

GET_TS_EVT_M Description

Description of the Function

The GET_TS_EVT_M function block gets the time stamped data in:

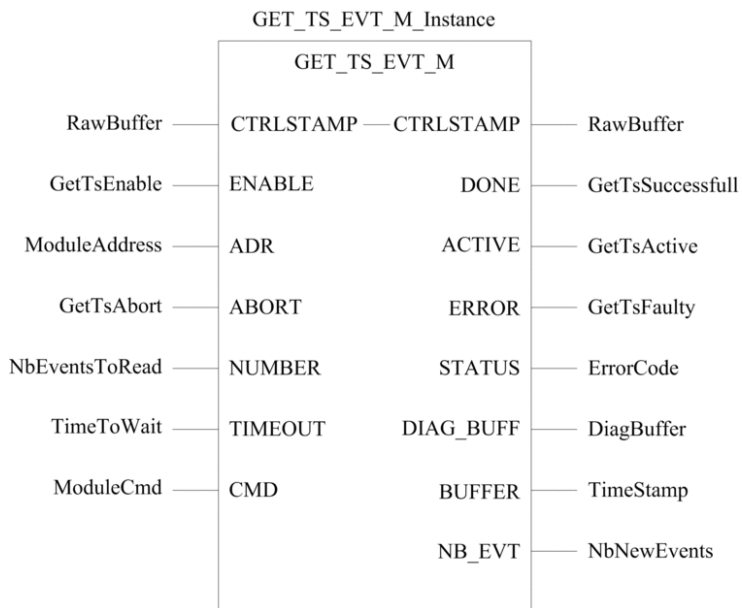
- OBMX ERT 1604T when located in Modicon M340 local drop,
- OBMX ERT 1604T when located in Modicon M580 local and remote drops,
- OBME P58 xxx Modicon M580 CPU,
- OBMX CRA 312 10 of a Modicon M580,
- OBME CRA 312 10 of a Modicon M580.

This function allows reading the event buffer to make it available for the PLC application.

The additional parameters EN and ENO can be configured.

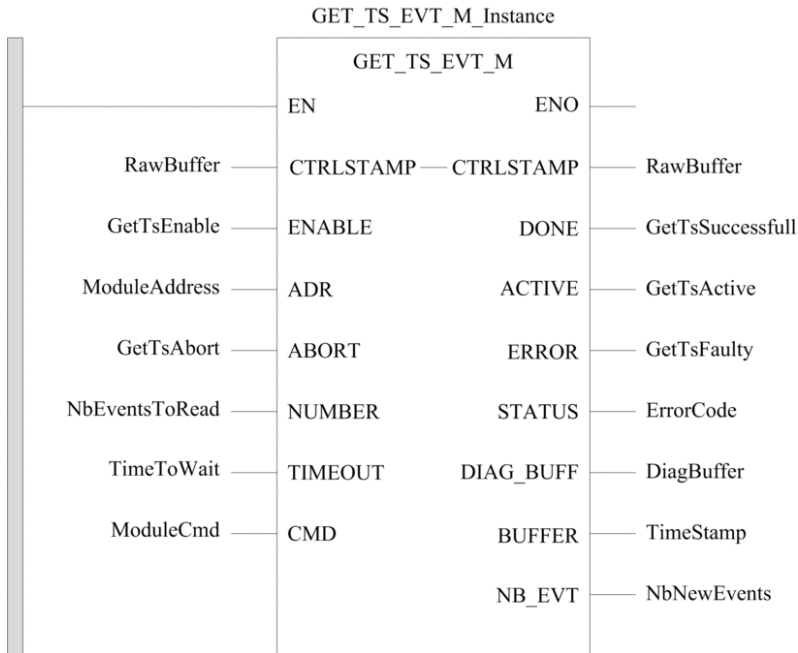
FBD Representation

Representation:



LD Representation

Representation:



IL Representation

Representation:

```
CAL GET_TS_EVT_M_Instance (CTRLSTAMP:=RawBuffer, ENABLE:=GetTsEnable,
ADR:=ModuleAddress, ABORT:=GetTsAbort, NUMBER:=NbEventsToRead, TIMEOUT:=TimeToWait,
CMD:=ModuleCmd, DONE=>GetTsSuccessfull, ACTIVE=>GetTsActive, ERROR=>GetTsFaulty,
STATUS=>ErrorCode, DIAG_BUFFER=>DiagBuffer, BUFFER=>TimeStamp, NB_EVT=>NbNewEvents)
```

ST Representation

Representation:

```
GET_TS_EVT_M_Instance (CTRLSTAMP:=RawBuffer, ENABLE:=GetTsEnable, ADR:=ModuleAddress,
ABORT:=GetTsAbort, NUMBER:=NbEventsToRead, TIMEOUT:=TimeToWait, CMD:=ModuleCmd,
DONE=>GetTsSuccessfull, ACTIVE=>GetTsActive, ERROR=>GetTsFaulty, STATUS=>ErrorCode,
DIAG_BUFFER=>DiagBuffer, BUFFER=>TimeStamp, NB_EVT=>NbNewEvents)
```

Description of the Parameters

The following table describes the input parameters:

Parameter	Type	Comment
ENABLE	BOOL	Set to 1 to send the request to the concerned module.
ADR	ANY_ARRAY_INT	Array containing the Modbus slave address, result of ADDM or ADDMX function.
ABORT	BOOL	Set to 1 to abort the current operation.
NUMBER	INT	Maximum number of events to be read in the module local buffer. NOTE: Changing the NUMBER value requires to set previously ENABLE to 0.
TIMEOUT	INT	Maximum time to wait for the drop reply. The time base for this parameter is 100 ms. NOTE: TIMEOUT = 0 corresponds to an infinite waiting value.
CMD	INT	Set to: ○ 0: read the module buffer

- 1: reset the module buffer

The following table describes the output parameters:

Parameter	Type	Comment
DONE	BOOL	Function block completed indication. Set to 1 when the execution of the function block is completed successfully.
ACTIVE	BOOL	Function block in progress indication. Set to 1 when the execution of the function block is in progress.
ERROR	BOOL	Set to 1 if an error is detected by the function block.
STATUS	WORD	Code providing a communication and operation report. <ul style="list-style-type: none"> ○ Byte 0: communication report ○ Byte 1: operation report
DIAG_BUFF	ANY_ARRAY_INT	<p>Description:</p> <ul style="list-style-type: none"> ○ Byte 0: Percentage of buffer filled ○ Byte 1: Diagnostic flags: <ul style="list-style-type: none"> ○ Bit 0 = 1: The time is OK and synchronized. ○ Bit 1 = 1: Buffer is full, events may be lost. <p>This occurs in the Overwrite mode, some or all events since the last response has been overwritten.</p> <p>The Overwrite mode is only available if the BM• CRA 312 10 is in the Applicative mode and is not supported by the BMX ERT 1604T module.</p> ○ Bit 2 = 1: Buffer is full, time stamping has stopped. ○ Bit 3 = 1: The read request is the first one since the 140 CRA 312 10, or BM• CRA 312 10 was powered-on (this bit is not used by the BMX ERT 1604T). <p>This occurs in the Stop When Full mode, which is supported by the BMX ERT 1604T module, and the BM• CRA 312 10 in the System mode.</p> <p>Any new events detected before reading the event buffer are lost.</p>
BUFFER	ANY_ARRAY_INT	<p>Raw buffer containing event time stamp entries:</p> <ul style="list-style-type: none"> ○ Word 0: <ul style="list-style-type: none"> ○ Byte 0: Reserved ○ Byte 1: Value of the variable after change detection ○ Word 1: Event ID ○ Word 2: Number of seconds since 1970/01/01 midnight (bit 15 to 0) ○ Word 3: Number of seconds since 1970/01/01 midnight (bit 31 to 16) ○ Word 4: Fraction of second (bit 15 to 0) ○ Word 5: <ul style="list-style-type: none"> ○ Byte 0: Fraction of second (bit 23 to 16) ○ Byte 1: Time quality <p>NOTE: The minimum <code>BUFFER</code> size is 12 Words and should be a multiple of 6 otherwise error 16#0009 will be raised.</p>
NB_EVENT	INT	Number of new events read in the BMX ERT 1604T local buffer.

The following table describes the input/output parameters:

Parameter	Type	Comment
CTRLSTAMP	DDT CTRL_STAMP	Specifies the CPU raw record buffer: <ul style="list-style-type: none"> ○ BEGIN index: INT ○ END index: INT ○ Operating MODE (a detailed description of the Operating mode word is given below): WORD.

STATUS Parameter Description

The following table describes the STATUS parameter:

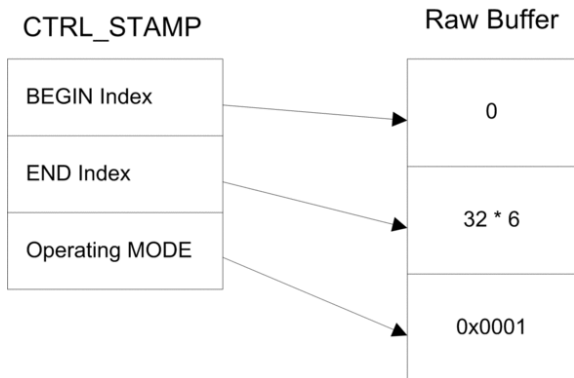
Communication Report (Byte 0)		Operation Report (Byte 1)	
Value	Description	Value	Description
00 hex	Correct exchange (request processed successfully)	00 hex	Positive result
		01 hex	The number of events in the PLC buffer reaches the maximal value.
		02 hex	The buffer is full and events have been overwritten since the last exchange.
		04 hex	Buffer is full and the recording stops
01 hex	Exchange stop on timeout	00 hex	Default value
02 hex	Exchange stop on user request (CANCEL)	00 hex	Default value
03 hex	Incorrect address format	00 hex	Default value
04 hex	Incorrect destination address	00 hex	Default value
06 hex	Incorrect specific parameters	01 hex	Invalid CMD parameter.
		02 hex	User parameters have changed between 2 invocations while the EFB execution is active.
07 hex	Problem in sending to the destination	00 hex	Default value
09 hex	Insufficient receive buffer size (<1 EVT) or buffer size is not a multiple of 6 integers	00 hex	Default value
0B hex	No processor system resources	00 hex	Default value
FF hex	Incorrect exchange (request processed with failure)	FF hex	General communication error

NOTE: If an error code is not described in the table above, see [Other error codes](#).

NOTE: ENO parameter is set to 1 on a correct exchange.

CTRLSTAMP Parameter Description

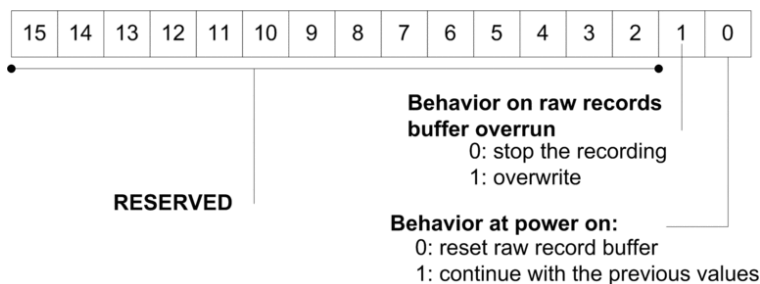
CTRL_STAMP DDT structure example and link with the PLC buffer:



The example above shows the CTRL_STAMP content after writing 32 events (1 event entry is 6 words long) in the PLC buffer configured as follows:

- PLC buffer is located and there are 32 events to write
- Stop the recording when the buffer is full and continue with the previous value on power on.

Operating MODE word structure:



Buffer level:

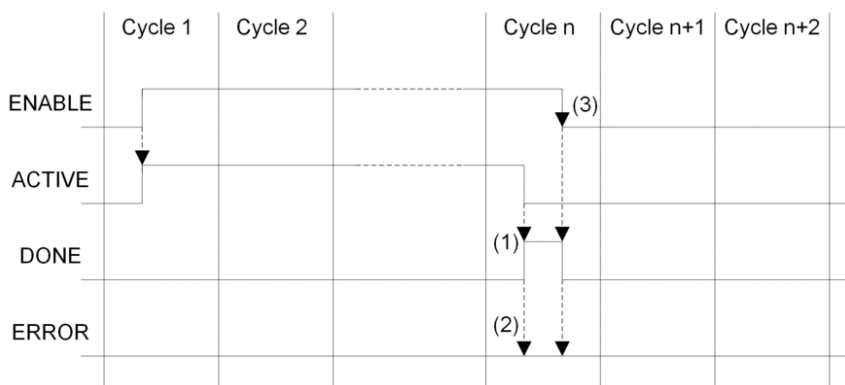
- If BEGIN Index = END Index then the buffer is empty
- If END Index + 6 = BEGIN Index then the buffer is full (in the preceding equation, 6 represents the size of 1 event). The buffer is full when there is one space left for 1 event (6 x INT).

EFB behavior on buffer full depends on the Operating MODE parameter, bit 1 value:

- If **Behavior on raw records buffer overrun** is set to 0 (stop the recording), the buffer is not filled with new data.
- If **Behavior on raw records buffer overrun** is set to 1 (overwrite buffer), elder data or overwritten with newer data. In this case, the EFB updates both BEGIN Index and END Index.

Operating Mode of Enable, Active, Done and Error Parameters

The ENABLE, ACTIVE, DONE (or SUCCESS) and ERROR parameters operate as follows:



(1) DONE = 1 if no error, DONE = 0 if error

(2) ERROR = 0 if no error, ERROR = 1 if error

(3) If the `ENABLE` bit is reset to 0 before completion, the function block is stopped (active bit to 0). To have a complete block execution, the value of 1 must be applied on the `ENABLE` bit until the operation has finished or until an error has occurred.

The `ENABLE` parameter is written by the application.

The `ACTIVE`, `DONE` and `ERROR` parameters are read by the application.

To launch the communication function only once, the `ENABLE` signal needs to be reset to 0 once the `ACTIVE` parameter is set to 0. If the `ENABLE` parameter is maintained to 1 once the `ACTIVE` parameter is set to 0, the communication function is launched again and the `ACTIVE` parameter will be set to 1 on the next cycle.