Siemens
Industry
Online
Support

APPLICATION EXAMPLE

S7-1500 TM FAST – Getting Started

S7-1500 TM FAST / TIA Portal V17

SIEMENS

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

Table of contents

1. Introduction

Overview

The user-programmable TM FAST technology module in the S7-1500 / ET 200MP gives you the ability to control extremely fast processes. Reaction times in the microsecond to nanosecond range are possible.

The module's function is programmed specifically for the application. For this purpose, an application is created using the Intel® Quartus® Prime engineering tool chain from Intel, which is downloaded to the TM FAST, where it is executed by an FPGA (field programmable gate array).

Applications

The TM FAST module can be deployed wherever extremely fast and high-precision responses are essential, and where a standard PLC is no longer sufficient. Solutions with the TM FAST outperform a standard PLC with respect to precision, resolution and reaction time by a factor of up to 1000.

Examples include:

- Short, adjustable and reproducible responses, e.g. for
  - Ejection of faulty parts
  - Sorting plants
  - Quick shutdown to protect the machine
- Position detection using
  - Incremental encoders
  - Absolute encoders
- Output of precise pulses and pulse patterns, e.g. for
  - Output of pulse or pixel patterns via multiple parallel digital outputs
  - Output of pulse patterns with freely defined pulses and pauses
  - Output of pulse-width modulated signals
  - Output of pulses of precisely defined length
- Acquisition of fast signals, e.g. for
  - Counting events
  - Measuring a frequency
  - Measuring a pulse duration
  - Starting an output sequence without delay

What do I get with this example?

The example project will

- help you familiarize yourself with the essential concepts of the TM FAST module;
- present all steps in a detailed manner, from compiling the logic with the Quartus® software to controlling to module with a watch table in the TIA Portal project;
- provide the example code in the SIMATIC CPU and the TM FAST with explanations.

Requirements

We recommend starting with our introductory "Hello World" example in the programming manual (\4\) unless you are already familiar with the essential tools: Intel® Quartus® Prime, MultiFieldbus Configuration Tool (MFCT) and TIA Portal.

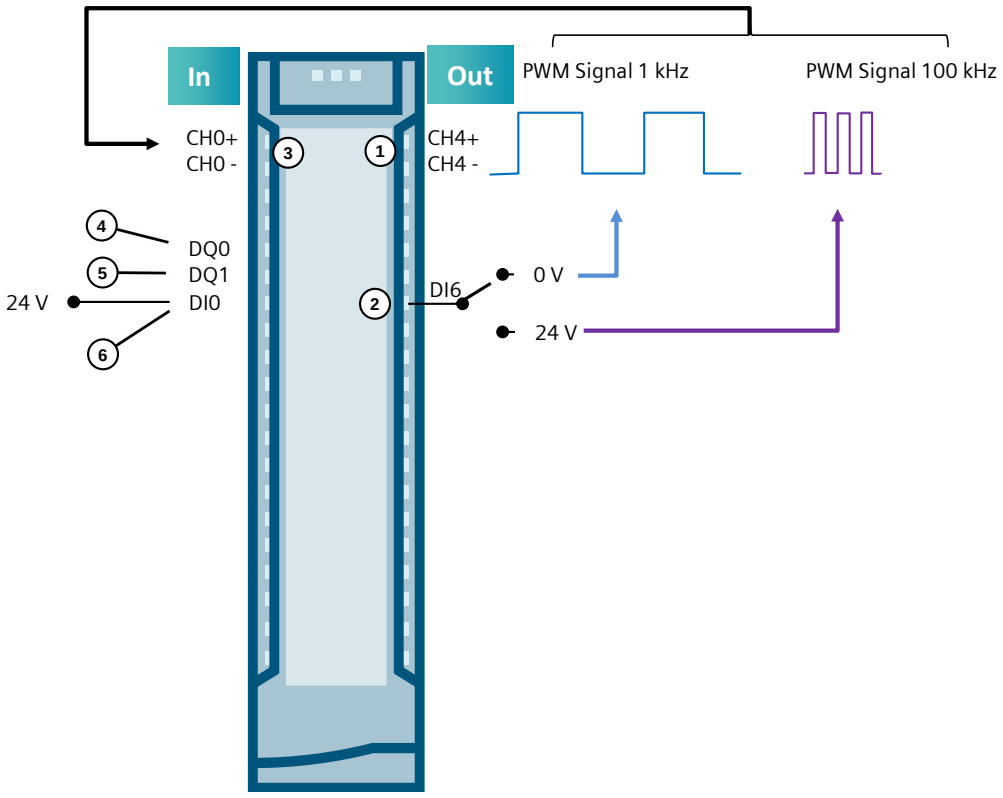
This example assumes you have mastered the basics listed above.

1.1. Principle of operation

1.1.1. Overview

Implemented functions

The graphic below shows the principal functions that are implemented in this example.



1. The module outputs a PWM signal with a 50% duty cycle at terminals CH4+/CH4-.
2. The output frequency is selected with the "DI6" input.
   If the input is "0", the output frequency will be 1 kHz. If it is "1" (24 V), the output frequency will be 100 kHz.
3. Terminals CH0+/CH0- serve as a counter input. The logic counts the number of pulses with a count width of 16 bits. That is, it counts from 0x0000 to 0xFFFF, then the count value (CntVal) restarts at 0x0000.
4. The DQ0 output is "1" if the current count value (CntVal) is between 0x2000 and 0x4000, otherwise it is "0".
5. The DQ1 output is "1" if the current count value is less than the value "Cam1OffVal", which is specified by the CPU via the control interface.
6. The DI0 input serves as an "alive bit" to help the CPU detect whether the application is active on the module.
   In the example, the input is hardwired to 24 V. The state of the digital inputs in the example is displayed by the user logic in the feedback interface. Once the logic is active, a "1" is displayed in the feedback interface at byte 3 / bit 0.

To represent the various ways of communicating with the CPU, the user-defined write-data record TMFASTUserWriteRec (set to 4 bytes in length) is mirrored back to the feedback interface (FB_IF(3)).

Description of the control interface: CPU -> TM FAST

The example uses bytes 0 to 3, of the maximum possible 32 bytes of output addresses in the control interface.

Register	ByteNo	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CTRL_IF(0)	0 .. 1	not used (0x0000)							
	2 - 3	Cam1OffVal (0x0000 – 0xFFFF)							
	4..31	not used (0x0000)							

Table 1-1

Description of the feedback interface: TM FAST -> CPU

The example uses bytes 0 to 19, of the maximum possible 32 bytes of input addresses in the feedback interface.

Register	ByteNo	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FB_IF(0)	0	"0"	"0"	"0"	"0"	Status DQ11	Status DQ10	Status DQ9	Status DQ8
	1	Status DQ7	Status DQ6	Status DQ5	Status DQ4	Status DQ3	Status DQ2	Status DQ1	Status DQ0
	2	"0"	"0"	"0"	"0"	Status DI11	Status DI10	Status DI9	Status DI8
	3	Status DI7	Status DI6	Status DI5	Status DI4	Status DI3	Status DI2	Status DI1	Status DI0
FB_IF(1)	4	"0"	"0"	"0"	"0"	Quality DQ QI11	Quality DQ QI10	Quality DQ QI9	Quality DQ QI8
	5	Quality DQ QI7	Quality DQ QI6	Quality DQ QI5	Quality DQ QI4	Quality DQ QI3	Quality DQ QI2	Quality DQ QI1	Quality DQ QI0
	6	"0"	"0"	"0"	"0"	Quality DQ QI11	Quality DQ QI10	Quality DQ QI9	Quality DQ QI8
	7	Quality DI QI7	Quality DI QI6	Quality DI QI5	Quality DI QI4	Quality DI QI3	Quality DI QI2	Quality DI QI1	Quality DI QI0
FB_IF(2)	8	OE CH7	OE CH6	OE CH5	OE CH4	OE CH3	OE CH2	OE CH1	OE CH0
	9	QI CH7	QI CH6	QI CH5	QI CH4	QI CH3	QI CH2	QI CH1	QI CH0
	10	Status CH7 Tx	Status CH6 Tx	Status CH5 Tx	Status CH4 Tx	Status CH3 Tx	Status CH2 Tx	Status CH1 Tx	Status CH0 Tx
	11	Status CH7 Rx	Status CH6 Rx	Status CH5 Rx	Status CH4 Rx	Status CH3 Rx	Status CH2 Rx	Status CH1 Rx	Status CH0 Rx
FB_IF(3)	12 – 15	UserWriteVal (last value written to the TM FAST via user data)							
FB_IF(4)	16 – 17	not used (0x0000)							
	18 – 19	CntVal (last value for pulses counted to CH0+/CH0- infinite counter 0x0000 – 0xFFFF)							
	20 – 31	not used (0x0000)							

Table 1-2

NOTE Chapter 3 contains a complete list of the control interface and the feedback interface PLC format and TM FAST format.

1.1.2. User logic of the Quartus® project

The logic realized in the Quartus® project is divided into the functionalities from chapter 1.1.1 and is explained in this chapter by way of excerpts.

1.1.2.1. Declarations

The aliases in the declarations section of the program make it easier to read the data in the feedback interface.

The statuses of the individual digital inputs and outputs (STATUS_DI / STATUS_DQ) are each represented as an array with a single alias.

QUALITY_DI and QUALITY DQ contains the quality information for the respective input or output ("1" means "bad" and indicates an error).

For the RS 485 channels, the information is tagged as STATUS_RX (receiving direction) or STATUS_TX (transmit direction). OE_TXRX (Output Enable for RS 485 channels) lets you switch between transmit and receive mode.

```
17  -- Control Interface
18  alias   Cam1OffVal:   std_logic_vector is CTRL_IF(0)( 15 downto 0);
19
20  -- Feedback Interface
21  alias   STATUS_DI:   std_logic_vector is FB_IF(0)( 11 downto 0);
22  alias   STATUS_DQ:   std_logic_vector is FB_IF(0)( 27 downto 16);
23
24  alias   QUALITY_DI:   std_logic_vector is FB_IF(1)( 11 downto 0);
25  alias   QUALITY_DQ:   std_logic_vector is FB_IF(1)( 27 downto 16);
26
27  alias   STATUS_RX:   std_logic_vector is FB_IF(2)( 7 downto 0);
28  alias   STATUS_TX:   std_logic_vector is FB_IF(2)( 15 downto 8);
29  alias   QI_TXRX:   std_logic_vector is FB_IF(2)( 23 downto 16);
30  alias   OE_TXRX:   std_logic_vector is FB_IF(2)( 31 downto 24);
31
32  alias   UserWriteVal:   std_logic_vector is FB_IF(3);
33
34  alias   CntVal:   std_logic_vector is FB_IF(4)( 15 downto 0);
```

1.1.2.2. IF_PROC process

The IF_PROC process shows the status of all I/Os in the feedback interface (see Table 1-2) by writing the statuses to the feedback interface for the CPU.

```
53  IF_PROC: PROCESS(CLK,RST)
54  BEGIN
55     if RST = '1' then
56        FB_IF <= (others => (others => '0'));
57     elsif Rising_edge(clk) then
58        STATUS_DI <= DI;
59        STATUS_DQ <= DQ;
60        QUALITY_DI <= DI_QI_BAD;
61        QUALITY_DQ <= DQ_QI_BAD;
62        STATUS_TX <= RS485_TX;
63        STATUS_RX <= RS485_RX;
64        QI_TXRX <= RS485_QI_BAD;
65        OE_TXRX <= RS485_OE;
66        CntVal <= S_CntVal;
67        UserWriteVal <= WR_REC(0);
68     end if;
69  END PROCESS;
```

NOTE

The state of digital input DI0 (in the logic, FB_IF(0)(0)) is shown from the CPU's perspective in byte 3 bit 0. This is because a conversion between Little Endian and Big Endian takes place between the module and TIA Portal.

If you connected input DI0 with 24 V, you can use the value of this "alive bit" in the CPU to see whether the logic is active.

1.1.2.3. FUNC_PWM process

Depending on the value of input DI6, the FUNC_PWM process outputs a pulse-width modulated signal at the CH4- output. The pulse/pause ratio is 50/50.

- DI6 = "0": Output frequency = 1 kHz
- DI6 = "1": Output frequency = 100 kHz

```
91  FUNCT_PWM: PROCESS(CLK, RST, CTRL_IF)
92  BEGIN
93     if RST = '1' then
94        RS485_OE(4)   <= '0';
95        RS485_TX(4)   <= '0';
96     elsif rising_edge(clk) then
97        RS485_OE(4)   <= '1';
98        if  DI(6) = '0'  then
99           RS485_TX(4)    <= '0';
100          PWM_CTRL_PERIOD_RS485 <= PWM_CTRL_PERIOD_1K;   -- load PWM periode value
101          PULSE_LENGTH_RS485 <= PULSE_LENGTH_1K;      -- load pulse width value
102       else
103          RS485_TX(4)    <= '0';
104          PWM_CTRL_PERIOD_RS485 <= PWM_CTRL_PERIOD_100K;   -- load PWM periode value
105          PULSE_LENGTH_RS485 <= PULSE_LENGTH_100K;   -- load pulse width value
106       end if;
107       if ( unsigned ( PWM_COUNTER ) <= 1 ) then
108          PWM_COUNTER <= PWM_CTRL_PERIOD_RS485;
109          RS485_TX(4)    <= '0';
110       else
111          PWM_COUNTER <= std_logic_vector ( unsigned ( PWM_COUNTER ) - 1 );
112       end if;
113        -- pulse width comparator (Duty Cycle)
114       if unsigned ( PWM_COUNTER ) <= unsigned ( PULSE_LENGTH_RS485 ) then
115          RS485_TX(4)    <= '1';
116       else
117          RS485_TX(4)    <= '0';
118       end if;
119    end if;
120 END PROCESS;
```

1.1.2.4. FUNC_INC_CH0 process

The 16-bit counter at CH0 is wired to output CH4 (PWM signal). The counter increments its value CntVal every time there is a positive edge at CH0 until it reaches the value 0xFFFF, then it jumps back to 0x0000. The count value is represented in the <CntVal> tag as a 16-bit value; it is reflected in the feedback interface in bytes 18 and 19.

```
71  FUNCT_INC_CH0: PROCESS(CLK, RST)
72  variable cnt_edge : integer := 0;
73  BEGIN
74     if RST = '1' then
75        S_CntVal <= (others => '0');
76     elsif rising_edge(clk) then
77        d_bit(0) <= RS485_RX(0); -- Positiv Edge Detection
78        d_bit(1) <= d_bit(0);
79        S_CntVal <= std_logic_vector ( to_unsigned(cnt_edge,S_CntVal'length));
80        if d_bit = "01" then
81           if cnt_edge = 65535 then
82              cnt_edge := 0;
83           else
84              cnt_edge := cnt_edge + 1;   -- Increment
85           end if;
86        end if;
87     end if;
88  END PROCESS;
```

1.1.2.5. FUNC_CAM_2 process

The FUNC_CAM_2 function compares the count value with two fixed comparison values and issues the result at output DQ0.

The count value <CntVal> is compared with the two variables <hex_compare_a> (0x2000) and <hex_compare_b> (0x4000). DQ0 is set to "1" if and only if CntVal is between the two values.

```
122  FUNCT_CAM_2: PROCESS(CLK, RST, CntVal)
123   variable hex_compare_a : std_logic_vector(15 downto 0) := x"2000";
124   variable hex_compare_b : std_logic_vector(15 downto 0) := x"4000";
125   BEGIN
126  if RST = '1' then
127     DQ(0) <= '0';
128  elsif Rising_edge(CLK) then
129     if hex_compare_a <= S_CntVal AND S_CntVal <= hex_compare_b then
130        DQ(0) <= '1';
131     else
132        DQ(0) <= '0';
133     end if;
134  end if;
135   END PROCESS;
```

1.1.2.6. FUNC_CAM_1 process

The FUNC_CAM_1 function compares the count value CntVal with a comparison value from the control interface and outputs the result at output DQ1.

```
137  FUNCT_CAM_1: PROCESS(CLK, RST, CntVal)
138   BEGIN
139  if RST = '1' then
140     DQ(1) <= '0';
141  elsif Rising_edge(CLK) then
142     if S_CntVal <= Cam1OffVal then
143        DQ(1) <= '1';
144     else
145        DQ(1) <= '0';
146     end if;
147  end if;
148   END PROCESS;
```

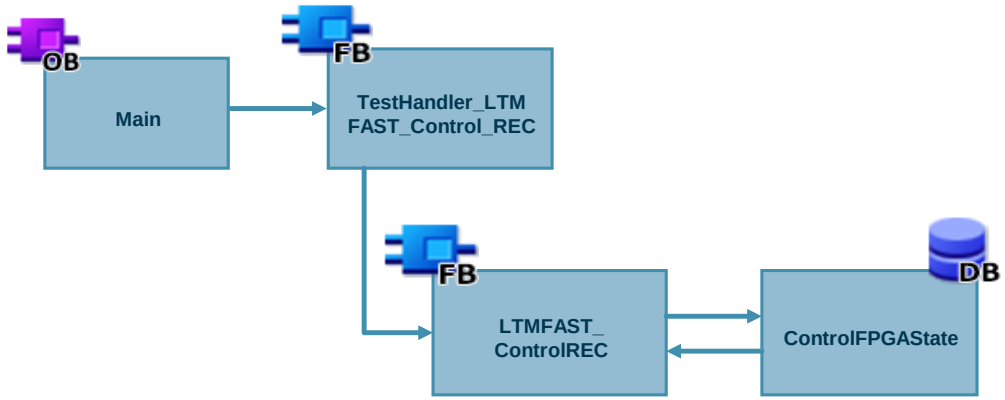
1.1.3. STEP 7 program

The task of the STEP 7 program in this example is simply to manage the TM FAST application and to receive status information.

The control/feedback interface is accessible via the configured process image. For easier operability in the watch table (see chapter 2.3), the two ranges are mapped to custom data types.

Diagram

The graphic below shows the program structure in the S7-1500 CPU.



Program blocks

The user program of the SIMATIC S7-1500 CPU consists of the following elements:

Element	Description
Main	In OB1, the function block "TestHandler_LTMFAST_ControlREC" (including the associated instance data block) is called cyclically.
FB "TestHandler_LTMFAST_ControlREC"	This block internally calls the block "LTMFAST_ControlREC".
FB "LTMFAST_ControlREC"	This block from the LTMFAST library manages the TM FAST application.
DB "ControlFPGAstate"	Global DB for holding the parameters of the FB "LTMFAST_ControlREC"
PLC data type "TMFASTGS01Control"	Data type for mapping the control interface to the configured process image of the outputs of the TM FAST (see Table 1-1)
PLC data type "TMFASTGS01Feedback	Data type for mapping the feedback interface to the configured process image of the inputs of the TM FAST (see Table 1-2)

Table 1-3

Tags in the control interface and feedback interface

The tags <FAST_FB> and <FAST_CTRL> map the configured I/O process image ranges with the tag types "TMFASTGS01Feedback" and "TMFASTGS01Control" to the defined control/feedback interfaces from Table 1-1 and Table 1-2.

Name	Data type	Address
FAST_FB	"TMFASTGS01Feedback"	%I0.0
Status DQ[11..8]	Byte	%IB0
Status DQ[7..0]	Byte	%IB1
Status DI[11..8]	Byte	%IB2
Status DI[7..0]	Byte	%IB3
Quality DQ[11..8]	Byte	%IB4
Quality DQ[7..0]	Byte	%IB5
Quality DI[11..8]	Byte	%IB6
Quality DI[7..0]	Byte	%IB7
Status OE	Byte	%IB8
Quality CHx	Byte	%IB9
Status TX	Byte	%IB10
Status RX	Byte	%IB11
UserWriteVal	UDInt	%ID12
CntVal	UDInt	%ID16
FAST_CTRL	"TMFASTGS01Control"	%Q0.0
Reserved	Word	%QW0
Cam1OffVal	Word	%QW2

1.2. Components used

The following hardware and software components were used to create this application example:

Component	Quantity	Article number	Note
CPU 1516-3PN/DP	1	6ES7 516-3AN02-0AB0	Or a similar S7-1500 CPU
TM FAST	1	6ES7 554-1AA00-0AB0	
TIA Portal V17		6ES7 7810...	Or higher

Table 1-4

The listed components can be purchased (e.g. via the Siemens Industry Mall).

The following additional software components are also required in order to use this example. Download the files from the specified links in SiePortal or the respective vendors, then install the packages to your PC.

	Component	File	Download link
1.	Intel® Quartus® Prime V22.1		\8\
2.	Quartus® TM FAST base project with Siemens system logic	MP_FAST_1_V1.0.1.zip	\5\
3.	MultiFieldbus Configuration Tool (MFCT) V1.5 or later	MFCT_1_5_0_0.zip	\6\
4.	HSP368 (only for TIA Portal V17)	TIA_Portal_V17_HSP.zip	\7\

Table 1-5

This application example consists of the following components:

Component	File name	Note
This documentation	109823442_Getting_Started_TM-FAST_DOC_V1.0_en	
TIA Portal project (V17)	109823442_TM_FAST_Getting_Started_PROJ_V1.0.zip	Compressed TIA Portal project (*.zap)
FPGA project	109823442_ TFL_FASTGS01_PROJECT_a.zip	

Table 1-6

2. Engineering

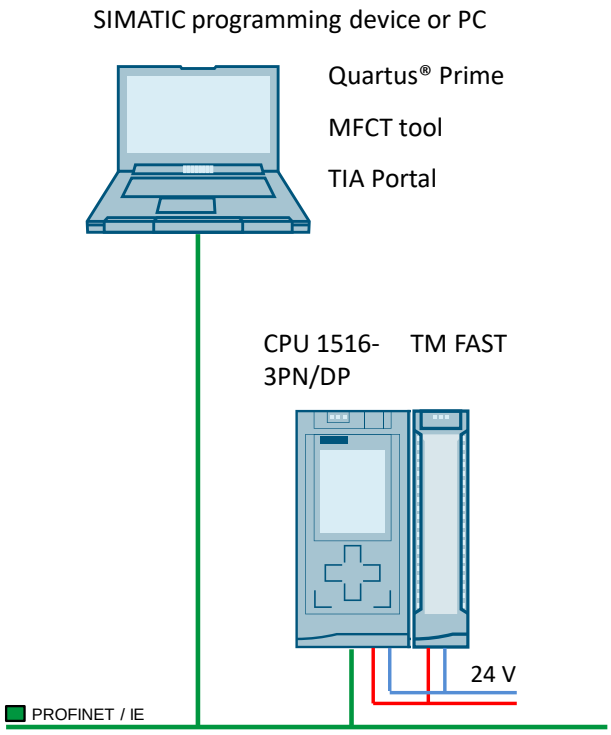
2.1. Hardware setup

The setup

Chapter 1.2 lists the required hardware components.

ATTENTION	Observe the SIMATIC S7-1500 setup guidelines. Please read the corresponding device manuals. Only switch on the power supply after you have completed and checked the assembly!

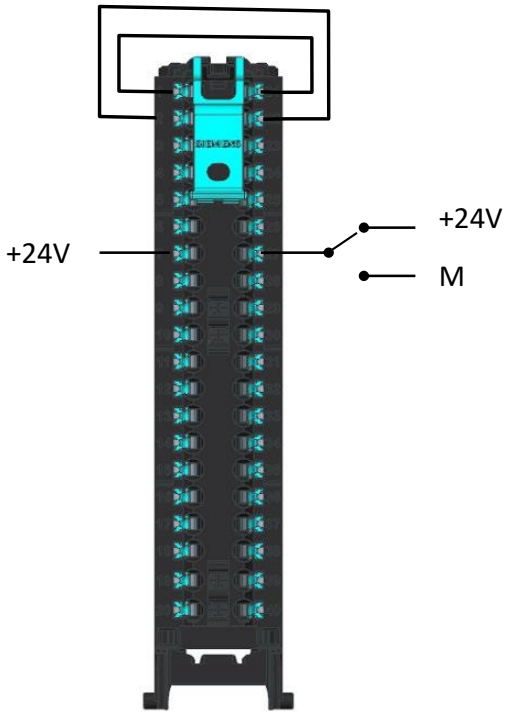
The following graphic shows the hardware setup implemented in this example.



Wiring the TM FAST module connections

The user program in the TM FAST generates various signals (see chapter 1.1.1), which are read back via the counter inputs.

Wire the 40-pin front panel connector of the TM FAST as shown in the illustration and table below.

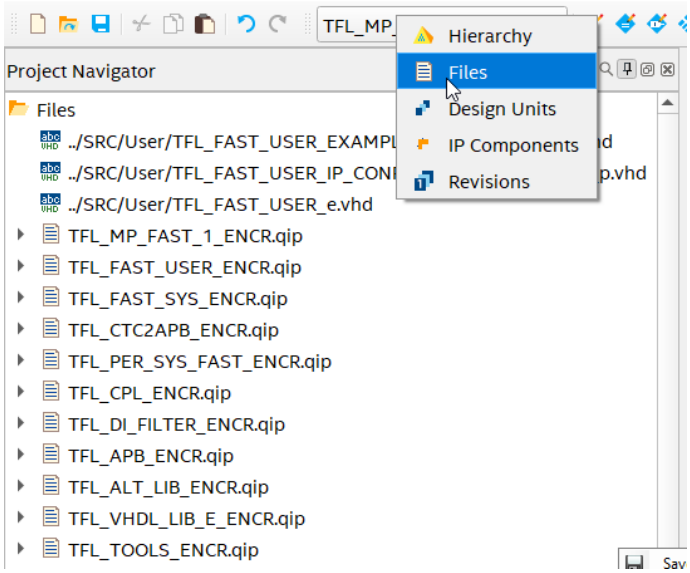


Terminal		Function		Function		Terminal
		Left connector side		Right connector side		
1	CH0+	Pulse Input+ → wired to CH4+		Pulse Output+ → wired to CH0+	CH4+	21
2	CH0-	Pulse Input- → wired to CH4-		Pulse Output- → wired to CH0-	CH4-	22
3	CH1+				CH5+	23
4	CH1-				CH5-	24
5	DQ0	TRUE if 0x2000 < <CntVal> < 0x4000			DQ6	25
6	DQ1	TRUE if <CntVal> < <Cam1OffVal>			DQ7	26
7	DI0	→ hardwired to 24 V		0V → 1 kHz at CH4 24V → 100 kHz at CH4	DI6	27
8	DI1				DI7	28
9	DIQ2				DIQ8	29
10	CH2+				CH6+	30
11	CH2-				CH6-	31
12	CH3+				CH7+	32
13	CH3-				CH7-	33
14	DQ3				DQ9	34
15	DQ4				DQ10	35
16	DI3				DI9	36
17	DI4				DI10	37
18	DIQ5				DIQ11	38
19	-				M	39
20	-				M	40

2.2. Project engineering / configuration

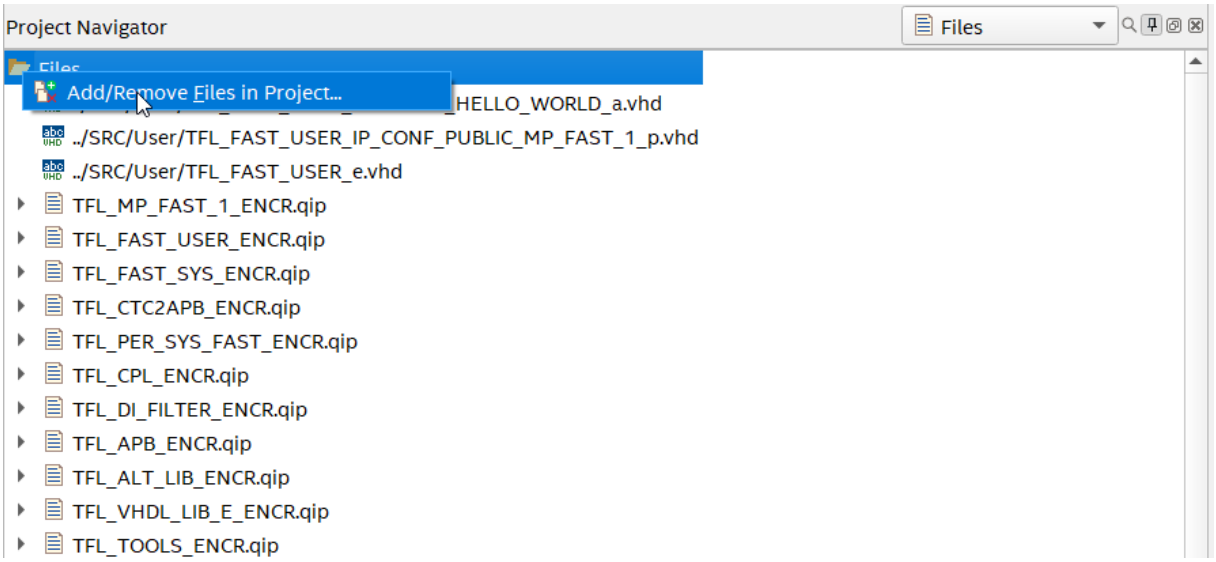
Quartus® Prime engineering

1. Go to the SiePortal page "Supplementary SW Components for TM FAST" (\5\), then download and unzip the file "MP_FAST_1_V1.0.1.zip". The extracted file is provided as the Intel® Quartus® Prime archive file "MP_FAST_1.qar". It contains the complete TM FAST base project with the Siemens system logic (TFL_MP_FAST_1.qpf) for the Cyclone® 10 FPGA located on the TM FAST module.
2. Launch the Quartus® Prime software and open the extracted Quartus® project "TFL_MP_FAST_1.qpf".
3. Select the "Files" option in the Project Navigator.

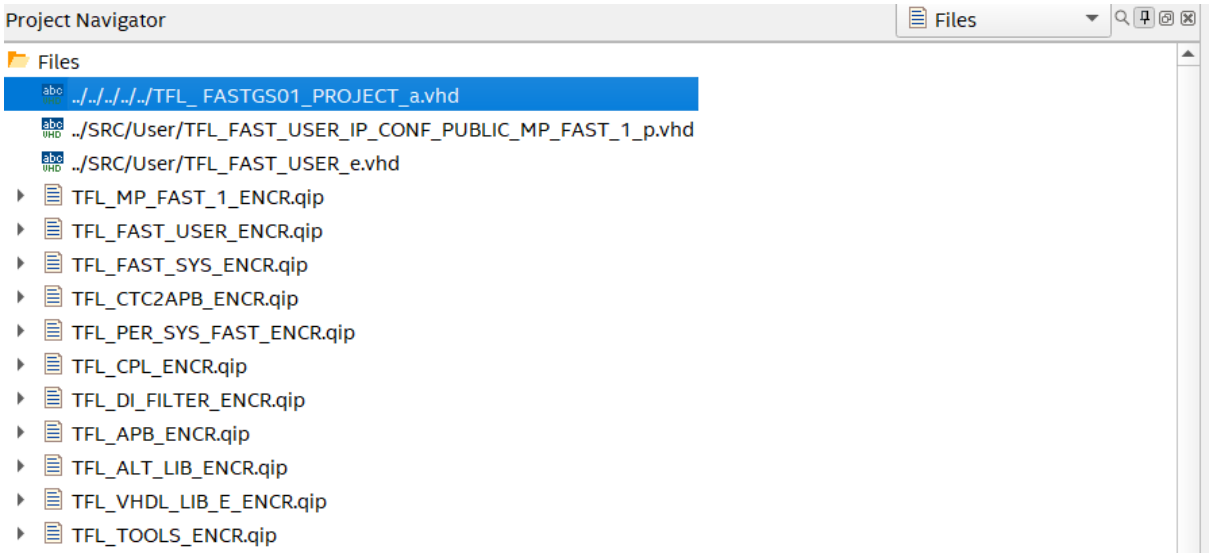


4. In the open base project, prepare the function of the Getting Started example.
   To do this, swap the architecture file:

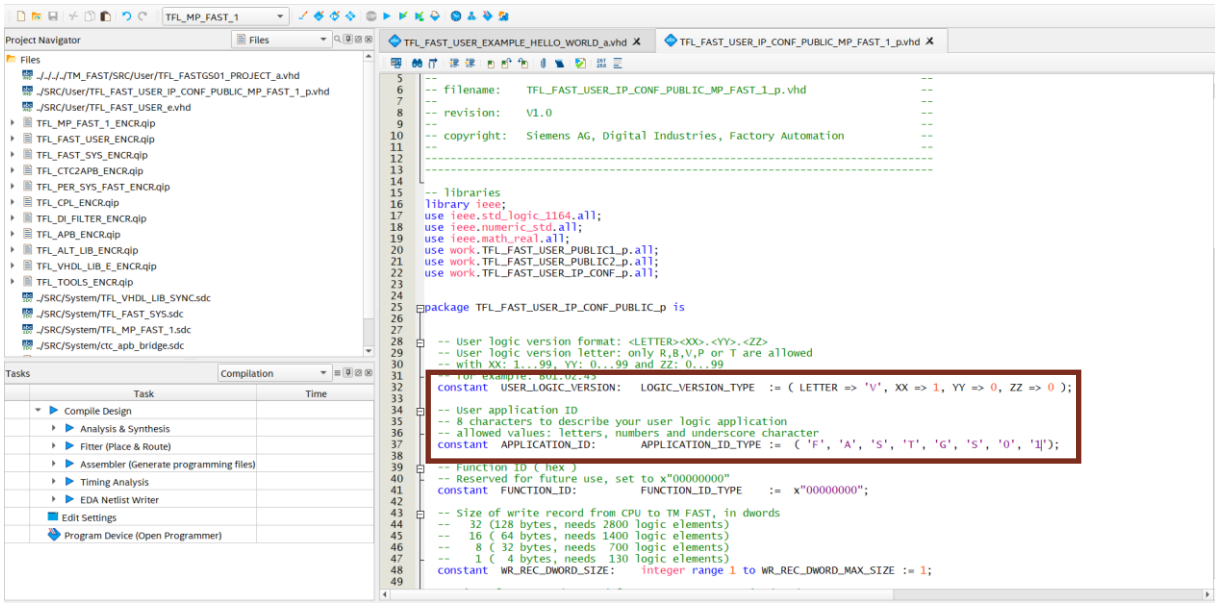
   - Right-click in the Project Navigator on "Files", then select "Add/Remove Files in Project...".
   - Remove the file TFL_USER_EXAMPLE_HELLO_WORLD_a.vhd from the project.
   - Add the file TFL_FASTGS01_PROJECT_a.vhd to the project.



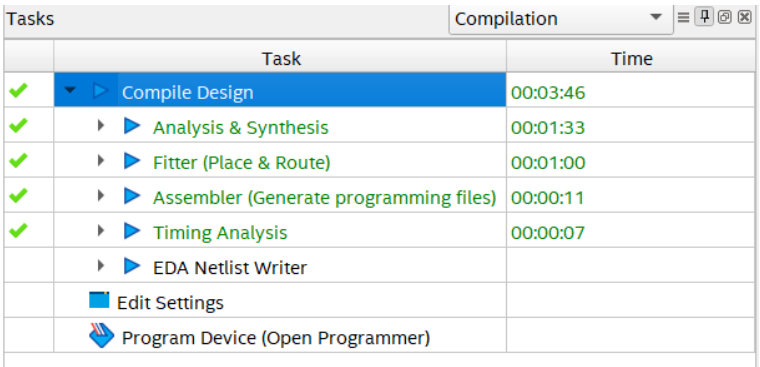
Result:
You have replaced the file TFL_USER_EXAMPLE_HELLO_WORLD_a.vhd with a new example architecture file, TLF_FASTGS01_PROJECT_a.vhd.



5. Now modify the user logic version (here: V1.0.0) and Application ID (here: FASTGS01) in the file TFL_FAST_USER_IP_CONF_PUBLIC_p.vhd.



6. Compile the project.



NOTE We recommend first performing the "Analysis & Synthesis" process before starting the complete Compile Design process.

Result:

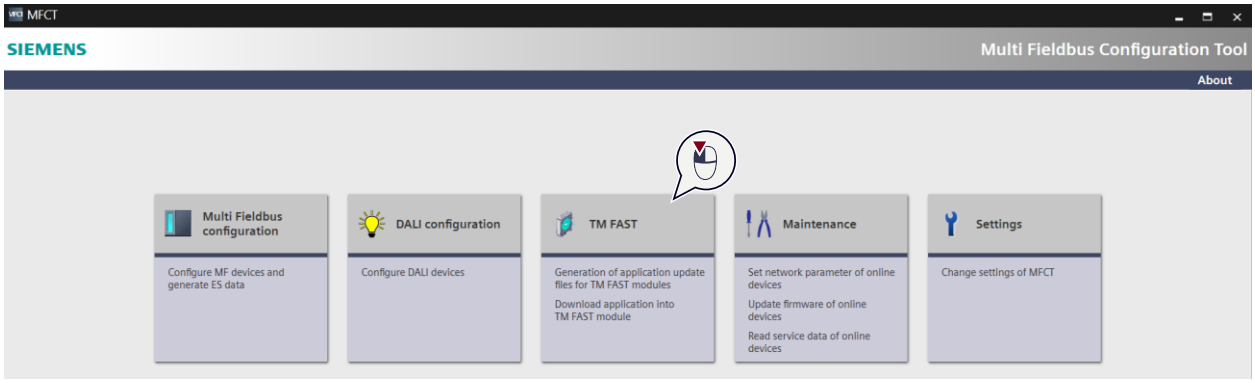
The compilation gives you two files in the project folder:

- TFL_MP_FAST_1.rbf:
  Raw Binary File. You will need this file to generate the UPD file with the MFCT tool.
- TFL_MP_FAST_1.sof:
  SRAM Object File. This file allows you to load the logic directly to the FPGA using the download cable and the TM FAST Debug Connector.

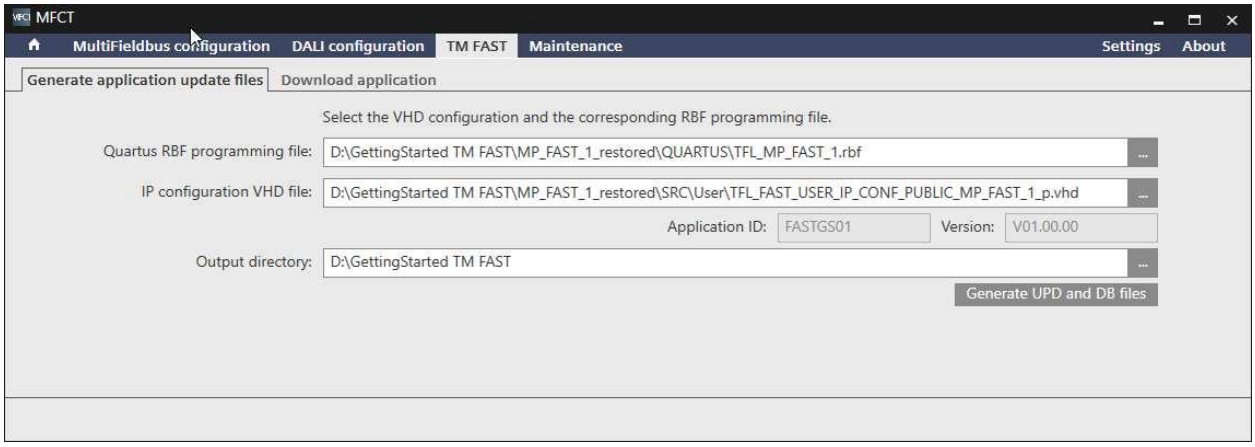
Actions in the MFCT tool

To load the compiled logic to the non-volatile flash memory of the TM FAST, you must now generate a UPD file with the MFCT tool.

1. Open the MFCT tool and click on the "TM FAST" tile.



2. Enter the following files in the respective fields:
   - the Raw Binary File "TFL_MP_FAST_1.rbf" (result of the Quartus® compilation)
   - the file "TFL_FAST_USER_IP_CONF_PUBLIC_MP_FAST_1_p.vhd" (for the application ID and the version)



NOTE

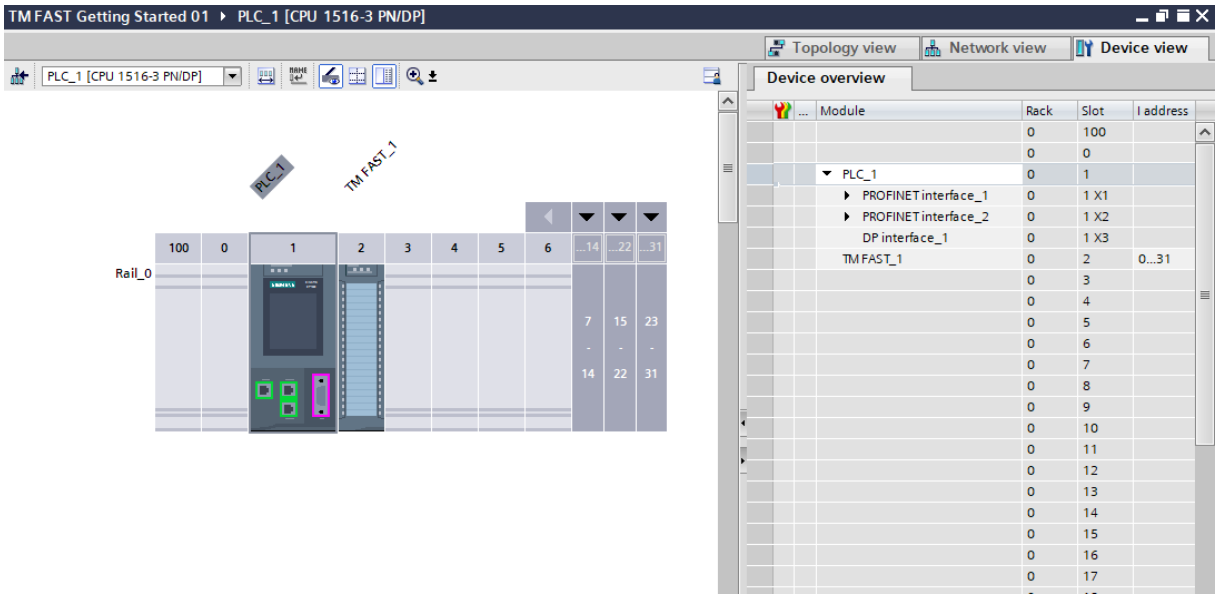
In a distributed architecture, you can download the UPD file to the module's flash memory with the MFCT tool. As of version 1.5.1 of the MFCT tool, the download is also supported for a centralized architecture.

The TM FAST module in our example is configured for a centralized architecture. You can download the UPD file straight to the module from the TIA Portal project.

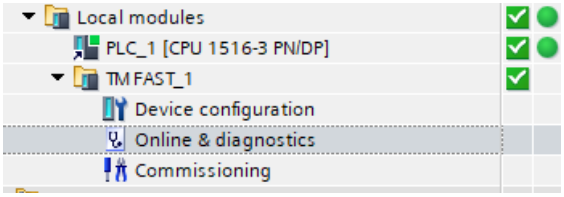
3. Launch TIA Portal and extract the archive "TM FAST Getting Started 01.zap17".

4.

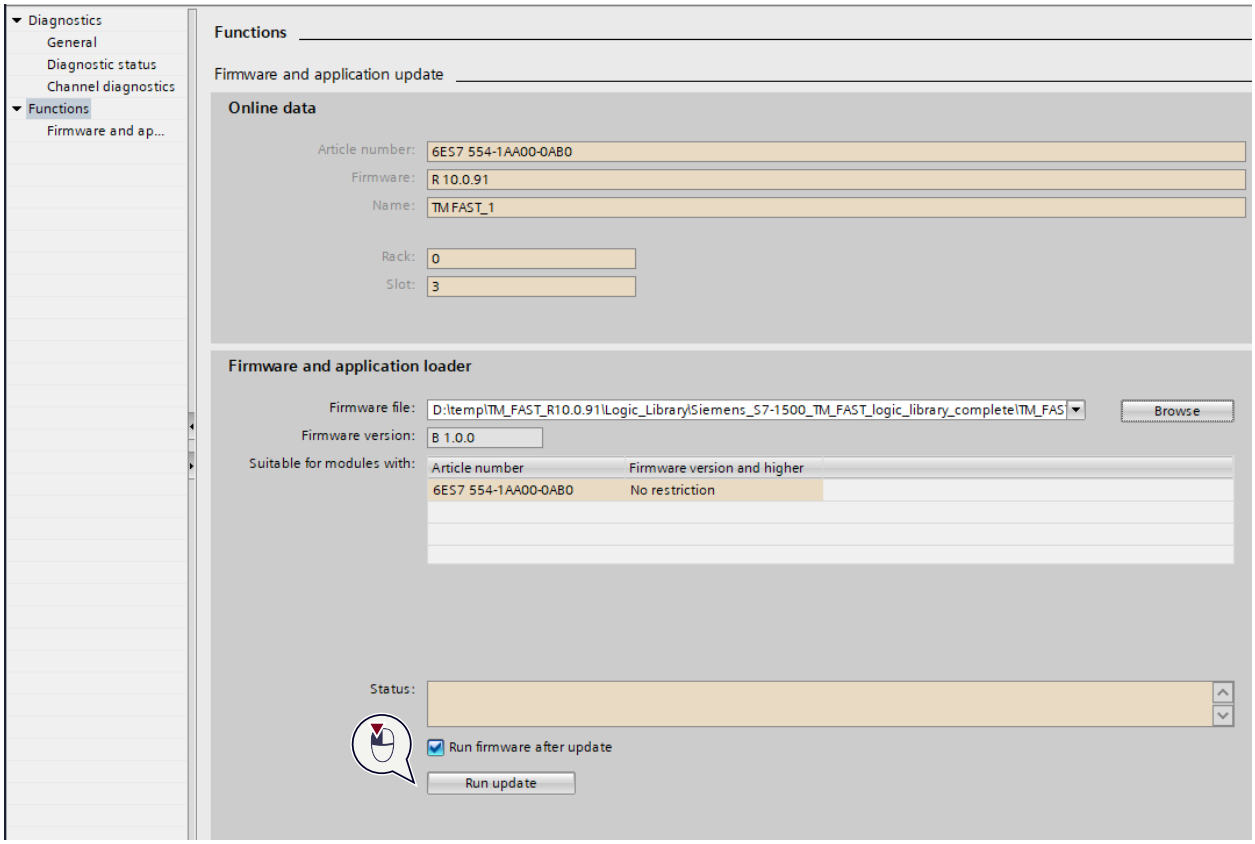
NOTE The TM FAST module is already part of TIA Portal V18. For TIA Portal V17, you will also need HSP368.



5. Go online and, in the project tree under "Local modules", select "Online & diagnostics" in the menu for the TM FAST.

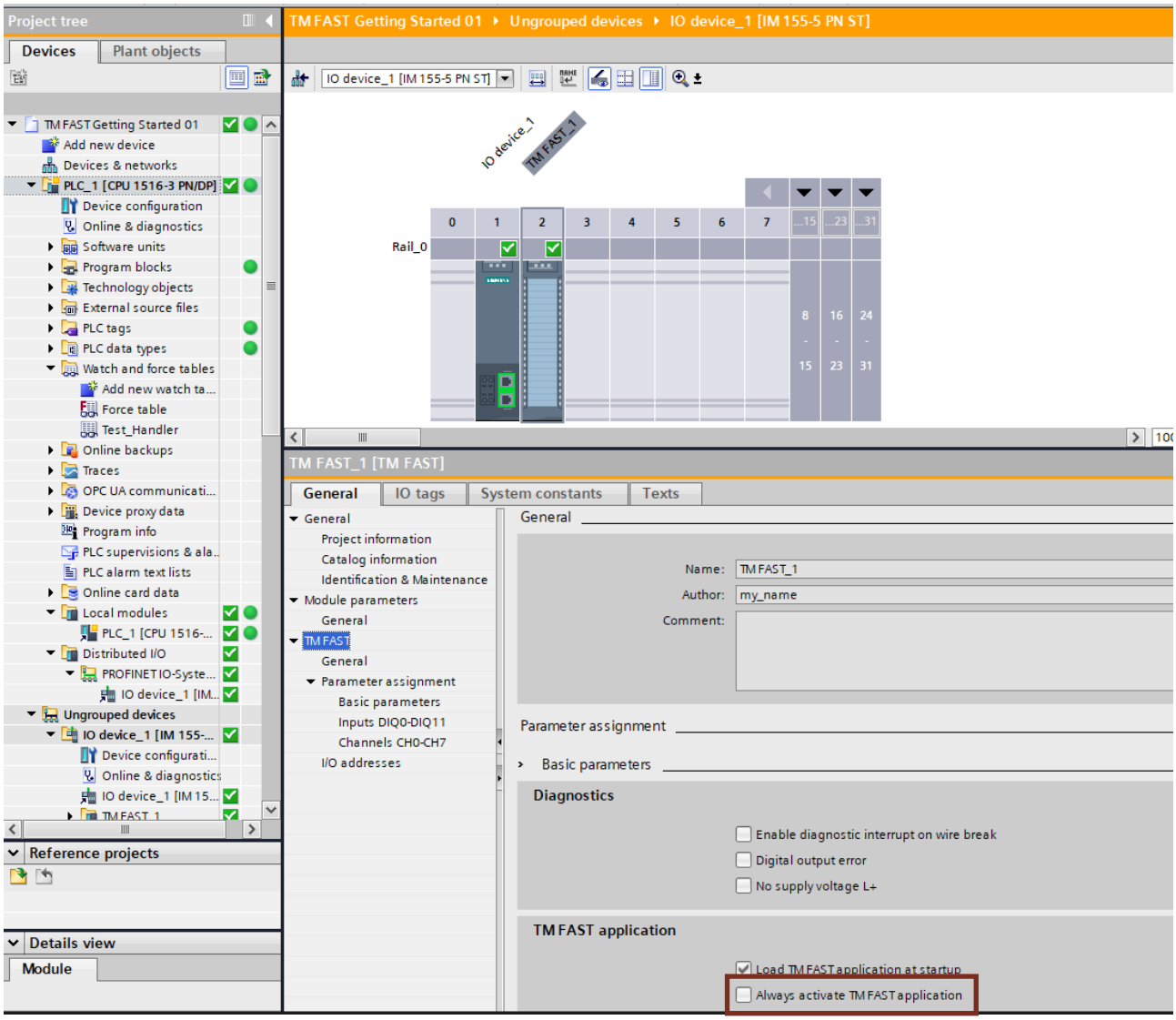


6. Under "Functions > Firmware and application loader", select the UPD file that you created earlier with the MFCT tool. Then click "Run update".



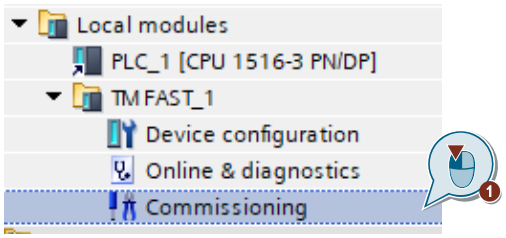
Result:

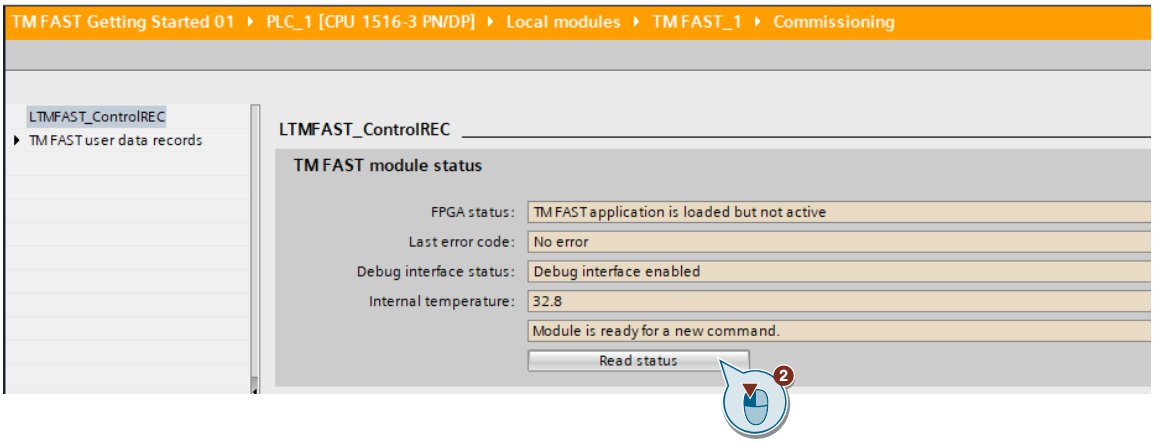
The logic has now been downloaded to the flash memory of the TM FAST, but it is not yet active. This is due to the setting in the TM FAST properties.



Check FPGA status

You can check the FPGA status in the project tree via "Local modules > TM FAST_1 > Commissioning" and the "Commissioning" window, or via the "Test_Handler" tag table.





2.3. Operation

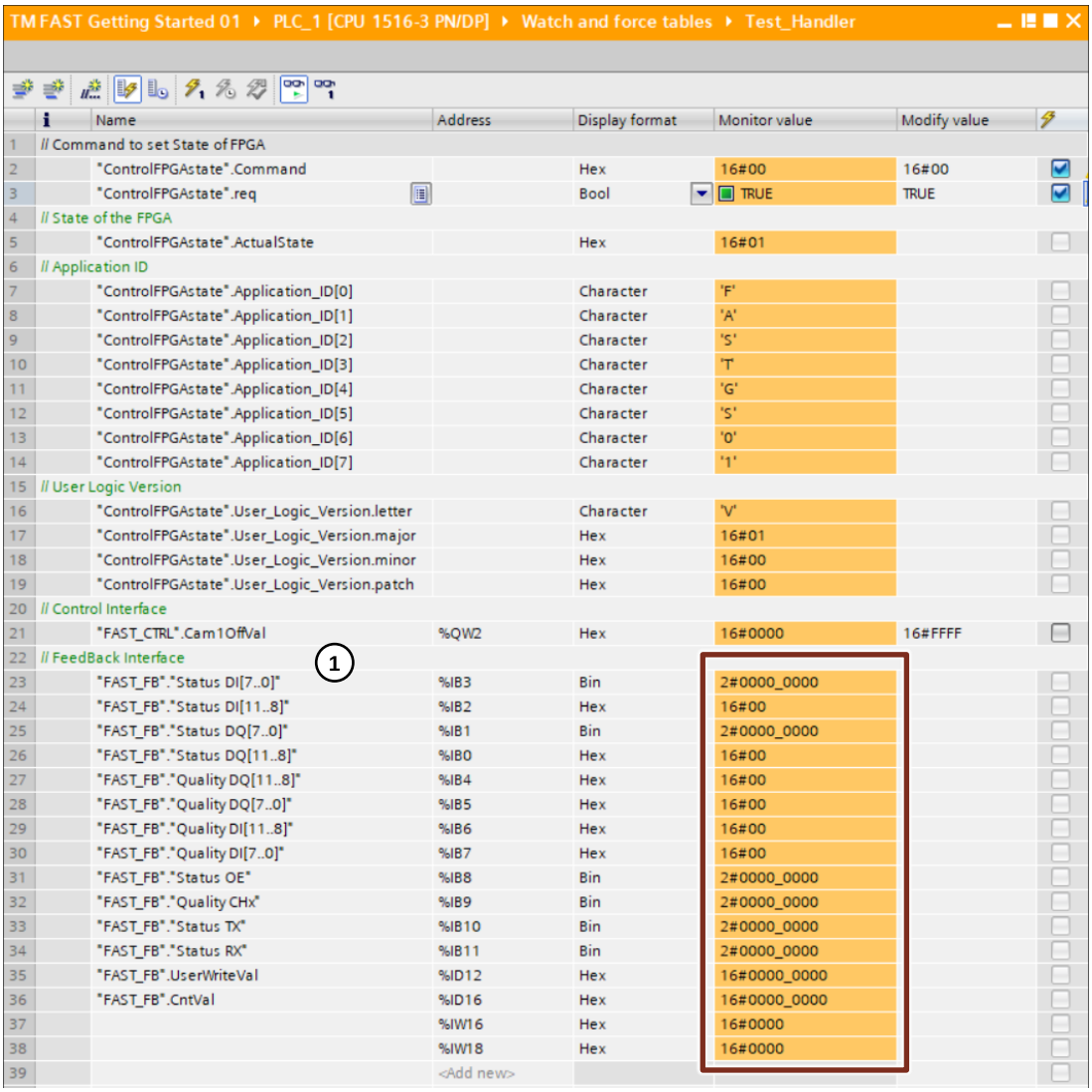
The example is monitored and controlled via the tag table "Test_Handler". It lets you do the following:

- Control the state of the FPGA.
- Set the value of the "Cam1Offval" parameter.
- Read all I/O state values (DI, DQ, RS485).
- Read the "CntVal" counter.

You can manage the TM FAST application and read status information from the TM FAST application by calling the instruction "LTMFAST_ControlREC" in OB1.

Initial state

Open the "Test_Handler" tag table via the project tree.



	Name	Address	Display format	Monitor value	Modify value
1	// Command to set State of FPGA				
2	"ControlFPGAstate".Command		Hex	16#00	16#00
3	"ControlFPGAstate".req		Bool	TRUE	TRUE
4	// State of the FPGA				
5	"ControlFPGAstate".ActualState		Hex	16#01	
6	// Application ID				
7	"ControlFPGAstate".Application_ID[0]		Character	'F'	
8	"ControlFPGAstate".Application_ID[1]		Character	'A'	
9	"ControlFPGAstate".Application_ID[2]		Character	'S'	
10	"ControlFPGAstate".Application_ID[3]		Character	'T'	
11	"ControlFPGAstate".Application_ID[4]		Character	'G'	
12	"ControlFPGAstate".Application_ID[5]		Character	'S'	
13	"ControlFPGAstate".Application_ID[6]		Character	'0'	
14	"ControlFPGAstate".Application_ID[7]		Character	'1'	
15	// User Logic Version				
16	"ControlFPGAstate".User_Logic_Version.letter		Character	'V'	
17	"ControlFPGAstate".User_Logic_Version.major		Hex	16#01	
18	"ControlFPGAstate".User_Logic_Version.minor		Hex	16#00	
19	"ControlFPGAstate".User_Logic_Version.patch		Hex	16#00	
20	// Control Interface				
21	"FAST_CTRL".Cam1OffVal	%QW2	Hex	16#0000	16#FFFF
22	// FeedBack Interface				
23	"FAST_FB"."Status DI[7..0]"	%IB3	Bin	2#0000_0000	
24	"FAST_FB"."Status DI[11..8]"	%IB2	Hex	16#00	
25	"FAST_FB"."Status DQ[7..0]"	%IB1	Bin	2#0000_0000	
26	"FAST_FB"."Status DQ[11..8]"	%IB0	Hex	16#00	
27	"FAST_FB"."Quality DQ[11..8]"	%IB4	Hex	16#00	
28	"FAST_FB"."Quality DQ[7..0]"	%IB5	Hex	16#00	
29	"FAST_FB"."Quality DI[11..8]"	%IB6	Hex	16#00	
30	"FAST_FB"."Quality DI[7..0]"	%IB7	Hex	16#00	
31	"FAST_FB"."Status OE"	%IB8	Bin	2#0000_0000	
32	"FAST_FB"."Quality CHx"	%IB9	Bin	2#0000_0000	
33	"FAST_FB"."Status TX"	%IB10	Bin	2#0000_0000	
34	"FAST_FB"."Status RX"	%IB11	Bin	2#0000_0000	
35	"FAST_FB".UserWriteVal	%ID12	Hex	16#0000_0000	
36	"FAST_FB".CntVal	%ID16	Hex	16#0000_0000	
37		%IW16	Hex	16#0000	
38		%IW18	Hex	16#0000	
39		<Add new>			

Up until this point, the logic of the TM FAST has not yet been activated. Therefore, the values in the "Feedback Interface" (1) area are still all "0".

Activate logic

You can control the status of the logic by using the command line to modify the tags in the tag table in the "Command to set State of FPGA" area.

In the "State of the FPGA" area, you can see the result of the command reflected in the tag <ActualState>.

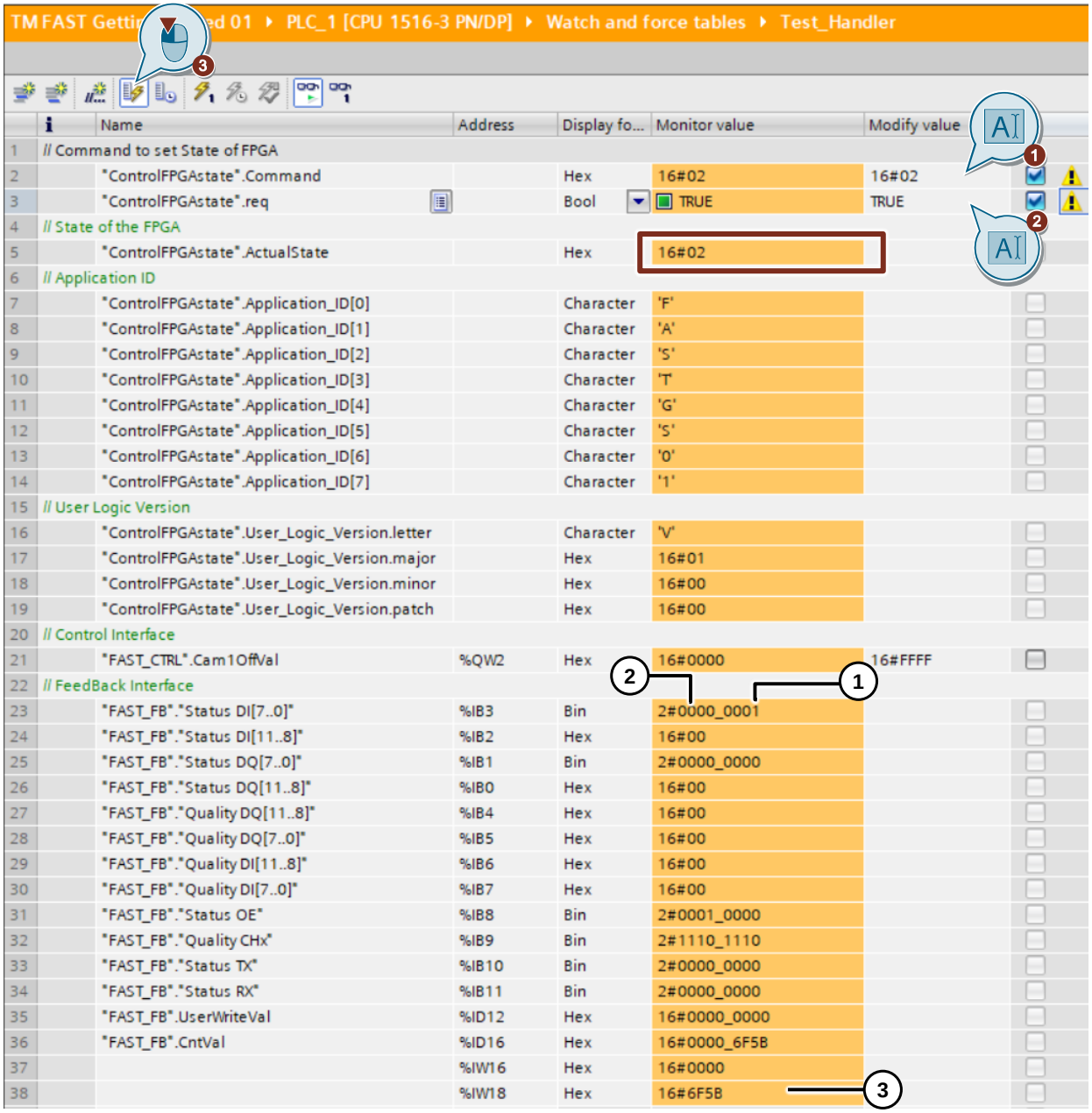
Possible commands are:

- 0 : Read module status
- 1 : Download TM FAST application to the FPGA from flash memory
- 2 : Activating TM FAST application in FPGA
- 3 : Remove TM FAST application from FPGA
- 4 : Delete TM FAST application from flash memory

Possible responses from the command interface are:

- 0 : TM FAST application has been deleted.
- 1 : TM FAST application is loaded, but not yet active.
- 2 : TM FAST application is loaded and active.

1. For the tag <ControlFPGState.Command>, enter the "Monitor value" "16#02" (Activate TM FAST application in FPGA). For the tag <ControlFPGState.req>, enter the "Monitor value" "TRUE" in the "Modify value" column, then activate the modify process.



	i	Name	Address	Display fo...	Monitor value	Modify value
1		// Command to set State of FPGA				
2		"ControlFPGAstate".Command		Hex	16#02	16#02
3		"ControlFPGAstate".req		Bool	TRUE	TRUE
4		// State of the FPGA				
5		"ControlFPGAstate".ActualState		Hex	16#02	
6		// Application ID				
7		"ControlFPGAstate".Application_ID[0]		Character	'F'	
8		"ControlFPGAstate".Application_ID[1]		Character	'A'	
9		"ControlFPGAstate".Application_ID[2]		Character	'S'	
10		"ControlFPGAstate".Application_ID[3]		Character	'T'	
11		"ControlFPGAstate".Application_ID[4]		Character	'G'	
12		"ControlFPGAstate".Application_ID[5]		Character	'S'	
13		"ControlFPGAstate".Application_ID[6]		Character	'0'	
14		"ControlFPGAstate".Application_ID[7]		Character	'1'	
15		// User Logic Version				
16		"ControlFPGAstate".User_Logic_Version.letter		Character	'V'	
17		"ControlFPGAstate".User_Logic_Version.major		Hex	16#01	
18		"ControlFPGAstate".User_Logic_Version.minor		Hex	16#00	
19		"ControlFPGAstate".User_Logic_Version.patch		Hex	16#00	
20		// Control Interface				
21		"FAST_CTRL".Cam1OffVal	%QW2	Hex	16#0000	16#FFFF
22		// FeedBack Interface				
23		"FAST_FB"."Status DI[7..0]"	%IB3	Bin	2#0000_0001	
24		"FAST_FB"."Status DI[11..8]"	%IB2	Hex	16#00	
25		"FAST_FB"."Status DQ[7..0]"	%IB1	Bin	2#0000_0000	
26		"FAST_FB"."Status DQ[11..8]"	%IB0	Hex	16#00	
27		"FAST_FB"."Quality DQ[11..8]"	%IB4	Hex	16#00	
28		"FAST_FB"."Quality DQ[7..0]"	%IB5	Hex	16#00	
29		"FAST_FB"."Quality DI[11..8]"	%IB6	Hex	16#00	
30		"FAST_FB"."Quality DI[7..0]"	%IB7	Hex	16#00	
31		"FAST_FB"."Status OE"	%IB8	Bin	2#0001_0000	
32		"FAST_FB"."Quality CHx"	%IB9	Bin	2#1110_1110	
33		"FAST_FB"."Status TX"	%IB10	Bin	2#0000_0000	
34		"FAST_FB"."Status RX"	%IB11	Bin	2#0000_0000	
35		"FAST_FB".UserWriteVal	%ID12	Hex	16#0000_0000	
36		"FAST_FB".CntVal	%ID16	Hex	16#0000_6F5B	
37			%IW16	Hex	16#0000	
38			%IW18	Hex	16#6F5B	

2. The value of the tag <ControlFPGAstate.ActualState> will then change to state 16#02 (TM FAST application is loaded and active).
   The values in the "Feedback interface" will also change (see Table 1-2), which proves that the logic is active.
3. 1: Alive bit = 1 (DI0)
   2: Output frequency selection, DI6 (here, 0 = 1 kHz)
   3: current count value (0x000 – 0xFFFF)

3. Useful information

The following Tables provide an overview of the assignment of the complete FPGA register to the PLC's process image. The process image in this case is configured starting from address 0 in the HW Config. It also illustrates the conversion of the bit sequence between the TM FAST module and TIA Portal.

PLC (BYTE)	PLC(WORD)	PLC (DWORD)	VHDL ENTITY
QB0	QW0	QD0	CTRL_IF_0 [31..24]
QB1			CTRL_IF_0 [23..16]
QB2	QW2		CTRL_IF_0 [15..8]
QB3			CTRL_IF_0 [7..0]
QB4	QW4	QD4	CTRL_IF_1 [31..24]
QB5			CTRL_IF_1 [23..16]
QB6	QW6		CTRL_IF_1 [15..8]
QB7			CTRL_IF_1 [7..0]
QB8	QW8	QD8	CTRL_IF_2 [31..24]
QB9			CTRL_IF_2 [23..16]
QB10	QW10		CTRL_IF_2 [15..8]
QB11			CTRL_IF_2 [7..0]
QB12	QW12	QD12	CTRL_IF_3 [31..24]
QB13			CTRL_IF_3 [23..16]
QB14	QW14		CTRL_IF_3 [15..8]
QB15			CTRL_IF_3 [7..0]
QB16	QW16	QD16	CTRL_IF_4 [31..24]
QB17			CTRL_IF_4 [23..16]
QB18	QW18		CTRL_IF_4 [15..8]
QB19			CTRL_IF_4 [7..0]
QB20	QW20	QD20	CTRL_IF_5 [31..24]
QB21			CTRL_IF_5 [23..16]
QB22	QW22		CTRL_IF_5 [15..8]
QB23			CTRL_IF_5 [7..0]
QB24	QW24	QD24	CTRL_IF_6 [31..24]
QB25			CTRL_IF_6 [23..16]
QB26	QW26		CTRL_IF_6 [15..8]
QB27			CTRL_IF_6 [7..0]
QB28	QW28	QD28	CTRL_IF_7 [31..24]
QB29			CTRL_IF_7 [23..16]
QB30	QW30		CTRL_IF_7 [15..8]
QB31			CTRL_IF_7 [7..0]

PLC (BYTE)	PLC(WORD)	PLC (DWORD)	VHDL ENTITY
IB0	IW0	ID0	FB_IF_0 [31..24]
IB1			FB_IF_0 [23..16]
IB2	IW2		FB_IF_0 [15..8]
IB3			FB_IF_0 [7..0]
IB4	IW4	ID4	FB_IF_1 [31..24]
IB5			FB_IF_1 [23..16]
IB6	IW6		FB_IF_1 [15..8]
IB7			FB_IF_1 [7..0]
IB8	IW8	ID8	FB_IF_2 [31..24]
IB9			FB_IF_2 [23..16]
IB10	IW10		FB_IF_2 [15..8]
IB11			FB_IF_2 [7..0]
IB12	IW12	ID12	FB_IF_3 [31..24]
IB13			FB_IF_3 [23..16]
IB14	IW14		FB_IF_3 [15..8]
IB15			FB_IF_3 [7..0]
IB16	IW16	ID16	FB_IF_4 [31..24]
IB17			FB_IF_4 [23..16]
IB18	IW18		FB_IF_4 [15..8]
IB19			FB_IF_4 [7..0]
IB20	IW20	ID20	FB_IF_5 [31..24]
IB21			FB_IF_5 [23..16]
IB22	IW22		FB_IF_5 [15..8]
IB23			FB_IF_5 [7..0]
IB24	IW24	ID24	FB_IF_6 [31..24]
IB25			FB_IF_6 [23..16]
IB26	IW26		FB_IF_6 [15..8]
IB27			FB_IF_6 [7..0]
IB28	IW28	ID28	FB_IF_7 [31..24]
IB29			FB_IF_7 [23..16]
IB30	IW30		FB_IF_7 [15..8]
IB31			FB_IF_7 [7..0]

Table 3-1 - Feedback interface

4. Appendix

4.1. Service and support

SiePortal

The integrated platform for product selection, purchasing and support - and connection of Industry Mall and Online support. The SiePortal home page replaces the previous home pages of the Industry Mall and the Online Support Portal (SIOS) and combines them.

- Products & Services
  In Products & Services, you can find all our offerings as previously available in Mall Catalog.
- Support
  In Support, you can find all information helpful for resolving technical issues with our products.
- mySieportal
  mySiePortal collects all your personal data and processes, from your account to current orders, service requests and more. You can only see the full range of functions here after you have logged in.

You can access SiePortal via this address: sieportal.siemens.com

Industry Online Support

Industry Online Support is the previous address for information on our products, solutions and services.

Product information, manuals, downloads, FAQs and application examples - all information is available with just a few mouse clicks: support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.
Please send queries to Technical Support via Web form: support.industry.siemens.com/cs/my/src

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.
For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page: siemens.com/sitrain

Industry Online Support app

You will receive optimum support wherever you are with the "Industry Online Support" app. The app is available for iOS and Android:

4.2. Links and literature

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the application example https://support.industry.siemens.com/cs/ww/en/view/109823442
\3\	Programming manual – Creating a TM FAST application https://support.industry.siemens.com/cs/ww/en/view/109816088
\4\	Equipment manual – TM FAST technology module https://support.industry.siemens.com/cs/ww/en/view/109816087
\5\	Supplementary software components https://support.industry.siemens.com/cs/ww/en/view/109817062
\6\	MultiFieldbus Configuration Tool (MFCT) https://support.industry.siemens.com/cs/ww/en/view/109773881
\7\	Hardware catalog support package for integrating the TM FAST into TIA Portal V17 https://support.industry.siemens.com/cs/ww/en/view/72341852
\8\	Intel® Quartus® Prime http://www.intel.com/quartus

Table 4-1

4.3. Change documentation

Version	Date	Change
V1.0	09/2023	First edition

Table 4-2