SENSOR SYSTEMS

STM32

M01 – Installation guide

PDF on how to install STM32CubeMX, Keil and the drivers (St-Link Driver)

Notes: none.

M02 – ARM Microcontrollers

Theory.

Introduction to:

- Generic ARM microcontrollers
- NUCLE-F401RE, microprocessor and peripherals
- PoliMi expansion board

Notes: needs to be printed.

M03 – First trial

Briefly explain how to setup a project in Cube and Keil. No actual code, just hardware check and a generic explanation of the different functions.

After making a blank project, compile and run the code in debug mode. The red LED on the bottom (LDSK1) should be on.

Notes: none.

M04 – GPIO and interrupts

HAL functions:

- HAL_GPIO_ReadPin
- HAL_GPIO_WritePin
- HAL_GPIO_TogglePin
- HAL_GPIO_EXTI_Callback

Interrupts contained in the file **stm32f4xx_it.c**

Projects:

1A: pushbutton – polling

Objective of this project is to switch on the green LED on NUCLEO board (LD2), every time the blue pushbutton is pressed. A polling operation will be used to monitor the state of the pushbutton.

1B: Pushbutton -interrupt

Objective of this project is to switch on an LED every time the blue pushbutton is pressed, and to switch it off when the pushbutton is released. The LED input will be used in interrupt mode.

M05 – Timers

Timers structure

PWM explained

Registers

Timer HAL functions:

- HAL_TIM_PWM_Start
- HAL_TIM_PWM_ConfigChannel
- HAL_TIM_PWM_Stop
- HAL_TIM_OC_DelayElapsedCallback

Projects:

1C: Blinking LED – PWM

Objective of this project is to blink the NUCLEO board green LED at 1 Hz with 50% DC, using a PWM.

M06 – USART

Description of the USART protocol

USART HAL functions:

- HAL_UART_Receive
- HAL_UART_Transmit

Projects:

2A: Sending text via USART

Objective of this project is to send information from the microcontroller to the PC, using the USART interface for the Virtual COM.

You will send a string containing your name and your year of birth followed by a new lineevery one second.

M07 – ADC

- ADC features:
  - What type: SAR
  - How many bits: 12
  - Right/left alignment
  - DMA
- ADC block diagram
- CUBE configurations
- ADC HAL functions

Projects:

3A: ADC started by software

Objective of the project is to acquire the voltage of the potentiometer on the POLIMI board, starting the conversion by software and then send the value to PC on a remote terminal.

3A-I: ADC single acquisition polling

Objective of this project is to acquire the voltage of the potentiometer every 1 second and send this value to a remote terminal. The ADC will be used in polling mode.

3A-II: ADC single acquisition interrupt

Objective of this project is to acquire the voltage of the potentiometer and send this value to a remote terminal every 1 s. The ADC will be used in interrupt mode.

3B: ADC triggered by TIM

Objective of the project is to acquire the potentiometer voltage using a timer to trigger a conversion at a regular conversion rate of 1 Hz.

4: Light Dependent Resistor

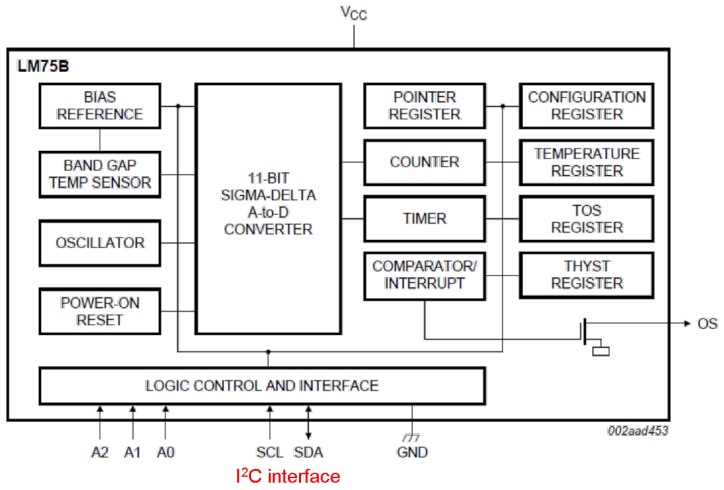
Objective of the project is to acquire the signal from the LDR (by Sunrom technology) on the POLIMI board. Convert one value every 1ms and store 1,000 values using the DMA. Every 1s, display on a remote terminal the average light power (expressed in lux) of the previous second.

3B: ADC scan using DMA

Objective of the project is to acquire 3 voltages (potentiometer, temperature sensor, Vref) every 1 s and to send them to a remote terminal. The acquisition are started by software and data are saved in the microcontroller memory using DMA.

M08 – Temperature sensor

- Temperature sensor on PoliMi board:



- LM75B description
- I2C interface
- I2C protocol
- ADC HAL functions

Projects:

5A: Temp MSB

Read the temperature measured by the LM75 and send it to a remote terminal every 1 second. As a first step we will read only the MSB 8 bit.

5B: Temp FSR

Now we will modify the code to read all 11 bits

M10 – Accelerometer

We only have the explanation of the project.

Acquire the x, y, z acceleration and send the 3 values to a remote terminal, about every 2 seconds.

The value of acceleration must be expressed in g, with a precision of 0.01 g and the correct sign.

The full-scale range must be +/-2 g.

The output format of the data must be like the one in the example:

X: + 0.05 g

Y: -0.22 g

Z: + 1.00 g