

STRATON EN / IEC60870-5-101 Driver

for

RTU32

V. 0.01 / May 2006 / Doc 40233



Content

1. Synopsis.....	3
2. Requirements.....	3
Application Structure	4
3. StartUp.....	4
4. IEC101	5
SendSP.....	6
SendDP	6
SendMVN.....	7
deltaIntegrator	7
SingleCMD	8
Output parameters.....	8
DoubleCMD	9
SetPointCMDNV	10

1. Synopsis

The implementation of EN/IEC60870-5-101 driver in STRATON is explained in this document. An example code is used for demonstration purpose. The application runs on Brodersen RTU32.

2. Requirements

Knowledge of IEC870-5-101 protocol and the IEC61131-3 standard is required, for the understanding of this document, and for the implementation of the drivers in STRATON.

The IEC870 drivers are, function-blocks made in C and imported in the STRATON hardware definition folder. The C function-blocks can be used in any STRATON application simply by declaring an instance of them.

Implemented ASDU types in this example are in:

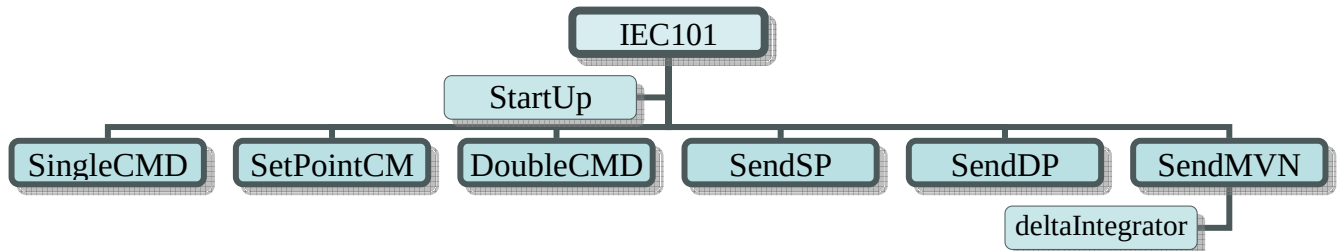
Process information in monitor direction

- <1> := single-point information
- <2> := single-point information with time-tag
- <3> := double-point information
- <4> := double-point information with time-tag
- <9> := measured value, normalised value
- <70> := end of initialisation

Process information in control direction

- <45> := single command
- <46> := double command
- <48> := set-point command normalised value
- <100> := general interrogation command
- <103> := clock synchronisation command

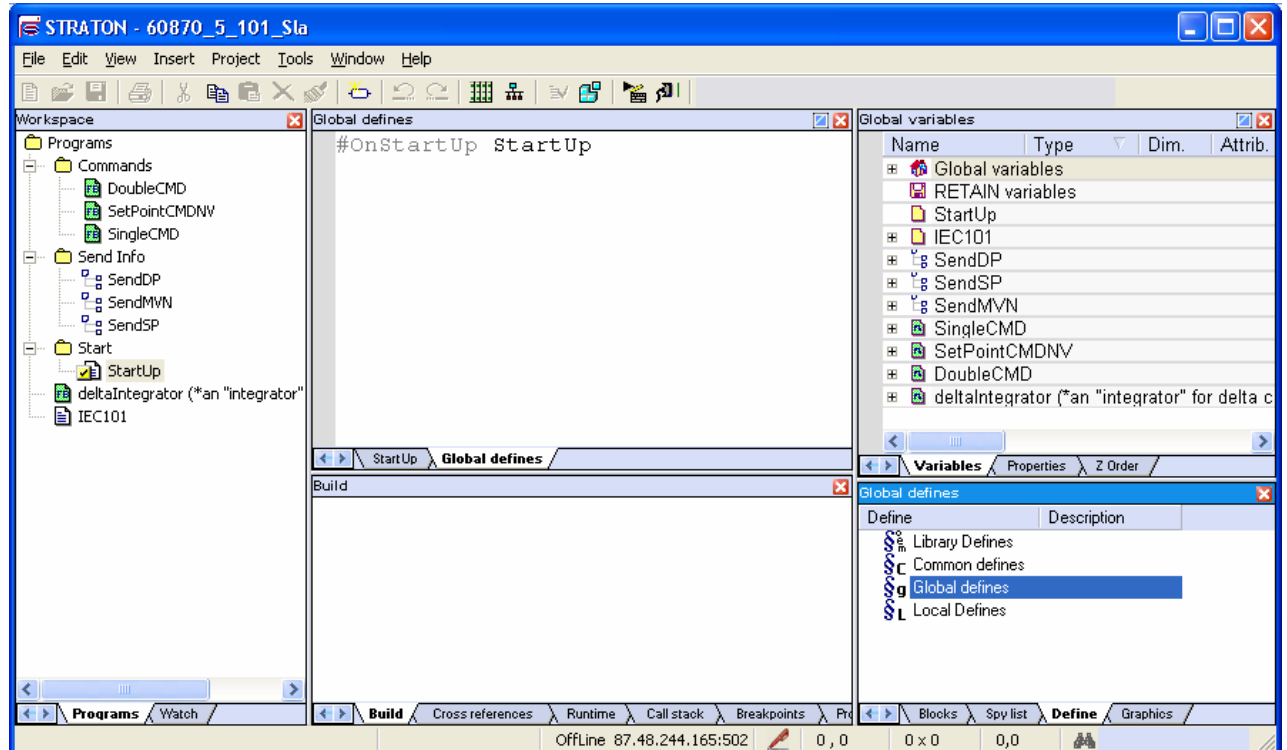
Application Structure



3. StartUp

The StartUp program is made for initialization purpose. It has been defined in STRATON in the "Global Define" manager to run on start up (only once). See below fig. 1.

Fig 1. – Global Define Manager



The "StartUp" program instantiates a handler. The handler can then be used in the whole application when needed.

To indicate that connection is established an "end of initialisation" ASDU is send.

4. IEC101

This is basically the main program. It runs after "StartUp" and calls all the necessary programs and function-blocks.

It sends single and double point information ASDU's spontaneously, when digital inputs are switched on/off, by calling "SendSP" and "SendDP".

By using the "AsduAvailable0()" function-block it checks for incoming commands and reply's. This is done using a case structure, where case number is equivalent to the type ID of the ASDU received from master.

The case structure also checks for COA (Common address of ASDU). If COA matches, or COA is equal to 255 (broadcast address), the program will proceed.

The commands implemented in this example code are 45, 46, 48, 100 and 103. Most likely there will in future updates be implemented more commands.

The program tests if pulses are still active and sets the outputs with the currently received values, by using the instances of "SingleCMD" and "DoubleCMD".

By evaluating the status of the physical outputs, the IEC101 will report any changes using "SendSP" and "SendDP".

An instance of "deltaIntegrator" is used to report any changes made on the analogue physical input greater than 5%.

SendSP

Input parameters

Handle	: DINT ; (Handler instantiated in "StartUp")
IOA	: DINT ; (Information Object Address)
COT	: DINT ; (Cause Of Transmission)
Value	: BOOL ; (the current value that should be sent in the ASDU (in this case physical input))
Trigger	: BOOL ; (trigger)
TimeTag	: BOOL ; (If true, time-tag is selected)

The program sends a single point information when called by using the "PutSglPoint()" function block. It also checks if time-tag is requested, and if so, implements the time-tag in the send single point information ASDU.

SendDP

Input parameters:

Handle	: DINT ; (Handler instantiated in "StartUp")
IOA	: DINT ; (Information Object Address)
COT	: DINT ; (Cause Of Transmission)
Value	: BOOL; (the current value that should be sent in the ASDU (in this case physical input))
Value1	: BOOL; (the current value that should be sent in the ASDU (in this case physical input))
Trigger	: BOOL; (trigger)
TimeTag	: BOOL; (If true, time-tag is selected)

The program sends a double point information when called by using the "PutDblPoint()" function block. It also checks if time-tag is requested, and if so, implements the time-tag in the send single point information ASDU.

SendMVN

Input parameters:

Handle	: DINT ; (Handler instantiated in "StartUp")
IOA	: DINT ; (Information Object Address)
COT	: DINT ; (Cause Of Transmission)
NVA	: DINT ; (the normalized analogue value (physical input))

The program sends the normalised measured value when called. No time-tag is implemented.

deltaIntegrator

Input parameters

deactivate	: BOOL ;	(turn off the function)
curW	: INT ;	(current analogue value)
sampleMillSec	: DINT ;	(time between samples)
threshold	: INT ;	(threshold in promille)

Output parameters

iecSend	: BOOL ;	(to indicate whether to send or not)
toSend	: INT ;	(sends the analogue)
debugSum	: REAL ;	(Only for debug purposes)

The "deltaIntegrator" is a "User Defined Function Block".

It is used in this context, for reporting any changes made on the analogue input. The "threshold" variable specifies how the deviation of the analogue input before it's reported.

SingleCMD

Input parameters

SCO	: SINT ;	(single command value received from master)
newValue	: BOOL ;	(Trigger)
ASDUx	: ARRAY	[0 .. 254] OF SINT ; (ASDU received from Master)
ASDULenx	: SINT ;	(Length of the ASDU)
ScHandle	: DINT ;	(Handler instantiated in "StartUp")
SE	: BOOL ;	(Select and execution)

Output parameters

Value : BOOL ; (Single Command value)

This "User Defined Function Block" is designed to receive single commands from the master and set the physical outputs. In this case output bit 0 and 1 are dedicated single command output bits.

To define whether to use S/E (select and execution), the "SE" parameter is set to true or false.

In this particular example S/E is set to true.

When using S/E, a single command ASDU must be received from the master, with the selection bit switched to 1, indicating a selection. Within 10 sec. the same single command ASDU must be received, with the selection bit switched to 0, indicating an execution. If not, the ASDU will be discarded. This will guaranty, that the single command is sent to the correct node (RTU), and that the ASDU is not too old.

The SingleCMD has also implemented an internal timer, for setting the outputs. This timer can be defined to set the outputs true for: 500ms, 5000ms or permanently.

DoubleCMD

Input parameters

SCO	: SINT ;	(double command value received from master)
newValue	: BOOL ;	(Trigger)
ASDUx	: ARRAY [0 .. 254] OF SINT ;	(ASDU received from master)
ASDULenx	: SINT ;	(Length of the ASDU)
ScHandle	: DINT ;	(Handler instantiated in "StartUp")
SE	: BOOL ;	(Select and execution)

Output parameters

Value	: BOOL ;	(1. value of Double Command)
Value1	: BOOL ;	(2. value of Double Command)

This "User Defined Function Block" is designed to receive double commands from the master and set the physical outputs. In this case output bits 2-7 dedicated double command output bits.

To define whether to use S/E (select and execution), the "SE" parameter is set to true or false.

In this particular example S/E is set to true.

When using S/E, a double command ASDU must be received from the master, with the selection bit switched to 1, indicating a selection. Within 10 sec. the same double command ASDU must be received, with the selection bit switched to 0, indicating an execution. If not, the ASDU will be discarded. This will guaranty, that the double command is sent to the correct node (RTU), and that the ASDU is not too old.

The DoubleCMD has also implemented an internal timer, for setting the outputs. This timer can be defined to set the outputs true for: 500ms, 5000ms or permanently.

SetPointCMDNV

Input parameters

SCO	: SINT ;	(the normalised value received from master)
NewValue	: BOOL ;	(trigger)
ASDUX	: ARRAY [0 .. 254] OF SINT ;	(ASDU received from master)
ASDULenx	: SINT ;	(length of the ASDU received from master)
ScHandle	: DINT ;	(handler instantiated in "StartUp")
SE	: BOOL ;	(Select and execute)

Output parameters

Value	: DINT ;	(the normalised value)
SetAnalogue	: BOOL ;	(this variable is used to ensure no output is changed during selection)

This function block sets a physical analogue output, to a normalised value.

To define whether to use S/E (select and execution), the "SE" parameter is set to true or false.

In this particular example S/E is set to true.

When using S/E, a set-point ASDU must be received from the master, with the selection bit switched to 1, indicating a selection. Within 10 sec. the same SCO ASDU must be received, with the selection bit switched to 0, indicating an execution. If not, the ASDU will be discarded. This will guaranty, that the set-point is sent to the correct node (RTU), and that the ASDU is not too old.