Robotics I – Extra Credit 15 points total Name:_________________________________

Write a single program (Arduino sketch based on diagram below) that will include these features (3 points each towards class grade, 15 points total). Note: LED = pin 5, Switch = pin 7, Servo = pin 11, Potentiometer= pin A0

You must demonstrate the lab instructor that this is working to get the proper credit for it. You can get partial credit for the parts that work.

1. Create a menu so that a user can select the following in the Serial Monitor:

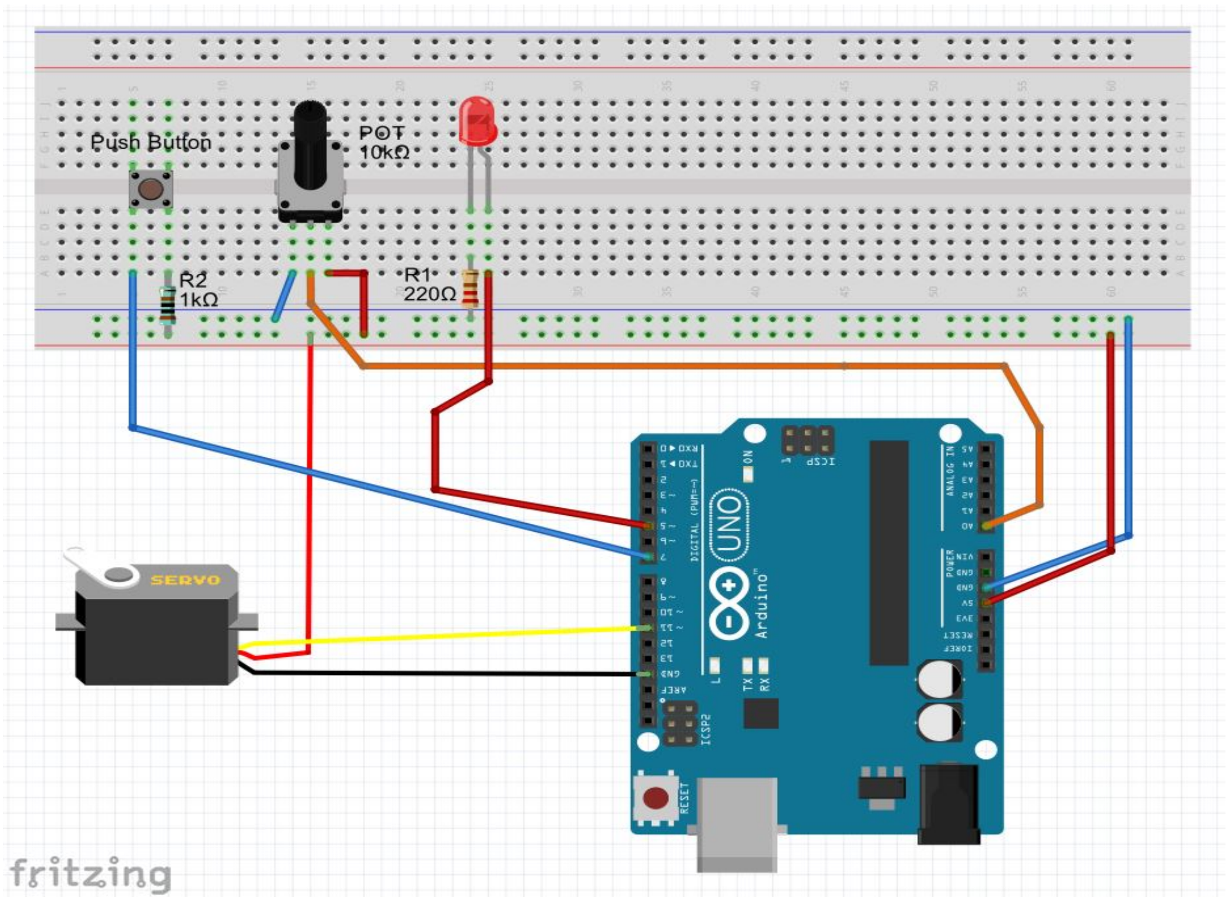
   Please choose one of following options:
   Press 1 to turn ON & OFF led
   Press 2 to adjust LED blink rate
   Press 3 to control LED brightness
   Press 4 to control Servo sweep

2. When you type 1 on the keyboard it will allow you to turn "LED on" when you press push button and "LED off" when you press button again.

3. When you type 2 on the keyboard it will allow you to turn "LED on" when you press a push button and "LED off" when you press the push button again. Also:
   a. LED will blink when is on.
   b. LED **blink rate** will be adjustable with connected variable resistor (pot) with adjustable blink time will be between 100µs to 1000µs

4. When you type 3 on the keyboard it will allow you to turn "LED on" when you press push button and "LED off" when you press button again.
   a. LED **brightness** will be adjustable with connected variable resistor

5. When you type 4 on the keyboard potentiometer (variable resistor) will control 180° sweep of servo motor arm.



Once you select, program will display your choice, for example after selecting option number 3, computer (serial monitor) will display a message “Running LED brightness control program.”

The lectures from http://www.roboticscity.com/learn-robotics.html will help you solve these problems.

Suggested commands:

1. `Serial.begin()`
2. `Serial.available()`
3. `Serial.read() or Serial.parseInt()`
4. `Serial.print() and Serial.println()`
5. `if()`
6. `switch case`
7. `map()`
8. `analogRead()`
9. `digitalRead()`

Example to check the state of the button:
/* State change detection (edge detection)
Often, you don't need to know the state of a digital input all the time, but you just need to know when the input changes from one state to another. For example, you want to know when a button goes from OFF to ON. This is called state change detection, or edge detection. This example shows how to detect when a button or button changes from off to on and on to off.
 The circuit:
 * pushbutton attached to pin 2 from +5V

```
  * 10K resistor attached to pin 2 from ground
  * LED attached from pin 13 to ground (or use the built-in LED on most Arduino boards)
http://arduino.cc/en/Tutorial/ButtonStateChange */

// this constant won't change:
const int  buttonPin = 2;    // the pin that the pushbutton is attached to

const int ledPin = 13;       // the pin that the LED is attached to

// Variables will change:

int buttonPushCounter = 0;   // counter for the number of button presses

int buttonState = 0;         // current state of the button

int lastButtonState = 0;     // previous state of the button

void setup() {

  // initialize the button pin as a input:

  pinMode(buttonPin, INPUT);

  // initialize the LED as an output:

  pinMode(ledPin, OUTPUT);

  // initialize serial communication:

  Serial.begin(9600);

}

void loop() {

  // read the pushbutton input pin:

  buttonState = digitalRead(buttonPin);

  // compare the buttonState to its previous state

  if (buttonState != lastButtonState) {

    // if the state has changed, increment the counter

    if (buttonState == HIGH) {

      // if the current state is HIGH then the button

      // wend from off to on:

      buttonPushCounter++;
      Serial.println("on");

      Serial.print("number of button pushes:  ");

      Serial.println(buttonPushCounter);

    }
    else {

      // if the current state is LOW then the button

      // wend from on to off:

      Serial.println("off");

    }

  }

  // save the current state as the last state,

  //for next time through the loop

  lastButtonState = buttonState;

  // turns on the LED every four button pushes by checking the modulo of the button push counter.

  // the modulo function gives you the remainder of the division of two numbers:

  if (buttonPushCounter % 4 == 0) {

    digitalWrite(ledPin, HIGH);

  }
  else {

    digitalWrite(ledPin, LOW);

  }

}
```

Reference:
https://opensourcehardwaregroup.com/tutorial-18-state-change-detection-and-the-modulo-operator-old-version/