

```

/*
*
-----
* Backyard Brains 2015
* Muscle SpikerShield Arduino UNO Code for Control of onboard relay
*
* Code monitors amplitude of EMG envelope, displays EMG strength on LED
bar and can control
* (ON/OFF) relay on the Muscle SpikerShield board.
*
* V1.0
* Written by Marcio Amorim
* Updated by Stanislav Mircic
*
* Tested with Muscle SpikerShield V2.31
*
-----
*/
#define NUMBER_OF_LEDS 6 //Number of LEDs in LED bar
#define OUTPUT_PIN 3 //Relay pin
#define OPERATION_MODE_BUTTON_PIN 7 //pin for button that
controls operation mode
#define SENSITIVITY_BUTTON_PIN 4 //pin for button that
controls sensitivity
#define INDICATOR_LED_PIN 13 //pin for LED that indicates
state of output

#define OPERATION_MODE_ON_OFF 1 //operation mode ON/OFF
#define OPERATION_MODE_FLICKER 2 //operation mode flickering

#define OUTPUT_STATE_ON 1 //output is in active (HIGH)
state
#define OUTPUT_STATE_OFF 0 //output is in inactive (LOW)
state

int emgMeasurement = 0; //current value of measured
EMG
int buttonState; //temporary variable for
reading of button state

int outputState = OUTPUT_STATE_OFF;
int currentOperationMode = 2;

```

```

//EMG saturation values (when EMG reaches this value the output will be
activated/deactivated)
int sensitivities[] = {200, 350, 520, 680, 840, 1000};
int lastSensitivitiesIndex = 2;           //set initial sensitivity
index
int sensitivity;                          //current sensitivity
threshold

byte ledbarHeight = 0;                   //temporary variable for led
bar height
byte ledPins[] = {8, 9, 10, 11, 12, 13}; //pins for LEDs in LED bar

//-----
//      Setup inputs and outputs
//
//-----
void setup()
{
    //set realy pin as output
    pinMode(OUTPUT_PIN, OUTPUT);

    //set button pins as inputs
    pinMode(OPERATION_MODE_BUTTON_PIN, INPUT);
    pinMode(SENSITIVITY_BUTTON_PIN, INPUT);

    //set all pins for LED bar to output
    for(int i = 0; i < NUMBER_OF_LEDS; i++)
    {
        pinMode(ledPins[i], OUTPUT);
    }

    //get the current sensitivity
    sensitivity = sensitivities[lastSensitivitiesIndex];
}

//-----
//      Main loop

```

```
//  
// - Checks state of sesitivity button  
// - Checks state of default-claw-state button  
// - Measure EMG  
// - Shows EMG strength on LED bar  
// - Sets state of the Relay ON/OFF  
//
```

```
-----  
void loop()  
{
```

```
    //----- Switch mode of operation  
-----
```

```
    buttonState = digitalRead(OPERATION_MODE_BUTTON_PIN);  
    if(buttonState == HIGH)  
    {  
        if(currentOperationMode == OPERATION_MODE_ON_OFF)  
        {  
            flickeringEffect();  
            currentOperationMode = OPERATION_MODE_FLICKER;  
        }  
        else  
        {  
            onOffEffect();  
            currentOperationMode = OPERATION_MODE_ON_OFF;  
        }  
    }  
}
```

```
    //----- Switch sensitivity  
-----
```

```
    //check if button is pressed (HIGH)  
    if (digitalRead(SENSITIVITY_BUTTON_PIN))  
    {  
        turnOffAllLeds();  
  
        //increment sensitivity index  
        lastSensitivitiesIndex++;  
        if(lastSensitivitiesIndex==NUMBER_OF_LEDS)  
        {  
            lastSensitivitiesIndex = 0;  
        }  
  
        //get current sensitivity value
```

```

sensitivity = sensitivities[lastSensitivitiesIndex];

//light up LED at lastSensitivitiesIndex position for visual
feedback
digitalWrite(ledPins[lastSensitivitiesIndex], HIGH);

//wait user to release button
while (digitalRead(SENSITIVITY_BUTTON_PIN))
{
    delay(10);
}
//wait a bit more so that LED light feedback is always visible
delay(100);
}

//----- Measure EMG signal
-----

emgMeasurement = analogRead(A0);

//----- Display EMG strength on LED bar
-----

turnOffAllLeds();
emgMeasurement= constrain(emgMeasurement, 0, sensitivity);
ledbarHeight = map(emgMeasurement, 0, sensitivity, 0, NUMBER_OF_LEDS);

for(int k = 0; k < ledbarHeight; k++)
{
    digitalWrite(ledPins[k], HIGH);
}

//----- Control output based on EMG strength
-----

//Control for ON/OFF mode of operation
if(currentOperationMode == OPERATION_MODE_ON_OFF)
{
    //If EMG reach threshold toggle current state
    //of output and wait for half the second
    //before next measurement of EMG

    if(emgMeasurement >= sensitivity)

```

```

    {
        if(outputState == OUTPUT_STATE_OFF)
        {
            digitalWrite(OUTPUT_PIN, HIGH);
            outputState = OUTPUT_STATE_ON;
        }
        else
        {
            digitalWrite(OUTPUT_PIN, LOW);
            outputState = OUTPUT_STATE_OFF;
        }
        delay(500);
    }
}

```

```

//Control for flickering mode of operation control
if(currentOperationMode == OPERATION_MODE_FLICKER)

```

```

{
    //If EMG reach threshold turn ON output
    //otherwise turn OFF output
    if(emgMeasurement >= sensitivity )
    {
        digitalWrite(OUTPUT_PIN, HIGH);
        outputState = OUTPUT_STATE_ON;
    }
    else
    {
        digitalWrite(OUTPUT_PIN, LOW);
        outputState = OUTPUT_STATE_OFF;
    }
    delay(20);
}

```

```

//----- Show indicator of output state
-----

```

```

if(outputState == OUTPUT_STATE_OFF)
{
    digitalWrite(INDICATOR_LED_PIN, LOW);
}
else
{
    digitalWrite(INDICATOR_LED_PIN, HIGH);
}
}

```

```
//-----  
-  
//  
//           Helper functions  
//  
//-----  
-  
  
//-----  
-  
// Turn OFF all LEDs in LED bar  
  
//-----  
-  
void turnOffAllLeds()  
{  
    for(int k = 0; k < NUMBER_OF_LEDS; k++)  
    {  
        digitalWrite(ledPins[k], LOW);  
    }  
}  
  
//-----  
-  
// Turn ON all LEDs in LED bar  
  
//-----  
-  
void turnOnAllLeds()  
{  
    for(int k = 0; k < NUMBER_OF_LEDS; k++)  
    {  
        digitalWrite(ledPins[k], HIGH);  
    }  
}
```

```
//-----  
-  
// Flicker all LEDs in LED bar just that user knows that we are in  
// flickering mode
```

```
//-----  
-  
void flickeringEffect()  
{  
    for(int i=0;i<5;i++)  
    {  
        turnOnAllLeds();  
        delay(100);  
        turnOffAllLeds();  
        delay(100);  
    }  
}
```

```
//-----  
-  
// Flicker ON all LEDs in LED bar wait for half the second and turn it  
off.  
// Just that user knows that we are in ON/OFF mode
```

```
//-----  
-  
void onOffEffect()  
{  
    turnOnAllLeds();  
    delay(500);  
    turnOffAllLeds();  
}
```