



Through the demonstration of counting the number of fist clenching, how to identify whether there is an EMG signal is illustrated. Identifying whether there is an EMG signal is to count muscle activity, which can be used for push-up counting, dumbbell counting, etc. It can also be used for human-computer interaction, such as playing the Flappy Bird game with the EMG signal generated by clenching a fist.

#### 1. Hardware configuration

- 1 x Arduino UNO board (or similar)
- 1 x EMG sensor signal processing board
- 1 x dry electrode connection wire
- 1 x 3P analog signal cable

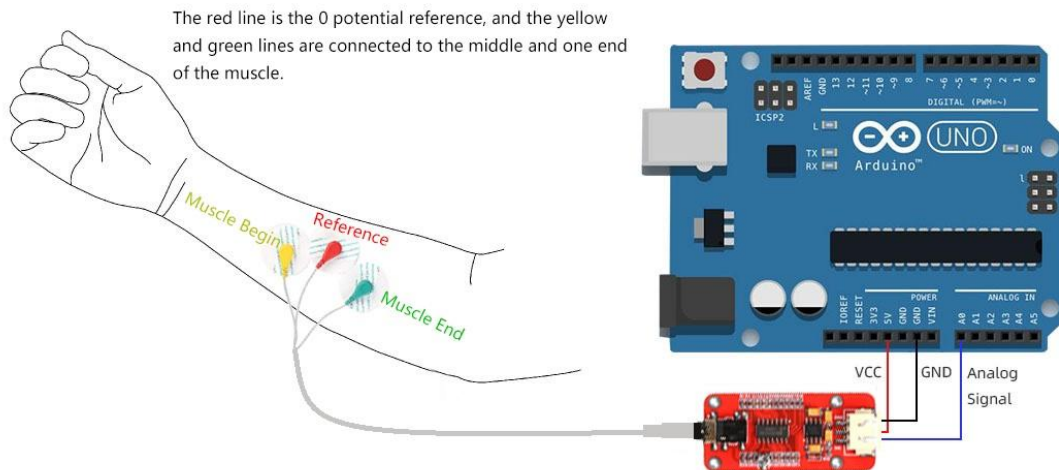
#### 2. Software configuration

Arduino IDE (1.8.2 and above recommended) (<https://www.arduino.cc/en/Main/Software>)

#### 3. Library files

EMG\_Filter-master.zip file inside the folder.

#### 4. Wiring diagram



#### 5. Demo code

Key point: When using it, you must disconnect the power supply of the laptop to get the correct waveform

```
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif
#include "EMGFilters.h"
#define SensorInputPin A0 //sensor input pin number
unsigned long threshold = 0; // threshold: Relaxed baseline values.(threshold=0:in the calibration process)
unsigned long EMG_num = 0; // EMG_num: The number of statistical signals
EMGFilters myFilter;
SAMPLE_FREQUENCY sampleRate = SAMPLE_FREQ_500HZ;
NOTCH_FREQUENCY humFreq = NOTCH_FREQ_50HZ;
void setup()
{
    myFilter.init(sampleRate, humFreq, true, true, true);
    Serial.begin(115200);
}
void loop()
{
    int data = analogRead(SensorInputPin);
    int dataAfterFilter = myFilter.update(data);
    int envelope = sq(dataAfterFilter);
    envelope = (envelope > threshold) ? envelope : 0;
    if (threshold > 0)
    {
        if (getEMGCount(envelope))
    }
```

```

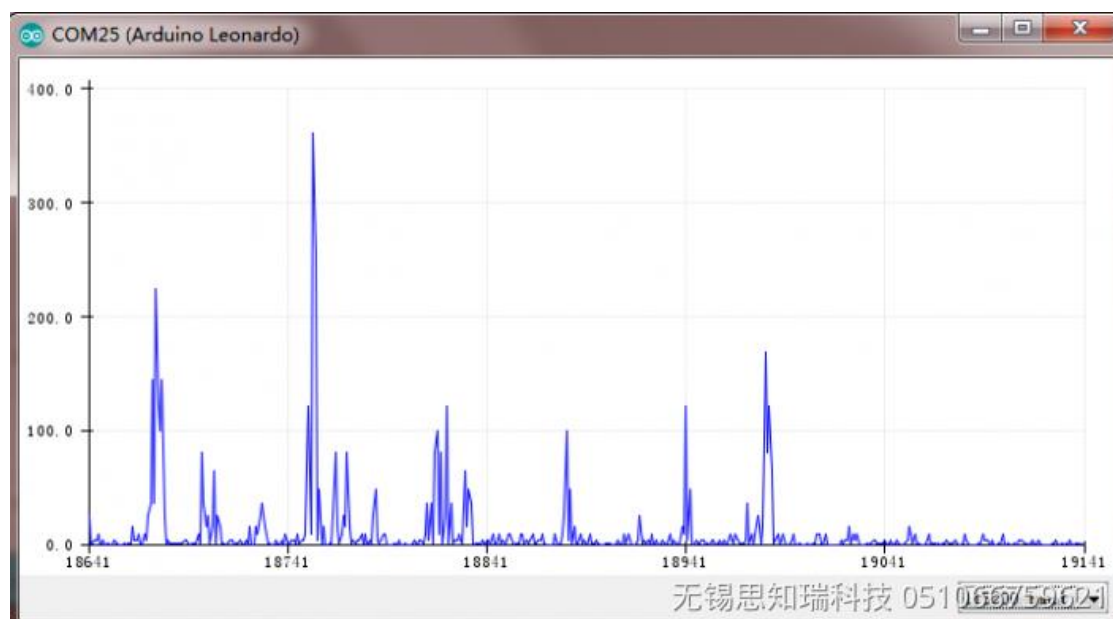
    { EMG_num++;
      Serial.print("EMG_num: ");
      Serial.println(EMG_num);
    }
  }
  else {
    Serial.println(envelope);
  }
  delayMicroseconds(500);
}

int getEMGCount(int gforce_envelope)
{
  static long integralData = 0;
  static long integralDataEve = 0;
  static bool remainFlag = false;
  static unsigned long timeMillis = 0;
  static unsigned long timeBeginzero = 0;
  static long fistNum = 0;
  static int TimeStandard = 200;
  integralDataEve = integralData;
  integralData += gforce_envelope;
  if ((integralDataEve == integralData) && (integralDataEve != 0))
  {
    timeMillis = millis();
    if (remainFlag)
    {
      timeBeginzero = timeMillis;
      remainFlag = false;
      return 0;
    }
    if ((timeMillis - timeBeginzero) > TimeStandard)
    {
      integralDataEve = integralData = 0;
      return 1;
    }
    return 0;
  }
  else {
    remainFlag = true;
    return 0;
  }
}

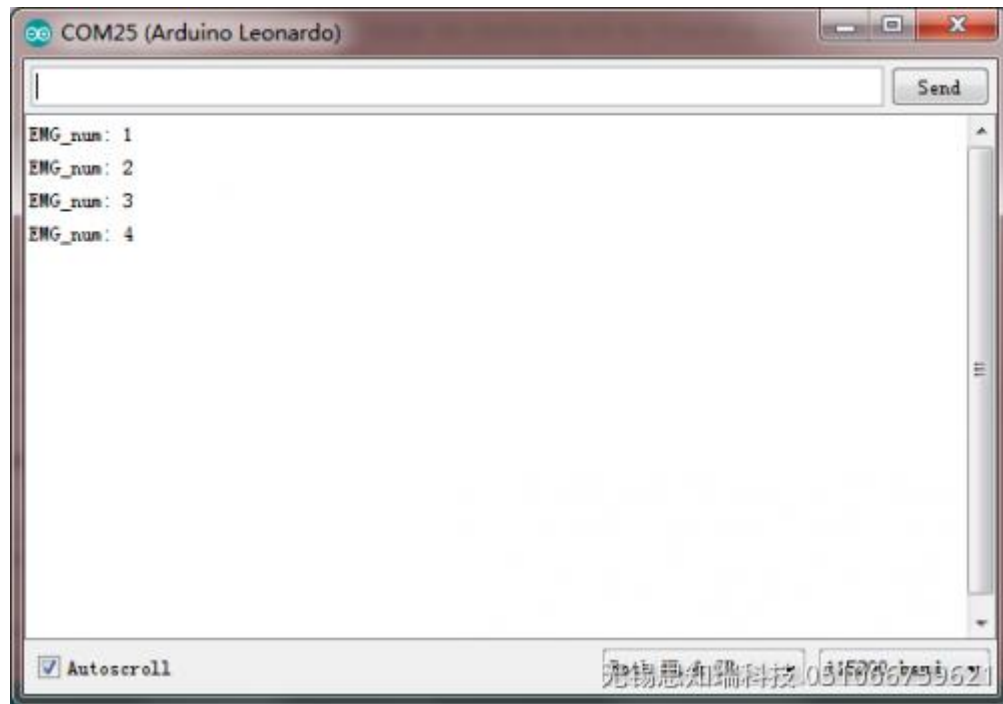
```

## 6. Code Description

1. In this code, the value of the threshold variable is 0 by default, indicating that it is in calibration mode. Change to a non-zero value to exit the calibration mode. Therefore, if you want to re-calibrate, just change the threshold variable in the code to 0, that is: unsigned long threshold = 0;
2. Upload the code to the arduino main control board, then open the Serial Plotter of the arduino IDE, and observe the maximum value of the y-axis.
3. Gently clenched the fist, then loosened the fist slightly, and so on. Observe the maximum value that appears on the y-axis in Serial Plotter and record it. As shown below, the maximum value is around 400.



4. Change the threshold variable in the code to the maximum value recorded just now. It is recommended to change it to: the maximum value +100 to improve the anti-interference ability. Re-upload the sample code to the Arduino main control board.
5. Open the Serial Monitor of the arduino IDE, make a fist and release it once, and you can see the printed count. A light clenched fist doesn't count. If you clench your fist tightly, the count will not count up. You must clench and release it to count. In this way, precise counting of the clenched fists is achieved. If it is found that the count is accumulated multiple times when the fist is released, it means that the threshold is set low, causing misjudgment. You can add 100 to the threshold variable value in the code, re-upload the sample code to the arduino UNO main control board, and check the effect again until the threshold variable is added to the satisfactory effect.



Place the sensor at the position where muscle activity needs to be detected, and then calibrate it to accurately count muscle activity, such as push-up counting, dumbbell counting

It can also be used in fitness occasions such as numbers, and can also be used for human-computer interaction.

My store <https://python.aliexpress.com>