Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Integración de un sistema multi-dimensional de medida de bioseñales para la identificación de la respuesta emocional ante estímulos musicales

Autor: Luis Rodríguez Cortés

Tutor: María del Mar Elena Pérez

Dpto. Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, Septiembre de 2020

Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Integración de un sistema multi-dimensional de medida de bioseñales para la identificación de la respuesta emocional ante estímulos musicales

Autor:
Luis Rodríguez Cortés

Tutor:
María del Mar Elena Pérez
Profesora Contratada Doctora

Dpto. Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, septiembre de 2020

Proyecto Fin de Carrera: Integración de un sistema multi-dimensional de medida de bioseñales para la identificación de la respuesta emocional ante estímulos musicales

Autor: Luis Rodríguez Cortés

Tutor: María del Mar Elena Pérez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mis amigos

A mi familia

A mis maestros

Agradecimientos

Me gustaría mencionar a todas las personas que han estado junto a mí estos años de grado, pues el camino no ha sido nada fácil, y sin ellos no habría sido posible continuar hasta el final. Se podría decir que han sido mi motivación para seguir avanzando, mostrando comprensión y apoyo en los momentos difíciles.

A Pablo y Fran, por esas quedadas esporádicas, en las que a pesar de llevar semanas o meses sin vernos, las conversaciones fluyen con la confianza típica de personas que se ven todos los días.

A Nacho, Fran y Jesús, por esos domingos, miércoles o cualquier día de la semana en los que el deporte, la cerveza y la buena compañía te liberan de la tensión del estudio continuo, formando cuerpo y mente. Especialmente a Jesús, por haber sido mi compañero de vida desde que nací, en los momentos buenos y en los no tan buenos. Nunca tuve hermanos, pero estando él, tampoco los eché de menos.

A mi familia, porque siempre me han incitado a seguir formándome, intentando abrirme el máximo número de puertas de cara al futuro. Especialmente a mis padres, que además de en lo académico, han sido un gran apoyo psicológico en los momentos en los que todo iba mal, ayudándome a no tirar la toalla (todos sabemos lo complicados que son los primeros años del Grado). A mi padre por inculcarme la cultura del esfuerzo, demostrándome que con trabajo podemos llegar a donde queramos en la vida. A mi madre, por enseñarme a no darle importancia a las cosas malas y centrarme en las buenas. Y una mención especial a mis abuelos, que han formado una gran familia con una unión envidiable entre los que pertenecemos a ella, superando todas las dificultades que se han puesto por delante.

A todas las personas maravillosas que he conocido en la carrera, profesores y alumnos. A Víctor, Juan, Pedro, Manu, César, Mati, Ale… por esos días inolvidables comiendo en la Fcom, en la sala de rol, en el parque, incluso en las salas de la biblioteca de Fcom... sin duda sois el mejor recuerdo que me llevo de estos años. Aún así, no puedo olvidarme del resto de personas maravillosas que me han acompañado estos años, compañeros de clase, el maravilloso grupo de Electrónica, los telemáticos que en muchas ocasiones comieron con nosotros en Fcom… y Paquito. Por último, a mi profesora del TFG, Mar, que a pesar de realizar el trabajo en un tiempo limitado que incluía las vacaciones, siempre ha estado muy atenta para facilitarme tanto ayuda como los materiales necesarios en cada momento.

Muchas gracias a todos, este Grado es tanto mío como vuestro.

Luis Rodríguez Cortés

Estudiante de Grado en Ingeniería de las Tecnologías de Telecomunicación

Sevilla, 2020

Resumen

Hoy en día, la tecnología se ha convertido en una parte fundamental de nuestro día a día, tanto para fines laborales como para ocio o investigación. En este sentido, los sensores han tomado una importancia vital en la medida de señales de nuestro entorno.

En este proyecto nos centramos en la medida de señales fisiológicas, ya que varios estudios han comprobado que hay relación entre ellas y las emociones que sienten las personas. Partiendo de sensores de pulso cardíaco, conductancia de la piel, tensión muscular y movimiento ocular, se recogerán los datos de forma sincronizada en un ordenador. Una vez hecho esto, se procesarán los datos de forma conjunta, consiguiendo así una lectura útil de dichas señales fisiológicas.

En el futuro, este proyecto podría ayudar a desarrollar estudios acerca de otras correlaciones entre la música y la emoción, además de tener aplicaciones en otros campos como la medicina.

Abstract

Nowadays, technology has become a fundamental part of our daily lives, wether it be for work, leisure or research. In this scope, sensors have been given a crucial importance at measuring our surrounding signals.

In this project, we are focusing on physiological signals measuring, as several studies have shown there is a close relationship between them and people's emotion. Starting from heart rate, skin conductance, muscular tension and eye movement sensors, data will be gathered synchronously, and saved to a computer. Once this is done, data will be processed altogether, achieving a useful reading of the before-mentioned physiological signals.

In the future, this project could help develop studies about some other correlations between music and emotion, in addition of having applications in other fields such as Medicine.

Índice

Agradecimientos **ix**

Resumen **xi**

Abstract **xiii**

Índice **xiv**

Índice de Tablas **xvi**

Índice de Figuras **xviii**

1 Introducción **1**
1.1 Estado del Arte *1*
1.2 Alcance *3*
1.3 Requisitos Técnicos *4*

2 Objetivos **5**

3 Planificación **7**

4 Sistema de medidas **9**
4.1 Introducción *9*
4.2 Componentes *10*
4.2.1 Sensores conectados a mirocontrolador 10
4.2.2 Sensor Electrooculograma 18
4.2.3 Cámara Panasonic Lumix DMC-LX7 20
4.3 Sincronización *21*
4.3.1 PLX-DAQ 22
4.3.2 Imágenes guardadas desde Matlab con la hora actual 23
4.3.3 Grabación con la Cámara de pantalla y sujeto experimental 24

5 Procesamiento de los Datos **27**
5.1 Procesamiento en Matlab *27*
5.2 Validación del Sistema *29*

6 Conclusiones **34**

7 Anexos **36**
7.1 Matriz de Verificación de requisitos técnicos *36*
7.2 Presupuesto *37*
7.3 Hojas Características de los sensores *38*
7.3.1 Pulse Sensor 38
7.3.2 GSR Sensor 41
7.3.3 MyoWare Muscle Sensor 45
7.4 Códigos *54*
7.4.1 Arduino Uno 54
7.4.2 Arduino Pro Mini 59
7.4.3 Matlab Capturas Electrooculograma 60
7.4.4 Matlab Señales Fisiológicas 63

Referencias **65**

Glosario **69**

ÍNDICE DE TABLAS

Tabla 3-1. Planificación temporal del Proyecto. 7
Tabla 7-1. Matriz de Pruebas. 36
Tabla 7-2. Presupuesto del proyecto. 37

ÍNDICE DE FIGURAS

Figura 1-1. Elección de humor de Moody [5]. 1
Figura 1-2. Coincidencia de Piloerección con bajada de GSR [6]. 2
Figura 1-3. Ejemplo de dispositivos de captura de señales fisiológicas [7]. 3
Figura 4-1. Esquema del Sistema Completo. 9
Figura 4-2. Placa de Arduino Uno. 10
Figura 4-3. Llamada a las funciones de cada sensor. 10
Figura 4-4. Configuración de la función del Pulse Sensor. 11
Figura 4-5. Inicialización del Pulse Sensor. 11
Figura 4-6. Conexionado del Pulse Sensor (Pin "A0"). 12
Figura 4-7. Pulse Sensor [10]. 12
Figura 4-8. Colocación del sensor GSR en los dedos índice y corazón. 13
Figura 4-9. Circuito equivalente del GSR Sensor [11]. 14
Figura 4-10. Obtención de datos del GSR Sensor por promedio. 14
Figura 4-11. Conexionado del GSR Sensor (Pin "A2"). 15
Figura 4-12. GSR Sensor [13]. 15
Figura 4-13. Colocación del sensor EMG en el músculo masetero de la mandíbula. 16
Figura 4-14. Obtención de datos del sensor EMG por promedio. 16
Figura 4-15. Conexionado del EMG Sensor (Pin "A4"). 17
Figura 4-16. EMG Sensor [16]. 17
Figura 4-17. Webcam sin filtro de infrarrojos rodeada de cuatro leds infrarrojos. 18
Figura 4-18. Sensor Electrooculograma y Arduino Pro Mini con conexiones. 18
Figura 4-19. Código Arduino para el sensor Electrooculograma. 19
Figura 4-20. Conexionado Arduino Pro Mini. 19
Figura 4-21. Cámara Panasonic Lumix DMC-LX7. 20
Figura 4-22. Conexionado del USB Hub de Black UFO [20]. 21
Figura 4-23. Sincronización de los sensores de Arduino Uno con PLX-DAQ [19]. 22
Figura 4-24. Interfaz del PLX-DAQ. 22
Figura 4-25. Capturas guardadas con la hora en que se toman. 23
Figura 4-26. Código para guardar las capturas con la hora en el nombre. 23
Figura 4-27. Widget de reloj digital para el escritorio. 24
Figura 4-28. Sujeto experimental con los cuatro sensores midiendo. 25
Figura 5-1. Importación de los datos a Matlab (I). 27
Figura 5-2. Importación de los datos a Matlab (II). 27
Figura 5-3. GSR, EMG, BPM y Hora en función del tiempo transcurrido. 28

Figura 5-4. Cámara grabando sujeto experimental y hora. 29
Figura 5-5. Señales Fisiológicas medidas en función del tiempo transcurrido. 30
Figura 5-6. Electrooculograma en los primeros compases del experimento. 31
Figura 5-7. Electrooculograma en los últimos compases del experimento. 31

1 INTRODUCCIÓN

1.1 Estado del Arte

En los últimos años se está investigando en el campo de la Recuperación de Información Musical (MIR). Esta técnica está siendo utilizada para categorizar, manipular e incluso crear música. Algunos ejemplos de aplicaciones concretas serían Sistemas de Recomendación Musical, separación de pistas, reconocimiento de instrumentos o transcripción automática de música [1].

En 2007, Pedro J. Ponce de León y José M. Iñesta crean un sistema de identificación de estilos musicales mediante reconocimiento de patrones, analizando características intrínsecas de géneros como el jazz o música clásica. Llegan a conseguir un acierto del 90% en la identificación del género [2].

Por otra parte, en 2013 un estudio muestra un experimento para determinar la relación entre música y emoción. En dicho estudio, se realiza un experimento en el cuál se reproducen fragmentos musicales con diferentes tempos, además de reproducirlos marcha atrás. Los usuarios escuchan los fragmentos en un ambiente tranquilo y valoran si lo que oyen les parece agradable o desagradable. El tempo se asocia con la excitación, causando mayor excitación los tempos rápidos respecto a los lentos. En cuanto a la valencia, se comprueba que los usuarios consideran más agradables los fragmentos reproducidos con normalidad respecto a aquellos reproducidos marcha atrás [3].

En septiembre de 2018, se realiza un estudio midiendo algunas señales fisiológicas humanas mientras los sujetos de estudio escuchaban música. Mediante un sistema llamado Moody [4], los usuarios eligen diferentes pistas musicales que escuchan, y después informan al sistema acerca de su humor y excitación. Una vez recogidos los datos de los usuarios y las señales fisiológicas, se concluye que la conductancia de la piel y el pulso cardíaco son las señales que mejor representan la emoción del usuario al escuchar música.

Figura 1-1. Elección de humor de Moody [5].

En 2020, María Castrillo hace un estudio de las respuestas fisiológicas de personas mientras escuchaban música, en algunos casos sin estímulo visual y en otros con vídeo. En concreto se midieron el pulso cardíaco y la conductancia de la piel, además de grabar a los usuarios con una cámara para buscar fenómenos como la piloerección. Los resultados son muy interesantes, ya que se encuentra una correlación entre la piloerección y las señales fisiológicas medidas. En cuanto al parámetro relacionado con la conductancia (GSR), en caso de piloerección se nota una leve subida y después una bajada pronunciada, como observamos en la Figura 1-2 [6].

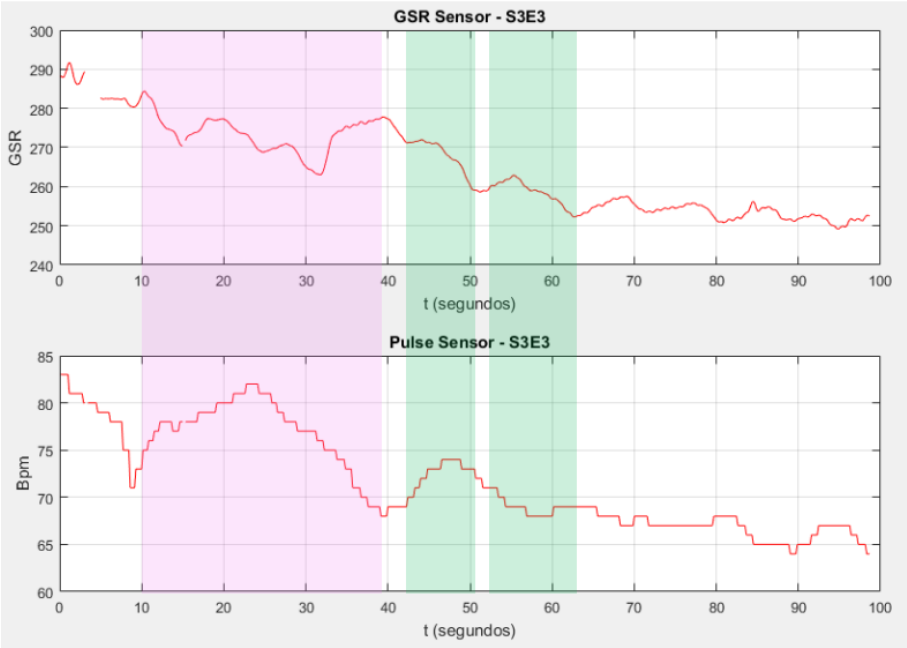


Figura 1-2. Coincidencia de Piloerección con bajada de GSR [6].

1.2 Alcance

Al comienzo del proyecto, se planteó utilizar el sistema de medidas de María para realizar un experimento con intérpretes musicales. La idea era medir las señales fisiológicas previamente mencionadas en una serie de músicos, mientras estos interpretaban algunas piezas. Así se buscaría una correlación entre la música interpretada y la emoción inducida en el artista. Sin embargo, finalmente se planteó un proyecto diferente.

En 2017, Alicia Fernández Sotos publica su tesis [7] sobre percepción de emoción en la música, y en ella se plantea un sistema de medidas fisiológicas como frecuencia cardíaca, actividad electrodérmica o tensión muscular. Este proyecto intenta emular el sistema mencionado, un ejemplo sería la Figura 1-3. El objetivo del proyecto es, por tanto, conseguir tomar el máximo número de medidas fisiológicas de forma sincronizada.

Figura 1-3. Ejemplo de dispositivos de captura de señales fisiológicas [7].

Se realizará la modificación e integración de sensores de medida de pulso cardíaco, conductancia de la piel, tensión muscular y movimiento ocular. Además, se utilizará un microcontrolador para programar el funcionamiento conjunto de dichos sensores, y se validará la funcionalidad de los mismos.

El proyecto no incluye la realización de experimentos, ni el procesado de los resultados obtenidos.

1.3 Requisitos Técnicos

Los requisitos técnicos del Proyecto se dividen en las siguientes categorías:

Funcionales:

- F.1: El sistema medirá las señales fisiológicas pulso cardíaco, conductancia de la piel, tensión muscular y movimiento ocular de un sujeto experimental.
- F.2: El sistema obtendrá las señales en tiempo real mediante un software específico que gestione la toma de datos.
- F.3: El sistema mostrará los datos ordenados.
- F.4: El sistema medirá el pulso cardíaco.
- F.5: El sistema tomará imágenes del ojo.

Prestaciones:

- P.1.1: El sistema obtendrá las señales de forma síncrona midiendo rangos de señal y realizando capturas fotográficas.
- P.3.1: El sistema mostrará los datos en forma de gráficas por pantalla.
- P.4.1: El rango del pulso cardíaco estará entre 0 y 250 pulsaciones por minuto.
- P.5.1: Las imágenes se tomarán con luz infrarroja, por lo que serán en blanco y negro.

Diseño:

- D.1: Las dimensiones del sistema no superarán 100x100x50 cm.
- D.2: El sistema estará formado por plástico y metal.
- D.3: El sistema no tendá un peso superior a 1Kg

Operación:

- O.1: El sistema se controlará desde un ordenador mediante la inicialización y visualización de la toma de datos por pantalla.

Restricciones:

- R.1: El sensor utilizado para medir el pulso cardíaco será el Pulse Sensor [10].
- R.2: El sensor utilizado para medir la conductancia de la piel será el GSR Sensor [13].
- R.3: El sensor utilizado para medir la tensión muscular será el Myoware Muscle Sensor [16].
- R.4: El sensor utilizado para medir el movimiento ocular será el implementado por Adán [17].

Seguridad:

- S.1: Todos los componentes utilizados deben cumplir la norma UNE-EN 301908-12 sobre compatibilidad electromagnética.
- S.2: Todos los sensores utilizados deben estar construidos con materiales biocompatibles.
- S.3: Todos los sensores utilizados deben tener protección contra descargas eléctricas.
- S.4: La alimentación de los sensores no debe estar conectada a la red eléctrica, sino a baterías u ordenadores portátiles.

2 OBJETIVOS

El proyecto va a consistir en la realización de un sistema síncrono de medidas de señales fisiológicas, a partir de los sensores mencionados previamente. Como se expondrá posteriormente en el apartado 4 (Sistema de Medidas), estos sensores ya han sido utilizados por otros alumnos en otros trabajos de fin de Grado. Por tanto, la complicación de este proyecto reside en la comprensión del funcionamiento de cada uno de los sensores, además de la integración de estos de forma síncrona. Conseguir obtener las señales de forma síncrona es fundamental, ya que solo así se pueden contrastar fenómenos puntuales entre unas señales y otras. Se plantean los siguientes objetivos en la realización del trabajo:

- Utilizar diferentes sensores para obtener señales fisiológicas de un usuario, de varias fuentes de señal distintas. En este caso, las señales a obtener son pulso cardíaco, conductancia de la piel, tensión muscular y movimiento ocular.
- Almacenar de forma organizada las señales recogidas en una base de datos, para facilitar el posterior uso de las mismas.
- Procesar y representar las señales obtenidas, para el posterior análisis de los datos recogidos.
- Aprender a trabajar con material y código de distintos profesionales, incorporándolos de forma sincronizada para la realización de experimentos, permitiendo así el control centralizado de los mismos.

3 PLANIFICACIÓN

Este proyecto se ha desarrollado durante junio y a lo largo del verano de 2020. A continuación se expone la planificación llevada a cabo.

	1-30 Junio	1-22 Julio	22-31 Julio	Agosto	1-14 Septiembre	15-30 Septiembre
Investigación Previa						
Definición de Objetivos						
Definición de Alcance						
Definición de Requisitos						
Estado del Arte						
Realización de la Parte Técnica						
Realización de la Memoria						
Revisión						
Presentación						
Tutorización						

Tabla 3-1. Planificación temporal del Proyecto.

La parte técnica corresponde a montar los circuitos necesarios y programar sus respectivos controladores, para la adquisición de datos de forma síncrona. El tiempo dedicado al proyecto consiste en 8 horas diarias entre semana, descansando los fines de semana.

4 SISTEMA DE MEDIDAS

4.1 Introducción

El sistema de medidas está formado por algunas placas de Arduino y una serie de sensores, que nos permiten obtener diferentes tipos de señales fisiológicas de forma sincronizada.

Los sensores empleados son los siguientes:

- **Sensor medidor de pulso cardíaco**
- **Sensor medidor de actividad electrodérmica de la piel (GSM)**
- **Sensor medidor de tensión muscular (electromiografía)**
- **Sensor medidor de movimiento ocular**
- **Cámara de vídeo**

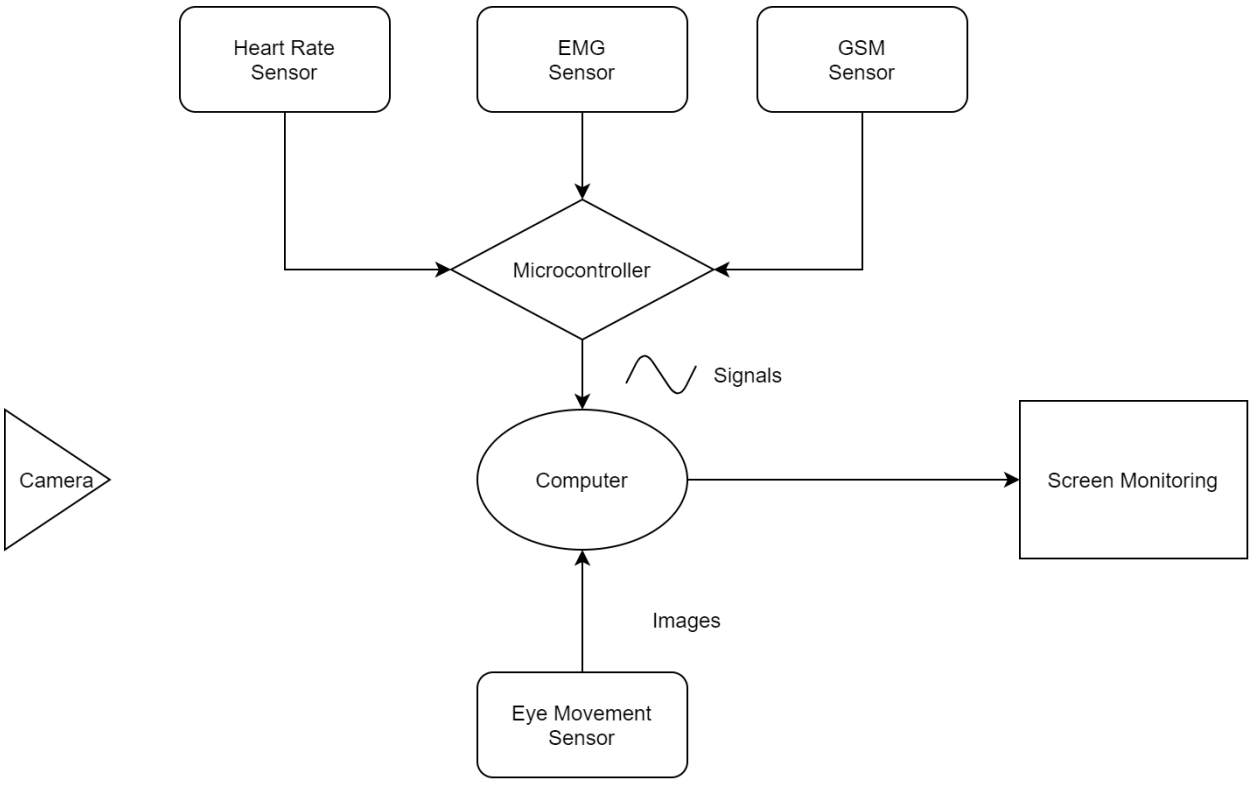


Figura 4-1. Esquema del Sistema Completo.

4.2 Componentes

A continuación, se exponen de forma detallada las características de cada uno de los sensores que conforman el sistema. Todos los códigos completos se encuentran en los anexos al final de la memoria.

4.2.1 Sensores conectados a mirocontrolador

a) Placa de Arduino Uno™ (Bloque Principal)

Para los tres primeros sensores se utiliza como interfaz una placa de Arduino Uno™ [8], haciendo uso además de una Protoboard para facilitar el conexionado. En la Figura 4-2, los tres cables que entran a la izquierda corresponden a las salidas de los sensores (entradas analógicas en la placa), y los dos de la derecha (rojo y marrón) salen a la Protoboard como Vcc (5V) y tierra (GND) respectivamente.

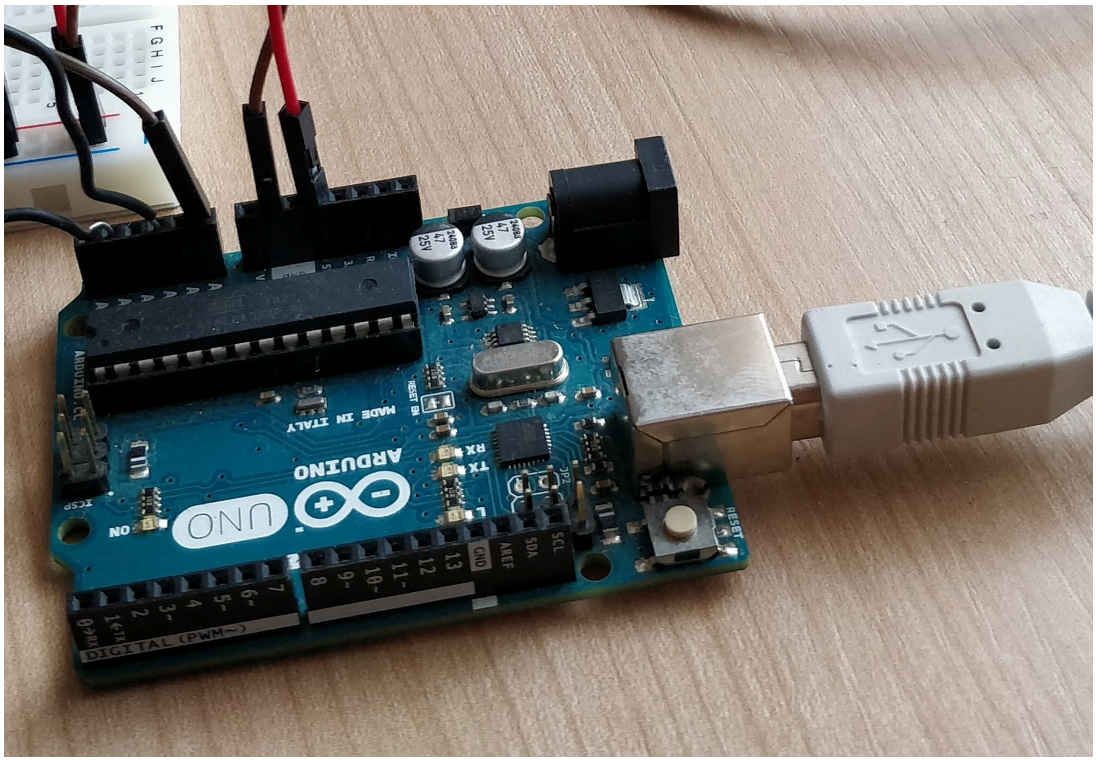


Figura 4-2. Placa de Arduino Uno.

Dado que este bloque debe controlar la captura de datos de los sensores, se ha decidido dividir el código de Arduino en partes. En este caso tenemos el programa principal, "Sensores.ino", que se encarga de instanciar la lectura de cada uno de los sensores por puerto serie. Esto se hace de forma escalonada, ya que así conseguimos recibir los datos en el orden adecuado para poder procesarlos individualmente según el sensor (Figura 4-3). Posteriormente, en el apartado de sincronización se comentará algo más acerca de este asunto.

```
//llamamos previamente a la funcion
Serial.print(gsr());
Serial.print(",");

//llamamos previamente a la funcion
Serial.print(emg());
Serial.print(",");


//llamamos previamente a la funcion
Serial.print(pulse());
Serial.println("");
```

Figura 4-3. Llamada a las funciones de cada sensor.

b) Pulse Sensor

Se trata de un sensor para Arduino que permite medir el pulso cardíaco (pulsaciones por minuto). Como explica María en su TFG [6] en la sección 4.4.3, el sensor funciona por fotoplestimografía, gracias a un led verde que incluye. A la hora de utilizarlo, debemos esperar un tiempo de establecimiento de entre 10 y 20 segundos, consiguiendo así empezar a medir de forma fiable. En este caso el sensor está conectado a la entrada analógica de Arduino Uno "A0", como se puede observar en la Figura 4-6. En la Figura 4-4 podemos observar el código utilizado para configurar la función del sensor de pulso, posteriormente llamada desde el código principal ("Sensores.ino"). En la Figura 4-5 podemos ver la inicialización del sensor, necesaria para su correcto funcionamiento. Estos códigos han sido tomados del trabajo de María [6], y modificados ligeramente para adaptarlos a una mayor cantidad de sensores.

```
int pulse(){ //función para PulseSensor
  /*
    * Introducimos un cierto retraso entre muestra y muestra a enviar
    * debido a que la velocidad de transmisión no admite tantas E/S
  */
  delay(20);
  pulseSensor.outputSample();
  pulseSensor.outputBeat();
  /*
    * Si se ha tenido un latido desde la última vez que se comprobó,
    * se escribirá la información por latido en serie.
  */
  if (pulseSensor.sawStartOfBeat()) {
    pulseSensor.outputBeat();
  }
}
```

Figura 4-4. Configuración de la función del Pulse Sensor.

```
pulseSensor.analogInput(PIN_INPUT);
//pulseSensor.blinkOnPulse(PIN_BLINK);
//pulseSensor.fadeOnPulse(PIN_FADE);

pulseSensor.setSerial(Serial);
pulseSensor.setOutputType(OUTPUT_TYPE);
pulseSensor.setThreshold(THRESHOLD);

if (!pulseSensor.begin()) { //en el caso de que el PulseSensor no se inicie
  /*
    PulseSensor initialization failed,
    likely because our particular Arduino platform interrupts
    aren't supported yet.
    If your Sketch hangs here, try ProcessEverySample.ino,
    which doesn't use interrupts.
  */
  for(;;) {
    // Flash the led to show things didn't work.
    digitalWrite(PIN_BLINK, LOW);
    delay(100);
    digitalWrite(PIN_BLINK, HIGH);
    delay(100);
  }
}
```

Figura 4-5. Inicialización del Pulse Sensor.

En la Figura 4-6, podemos observar el conexionado del Pulse Sensor, con dos pines a la alimentación (+ y – de la fila derecha de la Protoboard) y otro al pin “A0” del Arduino Uno.

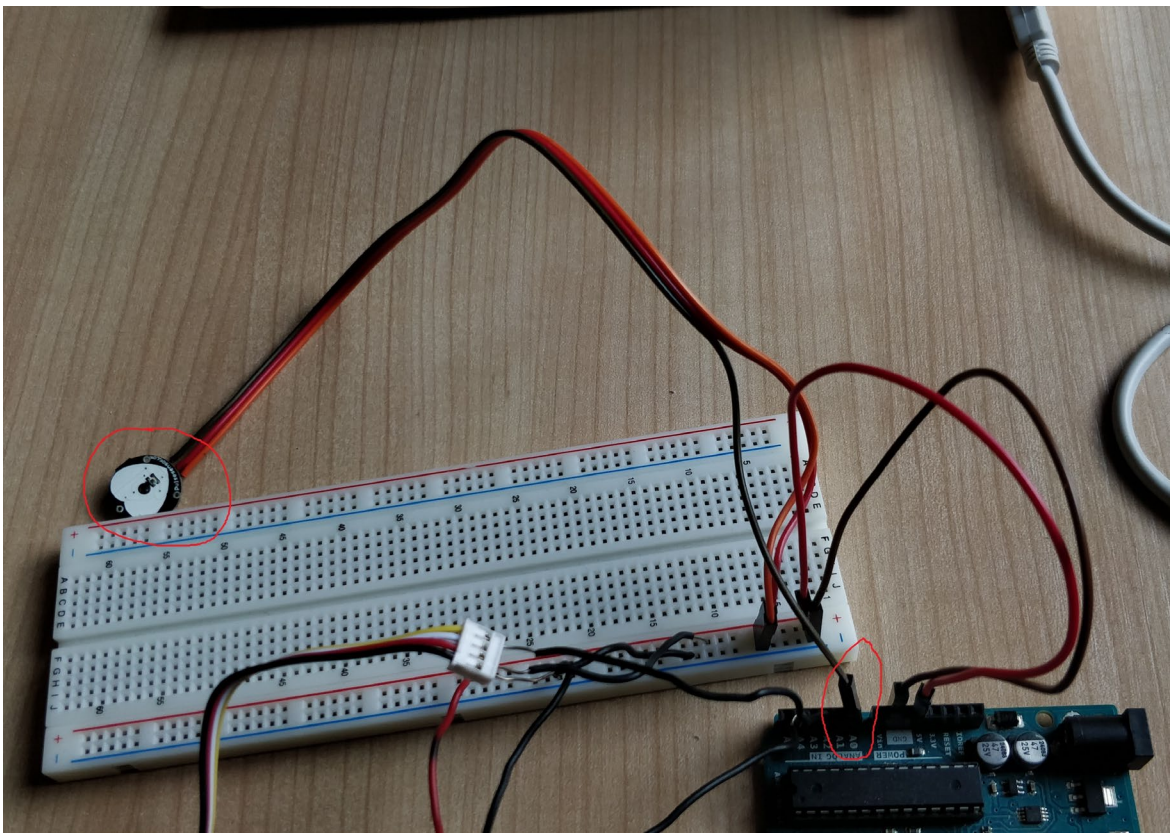


Figura 4-6. Conexionado del Pulse Sensor (Pin “A0”).

Por último, es importante comentar que se necesita instalar la librería PulseSensor Playground [9] de Arduino para el correcto funcionamiento del código.

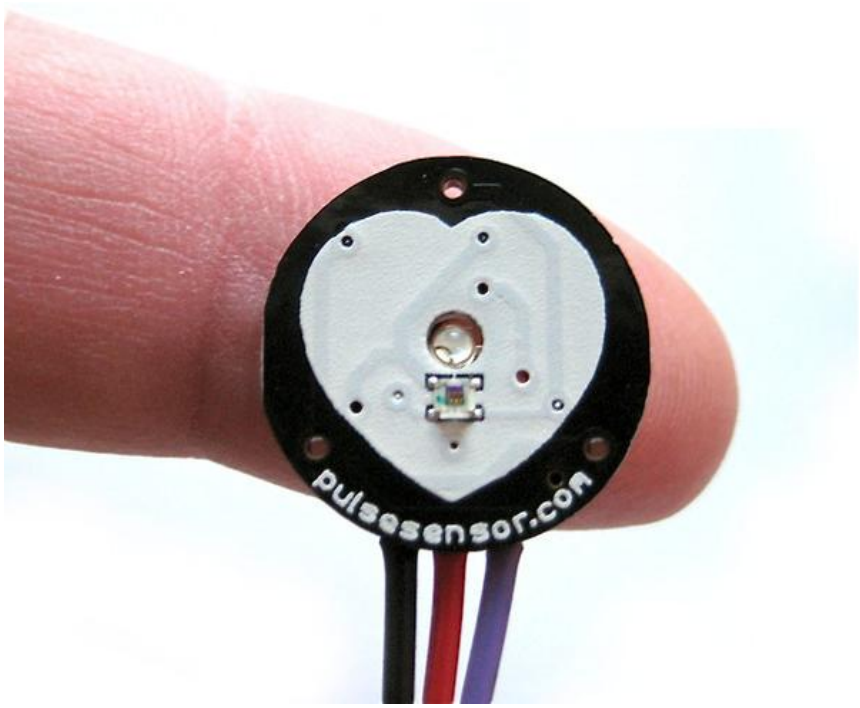


Figura 4-7. Pulse Sensor [10]

c) GSR Sensor

Este sensor se encarga de medir la EDA de la piel. En el TFG de María [6] se comenta un modo de empleo que será el elegido en este caso para poder utilizar sus resultados de forma fiable. Los electrodos se colocan en las falanges medias de los dedos índice y corazón, como se observa en la Figura 4-8.

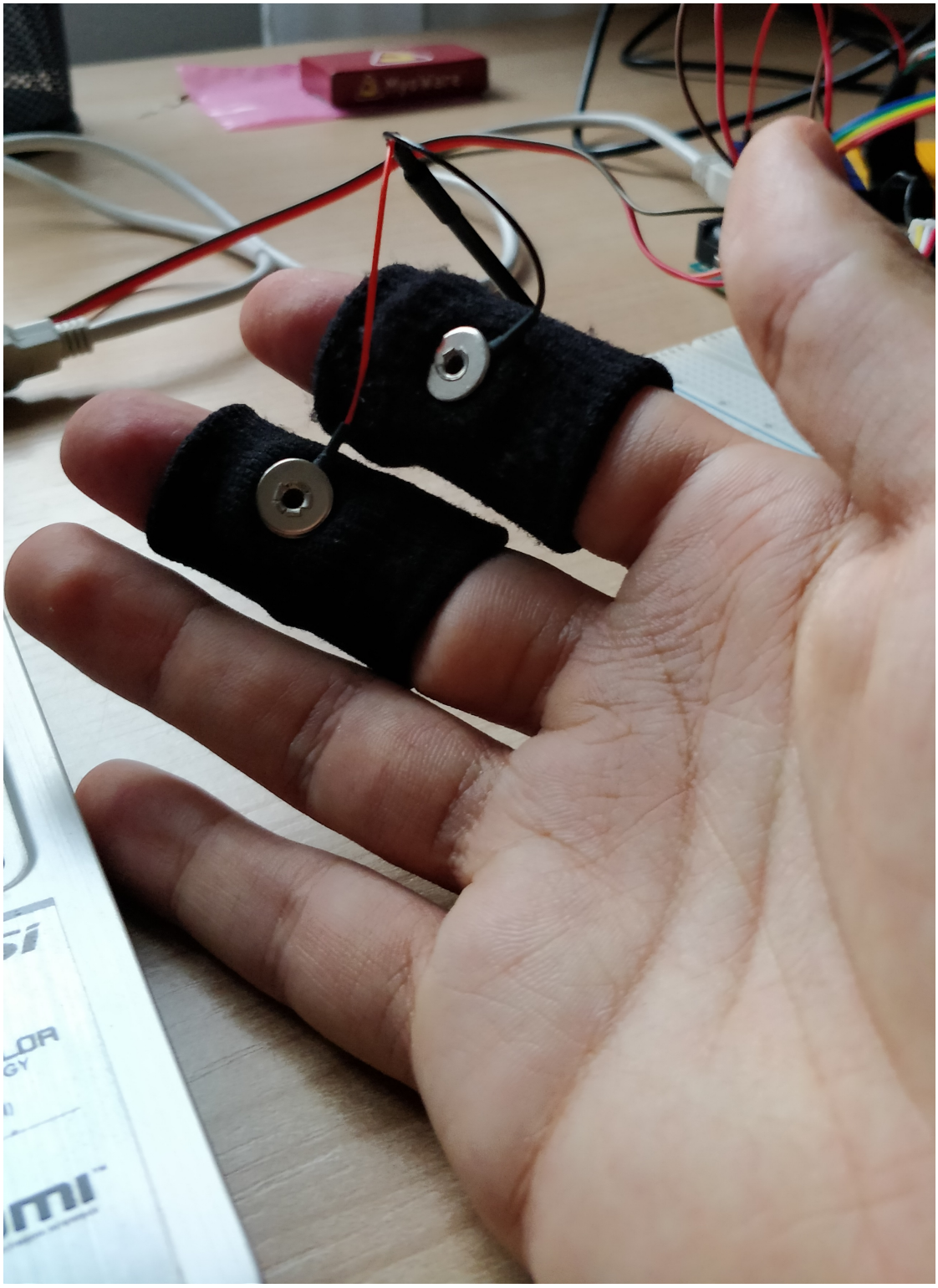


Figura 4-8. Colocación del sensor GSR en los dedos índice y corazón.

Este sensor funciona aplicando una tensión entre los dos electrodos, y se mide el flujo que circula a través de la piel. Este va a depender de la resistencia de la piel, que varía según la sudoración y la piloerección (Figura 4-9).

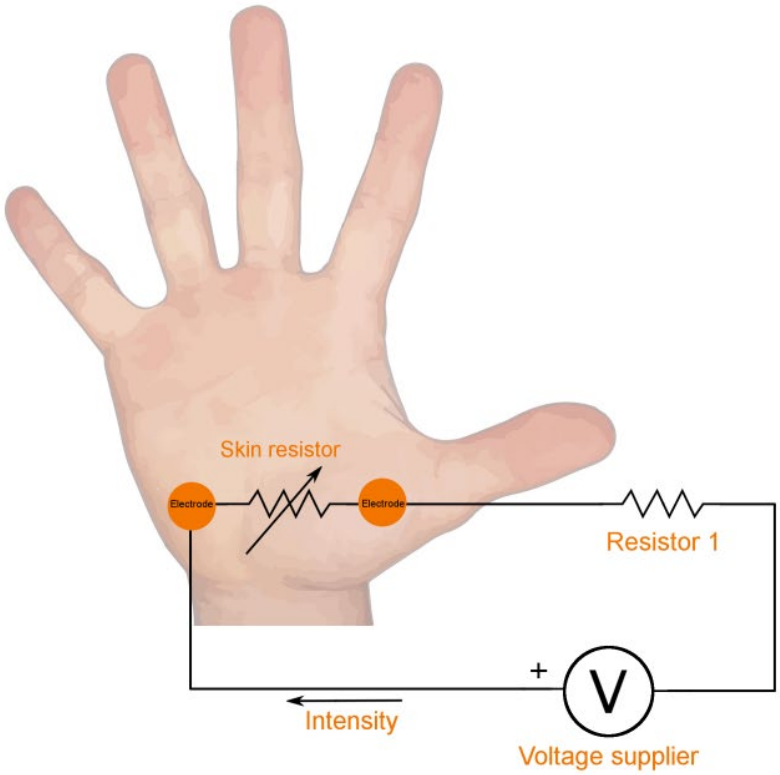


Figura 4-9. Circuito equivalente del GSR Sensor [11].

Con el código utilizado por María [12], se obtiene un parámetro en función de la conductancia de la piel, que nos permite estudiar las reacciones de piloerección (Figura 4-10).

```
int gsr(){ //función para GSR Sensor
  long sum=0;
  for(int i=0;i<10;i++) //hacemos el promedio entre 10 muestras
  {
    sensorValue=analogRead(GSR);
    sum += sensorValue;
    delay(15);
  }
  gsr_average = sum/100;
  Serial.print(gsr_average); //dato a mostrar por pantalla
}
```

Figura 4-10. Obtención de datos del GSR Sensor por promedio.

En cuanto a la conexión, en la Figura 4-11 observamos que el GSR Sensor está conectado al pin “A2” de entrada de Arduino y a Vcc y GND en la Protoboard.

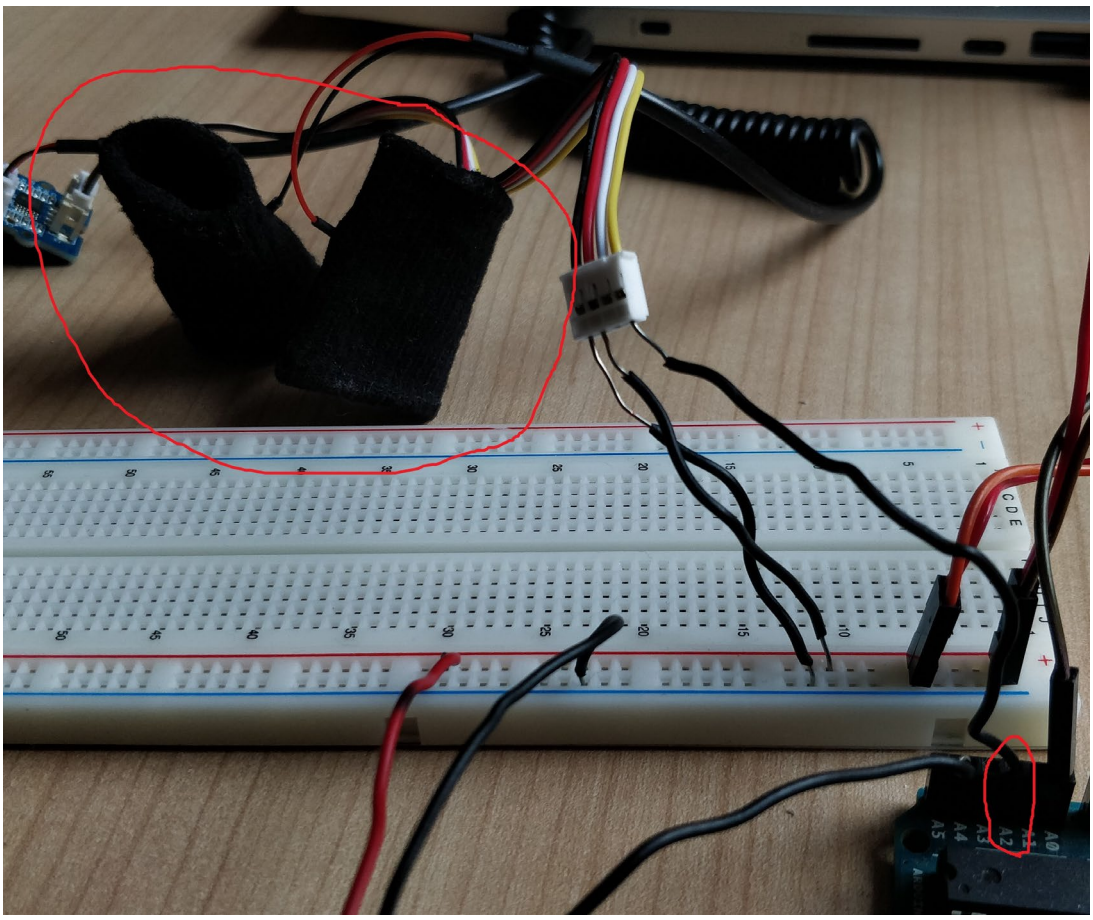


Figura 4-11. Conexionado del GSR Sensor (Pin “A2”).

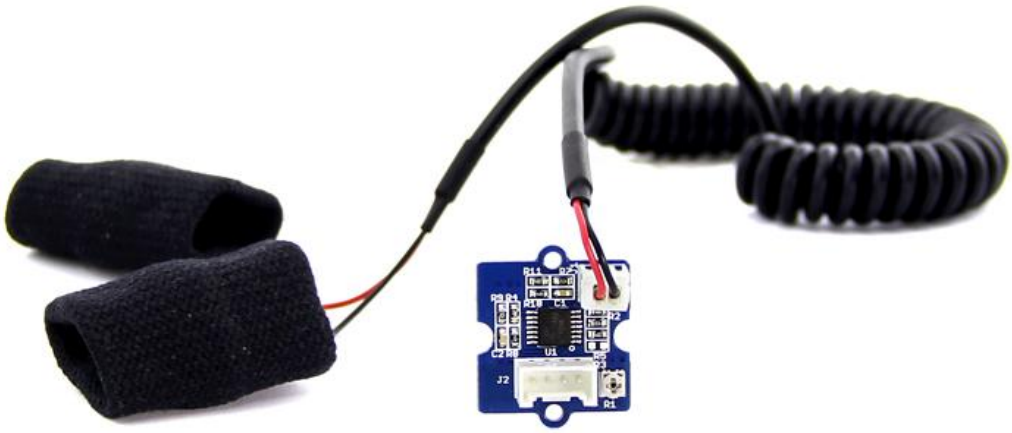


Figura 4-12. GSR Sensor [13].

d) EMG Sensor

Este sensor mide la tensión muscular a través de la aplicación de un potencial eléctrico sobre el músculo. El sensor fue utilizado por Jesús Suárez Luque en su TFG [14] para tratamiento del Bruxismo. En este caso se colocaba el sensor en la mandíbula del paciente para comprobar la fuerza aplicada por la misma mientras dormía, diagnosticando así la enfermedad. En este caso, nos interesa relacionar la fuerza de la mandíbula como señal fisiológica con la emoción experimentada por un sujeto experimental, mientras escucha música por ejemplo (Figura 4-13).

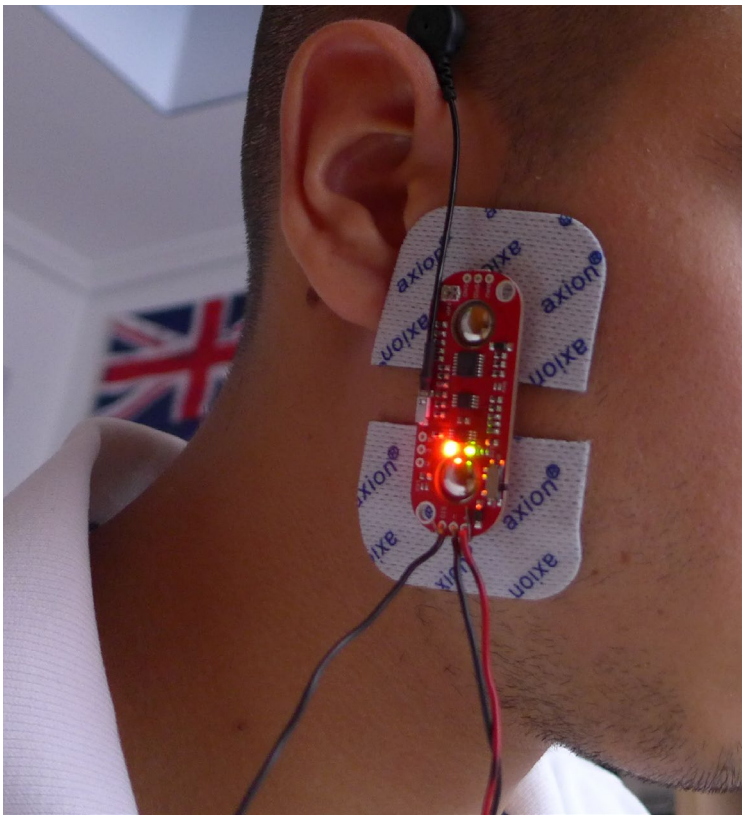


Figura 4-13. Colocación del sensor EMG en el músculo masetero de la mandíbula.

El código para sacar los datos del sensor está sacado de la web del sensor [15], y modificado para implementarlo junto a los otros sensores y obtener valores promedio (Figura 4-14). El parámetro recibido por puerto serie es una tensión entre 0V y 5V, de menos a más fuerza ejercida por el músculo.

```
int emg(){ //función para EMG
  float sum=0;
  for(int i=0;i<10;i++) //hacemos el promedio entre 10 muestras
  {
    sensEMG=analogRead(EMG);

    // Convert the analog reading (which goes from 0 - 10230) t
    float voltage = sensEMG * (5.0 / 10230.0);
    sum += voltage;
    delay(15);
  }
  emg_average = sum;
  Serial.print(emg_average); //dato a mostrar por pantalla
}
```

Figura 4-14. Obtención de datos del sensor EMG por promedio.

En la Figura 4-15 observamos la conexión, usando el pin de entrada “A4” de Arduino, y las conexiones a Vcc y GND en la Protoboard.

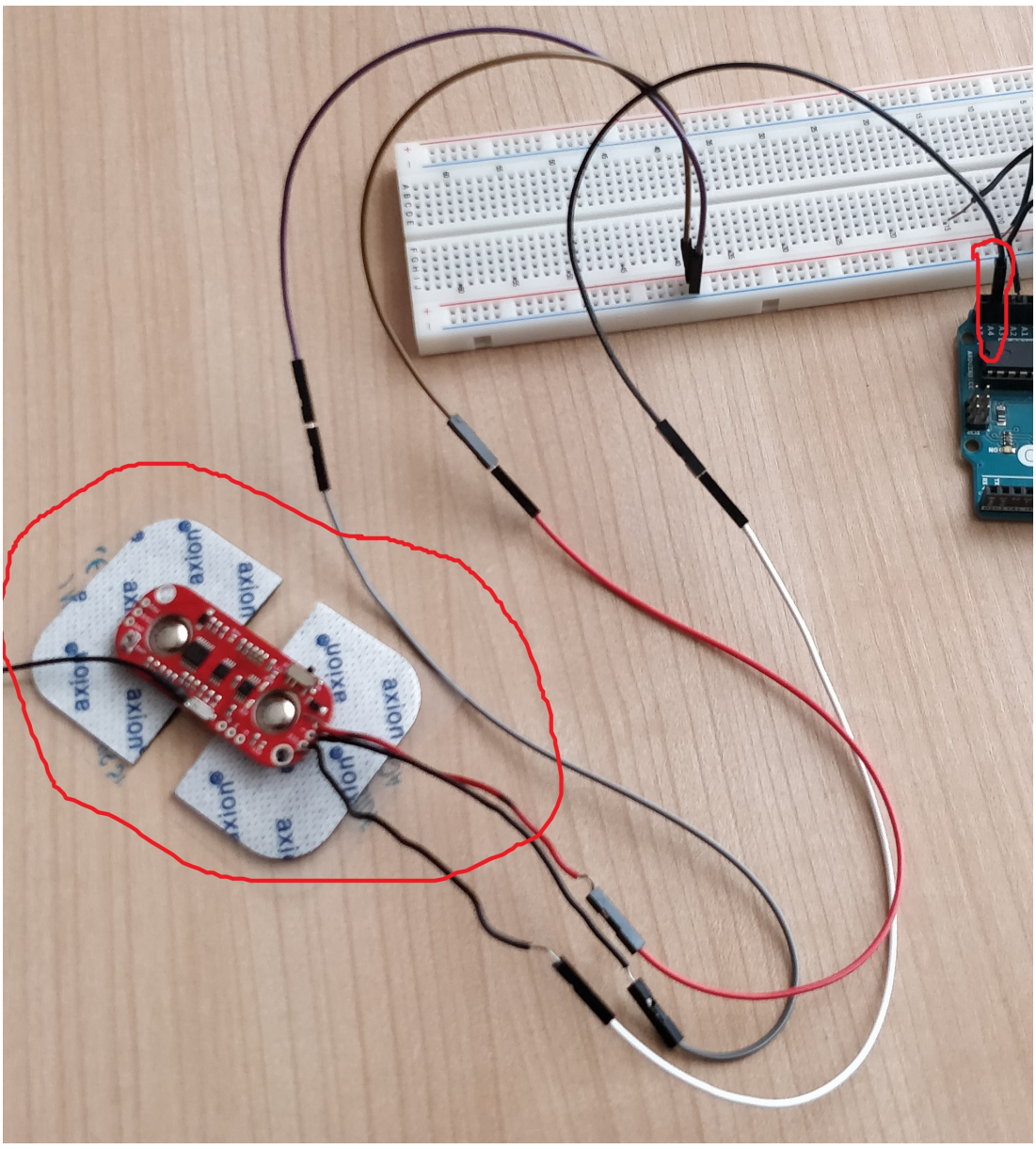


Figura 4-15. Conexionado del EMG Sensor (Pin “A4”).

Figura 4-16. EMG Sensor [16].

4.2.2 Sensor Electrooculograma

Este sensor detecta el movimiento del ojo a partir de una cámara web sin el filtro de infrarrojos, además de cuatro leds de infrarrojo alrededor de ella (Figura 4-17). Está integrada en un soporte similar a unas gafas de realidad virtual, que fue utilizado por Adán Montero en su TFG [17]. La cámara está conectada a una placa de Arduino Pro Mini [18], que a su vez va conectada al ordenador para transmitir los datos por puerto serie. Además, la cámara está conectada por otro USB a la alimentación (Figura 4-18).

Figura 4-17. Webcam sin filtro de infrarrojos rodeada de cuatro leds infrarrojos.

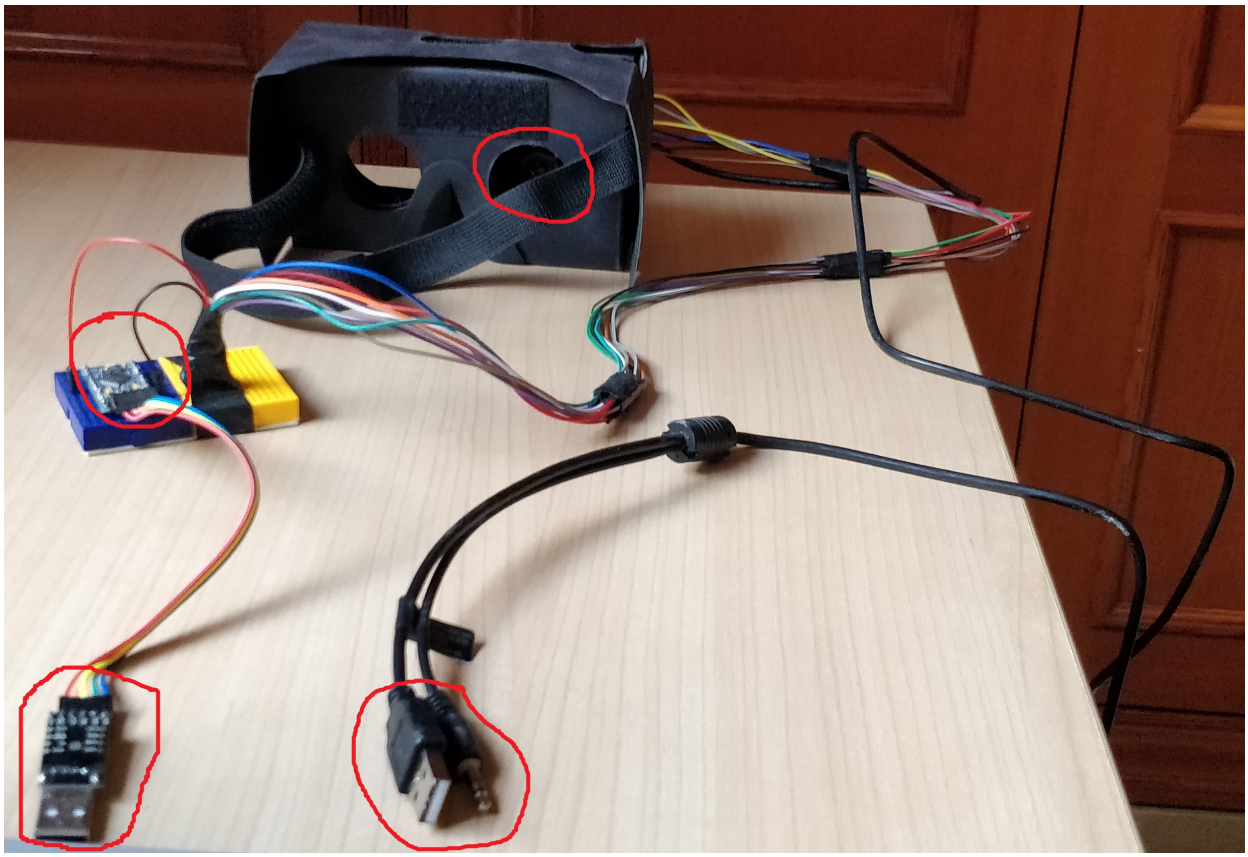


Figura 4-18. Sensor Electrooculograma y Arduino Pro Mini con conexiones.

El código Arduino de Adán nos permite leer la información por puerto serie, además de controlar el encendido de los leds infrarrojos, conectados al pin 13 de la placa Arduino (Figura 4-19).

```
if(Serial.available()>0) // if there is data to read
{
 matlabData=Serial.read(); // read data

  if (matlabData==3)
  //blinkState = !blinkState;
  digitalWrite(ledPin2,HIGH); // turn light on
}
```

Figura 4-19. Código Arduino para el sensor Electrooculograma.

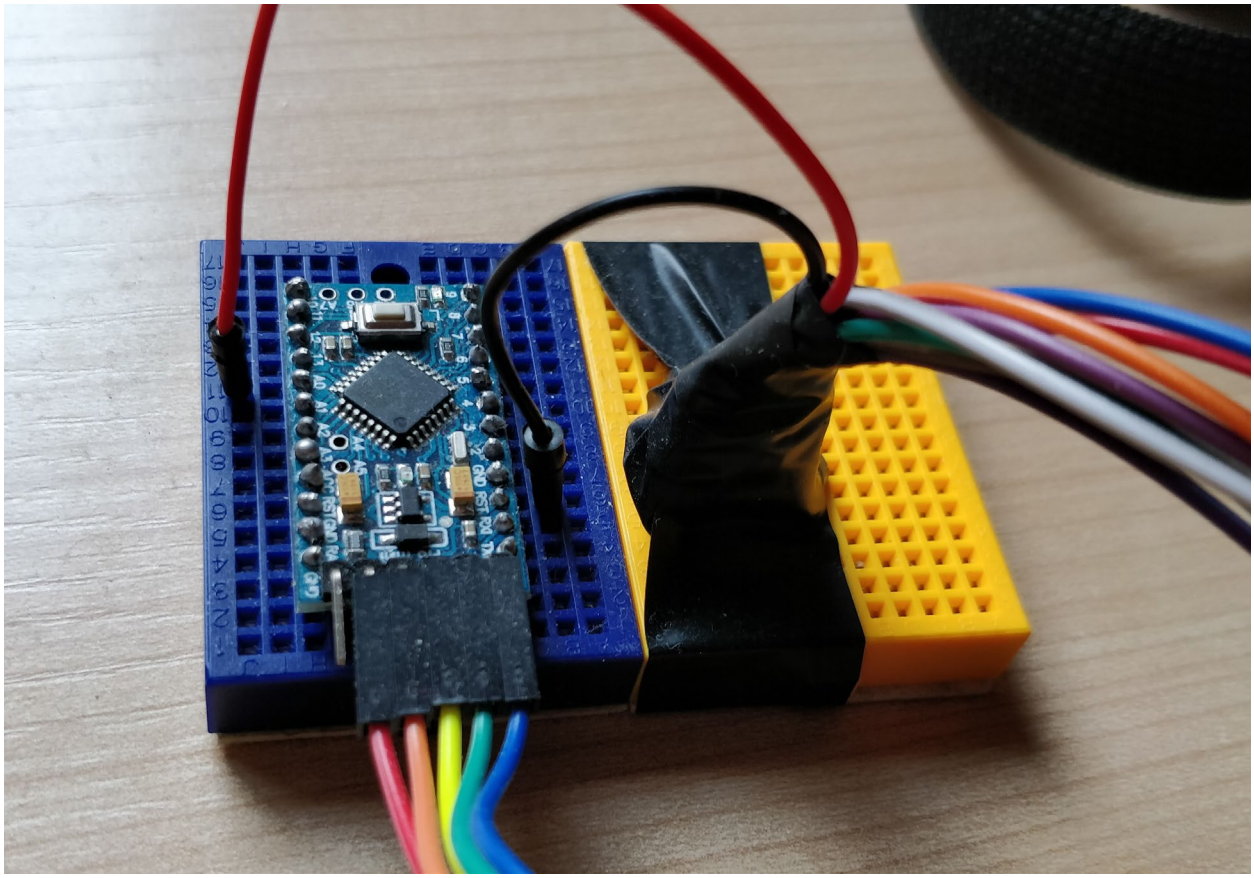


Figura 4-20. Conexionado Arduino Pro Mini.

4.2.3 Cámara Panasonic Lumix DMC-LX7

Con la cámara Panasonic Lumix DMC-LX7 [19] (Figura 4-21), se pretende grabar el experimento que se realice con este sistema. Además, se puede utilizar para contrastar información de los sensores, como la búsqueda de piloerección por ejemplo.

Figura 4-21. Cámara Panasonic Lumix DMC-LX7.

4.3 Sincronización

Después de haber descrito cada uno de los sensores utilizados, ahora viene la parte más importante del proyecto: conseguir la sincronización de los datos que envían estos sensores. Para ello, se han utilizado diferentes técnicas en función del modo de adquisición de los datos en cada uno de los casos. Además, se ha utilizado un USB Hub que nos permite conectar varios USB a un mismo puerto, ya que se necesitan más puertos de los que tiene el ordenador. El dispositivo elegido es USB Hub de Black UFO [20], y su conexión se puede apreciar en la Figura 4-22.

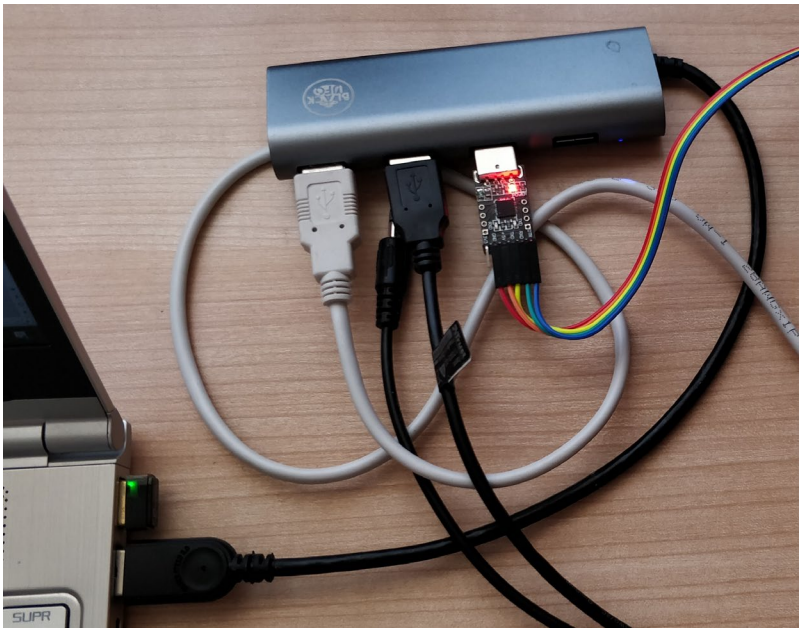


Figura 4-22. Conexionado del USB Hub de Black UFO [20].

El motivo por el que la sincronización es un elemento fundamental es la posibilidad de contrastación entre señales. Por ejemplo, si en un experimento se mide la piloerección, se ha comprobado en [6] que tanto la señal de pulso cardíaco como la de conductancia de la piel se ven afectadas. Esto implica que hay fenómenos que se ven reflejados en varias señales fisiológicas, por lo que podemos utilizar los datos recogidos de unas para contrastar los resultados de las otras, ganando fiabilidad en las conclusiones del experimento.

4.3.1 PLX-DAQ

En primer lugar, hablaremos de este complemento para Excel, ya utilizado por la autora en [6]. Este complemento nos permite leer los datos transmitidos por puerto serie, consiguiendo así poder sincronizar la lectura de los sensores conectados al Arduino Uno (Pulso, GSM y EMG). Tal y como se planteó en [6], se imprime por puerto serie la hora, la suma de los tiempos entre medidas y las medidas de los sensores, como se puede observar en la Figura 4-23. Una vez hecha la implementación del EMG, se añade su contribución al sistema, consiguiendo así la recogida síncrona de datos de los tres primeros sensores.

Hora	Tiempo	GSR	EMG	BPM	tiempo entre latidos	Pulso
20:21:13	0.32	520	3.71	0	750	501
20:21:13	0.64	520	4.49	0	750	501
20:21:13	0.97	530	3.25	0	750	499
20:21:14	1.29	530	2.76	0	750	494
20:21:14	1.62	540	4.4	0	750	506
20:21:14	1.94	540	4.75	0	750	506
20:21:15	2.27	540	4.17	0	750	504
20:21:15	2.59	550	2.84	0	2332	504
20:21:15	2.91	550	2.69	0	2332	503
20:21:16	3.24	550	4.22	0	2332	496
20:21:16	3.56	550	4.63	0	2332	507
20:21:16	3.88	550	3.96	0	2332	508
20:21:16	4.21	550	3.06	37	1588	524
20:21:17	4.54	560	2.59	37	1588	513
20:21:17	4.86	550	3.75	37	1588	505
20:21:17	5.18	550	4.33	39	974	508
20:21:18	5.51	550	4.23	39	974	499
20:21:18	5.83	550	2.99	41	860	513
20:21:18	6.16	550	2.34	41	860	513
20:21:19	6.48	550	3.07	41	860	516
20:21:19	6.80	550	4.3	43	972	505
20:21:19	7.13	550	3.41	43	972	509
20:21:20	7.45	540	1.5	43	972	511
20:21:20	7.77	520	3.66	45	942	497
20:21:20	8.10	510	1.57	45	942	518
20:21:21	8.43	510	1.6	45	942	501
20:21:21	8.75	520	2.31	45	942	510
20:21:21	9.08	520	3.95	45	942	489

Figura 4-23. Sincronización de los sensores de Arduino Uno con PLX-DAQ [21].

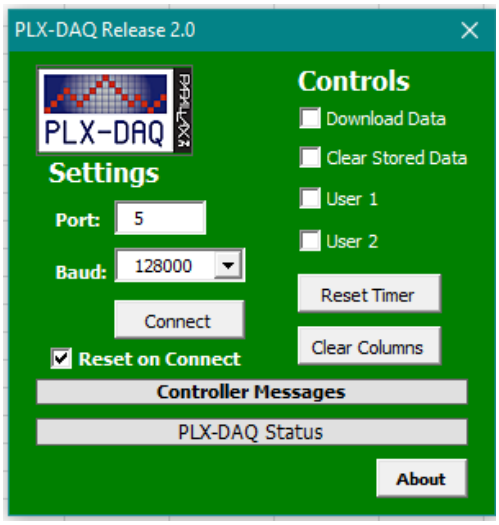


Figura 4-24. Interfaz del PLX-DAQ.

4.3.2 Imágenes guardadas desde Matlab con la hora actual

En cuanto al Electrooculograma, se ha decidido tomar una solución muy sencilla al problema de la sincronización. En el código de Matlab del autor en [17] se guardan capturas del ojo cada cierto tiempo, por lo que se ha decidido guardar la hora directamente en el nombre de las capturas (Figura 4-25). Contrastando con la hora de las medidas tomadas en el PLX-DAQ, ya tendríamos las capturas del electrooculograma sincronizadas con las medidas de los otros tres sensores.

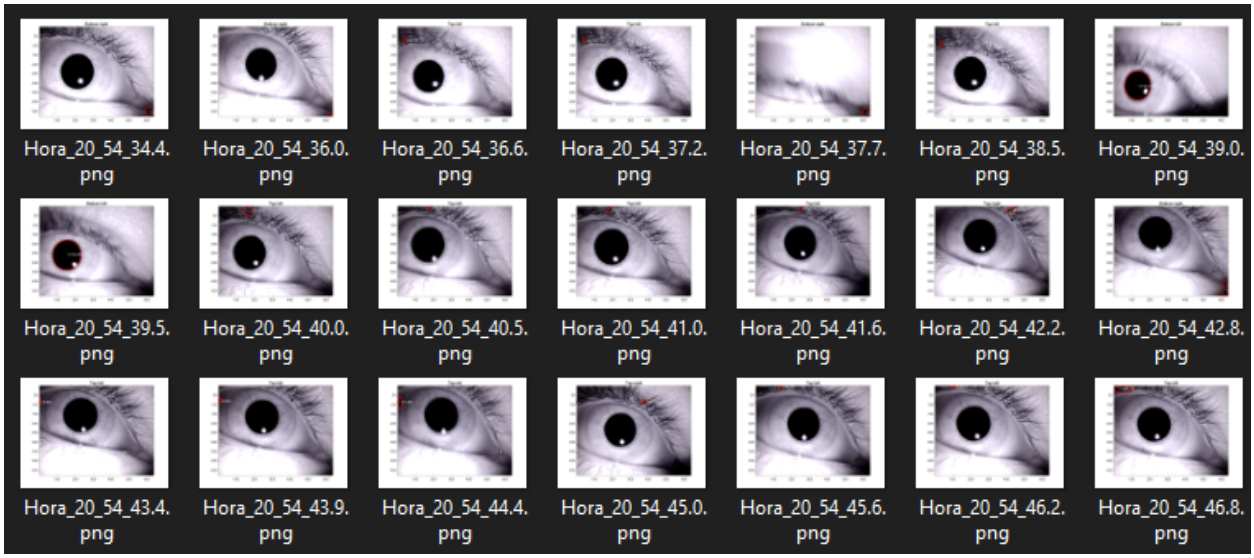


Figura 4-25. Capturas guardadas con la hora en que se toman

El código que nos permite realizar esto es el mostrado en la Figura 4-26, mediante la función clock de Matlab. Configurando el formato, se guarda la captura con dos enteros de hora y minuto, y un decimal para los segundos.

```
    moment=clock;
%% save current figure in a folder. IMPORTANT !!! Set your directory in the following command
saveas(gcf,['/Users/luigy/Desktop/Img_Cam/Hora_', sprintf('%02d_', moment(4)),
    sprintf('%02d_', moment(5)), sprintf('%.1f', moment(6)),'.png']);
```

Figura 4-26. Código para guardar las capturas con la hora en el nombre.

4.3.3 Grabación con la Cámara de pantalla y sujeto experimental

A la hora de realizar el experimento, se ha decidido mostrar en la pantalla del ordenador la hoja Excel con los resultados de los primeros tres sensores, Matlab abierto con las capturas del Electrooculograma y un reloj con la hora exacta del sistema, de tal forma que la cámara capture la hora a la que se toman las medidas, consiguiendo así la sincronización total de los sensores. Para esto, se ha decidido instalar un complemento para Windows 10 llamado Widget Launcher [22], que permite colocar Widgets en el escritorio del ordenador. En este caso el Widget elegido es un reloj digital, que podemos observar en la Figura 4-27.

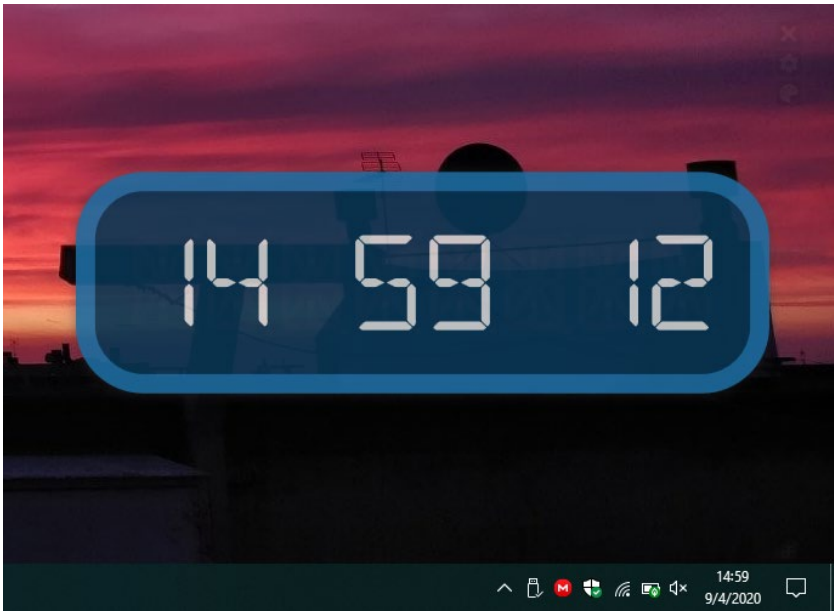


Figura 4-27. Widget de reloj digital para el escritorio.

Por último, en la Figura 4-28 podemos observar a un sujeto experimental con todos los sensores colocados en su posición, y midiendo las señales fisiológicas. En la mano derecha, el sensor de pulso mide el pulso cardíaco. En la mano izquierda, el sensor GSR mide la la conductancia de la piel. En la mejilla derecha, el sensor EMG mide la tensión muscular ejercida por la mandíbula. Por último, en el soporte de gafas de realidad virtual, el sensor electrooculograma toma capturas del movimiento del ojo. Así, el sistema mide todas las señales fisiológicas de forma sincronizada.

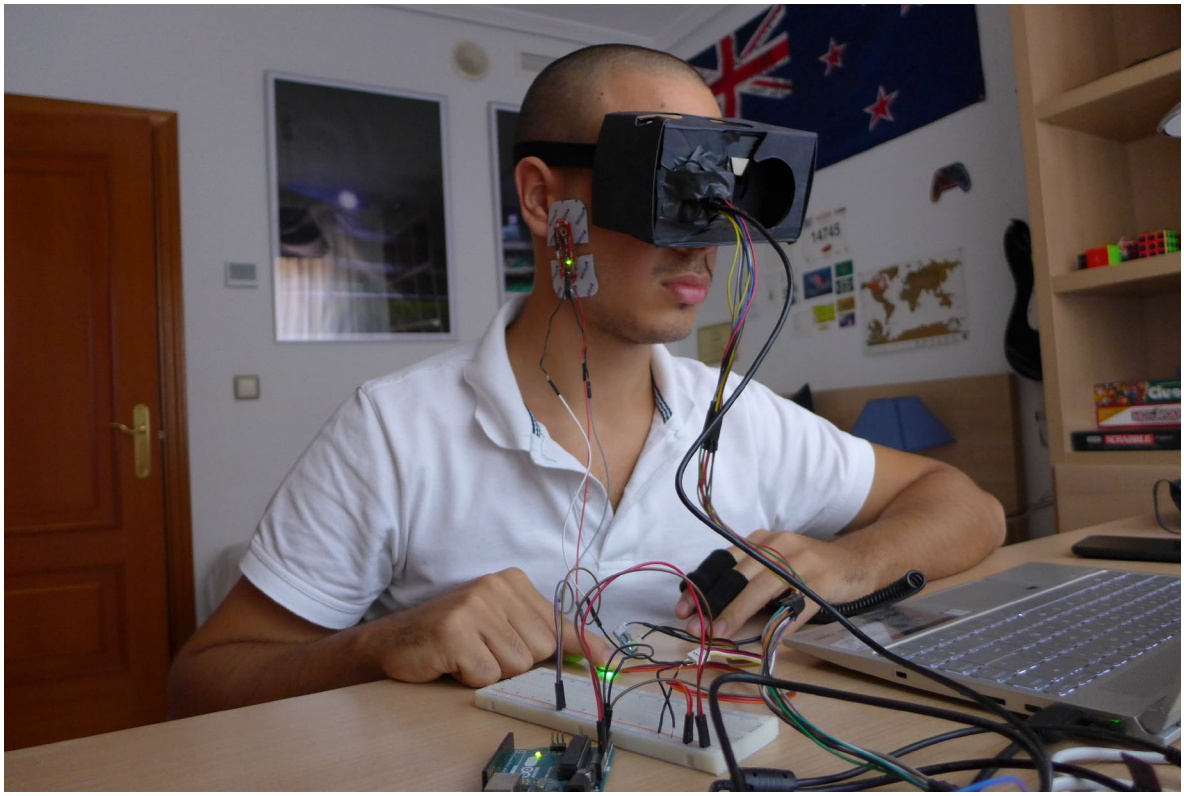


Figura 4-28. Sujeto experimental con los cuatro sensores midiendo.

5 PROCESAMIENTO DE LOS DATOS

5.1 Procesamiento en Matlab

En primer lugar, hablaremos del procesamiento de los datos de los sensores que enviaban información a través del puerto serie de Arduino Uno. Esta información, como se comentó en el apartado 4.3.1 de Sincronización, se guarda en una hoja Excel con el complemento PLX-DAQ. A continuación, esta base de datos se importa a Matlab, desde donde se procesarán los datos (Figura 5-1 y Figura 5-2).

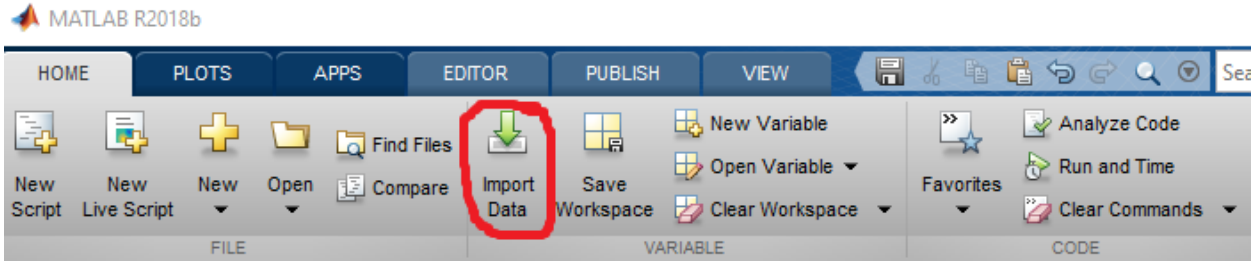


Figura 5-1. Importación de los datos a Matlab (I).

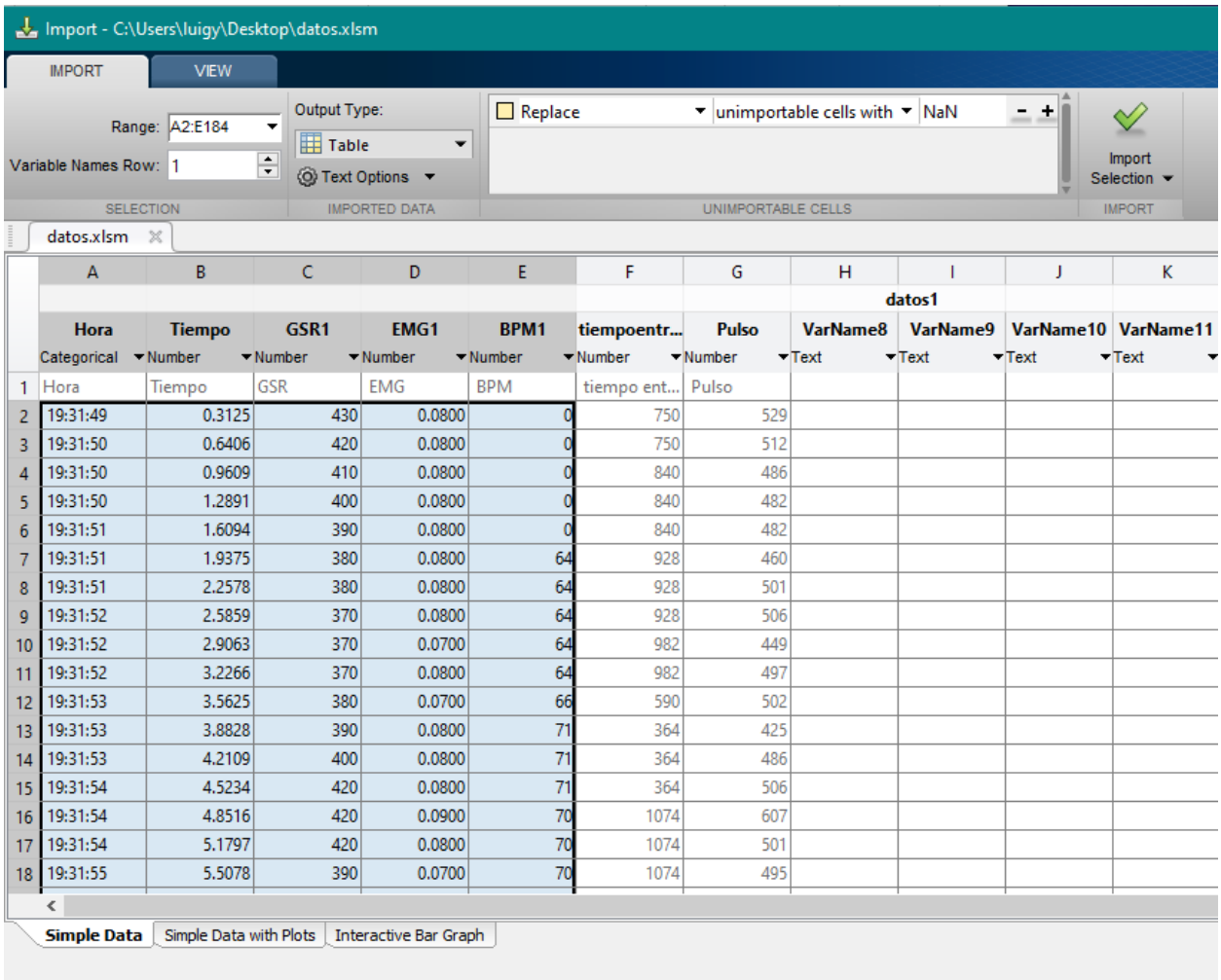


Figura 5-2. Importación de los datos a Matlab (II).

Una vez se han importado los datos en Matlab, se procesan como vectores para poder representar gráficamente los datos en función del tiempo. Para ello, se ha decidido representar GSR, EMG y BPM en función del tiempo transcurrido. Además, se ha representado también la hora en función del tiempo (Figura 5-3).

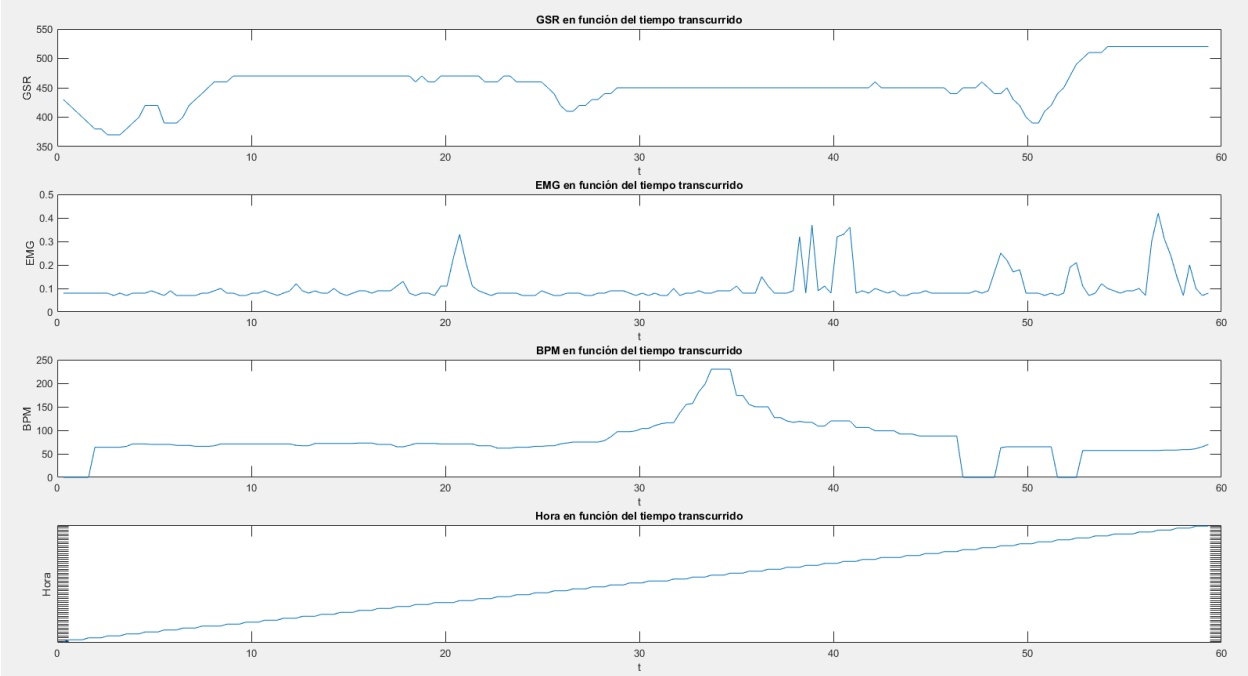


Figura 5-3. GSR, EMG, BPM y Hora en función del tiempo transcurrido.

El motivo por el que no se representan los datos en función de la hora directamente, es que el tiempo transcurrido nos da más precisión (hay varios valores de tiempo transcurrido por cada segundo). En cuanto a las imágenes recogidas por el electrooculograma, como se dijo en la sección 4.3.2, se guardan con la hora en la que fueron capturadas. Por tanto, a partir de las gráficas de la Figura 5-3, se puede elegir una hora determinada en la que interese comprobar las fotografías del electrooculograma, fáciles de encontrar gracias a su nombre. Posteriormente, en la sección 5.2, se expondrá un ejemplo que ilustrará de forma más clara este caso.

5.2 Validación del Sistema

Se ha realizado un experimento en un entorno tranquilo y silencioso, de poca luminosidad, intentando así crear un clima de concentración adecuado y evitar interferencias externas a las medidas. Se han tomado las señales fisiológicas de un sujeto experimental mientras escuchaba una canción. En la Figura 5-4 se puede observar una captura del vídeo grabado por la cámara, apuntando a la pantalla del ordenador para conocer la hora en todo momento, como se indicaba en la sección 4.3.3.

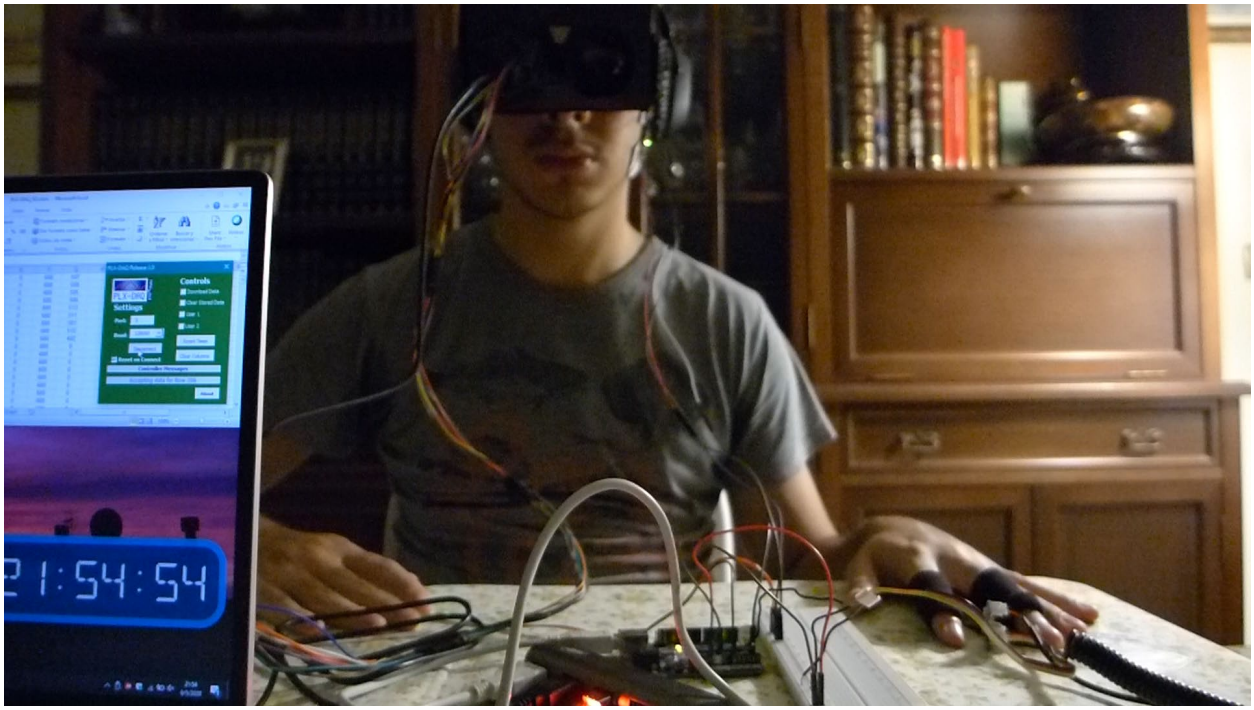


Figura 5-4. Cámara grabando sujeto experimental y hora.

En el vídeo se puede observar que el sujeto termina de acomodarse en torno a las 21:54:21, por lo que se analizarán las señales fisiológicas a partir de esa hora. También se observa en el vídeo que el sujeto comienza a desprenderse de los sensores al terminar la canción, a las 22:00:28, por lo que tampoco se tendrán en cuenta las medidas a partir de esa hora. Se importa por tanto la base de datos de señales fisiológicas desde las 21:54:21 hasta las 22:00:28. En la Figura 5-5 se observan dichas señales en función del tiempo.

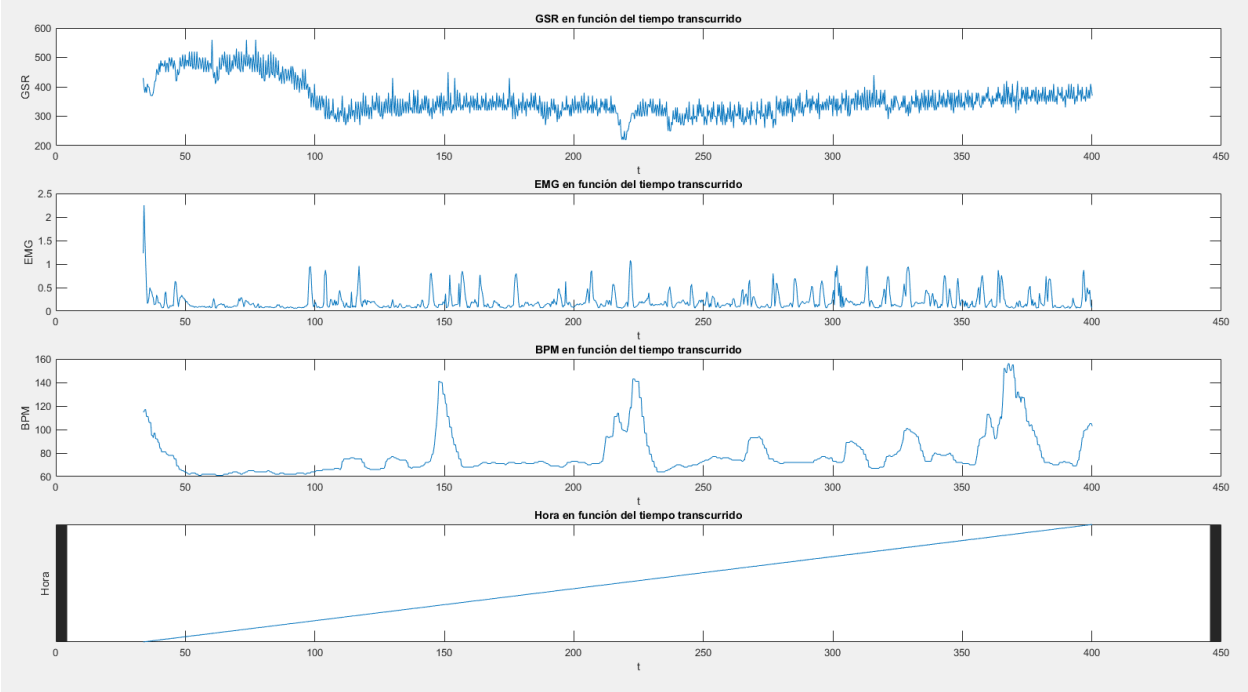


Figura 5-5. Señales Fisiológicas medidas en función del tiempo transcurrido.

No hubo casos de piloerección, debido en parte a la incomodidad del sujeto experimental por la invasividad de los sensores, especialmente el electrooculograma. De hecho, se puede observar un incremento en la media y la irregularidad del pulso cardíaco debido a esta incomodidad a lo largo del tiempo. En cuanto al sensor EMG, la fuerza aplicada por la mandíbula es bastante baja de media (en torno a 0.1V de un máximo de 5V), con algunos picos de hasta 0.5V, que también se ven incrementados conforme avanza el tiempo. El parámetro de la GSR se mantiene en torno a 320 de media, aunque al final también sube ligeramente. Todos los fenómenos que se acentúan con el avance del tiempo podrían estar relacionados con la incomodidad creciente del sujeto.

En cuanto a electrooculograma, se observa menos movimiento en los momentos iniciales (Figura 5-6) que en los finales (Figura 5-7).

Figura 5-6. Electrooculograma en los primeros compases del experimento.

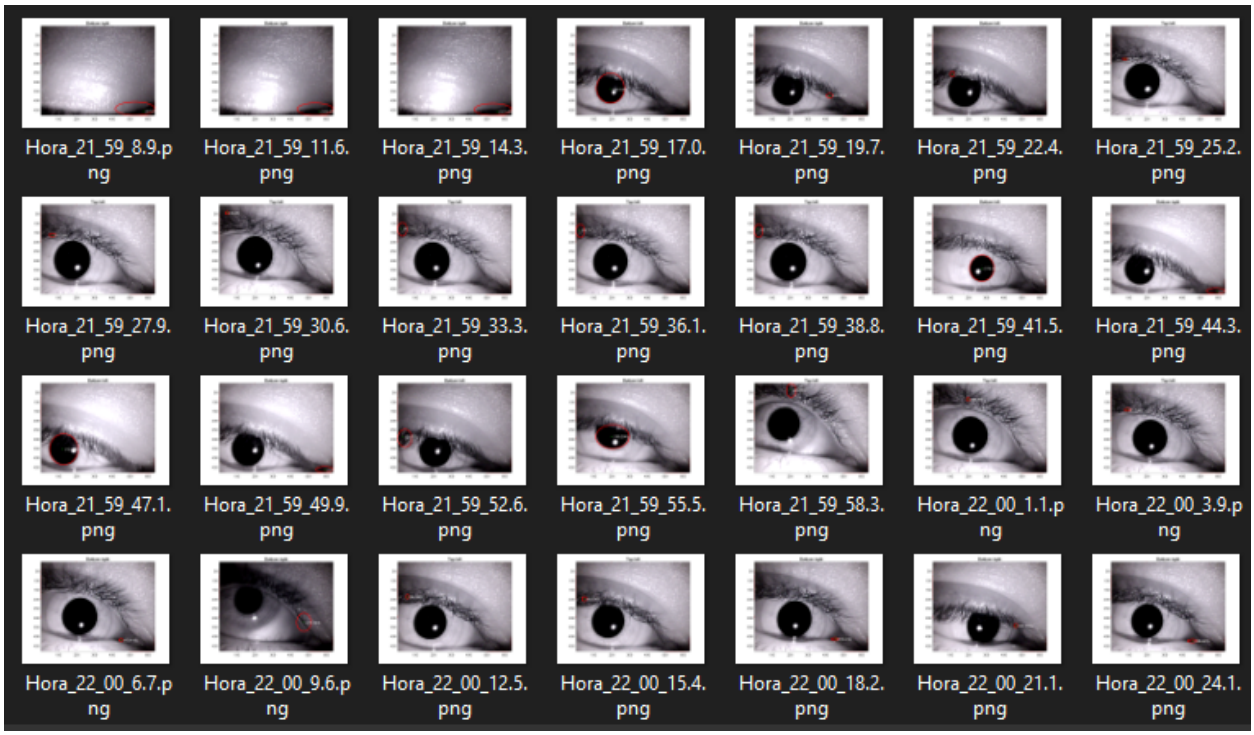


Figura 5-7. Electrooculograma en los últimos compases del experimento.

Este suceso denota una mayor distracción del sujeto experimental en los últimos compases del experimento respecto a los primeros, lo que concuerda con la incomodidad antes mencionada en vista a las otras señales fisiológicas.

6 CONCLUSIONES

Finalmente, se han conseguido cumplir los objetivos del proyecto:

- En primer lugar, se ha conseguido el funcionamiento de cada uno de los sensores, adquiriendo los datos de las señales fisiológicas de forma adecuada. Además, se ha encontrado la forma de sincronizar la adquisición de todos los datos, consiguiendo así una mayor fiabilidad a la hora de hacer experimentos.
- En cuanto a la colocación de los sensores, es importante reflejar que el sistema completo puede llegar a ser invasivo para el sujeto experimental. Por tanto, una de las posibles mejoras para futuros usos o experimentos del sistema, podría ser reducir el tamaño o la invasividad de alguno de los sensores, especialmente el electrooculograma, ya que así se reduciría la influencia del propio sistema en los resultados del experimento.
- Por último, se ha conseguido un procesamiento de los datos que permite comparar las medidas de los sensores de forma sencilla, gracias al sincronismo comentado anteriormente, y al procesamiento de las lecturas de los sensores en gráficas de Matlab.
- Finalmente se decidió no incluir el sensor de electroencefalograma al sistema. Esto se debe a dos motivos. En primer lugar, el sensor que teníamos disponible no aporta realmente la señal EEG, sino una aproximación de la misma. Esto se debe a que un sensor de encefalograma normal cuenta con diez electrodos o más, mientras que el nuestro solo tiene tres. Además, la obtención de los datos requiere un método muy diferente al de los otros sensores, lo que dificulta su integración en el sistema de manera sincronizada.

El proyecto ha estado lleno de dificultades, desde algunos problemas en la organización inicial , pasando por la complicada situación que vivimos con la COVID-19, la necesidad de adquirir materiales por internet en un tiempo reducido… Por suerte, se ha podido completar a tiempo, y en un futuro podría servir a otras personas para futuras líneas de investigación:

- Un posible experimento sería la medida de las señales fisiológicas de un cantante mientras interpreta una pieza musical. Se podría así, seguir investigando en el campo de la búsqueda de emociones relacionadas con la música gracias a este sistema.
- También se podría utilizar en otros campos, por ejemplo en un reconocimiento médico, o para otro tipo de investigaciones sobre emoción que no tengan que estar necesariamente relacionadas con la música.

7 ANEXOS

7.1 Matriz de Verificación de requisitos técnicos

Req.	Verificación				Nombre de la Prueba	Estado
	I	A	D	T		
F.1			x		Medida de datos por cada sensor	ok
F.2			x		Recogida de datos en tiempo real	ok
F.3			x		Muestra de datos recogidos	ok
F.4			x		Medida de pulso	ok
F.5			x		Toma de imagen ocular	ok
P.1.1			x		Recogida de medidas síncronas de todos los sensores	ok
P.3.1		x			Muestra de gráficas	ok
P.4.1			x		Medida de pulso	ok
P.5.1			x		Toma de imagen ocular	ok
D.1	x				Observación del sistema	ok
D.2	x				Observación del sistema	ok
D.3		x			Medida de peso	ok
O.1		x			Control del sistema	ok
R.1	x				Comprobación de sensor de pulso adecuado	ok
R.2	x				Comprobación de sensor de GSR adecuado	ok
R.3	x				Comprobación de sensor de EMG adecuado	ok
R.4	x				Comprobación de sensor de movimiento ocular adecuado	ok
S.1		x			Comprobación de normativa	pendiente
S.2		x			Comprobación de materiales	ok
S.3		x			Comprobación de materiales	ok
S.4		x			Comprobación de conexionado	ok

Tabla 7-1. Matriz de Pruebas.

7.2 Presupuesto

COMPONENTE	PROVEEDOR	PRECIO (IVA Y ENVÍO INCLUIDOS)
Placa Arduino Uno	Store.arduino.cc	30.86€
Cables macho-hembra, macho-macho y hembra-hembra para Arduino y Raspberry pi, 3x40 piezas, 20cm	Amazon.es	5.49€
Pulse Sensor	Pulsesensor.com	33.72€
GSR Sensor	Seeedstudio.com	8.36€ (+ envío)
MyoWare Muscle Sensor	Sparkfun.com	64.03€
Electrodos para sensor muscular (12 unidades)	Amazon.es	13.49€
Hub USB de 4 puertos	Amazon.es	9.89€
Cámara Panasonic Lumix DMC-LX7	eBay	135€
Mano de obra del Ingeniero (20€/hora)		300x20=6000€
TOTAL		6300.84€

Tabla 7-2. Presupuesto del proyecto.

7.3 Hojas Características de los sensores

7.3.1 Pulse Sensor

WORLD FAMOUS ELECTRONICS llc.

www.pulsesensor.com

PULSE SENSOR
EASY TO USE HEART RATE SENSOR & KIT

General Description	Features
The Pulse Sensor is the original low-cost optical heart rate sensor (PPG) for Arduino and other microcontrollers. It's designed and made by World Famous Electronics, who actively maintain extensive example projects and code at: www.pulsesensor.com	• Includes Kit accessories for high-quality sensor readings • Designed for Plug and Play • Small size and embeddable into wearables • Works with any MCU with an ADC • Works with 3 Volts or 5 Volts • Well-documented Arduino library

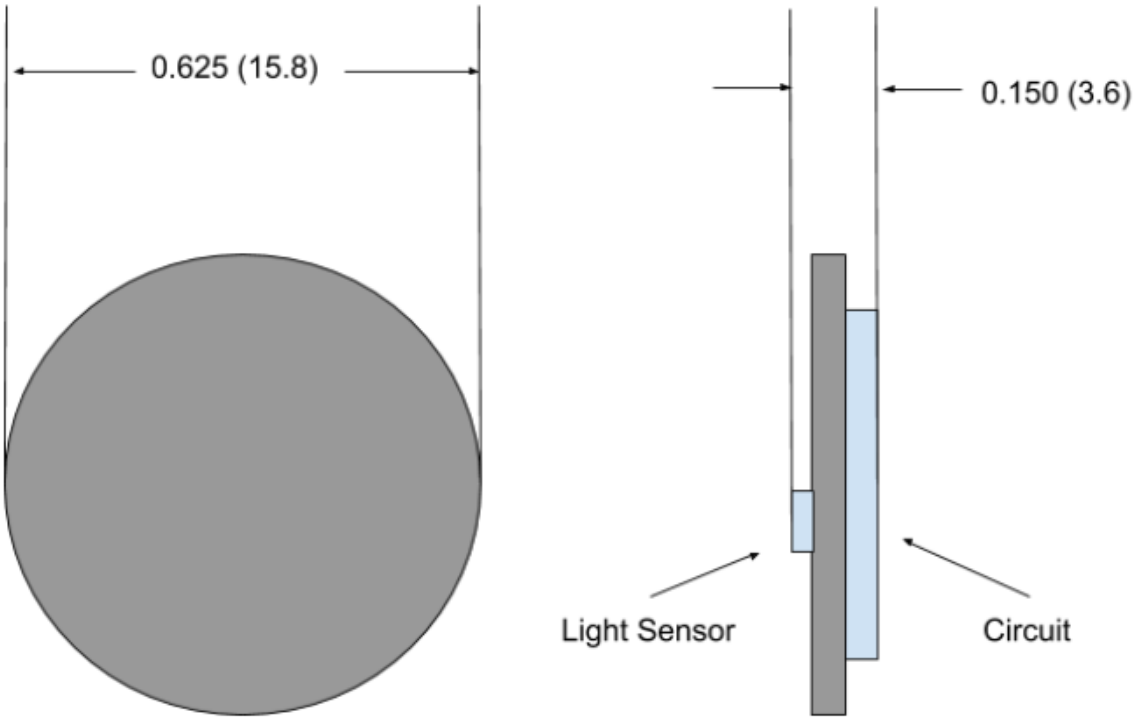
Absolute Maximum Ratings	Min	Typ	Max	Unit
Operating Temperature Range	-40		+85	°C
Input Voltage Range	3		5.5	V
Output Voltage Range	0.3	Vdd/2	Vdd	V
Supply Current	3		4	mA

Pulse Sensor Kit Contents

Pulse Sensor Optical Heart Rate Monitor

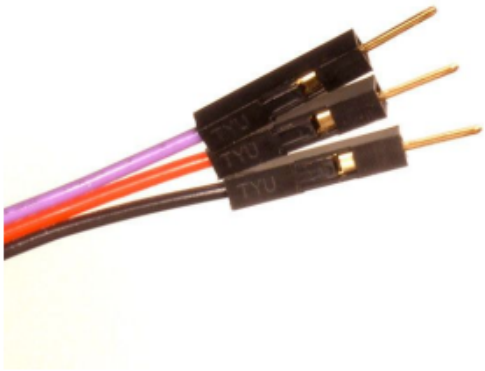
1/2

Physical Dimensions PCB inch(mm)



Cable Specs

- **Length 610 mm (24 inches)**
- **26 Gauge**
- **PVC Insulation, Ribbon Style**
- **Male Header Termination**
  - **Black Wire = GND**
  - **Red Wire = Vdd**
  - **Purple Wire = Pulse Signal**



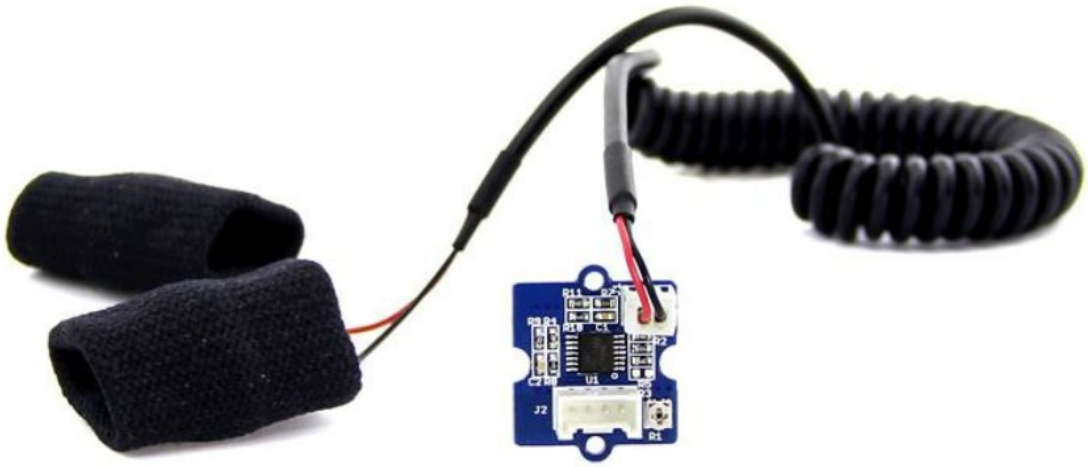
Pulse Sensor is certified as Open Source with the Open Source Hardware Association

OSHW US000075

2/2

7.3.2 GSR Sensor

Grove - GSR Sensor



GSR stands 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. It is an interesting to create emotion related projects like sleep quality monitor.

!!!Warning
Grove-GSR Sensor measures the resistance of the people, NOT Conductivity!

Version

Product Version	Changes	Released Date
Grove - GSR_Sensor V1.0	Initial	June 19, 2013
Grove - GSR_Sensor V1.2	Add C3 100nf between M324PW-TSSOP14 and GND	July 31, 2014

Specification

Parameter	Value/Range
Operating voltage	3.3V/5V
Sensitivity	Adjustable via a potentiometer
Input Signal	Resistance, NOT Conductivity
Output Signal	Voltage, analog reading
Finger contact material	Nickel

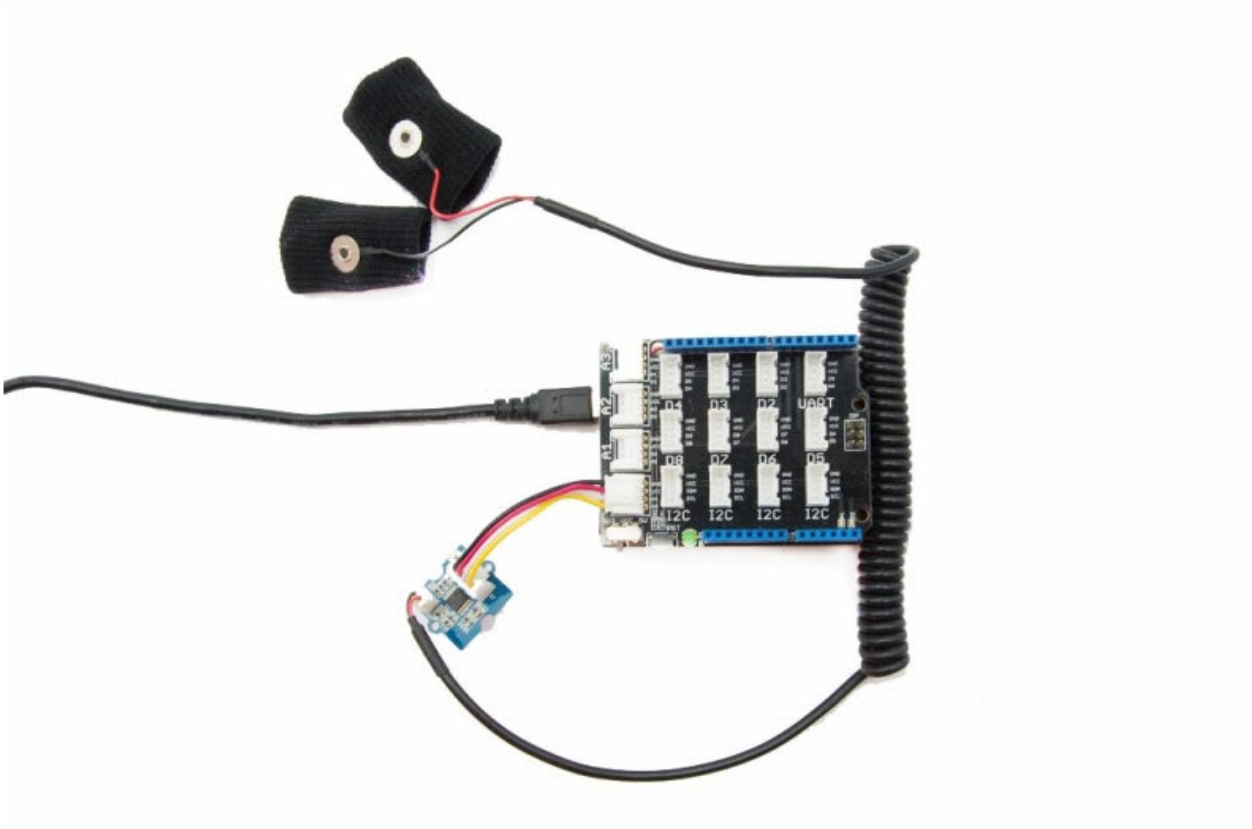
Getting Started

Hardware

- Step 1. We need to prepare the below stuffs:

Seeeduino V4.2	Base Shield	Grove - GSR
		
Get ONE Now	Get ONE Now	Get ONE Now

- Step 2. Connect the Grove-GSR to **A0** on Base Shield.
- Step 3. Plug the base Shield into Seeeduino-V4.2.
- Step 4. Connect Seeeduino-V4.2 to PC by using a USB cable.



GND	Black
5V	Red
NC	White
A0	Yellow

Software

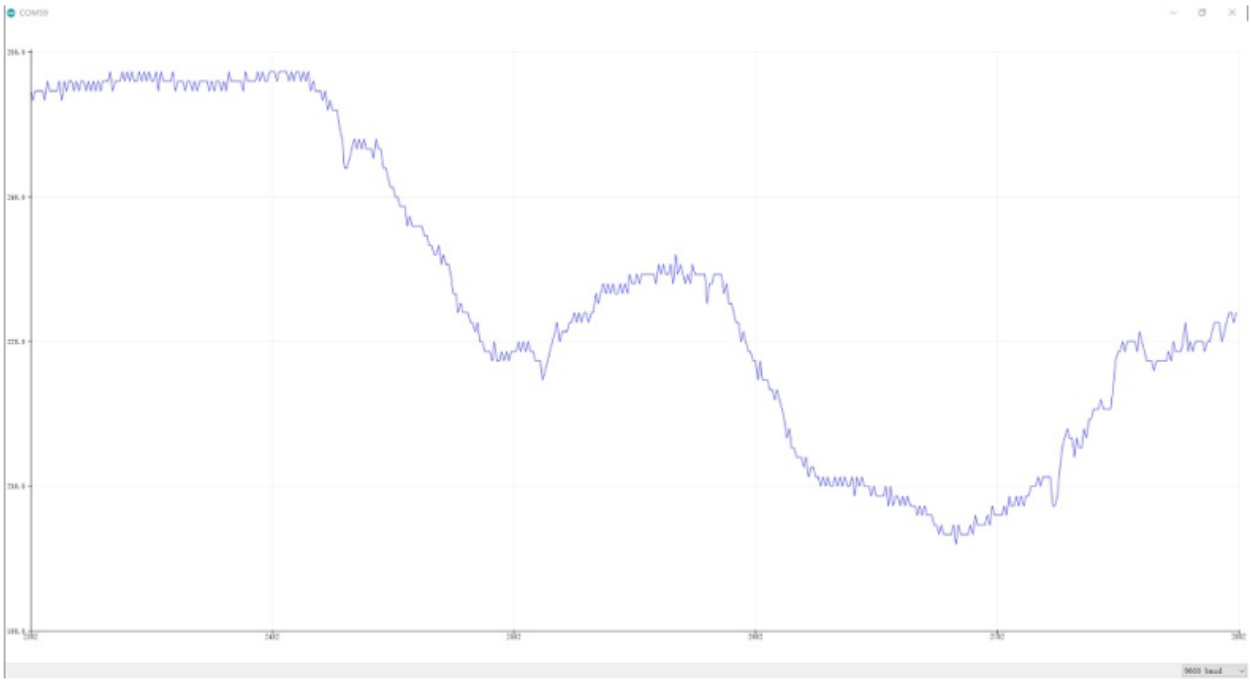
- Step 1. Copy the code into Arduino IDE and upload.

```
const int GSR=A0;
int sensorValue=0;
int gsr_average=0;

void setup(){
  Serial.begin(9600);
}

void loop(){
  long sum=0;
  for(int i=0;i<10;i++)           //Average the 10 measurements to remove the glitch
      {
      sensorValue=analogRead(GSR);
      sum += sensorValue;
      delay(5);
      }
   gsr_average = sum/10;
   Serial.println(gsr_average);
}
```

- Step 2. Wear the GSR sensor
- Step 3. Click the Tools-> Serial Plotter from Arduino IDE
- Step 4. We will see the below graph. Please deep breath and see the trends.



Human Resistance = (1024+2*Serial_Port_Reading)*10000/(512-Serial_Port_Reading), unit is ohm, Serial_Port_Reading is the value display on Serial Port(between 0~1023)

7.3.3 MyoWare Muscle Sensor

3-lead Muscle / Electromyography Sensor for Microcontroller Applications

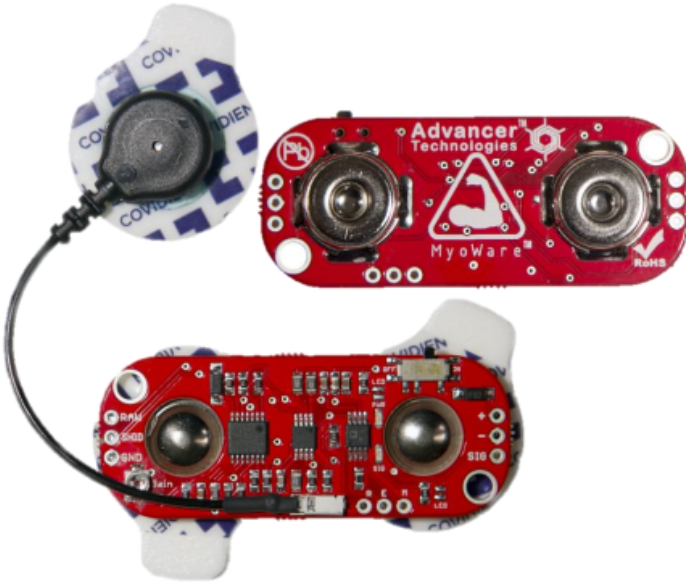
MyoWare™ Muscle Sensor (AT-04-001) DATASHEET

FEATURES

- *NEW* - Wearable Design
- *NEW* - Single Supply
  - +3.1V to +5.9V
  - Polarity reversal protection
- *NEW* - Two Output Modes
  - EMG Envelope
  - Raw EMG
- *NEW* - Expandable via Shields
- *NEW* - LED Indicators
- Specially Designed For Microcontrollers
- Adjustable Gain

APPLICATIONS

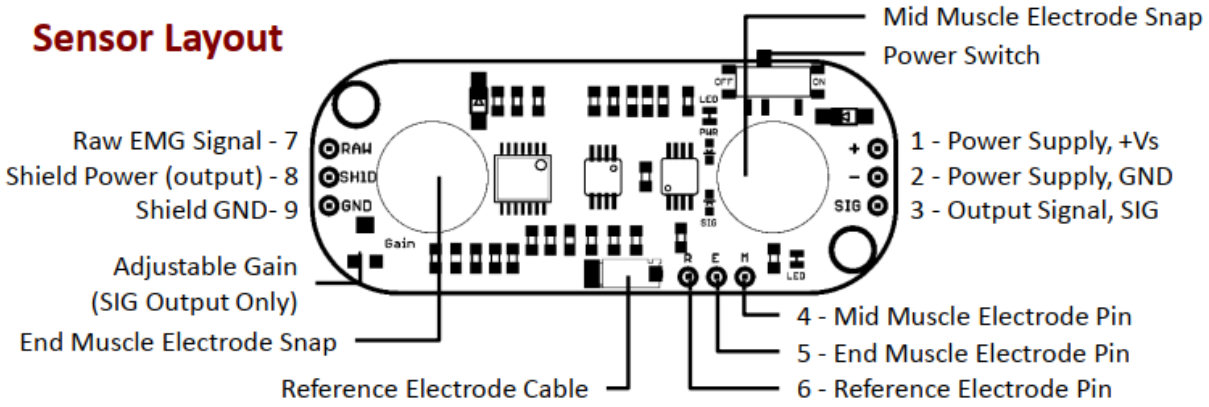
- Video games
- Robotics
- Medical Devices
- Wearable/Mobile Electronics
- Prosthetics/Orthotics



What is electromyography?

Measuring muscle activation via electric potential, referred to as electromyography (EMG), has traditionally been used for medical research and diagnosis of neuromuscular disorders. However, with the advent of ever shrinking yet more powerful microcontrollers and integrated circuits, EMG circuits and sensors have found their way into prosthetics, robotics and other control systems.

Sensor Layout



EMAIL: support@advancer.co

Advancer™ Technologies

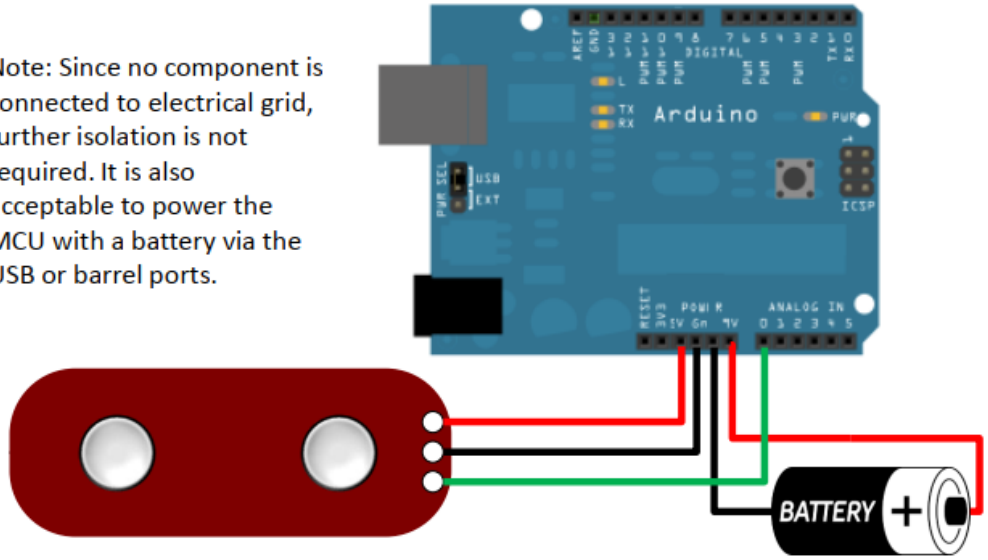
www.AdvancerTechnologies.com

Setup Configurations *(Arduino is shown but MyoWare is compatible with most development boards)*

a) Battery powered with isolation via no direct external connections

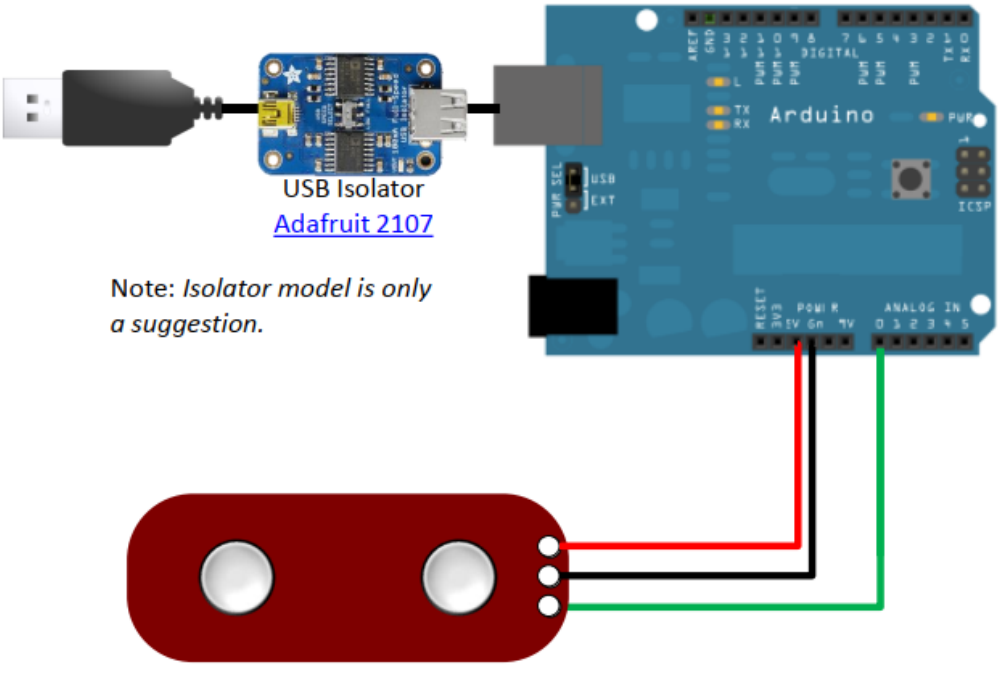
RECOMMENDED

Note: Since no component is connected to electrical grid, further isolation is not required. It is also acceptable to power the MCU with a battery via the USB or barrel ports.



b) Grid powered with USB isolation

RECOMMENDED



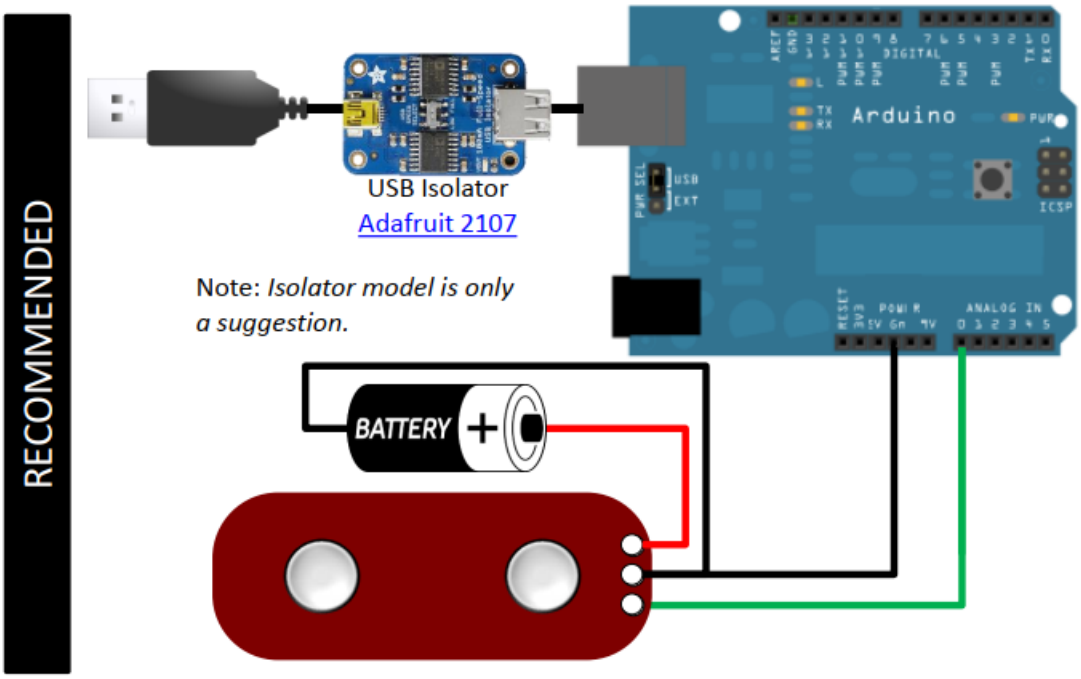
USB Isolator
Adafruit 2107

Note: *Isolator model is only a suggestion.*

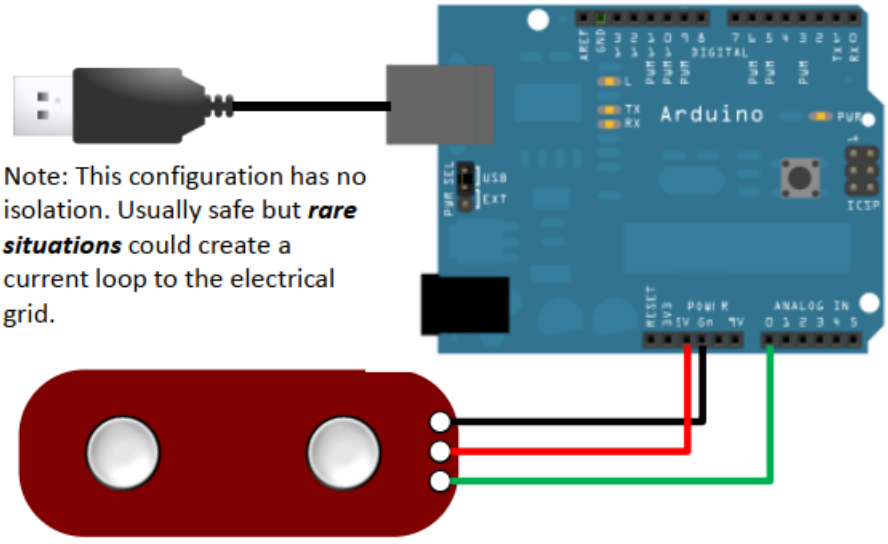
(Note: Arduino and batteries not included. Arduino setup is only an example; sensor will work with numerous other devices.)

Setup Configurations (cont'd)

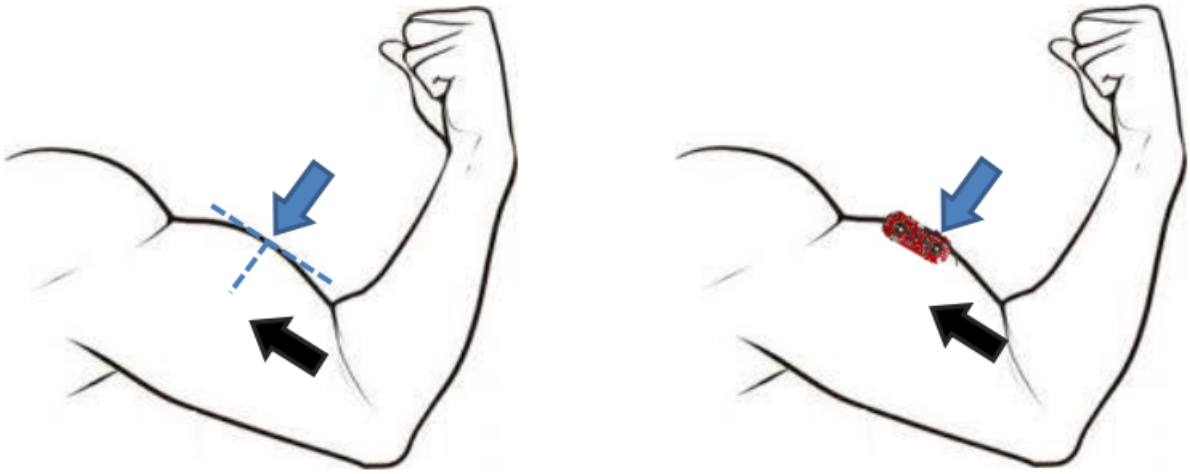
c) Battery powered sensor, Grid powered MCU with USB isolation



d) Grid powered. ***Warning: No isolation.***



Setup Instructions



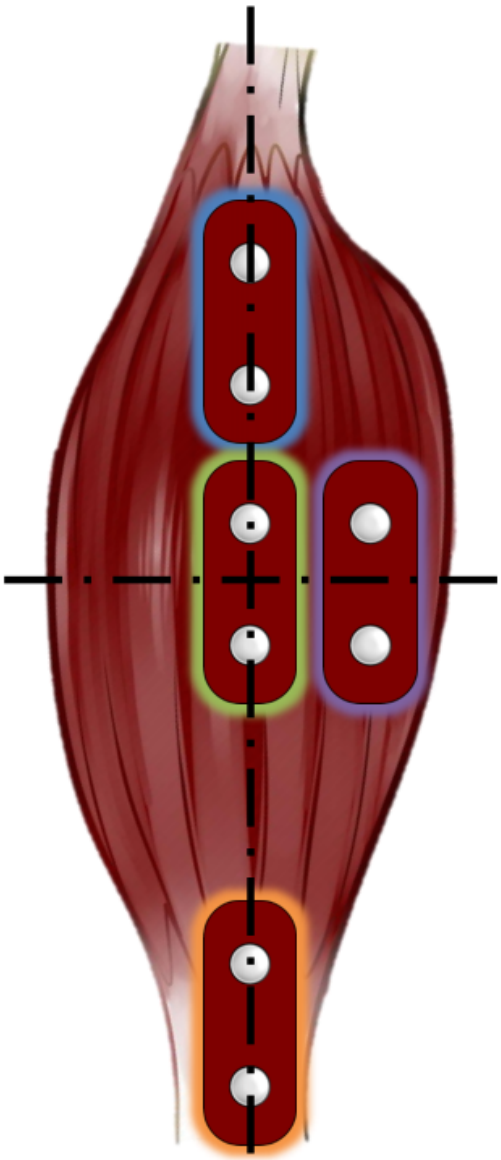
Note: Not To Scale

Example Sensor Location for Bicep

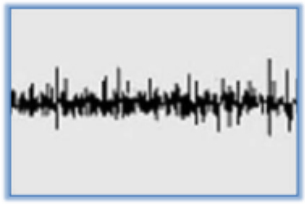
1) Thoroughly clean the intended area with soap to remove dirt and oil
2) Snap electrodes to the sensor's snap connectors

   (Note: While you can snap the sensor to the electrodes after they've been placed on the muscle, we do not recommend doing so due to the possibility of excessive force being applied and bruising the skin.)
3) Place the sensor on the desired muscle
   a. After determining which muscle group you want to target (e.g. bicep, forearm, calf), clean the skin thoroughly
   b. Place the sensor so one of the connected electrodes is in the middle of the muscle body. The other electrode should line up in the direction of the muscle length
   c. Peel off the backs of the electrodes to expose the adhesive and apply them to the skin
   d. Place the reference electrode on a bony or nonadjacent muscular part of your body near the targeted muscle
4) Connect to a development board (e.g. Arduino, RaspberryPi), microcontroller, or ADC
   a. See configurations previously shown

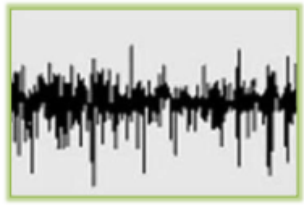
Why is electrode placement important?



Raw EMG output



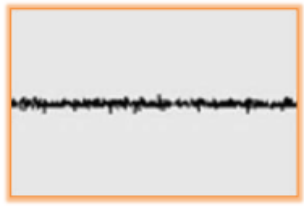
Innervation Zone



Correct Placement

Midline of the muscle belly between an innervation zone and a myotendon junction

Midline Offset

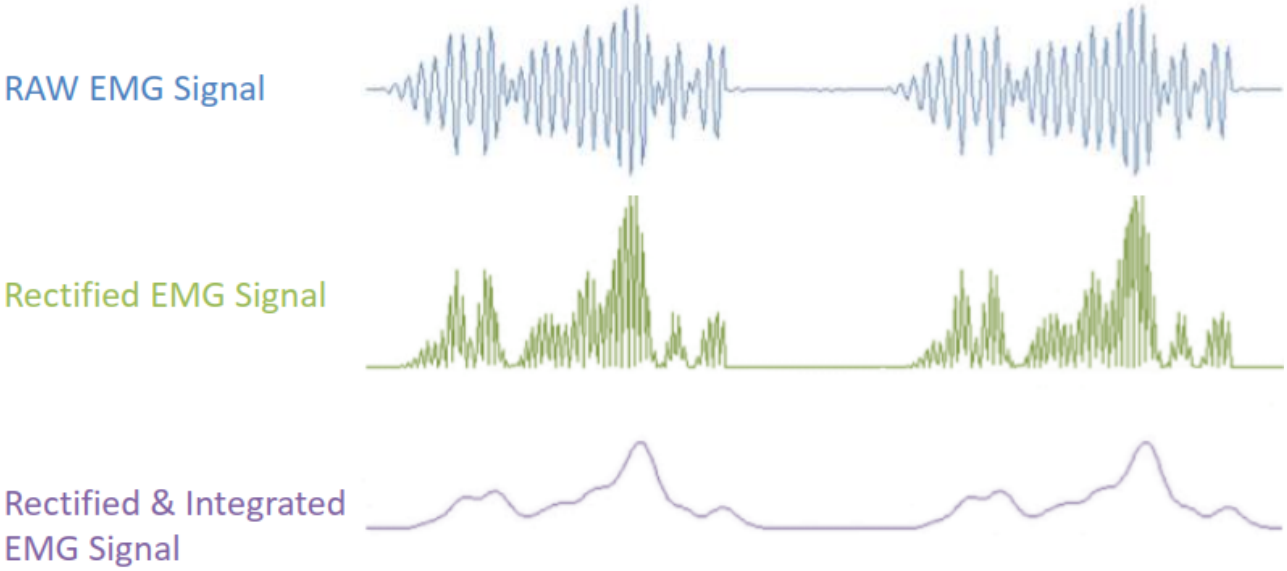


Myotendon Junction

Position and orientation of the muscle sensor electrodes has a vast effect on the strength of the signal. The electrodes should be place in the middle of the muscle body and should be aligned with the orientation of the muscle fibers. Placing the sensor in other locations will reduce the strength and quality of the sensor's signal due to a reduction of the number of motor units measured and interference attributed to crosstalk.

RAW EMG vs EMG Envelope

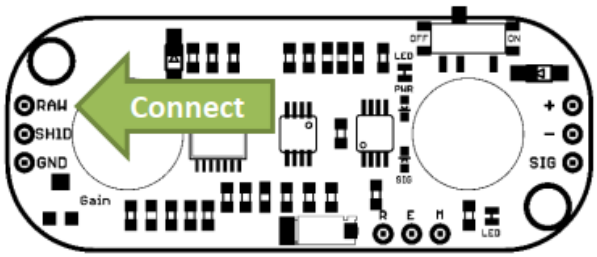
Our Muscle Sensors are designed to be used directly with a microcontroller. Therefore, our sensors primary output is not a RAW EMG signal but rather an amplified, rectified, and integrated signal (AKA the EMG's envelope) that will work well with a microcontroller's analog-to-digital converter (ADC). This difference is illustrated below using a representative EMG signal. *Note: Actual sensor output not shown.*



Reconfigure for Raw EMG Output

This new version has the ability to output an amplified raw EMG signal.
To output the raw EMG signal, simply connect the raw EMG signal pin to your measuring device instead of the SIG pin.

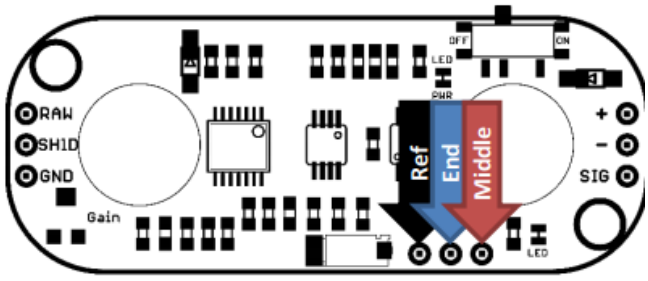
***Note**: The RAW output is centered about an offset voltage of +Vs/2, see above. It is important to ensure +Vs is the max voltage of the MCU's analog to digital converter. This will assure that you completely see both positive and negative portions of the waveform.*

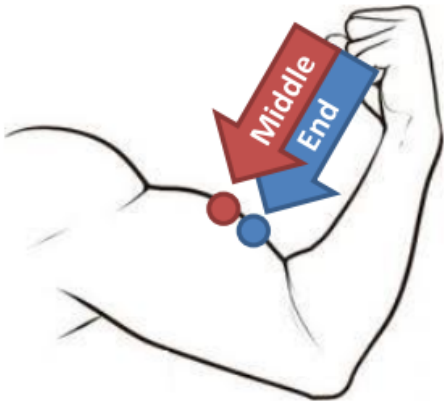


***Note**: The amplification for the RAW output **is <u>not</u> adjustable** via the GAIN potentiometer.*

Connecting external electrode cables

This new version has embedded electrode snaps right on the sensor board itself, replacing the need for a cable. However, if the on board snaps do not fit a user's specific application, an external cable can be connected to the board through three through hole pads shown above.





Middle

Connect this pad to the cable leading to an electrode placed in the middle of the muscle body.

End

Connect this to the cable leading to an electrode placed adjacent to the middle electrode towards the end of the muscle body.

Ref

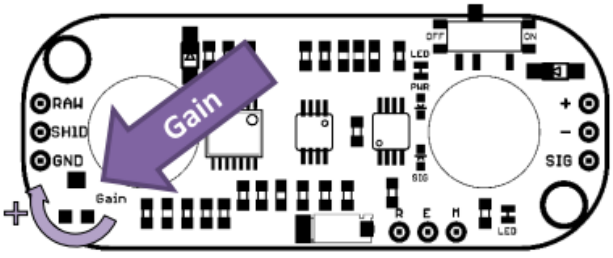
Connect this to the reference electrode. The reference electrode should be placed on an separate section of the body, such as the bony portion of the elbow or a nonadjacent muscle

Adjusting the gain

We recommend for users to get their sensor setup working reliably prior to adjusting the gain. The default gain setting should be appropriate for most applications.

To adjust the gain, locate the gain potentiometer in the lower left corner of the sensor (marked as "GAIN"). Using a Phillips screwdriver, turn the potentiometer clockwise to increase the output gain; turn the potentiometer counterclockwise to reduce the gain.

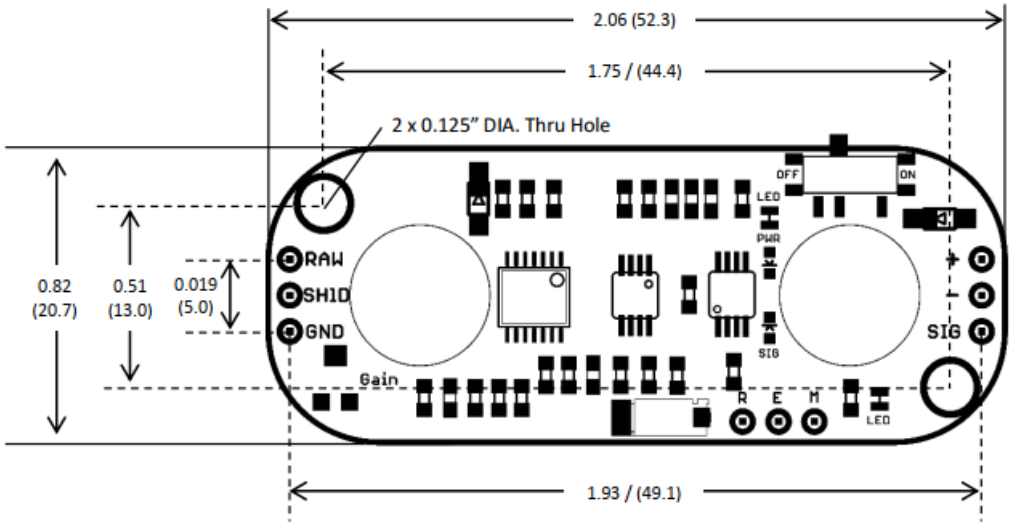
Note: In order to reduce the required voltage for the sensor, the redesign switch out a JFET amplifier for a CMOS amplifier. However CMOS amplifiers tend to have slower recovery times when saturated. Therefore, we advise users to adjust the gain such that the output signal will not saturate the amplifier.



Electrical Specifications

Parameter	Min	TYP	Max
Supply Voltage	+3.1V	+3.3V or +5V	+6.3V
Adjustable Gain Potentiometer, R_{gain} (G = 201 * R_{gain} / 1 kΩ)	0.01 Ω	50 kΩ	100 kΩ
Output Signal Voltage EMG Envelope Raw EMG (centered about +Vs/2)	 0V 0V	 -- --	 +Vs +Vs
Input Impedance	--	110 GΩ	--
Supply Current	--	9 mA	14 mA
Common Mode Rejection Ratio (CMRR)	--	110	--
Input Bias	--	1 pA	--

Dimensions



7.4 Códigos

7.4.1 Arduino Uno

Por una parte tenemos el proyecto de Arduino Uno, formado por un fichero principal “Sensores.ino”, que a su vez llama a las funciones de cada uno de los sensores conectados a esta placa.

a) Sensores.ino:

```
#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>

const int OUTPUT_TYPE = SERIAL_PLOTTER;

/*
   Pinout:
     PULSE_INPUT = Analog Input. Connected to the pulse sensor
      purple (signal) wire.
     PULSE_BLINK = digital Output. Connected to an LED (and 220 ohm
resistor)
      that will flash on each detected pulse.
     PULSE_FADE = digital Output. PWM pin connected to an LED (and
resistor)
      that will smoothly fade with each pulse.
      NOTE: PULSE_FADE must be a pin that supports PWM. Do not use
      pin 9 or 10, because those pins' PWM interferes with the sample
timer.
*/

/*
************************************************
* PINES DE PULSE SENSOR   ** PINES DE GSR        *
*   MARRON S A0           **  GND BLACK/GND      *
*   ROJO + VCC            **  5V RED/VCC         *
*   NARANJA - GND         **  A2 YELLOW/SIG      *
*                         **  NC WHITE/RED LED   *
* ************************************************
*/
```

```
/*
 GSR Sensor mano derecha
  Dedo corazón -> cable rojo
  Dedo índice -> cable negro
*/
/* PulseSensor mano izquierda -> dedo índice */
/************************/
const int PIN_INPUT = A0;   // entrada pulse sensor
const int PIN_BLINK = 13;   // LED1
//const int PIN_FADE = 5;     // LED2
const int THRESHOLD = 550;  // Adjust this number to avoid noise when
idle
/************************/
const int GSR=A2;           // entrada GSR sensor
int sensorValue=0;
int gsr_average=0;
/************************/
const int EMG=A4;           // entrada EMG sensor
int sensEMG=0;
float emg_average=0;

/*
  All the PulseSensor Playground functions.
*/
PulseSensorPlayground pulseSensor;

void setup(){
  Serial.begin(128000);     //velocidad en baudios serie para excel

  //líneas necesarias para imprimir los datos en una hoja excel
  Serial.println("CLEARDATA");
  Serial.println("LABEL,Hora,Tiempo,GSR,   EMG,   BPM,   tiempo   entre
latidos, Pulso"); //etiquetas para las columnas de excel
  Serial.println("RESETTIMER");

  pulseSensor.analogInput(PIN_INPUT);
  pulseSensor.setSerial(Serial);
  pulseSensor.setOutputType(OUTPUT_TYPE);
  pulseSensor.setThreshold(THRESHOLD);
```

```
  if (!pulseSensor.begin()) { //en el caso de que el PulseSensor no se
inicie
    /*
      PulseSensor initialization failed,
      likely because our particular Arduino platform interrupts
      aren't supported yet.
      If your Sketch hangs here, try ProcessEverySample.ino,
      which doesn't use interrupts.
    */
    for(;;) {
      // Flash the led to show things didn't work.
      digitalWrite(PIN_BLINK, LOW);
      delay(100);
      digitalWrite(PIN_BLINK, HIGH);
      delay(100);
    }
  }
}


void loop(){
  Serial.print("DATA, TIME, TIMER, "); //imprime en excel la hora y un
temporizador


  //llamamos previamente a la funcion gsr con las instrucciones del
GSR Sensor e imprimimos el valor:
  Serial.print(gsr());
  Serial.print(",");


  //llamamos previamente a la funcion emg con las instrucciones del
sensor EMG e imprimimos el valor:
  Serial.print(emg());
  Serial.print(",");



  //llamamos previamente a la funcion pulse con las instrucciones del
PulseSensor e imprimimos los valores indicados
  Serial.print(pulse());
  Serial.println("");
}
```

b) GSR.ino

```
int gsr(){ //función para GSR Sensor
  long sum=0;
  for(int i=0;i<10;i++) //hacemos el promedio entre 10 muestras para eliminar posible fallo
  {
    sensorValue=analogRead(GSR);
    sum += sensorValue;
    delay(15);
  }
  gsr_average = sum/100;
  Serial.print(gsr_average); //dato a mostrar por pantalla
}
```

c) EMG.ino

```
int emg(){ //función para EMG
  float sum=0;
  for(int i=0;i<10;i++) //hacemos el promedio entre 10 muestras para eliminar posible fallo
  {
    sensEMG=analogRead(EMG);

    // Convert the analog reading (which goes from 0 - 10230) to a voltage (0 - 5V):
    float voltage = sensEMG * (5.0 / 10230.0);
    sum += voltage;
    delay(15);
  }
  emg_average = sum;
  Serial.print(emg_average); //dato a mostrar por pantalla
}
```

d) Pulse.ino

```
int pulse(){ //función para PulseSensor
  /*
    * Introducimos un cierto retraso entre muestra y muestra a enviar
    * debido a que la velocidad de transmisión no admite tantas E/S
  */
  delay(20);
  pulseSensor.outputSample();
  pulseSensor.outputBeat();
  /*
    * Si se ha tenido un latido desde la última vez que se comprobó,
    * se escribirá la información por latido en serie.
  */
  if (pulseSensor.sawStartOfBeat()) {
    pulseSensor.outputBeat();
  }
}
```

7.4.2 Arduino Pro Mini

Este código de Adán [17] permite el control de los leds infrarrojos del electrooculograma (Arduino_Matlab_Serial.ino).

```
#define LED_PIN 13
bool blinkState = false;

int ledPin2 = 2;



int matlabData;

void setup()
{
  // configure LED for output
  pinMode(ledPin2,OUTPUT);
  pinMode(LED_PIN, OUTPUT);

  Serial.begin(9600);
}

void loop()
{

   if(Serial.available()>0) // if there is data to read
   {
    matlabData=Serial.read(); // read data

     if (matlabData==3)
     //blinkState = !blinkState;
     digitalWrite(ledPin2,HIGH); // turn light on
  }

    blinkState = !blinkState;
    //digitalWrite(LED_PIN, blinkState);
    digitalWrite(LED_PIN,blinkState); // turn light on
}
```

7.4.3 Matlab Capturas Electrooculograma

```
clc
clear all

if ~isempty(instrfind)
   fclose(instrfind);
   delete(instrfind);
end

% create an arduino object. instead of "/dev/cu.usbmodem1411','BaudRate"
% use your serail port name. You can check it in arduino -> tools -> port.
% Write the name of your port before the command 'BaudRate' in the
% following string.
%ar = arduino('/dev/tty.usbmodem14101');
ar = serial('COM6','BaudRate',9600);
% Check for the camera name with the command
% webcamlist() in matlab.
cameras= webcamlist()
% type "cameras" in the command window and see which camera correspond to
your modified webcam. select that camera
% just by entering the index within the curly brackets.
% It could be 1, 2, 3... depends on how many webcams you have attached on
your computer
mycamera = cameras{1}
% create a 'cam' variable with your camera name
cam = webcam(char(mycamera))

%open arduino USB port
fopen(ar);

% send the pulse to the port number 2 of arduino
fprintf(ar,'%s',char(3));

pause(5);

%check if the camera is positioned in the right way, with the focus on the
%center of the eye. The "preview" command let you see what the camera see.

preview(cam)


% preallocate the matrix with the cooridnates data
coordinate = zeros(50,2);

for i = 1:500;

% take a snapshot
img = snapshot(cam);

% calculate the center of the image

filas=size(img,1);
columnas=size(img,2);
% Center
centro_fila=round(filas/2);
centro_columna=round(columnas/2);
% transform the image in a BW image
if size(img,3)==3
```

```
    la_imagen=img;
end

 piel=~im2bw(la_imagen,0.1);
%     --
 piel=bwmorph(piel,'close');
 piel=bwmorph(piel,'open');
 piel=bwareaopen(piel,200);
 piel=imfill(piel,'holes');

% Tagged objects in BW image
L=bwlabel(piel);
% Get areas and tracking rectangle
out_a=regionprops(L);
% Count the number of objects
N=size(out_a,1);
while N < 1 || isempty(out_a) % Returns if no object in the image
    solo_cara=[ ];
    continue
end

% Select the area
areas=[out_a.Area];
[area_min pam_min]=min(areas);
[area_max pam_max]=max(areas);

% since there is a problem with the shades in the BW images (the algorithm
detect the size of the black
% shades in the image), we have to create an if statement where we declare
that if the algorithm
% detect a too big black area (threshold set on 10000), it has to consider
the
% smallest detected area as the pupil.
% On the other hand, if the black area is below a threshold of
% 10000 it considers the biggest black area as the pupil.

if area_max > 20000;

% show the BW image
imagesc(la_imagen);
colormap gray
hold on
% draw the red area around the pupil
rectangle('Position',out_a(pam_min).BoundingBox,'EdgeColor',[1 0 0],...
    'Curvature', [1,1],'LineWidth',2)
centro=round(out_a(pam_min).Centroid);

% detect the X and Y coordinates
X=centro(1);
Y=centro(2);

% save X and Y coordinates in the coordinates matrix
coordinate(i,1) = X;
coordinate(i,2) = Y;

% draw the cross at the center of the pupil
plot(X,Y,'g+')
%
text(X+10,Y,['(',num2str(X),',',num2str(Y),')'],'Color',[1 1 1])
if X<centro_columna && Y<centro_fila
```

```
    title('Top left')
elseif X>centro_columna && Y<centro_fila
    title('Top right')
elseif X<centro_columna && Y>centro_fila
    title('Bottom left')
else
    title('Bottom right')
end

elseif area_max < 20000;

% show the BW image
imagesc(la_imagen);
colormap gray
hold on
% draw the red area around the pupil
rectangle('Position',out_a(pam_max).BoundingBox,'EdgeColor',[1 0 0],...
    'Curvature', [1,1],'LineWidth',2)
centro=round(out_a(pam_max).Centroid);

% detect the X and Y coordinates
X=centro(1);
Y=centro(2);

% save X and Y coordinates in the coordinates matrix
coordinate(i,1) = X;
coordinate(i,2) = Y;

% draw the cross at the center of the pupil
plot(X,Y,'g+')
%
text(X+10,Y,['(',num2str(X),',',num2str(Y),')'],'Color',[1 1 1])
if X<centro_columna && Y<centro_fila
    title('Top left')
elseif X>centro_columna && Y<centro_fila
    title('Top right')
elseif X<centro_columna && Y>centro_fila
    title('Bottom left')
else
    title('Bottom right')
end

end
    moment=clock;
%% save current figure in a folder. IMPORTANT !!! Set your directory in the
following command
saveas(gcf,['/Users/luigy/Desktop/Img_Cam/Hora_', sprintf('%02d_',
moment(4)), sprintf('%02d_', moment(5)), sprintf('%.1f', moment(6)),'.png']);

end

%close arduino port number 2
fprintf(ar,'%s',char(4));
% close arduino USB port
fclose(ar);
```

7.4.4 Matlab Señales Fisiológicas

Este código permite, una vez importados los datos de Excel, representar las señales en función del tiempo.

```
datos=DatosExperimento;

hora=table2array(datos(1:height(datos),1));
tiempo=table2array(datos(1:height(datos),2));
GSR=table2array(datos(1:height(datos),3));
EMG=table2array(datos(1:height(datos),4));
BPM=table2array(datos(1:height(datos),5));


subplot(4,1,1)
plot(tiempo, GSR)
title('GSR en función del tiempo transcurrido')
xlabel('t')
ylabel('GSR')

subplot(4,1,2)
plot(tiempo, EMG)
title('EMG en función del tiempo transcurrido')
xlabel('t')
ylabel('EMG')

subplot(4,1,3)
plot(tiempo, BPM)
title('BPM en función del tiempo transcurrido')
xlabel('t')
ylabel('BPM')

subplot(4,1,4)
plot(tiempo, hora)
set(gca,'yticklabel',[])
title('Hora en función del tiempo transcurrido')
xlabel('t')
ylabel('Hora')
```

REFERENCIAS

[1] Wikipedia, «Music Information Retrieval,» 2010. [En línea]. Available:

https://en.wikipedia.org/wiki/Music_information_retrieval

[2] León, Pedro & Iñesta, Jose. «Pattern Recognition Approach for Music Style Identification Using Shallow Statistical Descriptors, » Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 37. 248 - 257. 10.1109/TSMCC.2006.876045, 2007.

[3] DROIT-VOLET SYLVIE, Ramos danilo, Bueno Lino, Bigand Emmanuel, «Music, emotion, and time perception: the influence of subjective emotional valence and arousal?, » *Frontiers in Psychology vol. 4*, 2013 Available online:

https://www.frontiersin.org/article/10.3389/fpsyg.2013.00417

[4] X. Hu, V. Sanghvi, B. Vong, P. J. On, C. Leong, and J. Angelica. "Moody: A web-based music mood classification and recommendation system", *Proc. of 9th International Conference on Music Information Retrieval*, Philadelphia, U.S. 2008

[5] Xiao Hu, Fanjie Li, & Jeremy T. D. Ng. (2018). On the Relationships between Music-induced Emotion and Physiological Signals. In Proceedings of the 19th International Society for Music Information Retrieval Conference (pp. 362–369). Paris, France: ISMIR. http://doi.org/10.5281/zenodo.1492425

[6] Castrillo Navarro, M. (2020). Integración y validación de un sistema biométrico para la detección de piloerección. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla.

[7] Tizón, Manuel. (2017). Música y emociones: parámetros que modulan la emoción percibida. Musicaenclave. 11.

[8] Arduino, "Arduino Uno rev. 3", Available Online:

https://store.arduino.cc/arduino-uno-rev3

[9] Joel Murphy, Yury Gitman, Brad Needham, "PulseSensor Playground", Available Online:

https://pulsesensor.com/pages/installing-our-playground-for-pulsesensor-arduino

[10] Pulse Sensor Order, "Pulse Sensor Amped", Available Online:

https://pulsesensor.com/products/pulse-sensor-amped

[11] «tobiipro,» [En línea]. Available: https://www.tobiipro.com/learn-and-support/learn/GSR-essentials/how-does-a-gsr-sensor-work/. [Último acceso: 6 Julio 2019].

[12] «Seeed Studio,» [En línea]. Available: http://wiki.seeedstudio.com/Grove-GSR_Sensor/. [Último acceso: 17 Julio 2019].

[13] GSR Sensor, "Grove GSR Sensor", Available Online:

https://wiki.seeedstudio.com/Grove-GSR_Sensor/

[14] Suárez Luque, J. (2019). Integración de sensor EMG para control de electroestimulación. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla.

[15] Myoware Muscle Sensor, "Arduino Example Code," Available Online:

https://cdn.sparkfun.com/assets/e/8/e/a/d/MyoWareSampleCode.zip

[16] Sparkfun, "Myoware Muscle Sensor" Available Online:

https://www.sparkfun.com/products/13723

[17] Montero Torres, A. (2019). Desarrollo de entorno de realidad virtual con seguimiento ocular a través de tecnología de infrarrojos para terapia EMDR. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla.

[18] Arduino, "Arduino Pro Mini", Available Online:

https://store.arduino.cc/arduino-pro-mini

[19] Panasonic, "Lumix DMC-LX7", Available Online:

https://www.panasonic.com/middleeast/en/support/product-archive/camera/dmc-lx7.html

[20] Amazon, " HUB USB 3.0 Black UFO with 4 USB ports. High speed USB connector to transfer data to your computer. Compatible with Windows and Mac USB ports. USB 3.0 metallic graphite.", Available Online:

https://www.amazon.co.uk/HUB-USB-3-0-connector-Compatible-Black/dp/B01M01MYZY

[21] Parallax, "PLX-DAQ", Available Online:

https://www.parallax.com/downloads/plx-daq

[22] Microsoft Store, "Widget Launcher", Available Online:

https://www.microsoft.com/en-us/p/widget-launcher/9wzdncrdqfbt?activetab=pivot:overviewtab

GLOSARIO

MIR: Music Information Recognition 1

GSR: Galvanic Skin Response 2

GND: Tierra del circuito (0V) 10

Vcc: Tensión de alimentación de 5V 10

TFG: Trabajo de Fin de Grado 11

EDA: Actividad Electrodérmica 13

EMG: Electromiografía 16

BPM: Beats Per Minute 28

EEG: Electroencefalograma 34