TERRAN BLAKE // LAB: 7:30 WEDNESDAY // ECE 241 2016

```
enum ProcStates { Idle, Wait2, Low2};

#include<LiquidCrystal.h>                                    //These two lines setup the drivers for the
display
LiquidCrystal LcdDriver(11, 9, 5, 6, 7, 8 );

unsigned long currenttime = 0;
unsigned long timer = 0;
unsigned long timeOfLastButtonEvent = 0;

int interval = 5;
int ProcStates;
int buttonPin = 4;                                           //integer for button pin
int buttonPressCount = -1;
int tracker = 0;
int debounceInterval = 5;

boolean Input = LOW;                                         //Stores the value of the current state
boolean Wait = LOW;                                          //Stores the last state so that it loops properly
boolean Low = LOW;
                                        //Lets the button settle for 5ms                    //store
the last time the button state changed
void setup() {
  pinMode(buttonPin, INPUT);                                 //Sets pin 4 as an Input
  Serial.begin(9600);                                        //Begins Lcd on pin 16 and 2
  LcdDriver.begin(16, 2);
  LcdDriver.clear();
  LcdDriver.setCursor(0,0);
  ProcStates = Idle;
```

```
  pinMode(10,OUTPUT);
}

int NextState(int ProcStates) {

   switch(ProcStates) {
      case Idle:
         if (Input = LOW) {
            timer = millis();
            return Wait2;
         }
         break;
      case Wait2:
         if (Input = HIGH) {
            return Idle;
         }else if (timer - currenttime >= interval) {
            // Serial.println("indicator light is on");
            currenttime = timer;
            return Low2;

         }
         break;
      case Low2:
         if (Input = HIGH) {
            return Idle;
         }
    }
   }

void loop() {
```

```
 NextState(ProcStates);


 Input = digitalRead(buttonPin);
 unsigned long currentTime = millis();


 if (Input != Wait) {
    timeOfLastButtonEvent = currentTime;
 }


 if (currentTime - timeOfLastButtonEvent > debounceInterval){                    //Checks on the voltage
based on timer
    if (Input != Low) {                  //If the voltage has changed, switch states
       Low = Input;                                    //Updates the state


       //trigger an event
       if (Low == HIGH) {                                        //Does a serial print when button is pressed
          Serial.println("released");
          buttonPressCount++;
                        //Question 2, ads to counter when pressed and depressed
       } else {
          Serial.println("pressed");       //Serial print and adds to button press count
          //buttonPressCount++; This is for part 1
       }
    }
 }


 unsigned long oldMillis = 0;
 unsigned long currentMillis = millis();
 int interval = 500;
```

```
  if (buttonPressCount > tracker) {                    //Prints the amount of button presses based
on timer
     tracker++;
     LcdDriver.clear();
     LcdDriver.print(buttonPressCount);
  }

  Wait = Input;

}
```