dint, 7=Udint, 8=Dword
Event logg data array	44A56000	= 1323.0 V

1.11. MainEvent (Address 1700-1712)

Reading the MainEvent is optional (you can control the modulator without this)
Actual modulator state in the event format, priority:

1. Errors (when not active first interlock is showed instead)
2. First interlock (when not active warning is showed instead)
3. Warning (when not active state is showed instead)
4. State

1.12. FirstInterlockEvent (Address 1715-1727)

Reading the FirstInterlockEvent is optional (you can control the modulator without this). This is a copy of the first interlock since previous reset, after reset this event type is set to 0 to indicate not active

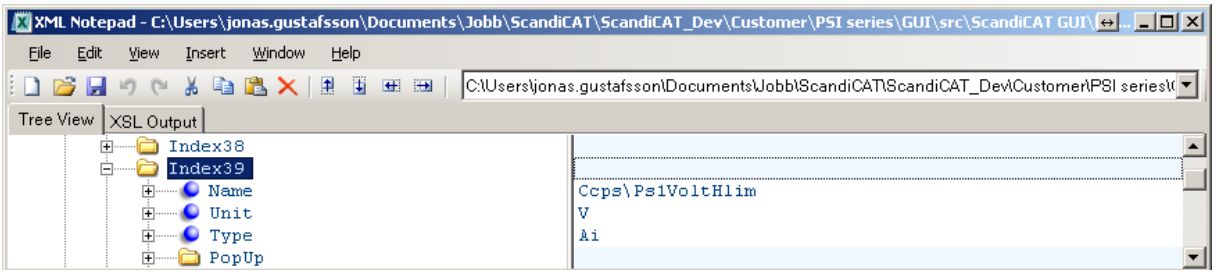
1.13. State read array (Address 2000-2255)

This array is used to get the actual state of each SSM* (Slave State Machine) in the modulator. Each SSM has its own index in this array, from index 0 (at address 2000) to index 255 (at address 2255).

A value from an element in this matrix is the current state for the corresponding SSM. See section 1.6 for how to parse state values.

To get the name of an SSM with a certain index you use the file Resource.xml and look under GUI/MatrixItems.

Example: At Modbus address 2039 you find the current state of the SSM named Ccps\Ps1VoltHlim, since that name is found under GUI/MatrixItems/Index39/Name in Resource.xml.



See also section 1.14 to get more info on how to interpret the values of the state matrixes.

1.14. Target state array

This array is not available in Modbus. It contains the same info as the Matrix page in ScandiCAT GUI. However, it is only changed by ScandiNova so a static copy of it is good enough, since it does not change during normal operation.

There is a tool, ScandiCATMatrixReader, that can be used to get the info from this matrix. For more info see DOC-011292.

* See the Control System Manual for further explanation

2. ModbusTCP registers

Registers mapping						
InputRegisters						
Start Address	**End Address**	**Type**	**Unit**	**General**	**Function**	**Name**
0	0	Int16		Modbus-TCP protocol identification number	Id number of protocol (can be different for different customers)	Variable name: .gp_iModbusProtocolId
1	1	Int16		Modbus-TCP protocol revision number	Rev number of protocol (updated each time this document changes)	Variable name: .gp_iModbusProtocolRev
2	2	Uint16		The previously received Modbus-TCP watchdog value	Copied from the ModbusTcpWatchdog holding register, can be used by the master to monitor this communication	Variable name: SR_SlowCall.IFB_ExtComSts.uiModbusTcpWatchdogPrev
3	3	Int16		The actual state of the modulator	(* See Strings\State in GUI Resource.xml *) This is the actual state of the MSM (Main State Machine), example: StateTarget and StateRead are both "Trig" (13) A solenoid interlock is tripped, this interlock was configured to interlock the "Hv" state (9), therefore StateTarget is immediately set to the target level below which is "StandBy" (5). Now all units immediately go to "StandBy" and report their state to the MSM, when the MSM find that all units are at "StandBy" it sets the iStateRead to "StandBy". At the same time StatusBits.HvInlkExist is set to TRUE Also two events are added to the EventLogg, first the interlock event and then the state event (“StandBy”) Additionally the interlock event is copied to FirstInterlockEvent in case multiple interlocks occur in sequence this first interlock copy will not be overwritten until a reset is received)	Variable name: .g_iStateRead
4	4	Uint16		A word containing 16 status bit's	Bit0: StbInlkExist Bit1: HvInlkExist Bit2: TrigInlkExist Bit3: WarningExist Bit4: OutsideLimits Bit5: Error Bit6: Spare Bit7: Spare Bit8: Spare Bit9: Spare Bit10: Spare Bit11: Spare Bit12: Spare Bit13: Spare Bit14: Spare Bit15: Spare	Variable name: .g_wStatusBits
5	5	Int16		The current access level	See Strings\Message 0-3 in GUI Resource.xml	Variable name: .gp_iAccessLevel
6	7	Single		Remaining time of the delay	Filament warm-up timer	Kly\HeaterDelay1 Variable name: SR_SlowCall.IFB_SlowKly.IFB_Delay[1].rTimeRemaining
8	9	Single		Pulse repetition frequency	Pulse repetition frequency read value	T&i\PrfRead Variable name: SR_FastCall.IFB_FastTi.IFB_PrfRead.atq_rPrfRead

20	20	Int16		Modulator target state	Currently used setvalue	Variable name: .g_iStateTarget
21	22	Single	V	Voltage setvalue of all Ccps	Currently used setvalue	Ccps\VoltSet Variable name: .gp_rVoltSet
23	24	Single	A	Filament current setvalue	Currently used setvalue	Kly\FpsCurrSet1 Variable name: SR_FastCall.IFB_FastKly.IFB_FastFilPS.IFB_CurrSet.ip_rSet
25	26	Single	µs	Pulse width setvalue of all SU's	Currently used setvalue	Switch\PlswthSet Variable name: .gp_rPlswthSet
100	101	Single	V	Voltage read value of Capacitor Charging Power Supply no 1	Scaled readvalue	Ccps\Ps1VoltRead Variable name: SR_FastCall.IFB_Ccps.IFB_CcpsUnit[1].IFB_VoltRead.atq_rRead
120	120	Word		Interlock status bits of Capacitor Charging Power Supply no 1	Bit0 "Mains interlock" Bit1 "PwmPulseCount interlock" Bit2 "IsaqCom interlock" Bit3 "SoftStart interlock" Bit4 "Igbt interlock" Bit5 "PhaseLoss interlock" Bit6 "TransformerTemp interlock" Bit7 "RectifierTemp interlock" Bit8 "IgbtTemp interlock" Bit9 "OverVoltage interlock" Bit10 "OverCurrent interlock" Bit11 "OptoFiberDarkTimeout"	Ccps\Ps1SumSts Variable name: SR_FastCall.IFB_Ccps.IFB_CcpsUnit[1].IFB_DigitalInputWord.wTempWord
200	201	Single	A	Current readvalue of the filament power supply	Scaled readvalue	Kly\FpsCurrRead Variable name: SR_FastCall.IFB_FastKly.IFB_FastFilPs.IFB_CurrRead.atq_rRead
202	203	Single	V	Voltage readvalue of the filament power supply	Scaled readvalue	Variable name: SR_FastCall.IFB_FastKly.IFB_FastFilPs.IFB_VoltRead.atq_rRead
300	301	Single	nA	Current readvalue of the ion pump controller 1	Scaled readvalue	Kly\Ipc1CurrRead Variable name: SR_FastCall.IFB_FastKly.IFB_FastIonPS[1].IFB_CurrRead.atq_rRead
302	303	Single	kV	Voltage readvalue of the ion pump controller 1	Scaled readvalue	Kly\Ipc1VoltRead Variable name: SR_FastCall.IFB_FastKly.IFB_FastIonPS[1].IFB_VoltRead.atq_rRead
400	401	Single	A	Current readvalue of solenoid power supply 1	Scaled readvalue	Kly\Sps1CurrRead Variable name: SR_FastCall.IFB_FastKly.IFB_Sps.IFB_GEN[1].IFB_CurrRead.atq_rRead
402	403	Single	V	Voltage readvalue of solenoid power supply 1	Scaled readvalue	Kly\Sps1VoltRead Variable name: SR_FastCall.IFB_FastKly.IFB_Sps.IFB_GEN[1].IFB_VoltRead.atq_rRead
404	405	Single	A	Current readvalue of solenoid power supply 2	Scaled readvalue	Kly\Sps2CurrRead Variable name: SR_FastCall.IFB_FastKly.IFB_Sp

						s.IFB_GEN[2].IFB_CurrRead.atq_rRead
406	407	Single	V	Voltage readvalue of solenoid power supply 2	Scaled readvalue	Kly\Sps2VoltRead Variable name: SR_FastCall.IFB_FastKly.IFB_Sps.IFB_GEN[2].IFB_VoltRead.atq_rRead
408	409	Single	A	Current readvalue of solenoid power supply 3	Scaled readvalue	Kly\Sps3CurrRead Variable name: SR_FastCall.IFB_FastKly.IFB_Sps.IFB_GEN[3].IFB_CurrRead.atq_rRead
410	411	Single	V	Voltage readvalue of solenoid power supply 3	Scaled readvalue	Kly\Sps3VoltRead Variable name: SR_FastCall.IFB_FastKly.IFB_Sps.IFB_GEN[3].IFB_VoltRead.atq_rRead
500	501	Single	A	Current Transformer readvalue	Scaled read value of the pulse current amplitude	Tank\DigiCtRead Variable name: SR_FastCall.IFB_FastTank.IFB_Digi.IFB_CtRead.atq_rRead
502	503	Single	kV	Capacitive voltage-divider readvalue	Scaled read value of the pulse voltage amplitude	Tank\DigiCvdRead Variable name: SR_FastCall.IFB_FastTank.IFB_Digi.IFB_CvdRead.atq_rRead
504	505	Single	µs	Full Width Half Maximum readvalue	Scaled read value of the pulse width at 50% height of the current pulse	Tank\DigiFwhmRead Variable name: SR_FastCall.IFB_FastTank.IFB_Digi.IFB_FwhmRead.atq_rRead
600	601	Single	ºC	Oil temperature readvalue	Scaled readvalue	Tank\OilTempRead Variable name: SR_SlowCall.IFB_SlowTank.IFB_OilTemperature.atq_rRead
602	603	Single	mm	Oil level readvalue	Scaled readvalue of the oil level (0mm at Klystron min specification)	Tank\OilLevRead Variable name: SR_SlowCall.IFB_SlowTank.IFB_OilLevel.atq_rRead
700	701	Single	ºC	Cool\InletWaterTemp	Scaled readvalue	Cool\InletWaterTempRead Variable name: SR_SlowCall.IFB_Cool.IFB_TempSensors[1].atq_rRead
702	703	Single	ºC	Cool\ColRtnTemp	Scaled readvalue	Cool\ColRtnTempRead Variable name: SR_SlowCall.IFB_Cool.IFB_TempSensors[2].atq_rRead
704	705	Single	ºC	Cool\SolTemp	Scaled readvalue	Cool\BodyRtnTempRead Variable name: SR_SlowCall.IFB_Cool.IFB_TempSensors[3].atq_rRead
820	821	Single	dBm	RF forward power	Scaled readvalue	Kly\RF Fwd Read Variable name: SR_FastCall.IFB_FastKly.IFB_RfOutDigi.IFB_FwdRead.atq_rRead
822	823	Single	dBm	RF reflected power	Scaled readvalue	Kly\RF Rfl Read Variable name:

						SR_FastCall.IFB_FastKly.IFB_RfOutDigi.IFB_RefRead.atq_rRead
824	825	Single		RF VSWR	Scaled readvalue	Kly\RF VSWR Read Variable name: SR_FastCall.IFB_FastKly.IFB_RfOutDigi.atq_rVswrRead
826	827	Single	µs	RF pulse length	Scaled readvalue	Kly\RF Pulse Len Read Variable name: SR_FastCall.IFB_FastKly.IFB_RfOutDigi.IFB_FwhmRead.atq_rRead
900	901	Single	l/m	Cooling water flow rate	Scaled readvalue	Cool\CcpsSuFlow1 Variable name: SR_SlowCall.IFB_Cool.IFB_FlowReads[1].atq_rRead
902	903	Single	l/m	Cooling water flow rate	Scaled readvalue	Cool\CcpsSuFlow2 Variable name: SR_SlowCall.IFB_Cool.IFB_FlowReads[2].atq_rRead
904	905	Single	l/m	Cooling water flow rate	Scaled readvalue	Cool\CcpsSuFlow3 Variable name: SR_SlowCall.IFB_Cool.IFB_FlowReads[3].atq_rRead
908	909	Single	l/m	Cooling water flow rate	Scaled readvalue	Cool\BodWinFlow Variable name: SR_SlowCall.IFB_Cool.IFB_FlowReads[5].atq_rRead
910	911	Single	l/m	Cooling water flow rate	Scaled readvalue	Cool\CollectorFlow Variable name: SR_SlowCall.IFB_Cool.IFB_FlowReads[6].atq_rRead
912	913	Single	l/m	Cooling water flow rate	Scaled readvalue	Cool\SolenoidFlow Variable name: SR_SlowCall.IFB_Cool.IFB_FlowReads[7].atq_rRead
1000	1049	Uint16		Event logg increment array	Incremented for each new event, this enables GUI to see if new events has ocurred	Variable name: .g_aEventsIncr
1050	1099	Uint16		Event logg type array	0=State, 1=Warning, 2=Interlock, 3=Error, 4=Parameter, 5=Message	Variable name: .g_aEventsType
1100	1299	Uint64		Event logg time-stamp array	TYPE T_FILETIME : STRUCT dwLowDateTime : DWORD; dwHighDateTime : DWORD; END_STRUCT END_TYPE The T_FILETIME structure is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC). (Since modbus doesn't support Uint64 you have to read four words per timestamp)	Variable name: .g_aEventsTime
1300	1399	Uint32		Event logg trig id array	The current trig id/count	Variable name: .g_aEventsTrigId
1400	1449	Uint16		Event logg index array	Shows which item that generated this event, see Matrixltems in GUI Resource.xml	Variable name: .g_aEventsIndex
1450	1499	Uint16		Event logg text number array	See GUI Resource.xml\Strings, Type determines sub-element	Variable name: .g_aEventsTextNo
1500	1549	Uint16		Event logg data type array	0=No data, 1=Real, 2=Bool, 3=Int, 4=Uint, 5=Word, 6=Dint, 7=Udint, 8=Dword	Variable name: .g_aEventsDataType

1550	1649	Uint32		Event logg data array	Here data for the event can be entered, if DataType indicates NoData then this value is random	Variable name: .g_aEventsData
1700	1712	Struct		Main event struct	This can be used to present the actual state of the modulator, you see State, Warnings, FirstInterlock and Error messages STRUCT Incr : Uint16; (addr: 1700) Type : Uint16; (addr: 1701) TimeStamp : Uint64; (addr: 1702) TrigId : Uint32; (addr: 1706) Index : Uint16; (addr: 1708) TextNo : Uint16; (addr: 1709) DataType : Uint16; (addr: 1710) Data : Uint32; (addr: 1711) END_STRUCT	Variable name: .g_stMainEvent
1715	1727	Struct		Main event struct	Displays the first interlock event since previous reset, the Type element will indicate active interlock whith the value 2 and inactive/resetted with the value 0 STRUCT Incr : Uint16; (addr: 1715) Type : Uint16; (addr: 1716) TimeStamp : Uint64; (addr: 1717) TrigId : Uint32; (addr: 1721) Index : Uint16; (addr: 1723) TextNo : Uint16; (addr: 1724) DataType : Uint16; (addr: 1725) Data : Uint32; (addr: 1726) END_STRUCT	Variable name: .g_stFirstInterlockEvent
2000	2255	Uint16		State read array	In this array you can see the actual state of all items in the modulator, see Strings\State in GUI Resource.xml	Variable name: .g_aReadMatrixStateRead
2300	2555	Uint16		Status bit's array	In this array you can see the unlatched status bit's of all items in the modulator, Bit0: Warning condition exists Bit1: Interlock condition exists	Variable name: .g_aReadMatrixStatusBits
3000	3000	Int16		Customer waveform sequence	Sequence number reply when a new waveform is received This variable will reply: 99 if fetching did not produce any result 100 when Cvd T-0 is received 101 when Cvd T-1 is received 102 when Cvd T-2 is received 103 when Cvd T-3 is received 104 when Cvd T-4 is received 105 when Cvd saved reference is received 106 when Cvd upper interlock bondary is received 107 when Cvd lower interlock bondary is received	Variable name: SR_FastCall.IFB_FastDigitizer.iCustSeq

					108 when Cvd upper warning bondary is received 109 when Cvd lower warning bondary is received 110 when Ct T-0 is received 111 when Ct T-1 is received 112 when Ct T-2 is received 113 when Ct T-3 is received 114 when Ct T-4 is received 115 when Ct saved reference is received 116 when Ct upper interlock bondary is received 117 when Ct lower interlock bondary is received 118 when Ct upper warning bondary is received 119 when Ct lower warning bondary is received 120 when RfFwd T-0 is received 121 when RfFwd T-1 is received 122 when RfFwd T-2 is received 123 when RfFwd T-3 is received 124 when RfFwd T-4 is received 125 when RfFwd saved reference is received 126 when RfFwd upper interlock bondary is received 127 when RfFwd lower interlock bondary is received 128 when RfFwd upper warning bondary is received 129 when RfFwd lower warning bondary is received 130 when RfRfl T-0 is received 131 when RfRfl T-1 is received 132 when RfRfl T-2 is received 133 when RfRfl T-3 is received 134 when RfRfl T-4 is received 135 when RfRfl saved reference is received 136 when RfRfl upper interlock bondary is received 137 when RfRfl lower interlock bondary is received 138 when RfRfl upper warning bondary is received 139 when RfRfl lower warning bondary is received	
3001	3512	Array of int16		Customer waveform	Byte 0-7 TimeStamp (either in Windows FileTime or in EpicsTime) Byte 8-15 PulseId Byte 16-17 NoOfSamples Byte 18-1023 Waveform	Variable name: SR_FastCall.IFB_FastDigitizer.aCust

OutputRegisters						
Start Addr	**End Addr**	**Type**	**Unit**	**General**	**Function**	**Name**
0	0	Uint16		Communication watchdog	Increment this value at least every second (depending on timeout setting in local GUI), used by the modulator to monitor this communication	Variable name: SR_SlowCall.IFB_ExtComSts.uiModbusTcpWatchdog
1	1	Int16		Modulator target state	This is the target state of the MSM (Main State Machine), example: iStateTarget and iStateRead are both "Off" (1) You request the modulator to "Hv" state by writing 9 to StateTarget See Strings\State in GUI Resource.xml	Variable name: .g_iStateTargetRem
2	2	Uint16		Command bit's Bit0: Reset	A word containing 16 command bit's Bit0: Reset (remember to set it back to zero	Variable name: .g_wCommandBitsRem

				Bit1: ECAT control (use .g_rVoltSetEcat)	afterwards) Bit1: ECAT control (use .g_rVoltSetEcat)	
100	101	Single	V	Voltage setvalue of all Ccps	Scaled setvalue, range 0-1200VDC	Variable name: .g_rVoltSetRem
200	201	Single	A	Filament current setvalue	Scaled setvalue, range 0-30ADC	Variable name: SR_FastCall.IFB_FastKly.IFB_FastFilPS.IFB_CurrSet.ati_rSetRem
300	301	Single	µs	Pulse width setvalue of all SU's	Scaled setvalue, range 1-5µs	Variable name: .g_rPlswthSetRem
3000	3000	Int16		Customer waveform sequence	Sequence number used to fetch waveforms Set this variable to a specific value to start fetching a specific waveform: 0 Cvd T-0 1 Cvd T-1 2 Cvd T-2 3 Cvd T-3 4 Cvd T-4 5 Cvd saved reference 6 Cvd upper interlock bondary 7 Cvd lower interlock bondary 8 Cvd upper warning bondary 9 Cvd lower warning bondary 10 Ct T-0 11 Ct T-1 12 Ct T-2 13 Ct T-3 14 Ct T-4 15 Ct saved reference 16 Ct upper interlock bondary 17 Ct lower interlock bondary 18 Ct upper warning bondary 19 Ct lower warning bondary 20 RfFwd T-0 21 RfFwd T-1 22 RfFwd T-2 23 RfFwd T-3 24 RfFwd T-4 25 RfFwd saved reference 26 RfFwd upper interlock bondary 27 RfFwd lower interlock bondary 28 RfFwd upper warning bondary 29 RfFwd lower warning bondary 30 RfRfl T-0 31 RfRfl T-1 32 RfRfl T-2 33 RfRfl T-3 34 RfRfl T-4 35 RfRfl saved reference 36 RfRfl upper interlock bondary 37 RfRfl lower interlock bondary 38 RfRfl upper warning bondary 39 RfRfl lower warning bondary	Variable name: SR_FastCall.IFB_FastDigitizer.iCustSeq