ECEN-361 Lab-01:STM32Cube IDE Introduction

NAME: _____

Introduction and Objective of the Lab

With the change away from the TI-MSP432 board (obsoleted) there are likely some in this class who are unfamiliar with the development environment comes with the new board to be used: STMicro's STM32-Nucleo-F476RG. Both boards are centered on an ARM-Cortex processor and most of the fundamental principles learned in the pre-requisite ECEN-260 are unchanged.

Similarly, the integrated development environment (IDE) will necessarily change from T.I.'s Code Composer Studio, to STMicro's customized one: STM32CubeIDE. Both of these IDEs, however, are based on the open-source Eclipse framework and the transition should be very simple.

The objectives of this first lab are as follows:
- Part 1: Install, run, and do a first example project with STMCubeIDE
- Part 2: Become familiar with some of the Hardware Abstraction Level (HAL) routines provided

For each of the parts, follow the instructions, then fill in answers to the questions. Expected answers are indicated in the boxes with red text/spaces to fill in answers.

The submission for this lab is simply this document, with your responses filled in. NO CODE needs to be submitted.

Instructions

Part 1: First Program with STM32CubeIDE

STM32 CubeIDE needs to be installed on your personal computer. Rather than go thru the instructions in this document, the first part of this lab is for you to replicate the steps shown in a YouTube video:

Note that it may be easier for you to replicate the steps in this video by following the written instructions HERE.

When you complete the installation, and do the first "blinky" program, you should have a board that blinks the LED on/off.   Experiment with the various toolbar buttons:

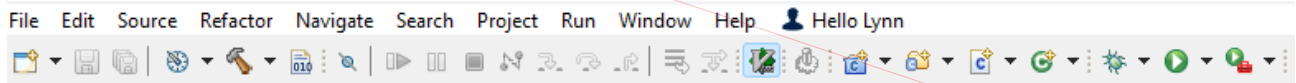When completed answer the following questions:

Platform used (PC/Mac/Linux): _____

Version of STM32CubeIDE installed: _____

What GPIO Pin is tied to the LED on the board?: _____

At what frequency did your light blink?: _____

What does the "bug" button do?

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help   👤 Hello Lynn

Part 2: Loading/Running a pre-built project

For this part of the lab, you won't need to write any code, but you'll use the STM32CubeIDE to load a project already built for you, and then use that code to test out the MultiFunction add-on board. We'll use this board for many other labs.

The MultiFunction add-on board uses an Arduino shield compatible form factor to add switches, 7-segment LED digits, a buzzer, potentiometer, and more LEDs. Documentation on the board is found in the iLearn module (**Course Resources/MultiFunction Board Docs**)

Steps:
1. Make sure your Nucleo Board is not powered on
2. Plug in the MultiFunction board into the mating CN8 & CN9 connectors on top of the Nucleo.
3. Install the MultiFunction_Example_Code. This is done by:
   a. Accept the lab assignment in the GitHub Classroom. This will make a customized repository for you to clone: If you're reading this document, it's likely you've already been successful cloning the lab repo.   More can be found here: "Working with Labs from Github Classroom.
      Open STM32CubeIDE
   b. Note the directory into which you cloned the repository. Bring it into your STM32CubeIDE workspace with : File/Import/<your directory name>
4. Plug in the board to the USB, clean/build/run the project.
5. Experiment with the buttons, note the operation. Look at the code in :
   Core/Src/main.c to help with the questions that follow.
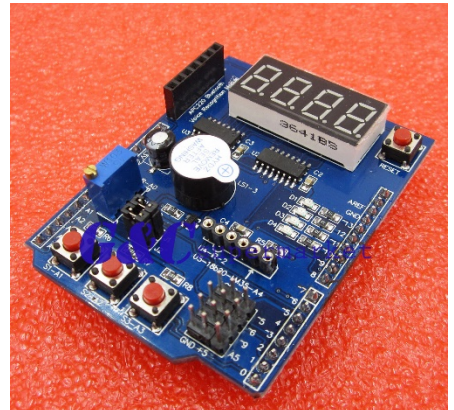
6. Open the project's configuration file GUI (ECEN-361-STM32-Lab-01-multiboard.ioc). You should see a graphical representation of the chip along with name definitions, and other parts inside the chip that have been programmed: Timers, Interrupts, etc.
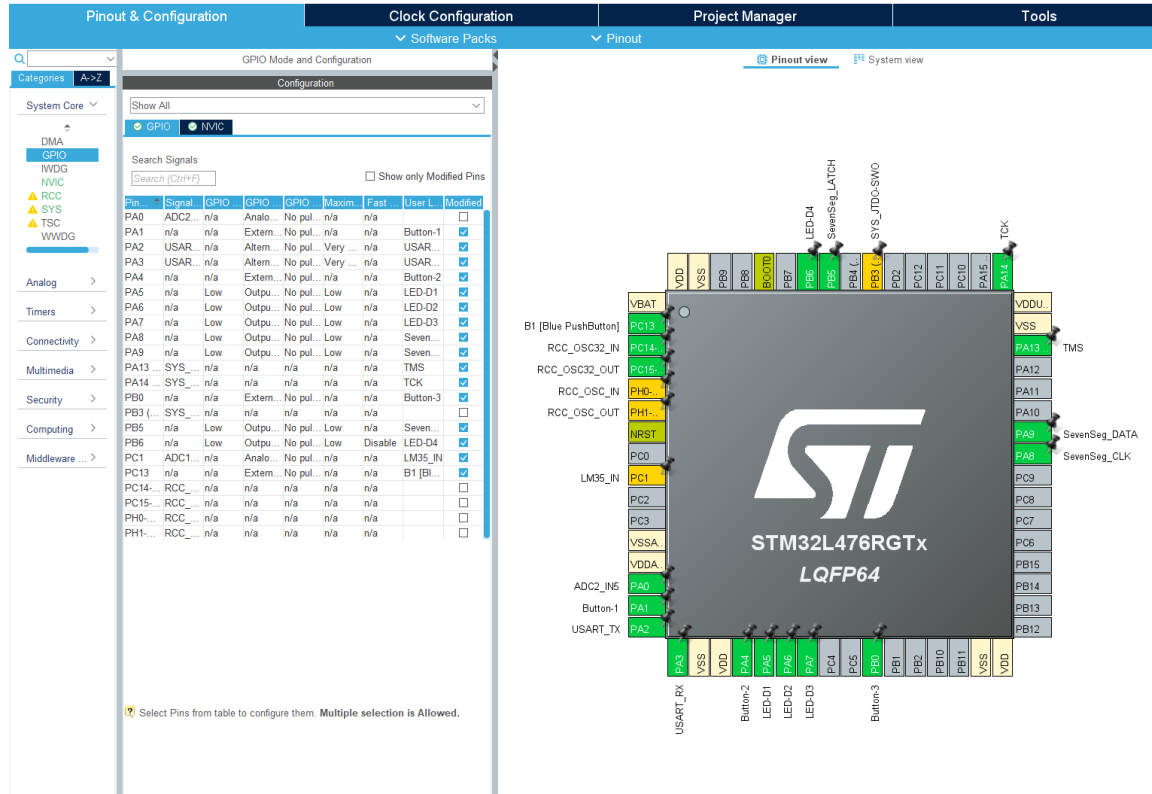
Look through these definitions and compare them to the code that gets generated automatically in:

```
main.c,                 (main execution loop)
stm32l4xx_it.c          (interrupt handlers and config), and
stm32l4xx_hal_msp.c     (hardware abstraction layer)
```

Part 2 Questions (5pts)

What does switch 1 (S1) do?: _____

What does switch 2 (S2) do?: _____

What does switch 3 (S3) do?: _____

What variable would you change to make the first number shown on the 7-segment digits be '1234'?

_____

What pin on the chip is tied to the potentiometer?: _____

- In addition to the changing digits, there is a serial output terminal stream that is reporting status abut every second or so. Connect a terminal emulator to this stream (Windows: 'PuTTY', Mac: 'screen') and paste a line from its output.

  _____

- What does the potentiometer do in the program?

  _____