## ED5502 (Specialist Diploma) Spring 2019 Project 1 (Weeks 4-8)

(Version 1.1, 22 Feb 2019)

This project is worth **20%** of the total module assessment.

**General**

This project is designed to be run on an Nucleo64-STM32L476RG board and an added Arduino UNO Shield with a LED Array, two Pushbutton Switches, and a simple potentiometer as an ADC input source. The connections on the Shield board are as follows:

The LEDs are connected as follows:

LED0, LED1 not connected

LED2 - D2 - PA10

LED3 - D3 - PB3

LED4 - D4 - PB5

LED5 - D5 - PB4

LED6 - D6 - PB10

LED7 - D7 - PA8

LED8 - A5 - PC1

LED9 - A4 - PC0

The Pushbutton Switches on the board are connected as

S1: D12 – PA6

S2: D13 – PA5 – Not used in this version of the project

The Potentiometer is connected to PA0

==========================================================================

**Project Tasks:**

Write a program to do the following tasks in the main program loop:

Read the state of switch S1.

S1 at logic ‘1’

If the input state of S1 is at logic ‘1’ level, then flash the LEDs based on the state of a variable named DisplayMode, at a rate of 0.25s per LED.

If the DisplayMode is logic ‘1’ then turn ON one LED at a time moving forward and back in Cylon fashion.

If the DisplayMode is logic ‘0’ the turn OFF one LED at a time, with all other LEDs ON, again moving forward and back in Cylon fashion.

At the end of the LED display sequence in either case, send a message to the user PC on USART2, the message should be ‘LED Cylon Display’. The end of the display sequence is found when LED2 has been changed.

S1 at logic ‘0’

Read the result of the ADC conversion of an analog voltage on pin PA0. The ADC will return a 12-bit value in the range 0 – 4095.

Display an output on the LED array based on the ADC result, as follows:

|  |  |
| --- | --- |
| ADC Result | LED Array Setting |
| 0 – 511 | 0 0 0 0 0 0 0 1 |
| 512 – 1023 | 0 0 0 0 0 0 1 1 |
| 1024 – 1535 | 0 0 0 0 0 1 1 1 |
| 1536 – 2047 | 0 0 0 0 1 1 1 1 |
| 2048 – 2559 | 0 0 0 1 1 1 1 1 |
| 2560 – 3071 | 0 0 1 1 1 1 1 1 |
| 3072 – 3583 | 0 1 1 1 1 1 1 1 |
| 3584 – 4095 | 1 1 1 1 1 1 1 1 |

After each conversion, send a message to the User PC on USART2 showing the ADC result.

‘ADC Result = xxxx’ where ‘xxxx’ is the ADC result read from the ADC.

**Settings:**

Set the Port pins used for LED outputs as General-Purpose Output Pins.

Set the Port pins used as Switch inputs as General-Purpose Input Pins.

Set PA0 as an Analog Pin

Set USART2 Baud Rate to 115200 Baud, 8 data bits, 1 stop bit, no parity

Set PA2 as an output pin, set PA3 as input pin

ADC Settings will be defined after next lecture.

For sending data to the PC, I suggest using the techniques in the Mazidi program 4\_1 (modified for STM32L476RG). To display the ADC result you may like to use the C library functions itoa().

**Code Structure**

Initialisation Section:

Initialise relevant variables

Initialise Port settings

Initialise USART2

Initialise ADC (after next lecture)

while(1)

{

Read S1

If S1 set

{

If DisplayMode == Normal

Display Cylon pattern (moving one) on the LED array

Else

Display Inverse Cylon pattern (moving zero) on the LED array

If DisplayMode == Normal

DisplayMode = Inverse

Else

DisplayMode = Normal

Send a message to the User PC on USART2

}

Else /\* S1 clear \*/

{

Start the ADC using a software trigger

Wait for the conversion to be completed

Read the ADC conversion result

Display a pattern on the LED array based on the conversion

Send a message providing the ADC result to the User PC on USART2

}

**Style:**

Please structure your programs for readability and use functions where appropriate

**Marking Scheme:**

Initialisation (including variables) and overall programming style: 4%

LED Array (Cylon): 4%

PC Display: 4%

ADC conversion: 4%

PC Display of ADC result: 4%

(You may use STMCubeMX to set up the GPIO, USART2 and ADC options but ideally I would like you to write your own initialisation functions for these on-chip peripherals, ie for full 4% initialisation).

**Timetable:**

Demonstrate your program running on the STM32L476RG + shield in the lecture/lab session in week 8.

Submit the final versions of your program using Sulis, before 5pm, Wednesday 13 March 2019.