UM2092
User manual

Basic metrology firmware for the STM32F103RD and the STPM32 devices

Introduction

The following document describes a firmware for the STM32F103RD microcontroller to manage the STPM32 metrology device. This firmware can be tested using two evaluation boards, the STEVAL-IPE023V1 and the EVALSTPM32.

The package includes:

- The initialization of the STM32F103RD
- The reset sequence of the STPM32
- The communication through UART or SPI
- The block transfer in read and write mode
- The data latch through SYN pin
- A basic metrology application, based on the STM32F103RD timer, to read energy and voltage

The STM32F103RD initialization uses the Cube MX tool chain.

The STPM32 driver integration is based on IAR tool chain with I-JET pod.

For further reference please refer to STPM32 and STM32F103RD datasheets.

Contents

1 Cube MX configuration

The STEVAL-IPE023V1 contains the STM32F103RDT microcontroller, configured as follows:

- RCC high speed clock HSE (crystal/ceramic resonator) and RCC high speed clock LSE (crystal/ceramic resonator)
- SYS debug JTAG (5 pin)
- SPI 1 is used in full duplex mode with hardware NSS signal disable
  - Frame format: Motorola
  - Data size: 8 bits
  - First bit: MSB first
  - Clock polarity: high
  - Clock phase: 2 edge
  - CRC calculation: disabled
- TIM3 clock source internal clock
- USART3 in asynchronous mode with hardware flow control disable
  - Baud rate: 9600 bits/s
  - Word length: 8 bits
  - Parity: none
  - Stop bits: 1
  - Data direction: receive and transmit
- TIM3 is used for basic metering application
- SPI and USART configuration depends on the topology used

2 STPM32 configuration

2.1 Communication mode

The STPM32 is configured during the startup in SPI or in UART mode.

If the SCS is set low and EN pin goes from low to high, the STPM32 communicates by SPI mode. If the SCS is set high and EN pin goes from low to high, the STPM32 communicates by UART mode. The STPM32 EN pin can be tied to VDD thanks to a pull-up if it is not driven (EVALSTPM32 board). In this case, no additional wire between the STM32F103RD and the STPM32 is needed.

The STM32F103RD pins are in input floating mode at the startup, so on SCS is needed:

- Pull-down network to communicate by SPI
- Pull-up network to communicate by UART

In the FW, there is not any code related to this since it is pure HW mechanism.

Figure 1: UART communication peripheral selection

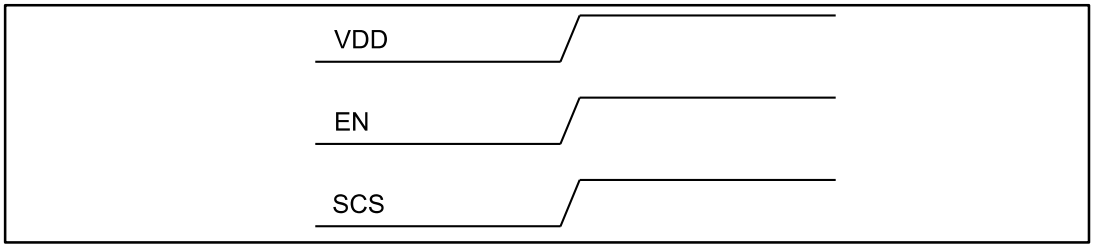
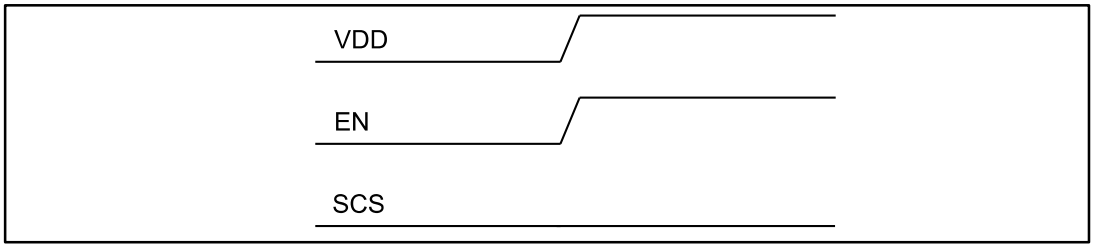


Figure 2: SPI communication peripheral selection



EN pin can be also driven by FW:

- Put EN low to reset the configuration
- Set the desired level on SCS (for SPI or UART)
- Put EN high level to latch the configuration

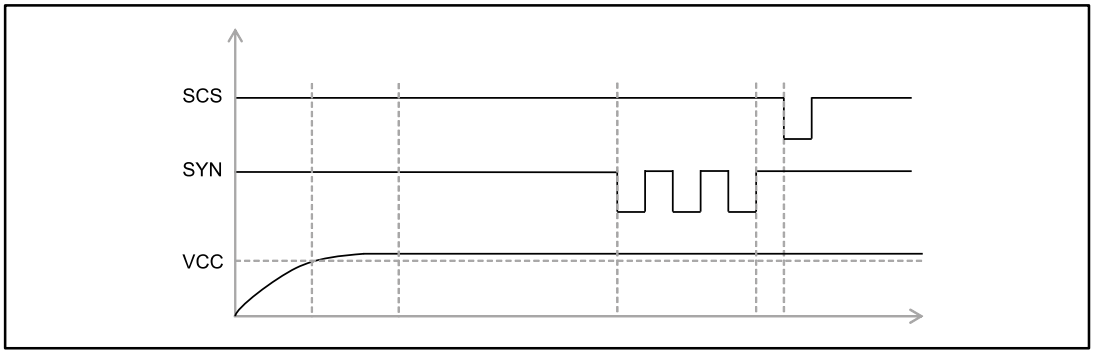
In the FW, this service is included into Metro_Init() and it is called Metro_power_up_device().

The choice of the GPIO is in st_device.h.

2.2 Reset

As mentioned in the STPM32 datasheet, after the POR, the chip must be reset by toggling 3 times SYN pin and once SCS pin.

Figure 3: Reset signal



In the FW, the complete sequence is in the function Metro_Init().

2.3 SPI and UART communication modes

In SPI mode the communication speed is based on the SPI clock.

In UART mode, by default, the STPM32 works at 9600 bauds. In order to speed up the communication, please use Metro_UartSpeed(uint32_t baudrate) function to change the baud rate.

2.4 Latch data measure

The STPM32 registers can be latched in different ways:

- By FW, by setting latch bits in DSPCTRL3 register
- By HW, using SYN pin (this method is used in the FW)

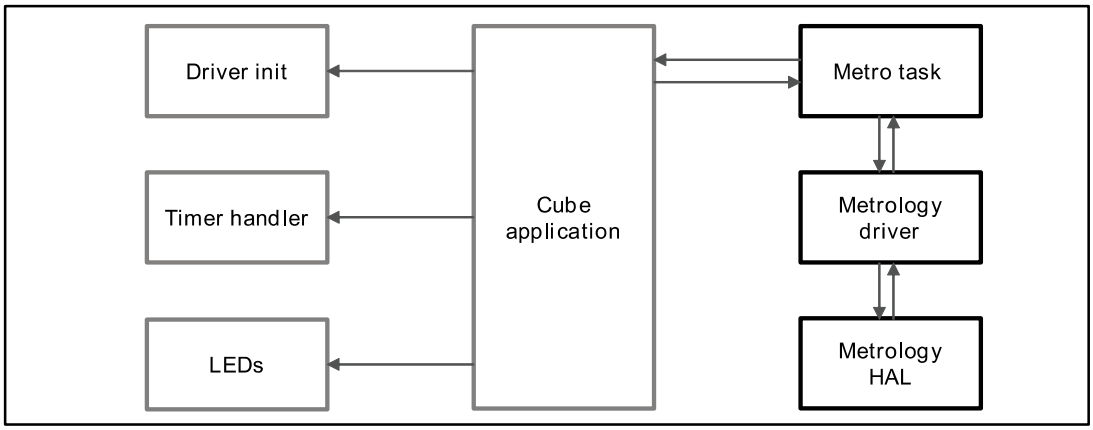
The choice of the method depends on the function uint8_t Metro_Set_Latch_device_type (METRO_NB_Device_t in_Metro_Device, METRO_Latch_Device_Type_t in_Metro_Latch_Device_Type).

3 Firmware block diagram

The metrology package includes:

- STPM32 reset management
- STPM32 latch configuration
- STPM32 latch management
- STPM32 read and write access
- STPM32 computation for metrology measurement (in basic application energy and RMS measure)

Figure 4: Firmware block diagram

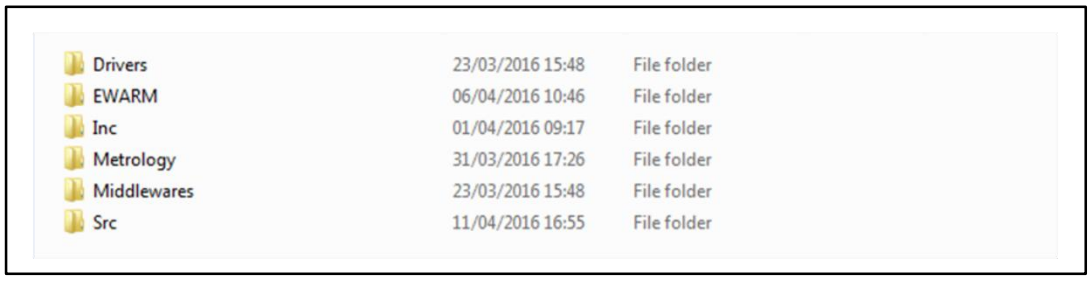


3.1 Integration with IAR tool chain

The firmware has been developed for the STM32F103RD, but it can be easily ported to other STM32 microcontrollers with minor changes. The microcontroller part number is selected into the IAR configuration and in main.c.

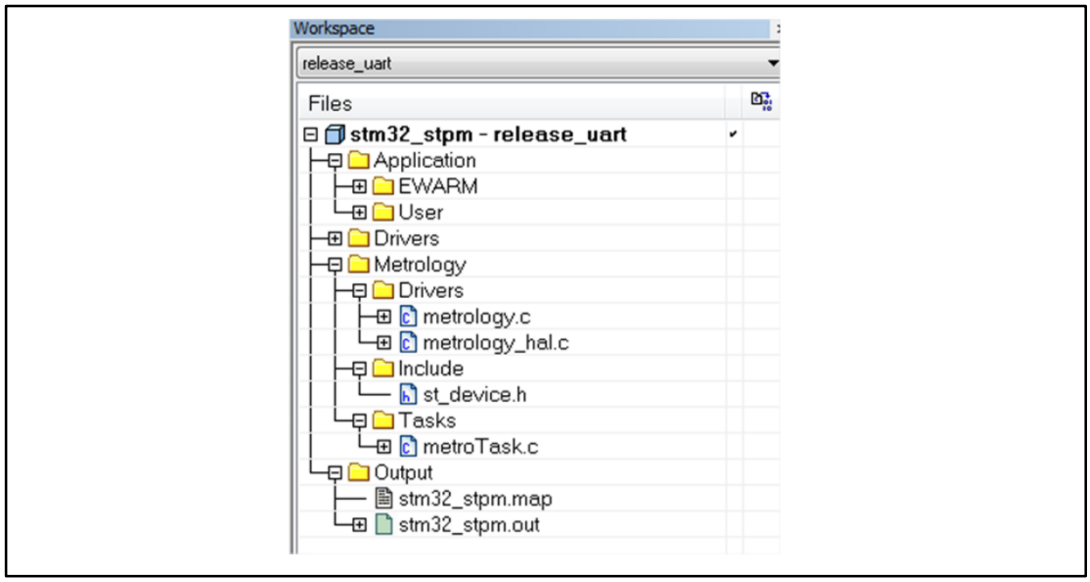
The metrology package is put at the same level as the other folders in Cube.

Figure 5: Folder organization



In the IAR tool chain, folders and files need to be added. The following organization is used:

Figure 6: IAR file organization



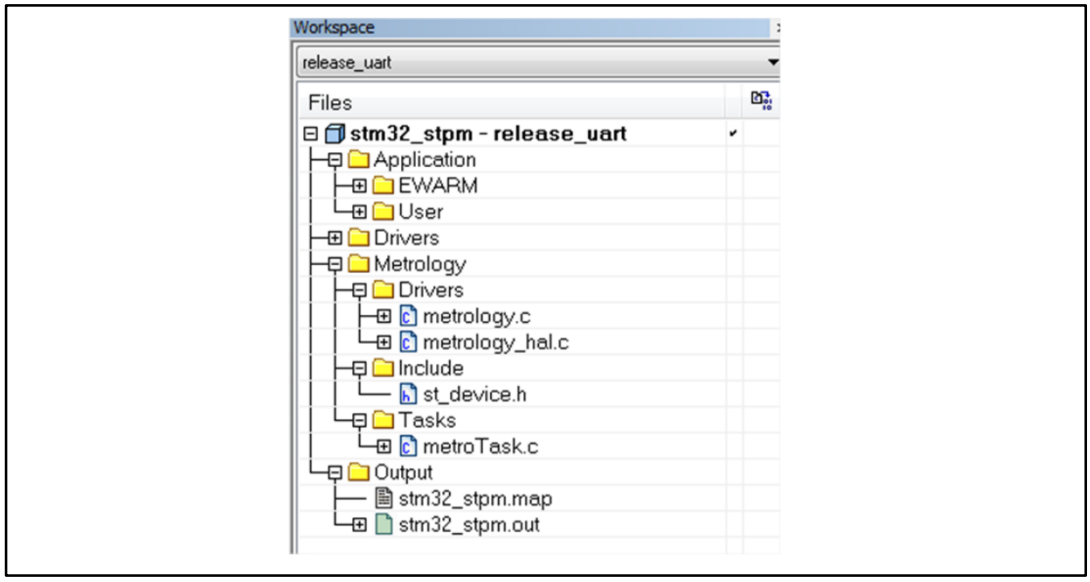
After the integration of files into the project, they need to be included.

The following symbols must be defined for the compiler:

- SPI_XFER_STPM3X for SPI communication
- UART_XFER_STPM3X for UART communication

It is possible to choose directly the configuration in the project by selecting the right compile flag.

Figure 7: SPI/UART release



There are two sections in which C/C++ compiler and assembler options can be updated.

Figure 8: Compiler update

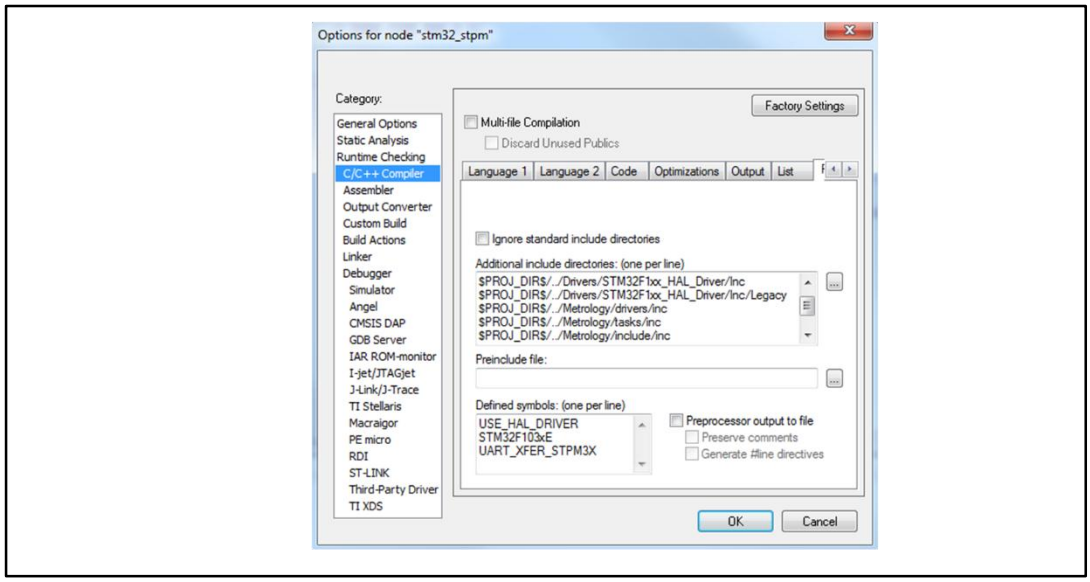
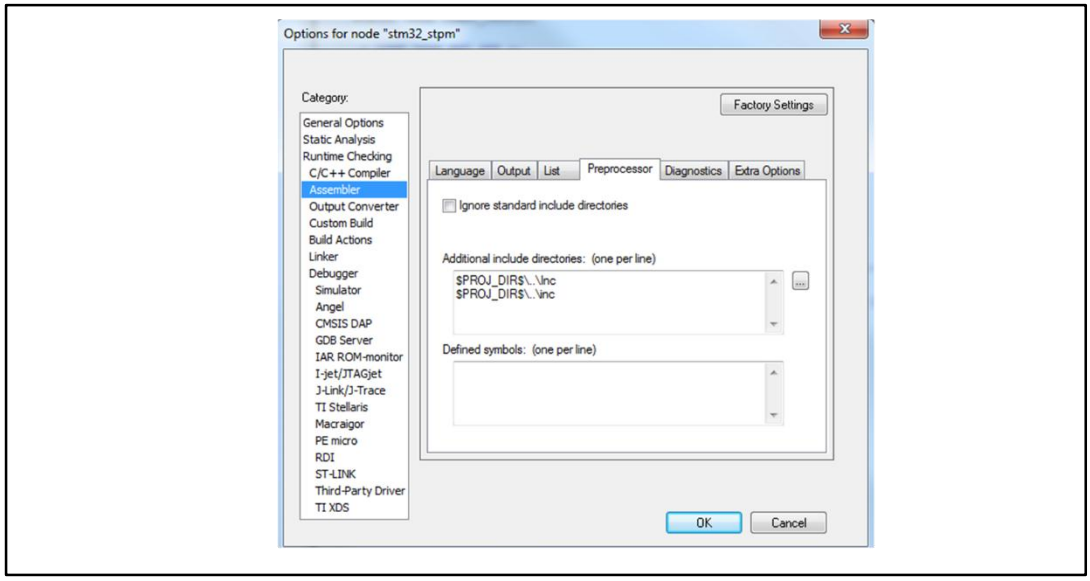


Figure 9: Assembler update



Into main.c file the configuration functions must be changed to be adapted to the HW configuration.

3.2 St_device.h

This file includes all #define used mainly for the communication, for example:

- SPI used
- UART used
- UART speed after initialization
- CS and SYN pins

This file must be updated according to the CUBE MX definitions for the specific STM32 part number .

3.3 Metro Task.c

This file includes the functions:

- METRO_Init(): this function initializes the STPM3x (POR sequence), configures the STPM3x by writing internal registers and selects the latch type
- METRO_Update_Measures: it reads the data from the STPM3x
- METRO_latch_Measures: it handles the STPM3x latch

3.4 STPM3x configuration

The FW handles up to four different STPM3x devices, but in the present deployment one device only, called EXT1, is defined.

Besides, the code can support different STPM3x devices (STPM32, STPM33, STPM34).

The STPM3x configuration register settings are defined into metroDefault (handler_metrology.c).

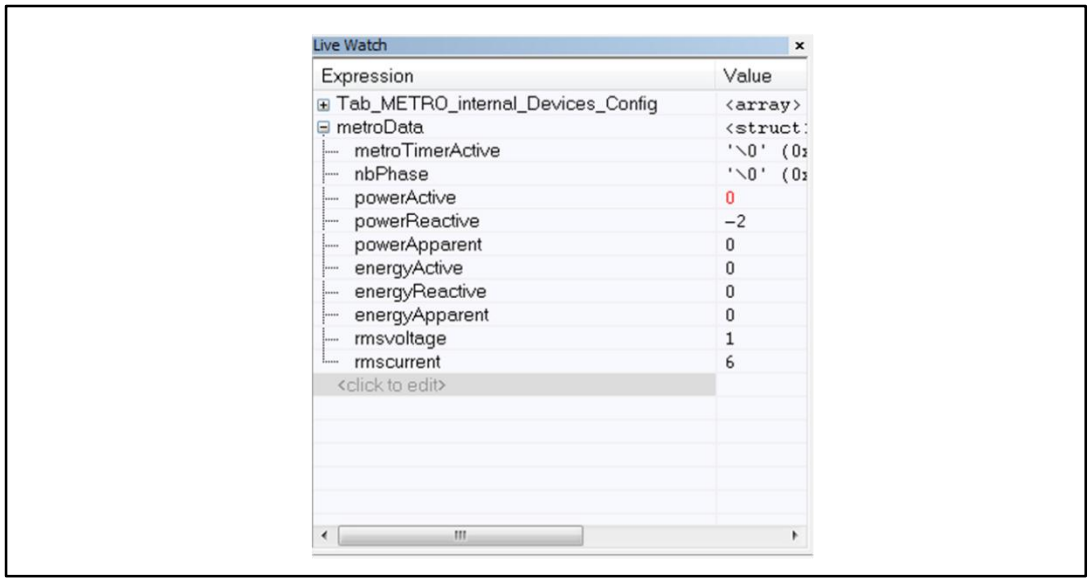
- The first U32 word is the STPM3x type and number of supported channels
- 19 x U32 words contain the settings to write into DSPCTRL registers
- 2 x U32 words contain the power conversion factor for conversion of register value to Watt. The value depends on the AFE components of the application
- 2 x U32 words contain the voltage conversion factor for conversion of register value to Volt
- 2 x U32 words contain the current conversion factor for conversion of register value to Ampere

3.5 Data with Live Watch

Data can be read in live mode debug using "Live Watch" view.

By adding the variables, Tab_METRO_internal_Devices_Config and metroData to "Live Watch" window, the raw registers can be monitored from the STPM32 and the measurement data calculated by the basic metrology application.

Figure 10: Live Watch



4 STEVAL-IPE023V1 board update

4.1 SPI configuration

In order to have the STPM32 available for the communication, a 10 k pull-down resistor can be placed on SPI1-NSS line (host side).

4.2 UART configuration

The STEVAL-IPE023V1 board does not support by default the UART mode.

A simple UART communication can be set up, but in this case the metrology board works without galvanic isolation.

To enable UART communication, on the STEVAL-IPE023V1 board:

- Remove the connection STPMxx_MOSI between the STPM32 connector and U5
- Remove the connection STPMxx_SDA_MISO between the STPM32 connector and U3/U4
- Connect both of GND (isolated and no-isolated)
- Connect J3-2 to PB10/USART3-TX
- Connect J3-4 to PB11/USART3-RX
- Place 10 k pull-up on SPI1-NSS line

To isolate from mains, place an isolator over RX and TX line and no connection between the two GND.

5 Revision history

Table 1: Document revision history

Date	Revision	Changes
26-Jul-2016	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.