Course Outline

- Chapter 1: Introduction
- Chapter 2: Basics
- Chapter 3: Digital
- Chapter 4: Analog
- Chapter 5: Communication
- Chapter 6: Control Structure
- **Chapter 7: Sensor**
- Chapter 8: Wi-Fi
- Chapter 9: Android
- Chapter 10: Xively
- Chapter 11: Bluetooth
- Chapter 12: Remote Car

Arduino Tutorial
Chapter 7: Sensor (part 4)
Playground of Sensors

National Chiao Tung University

2016/11

Content

1. Touch Sensor
2. Piezo Vibration Sensor
3. Moisture Sensor
4. Fire Alarm: Flame Sensor & Smoke Sensor
5. Galvanic Skin Response (GSR) Sensor
6. Ear-clip Heart Rate
7. Electromyography (EMG) Sensor

Sensors

Touch Sensor

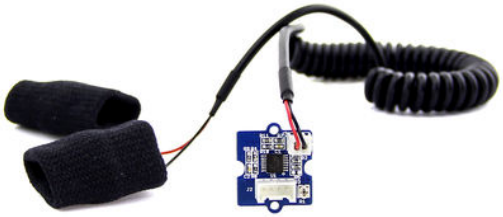
Vibration Sensor

Moisture Sensor

Flame Sensor

Smoke Sensor

Galvanic Skin Response

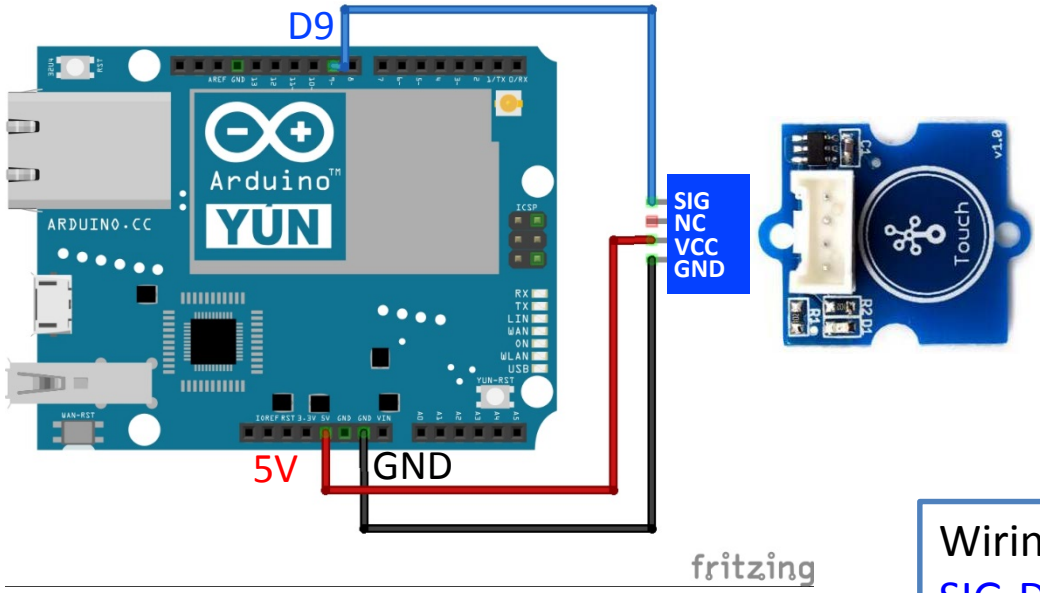


Ear-clip Heart Rate

EMG Sensor

Lab 1. Touch Sensor

- Goal: Use Grove touch sensor to turn on/off an LED .
- Hardware Required
  - Arduino Board
  - Grove touch sensor



Wiring:
SIG-D9
VCC-5V
GND-GND

Lab 1. Touch Sensor (Cont.)

- TTP223 is a touch pad detector IC which offers 1 touch key.
  - replacing traditional direct button key with diverse pad size.
  - Low power consumption and wide operating voltage are the contact key features for DC or AC application.
- After power-on have about 0.5sec stable-time, during the time do not touch the key pad, and the function is disabled

Lab 1. Touch Sensor (Cont.)

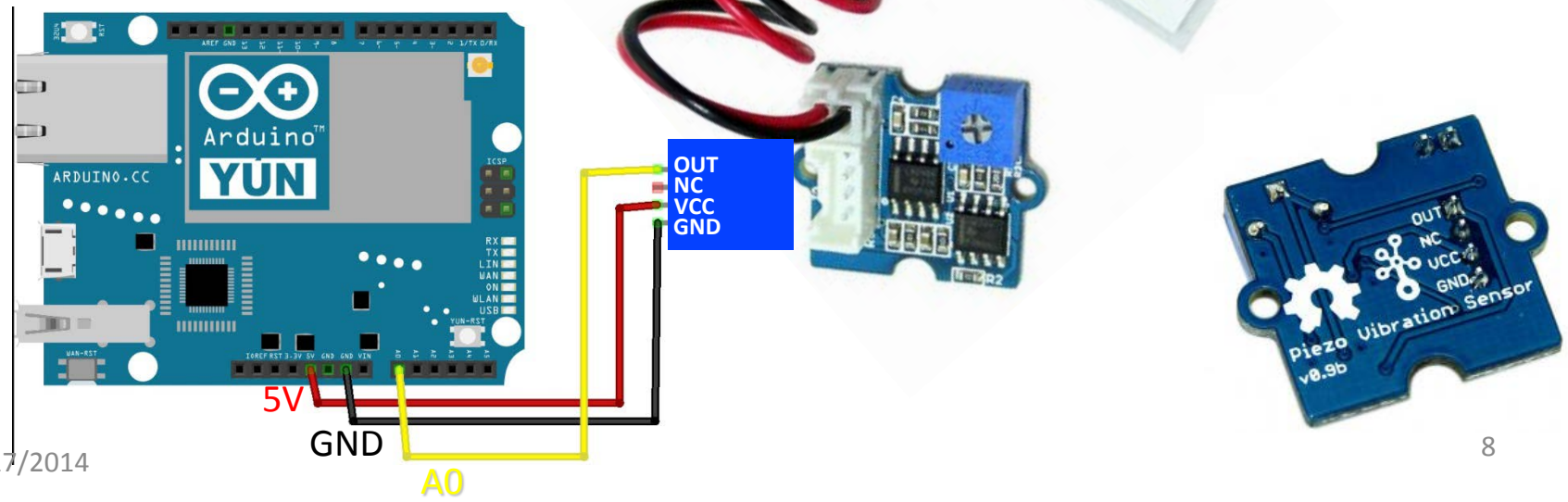
```
const int TouchPin=9; // touch sensor pin
const int ledPin=13;   // built-in LED
void setup() {
  pinMode(TouchPin, INPUT);
  pinMode(ledPin,OUTPUT);
}
```

```
void loop() {
  int sensorValue = digitalRead(TouchPin);
  if(sensorValue==1)
  {
    digitalWrite(ledPin,HIGH);
  }
  else
  {
    digitalWrite(ledPin,LOW);
  }
}
```

Lab 2. Piezo Vibration Sensor

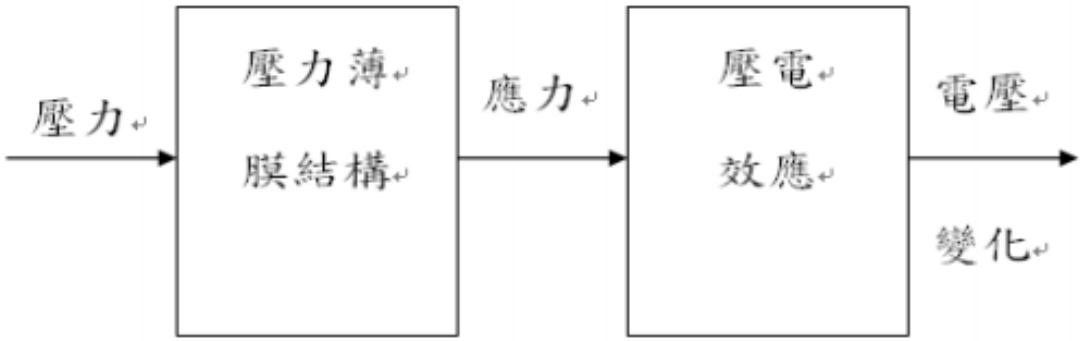
- Goal. Use Piezo Vibration Sensor to control LED.
  When the vibration is detected, this sensor outputs a logic high signal (the sensitivity can be changed by adjusting the potentiometer), and LED lights up.
- Hardware Required
  - Arduino Board
  - Grove piezo vibration sensor

Wiring
OUT-A0
VCC-5V
GND-GND



Lab 2. Piezo Vibration Sensor (cont.)

- The LDT0-028K is a flexible component comprising a 28 μm thick piezoelectric PVDF (聚氟化亞乙烯) polymer film with screen-printed Ag-ink electrodes, laminated to a 0.125 mm polyester substrate, and fitted with two crimped contacts.
  - Bending
  - Vibration



Lab 2. Piezo Vibration Sensor (cont.)

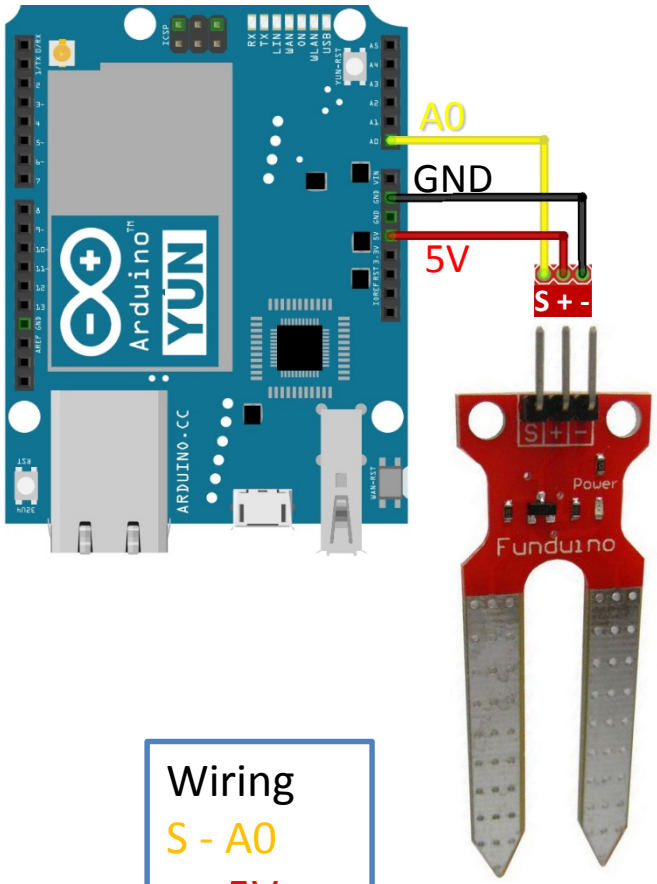
```
const int ledPin=13;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin,OUTPUT);
}
```

```
void loop() {
  int sensorValue = analogRead(A0);
 Serial.println(sensorValue);
 delay(1000);
 if(sensorValue>800)
 {
   digitalWrite(ledPin,HIGH);
 }
 else
 {
    digitalWrite(ledPin,LOW);
 }
}
```

Lab 3. Moisture Sensor

- Goal: Use Moisture Sensor to detect if any water exists.
- Hardware Required
  - Arduino Board
  - Moisture Sensor

水份感測器模組，可以用檢測土壤濕度、監控浴缸的水位、或是浴室是否有積水。感測器使用兩個探頭，通過讀取兩個電極之間的電流的變化，若有介質(乾土壤、水)，可讓電流通過，然後讀取該電阻得到的水分含量。水比較多的情況下，會更容易地進行電力（減少阻力），而乾燥的土壤導電性差。



Wiring
S - A0
+ - 5V
- - GND

Lab 3. Moisture Sensor (cont.)

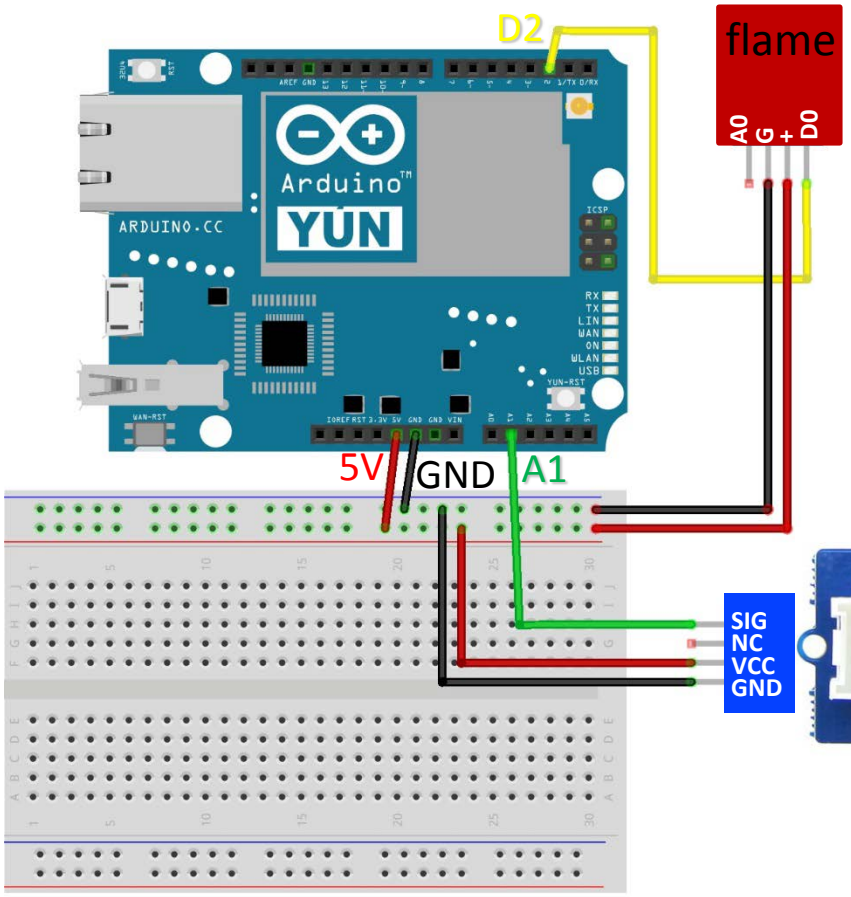
Lab 3. Moisture Sensor (cont.)

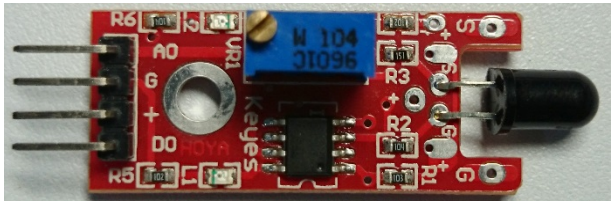
```
void setup(){
 Serial.begin(57600);
}

void loop(){
 Serial.print("Moisture Sensor Value:");
 Serial.println(analogRead(A0));
delay(100);
}
```

```
# the sensor value description
# 0 ~300 dry soil
# 300~700 humid soil
# 700~950 in water
```

Lab 4. Fire Alarm

- Goal: Use a flame sensor and a smoke sensor to detect if any fire or smoke exists, and then alarm





Flame sensor

Wiring
G - GND
+ - 5V
D0 - D2

- Hardware Required
  - Arduino Board
  - Flame Sensor
  - Smoke Sensor (MQ2)

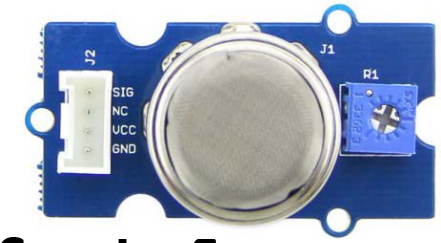
Smoke sensor

Wiring
SIG - A1
VCC - 5V
GND - GND

Lab 4. Fire Alarm (cont.)

Flame Sensor

- Used for short range fire detection and can be used to monitor projects or as a safety precaution to cut devices off / on.
- Very sensitive to IR wavelength at 760 nm ~ 1100 nm light, especially fire wavelength.



Smoke Sensor

- MQ-2:
  - Sensitive material SnO_2
  - Lower conductivity in clean air
  - When target combustible gas exists, the sensor's conductivity is more higher along with the gas concentration rising.
  - Convert change of conductivity to fit corresponding gas concentration

Lab 4. Fire Alarm (cont.)

```
const int ledpin=13;   //LED pin
const int flamepin=2; //flame sensor pin

void setup()
{
  pinMode(ledpin,OUTPUT);
  //LED pin as OUTPUT
  pinMode(flamepin,INPUT);
//FLAME SENSOR pin as INPUT

  Serial.begin(9600);
  Serial.println("Start detecting...");
  digitalWrite(ledpin,LOW);
}
```

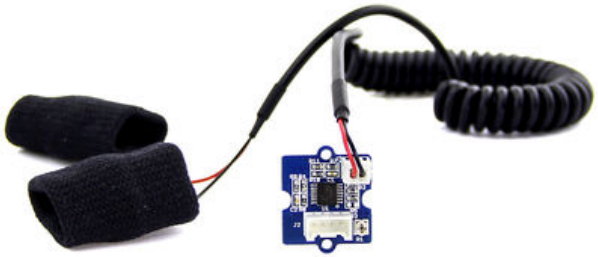
```
void loop()
{//variable declaration

   flameval=digitalRead(flamepin);
  //read flame sensor pin and assign to
   //flameval
   gasreading = analogRead(A1);

   Serial.print(flameval);
   Serial.print(",");
   Serial.println(gasreading);

  //check fire & smoke concentration
   if((flameval==HIGH) || (gasreading>200))
        digitalWrite(ledpin,HIGH);
   else
     digitalWrite(ledpin,LOW);
   delay(1500);
}
```

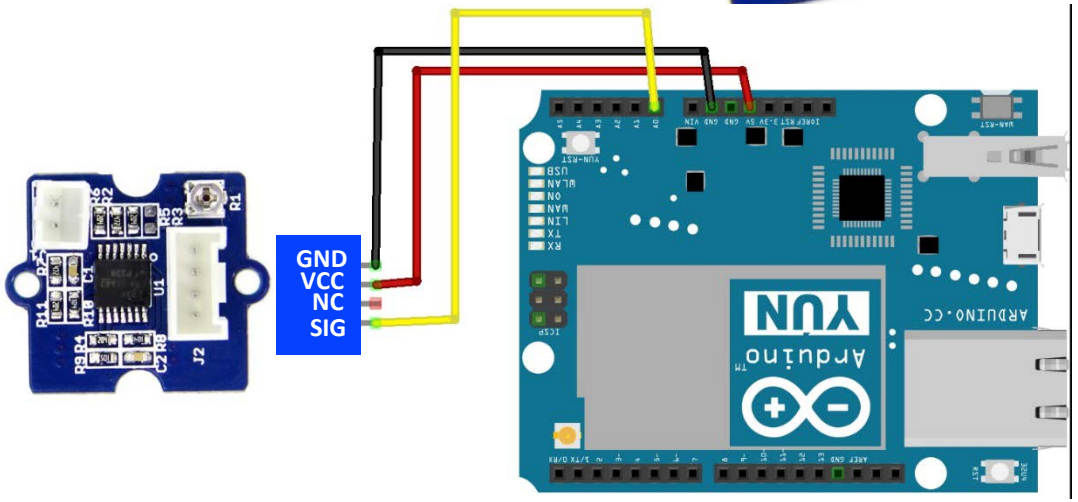
Lab 5. Galvanic Skin Response (GSR)

- Goal: Use GSR sensor to observe the analog output of some strong emotions by simple attaching two electrodes to two fingers on one hand, such as deep breath.

- Hardware Required
  - Arduino Board
  - Grove GSR Sensor

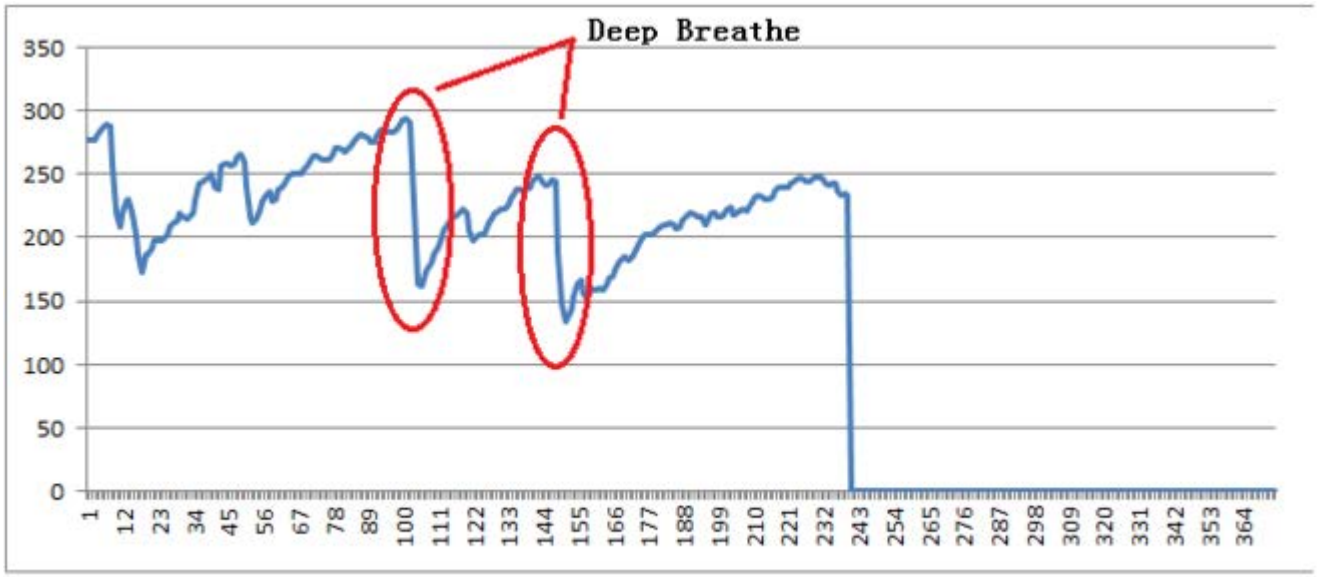
Wiring
GND - GND
VCC - 5V
SIG - A0



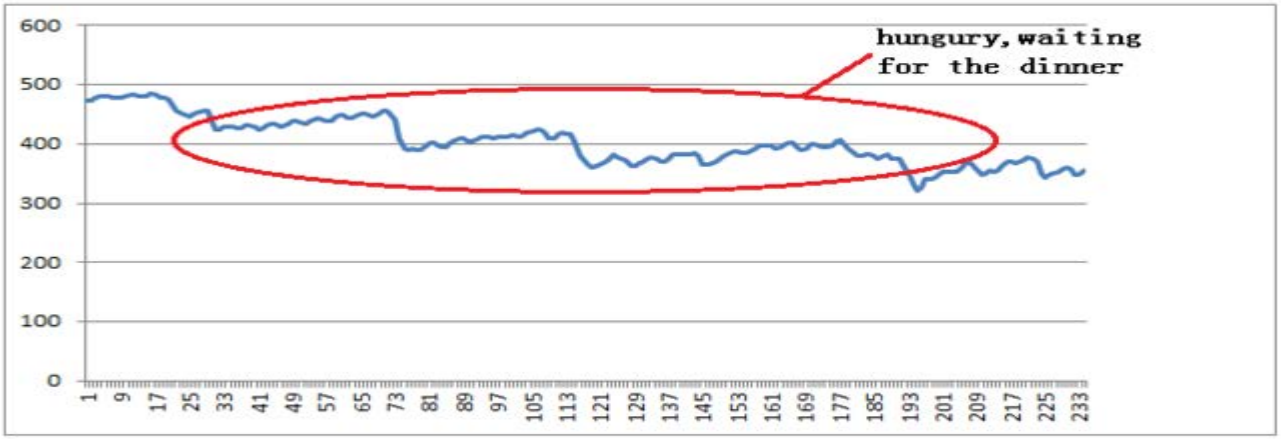


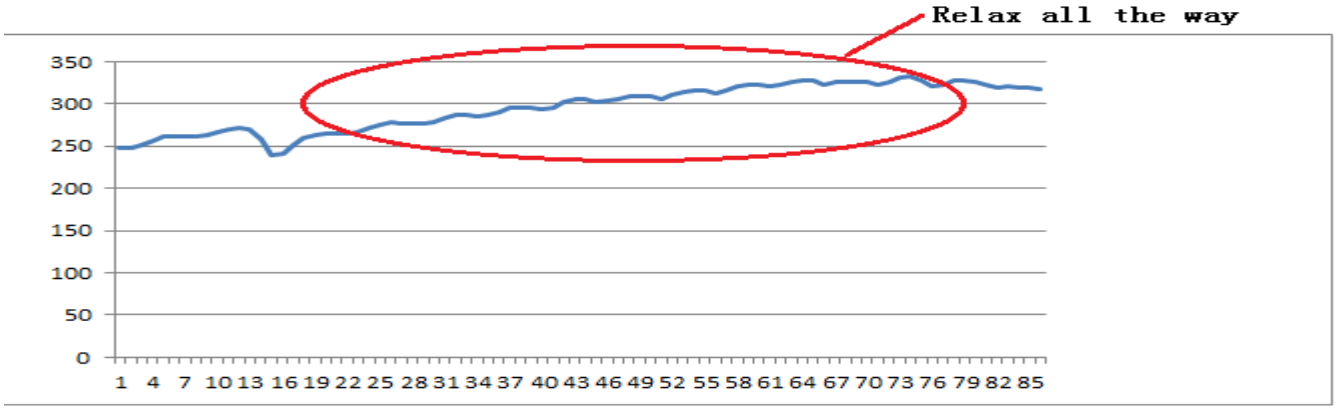
Lab 5. Galvanic Skin Response (GSR)

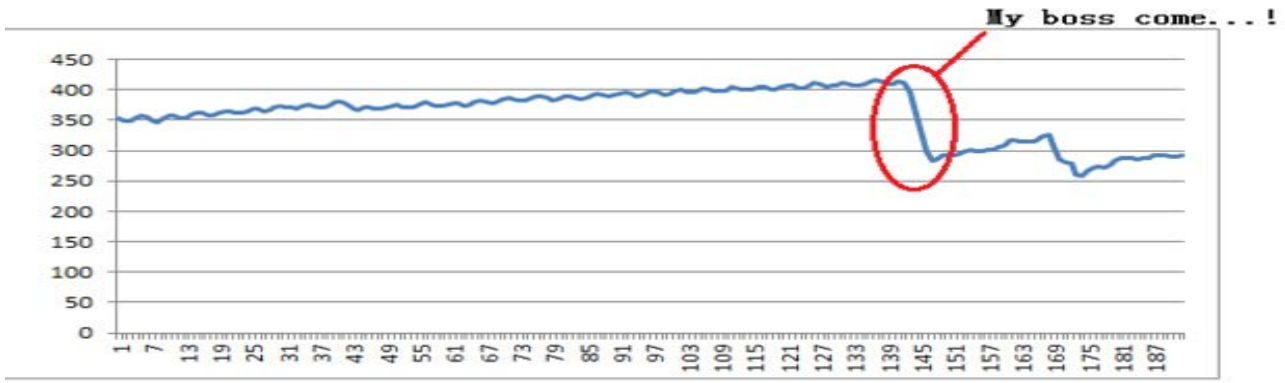
- GSR, standing for galvanic skin response, is a method of measuring the electrical conductance of the skin.
  - Strong emotion can cause stimulus to your sympathetic nervous system, resulting more sweat being secreted by the sweat glands. Grove – GSR allows you to spot such strong emotions by simple attaching two electrodes to two fingers on one hand, an interesting gear to create emotion related projects, like sleep quality monitor.



Lab 5. Galvanic Skin Response (GSR)







Lab 5. Galvanic Skin Response (GSR)

```
const int BUZZER=13;
const int GSR=A0;
int threshold=0;
int sensorValue;
```

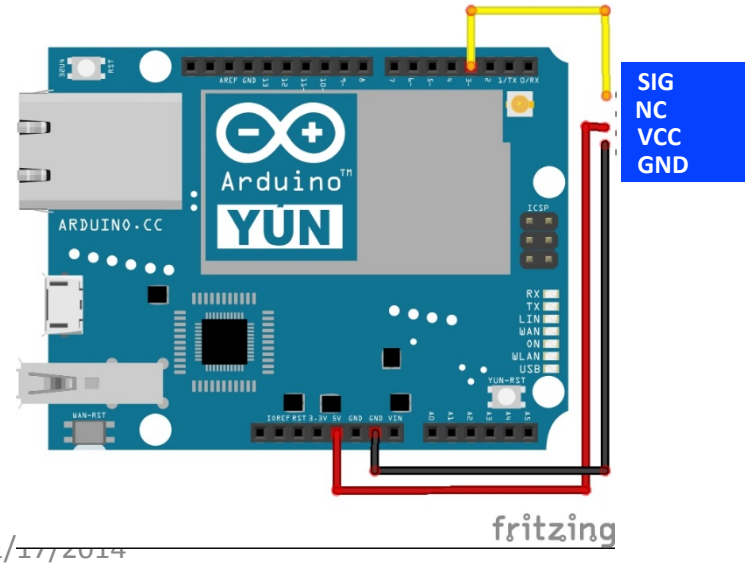
```
void setup(){
  long sum=0;
  Serial.begin(9600);
  pinMode(BUZZER,OUTPUT);
  digitalWrite(BUZZER,LOW);
  delay(1000);

  for(int i=0;i<500;i++)
  {
    sensorValue=analogRead(GSR);
    sum += sensorValue;
    delay(5);
  }
  threshold = sum/500;
   Serial.print("threshold =");
   Serial.println(threshold);
  }
```

Lab 5. Galvanic Skin Response (GSR)

```
void loop(){
  int temp;
  sensorValue=analogRead(GSR);
  Serial.print("sensorValue=");
  Serial.println(sensorValue);
  temp = threshold - sensorValue;
  if(abs(temp)>50)
  {
    sensorValue=analogRead(GSR);
    temp = threshold - sensorValue;
    if(abs(temp)>50){
    digitalWrite(BUZZER,HIGH);
    Serial.println("YES!");
    delay(3000);
    digitalWrite(BUZZER,LOW);
    delay(1000);}
  }
}
```

Lab 6. Ear-clip Heart Rate

- Goal: Use ear clip heart rate sensor to measure your heart rate
- Hardware Required
  - Arduino Board
  - Grove Ear-clip heart Sensor



Wiring
SIG - D3
VCC - 5V
GND - GND

Lab 6. Ear Clip Heart Rate (cont.)

- attachInterrupt()

attachInterrupt(interrupt, ISR, mode)

interrupt:	the number of the interrupt (int)
ISR:	the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine.
mode:	defines when the interrupt should be triggered. Four contstants are predefined as valid values: •LOW to trigger the interrupt whenever the pin is low, •CHANGE to trigger the interrupt whenever the pin changes value **•RISING to trigger when the pin goes from low to high,** •FALLING for when the pin goes from high to low.

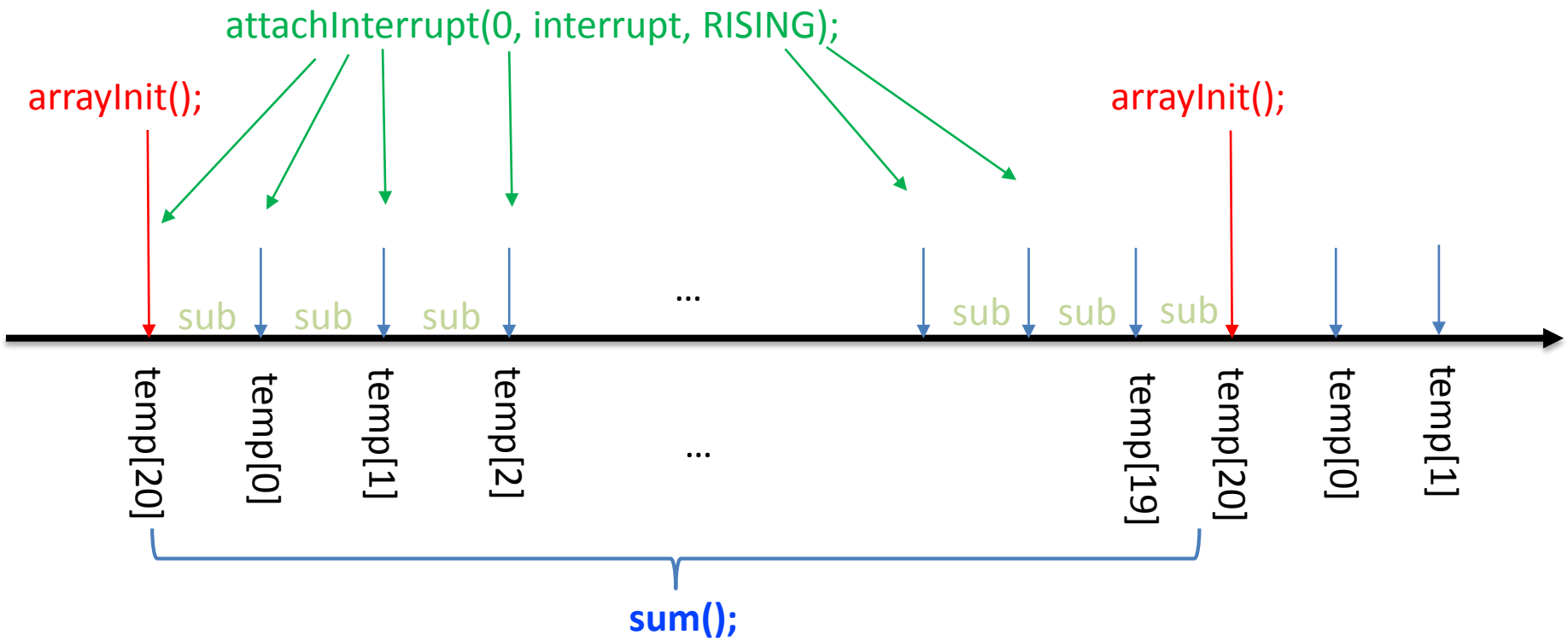
```
//interrupt example

void setup()
{
  attachInterrupt(0, interrupt, RISING);
  //set interrupt 0,digital pin 3 of yun
}

void loop()
{
   …
}

void interrupt()
{
  // do something
}
```

Lab 6. Ear Clip Heart Rate (cont.)

- External Interrupts of Yún:
  - 3 (interrupt 0)
  - 2 (interrupt 1)
  - 0 (interrupt 2), RX of Yún
  - 1 (interrupt 3), TX of Yún
  - 7 (interrupt 4), connected to the AR9331 processor and it may be used as handshake signal in future.

Lab 6. Ear Clip Heart Rate (cont.)

- How to calculate the heart rate?



sub < 2 seconds

Lab 6. Ear Clip Heart Rate (cont.)

```
#define LED 13
boolean led_state = LOW;
//state of LED, each time an external interrupt
//will change the state of LED
unsigned char counter;
unsigned long temp[21];
unsigned long sub;
bool data_effect=true;
unsigned int heart_rate;
//the measurement result of heart rate

const int max_heartpluse_duty = 2000;
//you can change it follow your system's
//request. 2000 meams 2 seconds. System
//return error if the duty overtrip 2 second.
```

```
void setup()
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  Serial.println("Please ready your ear clip.");
  delay(5000);
  arrayInit();
  Serial.println("Heart rate test begin.");
  attachInterrupt(0, interrupt, RISING);
  //set interrupt 0,digital pin 3 of yun
}
void loop()
{
  //Update the state of the indicator
  digitalWrite(LED, led_state);
}
```

```
for(unsigned char i=0;i < 20;i ++)
  {
                temp[i]=0;
  }
  temp[20]=millis();
```

Lab 6. Ear Clip Heart Rate (cont.)

```
/*Function: Interrupt service routine. Get the sigal from the
external interrupt*/
void interrupt()
{
   temp[counter]=millis();
   Serial.println(counter,DEC);
   Serial.println(temp[counter]);
   switch(counter)
   {
        case 0:
                sub=temp[counter]-temp[20];
                Serial.println(sub);
                break;
       default:
                sub=temp[counter]-temp[counter-1];
                Serial.println(sub);
                break;
    }
```

```
if(sub>max_heartpulse_duty)//set 2 seconds as
   //max heart pluse duty
   {
                data_effect=0;//sign bit
                counter=0;
                Serial.println("Measure  error, restart!");
                arrayInit();
   }

   if (counter==20&&data_effect)
   {
                counter=0;
                sum();
   }
   else if(counter!=20&&data_effect)
                counter++;
   else
   {
                counter=0;
                data_effect=1;
   }
} //end of interrupt()
```

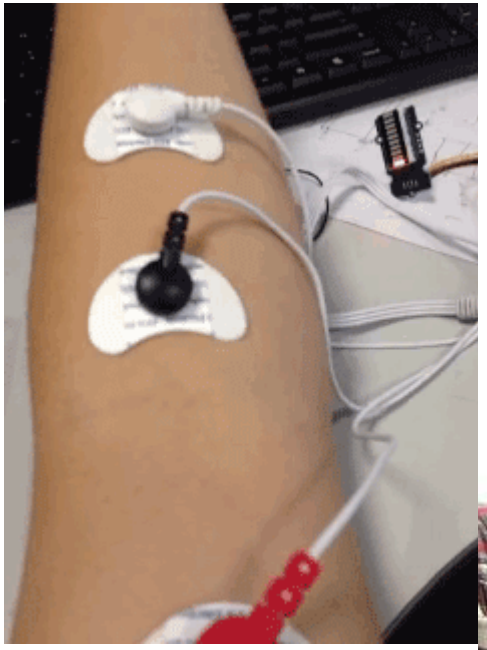
Lab 6. Ear Clip Heart Rate (cont.)

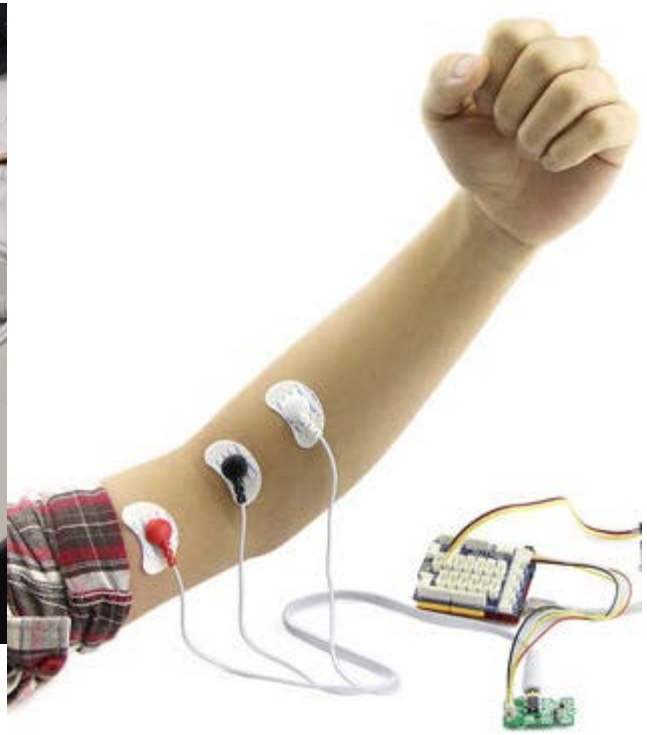
```
/*Function: calculate the heart rate*/
void sum()
{
 if(data_effect)
 {
    heart_rate=1200000/(temp[20]-temp[0]);
    Serial.print("Heart_rate_is:\t");
    Serial.println(heart_rate);
  }
   data_effect=1;//sign bit
}
```

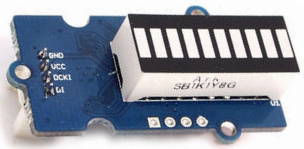
```
/*Function: Initialization for the array(temp)*/
void arrayInit()
{
  for(unsigned char i=0;i < 20;i ++)
  {
            temp[i]=0;
  }
  temp[20]=millis();
}
```

Lab 7. EMG Sensor

- Goal: Use EMG detector to observe your muscle activity, and show with an LED bar.
- Hardware Required
  - Arduino Board
  - Grove EMG Sensor
  - Grove LED bar



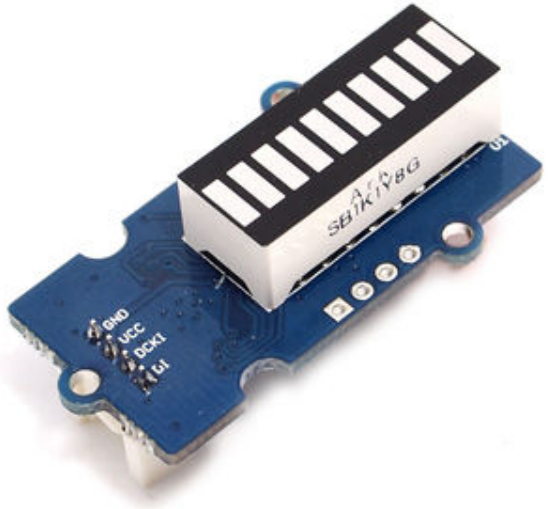


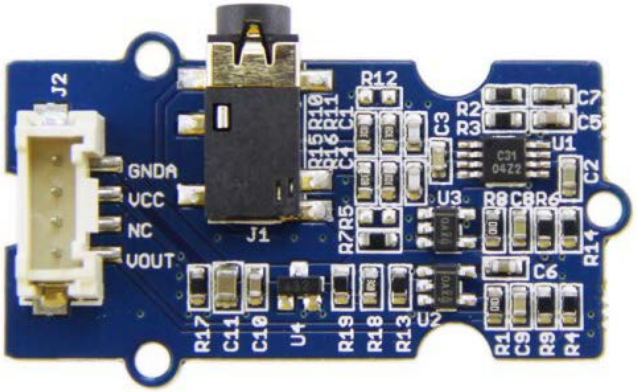


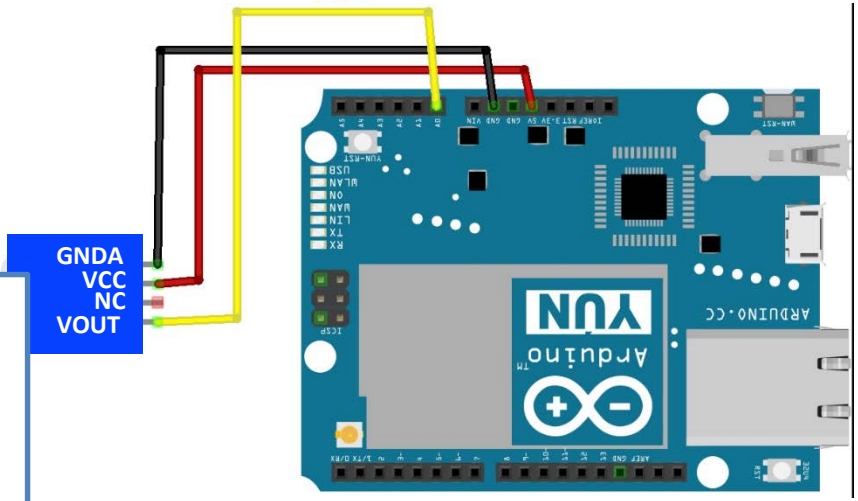
http://www.seeedstudio.com/wiki/images/7/7c/Grove_emg_demo_2.gif

Lab 7. EMG Sensor (cont.)

Wiring
GND - GND
VCC - VCC
DCKI – D9
GI – D8







Wiring
GNDA - GND
VCC - 5V
VOUT - A0

Lab 7. EMG Sensor (cont.)

- In standby mode, the output voltage is 1.5V. When detect muscle active, the output signal rise up, the maximum voltage is 3.3V. You can use this sensor in 3.3V or 5V system.

Lab 7. EMG Sensor (cont.)

```
#include <LED_Bar.h>
LED_Bar bar(9, 8);

int max_analog_dta   = 300;
int min_analog_dta    = 100
int static_analog_dta  = 0;

int getAnalog(int pin) // get analog value
{
   long sum = 0;
   for(int i=0; i<32; i++)
      sum += analogRead(pin);
   int dta = sum>>5;

   max_analog_dta = dta>max_analog_dta ? dta : max_analog_dta;      // if max data
   min_analog_dta = min_analog_dta>dta ? dta : min_analog_dta;      // if min data

   return sum>>5;
}
```

Lab 7. EMG Sensor (cont.)

```
void setup()
{
   Serial.begin(115200);
   long sum = 0;
   for(int i=0; i<=10; i++)
   {
      for(int j=0; j<100; j++)
      {
         sum += getAnalog(A0);
         delay(1);
      }
      bar.setLevel(10-i);
   }

   sum /= 1100;
   static_analog_dta = sum;
   Serial.print("static_analog_dta = ");
   Serial.println(static_analog_dta);
}
```

Lab 7. EMG Sensor (cont.)

```
int level      = 5;
int level_buf   = 5;

void loop()
{
// get Analog value
   int val = getAnalog(A0);
   Serial.println(val);

   int level2;

   if(val>static_analog_dta)
// larger than static_analog_dta
   {
      level2 = 5 + map(val, static_analog_dta, max_analog_dta, 0, 5);
   }
   else
   {
      level2 = 5 - map(val, min_analog_dta, static_analog_dta, 0, 5);
   }
// to smooth the change of led bar
   if(level2 > level)
   {
      level++;
   }
   else if(level2 < level)
   {
      level--;
   }

   if(level != level_buf)
   {
      level_buf = level;
      bar.setLevel(level);
   }
   delay(10);
}
```

Hands on

Arduino

- **Open--->**

  01.touch → touch.ino
  02.vibration → vibration.ino
  03.moisture → moisture.ino
  04.Flame → Flame.ino
  05.GSR → GSR.ino
  06.HeartRate → HeartRate.ino
  07.EMG → EMG.ino (Include LED_Bar library)

Target

- The quiz demo of previous Ch.7 Part 3
- 7 labs of today's course are for reference.
  - If you are interested in these, we can lend you the sensors to do the experiment and return them before you get out of class
- Grouping
  - 1 or 2 person for a group (Deadline: Nov. 29) https://goo.gl/forms/KpcbIPIt2q5Vx35k2
  - Creative and innovative applications
  - Proposal presentation on Dec. 6 (Main idea 1~2 slides)
  - Dec. 27: Final project preparation
  - Jan. 3: Demo & score