Summer practice project

*Guidelines*

Contents

[1. Introduction. Used tools. Needed Docs. (EF - 1 day) 2](#_Toc517015773)

[2. A Tutorial Introduction. GPIO (EF – 3 days) 2](#_Toc517015774)

[3. ADC and Timers 3](#_Toc517015775)

[4. BLDC 4](#_Toc517015776)

# Introduction. Used tools. Needed Docs. (EF - 1 day)

- Present board (uC, pin layout, debugger)

- provide ide, drivers (st-link)

- Present ide(create new C-only project, build, test run )

- Total commander?

- In the time remained give a few C exercises

# A Tutorial Introduction. GPIO (EF – 3 days)

* BOM:
  + 1x STM Nucleo-64
  + 1xMiniUSB cable
  + 1x breadboard
  + 2x Push button
  + 8x LED
  + 1xCap(value TBD)
  + 1xRes(value TBD)
  + Connection cables
* Scope:
  + get familiarized with the IDE. Start new arm proj. add the hal drivers
  + reading the mcu data sheet
  + init the board with proper settings (wrinting all the config code. **NO HAL**) : clock, buses, GPIO
  + organize code. No magic numbers. Specific variable names. Code modularization.
  + COMMENTS. as detailed as possible
  + downloading binary.
  + Intro into debugging
* TASKS:

**TASK\_1: Writing a small program that shall output on 1 port pin (*properly* connected to a LED).** Small steps:

* + Init the mcu clock at 84 Mhz (RCC)
  + Set up one pin on one port as output(GPIO).
  + Create a dummy delay function(loops)
  + Connect led. Test the software “Blinking led”

**TASK\_2** :**Bringing buttons into the mix.** **Write a small program shall read 2 port pins (coming from 2 buttons),**

* + Configure 2 pins on the same ports (if possible) as input.
  + Connect buttons. Test the software.
  + *TO KEEP IN MIND*:
    - the buttons need to be debounced; at least 2 software solutions, (extra) 1 hardware solution.
      * Extra tools needed (osc, 2 cap, 2 res)
    - Implement 3 logic operations : AND, OR and XOR

**TASK\_3** :**More on gpio.** Connect 8 leds, and 1/2 buttons.

* Output on leds integers/characters in binary – enforcing to create a function that converts int/char to binary. *Also creating a “driver” that outputs on the selected pins, further used for the LCD.*

# ADC and Timers

* + - 1x STM Nucleo-64
    - 1xMiniUSB cable
    - 1x breadboard
    - 2x Push button
    - 1xPot 10k
    - 8x LED
    - Connection cables

**TASK\_4: Configure the ADC to read a value from a pot.**

* The uC datasheet shall be massively consulted
* Config the adc in free running mode. Wait for the conversion to finish then fetch the converted value
* Depending on the pot value turn on/off in a seq the 8 leds

**TASK\_5: Configure a TIM output on a pin a PWM signal with a x kz frequency.**

* The uC datasheet shall be massively consulted
* Configure the TIM in free running mode. Enable interrupt. On each cnt update, update the dtc
* Update the dtc with the converted pot adc value(properly scaled)

**TASK\_6: Configure a TIM output on 6 pins a PWM signal with a xkz frequency.**

* The uC datasheet shall be massively consulted
* Configure the TIM in free running mode. Enable interrupt. On each counter update, update the dtc
* Update the dtc with the converted pot adc value(properly scaled)
* Emulate “volume” leds

# BLDC

* Create driver to read from hall sensors(using gpio)
* Create LKT with Phase vs Hall state correlation
* Configure TIM1 as pwm driver to command the motor. 10kHz
* Create ComutationHandler. Function in pwm\_driver that shall

read the hall state and output pwm on the correct phases with the correct polarity according with the LKT

* Config a timer as input capture. Connect H1 to it, in order to use it as a speed sensor.
* Implement a PI speed controller