Concept Version 2.1 – 2.5

IEC Function Blocks

User's Guide

Monaghan Engineering, Inc.
862 Las Colinas Drive
Dripping Springs, Texas 78620
Phone: 1-512-858-4271
Fax: 1-512-858-1355
e-mail: techsupport@monaghan-engineering.com

Publication #10012
Version 3.01

TABLE OF CONTENTS

1.0 SER SYSTEM OVERVIEW

The Sequence of Events Recorder (SER) provides a permanent record of events that occur within milliseconds of each other, such as the operation of circuit breakers or the shutdown sequences of compressors and other high speed devices. The SER will provide the time of the event, the state of the point, and the point identification.

Multiple SER cards may be synchronized together to provide a distributed event recording system. Interface cards are available to synchronize the SER cards with an external time standard such as a Global Positioning Satellite (GPS) receiver.

1.1 SER FUNCTION BLOCKS

The SER function block provides an easy means of both configuring and gathering data from the SER cards. When the controller is reset, the function block will configure the SER cards and scan for event data. When an event is detected, the function block will build a buffer containing the event data and either send it over Modbus Plus to a receiving drop, or hold the buffer until a host computer polls the PLC for the data.

2.0 INSTALLATION

In order for Concept to recognize the IEC library for the SER cards the library files must be placed in the proper subdirectories and the Concept library database must be rebuilt.

In addition to loading the IEC library files, the Concept I/O card database must also be updated to include support for the SER x53 00 and the GPS 100 00 card.

2.1 LIBRARY FILES

The IEC library files should be placed in the following subdirectories:

c:\concept\lib\mei\cod\dos16\mei.cod
c:\concept\lib\mei\cod\dos16\mei.x0
c:\concept\lib\mei\cod\dos16\mei.x1
c:\concept\lib\mei\cod\dos16\mei.x2
c:\concept\lib\mei\cod\dos32\mei.cod
c:\concept\lib\mei\cod\dos32\mei.x0
c:\concept\lib\mei\cod\dos32\mei.x1
c:\concept\lib\mei\cod\dos32\mei.x2
c:\concept\lib\mei\tpl\mei.dty
c:\concept\lib\mei\tpl\mei.p1
c:\concept\lib\mei\tpl\mei.p2
c:\concept\lib\mei\tpl\mei.p3
c:\concept\lib\mei\tpl\mei.p4
c:\concept\lib\mei\tpl\mei.q1
c:\concept\lib\mei\tpl\mei.q2
c:\concept\lib\mei\tpl\mei.tpl

After copying the library files to the proper subdirectories delete the following database files:

c:\concept\lib\~aitmp.p1
c:\concept\lib\~aitmp.p2
c:\concept\lib\~aitmp.p3
c:\concept\lib\~aitmp.p4
c:\concept\lib\~aitmp.q1
c:\concept\lib\~aitmp.q2

When Concept starts it will rebuild the library database and the MEI library should be available.

2.2 MODULE DATABASE

Concept 2.1 includes the utility, MODCONF.EXE, which is used to add new I/O cards to the module database. Copy the file mneng.mdc to the Concept subdirectory. Run the MODCONF.EXE program and open the mneng.mdc file to add support for the SER x53 00 and GPS 100 00 cards.

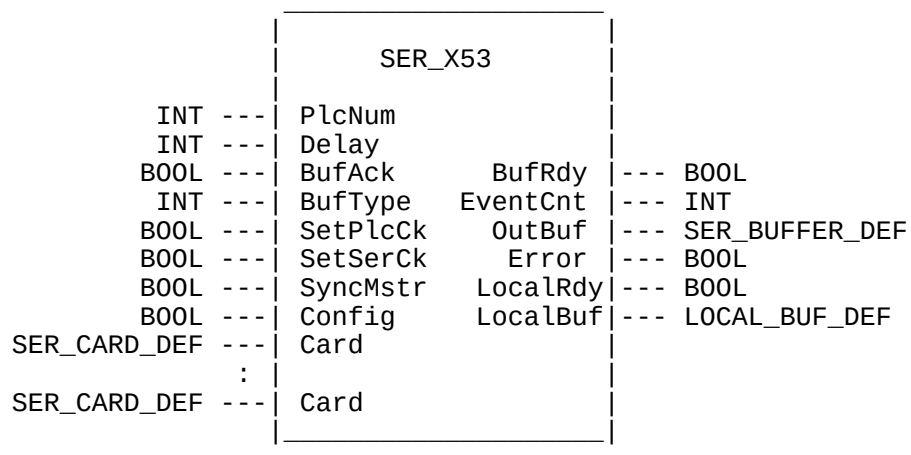
3.0 SER FUNCTION BLOCK

The SER Function block is an extendable block that performs three separate functions. Only one block is required in each PLC and the block will support up to 23 SER cards.

For every SER module mounted in the PLC, a variable of the type SER_CARD_DEF is required. This variable is used to define all of the configuration parameters for the card. The SER block will use this information to download configuration data to the card and to retrieve event data from the card.

The SER block is also responsible for managing a common buffer into which events are placed. When an event buffer is ready to be read, the SER function block will turn on the BufRdy output indicating that data is ready. This output can either be polled by master computer and the data gathered when the SER block indicates a buffer is ready, or the output may be connected to a MSTR block which will automatically send the event data over Modbus Plus to a BM85E Bridge Mux, or a PC with a SA85 card. After the data has been collected, the BufAck input is used to tell the SER block to clear the buffer and to start collecting new event data.

The SER function block will also set the PLC time of day clock from the time in the SER card, or set the time in the SER card from the PLC time of day clock.



3.1 SER FUNCTION BLOCK INPUTS

PlcNum
The value of the PlcNum input is placed in the output buffer to indicate which PLC the buffer is from.

Delay
The value of the Delay input is the number of 10mS intervals the block will wait from the last event until the BufRdy output is set true.

BufAck
A true value on the BufAck input will clear the event buffer and set the BufRdy output false.

BufType
The value of BufType is used to determine the structure of the output buffer. Currently only a value of zero is supported. This will generate a buffer in the format described later in this document. Future versions of this function block will support multiple buffer definitions.

SetPlcCk
A true value to the SetPlcCk input will set the PLC Time of Day clock from the first SER card.

SetSerCk
A true value on the SetSerCk input will set the first SER cards clock from the PLC Time of Day clock.

SyncMstr
A true value on the SyncMstr input will configure the first SER card as a time sync master.

Config
A true on the Config input will download the configuration data to all of the cards.

Card
The block can be expanded to accept up to 23 Card inputs, which expect a data type of the SER_CARD_DEF type.

```
TYPE
      SER_CARD_DEF: STRUCT
            ADDRESS0X:              INT;
            ADDRESS1X:              INT;
            TimeQuality:            INT;
            EventCount              INT;
            FirmwareVersion:        INT;
            CardError:              INT;
            Filter:                 ARRAY [0..31] OF UINT;
            Debounce:               ARRAY [0..31] OF UINT;
            Chatter:                ARRAY [0..31] OF UINT;
            OffScan:                ARRAY [0..31] OF BOOL;
            Enable:                 BOOL;
            Config1:                BOOL;
            ConfigPor:              BOOL;
            FilterInput:            BOOL;
            BatteryEnable:          BOOL;
            SyncLock:               BOOL;
      END_STRUCT;
```

Address0X - Address of First 0x Input in I/O Map
The SERX53 function block must write the commands to the SER module through 0x registers. . The number entered here is an offset from 000000. Register 000001 is addressed a 1. Please note that the address entered here must match a 0x address as entered in the PLC's I/O map for the SER module.

Address1X - Address of First 1x Input in I/O Map
The SERX53 function block must read the values from the SER module through 1x inputs. The number entered here is an offset from 100000. Register 100001 is addressed a 1. Please note that the address entered here must match a 1x address as entered in the PLC's I/O map for the SER module.

TimeQuality – Current Time Quality – READ ONLY
This register contains the current quality of the SER time stamps. The values are as follows:

Value	Quality
0	Good, accuracy is GMT +/- 1ms
1	Fair, accuracy is GMT +/- 50ms
2	Poor, accuracy is GMT +/- 50ms or worse
3	Bad, no time reference available

EventCount– SER Card Event Count
The number of events currently in the SER card buffer.

FirmwareVersion – Firmware Version of the SER Card – READ ONLY
This register contains the version of the firmware in the SER card.

CardError – Error Status of Card – READ ONLY
This register contains the SER Card error. Its primary purpose is to aid debugging during the programming and testing of the program. Whenever a bit is turned on, a certain error condition is present.

Bit	Error Condition
15	The 0x address entered in the function block is outside the valid 0x addresses as configured in the PLC's configuration. For example, if the inputs 10001 through 10512 are configured, the highest valid 1x address would be 10449, since the SER module requires 64 1x inputs.
14	The 0x address entered in the function block is outside the valid 0x addresses as configured in the PLC's configuration. For example, if the outputs 00001 through 01536 are configured, the highest valid 0x address would be 01473, since the SER module requires 64 0x outputs.
8	The Time of Day Clock must be configured.

Filter – SER Card Input Filter
The filter value for each input point is contained in an array of 32 unsigned integers. The purpose of the contact filter time is to eliminate false event messages caused by noise. The filter time is the amount of time that a point must stay in a new state in order to be recognized as an event. As an example, if the filter time for a point is set to 25 milliseconds, then when the input changes state it must remain in the new state for 25 milliseconds before and event message is generated. If the input returns to the previous state in less than 25 milliseconds, the filter is reset and the next time the state changes it will again have to remain in that state for 25 milliseconds before an event message is generated. The time that is associated with the event is the time at which the point first changed, not the time when the filter recognized the event. If two input points that different input filter values were to change at the same time, the event messages would be generated at different times but the time that was contained in the event message would be identical.

Debounce – SER Card Input Debounce
The debounce value for each input point is contained in an array of 32 unsigned integers. The purpose of the contact debounce time is to prevent multiple events from being generated from a single contact closure. The debounce time is the amount of time that input processing is disabled for a point after an event has been recorded

Chatter – SER Card Input Chatter
The debounce value for each input point is contained in an array of 32 unsigned integers. This parameter determines the maximum number of status changes allowed per minute before the point

is taken off scan. Once a point is taken off scan, it will be put on scan again after being at least one minute below the chatter count.

OffScan – SER Point Scan State
The offscan value for each input point is contained in an array of 32 Boolean values. If the offscan value is TRUE the SER card will not process SER data for that point.

Enable – SER Card Enable
A true value will enable the SER card.

Config1 – Configure All Points The Same As Point 1
For a FALSE value, all 32 points are configured with the parameters entered individually for every point. However, if all 32 points operate with the same parameters, these parameters need to be entered only once for point 1. For a TRUE value, the parameters for point 1 are used to configure all 32 points.

ConfigPor – Configure Card After Power On Reset
Since the SER module cannot preserve any parameterizations through a power cycle, all point parameters are reset to their default values after a power cycle. If this value is TRUE, the function block will reconfigure the SER card after a Power On Reset.

FilterInput – Use Input Filter On 1X Registers
For a FALSE value, the 1X register input points will be passed to the processor immediately when they change. For a TRUE value, the filter value for each point will be used to filter the 1X register inputs before they are passed to the processor.

BatteryEnable – Enable the Battery Backed Up Memory
For a FALSE value, the card firmware will not save event data when the card is powered down. For a TRUE value, the card firmware will retain old event data and generate a restart event when the card receives power.

SyncLock – Time Signal Synchronization – READ ONLY
This register contains the current sync status for the SER card. For a TRUE value, the card is receiving time sync messages. For a FALSE value, the card is not receiving time sync messages.

3.2 SER FUNCTION BLOCK OUTPUTS

BufRdy
The BufRdy output is set true when an event buffer is ready to be read.

EventCnt
The EventCnt output contains the number of events in the buffer.

OutBuf
The OutBuf output expects a data type of the SER_BUFFER_DEF type which is defined as follows:

```
TYPE
      SER_BUFFER_DEF: STRUCT
            PlcNumber:              INT;
            BufferType:             INT;
            NumberOfEvents:         INT;
            Nul1:                   INT;
            Nul2:                   INT;
            Nul3:                   INT;
            Nul4:                   INT;
```

```
            Nul5:                   INT;
            Nul6:                   INT;
            Version:                INT;
            Events:                 ARRAY [0..89] OF UINT;
      END_STRUCT;
```

Error
The Error output is set true when there is a configuration error.

LocalRdy
The LocalRdy output is set true for one scan cycle time when a new event is placed in the Local Buffer.

LocalBuf
The LocalBuf output expects a buffer of the LOCAL_BUF_DEF type, which is defined as follows:

```
TYPE
      LOCAL_BUF_DEF: STRUCT
            EventType:              INT;
            Card:                   INT;
            Point:                  INT;
            DateTime:               UDINT;
            Milliseconds:           INT;
            Quality:                INT:
            CurrentState:           BOOL;
      END_STRUCT;
```

The local buffer provides a second port for retrieving event data. The event data is only guaranteed to be valid for one scan cycle time. This port is used by other function blocks that need access to event data.

3.3 EVENT BUFFER STRUCTURE

The event buffer consists of two parts, the header and the buffer proper. The header is located in the first 10 registers. The first register contains the PLC identification, the second register contains the type of event buffer and the third register contains the number of events following. The tenth register contains the SER_x53 function block version number. The other six registers are presently unused.

3.3.1 EVENT BUFFER TYPE 0

The events are stored in an array of registers immediately following the header. The first event will be stored in registers 11 through 13, the second event in registers 14-16, and the nth event in registers (n-1)*3+10 through (n-1)*3+12. Up to 30 events can be stored in a single buffer. The three registers per event are used as follows:

Register	Bits	Usage
1	11-15	SER Card Identification (0..31) (from SER Card Function Block)
1	10	Current Status Value (0 or 1, from SER module)
1	5-9	Status Identifier (0..31, from SER module)
1	0-4	Event Type
		1: Status Change
		2: Point On Scan
		3: Point Off Scan
		4: Chatter On Scan
		5: Chatter Off Scan
		6: Power On Reset
		7: External Time Sync Lock
		8: External Time Sync Lost
		9: SER Card Event Buffer Overflow (from SER module)
		10: Scan Buffer Overflow
		11: Time Resync Old Time
		12: Time Resync New Time
		13: Hourly Time Update
		14: Time Resync New Date
		15: Reconfigure
		16: Output Point Change
		17: Restart Date
		18: Restart Time
		Events 1 – 12, 16, 18
2	10-15	Seconds (0..59, from SER module)
2	0-9	Milliseconds (0..999, from SER module)
		Events 13 – 15, 17
2	9-13	Hour (0..23)
2	4-8	Day (1..31)
2	0-3	Month (1..12)
		Events 1 – 12, 16, 18
	14-15	Time Quality
3	8-12	Hours (0..23, from SER module)
3	0-5	Minutes (0..59, from SER module)
		Events 13 – 15, 17
3	14-15	Time Quality
3	0-12	Year (0..4095)

Example:

Offset	Value	Description
0	23	Message originated in PLC 23
1	x	unused
2	2	Message contains two events
3	x	unused
4	x	unused
5	x	unused
6	x	unused
7	x	unused
8	x	unused
9	100	Loadable version 1.00
10	15873 = 00111 1 10000 00001	Identifier of first event SER card 7 Current status 1 Status point 16 Event type 1 (status change)
11	39228 = 100110 0100111100	Second/milliseconds of event 38 seconds/316 milliseconds
12	4399 = 000 10001 00 101111	Hours/minutes of event 17 hours/47 minutes => Status point 16 on SER card 7 of PLC 23 changed to value 1 at 17:47:38.316
13	15873 = 00101 0 00000 01001	Identifier of second event SER card 5 Current status not used Status point not used Event type 9 (SER buffer overflow)
14	39228 = 100110 0101110010	Second/milliseconds of event 38 seconds/370 milliseconds
15	4399 = 000 10001 00 101111	Hours/minutes of event 17 hours/47 minutes => Buffer overflow on SER card 5 of PLC 23 at 17:47:38.370

3.3.2 EVENT BUFFER TYPE 1

The events are stored in an array of registers immediately following the header. The buffer will only contain a single event and each data value will be placed in a separate register.

Register	Usage
1	Event Type
2	Point
3	Current State
4	Card
5	Millisecond
6	Second
7	Minute
8	Hour
9	Day
10	Month
11	Year
12	Quality

3.3.3 EVENT BUFFER TYPE 2

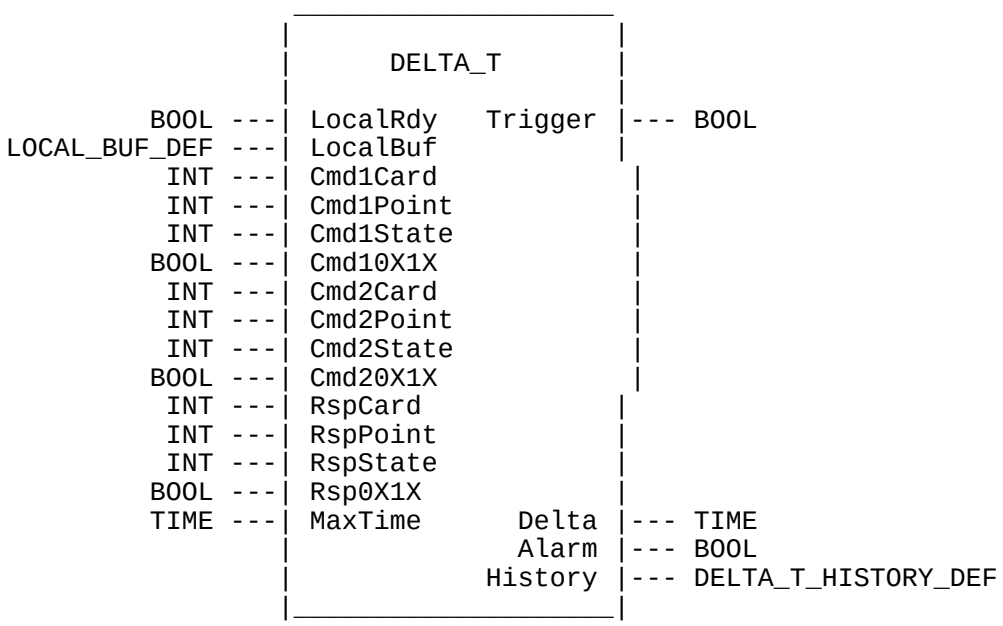
The events are stored in an array of registers immediately following the header. Each event takes four registers. The first event will be stored in registers 11 through 14, the second event in registers 15-18, and the nth event in registers (n-1)*4+10 through (n-1)*4+13. Up to 22 events can be stored in a single buffer. The four registers per event are used as follows:

Register	Bits	Usage
1	11-15	SER Card Identification (0..31) (from SER Card Function Block)
1	10	Current Status Value (0 or 1, from SER module)
1	5-9	Status Identifier (0..31, from SER module)
1	0-4	Event Type 1: Status Change 2: Point On Scan 3: Point Off Scan 4: Chatter On Scan 5: Chatter Off Scan 6: Power On Reset 7: External Time Sync Lock 8: External Time Sync Lost 9: SER Card Event Buffer Overflow (from SER module) 10: Scan Buffer Overflow 11: Time Resync Old Time 12: Time Resync New Time 13: Hourly Time Update 14: Time Resync New Date 15: Reconfigure 16: Output Point Change 17: Restart Date 18: Restart Time
2	0-9	Milliseconds (0..999, from SER module)
2	14-15	Time Quality
3	0-15	Date Time LSB (32 Bit Long Integer. Number of seconds since 1/1/1984)
4	0-15	Date Time MSB (32 Bit Long Integer. Number of seconds since 1/1/1984)

4.0 DELTA TIME FUNCTION BLOCK

The DELTA T Function Block is used to calculate the time difference between two events and set an alarm if the time exceeds a specified value.

The block will monitor the event stream from the local buffer of the SERX53 block. When the DELTA_T block detects an event that matches the Cmd1Card, Cmd1Point, Cmd1State and Cmd10X1X inputs, or the Cm21Card, Cmd2Point, Cmd2State and Cmd20X1X inputs, the trigger output will turn on, the Delta output will be set to zero and an internal timer will be started. When the block detects an event that matches the RspCard, RspPoint, RspState and Rsp0X1X inputs, the trigger output will be turned off, the time difference between the command and response events will be calculated and placed in the Delta output and if the time exceeds the MaxTime input, the Alarm output will be turned on. If the block does not receive a response event, when the internal timer exceeds the MaxTime input, the Alarm output will be turned on and the Trigger output will remain on.



4.1 DELTA_T FUNCTION BLOCK INPUTS

LocalRdy
The LocalRdy input is set true for one scan cycle time when a new event is available in the Local Buffer.

LocalBuf
The LocalBuf input expects a buffer of the LOCAL_BUF_DEF type, which is defined as follows:

```
TYPE
      LOCAL_BUF_DEF: STRUCT
            EventType:              INT;
            Card:                   INT;
            Point:                  INT;
            DateTime:               UDINT;
            Milliseconds:           INT;
            Quality:                INT:
            CurrentState:           BOOL;
      END_STRUCT;
```

Cmd1Card
This is the card number that the block will look for to trigger the time measurement. (0 – 22)

Cmd1Point
This is the point number that the block will look for to trigger the time measurement. (1- 32)

Cmd1Sate
This is the state that the block will look for to trigger the time measurement.

Cmd10X1X
A zero on this input will force the block to look for an event type 16, Output Change Event. A one on this input will force the block to look for an event type 1, Input Change Event.

Cmd2Card
This is the card number that the block will look for to trigger the time measurement. (0 – 22)

Cmd2Point
This is the point number that the block will look for to trigger the time measurement. (1- 32)

Cmd2Sate
This is the state that the block will look for to trigger the time measurement.

Cmd20X1X
A zero on this input will force the block to look for an event type 16, Output Change Event. A one on this input will force the block to look for an event type 1, Input Change Event.

RspCard
This is the card number that the block will look for to complete the time measurement. (0 – 22)

RspPoint
This is the point number that the block will look for to complete the time measurement. (1 – 32)

RspSate
This is the state that the block will look for to complete the time measurement.

Rsp0X1X
A zero on this input will force the block to look for an event type 16, Output Change Event. A one on this input will force the block to look for an event type 1, Input Change Event.

MaxTime
The maximum time allowed between events before the alarm output is turned on.

4.2 DELTA_T FUNCTION BLOCK OUTPUTS

Trigger
The Trigger output is set true when the command event is detected.

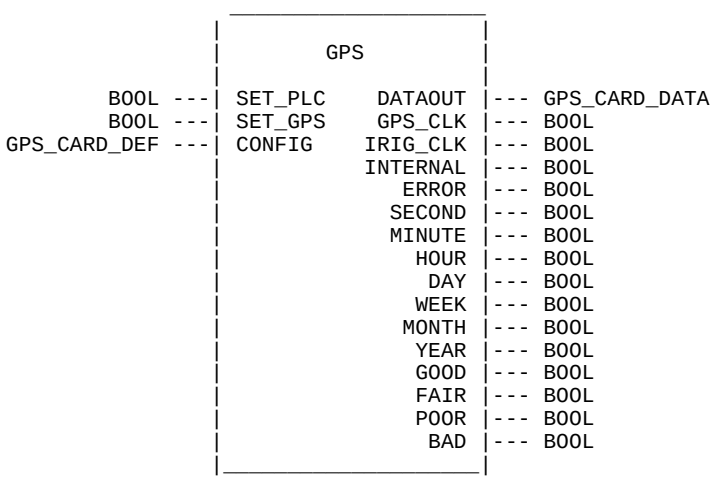
Delta
The time difference between the command event and the response event.

Alarm
The Alarm output is set true if the time difference between the command event and the response event exceeds the MaxTime input.

History
A structure of the DELTA_T_HISTORY_DEF type that contains the last 16 Delta Time measurements and an average of the last 16 measurements.

```
TYPE
      DELTA_T_HISTORY_DEF: STRUCT
            History:                ARRAY [0..15] OF TIME;
            Average:                TIME;
      END_STRUCT;
```

5.0 GPS FUNCTION BLOCK



The GPS function block performs 3 different functions:

1. When SET_PLC input to the block is active the function block will transfer the GPS 100 00 time to the PLC time of day clock.

2. When SET_GPS input to the block is active or when the GPS function block detects that the GPS 100 00 card has gone through a power on reset sequence the function block will transfer the PLC time of day clock to the GPS 100 00 card.

3. The GPS function block continuously updates the time registers, GPS receiver status registers and the GPS position and velocity registers.

5.1 GPS FUNCTION BLOCK INPUTS

SET_PLC
The SET_PLC input is a Boolean input. A true value on the SET_PLC input will cause the PLC time of day clock to be set from the GPS 100 card.

SET_GPS
The SET_GPS input is a Boolean input. A true value on the SET_GPS input will cause the GPS 100 card time to be set from the PLC time of day clock.

CONFIG
The CONFIG input expects a data type of GPS_CARD_DEF which is defined as follows:

```
typedef struct G_CARD_DEF
       {
       int    ADDRESS0X;
       int    ADDRESS3X;
       int    ADDRESS4X;
       int    GPS_IO_OPT;
       };
```

ADDRESS0X - Address of First 0x Output
The GPS function block will activate a group of 0X registers on a timed basis for easy use in ladder logic programming. The starting register for this group is defined here. The starting 0X register must be the start of a 16 bit word ((16 * n) +1). Acceptable values are starting resisters

such as 1, 17, 33, 49, 65 …. If a zero is entered as the starting register, the function block will disable this feature.

ADDRESS3X - Address of First 3x Input in I/O Map
The GPS function block must read the values from the GPS module through 3x inputs. The number entered here is an offset from 300000. Register 300001 is addressed a 1. Please note that the address entered here must match a 3x address as entered in the PLC's I/O map for the GPS module.

ADDRESS4X - Address of First 4x Input in I/O Map
The GPS function block must write the commands to the GPS module through 4x registers. . The number entered here is an offset from 400000. Register 400001 is addressed a 1. Please note that the address entered here must match a 4x address as entered in the PLC's I/O map for the GPS module.

GPS_IO_OPT – Currently not used.

5.2 GPS FUNCTION BLOCK OUTPUTS

DATAOUT
The DATAOUT output expects a data type of GPS_CARD_DATA which is defined as follows:

```
typedef struct GPS_CARD_DATA
       {
       int    EfbError;
       int    DayOfWeek;
       int    Month;
       int    Day;
       int    Year;
       int    Hour;
       int    Minute;
       int    Second;
       int    Millisecond;
       int    LongYear;
       int    EfbVersion;
       int    FirmwareVersion;
       int    GpsStatus;
       int    NumSats;
       float  Latitude;
       float  Longitude;
       float  Altitude;
       float  EastVelocity;
       float  NorthVelocity;
       float  UpVelocity;
       int    TimeQuality;
       };
```

EfbError - GPS Function Block Error Status
This register contains the GPS function block error status. Its primary purpose is to aid debugging during the programming and testing of the ladder logic program. Whenever a bit is turned on, a certain error condition is present.

Bit	Error Condition
15	The time of day clock has not been configured in the PLC's configuration.
14	An invalid 0X register has been entered.
13	An invalid 4X register has been entered.
12	An invalid 3X register has been entered.

DayOfWeek - Day of Week
1 = Sunday, 2 = Monday, … 7 = Saturday

Month - Month of Year
The current month of the year.

Day - Day of Month
The day of the month

Year – Year
The current year. (0 – 99)

Hour – Hour
The current hour.

Minute – Minute
The current minute.

Second – Second
The current second.

Millisecond – Millisecond
The current millisecond.

LongYear – Year
The current year as a 16 bit integer. (1984 – 65536)

EfbVersion – EFB Function Block Version Number
This register contains the GPS function block version number.

FirmwareVersion - GPS 100 00 Card Firmware Version Number
This register contains the GPS 100 00 card firmware version number.

GpsStatus - GPS Status
The GPS status register contains information about the satellite tracking status of the GPS receiver.
0 = Fix not available
1 = Non-differential GPS fix available
2 = Differential GPS fix available

NumSats – Number of Satellites
The number of satellites currently being tracked by the GPS receiver.

Latitude - Latitude Position (Single Precision)
The current latitude in +/- ddmm.mmmm format. Positive being North latitude and negative being South latitude.

Longitude – Longitude Position (Single Precision)
The current longitude in +/- dddmm.mmmm format. Positive being East longitude and negative being West longitude.

Altitude - Altitude Position (Single Precision)
The antenna height above or below mean sea level, -9999.9 to 99999.9 meters.

EastVelocity - East Velocity (Single Precision)
Not supported by the GPS 300 00 receiver.

NorthVelocity - North Velocity (Single Precision)
Not Supported by the GPS 300 00 receiver.

UpVelocity - Up Velocity (Single Precision)
Not Supported by the GPS 300 00 receiver.

TimeQuality - Time Quality
The current time quality.
0 = Good
1 = Fair
2 = Poor
3 = Bad.

GPS_CLK
The GPS_CLK output is of the type Boolean. It will be true when the GPS 100 card is using the GPS receiver as the time source.

IRIG_CLK
The IRIG_CLK output is of the type Boolean. It will be true when the GPS 100 card is using the IRIG-B time code signal as the time source.

INTERNAL
The INTERNAL output is of the type Boolean. It will be true when the GPS 100 card is using its crystal oscillator as the time source.

ERROR
The ERROR output is true when an error condition is present. The error code in the GPS_CARD_DATA.EFB_ERROR variable will define the error.

SECOND
The SECOND output is of the type Boolean. It will be true for one scan cycle time at the beginning of each second.

MINUTE
The MINUTE output is of the type Boolean. It will be true for one scan cycle time at the beginning of each minute.

HOUR
The HOUR output is of the type Boolean. It will be true for one scan cycle time at the beginning of each hour.

DAY
The DAY output is of the type Boolean. It will be true for one scan cycle time at the beginning of each day.

WEEK
The WEEK output is of the type Boolean. It will be true for one scan cycle time at the beginning of each week.

MONTH
The MONTH output is of the type Boolean. It will be true for one scan cycle time at the beginning of each month.

YEAR
The YEAR output is of the type Boolean. It will be true for one scan cycle time at the beginning of each year.

GOOD
The GOOD output is of the type Boolean. It will be true when the time quality is good.

FAIR
The FAIR output is of the type Boolean. It will be true when the time quality is fair.

POOR
The POOR output is of the type Boolean. It will be true when the time quality is poor.

BAD
The BAD output is of the type Boolean. It will be true when the time quality is bad.

5.3 GPS FUNCTION BLOCK 0X DATA AREA

The GPS function block will output status information to a group of 0X registers for easy use in ladder logic. The starting register 0X is defined in the function block configuration.

One Pulse per Second
Offset: +0
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every second.

One Pulse per Minute
Offset: +1
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every minute.

One Pulse per Hour
Offset: +2
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every hour.

One Pulse per Day
Offset: +3
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every day.

One Pulse per Week
Offset: +4
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every week.

One Pulse per Month
Offset: +5
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every month.

One Pulse per Year
Offset: +6
Usage: This 0X register will transition from a 0 to a 1 for one scan cycle at the start of every Year.

Good Time Quality
Offset: +7
Usage: This 0X register will set to a 1 when the accuracy of the time from the GPS 100 00 card is 1 mS or less.

Fair Time Quality

Offset: +8

Usage: This 0X register will set to a 1 when the accuracy of the time from the GPS 100 00 card is between 1 mS and 50 mS.

Poor Time Quality

Offset: +8

Usage: This 0X register will set to a 1 when the accuracy of the time from the GPS 100 00 card is worse then 50 mS.

Bad Time Quality

Offset: +9

Usage: This 0X register will set to a 1 when the GPS 100 00 card has gone through a power on reset sequence and has not received an initial time update from an external time reference of the PLC time of day clock through the GPS function block.