CSCE 336 Embedded Systems, Spring 2023
Lab 5 – Bonus Lab

Due: Friday, May 12th, 2023 (Close of Business)

Name: _____

1. Instructions

This Lab is an individual assignment, collaboration is not allowed. Show your work and describe your reasoning to get partial credit if your solution is incorrect. Complete all of the sections below and make sure to proof of functionality where required. You can upload proof of functionality to Canvas. Some checkoffs may require a picture, video, or other submission via Canvas. You should keep notes on what you complete since parts of future homework could build on this lab.

This assignment will be a completely online Canvas submission. **Turn in a copy of your code on Canvas!** Failing to electronically turn in your code will result in up to a 20 point penalty on this assignment. Points may also be deducted for coding errors, poor style, or poor commenting. You will build different functions for each portion of this lab and comment out the function call when not being used.

2. Analog to Digital Conversion

Connect your Analog Joystick to your Arduino using two of the analog inputs A0 through A5 for the VRx and VRy Analog Joystick pins and a digital input for the switch pushbutton.

Figure 1: Analog Joystick Module

Pin Configurations:
1. GND: ground
2. +5V: 5V DC
3. VRx: voltage proportional to x position
4. VRy: voltage proportional to y position
5. SW: switch pushbutton

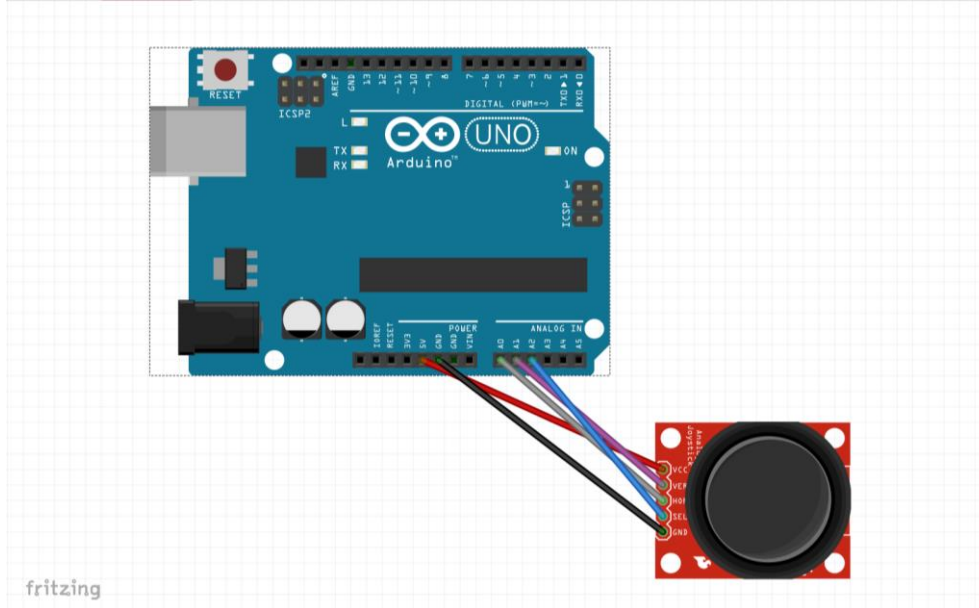Figure 2 below shows a sample configuration of connecting the Analog Joystick to the Arduino.

Figure 2: Connected Analog Joystick

Checkoff 1: *After you connect the Joystick to the Arduino take a picture. Upload a picture of the Joystick connected to your Arduino.*

Instructor sign off: _____

3.   "Analog Joystick Inputs without AnalogRead() Library"

Overview:

This portion of the lab is designed to assist you in learning the concepts associated with setting up the ADC registers to conduct an analog conversion on your Arduino by using the registers shown in class. The Analog Joystick in your parts kit will serve as an input to determine when the Joystick is pushed up, down, left, and right. The skills you will learn from this lab will come in handy in the future as you start to build the Etch-a-Sketch using a graphics display.

Checkoff 2: *Write code that reads two ADC inputs to read the proportional VRx and VRy values and the SW input on the Joystick and print values on the serial port that show when the Joystick is pushed Up, Down, Left, Right, and button press. Your joystick values need to have a threshold value so that if the joystick is released it should not show being pressed in any direction. Upload video proof. This can be combined with the Etch-a-Sketch functionality video.*

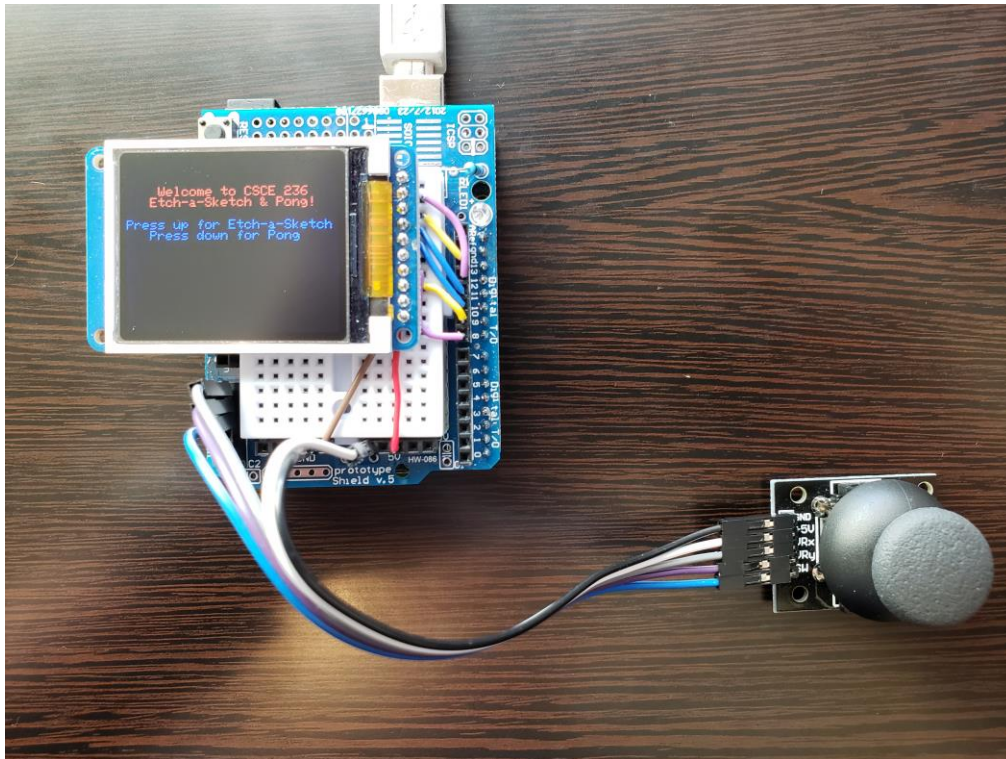Instructor sign off 2: _____

4. "Etch-a-Sketch"

Figure 3: Analog Joystick Module and 1.8" TFT Display connected to Arduino

Connect your Analog Joystick and 1.8 in TFT Graphics display to your Arduino using the pin configurations shown below and shown in Figure 3 above. For more information on the TFT display see https://learn.adafruit.com/1-8-tft-display/.

- **Lite** - this is the PWM input for the backlight control. Connect to 3-5VDC to turn on the backlight. Connect to ground to turn it off. Or, you can PWM at any frequency. **(I used 5V)**
- **MISO** - this is the SPI Microcontroller In Serial Out pin, its used for the SD card. It isn't used for the TFT display which is write-only **(Arduino Pin 12)**
- **SCLK** - this is the SPI clock input pin **(Arduino Pin 13)**
- **MOSI** - this is the SPI Microcontroller Out Serial In pin, it is used to send data from the microcontroller to the SD card and/or TFT **(Arduino Pin 11)**
- **TFT_CS** - this is the TFT SPI chip select pin **(I used Arduino Pin 10)**
- **Card CS** - this is the SD card chip select, used if you want to read from the SD card. (Not connected yet)
- **D/C** - this is the TFT SPI data or command selector pin **(I used Arduino Pin 8)**
- **RST** - this is the TFT reset pin. Connect to ground to reset the TFT! Its best to have this pin controlled by the library so the display is reset cleanly, but you can also connect it to the Arduino Reset pin, which works for most cases. **(I used Arduino Pin 9)**
- **Vcc** - this is the power pin, connect to 3-5VDC - it has reverse polarity protection but try to wire it right!
- **GND** - this is the power and signal ground pin

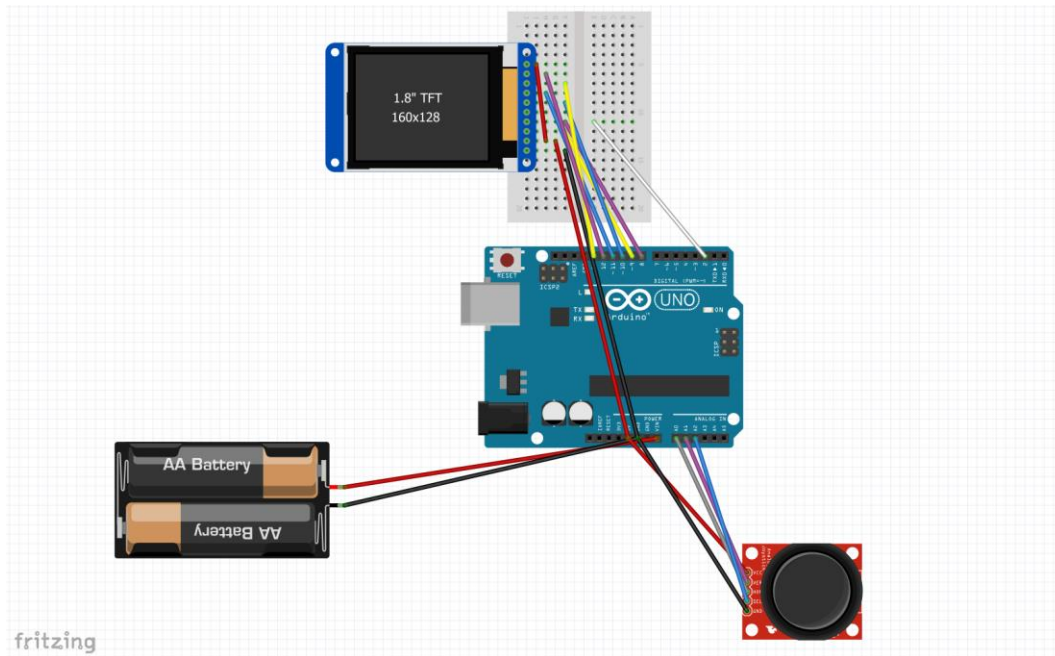Figure 4 below shows a sample configuration of connecting the Analog Joystick and TFT Display to the Arduino.

Figure 4: Analog Joystick Module and 1.8" TFT Display Sample Configuration

Checkoff 3: *After you connect the Joystick and TFT Display to the Arduino take a picture. Upload a picture of the Joystick and TFT display connected to your Arduino.*

Instructor sign off 3: _____

Overview:

Figure 5: Sample Etch-a-Sketch Screen

This portion of the lab is designed apply what you learned about ADC conversion using the Analog Joystick and to assist you in learning the concepts of serial communications with SPI to connect to a 1.8" TFT graphics display to build an Etch-a-Sketch.

Once you connect the 1.8" TFT Graphics display to the Arduino use the sample main code on the website to verify that the display will clear the screen and print a 4x4 pixel box in the center of the display using the following Adafruit function. The function is described in Figure 6 below.

```
tft.fillRect(TFT_width/2, TFT_height/2, box_width, box_height, color);
```

```
void Adafruit_SPITFT::fillRect(int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)

Draw a filled rectangle to the display. Self-contained and provides its own transaction as needed (see writeFillRect() or writeFillRectPreclipped()
for lower-level variants). Edge clipping and rejection is performed here.

Parameters:
 x  – Horizontal position of first corner.
 y  – Vertical position of first corner.
 w  – Rectangle width in pixels (positive = right of first corner, negative = left of first corner).
 h  – Rectangle height in pixels (positive = below first corner, negative = above first corner).
 color  – 16-bit fill color in '565' RGB format.
```

Figure 6: Adafruit fillRect Function

Checkoff 4: *Write code that creates small box (perhaps 4x4 pixels) on the screen and moves the box around on the display while leaving a trail of boxes using an Analog Joystick as input. The code will read the two proportional VRx and VRy values and the SW input on the Joystick to move the box Up, Down, Left, and Right. When the Joystick button is pressed it will change the color of the box being printed on the display. Upload video proof of the Etch-a-Sketch functionality.*

Instructor sign off 4: _____

5. "Pong"

Overview:

Figure 7: Sample Pong Screen

This portion of the lab is designed apply what you learned about ADC conversion using the Analog Joystick and the concepts of serial communications with SPI to connect to a 1.8" TFT graphics display to build a simple Pong game.

Required Functionality – Bouncing Ball

Using the given Pong header and implementation files given on the website to write the code to make the box (i.e. pong ball) to collide with and bounce off of all four sides of the display.

A-Functionality – Simple Pong Game

Once you get the ball the bounce then add in a simple paddle to move up and down on one side of the screen to collide with the ball keep it from hitting that side of the screen and ending the game. The paddle movement will be controlled with the Joystick.

Checkoff 5: *Write code that implements basic Pong functionality with a ball that bounces off of three sides of the screen and collides with a paddle on the fourth side otherwise ends the game. Upload video proof of the Pong functionality.*

Instructor sign off 5: _____