

# Instrucciones creación Miners Help

## Materiales

Pieza de agarre para sensor de humedad colocado desde la cabeza a la boca.



Pieza de agarre para Arduino a brazo (Anexo 1). Esta pieza puede ser hecha con material de impresión 3d o con cualquier material como metal o madera.



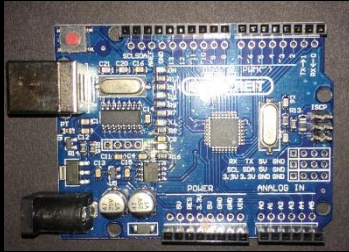
Cable.



Sensor de Humedad



Arduino.



Sensor Cardíaco AD8232



3 Electroodos



Cable de Conexión de Electroodos a Sensor AD8232

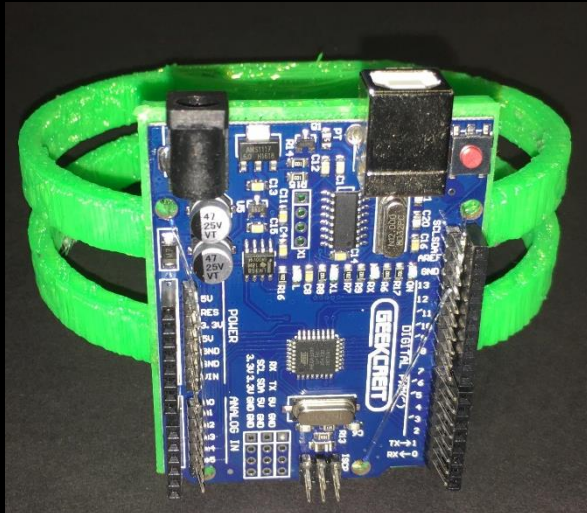


## Procedimiento

- 1- Colocar el sensor de Humedad en su base de cabeza a boca.

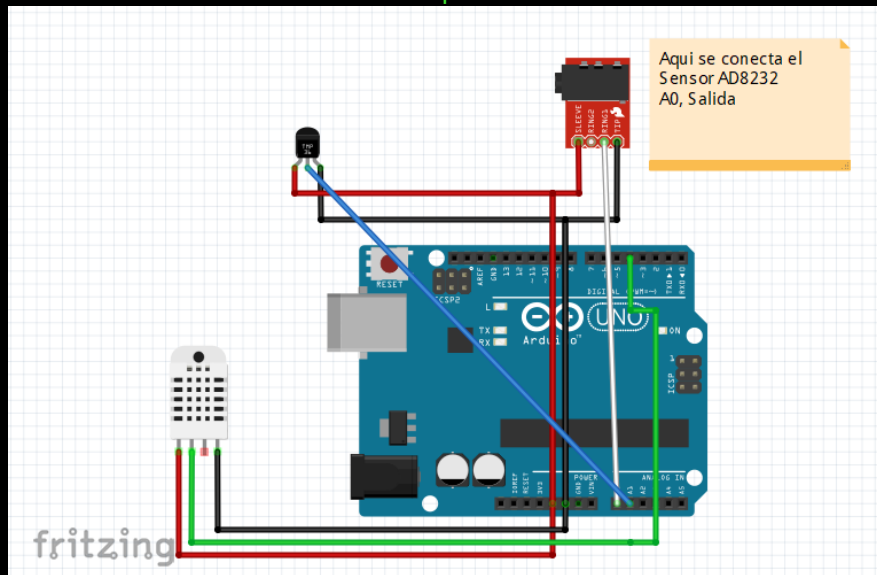


- 2- Colocar el Arduino en su respectiva pieza (Pieza 2).

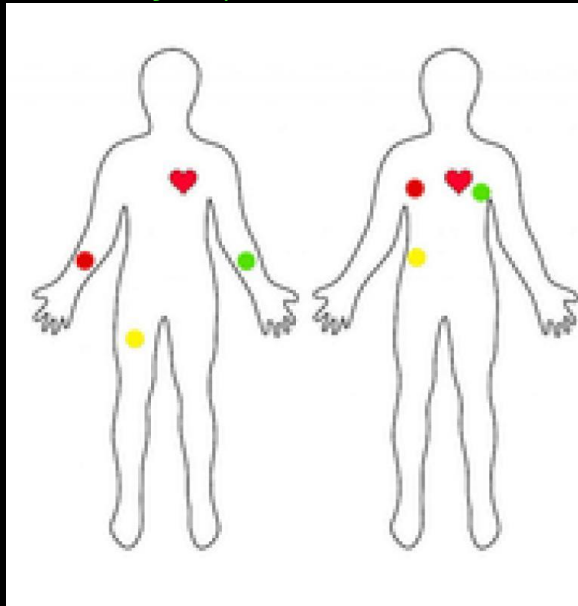


- 3- Sujetar las piezas antes armadas en la parte donde será su ubicación final, medir y cortar los trozos de cable para hacer la conexión. (esto recordando que el sensor de temperatura ira en la axila, la base con el sensor de humedad en la cabeza asegurándose de que este cerca de la boca, y el sensor cardiaco en la parte que se guste).

4- Realizar las conexiones correspondientes.



5- Colocar su Proyecto a la persona que lo necesite Sensor de temperatura en axila, sensor de humedad en la boca, y los electrodos en las partes indicadas y después al sensor AD8232.

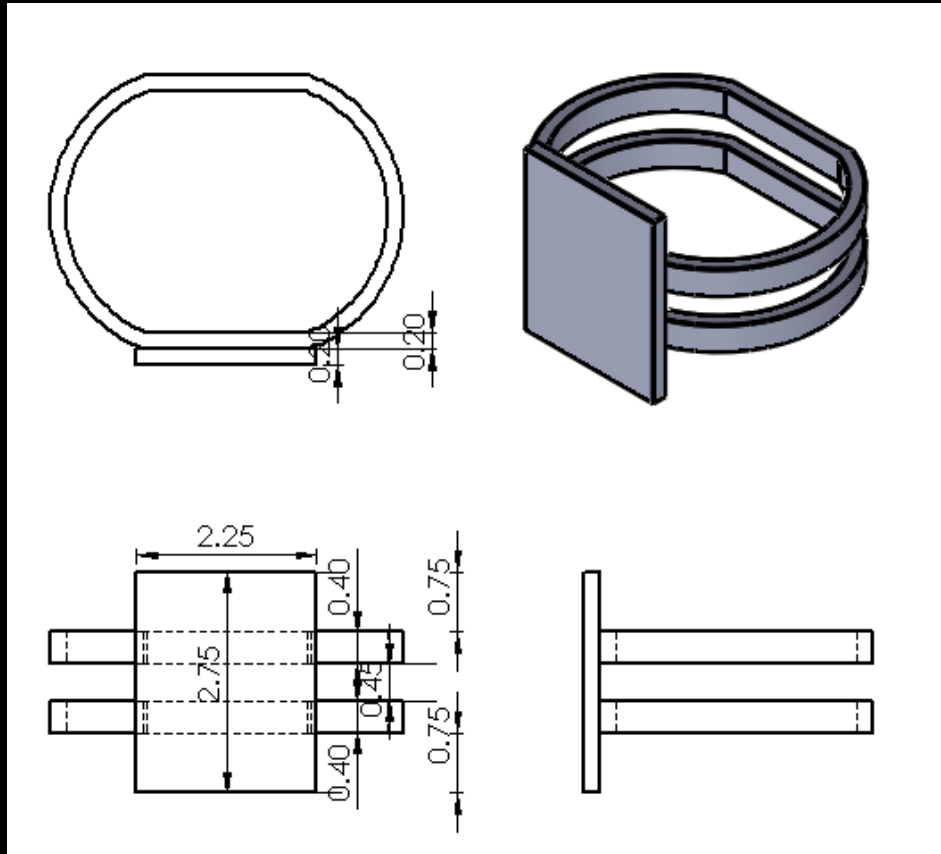


6- Ya esta listo para utilizar su nueva solución aplicando ciencia y tecnología por parte del equipo E-Hawks 6199, Cualquier duda y/o aclaración, estamos a sus ordenes

7- Recuerde que cada que se requiera calcular valores, es necesario cargar de nuevo el código.

## Anexos

### Anexo 1 (Pieza Arduino)



### Anexo 3

```
#include <DHT.h>
```

```
// Definimos el pin digital donde se conecta el sensor para respiracion
```

```
#define DHTPIN 4
```

```
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
//Variables para la lectura del electrocardiograma
```

```
float BPM;
```

```
float Ritmo;
```

```
int UpperThreshold = 518;
```

```
int LowerThreshold = 490;
```

```
int reading = 0;
```

```
int CR = 0;
```

```
bool IgnoreReading = false;
```

```
bool FirstPulseDetected = false;
```



```
unsigned long FirstPulseTime = 0;
unsigned long SecondPulseTime = 0;
unsigned long PulseInterval = 0;
const unsigned long delayTime = 10;
const unsigned long delayTime2 = 1000;
const unsigned long baudRate = 9600;
unsigned long previousMillis = 0;
unsigned long previousMillis2 = 0;
float PulsoFin;
```

```
//Variables para la lectura de temperatura
```

```
int Temp;
int TempFin;
int CT = 0;
```

```
//Variables la Respiracion
```

```
float Resp1;
int CS = 0;
int RespFin;
```

```
void setup(){
  //Serial
  Serial.begin(baudRate);
  Resp1 = Resp();
}
```

```
void loop(){
  //Codigo Pulso Cardiaco
  if(millis() < 60000){
    if(Pulso() < 30){ //Quitamos el error del sensor que lee una pulsacion de
      //menos de 30 que no puede tener un humano
      Ritmo = Ritmo+Pulso();
      CR++; //Contador del Ritmo (cuantas veces lo midio)
    }else{
      if(Pulso() > 140){
        Ritmo = Ritmo+140;
        CR++;
      }
    }
  }
  //Codigo Temperatura
  Temp = Temp+Temperatura();
  CT++;
  Serial.println(Ritmo);
  //Respiracion
```

```

if(millis()%5 == 0){
  if(Resp1+0.10>Resp() && Resp()>Resp1-0.10){
  }else{
    CS++;
    Resp1 = Resp();
  }
}
}else{
  //Pulso Promedio
  PulsoFin = Ritmo/CR;
  Serial.Println(PulsoFin);
  //Temperatura Promedio
  TempFin = Temp/CT;
  Serial.Println(TempFin);
  //Respiraciones Promedio
  RespFin = CS;
  Serial.Println(RespFin);
  delay(10000);
}
}
}

//Funciones para Ritmo Cardiac
float Pulso(){
  unsigned long currentMillis = millis();
  if(myTimer1(delayTime, currentMillis) == 1){
    reading = analogRead(0);
    if(reading > UpperThreshold && IgnoreReading == false){
      if(FirstPulseDetected == false){
        FirstPulseTime = millis();
        FirstPulseDetected = true;
      }
      else{
        SecondPulseTime = millis();
        PulseInterval = SecondPulseTime - FirstPulseTime;
        FirstPulseTime = SecondPulseTime;
      }
      IgnoreReading = true;
    }
    if(reading < LowerThreshold && IgnoreReading == true){
      IgnoreReading = false;
    }
    BPM = (1.0/PulseInterval) * 60.0 * 1000;
  }
  BPM = reading/3;
}

```

```

    if(myTimer2(delayTime2, currentMillis) == 1){
        Serial.print(reading);
        Serial.print("\t");
        Serial.print(PulseInterval);
        Serial.print("\t");
        Serial.print(BPM);
        Serial.println(" BPM");
        Serial.flush();
    }
    return BPM;
}

int myTimer1(long delayTime, long currentMillis){
    if(currentMillis - previousMillis >= delayTime){previousMillis =
currentMillis;return 1;}
    else{return 0;}
}

int myTimer2(long delayTime2, long currentMillis){
    if(currentMillis - previousMillis2 >= delayTime2){previousMillis2 =
currentMillis;return 1;}
    else{return 0;}
}

//Funcion para Tempetaruta
float Temperatura(){
    int dato;
    float c;
    dato=analogRead(1);
    c =(500.0 * dato)/1023;
    return (c);
}

//Funcion Respiraciones
float Resp() {
    // Esperamos 5 segundos entre medidas
    //delay(100); //Esto se quita del codigo original si no interferiria en los
demas calculos

    // Leemos la humedad relativa
    float h = dht.readHumidity();
    // Leemos la temperatura en grados centígrados (por defecto)
    float t = dht.readTemperature();
    // Leemos la temperatura en grados Fahrenheit
    float f = dht.readTemperature(true);

    // Comprobamos si ha habido algún error en la lectura
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Error obteniendo los datos del sensor DHT11");
        return;
    }
}

```



```
}

// Calcular el índice de calor en Fahrenheit
float hif = dht.computeHeatIndex(f, h);
// Calcular el índice de calor en grados centígrados
float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humedad: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperatura: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t");
Serial.print("Índice de calor: ");
Serial.print(hic);
Serial.print(" *C ");
Serial.print(hif);
Serial.println(" *F");
return hic;
}
```