***Efficient Embedded Course***

**ASSIGNMENT**

**INTERRUPT PROJECT:**

**HUMAN RESPONSE TIMER**

**Issue 1.0**

Contents

[1 Introduction 1](#_Toc78805198)

[1.1 Lab overview 1](#_Toc78805199)

[2 Requirements 1](#_Toc78805200)

[3 Details 2](#_Toc78805201)

[3.1 Hardware 2](#_Toc78805202)

[3.2 Software 2](#_Toc78805203)

[3.3 Testing 3](#_Toc78805204)

[3.4 Extra Credit Options 3](#_Toc78805205)

# Introduction

## Lab overview

For this project, you will create a device which measures how quickly a person can press a switch in response to an LED being lit. This will give you an idea of how much work the processor can do in the time it takes you to react to an event.

# Requirements

In this lab, we will be using the following hardware and software:

* **Keil µVision5 MDK IDE**
  + Please see the included Getting Started with Keil guide on how to download and install Keil.
* **STM32 Nucleo-L552ZE-Q**
  + For more information, click [here](https://www.st.com/en/evaluation-tools/nucleo-l552ze-q.html).
* **Logic Analyzer or Oscilloscope** 
  + Required to monitor the interrupt signals

# Details

## Hardware



Figure 1. Schematic diagram

Use a switch SW1 to control the system. Use the RGB LEDs as the output device.

Please see the Nucleo-L552ZE-Q User manual for the pinout of the Arduino-included Zio connectors for CN7, CN8, CN9 and CN10 using this link: https://www.st.com/resource/en/user\_manual/um2581-stm32l5-nucleo144-board-mb1361-stmicroelectronics.pdf

## Software

The main code should perform the following:

* Initialize peripherals
* Repeat the following
  + Turn off all LEDs
  + Clear counter
  + Wait a random amount of time (e.g. within 1-3 seconds)
  + Turn on one LED
  + Repeat until ISR has been triggered, as indicated by the flag being set
    - increment counter
  + Save counter value in memory
  + Wait for approximately 5 seconds

The ISR should perform the following:

* Set a flag indicating the ISR has executed

You will also need some support functions:

* Use the **leds.c** module to initialize and control the RGB LEDs.
* Use the C standard library function **rand()** to generate a random integer.
* Use the **delay\_ms** function provided by **delay.c** to wait for a number of milliseconds.

## Testing

To see the number of iterations counted, set a breakpoint in your main function after the switch press has been detected and examine the counter variable using the watch window.

## Extra Credit Options

Here are some ideas for extra credit:

* Modify the code to average ten response time measurements.
* Determine if the user’s response time changes depending on which color the LED generates (red, green, blue, orange, cyan, magenta, white).
* Determine if user’s response time is the same when the LED turns on vs. turns off.
* Determine how long it takes the user to recognize a color (e.g. green) when multiple possible colors may be generated. Modify the program to randomly generate one of the seven possible colors and have the program track the response time.