



**Universidad
Zaragoza**

Proyecto Fin de Carrera

e-Health Sensor Platform

Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas

Autor

Luis Antonio Martín Nuez

Director y Codirector

Roberto Casas Nebra
Álvaro Marco Marco

Ingeniería Industrial, EINA
Curso 2013-2014

e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas

Resumen

Este proyecto está enmarcado dentro de mis prácticas en la empresa Libelium Comunicaciones Distribuidas S.L. Su motivación surge del interés en la creación de un dispositivo sensorial de parámetros biométricos que le permita utilizar esta tecnología, actualmente demasiado cara, a todo el mundo.

Las redes sensoriales inalámbricas están formadas por una serie de nodos que miden parámetros de interés mediante sensores, y trabajan en conjunto para enviar esta información. La instalación de una red sensorial en el entorno de un paciente o persona con limitaciones, posibilita una atención y control constante, incluso en su propio domicilio.

Para este proyecto se decide trabajar sobre la plataforma Open Source Arduino. Siguiendo las líneas de ese dispositivo, el proyecto se apoyará en las bases definidas por los productos del movimiento Open Source: contenido libre, accesibilidad, bajo coste... Por lo tanto, se pretende integrar en un único dispositivo una serie de sensores biométricos, para posibilitar la monitorización de parámetros vitales. Este seguimiento tiene que ser en tiempo real, de manera que es conocido el estado del sujeto y sus posibles necesidades. El campo de aplicación incluye todo tipo de monitorización de pacientes de forma poco invasiva, y otros usos que dependan de mediciones biomédicas.

El primer paso consistirá en la búsqueda de las necesidades del usuario, investigación de sistemas ya existentes, y estudio de sensores en el mercado. Esto nos permitirá definir unos requisitos mínimos de la plataforma, con los que poder continuar trabajando. Una vez definidos los sensores se deben diseñar las diferentes etapas, el hardware formado por la electrónica de adaptación necesaria para los sensores elegidos. Con respecto al software, se implementará una librería con múltiples funciones y ejemplos, con el fin de facilitar su programación al usuario final.

El resultado final pretende ser un dispositivo real, que se pueda comercializar como un producto acabado y funcional, acercando la tecnología de la monitorización médica a todos los lugares del planeta a un precio asequible, y que sirva como base para un futuro producto utilizable como "mini hospital" o centro de diagnóstico médico de bajo coste en países en vías de desarrollo o subdesarrollados. Aplicable al campo de la telemedicina, se busca revolucionar esta tecnología, rompiendo con la privatización de este mercado y disminuyendo el precio del producto final en comparación con los prohibitivos precios de los dispositivos hospitalarios. Esto abrirá las puertas a un nuevo movimiento de tecnologías médicas libres.

Tabla de contenido

Índice de figuras	6
Índice de Tablas.....	12
Glosario	13
1 Introducción	14
1.1 Marco de trabajo.....	14
1.1.1 Libelium y sus productos	14
1.1.2 Las redes sensoriales	15
1.2 Justificación.....	17
1.2.1 Objeto.....	17
1.2.2 Motivación del proyecto.....	18
1.3 Objetivos	19
1.3.1 Alcance.....	20
1.4 Antecedentes	21
1.5 Desarrollo del proyecto	23
1.5.1 Planificación.....	24
1.5.2 Seguimiento.....	25
2 Estudio del estado del arte.....	27
2.1 Campo de aplicación	27
2.2 Análisis del mercado	30
2.2.1 Sistemas comerciales.....	30
2.2.2 Proyectos Open Source	32
2.3 Evaluación de las necesidades del usuario	35
3 Especificaciones del sistema	36
3.1 La tecnología de partida	36
3.1.1 Arduino	36
3.1.2 Waspote.....	39
3.1.3 Raspberry Pi	41
3.2 Especificaciones básicas iniciales	43
3.3 Parámetros de interés para la monitorización.....	46

3.4	Sistemas de visualización.....	47
3.5	Protocolos de comunicación.....	48
3.6	Especificaciones del diseño final	49
3.6.1	Requisitos añadidos.....	49
3.6.2	Resultados finales.....	50
4	Diseño del hardware.....	52
4.1	Diagrama de bloques del sistema	52
4.2	Sensores biométricos integrados	53
4.2.1	Pulso y oxígeno en sangre.....	53
4.2.2	Electrocardiograma	56
4.2.3	Respiración.....	58
4.2.4	Temperatura corporal	61
4.2.5	Tensión Sanguínea	63
4.2.6	Posición del paciente	68
4.2.7	Conductividad de la piel.....	70
4.2.8	Glucosa en sangre.....	72
4.2.9	Electromiografía.....	74
4.3	Electrónica de adaptación.....	76
4.3.1	Sensores.....	77
4.3.2	Otros circuitos	84
4.4	Prototipos de placas de circuito impreso (PCB).....	85
4.4.1	Placa e-Health Versión 1	87
4.4.2	Placa e-Health Versión 2.....	89
5	Diseño de software	91
5.1	Librería e-Health	92
5.1.1	Arduino	92
5.1.2	Raspberry Pi.....	93
5.2	Firmware asociado a los sensores	95
5.2.1	Funciones	95
5.2.2	Programas de ejemplo	107
5.3	Firmware asociado a otras funcionalidades.....	111

5.3.1	Funciones	111
5.3.2	Programas de ejemplo	114
6	Pruebas de laboratorio	117
6.1	Calibración sensores.....	117
6.2	Pruebas de medición.....	118
6.3	Pruebas de compatibilidad	119
7	Conclusiones	125
7.1	Conclusiones personales	126
7.2	Líneas futuras	127
8	Bibliografía	129
Anexo A	Información técnica.....	132
A.1.	Parámetros médicos de interés	132
A.1.1.	Pulso y oxígeno en sangre.....	132
A.1.2.	Electrocardiograma	133
A.1.3.	Respiración.....	134
A.1.4.	Temperatura corporal	135
A.1.5.	Tensión Sanguínea	136
A.1.6.	Posición del paciente	137
A.1.7.	Conductividad de la piel.....	137
A.1.8.	Glucosa en sangre.....	138
A.1.9.	Electromiografía.....	138
A.2	Calibración	139
Anexo B	Códigos de programación	143
B.1	Software de base: Arduino IDE.....	143
B.2	Firmware	144
Anexo C	Documentación online generada	156
C.1	Guía de producto	156
C.2	Videos y presentaciones	157
Anexo D	Eventos y prensa	158
D.1	Premios y eventos.....	158

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	<i>MEMORIA</i>
<i>Revisión n°:3</i>	<i>Fecha: 13/02/2014</i>

D.1.1	Premios.....	158
D.1.2	Eventos	159
D.2	Prensa	160
Anexo E	Kit de producto.....	162
E.1	Kit completo	162
Anexo F	Casos de estudio	164
F.1	Proyectos terminados	164
F.2	Proyectos en desarrollo	165

Índice de figuras

Figura 1 Esquema de una red sensorial inalámbrica	15
Figura 2 Ejemplo de aplicaciones de redes sensoriales inalámbricas	16
Figura 3 Red sensorial inalámbrica	16
Figura 4 Placa básica e-Health Sensor Platform	17
Figura 5 Logotipo del movimiento Open Source	20
Figura 6 Esquema básico telemedicina	21
Figura 7 Cronograma del proyecto.....	24
Figura 8 Imagen promocional e-Health Sensor Platform.....	27
Figura 9 Esquema de posible aplicación de la tecnología Polar en el proyecto.....	31
Figura 10 Producto Grove Heart Rate ear-clip como ejemplo de dispositivo final	33
Figura 11 Evolución de la placa del proyecto Tricoder de la Universidad de Berkeley	34
Figura 12 Placa básica Arduino UNO.....	36
Figura 13 Ejemplo de aplicación encendiendo un LED "Hello World!"	38
Figura 14 Imagen de Waspote con módulos XBee y GPRS incluidos	39
Figura 15 Plataforma Open Source Raspberry Pi	41
Figura 16 Esquema básico de bloques Raspberry Pi.....	42
Figura 17 Pinout básico de Arduino.....	43
Figura 18 Ejemplo de display LCD gráfico	47
Figura 19 Módulo XBee para posible aplicación final con la plataforma e-Health.....	48
Figura 20 Aplicación real del producto acabado.....	50
Figura 21 Diagrama de bloques general del sistema completo e-Health	52
Figura 22 Diagrama de bloques general del sistema e-Health Versión 1	53
Figura 23 Sensor SpO2	54
Figura 24 Sensor SpO2 conectado sobre la placa e-Health.....	55
Figura 25 Medición en proceso del sensor SpO2.....	55

Figura 26 Sensor electrocardiograma.....	56
Figura 27 Conexión de los cables de medida de ECG sobre e-Health	57
Figura 28 Polaridad de las conexiones del ECG	57
Figura 29 Conexión de los parches sobre el paciente para realizar la medida de ECG	58
Figura 30 Sensor de flujo respiratorio.....	59
Figura 31 Polaridad del sensor de respiración.....	60
Figura 32 Polaridad de las conexiones del sensor de respiración	60
Figura 33 Colocación del sensor de respiración sobre un paciente.....	61
Figura 34 Sensor de temperatura corporal	61
Figura 35 Conexión del sensor sobre la placa de e-Health.....	62
Figura 36 Colocación del sensor de temperatura corporal sobre el paciente	62
Figura 37 Sensor tensión sanguínea e-Health Versión 1.....	63
Figura 38 Sensor de tensión sanguínea e-Health Versión 2	64
Figura 39 Medición de tensión sanguínea Versión 1	65
Figura 40 Medida de tensión sanguínea obtenida y almacenada en e-Health.....	65
Figura 41 Medición de tensión sanguínea Versión 2	66
Figura 42 Conexión del sensor de tensión sanguínea Versión 2	66
Figura 43 Conexión sensor tensión sanguínea Versión 2 sobre e-Health Versión 1.....	67
Figura 44 Cable adaptador para hacer compatible el sensor de tensión sanguínea.....	67
Figura 45 Acelerómetro utilizado para detectar la posición del paciente	68
Figura 46 Sensor de posición del paciente	68
Figura 47 Conexión del sensor de posición sobre la placa e-Health	69
Figura 48 Colocación del sensor de posición en el paciente.....	69
Figura 49 Sensor de conductividad de piel	70
Figura 50 Conexión de los sensores conductividad de piel.....	70
Figura 51 Proceso de colocación del sensor de conductividad de piel.....	71

Figura 52 Colocación del sensor de conductividad de piel	71
Figura 53 Sensor de glucosa.....	72
Figura 54 Realización de una medida de glucosa	72
Figura 55 Obtención de la medida de glucosa	73
Figura 56 Conexión del sensor de glucosa	73
Figura 57 Sensor de electromiograma.....	74
Figura 58 Polaridad de los cables de medida de electromiografía	74
Figura 59 Conexión del sensor de electromiograma	75
Figura 60 Colocación del sensor de electromiograma sobre el paciente.....	75
Figura 61 Circuito de adaptación del sensor de SpO2	77
Figura 62 Circuito de adaptación del sensor de electrocardiograma.....	78
Figura 63 Circuito de adaptación del sensor de respiración.....	79
Figura 64 Circuito de adaptación del sensor de temperatura	79
Figura 65 Circuito de adaptación del sensor de tensión sanguínea.....	80
Figura 66 Circuito de adaptación del sensor de conductividad.....	81
Figura 67 Circuito de adaptación del sensor de glucosa	81
Figura 68 Circuito de adaptación del sensor de electromiograma	83
Figura 69 Circuitos y conectores destinados a otras aplicaciones.....	84
Figura 70 Evolución de las distintas placas de prototipado en todo el desarrollo.....	85
Figura 71 Esquemático definitivo de la placa e-Health Versión 2.....	86
Figura 72 Esquema y conexión de e-Health Versión 1	87
Figura 73 e-Health Versión 1 sobre Arduino y Raspberry Pi	87
Figura 74 Conexión final de todos los sensores sobre e-Health Versión 1	88
Figura 75 Conexión final de todos los sensores sobre e-Health Versión 2	89
Figura 76 Esquema y conexión de e-Health Versión 2	90
Figura 77 Layout definitivo de la placa e-Health Versión 2	90

Figura 78 Placa Arduino y placa e-Health	93
Figura 79 Placa Raspberry Pi, placa adaptadora y placa e-Health	93
Figura 80 Visualización de la medida de SpO2 por medio del monitor serie de Arduino	96
Figura 81 Visualización de la medida de respiración por medio del monitor serie de Arduino .	97
Figura 82 Visualización de la medida de temperatura por medio del monitor serie de Arduino	98
Figura 83 Visualización de la medida de tensión sanguínea por medio del monitor serie de Arduino	100
Figura 84 Posiciones detectadas por el sensor de posición	102
Figura 85 Visualización de la medida de posición por medio del monitor serie de Arduino ...	102
Figura 86 Visualización de la medida de conductividad por medio del monitor serie de Arduino	103
Figura 87 Visualización de la medida de glucosa por medio del monitor serie de Arduino.....	105
Figura 88 Visualización de la medida de electromiograma por medio del monitor serie de Arduino	106
Figura 89 Diagrama de flujo de los códigos básicos de lectura de sensores	107
Figura 90 Interfaz del ejemplo en el monitor serie de Arduino	108
Figura 91 Diagrama de flujo del código de interfaz monitor serie.....	109
Figura 92 Display LCD gráfico conectado a Arduino	111
Figura 93 Primera pantalla con parámetros básicos del ejemplo del GLCD en e-Health	113
Figura 94 Segunda pantalla con medición respiratoria del ejemplo del GLCD en e-Health	113
Figura 95 Tercera pantalla con medición ECG del ejemplo del GLCD en e-Health	113
Figura 96 Diagrama de flujo del código de realización de gráficas en tiempo real	114
Figura 97 Representación gráfica de la señal ECG	116
Figura 98 Interfaz básica del programa KST utilizado para graficar las señales de e-Health...	116
Figura 99 Salida de datos genérica por el monitor serie de Arduino	118
Figura 100 Representación gráfica de los valores biométricos medidos con e-Health	118

Figura 101 Módulo WiFi conectado sobre e-Health y Arduino utilizando el módulo de comunicaciones	119
Figura 102 Diagrama de flujo del código de comunicación WiFi con un Smartphone	120
Figura 103 Capturas de pantalla de la aplicación de e-Health para smartphones	122
Figura 104 Ejemplo combinado de e-Health con tecnología bluetooth	123
Figura 105 Ejemplo combinado de e-Health con tecnología XBee	123
Figura 106 Ejemplo combinado de e-Health con tecnología GPRS	124
Figura 107 Fotografía realizada con Arduino y tecnología 3G simulando una picadura de serpiente	124
Figura 108 Señal típica de electrocardiograma	133
Figura 109 Colocación básica del sensor de respiración.....	134
Figura 110 Colocación básica del sensor de temperatura en extremidades	135
Figura 111 Posicionamiento del brazo para realizar la medida de tensión sanguínea	136
Figura 112 Posiciones básicas de un paciente.....	137
Figura 113 Colocación básica del sensor de conductividad.....	138
Figura 114 Valores que hay que calibrar en el sensor de temperatura corporal.....	139
Figura 115 Utilización del multímetro para la calibración del sensor de temperatura.....	140
Figura 116 Valores que hay que calibrar en el sensor conductividad	141
Figura 117 Localización del potenciómetro de calibración del sensor EMG.....	142
Figura 118 Gráfica obtenida con los valores EMG	142
Figura 119 Captura de pantalla del IDE de Arduino.....	143
Figura 120 Impresión de pantalla del tutorial online	156
Figura 121 Video presentación/demostración de la plataforma e-Health	157
Figura 122 Presentación PowerPoint de la plataforma e-Health.....	157
Figura 123 Certificado de finalista internacional del concurso James Dyson Award 2013.....	158
Figura 124 Charla impartida en la Universidad de México	160
Figura 125 Captura de pantalla de la entrevista realizada para Teknautas.....	161

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	<i>MEMORIA</i>
<i>Revisión n°:3</i>	<i>Fecha: 13/02/2014</i>

Figura 126 Pegatina para el kit de e-Health Sensor Platform	162
Figura 127 Caja del kit completo de e-Health Sensor Platform.....	163
Figura 128 Producto final del proyecto OpenPhysio desarrollado en el Fablab Pisa, Italia	164
Figura 129 Captura de pantalla del programa de visualización 3D para e-Health en actual desarrollo.....	165

Índice de Tablas

Tabla 1 Análisis clínico de los valores de SpO2	132
Tabla 2 Ritmo cardíaco según la edad	133
Tabla 3 Reacciones en el ser humano frente a la temperatura corporal	135
Tabla 4 Niveles de tensión sistólica y diastólica de los distintos estados	136

Glosario

ADC	<i>Analog to digital converter</i> (Conversor analógico digital)
T^a	Temperatura
WSN	<i>Wireless sensor network</i> (Red sensorial inalámbrica)
PCB	<i>Printed Circuit Board</i> (Placa de circuito impreso)
SMD	<i>Surface Mount Device</i> (Dispositivo de montaje superficial)
IDE	<i>Integrated Development Environment</i> (Entorno de desarrollo integrado)
GPRS	<i>General Packet Radio Service</i> (Servicio general de paquetes vía radio)
GSM	Sistema global para comunicaciones móviles
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
ECG / EKG	Electrocardiograma
EMG	Electromiograma
EEG	Electroencefalograma
GSR	<i>Galvanic Skin Response</i> (Respuesta galvánica de la piel)
SpO₂	Saturación de oxígeno
SPI	<i>Serial Peripheral Interface</i> (Interfaz de comunicación serie)
EDA	<i>Electrodermal activity</i>
DIY	<i>Do It Yourself</i> (Hazlo tú mismo)

1 Introducción

1.1 Marco de trabajo

El presente Proyecto Fin de Carrera se ha realizado durante 2012-2013 en la empresa Libelium Comunicaciones Distribuidas S.L. bajo la dirección de D. Roberto Casas Nebra como director, profesor titular del departamento de Ingeniería Electrónica y Comunicaciones de la Universidad de Zaragoza, y Álvaro Marco Marco como co-director, profesor asociado del departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza.

El proyecto se ha llevado a cabo íntegramente en las instalaciones de Libelium por un periodo de año y medio, aunque actualmente se sigue trabajando y llevando a cabo un proceso de mejora constante. El proyecto se inició en Junio de 2012 bajo el formato de prácticas/proyecto en empresa de Universa.

1.1.1 Libelium y sus productos

Libelium [1] es una empresa de diseño y fabricación de hardware para la implementación de redes sensoriales inalámbricas, redes malladas y protocolos de comunicación para todo tipo de redes inalámbricas distribuidas.

Nace en 2006 como empresa spin-off de la Universidad de Zaragoza. Sus productos se han vendido en más de 40 países y sus esfuerzos se han visto reconocidos con diversos premios. En ella se utiliza en gran medida el software libre y el firmware tiene una licencia pública GPL (General Public License). Entre los productos de Libelium cabe destacar los más relevantes:

- Waspnote: dispositivo sensorial para la creación de redes sensoriales inalámbricas (actuando como nodo) de bajo consumo [2].
- Meshlium: dispositivo router multitecnología que integra tecnología WiFi, ZigBee, GPRS, GPS y Bluetooth [3].

1.1.2 Las redes sensoriales

Las redes sensoriales inalámbricas, también conocidas como WSN, son redes que buscan monitorizar el entorno. Compuestas por pequeñas máquinas como las de la Figura 1, equipadas con distintos tipos de sensores, trabajan de forma colaborativa para la medición de datos captados del exterior. A cada una de estas máquinas se le llama *mote* o nodo.

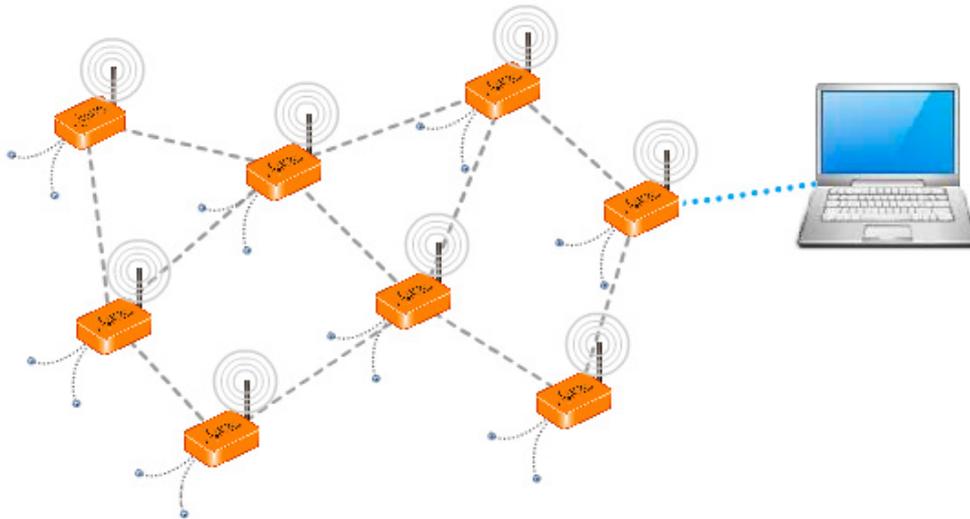


Figura 1 Esquema de una red sensorial inalámbrica

Entre sus ventajas destacar:

- Fácil implantación y escalabilidad (sin cables).
- Auto-organización.
- Escalabilidad.
- Robustez.
- Fiabilidad.
- Seguridad.
- Mínimo mantenimiento.



Figura 2 Ejemplo de aplicaciones de redes sensoriales inalámbricas

En sus orígenes estas redes se enfocaban a aplicaciones militares, pero actualmente están enfocadas a un mayor número de aplicaciones como se puede ver en la Figura 2, entre ellas destacar en aplicaciones de agricultura o ganadería, medio ambiente o todas las relacionadas con el concepto actual de “ciudades inteligentes”. Un ejemplo aplicado con la tecnología de Libelium es el de la Figura 3.

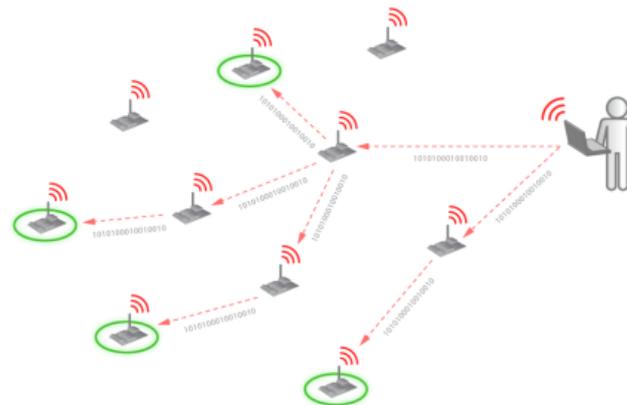


Figura 3 Red sensorial inalámbrica

El campo de aplicación que para este proyecto más interesa es el relacionado con la salud. La instalación de una red sensorial de comunicación en el entorno cercano a un paciente o persona con limitaciones [4], permite una atención y control constante. Es posible incluir sensores de parámetros biométricos, capaces de monitorizar los latidos, la respiración, movimientos, etc. De esta forma es posible dar un servicio de monitorización de constantes vitales desde el propio domicilio del paciente, de manera que su calidad de vida mejore. El pequeño tamaño de los motes posibilita controlar a pacientes que necesitan tener monitorizadas ciertas constantes vitales sin restarles calidad de vida.

1.2 Justificación

1.2.1 Objeto

El objeto del proyecto es el desarrollo de una placa electrónica de medición biométrica, para su uso como sistema de diagnóstico de pacientes en tiempo real, permitiéndonos además el envío de todas las medidas de manera inalámbrica.

El proyecto abarca multitud de campos y tareas a realizar, desde la investigación necesaria para la comprensión del funcionamiento de todos los sensores, hasta el estudio de los componentes para aprender a configurarlos y programarlos de manera correcta.

Como resultado final se espera un dispositivo de bajo coste, que sirva para la monitorización de pacientes en tiempo real.

Para ello comenzaré con una investigación de los dispositivos del mercado con características similares, los distintos sensores y parámetros que son necesarios monitorizar en un paciente y otros temas que se van a trabajar en el proyecto. Se continuará con distintas pruebas con diversos sensores, para adquirir información de las experiencias vividas, comprobando su funcionamiento y su calidad de medición. Tras esto se comenzara el diseño del dispositivo, la integración de los sensores y su fabricación. Finalizando, con una serie de pruebas para comprobar si funciona correctamente y se encuentra dentro de unos estándares mínimos de calidad. El resultado se pude observar en la Figura 4.

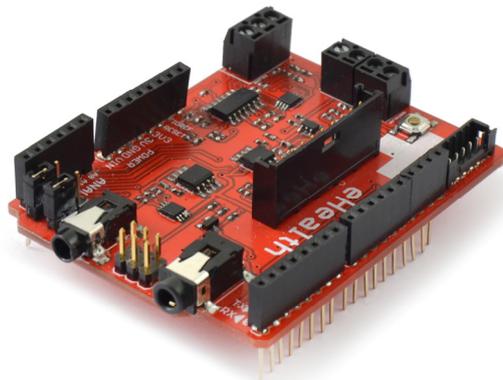


Figura 4 Placa básica e-Health Sensor Platform

1.2.2 Motivación del proyecto

Este proyecto surge de la necesidad de ampliar las posibilidades de la tecnología de Libelium, a un campo que nunca se había aplicado, la medicina y la monitorización de señales biométricas en tiempo real. Surge un tiempo después y apoyándose en el proyecto final de carrera del estudiante Yuri Carmona [5], también desarrollado en la empresa Libelium. En él se buscaba el diseño y la evaluación de un dispositivo biométrico autónomo para escenarios de monitorización, funcionando con la tecnología Waspnote.

El presente proyecto se encuentra englobado dentro del desarrollo de numerosas placas y dispositivos para la comunidad Maker, con el fin de aportar un nuevo enfoque a la tecnología de Libelium, abriéndole las puertas al mundo particular, y permitiendo la interconexión de distintos dispositivos de manera inalámbrica. El principal objetivo de la empresa es el de ampliar el campo de aplicación de su tecnología, pero manteniendo los niveles de rendimiento del sistema (bajo consumo, velocidad de transmisión...).

Este dispositivo permitirá monitorizar tanto a pacientes que se encuentren en un centro sanitario como los que se encuentren en su propio domicilio, incluso mientras duermen. En caso de ser detectado un valor anómalo, esta incidencia puede ser registrada en la red interna. Por ejemplo, pueden ser enviados SMS al Smartphone del médico, reduciendo el tiempo de aviso y por tanto el de actuación.

En vista de que los dispositivos electrónicos médicos tienen unos altos costes y son voluminosos, se busca la creación de un dispositivo que permita realizar esta tarea de manera libre, rápida y en tiempo real, aportándole también la ventaja de la comunicación inalámbrica, todo ello sin un precio elevado. Ante esta nueva posibilidad de mercado, se analizó la idea de que se estaban abriendo las puertas al mundo Open Source a todas las ramas de la tecnología y de la ciencia, pero en el campo de la medicina no ocurría. ¿Por qué? Pensamos que sería por miedo...por problemas con certificaciones....pero aun pensado todo esto, se planteó la posibilidad de obtener el primer sistema "libre" de medición biométrica, accesible para casi cualquier persona gracias a su bajo precio y una documentación realizada muy completa de cómo utilizarlo.

1.3 Objetivos

El objetivo principal de este PFC es el desarrollo, construcción, programación y puesta a punto de un prototipo electrónico autónomo para la monitorización de parámetros biométricos, y que disponga de numerosas otras funcionalidades como la visualización de parámetros en tiempo real o el envío de datos de manera inalámbrica.

Podemos distinguir también una serie de objetivos secundarios relacionados con la finalización del proyecto:

- Investigación en un primer lugar todo lo relacionado con el ámbito del proyecto (redes inalámbricas, la tecnología de partida,...).
- Estudio y selección de los parámetros biométricos más importantes para su aplicación médica. Para ello se recurrirá a la consulta de especialistas o a la búsqueda de redes sensoriales enfocadas a aplicaciones médicas.
- Investigación y prueba de los sensores utilizados y su mercado. Adquiriendo de esta manera los conocimientos necesarios para la realización de elecciones óptimas.
- Comprensión en profundidad del *hardware* utilizado (*datasheet*, documentación técnica...). Diseño de la electrónica de adaptación asociada a los sensores y al resto de componentes del prototipo. Compromiso entre costes y calidad.
- Desarrollo del código *firmware* necesario para controlar el dispositivo, con la consiguiente comprensión de todas las librerías utilizadas. Conocer el lenguaje de programación C-C++, siendo este la base de todos los códigos realizados.
- Diseño de un sistema adecuado, garantizando su autonomía y minimizando el consumo.

Gracias a que la tecnología utilizada se basa en el movimiento “*open hardware-software*”, la curva de aprendizaje es exponencial, lo que facilita notablemente la investigación. No obstante, al tratarse de un campo de aplicación no demasiado explorado, se han tenido que realizar búsquedas en profundidad de información.

1.3.1 Alcance

En estos últimos años, la tecnología basada en hardware y software libre representada con el logo de la Figura 5, como por ejemplo Arduino [6] (sistema desarrollado basado en microcontroladores AtMega), ha iniciado un movimiento muy importante entre la sociedad electrónica, propulsado por las numerosas ventajas que este tipo de dispositivos facilitan. Ya sea en el ámbito personal como en el profesional, las personas comparten sus avances con el resto de la comunidad, lo que permite tener una referencia continua para la realización de distintos proyectos.



Figura 5 Logotipo del movimiento Open Source

Entre este tipo de tecnologías destaca la desarrollada por Libelium bajo licencia Open Source, la cual está en continuo desarrollo y mejora. Este proyecto aporta un nuevo valor añadido a dicha tecnología, permitiendo en cierta medida introducir el hardware y software libre al campo de la medicina. Gracias a esto, se espera que un gran número de empresas que encuentren problemas tanto de costes como de otros tipos (volumen, mantenimiento...) para la adquisición de equipos o implementación de redes basadas en mediciones médicas, vean en este módulo una opción económica para su realización.

La domótica médica, robótica, todas las formas de instrumentación y las nanotecnologías son las principales ramas por las que se mueve el futuro de la raza humana. El mercado de la electrónica personal está creciendo de una manera incalculable, y servirá para crear nuevos productos que monitoricen nuestro entorno. Tendremos que ver y controlar nuestro entorno completo para vivir una vida sana y prolongada.

1.4 Antecedentes

Aunque el concepto de telemedicina puede parecer muy reciente y relacionado con la globalización de las comunicaciones y de internet, existen antecedentes de diagnósticos a distancia desde casi la invención del teléfono. Por ejemplo, en la década de los 60 se logró transmitir electrocardiogramas desde barcos. Hoy en día se cuenta con equipamiento y experiencias satisfactorias de diagnóstico remoto en prácticamente todas las especialidades médicas.

El concepto de historia clínica es el conjunto de documentos que contienen los datos, valoraciones e informaciones de cualquier índole, sobre la situación y la evolución clínica de un paciente a lo largo del proceso asistencial. En este sentido sí que se ha continuado avanzando, como resultado de incorporar las Tecnologías de la Información y la Comunicación (TIC) en el núcleo de la actividad sanitaria. El problema es que para la toma de valores o la realización de mediciones que aporten los datos necesarios para una historia clínica es necesario acudir a un centro sanitario.

Como complemento a esta historia clínica, la industria buscó en las nuevas tecnologías electrónicas una solución eficiente para monitorizar el estado físico de los pacientes. Todo esto ha servido para que el campo de la tele-monitorización se integre con la telemedicina, para poder llevar un control riguroso y en tiempo real de los pacientes siguiendo el esquema de la Figura 6.



Figura 6 Esquema básico telemedicina

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

Aun con todo esto presente, el diagnóstico en tiempo real de pacientes [6] solo podía ser utilizado por grandes empresas o por los distintos sistemas sanitarios de los países, dados sus altos precios. Estos aparatos, además de tener elevados costes no son portátiles ni adaptables a la situación de cada paciente. Son sistemas muy precisos y de altas prestaciones, que deben ser manipulados por personal con una elevada formación.

Esto suponía una barrera para la telemedicina debido a que no era viable el diagnóstico “en casa” de cada uno de los pacientes al no disponer de los dispositivos de medida necesarios, y provocaba graves problemas en los países, como sistemas sanitarios sobresaturados.

La fabricación de un dispositivo médico portable que no solo permitiera la medición biométrica, sino que lo aunara con la intercomunicación inalámbrica, y todo ello a un precio asequible, supondría un dispositivo con mucho potencial y una gran oportunidad en el mercado.

1.5 Desarrollo del proyecto

El desarrollo del proyecto se ha dividido en cuatro fases principalmente:

- En primer lugar se han estudiado las diferentes tecnologías que intervienen en el proyecto: los distintos protocolos de comunicaciones utilizados (UART, SPI...), la tecnología sobre la que se ha trabajado (Waspote, Arduino, Raspberry Pi [7]...), el lenguaje de programación y el entorno de programación que se ha utilizado (C-C++, IDE Arduino...), las distintas posibilidades de hardware a utilizar y todos los demás temas relacionados en menor medida con el proyecto.

En base a la información obtenida se han probado distintos dispositivos médicos comerciales, con el fin de comprobar la viabilidad del módulo como sistema de diagnóstico libre en tiempo real.

- En segundo lugar se ha implementado un prototipo, gracias a toda la investigación previa, a la par que se han probado e integrado todos los sensores posibles, y el estudio y validación de mediciones gracias a medidas y pruebas con dispositivos comerciales certificados.
- Por último, se ha revisado el prototipo para realizar una versión final, y se han llevado a cabo todas las pruebas necesarias para evaluar sus prestaciones y su correcto funcionamiento como dispositivo biomédico de diagnosis.

En paralelo a estas fases se ha ido realizando una memoria de proyecto, en la que se toma nota de todos los avances e investigaciones. Esta memoria ha sido la base de la actual documentación del proyecto.

1.5.1 Planificación

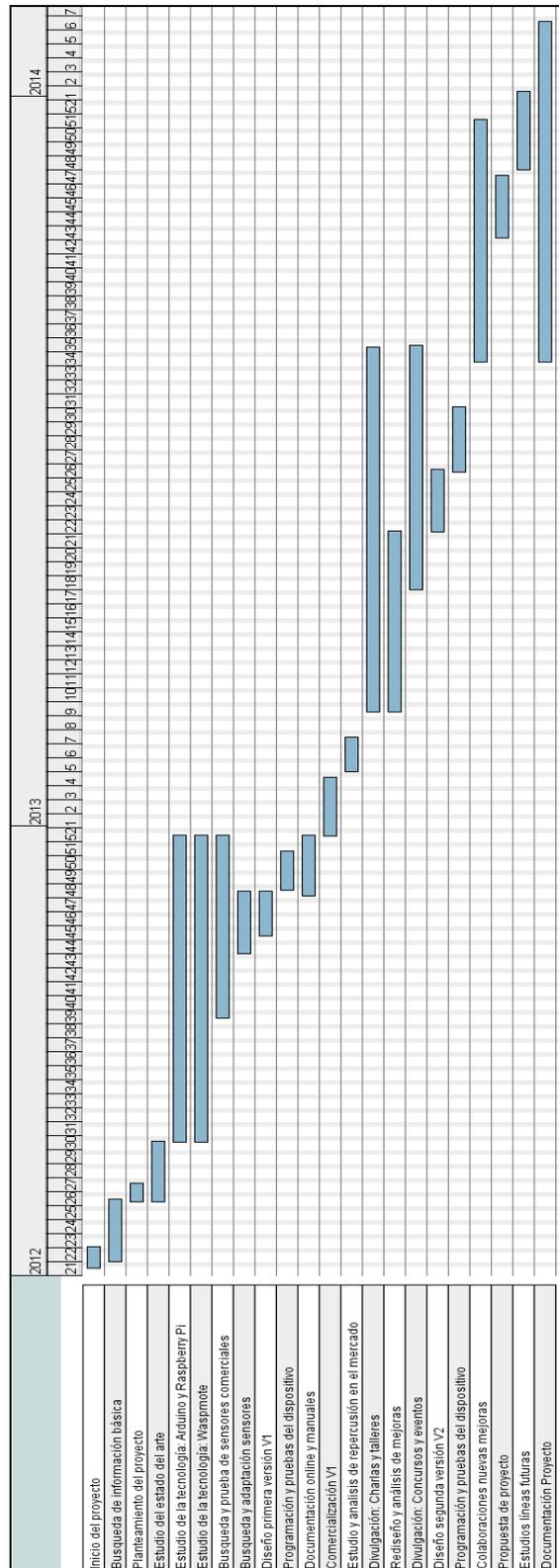


Figura 7 Cronograma del proyecto

1.5.2 Seguimiento

Se ha llevado a cabo un seguimiento de la búsqueda de información y de la ejecución del proyecto, tanto por parte de la Universidad, como por parte de la empresa.

Estas son las dos vías distintas de seguimiento que se han llevado a cabo:

- Con D. Roberto Casas Nebra, ponente y director del proyecto, profesor de la Universidad de Zaragoza y con Álvaro Marco Marco, ponente y co-director, profesor asociado de la Universidad de Zaragoza.

En dichas reuniones se han tratado los siguientes temas:

- Inicio de PFC.
 - Definición del proyecto.
 - Informes de avances.
 - Fijación de los distintos documentos a presentar y sus índices.
 - Revisión y modificación de documentos.
 - Dudas generales.
-
- En la empresa, semanalmente, se han presentado los avances en el proyecto en reuniones con D. David Gascón, y con ellos los problemas que nos han aparecido. Esto ha permitido un avance continuo, sin parones que lo ralenticen demasiado.

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

Los principales problemas tratados han sido:

- Errores de diseño.
- Problemas en la investigación.
- Poca información de ciertos dispositivos.
- Fallos de programación.
- Maquetación de producto.

También los distintos departamentos de la empresa (producción, diseño, I+D...) me han ayudado a resolver algunas de estas dudas o problemas que iban surgiendo en el desarrollo del proyecto.

2 Estudio del estado del arte

2.1 Campo de aplicación

e-Health Sensor Platform es algo más que un diseño, es algo diferente a lo que se ha visto hasta ahora en el campo de la telemedicina y lo que se quiere es que sea una revolución. Se trata de una plataforma de sensores biomédicos para prototipado y desarrollo de aplicaciones médicas donde la misión principal es monitorizar el cuerpo humano. Integra numerosos sensores a un precio relativamente bajo en comparación con los prohibitivos precios de los dispositivos hospitalarios [8].

Esta plataforma quiere dar a la comunidad de desarrolladores e innovadores, tan en auge en el panorama actual, las herramientas para desarrollar y poder trabajar con medidas biométricas en sus proyectos y productos. En la siguiente Figura 8 podemos observar una imagen promocional de la plataforma.



Figura 8 Imagen promocional e-Health Sensor Platform

Se quiere que sea la base de una nueva era de productos Open Source, gracias a que se basa en esta filosofía. Todo es libre, el código, los planos...cualquier persona puede cogerlos y replicarlos, lo único se pide a cambio es que mencionen las fuentes para que otra gente pueda conocer el producto, y suba sus mejoras a Internet, para que todos podamos beneficiarnos con su *feedback*.

El proyecto se enfoca al uso de particulares que deseen monitorizar distintos parámetros médicos o para empresas o grupos de investigación que quieran realizar pruebas básicas con alguno de los sensores biométricos integrados. Al disponer de numerosos protocolos de comunicaciones inalámbrica, las aplicaciones son inmensas y facilitan una monitorización menos molesta e intrusiva para el paciente [9].

Los beneficios del sistema son entre otros:

- Monitorización exhaustiva de los pacientes. Seguimiento continuo de las señales vitales para enfermos crónicos.
- Permite un análisis global de un paciente, y su correcta asistencia remota.
- Detección y prevención de diferentes enfermedades debido a un control diario.
- Interacción remota gracias a la utilización de comunicación inalámbrica.
- Ahorro de costes tanto económicos como en personal, favoreciendo a la no saturación de los sistemas sanitarios.
- Comparación de datos con meses o años anteriores. Sobre todo para el control de ancianos.
- Obtención de los parámetros biométricos en bruto, los que son aplicables para interactuar con proyectos de diversas índoles (como por ejemplo proyectos de arte).

Además del uso principal en monitorización remota, también es aplicable en los campos de la medicina deportiva y en la tele rehabilitación en centros de alto rendimiento.

Los principales beneficios que aportaría a este tipo de aplicaciones sería:

- Reducido tamaño.
- Permite un análisis rápido gracias a su sencillez de uso.
- Interacción remota.
- Reducción de costes.
- Implementación de rutinas para auto-diagnóstico dado su carácter “libre” tanto en hardware como en software.
- Posibilidad de monitorización continúa.

Sumándole a esto todos los beneficios propios de las redes sensoriales (WSN) como son el acceso instantáneo y remoto de datos, fiabilidad, eliminación de cableado, robustez, escalabilidad, sencillez de implantación, mínimo mantenimiento y consumo, precio reducido, flexibilidad y total posibilidad de personalización.

2.2 Análisis del mercado

Hasta la fecha, han sido realizados bastantes proyectos relacionados con las WSN y su aplicación en medicina. Aunque la mayoría de los dispositivos son poco portátiles debido a su gran tamaño, han aparecido algunos equipos "semiportátiles" comerciales que han sido utilizados hasta hace poco para monitorizar a la gente en sus casas. Estos dispositivos presentan muchos inconvenientes, como precio y tamaño, que los sistemas inalámbricos como el que se quiere desarrollar podrían solventar.

2.2.1 Sistemas comerciales

Son dispositivos muy genéricos, y aunque cubren prestaciones planteadas anteriormente como sencillez, bajo coste...., no son abiertos lo que no permite al usuario ningún tipo de versatilidad. No se permiten utilizar más que con unas características básicas definidas de fábrica, por lo que no abren puertas para la creación de proyectos o para el prototipado.

A continuación se presentan algunos de los sistemas comerciales más destacados, y aunque están muy enfocados en una sola funcionalidad, han servido de base para la definición de las especificaciones básicas de la plataforma:

1. Polar Heart Rate: De la empresa Polar [10], especializada en monitorización de deportistas, se trata de un cinturón de pecho que monitoriza distintos parámetros de los pacientes como respiración o posición. Tienen ciertos horizontes de investigación puestos en el mundo Open Source, ya que dejan abierta la comunicación de algunos de sus dispositivos con otras plataformas.

Durante el desarrollo, como se puede observar en la Figura 9, dada la versatilidad de esta tecnología y que disponía de una posibilidad de integración, se barajó la utilización dentro de la plataforma final e-Health Sensor Platform. Reducía costes de fabricación e incluso añadía características al producto, como la monitorización de diversos pacientes. Finalmente se deshecho dados sus elevados costes.

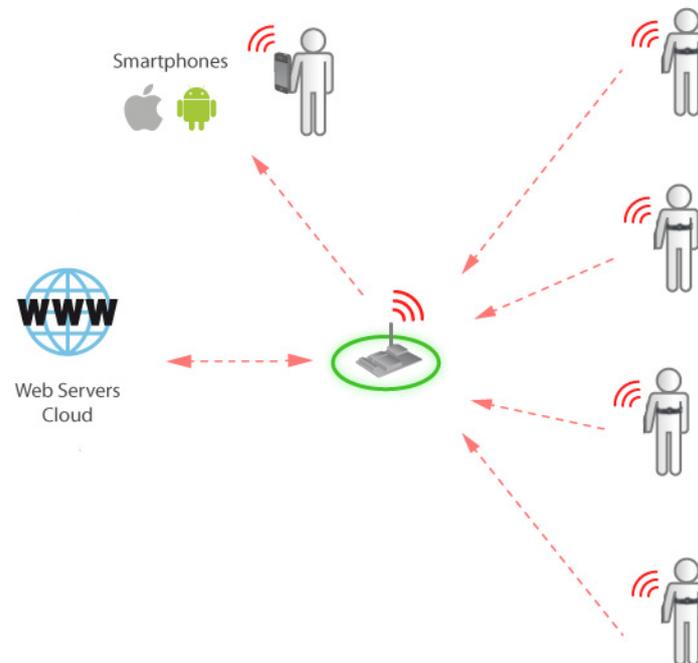


Figura 9 Esquema de posible aplicación de la tecnología Polar en el proyecto

2. Mindwave: Se trata de la empresa que distribuye el juego para niños que consiste en intentar mover con aire una pelota por medio de la mente. Disponen de dispositivos comerciales para la medida de señales EEG. Al igual que Polar, permiten la comunicación con otros dispositivos vía Bluetooth [11].
3. GoodDay: No se trata de un producto en sí, pero es una empresa que fue creada no hace mucho enfocada a la aplicación final de e-Health, o al menos a una de ellas. Se trata de una empresa que dispone de dispositivos de telemedicina, los cuales envían la información al teléfono móvil vía Bluetooth. Utilizan los Smartphone como la plataforma para adquisición, procesado y envío de datos [12].

2.2.2 Proyectos Open Source

En el presente panorama actual, gracias a la financiación colectiva y a diversos movimientos en torno a la filosofía Maker y OpenSource, han aparecido multitud de dispositivos comerciales acabados, con cierto matiz “libre” que sirven de base para todos los demás proyectos que seguirán surgiendo en torno a la telemedicina.

Han sido realizados varios proyectos que utilizan comunicación inalámbrica con intereses biomédicos [13]. Las numerosas aplicaciones encontradas integran sensores que tratan de medir parámetros similares. Este estudio de los actuales dispositivos del mercado ha servido para una primera definición de parámetros a monitorizar:

- Actividad eléctrica del corazón en forma de electrocardiograma (ECG o EKG).
- Análisis de pulsioximetría, medida de la saturación de oxígeno en sangre y de pulsaciones por minuto, así como la onda de pulso pletismográfica.
- Presión arterial.
- Respiración y picos de flujo respiratorio.
- Actividad cerebral mediante electroencefalograma (EEG).
- Temperatura corporal de un individuo.
- Actividad muscular mediante sensores de electromiograma (EMG). Así como el movimiento de los ojos en fases de sueño.
- Sensores de detección de ronquidos.
- Sudoración (respuesta galvánica de la piel). Indicador de estado emocional del sujeto.
- Diferentes sensores como el giroscopio o el acelerómetro, relacionados con la posición del individuo, movimiento de las extremidades corporales, etc.

- Parámetros ambientales que pueden afectar en el estado del individuo, véase: Temperatura ambiental, altura, presión, atmosférica, nivel de oxígeno ambiental, entre otros.

A continuación se presentan algunos de los sistemas Open Source más destacados, que han servido de base para la definición de las especificaciones básicas de la plataforma:

1. Meet Easy Pulse: se trata de un kit DIY que dispone de todos los componentes necesarios para construir un sensor de pulso cardíaco. Es libre y muy sencillo de construir, aunque no es compatible con otras plataformas [14].
2. Grove Heart Rate ear-clip: Similar al anterior, aunque tomando la medida en la oreja como se observa en la Figura 10. Se trata de un pequeño dispositivo libre que realiza una medición de pulso. Es de la empresa asiática Seedstudio, fabricante y distribuidor mundial de electrónica Open Source [15].



Figura 10 Producto Grove Heart Rate ear-clip como ejemplo de dispositivo final

3. EMG Muscle Sensor Kit: Dispositivo libre para la medición de impulsos musculares básicos. Disponible para su compra en la empresa Advancer Technologies [16].

4. Galvanic Skin Response Sensor for Iphone: Se trata de un proyecto que busca utilizar Arduino y un Smartphone para monitorizar las señales GSR de un paciente [17].
5. BioSensorBoard: Placa para Arduino que permite la monitorización del pulso y de la señal GSR de dos personas, únicamente colocando los dedos en zonas de la placa [18].
6. The Berkeley Tricoder: Placa muy similar a la plataforma de este proyecto desarrollada durante 5 años en la Universidad de Berkeley. En la Figura 11 podemos ver su evolución. Integra diversos sensores como: ECG, EMG, respiración, SPO2 y un acelerómetro. Cabe destacar que tiene incorporado un módulo Bluetooth y una SD para la adquisición, almacenamiento y envío de datos [19].

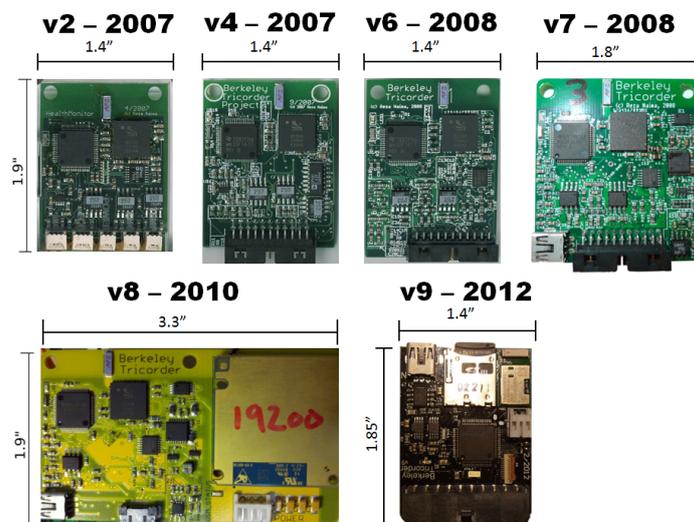


Figura 11 Evolución de la placa del proyecto Tricoder de la Universidad de Berkeley

El valor añadido de la plataforma presentada en este frente a las ya existentes es la versatilidad que aportara una tecnología de base tan modular y extendida como es Arduino y Raspberry Pi, permitiendo una interconexión y facilidad de prototipado muy rápida para implementación rápida de proyectos biométricos.

2.3 Evaluación de las necesidades del usuario

En primer lugar, fue muy importante distintas reuniones con personal médico con el fin de obtener su opinión y puntos de vista sobre la telemedicina. Esto sirvió junto al estudio del estado del arte, para una primera definición de características de la plataforma.

Por otra parte, en Libelium, se tuvo acceso a numerosos sistemas comerciales médicos, con los que se obtuvo una primera información sobre metodología y dispositivos actuales médicos.

Vía internet, he podido preguntar y obtener información relevante sobre los distintos protocolos utilizados y tecnologías [20].

Se han planteado los siguientes objetivos de cara a las necesidades del mercado y de los usuarios:

- Lo más relevante de cara a un dispositivo, es reducir el tamaño y precio de los actuales sistemas comercializados, para que cada persona pueda tenerlo en su hogar sin un gran desembolso económico.
- No todos los pacientes necesitan de todas las tecnologías y sensores, por lo que debe ser lo más modular posible.

En general, los sensores más necesitados son de temperatura corporal, pulso, acelerómetros y también ECG, EMG, EEG. Además de monitorizar el ambiente en el que se encuentra el paciente temperatura de la casa, humedad relativa, presión atmosférica, sensores de iluminación, micrófonos en partes diferentes del hogar, sensores de presión sensitiva en sofás, sillas y camas, entre otros. Aunque dado que trabajamos con una plataforma abierta, esto se puede dejar un poco apartado para futuras actualizaciones.

Como conclusiones se pudo discernir entre numerosos sensores, y se pensó que era muy difícil abarcar todas las patologías, por lo que sería conveniente integrar los sensores más básicos, enfocados a la monitorización de pacientes crónicos, como pueden ser los ancianos.

3 Especificaciones del sistema

3.1 La tecnología de partida

3.1.1 Arduino

Arduino es una plataforma de hardware abierto basada en los microcontroladores AVR de ATMEL. Se trata de una sencilla placa con entradas y salidas (E/S) analógicas y digitales que incluye un entorno de desarrollo para implementar el código en lenguaje C-C++. La placa básica de la Figura 12 es un Arduino UNO.

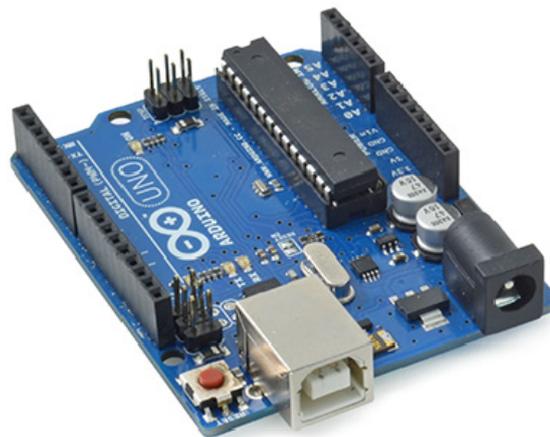


Figura 12 Placa básica Arduino UNO

Originalmente estaba pensada para el desarrollo de proyectos artísticos y entornos interactivos, pero debido a su versatilidad y facilidad de uso, se está usando en muchos otros proyectos.

Algunos de sus usos más comunes:

- Creación de arte electrónico.
- Sistemas para tomar lectura de sensores.
- Control de motores y robots.
- Sistemas de seguridad y control domótico.

Para conseguir todo ello se ha creado alrededor de Arduino una serie de comunidades que se encargan de diseñar placas para dar a la placa muchas más funcionalidades de las básicas.

Muchas placas se han desarrollado por empresas privadas o por desarrolladores aficionados lo que hace que la competitividad entre ellos por dar nuevas funcionalidades a Arduino sea constante. Esto contribuye a una continua publicación de proyectos y nuevos desarrollos en la red.

Ventajas de Arduino

Hay muchos otros microcontroladores y plataformas con microcontroladores disponibles para la computación física. Muchos otros ofrecen funcionalidades similares. Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en paquetes fáciles de usar. Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas a profesores y estudiantes:

- Asequible: Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos de 60€
- Multi-Plataforma: El software de Arduino funciona en los sistemas operativos Windows, Mac y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.
- Entorno de programación simple y directo: El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.
- Software ampliable y de código abierto: El software Arduino está publicado bajo una licencia libre y preparada para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la

programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente el código en tus programas si así lo deseas.

- Hardware ampliable y de código abierto: Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero. En la Figura 13 encontramos un ejemplo de aplicación.

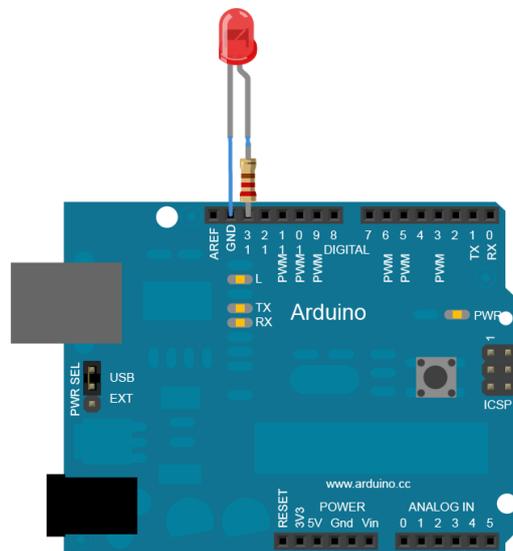


Figura 13 Ejemplo de aplicación encendiendo un LED "Hello World!"

Sin embargo Arduino también tiene algunos críticos. Aunque es evidente que la placa presenta unas características como herramienta de prototipado rápido, con un hardware potente y barato, y con unas herramientas de desarrollo software muy bien cuidadas, hay gente que duda de la capacidad de Arduino para ser una herramienta de aprendizaje real. Ya que muchos detalles están ocultos para los programadores y desarrolladores, y no enseñan las bases de los sistemas basados en micro-controladores. Por decirlo de otra manera, Arduino enseña a usar los micros, pero no enseña “qué” son los micros. Pero aun así, sigue siendo una herramienta muy buena para iniciar a la gente en el mundillo de la electrónica y que no dispone de estudios universitarios o superiores.

3.1.2 Wasmote

Wasmote es un dispositivo sensorial inalámbrico, autónomo y de bajo consumo pensado para monitorizar cualquier tipo de sensor. La Figura 14 nos muestra un Wasmote con diversos módulos conectados.



Figura 14 Imagen de Wasmote con módulos XBee y GPRS incluidos

Fue desarrollado por el equipo de Libelium bajo los términos de software abierto Open Source, siguiendo la filosofía de la plataforma Arduino. Esto significa que cualquier persona tiene acceso y está autorizada a modificar la plataforma a nivel de código.

En noviembre de 2009 fue lanzado a la venta el developer pack, que contiene cinco Wasmotes completos y un Wasmote Gateway (conecta la red a un ordenador por su USB).

Wasmote está basado en el microcontrolador ATmega1281, que funciona a 8 MHz (ejecuta una instrucción de bajo nivel cada 125 ns) y que le proporciona 8 kB de SRAM, 4 kB de EEPROM y 128 kB de memoria flash.

En conclusión, Waspote es el mejor dispositivo en el mercado porque sus características destacan frente a los demás:

- Alta modularidad: solo se compra lo que se necesita.
- Muy bajo consumo.
- Posibilidad de alimentarlo con energía solar (larga vida útil).
- Posibilidad de incluir casi cualquier tipo de sensor.
- Muchas entradas, tanto analógicas como digitales: se pueden conectar muchos sensores al mismo Waspote.
- Gran variedad de módulos de comunicaciones inalámbrica, con alcances muy grandes.
- Posibilidad de GPRS y módulos extra (GPS, SD).
- Plataforma abierta y personalizable.
- La comunidad de usuarios crea nuevas utilidades y mejora las existentes.
- Precio muy competitivo y ausencia de cuotas anuales.

3.1.3 Raspberry Pi

Raspberry Pi es un ordenador de placa reducida de bajo costo, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

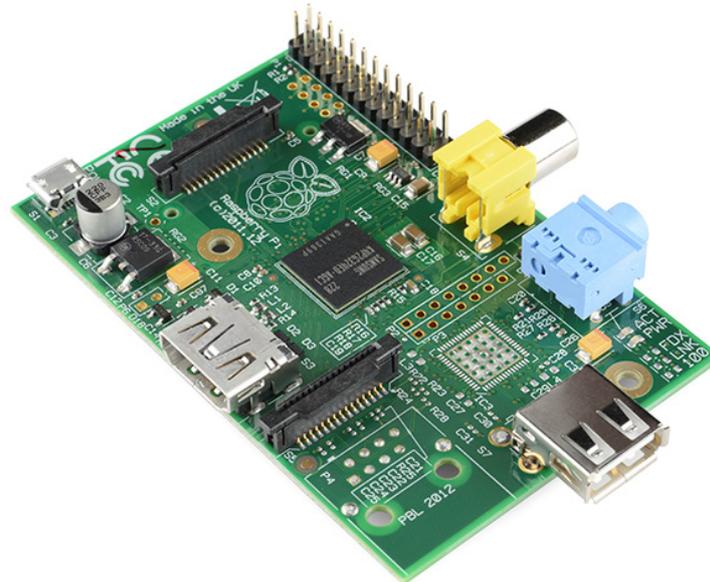


Figura 15 Plataforma Open Source Raspberry Pi

Con unas dimensiones de placa de 8.5 por 5.3 cm, en el modelo B de la Raspberry Pi que es, el que se comercializa ahora como muestra la Figura 15, nos encontramos con unas características muy interesantes. Raspberry Pi. En su corazón nos encontramos con un chip integrado Broadcom BCM2835, que contiene un procesador ARM11 con varias frecuencias de funcionamiento y la posibilidad de subirla (overclocking) hasta 1 GHz sin perder la garantía, un procesador gráfico VideoCore IV, y distintas cantidades de memoria RAM.

Las últimas Raspberry Pi cuentan con 512 MB de memoria. Todo ello equivale en la práctica a un ordenador con unas capacidades gráficas similares a la XBOX de Microsoft y con la posibilidad de reproducir vídeo en 1080p.

En la placa de la Raspberry Pi nos encontramos además con una salida de vídeo y audio a través de un conector HDMI, con lo que conseguiremos conectar la tarjeta tanto a televisores como a monitores que cuenten con dicha conexión como presenta la Figura 16. En cuanto a vídeo se refiere, también cuenta con una salida de vídeo compuesto y una salida de audio a través de un mini Jack.

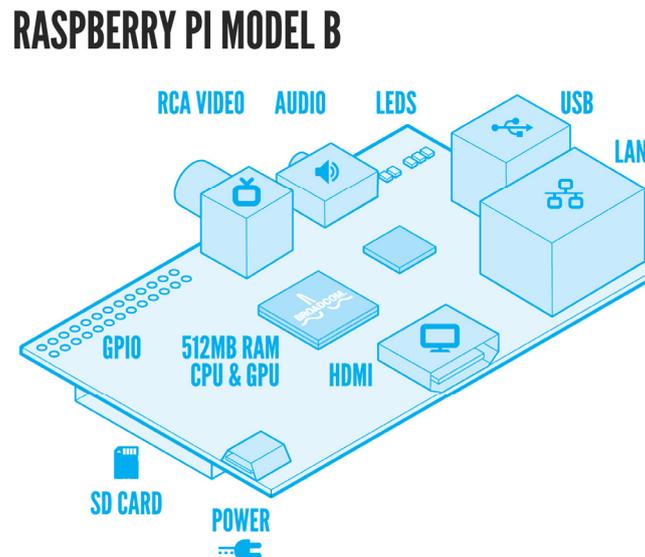


Figura 16 Esquema básico de bloques Raspberry Pi

La Raspberry Pi se entrega sin sistema operativo; éste deberemos instalarlo sobre una tarjeta SD que introduciremos en la ranura de la Raspberry Pi. Tenemos dos opciones, la primera es descargarnos desde la página oficial, todo el software para instalarlo en la tarjeta, o bien recurrir a algún programa que haga esto por nosotros. Entre estos programas encontramos BerryBoot, un programa que se encarga de todo el trabajo de instalación del software desde la propia Raspberry Pi. Una vez copiados los archivos a la tarjeta sd, la introducimos en nuestra Raspberry Pi y BerryBoot te permitirá elegir el sistema operativo descargándolo desde internet. Otra opción interesante es Noobs, una aplicación que facilita la instalación de diversas distribuciones Linux. RaspbianNoobs hace innecesario el acceso a Internet durante la instalación, y tan sólo tendremos que descargar Noobs y descomprimirlo en una tarjeta SD de al menos 4GB de capacidad.

3.2 Especificaciones básicas iniciales

El dispositivo va a ser un módulo central, y debe adaptarse a la placa de Arduino mostrado en la Figura 17, o sobre la que vaya conectarse, ya que no puede trabajar individualmente como sistema. La placa se conecta por medio de dos zócalos de conectores disponibles, mediante los que se alimenta y se comunica con el micro controlador principal. Para un primer planteamiento supondremos la utilización de Arduino como procesador principal.

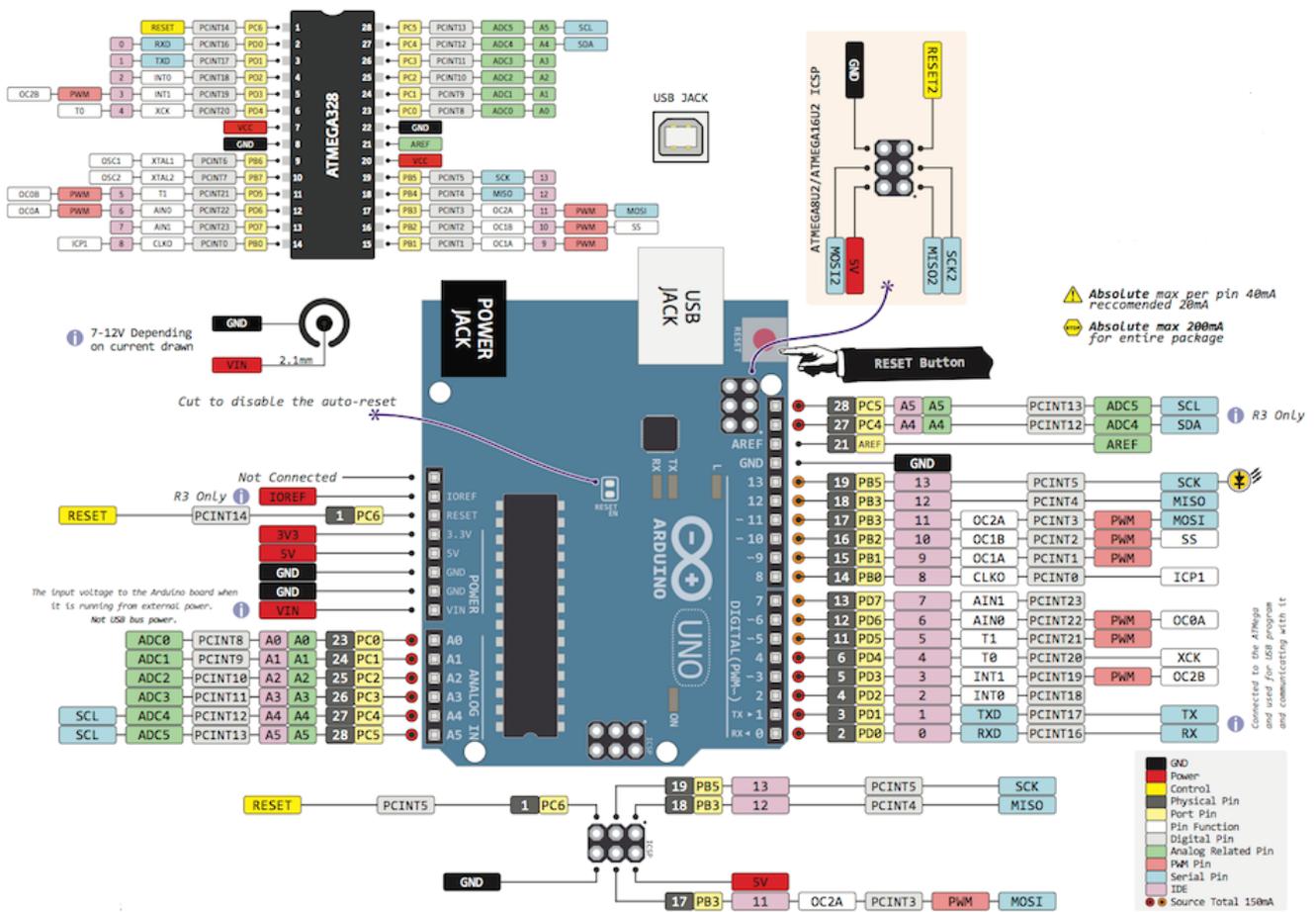


Figura 17 Pinout básico de Arduino

Estos conectores disponen de:

- Pines de Alimentación: GND, VIN, 5V y 3.3V.
- Pines de conexión serie: 0 (RX) y 1 (TX). Usado para recibir (RX) y transmitir (TX) datos a través de puerto serie TTL. Estos pines está conectados a los pines correspondientes del chip FTDI USB-to-TTL.
- Pines digitales PWM: 3, 5, 6, 9, 10, y 11. Proporciona una salida PWM (valores de 0 a 255).
- Conexión a las líneas del bus de comunicación serie SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines proporcionan comunicación SPI.
- Conexión a entradas para interrupción a nivel: 2 y 3. Estos pines se pueden configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH(5V) o viceversa), o en cambios de valor.
- Pines de entradas analógicas: cada una de ellas proporciona una resolución de 10 bits (1024 valores). Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar la cota superior de este rango usando el pin AREF.
- Conexión I2C: 4 (SDA) y 5 (SCL). Soporte del protocolo de comunicaciones I2C (TWI).

En cuanto a las especificaciones de diseño básicas cabe destacar:

- Topología: Tomará como hardware base el sistema Arduino y será compatible con Raspberry Pi.
- Alimentación: Dispondremos de alimentación de 5 o 3.3 voltios.
- Dimensiones: Las dimensiones del Arduino son: 68 x 53mm. El dispositivo debe adaptarse al tamaño y tener unas dimensiones similares a las de los módulos ya existentes para Arduino (XBee, RFID...).
- Peso: Aunque no hay condicionantes sobre el peso del dispositivo, es recomendable que sea el menor posible dadas las condiciones de utilización del módulo.
- Coste: Se pretende construir un equipo con un límite de coste de unos cientos de euros, sin ser demasiado restrictivo este tema. Se pone una primera meta de no superar los 500 euros por el dispositivo completo.
- Condiciones de temperatura: El dispositivo deberá trabajar en condiciones de temperatura ambiente, porque el módulo deberá poder operar entre -10 y 45 °C, rango marcado por lo que cada componente electrónico puede soportar.
- Autonomía: En esta especificación no existe demasiada restricción, pero siguiendo las especificaciones de diseño de la tecnología de Libelium, se intentará reducir al mínimo el consumo con el fin de mejorar la autonomía del conjunto.
- Limitaciones: Arduino tiene limitaciones en su diseño y hay que tenerlas muy en cuenta a la hora de diseñar un nuevo dispositivo integrable en él. Entre estas limitaciones, está su velocidad de comunicación UART (que está limitada a 115200 en la versión más nueva) o su memoria para programación (32KB).

3.3 Parámetros de interés para la monitorización

Después del estudio realizado, han sido escogidos los parámetros que muestran mayores posibilidades de integración dependiendo de diversos factores (dificultad y posibilidad de integración, aplicaciones posteriores...). Para todos estos parámetros se deberá buscar el sensor necesario para su medición y estudiar si es viable su implementación.

Se tendrán en cuenta distintas posibilidades:

- Trabajar con un sensor comercial e integrarlo en la plataforma.
- Realizar la integración y la electrónica para la medición de un parámetro.
- Fabricación del propio sensor de medida.

La elección de sensores se deja abierta en todo el desarrollo de la plataforma, esperando posibles mejoras aportadas por los miembros de la comunidad Open Source, o mejoras que se descubran sobre nuevos sensores.

Aunque se definen unos parámetros de base sobre los que investigar: oxígeno en sangre, pulso, ECG, respiración, temperatura corporal, tensión sanguínea, posición del paciente, conductividad de la piel, glucosa en sangre y EMG. Todos ellos se encuentran explicados en el Anexo A.

3.4 Sistemas de visualización

Se plantean distintos sistemas de visualización de los datos, pensando en aplicaciones finales y hacer el producto más atractivo para los usuarios:

- Monitorización directa en el ordenador, por medio de conexión vía USB.
- Ejemplo de monitorización y representación gráfica en tiempo real.
- APP para móviles como ejemplo de conexión inalámbrica del sistema.
- Monitorización del sistema de manera autónoma por medio de alguna pantalla o display gráfico como el que aparece en la Figura 18.
- Almacenamiento de datos en tiempo real. Puede servir para este ejemplo el almacenamiento en una SD.

Se establecen ciertas prioridades de implementación destacando: APP para móviles, LCD gráfico y monitorización en tiempo real por ordenador.

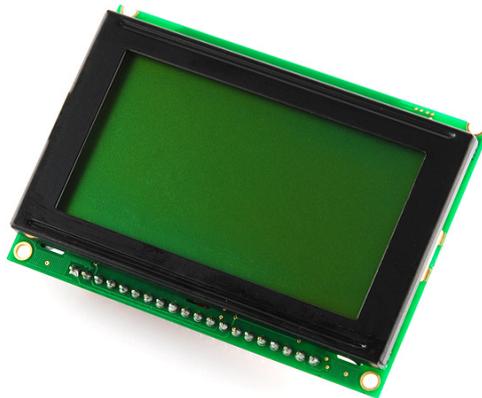


Figura 18 Ejemplo de display LCD gráfico

3.5 Protocolos de comunicación

Una de los principales usos de la telemedicina es el poder disponer de los datos médicos a distancia. Por eso una de las características básicas de nuestro sistema es la posibilidad de comunicarse de manera inalámbrica, conectarse a la nube o enviar las mediciones al médico. Al tratarse de un sistema libre y modular, podemos realizar ejemplos de aplicación con diversos protocolos:

- Ejemplos de comunicación WiFi con teléfonos móviles.
- Comunicación vía Bluetooth.
- Envío de datos por protocolos XBee (Figura 19) para aplicaciones de redes sensoriales.
- Utilización de tecnología GPRS y 3G para envío de SMS, subida de datos a la nube o realización y envío de imágenes para diagnóstico.

En el caso de protocolos de comunicación se decide investigar al máximo el número de posibilidades, por lo que se plantea la implementación de ejemplos combinados con todas las tecnologías.



Figura 19 Módulo XBee para posible aplicación final con la plataforma e-Health

3.6 Especificaciones del diseño final

Se partirá de los requisitos previamente comentados, aunque a lo largo del proceso de desarrollo pueden ser rechazados o mejorados en función de la situación y los datos analizados.

3.6.1 Requisitos añadidos

Tras las distintas investigaciones, experiencias y ejemplos aplicados en el campo de la telemedicina, se han barajado una serie de requisitos a cumplir, añadidos a los iniciales:

- Basar el *hardware* del dispositivo en integrados y etapas proporcionadas por las distintas casas ya que facilitan información y esquemas específicos para aplicaciones médicas [21] [22].
- Trabajar con tecnología SMD únicamente dado el pequeño tamaño del dispositivo y el lugar donde va a ir colocado. Esto reducirá en gran parte las dimensiones del prototipo, y facilitará el diseño del ruteado [23] [24] [25].
- Hacer compatible el dispositivo con Raspberry Pi, para conseguir una mayor aceptación de producto entre la comunidad y abrirle nuevas vías de desarrollo.
- Analizar si es más barato fabricar el sensor propiamente dicho, o adaptar uno ya comercial al sistema.

3.6.2 Resultados finales

El resultado final es un dispositivo, que dote a Arduino y Raspberry Pi del soporte físico necesario, para realizar diagnóstico de pacientes en tiempo real como el ejemplo de la Figura 20. e-Health Sensor Platform es una plataforma que quiere liberalizar el mercado de la medicina y la telemedicina. La mayor parte de los dispositivos médicos tienen licencias privadas y precios elevados, por lo que no es posible investigar con ellos a no ser que tengas una gran empresa a tus espaldas.

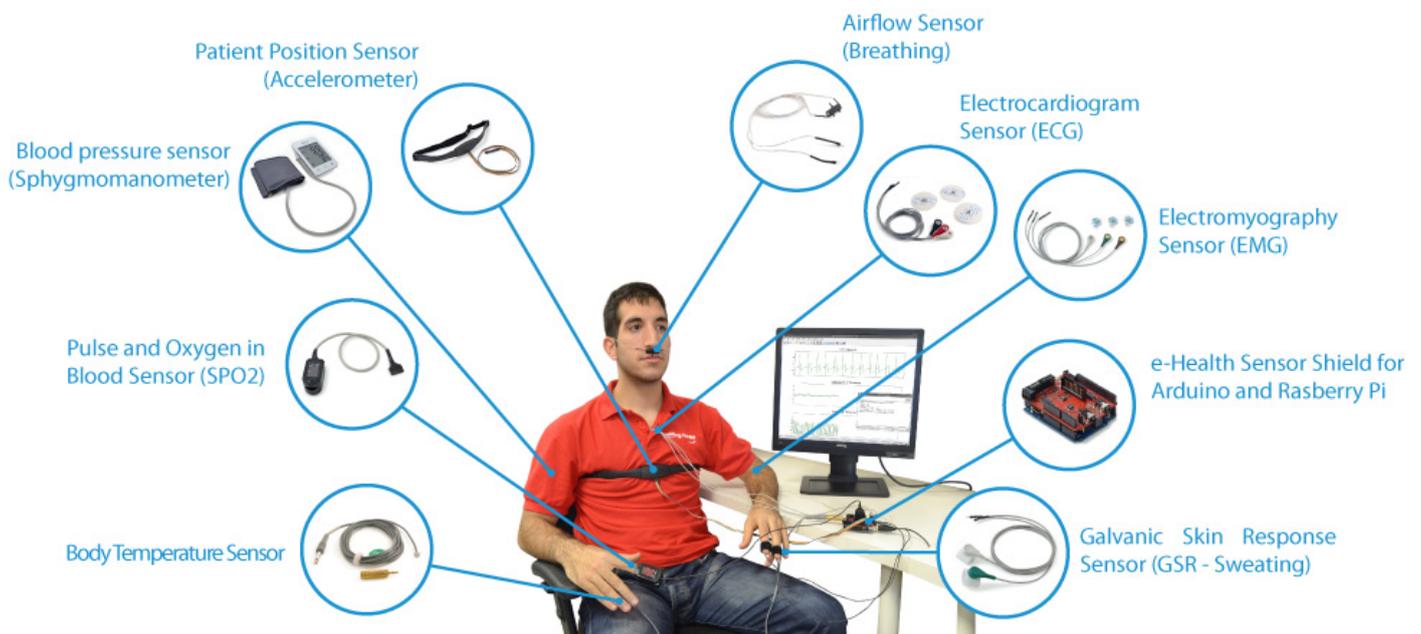


Figura 20 Aplicación real del producto acabado

Conectada a un microcontrolador (Arduino o Raspberry Pi) encargado de tramitar las señales que la placa e-Health mide con 10 distintos tipos de sensores médicos:

- Pulso cardíaco: capaz de medir el ritmo de pulsaciones del corazón.
- Oxígeno en sangre: medición de los niveles de oxígeno en el torrente sanguíneo.

- Respiración: captura del ritmo respiratorio, y de las ausencias de él. Diseñado para trastornos respiratorios o del sueño (apnea).
- Temperatura corporal: variación de la temperatura media del paciente.
- Electrocardiograma (ECG): medición de las señales eléctricas que se producen con el latido del corazón.
- Electromiograma (EMG): medición de las señales eléctricas que se producen con el movimiento de los músculos.
- Glucosa en sangre: monitorización de los niveles de azúcar en sangre, registrando hora y fecha, para poder realizar seguimiento y estadísticas del paciente.
- Tensión sanguínea: monitorización de los niveles de tensión sanguínea, registrando hora y fecha, para poder realizar seguimiento y estadísticas del paciente.
- Posición del paciente: capaz de detectar en qué posición se encuentra el paciente, o si ha sufrido caídas.
- Sudoración de la piel (GSR): basado en el principio de "la máquina de la verdad", mide la resistencia y la conductividad de la piel, detectando variaciones de sudoración.

Lo que por otro lado diferencia a e-Health de todos los productos del mercado es sus múltiples posibilidades de aplicación. Todos estos parámetros pueden ser tomados y almacenados en la memoria, transmitidos a un ordenador, o simplemente enviado por algún protocolo de comunicaciones gracias a una tecnología libre y modular: radiofrecuencia, WiFi, GPRS, Bluetooth, 3G...

4 Diseño del hardware

El *hardware* de este proyecto conforma la creación y diseño de la plataforma de monitorización biométrica e-Health Sensor Platform. En esta placa se incluyen los componentes e integrados necesarios para su correcto funcionamiento.

4.1 Diagrama de bloques del sistema

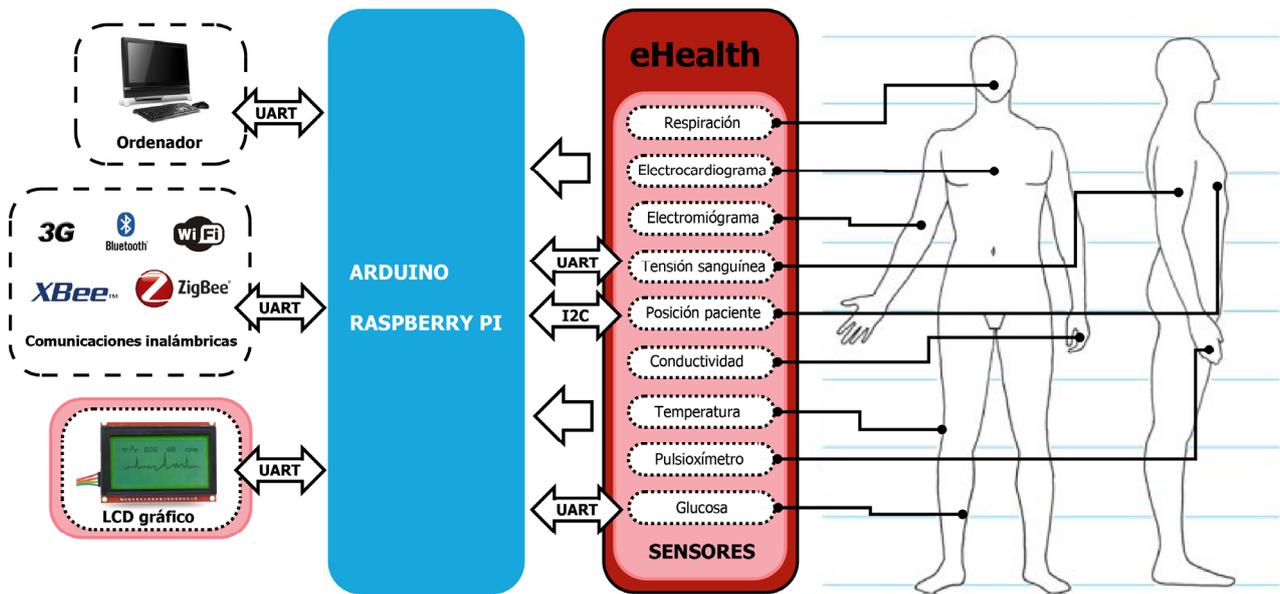


Figura 21 Diagrama de bloques general del sistema completo e-Health

Los datos son tomados por la placa de adaptación que se conecta mediante entradas y salidas disponibles en Arduino o Raspberry Pi. La mayoría de los sensores utilizan entradas analógicas. Sin embargo algunos de ellos se basan en protocolos de medida u obtención de datos distintos, como son: el sensor de tensión sanguínea por UART, el glucómetro por UART, el sensor de SPO2 por entradas digitales y el sensor de posición por I2C.

Como podemos observar en la Figura 21, disponemos de posibilidad de conectar diversos dispositivos y tecnologías por medio de comunicación UART. Desde la utilización de un ordenador como interfaz de visualización de datos, pasando por todos los posibles protocolos inalámbricos, hasta la utilización de un display LCD gráfico para tener un sistema autónomo de medida.

Previo a este diagrama de bloques general, asociado a la Versión 2 de la plataforma, se definió un primer diagrama de bloques que podemos observar en la Figura 22, con un menor número de requerimientos.

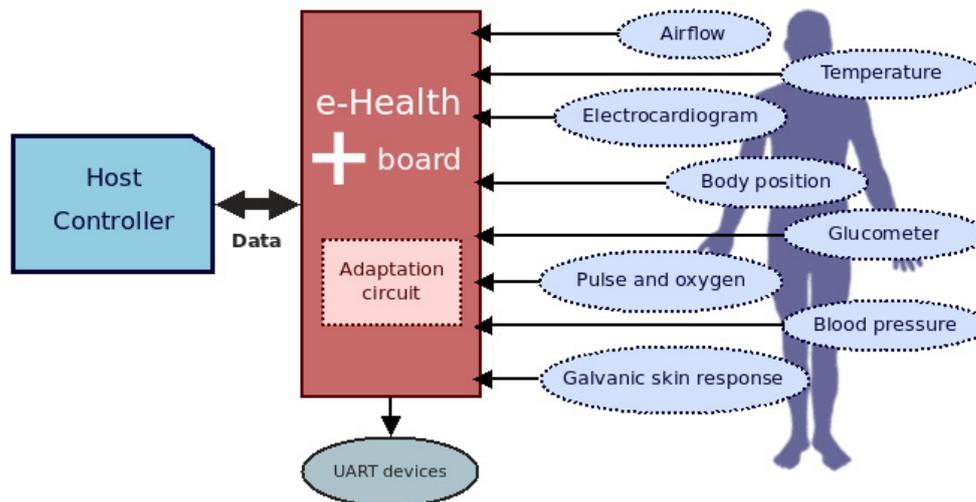


Figura 22 Diagrama de bloques general del sistema e-Health Versión 1

4.2 Sensores biométricos integrados

El estado actual de la placa es una segunda versión con nuevos sensores y modificaciones con respecto a la primera versión comercializada. En este apartado se tratarán todos los sensores independientemente de la versión en la que estén integrados, realizando las pertinentes aclaraciones en la sección referida a las versiones de la placa.

4.2.1 Pulso y oxígeno en sangre

La medición del pulso y oxigenación en sangre es posible ser realizada mediante un único dispositivo, el pulsioxímetro. La mayor desventaja de estos dispositivos es su coste. Esencialmente consiste de dos componentes:

- Una sonda.
- Un oxímetro.

La sonda es colocada normalmente en el dedo índice, aunque también admite otras localizaciones: el lóbulo de la oreja en adultos o los pies en neonatos. El sensor incluye unos LEDs y un fotodetector. Emite luz con dos longitudes de onda : 660nm (roja) y 940nm (infrarroja), que son longitudes de onda características de la oxihemoglobina y la hemoglobina reducida, respectivamente. La mayor parte de la luz es absorbida por el tejido conectivo, piel, hueso y sangre venosa en una cantidad constante, produciéndose un pequeño incremento de esta absorción en la sangre arterial con cada latido. El oxímetro de este sistema es el encargado de procesar la información. Mediante la comparación de la luz absorbida a lo largo del tiempo y el uso de algoritmos adecuados, son calculados los parámetros deseados.

Tras el análisis de estos dispositivos y observar sus altos precios, y descartar la posibilidad de implementar toda la electrónica por la misma causa, se opta por el “hacking” de un dispositivo comercial, que ya dispone de toda la electrónica integrada. El resultado se puede observar en la Figura 23.



Figura 23 Sensor SpO2

Para ello se compraron diversos dispositivos de medición independiente de SPO2 y se realizó ingeniería inversa con el fin de estudiar su funcionamiento, y la mejor manera de obtener los datos. Esto supone un coste extra en producción del sensor, pero aun con esto tenido en cuenta reduce notablemente el coste.

Colocación y uso

Este conector dispone de un conector de una única posición en la parte superior de la placa para su correcta colocación como muestra la Figura 24.



Figura 24 Sensor SpO2 conectado sobre la placa e-Health

Una vez conectado únicamente es necesario introducir el dedo y presionar el botón de ON. Tras unos segundos, como ocurre en la Figura 25, el valor es tomado por el sensor y mostrado en la pantalla, del cual se obtiene la medida en tiempo real.



Figura 25 Medición en proceso del sensor SpO2

4.2.2 Electrocardiograma

El electrocardiograma es medido a través de electrodos [26]. Actualmente, se encuentran dos formas diferentes de electrocardiografía. La primera y más notable, el “ECG estándar”, consta de 12 derivaciones, que son el resultado de la exploración superficial del corazón desde distintos planos a través de 12 electrodos.

La segunda forma de electrocardiografía está relacionada con los sistemas de “telemetría continua de electrocardiograma” para una monitorización prolongada [27]. Éstos incluyen el uso de ECG de dos o tres electrodos, como el de la Figura 26, para evaluar la actividad eléctrica del corazón durante un tiempo extenso.



Figura 26 Sensor electrocardiograma

Para ello se utilizan parches de medición reutilizables con sus pertinentes cables de conexionado. Lo que permite colocar en el paciente las tomas, y poder realizar el electrocardiograma de 3 derivaciones correctamente. Para la correcta medición es necesaria su correspondiente circuitería de adaptación.

Colocación y uso

Al tratarse de una toma de 3 derivaciones deberemos utilizar 3 parches ECG junto a sus respectivos cables.

Es necesario cablear cada uno de ellos a su correspondiente posición, asegurando la correcta polaridad, como indica la Figura 27.

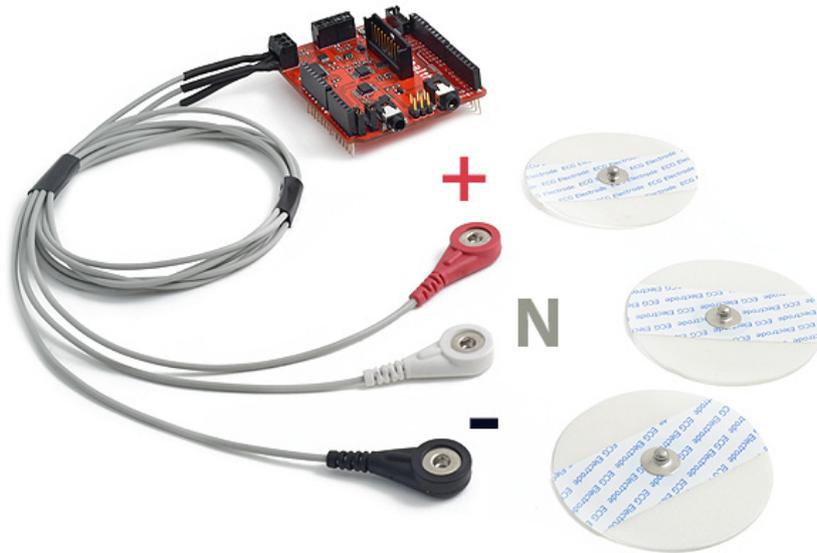


Figura 27 Conexión de los cables de medida de ECG sobre e-Health

Debemos posicionar los 3 parches en las zonas indicadas en la Figura 29, y a su vez conectarlos a los cables de medida como indica la Figura 28. La aplicación de un gel conductor previa colocación de los parches sirve para una mejor medida.

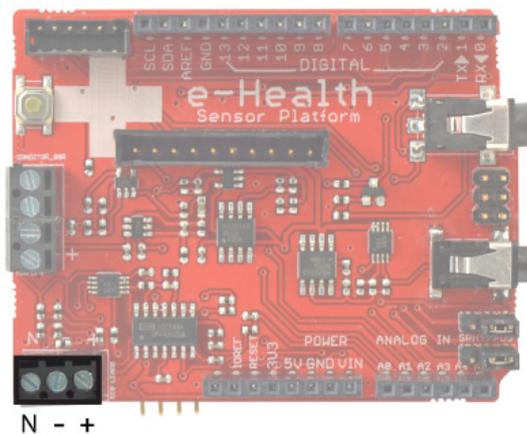


Figura 28 Polaridad de las conexiones del ECG

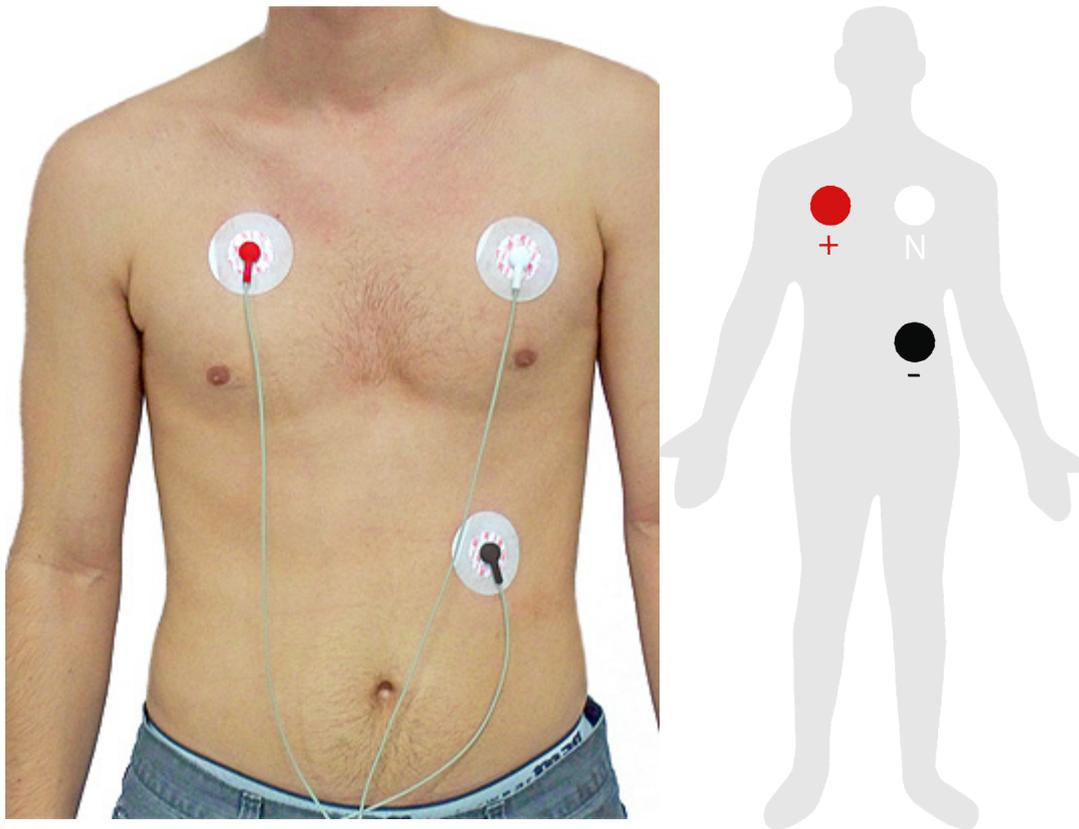


Figura 29 Conexión de los parches sobre el paciente para realizar la medida de ECG

4.2.3 Respiración

Existen diferentes modelos de sensores para la monitorización de la respiración. Los dos más utilizados son el cinturón de esfuerzo respiratorio y los sensores de flujo aéreo.

El cinturón de esfuerzo respiratorio utiliza un sensor piezoeléctrico. Suele ser colocado en el pecho o en el abdomen. Este sensor convierte el movimiento mecánico producido por la respiración en un voltaje oscilante que indica la cadencia de respiración. Esto es así debido a las propiedades de los materiales piezoeléctricos, que al ser sometidos a tensiones mecánicas adquieren una polarización eléctrica en su masa, apareciendo a la salida una diferencia de potencial.

La otra alternativa es la de los sensores de flujo aéreo, que sirven para monitorizar el flujo de aire exhalado. Principalmente, es posible encontrar termistores y termopares para la medida del flujo respiratorio.

Un termopar es una unión de dos metales que produce un voltaje relacionado con una diferencia de temperatura. En nuestro caso, la diferencia de temperatura es dada por el flujo aéreo del paciente. Por otra parte, el termistor se trata de un sensor resistivo en función de la temperatura del aire expulsado. A simple vista son el mismo tipo de sensor, aunque el precio inferior de los termopares los hace más atractivos a la hora de ser integrados.

Dado que los sensores de este tipo suponen unos altos costes, se plantea la fabricación de una cánula propia que permita medir el flujo respiratorio utilizando 2 termopares en serie, formando una termopila, para detectar diferencias de temperatura. Esto abarata el coste notablemente, aunque supone una mayor mano de obra a la hora de la fabricación del dispositivo como se observa en la Figura 30. Para integrar este sensor es necesario circuitería específica de adaptación de señal.



Figura 30 Sensor de flujo respiratorio

Se diseña una cánula de un solo canal oral o nasal reutilizables. Esta dispone de dos puntas de medida que deben ser colocadas en la zona de medición. Un adulto normal lleva a cabo una media de 15-30 respiraciones por minuto.

El sensor ha sido fabricado en resina epoxi que sirve para cubrir y fijar la unión de los dos termopares. Dispone de 2 conectores indicados en la Figura 31, uno positivo y otro negativo, que debe ser conectado de la manera correcta para su funcionamiento.



Figura 31 Polaridad del sensor de respiración

Colocación y uso

Al igual que otros sensores este dispone de dos cables con una determinada polaridad. Es necesario conectar los cables del sensor en la plataforma sirviéndose de los colores que indican la polaridad y de las indicaciones de la Figura 31.

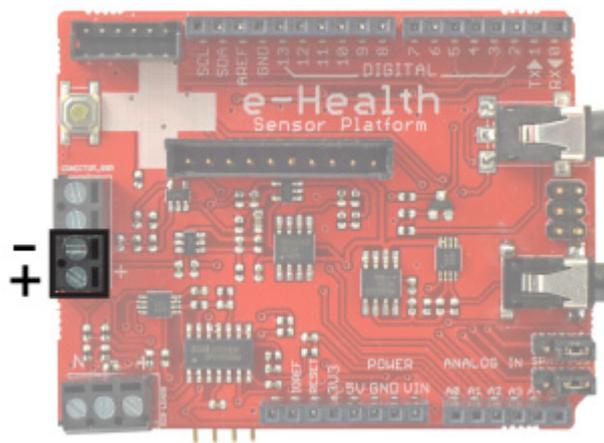


Figura 32 Polaridad de las conexiones del sensor de respiración

El sensor debe colarse en la zona entre la nariz y la boca, como se indica en la Figura 33, para realizar una correcta medición.

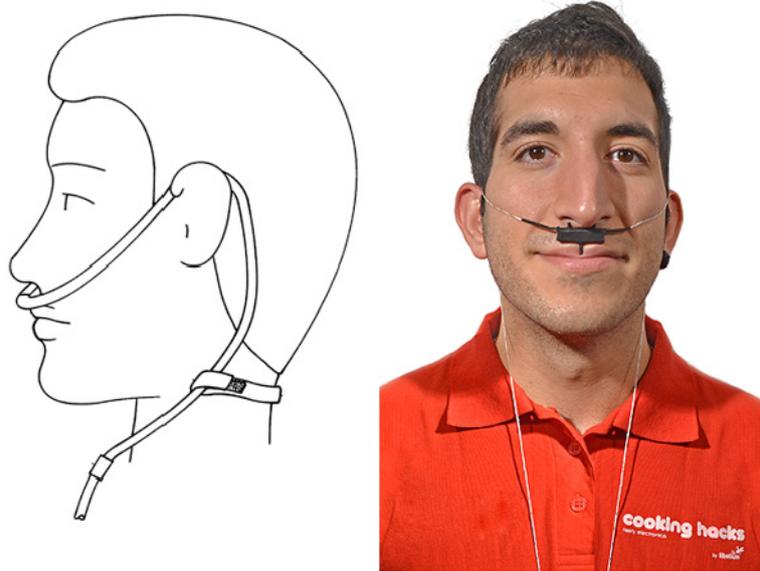


Figura 33 Colocación del sensor de respiración sobre un paciente

4.2.4 Temperatura corporal

Estos sensores son de tipo resistivo, como el de la Figura 34, ya que a su salida aportan una cierta impedancia dependiendo de la temperatura corporal. En principio, no es válido cualquier sensor porque el rango de interés no será ambiental, sino corporal. Se trata de un termistor de tipo NTC cuyo valor resistivo desciende a medida que aumenta la temperatura [28] [29].



Figura 34 Sensor de temperatura corporal

Puede ser utilizado en diversos puntos: tórax, axila, extremidades, yemas de los dedos y dedos del pie. El sensor es fijado mediante una cinta adhesiva para que no se mueva y produzca malas mediciones. Se trata de un sensor ya comercial que conectamos a nuestra placa por medio de un adaptador de tipo Jack.

Colocación y uso

El sensor utilizado necesita un adaptador como el de la Figura 35 del tamaño del Jack para poder ser utilizado sobre la plataforma.

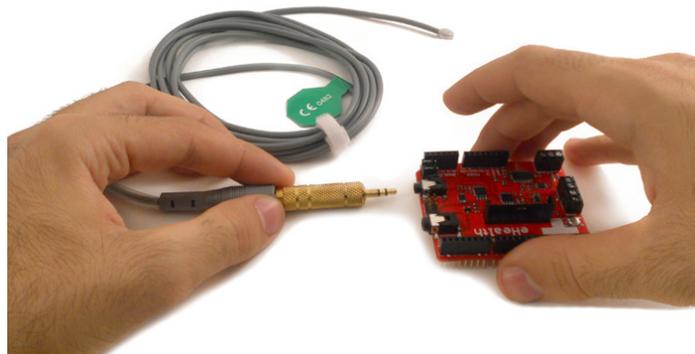


Figura 35 Conexión del sensor sobre la placa de e-Health

Debe colocarse en contacto la parte metálica con nuestra piel en cualquier dedo de alguna de las extremidades humanas al igual que en la Figura 36. Para ello podemos servirnos de un trozo de cinta adhesiva médica.



Figura 36 Colocación del sensor de temperatura corporal sobre el paciente

4.2.5 Tensión Sanguínea

Para el sensor de tensión sanguínea se ha recurrido a la utilización de un dispositivo comercial. Tras un primer estudio en el que se barajó la posibilidad de la implementación del circuito completo de medición, se descartó ya que suponía un gran gasto en investigación y en material (circuito de aire, sensores, adaptación...) [30].

Tras esto se adquirieron numerosos sensores de bajo coste, con el fin de estudiarlos y conseguir integrarlos en el sistema. En un primer estudio se consiguió utilizar la señal directa del sensor de presión interno de uno de los modelos para la medición de los valores de tensión. Para ello era necesaria la apertura del dispositivo y la colocación de un conector que permitiera sacar las señales fuera del dispositivo. El resultado final se puede observar en la Figura 37.



Figura 37 Sensor tensión sanguínea e-Health Versión 1

Tras unos meses de comercialización este sensor tuvo que ser rediseñado debido a problemas con el suministro. Con el fin de facilitar el montaje se realizaron pruebas con dispositivos que ya disponían de una salida de datos.



Figura 38 Sensor de tensión sanguínea e-Health Versión 2

Se recurrió finalmente al dispositivo comercial de la Figura 38 que disponía de una salida de conector USB [31] que transmitía los datos tomados por medio de comunicación serie UART, con las siguientes características:

- Método de medida: Sistema oscilométrico.
- Rango de medida: Presión 0-300 mmHg.
- Precisión: Presión $\leq \pm 3$ mmHg.
- Temperatura de trabajo: 10-40°C.

Colocación y uso

El tensionómetro de la Versión 1 de la plataforma debía colocarse alrededor de la muñeca como se indica en la Figura 39. Es importante mantener la palma hacia arriba y el brazo en la correcta posición alineado con el corazón, para obtener una medida correcta. Se utiliza un cable Jack de 3 polos para transmitir la señal del sensor.



Figura 39 Medición de tensión sanguínea Versión 1

Una vez conectado era necesario presionar el botón de ON para empezar la medición como en el ejemplo de la Figura 40.

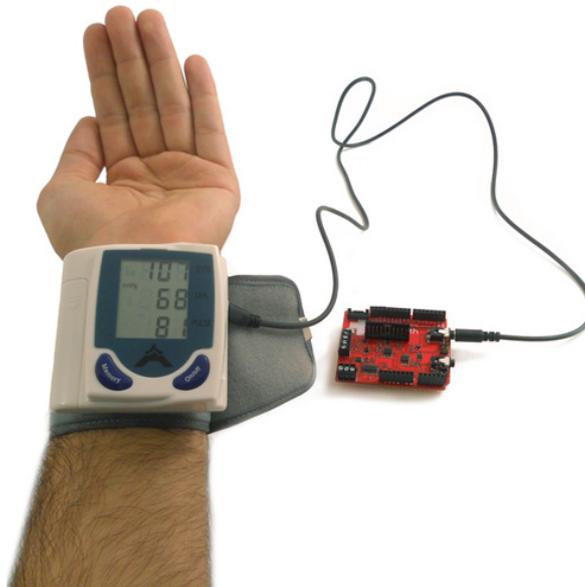


Figura 40 Medida de tensión sanguínea obtenida y almacenada en e-Health

En el caso del sensor de la Versión 2 la utilización es distinta. Antes de conectar el sensor es necesario realizar las mediciones como se indica en la Figura 41. Para ello únicamente debemos situar el brazalete del sensor en la zona del bíceps y encender el dispositivo. Tras unos segundos obtendremos en pantalla la medida.



Figura 41 Medición de tensión sanguínea Versión 2

Una vez tomadas las medidas, debemos conectar el cable adaptador en la disposición de la Figura 42 para poder extraerlas con la plataforma.



Figura 42 Conexión del sensor de tensión sanguínea Versión 2

Para indicar la correcta conexión el dispositivo muestra por pantalla el mensaje “UUU”.

En este caso no se necesita calibración, pero se ha implementado un sistema para hacer compatible este sensor con la versión anterior de e-Health (Figura 43). Como se trata de un sensor que utiliza la UART de Arduino, únicamente debemos utilizar un conector fabricado de manera específica, para cablear las señales en el conector del monitor de la pantalla gráfica como se muestra en la Figura 43.

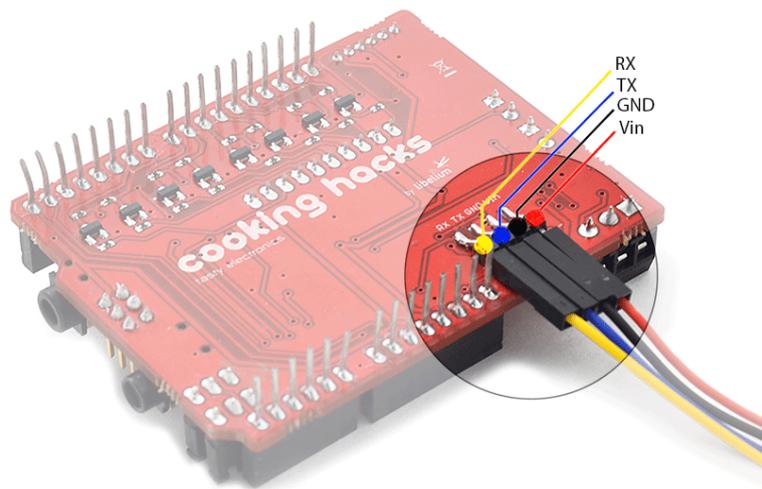


Figura 43 Conexión sensor tensión sanguínea Versión 2 sobre e-Health Versión 1

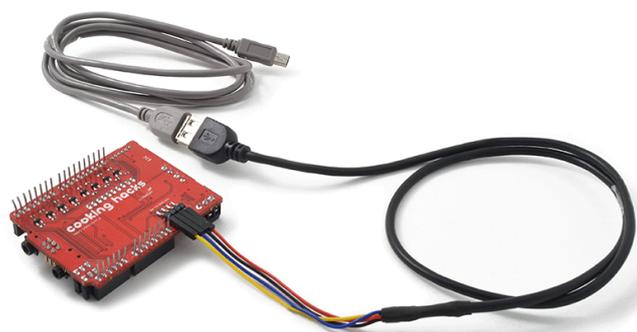


Figura 44 Cable adaptador para hacer compatible el sensor de tensión sanguínea

4.2.6 Posición del paciente

Para este sensor se recurrió al acelerómetro de 3 ejes de la Figura 45 que nos permite obtener en todo momento la posición del paciente, o la detección de caídas.



Figura 45 Acelerómetro utilizado para detectar la posición del paciente

Este acelerómetro [32] nos permite una programación flexible, con una alimentación de únicamente 3.3 voltios. Dispone de una escala ajustable de 2, 4 y hasta 8 g, a la vez que tiene 2 pines de interrupción. Se programa por medio del protocolo I2C. El sensor acabado es el cinturón de pecho de la Figura 46.



Figura 46 Sensor de posición del paciente

Colocación y uso

El sensor de posición dispone de un conector de una sola posición, lo que hace que su conexionado sea muy sencillo como podemos observar en la Figura 47.

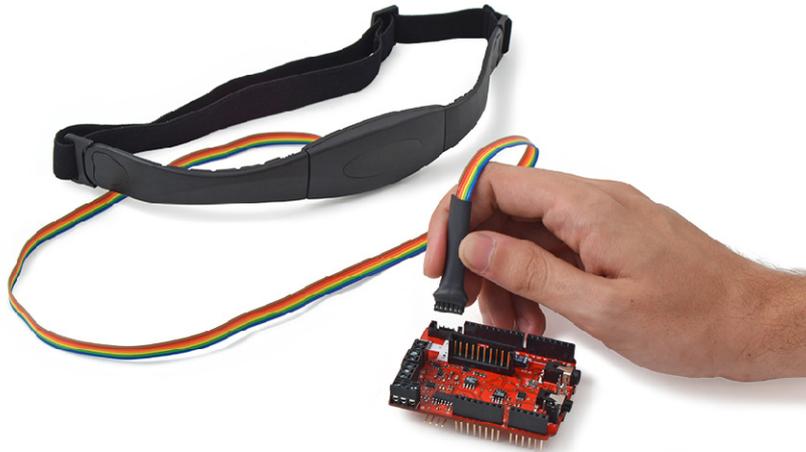


Figura 47 Conexión del sensor de posición sobre la placa e-Health

Una vez conectado podemos conectar la cinta alrededor de nuestro pecho como se observa en la Figura 48, con el cable hacia el suelo.



Figura 48 Colocación del sensor de posición en el paciente

4.2.7 Conductividad de la piel

En general, los electrodos utilizados son de Ag/AgCl y requieren el uso de la pasta de electrodo adecuada [33]. Estos electrodos suelen ser colocados en las falanges intermedias de los dedos índice y corazón. Ha sido necesaria la elección de un par de electrodos con velcro como los de la Figura 49 para su correcta sujeción en las falanges de los dedos índice y corazón. Se utilizan para su conexión a la placa e-Health dos cables similares a los utilizados en el sensor de ECG.



Figura 49 Sensor de conductividad de piel

Colocación y uso

A diferencia de otros sensores, este sensor no necesita una conexión con polaridad específica. Únicamente debemos conectar los cables como se indica en la Figura 50 en la plataforma en su correspondiente conector y colocar los dedos sobre los contactos metálicos.



Figura 50 Conexión de los sensores conductividad de piel

Una vez colocados podemos sujetarlos por medio de las cintas de velcro al igual que en la Figura 51 y la Figura 52. La utilización de un gel conductor ayudaría a una toma más precisa de la medida.

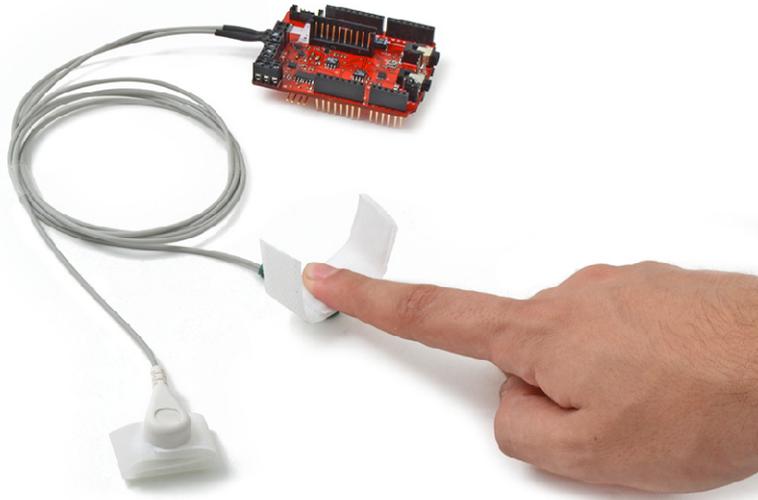


Figura 51 Proceso de colocación del sensor de conductividad de piel

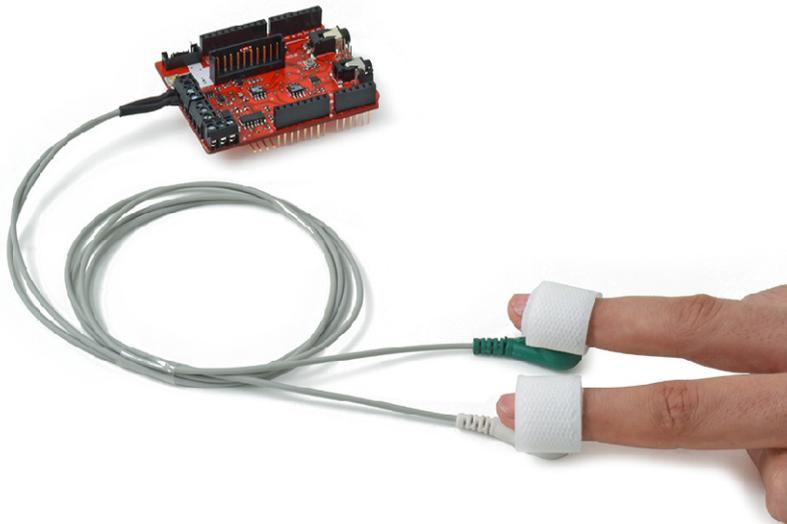


Figura 52 Colocación del sensor de conductividad de piel

4.2.8 Glucosa en sangre

En el caso del sensor de glucosa, se ha recurrido nuevamente al sensor comercial de la Figura 53, que de nuevo dispone de una interfaz para solicitar los datos almacenados. En este caso envía los datos por medio del protocolo 232, por lo que habrá que adaptar la comunicación para que Arduino o Raspberry Pi pueda conectarse a él sin problema.



Figura 53 Sensor de glucosa

Colocación y uso

Antes de conectar el glucómetro a la plataforma es necesario tomar una medida como en la Figura 54. Para ello es necesario conectar una tira de medición. En la pantalla del sensor observaremos una gota parpadeando que indica que se puede realizar la medida.



Figura 54 Realización de una medida de glucosa

Con la ayuda de una lanceta podemos realizar un pequeño orificio en nuestra piel para obtener una muestra de sangre, que colocaremos en la tira de medida.

Tras unos segundos el sensor nos mostrará el valor como en la Figura 55. Para poder obtener los datos del sensor es necesario utilizar el cable específico.



Figura 55 Obtención de la medida de glucosa

Para indicar la correcta conexión el dispositivo muestra por pantalla el mensaje “P-C” como se observa en la Figura 56.



Figura 56 Conexión del sensor de glucosa

4.2.9 Electromiografía

En este caso nos serviremos del mismo sistema que para el sensor de ECG que monitoriza los impulsos del corazón, en cuanto al sensor, para la medición de impulsos musculares. La señal deberá ser adaptada de manera distinta, con el fin de obtener otro tipo de onda, y los parches serán de un tamaño menor para poder colocarlos en cualquier músculo del cuerpo como se observa en la Figura 57.



Figura 57 Sensor de electromiograma

Los parches disponen de unos adhesivos con gel que no irrita la piel, y evita reacciones alérgicas.

Colocación y uso

El proceso de uso de este sensor es similar al de ECG. Primero debemos conectar los tres cables de medida en la plataforma siguiendo la polaridad indicada en la Figura 58.

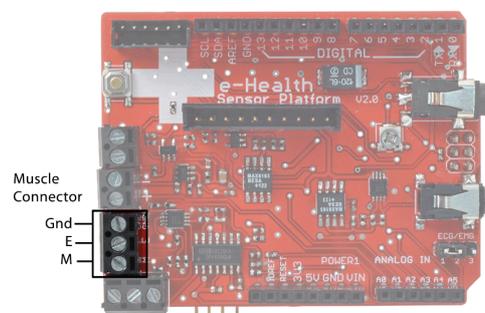


Figura 58 Polaridad de los cables de medida de electromiografía

Para este sensor utilizaremos parches de menos diámetro con el fin de poder monitorizar zonas de pequeño tamaño. Situaremos los parches en tres zonas del músculo: uno en el medio del músculo, otro en el inicio o final, y el último en una zona que no tenga contacto con músculos. Debemos respetar la polaridad de los cables y las conexiones como se observa en la Figura 59 y la Figura 60.

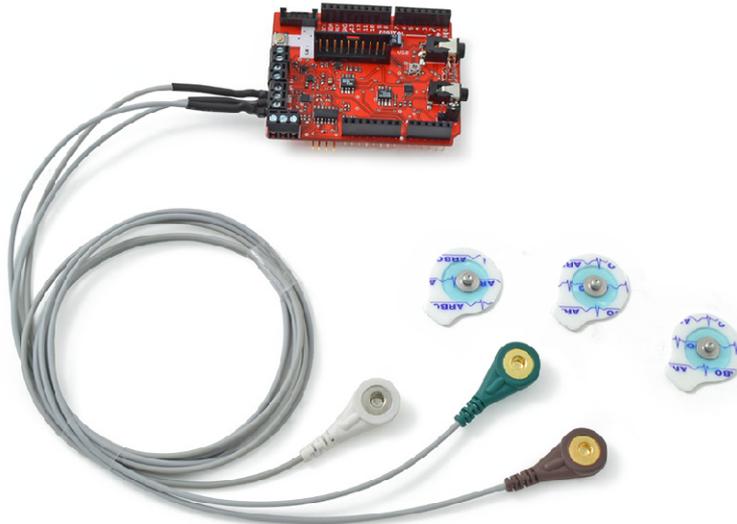


Figura 59 Conexión del sensor de electromiograma

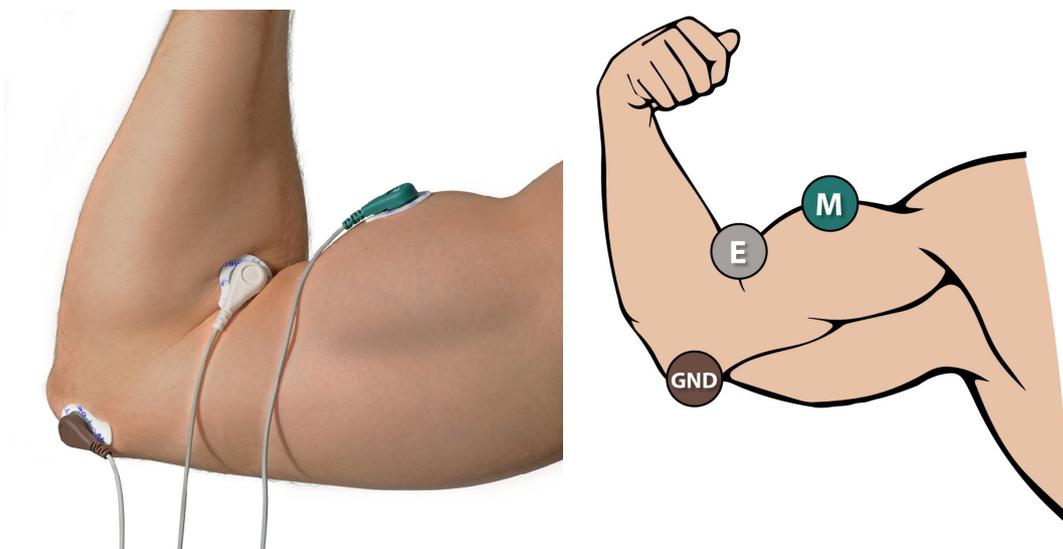


Figura 60 Colocación del sensor de electromiograma sobre el paciente

4.3 Electrónica de adaptación

Se ha realizado una búsqueda de distintos componentes con el fin de encontrar los que mejor encajan en la aplicación cumpliendo los objetivos y especificaciones marcadas de antemano. En este apartado vamos a tratar de analizar todas las posibles soluciones que se han ido planteando a medida que nos acercábamos el diseño final prototipo [34] [35] [36].

Para todas las decisiones se han seguido una serie de pautas, basadas en unos criterios económicos, sociales y tecnológicos mínimos según el punto de vista del diseñador [37] [38]. A continuación se presentan los temas que más influyen en la toma de decisiones, y el criterio que se ha aplicado en cada uno de los posibles casos:

- Versatilidad: una gran capacidad de cambio, sirviendo para multitud de situaciones y siendo útil para distintos tipos de aplicaciones. Se ha intentado siempre escoger los componentes más versátiles, o la solución que más rango de utilidad abarca.
- Precio: el valor de los componentes o lo que cuesta implementar cada una de las soluciones. Las soluciones que se han intentado escoger siempre han sido las más baratas para así reducir el precio final del diseño, pero sin dejar a un lado la calidad.
- Tamaño: las dimensiones del diseño una vez finalizado el producto, o simplemente la disposición de los componentes en su interior. En cuanto a dimensiones y las soluciones que a esto afectan, se ha seguido la pauta de buscar un tamaño que se adapte a la tecnología de base que utiliza el módulo.
- Prestaciones: el rendimiento o las características de un sistema. Las posibilidades que le ofrece al usuario. Se ha buscado que el producto final tenga unas buenas prestaciones, tanto de rapidez, información y fiabilidad, como de estética y solidez.

- Sencillez: en cuanto a diseño de la PCB, sin un número excesivo de componentes, como en lo que respecta a la sencillez de presentación y disminución de tamaño.
- Consumo y potencia: lo que el dispositivo consume, y la potencia disipada por los componentes más relevantes. La potencia no es algo demasiado relevante en este tipo de diseños, pero se ha buscado reducir al máximo el consumo.
- Estética: que el diseño final tanto desde el punto de vista del diseño electrónico, como el de presentación cara al público, sea lo suficientemente organizado y elegante. Hemos intentado que el diseño electrónico sea lo más ordenado posible con el fin de facilitar posibles reparaciones.

4.3.1 Sensores

En los próximos apartados será explicada brevemente la electrónica de adaptación diseñada para cada sensor.

Pulso y oxígeno en sangre

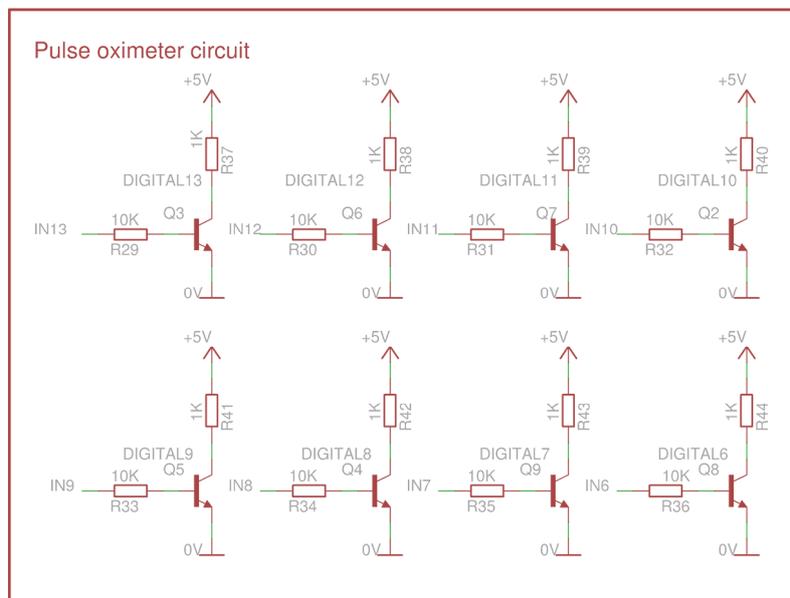


Figura 61 Circuito de adaptación del sensor de SpO2

Para este sensor no es necesaria una electrónica de adaptación específica por el tipo de sensor, sino que debe adaptarse a la circuitería interna del dispositivo comercial. Se utiliza etapas transistorizadas para detectar que valores está representando por pantalla el dispositivo, utilizando interrupciones como sistema de detección. En la Figura 61 podemos observar el circuito relacionado con este sensor.

El circuito utilizado para la medición con el pulxiómetro se basa en una lógica inversa de transistores. Dado que las señales que el sensor nos proporciona son de 2 voltios y no pueden ser detectadas directamente por nuestro Arduino, se han utilizado transistores a cuya base se llevan las múltiples señales del display del sensor. Esto nos permite conmutar la tensión de 5 voltios a entradas digitales y detectar que diodos del display 7 segmentos se encuentran encendidos.

Electrocardiograma

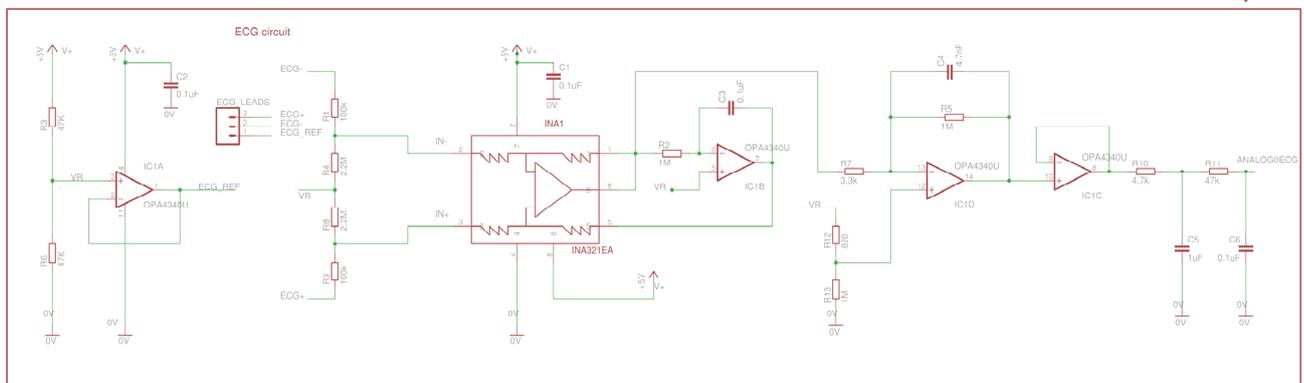


Figura 62 Circuito de adaptación del sensor de electrocardiograma

La electrónica de adaptación necesaria, ha sido basada en las especificaciones técnicas de un amplificador de instrumentación. Este dispositivo es el encargado de amplificar la señal diferencial de entrada proveniente de sendos electrodos izquierdo y derecho. Se utilizan amplificadores operacionales para completar las distintas etapas necesarias para la medición de la señal de ECG, siguiendo circuitos de ejemplo propios de los componentes. En la Figura 62 podemos observar el circuito relacionado con este sensor.

Respiración

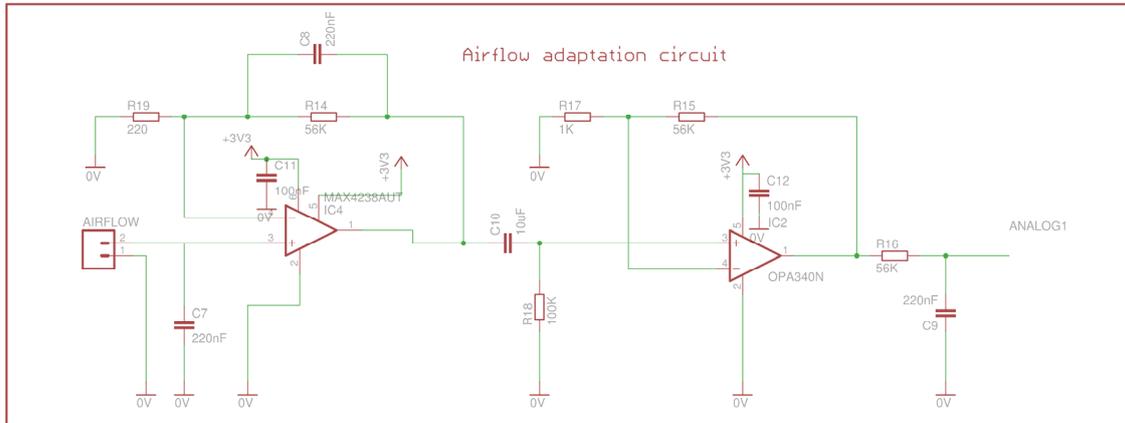


Figura 63 Circuito de adaptación del sensor de respiración

Se planteó una primera opción de circuito de amplificación utilizando amplificadores de instrumentación de bajo ruido, pero suponía un coste muy elevado. Por esto mismo se utiliza un amplificador operacional de bajo ruido, seguido por varios filtros de paso alto y bajo para discriminar las frecuencias no deseadas. Esto se completa con una última etapa de amplificación. En la Figura 63 podemos observar el circuito relacionado con este sensor. La última etapa de amplificación tiene unas restricciones menores de ruido, por lo que se utiliza un operacional de menor precio, tensión de offset adecuada y capaz de dar la ganancia requerida.

Temperatura corporal

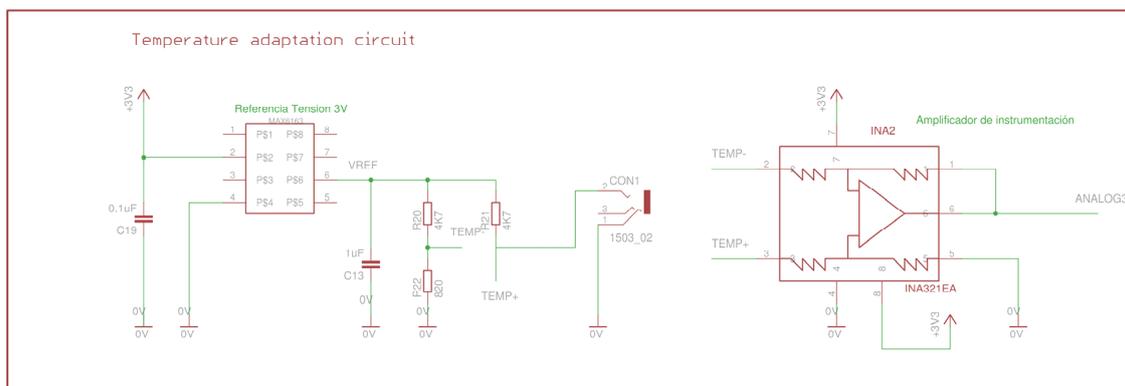


Figura 64 Circuito de adaptación del sensor de temperatura

El diseño escogido para el sensor de temperatura se ha basado en el puente de Wheatstone. El puente de Wheatstone ha sido diseñado para cubrir el rango de temperaturas de interés: entre 25°C y 50°C. En la Figura 64 podemos observar el circuito relacionado con este sensor.

La tensión de salida diferencial del puente de Wheatstone es amplificada y referenciada a tierra mediante un amplificador de instrumentación.

Tensión sanguínea

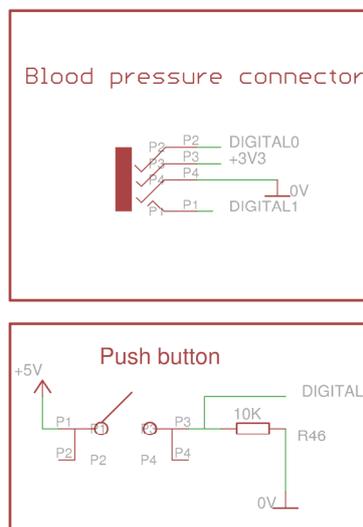


Figura 65 Circuito de adaptación del sensor de tensión sanguínea

Para este sensor no es necesario ningún tipo de adaptación, ya que únicamente dispone de conexión directa a los pines 0 y 1 de Arduino o Raspberry Pi, donde se encuentra disponible la comunicación UART serie. En la Figura 65 podemos observar el circuito relacionado con este sensor.

Posición del paciente

Al igual que en el caso anterior este sensor únicamente necesita conexión a las líneas del I2C del micro controlador, dado que al ser una placa comercial de un acelerómetro ya dispone de la electrónica de adaptación integrada.

Conductividad de piel

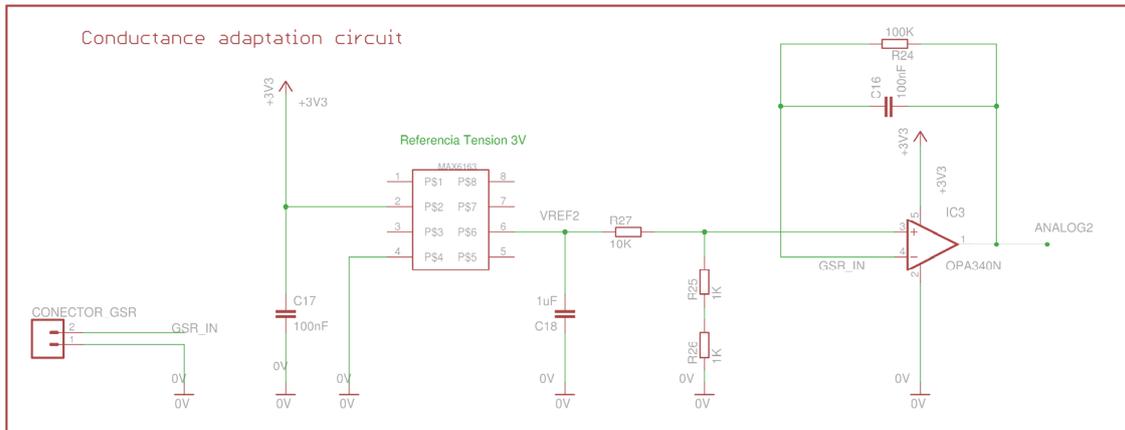


Figura 66 Circuito de adaptación del sensor de conductividad

Se ha tratado de medir la intensidad que discurre entre dos electrodos colocados en los dedos índice y corazón de la mano. El esquema se basa en un amplificador operacional que amplifica esa intensidad que es del orden de μ Amperios. Para que exista esa conducción entre ambos dedos, se aplican 0.5 voltios entre los electrodos. Mediante el conocimiento de la intensidad y el voltaje existente entre ambos electrodos, es posible deducir la resistencia, y por lo tanto la conductancia de la piel, mediante la ley de Ohm. En la Figura 66 podemos observar el circuito relacionado con este sensor.

Glucosa en sangre

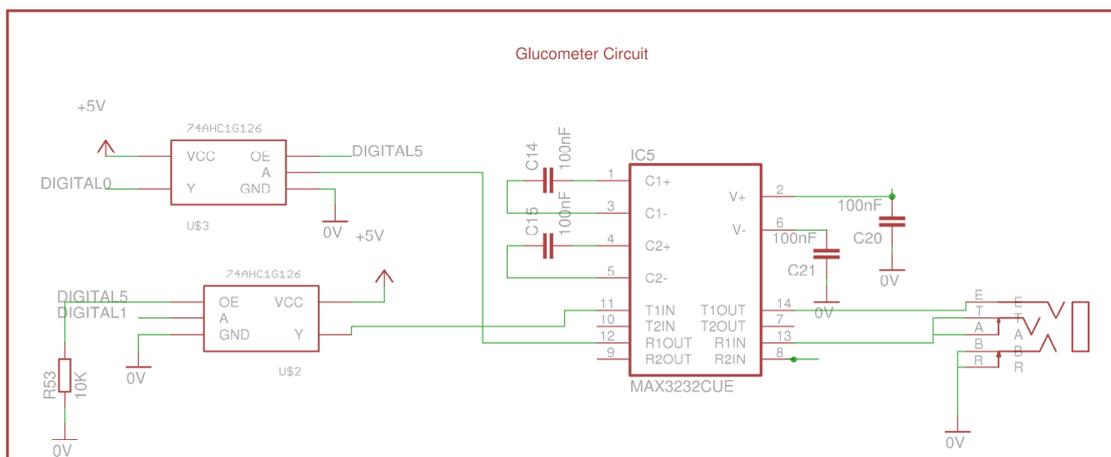


Figura 67 Circuito de adaptación del sensor de glucosa

Este sensor necesita de una etapa de adaptación del protocolo 232 a UART que es el protocolo que tenemos disponible en nuestras plataformas. Para poder tener conectado el dispositivo sin que afecte a la programación del micro controlador se ha implementado una etapa con interruptores digitales, para poder habilitar la lectura de sensor solo en el momento preciso. En la Figura 67 podemos observar el circuito relacionado con este sensor.

Electromiograma

Para este sensor nos basamos en una etapa ya implementada de medición de señales EMG. Para ello lo primero que tenemos que obtener es alimentación bipolar. Una vez realizado esto, alimentamos un amplificador que nos proporciona la señal medida de los impulsos musculares. Debe ser rectificadora y adaptada para poder obtener una señal analógica útil para nuestro sistema. En la Figura 68 podemos observar el circuito relacionado con este sensor.

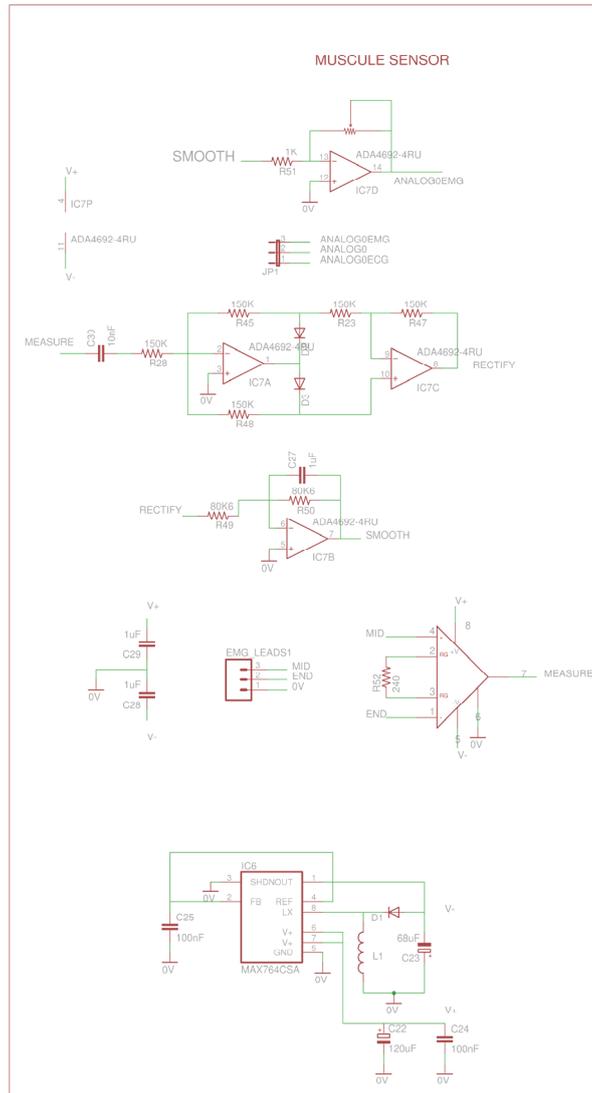


Figura 68 Circuito de adaptación del sensor de electromiograma

4.3.2 Otros circuitos

Para todos ellos se han elegido conexiones sencillas por medio de conectores directos sobre la placa. En la Figura 69 podemos observar las distintas conexiones extra.

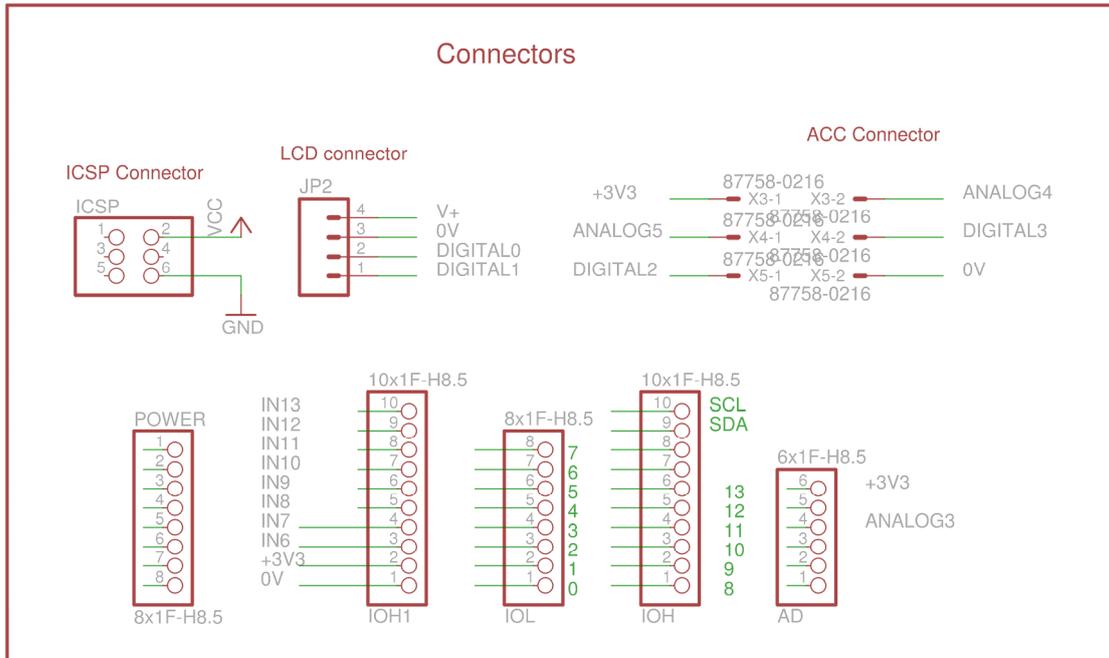


Figura 69 Circuitos y conectores destinados a otras aplicaciones

Se han dejado libres diversos pines con el fin de poder realizar ejemplos combinados con otras tecnologías. También se han habilitado conectores para conectar algunos periféricos, como por ejemplo una pantalla gráfica.

4.4 Prototipos de placas de circuito impreso (PCB)

Partiendo de la información proporcionada por los fabricantes de los distintos sensores y los conocimientos adquiridos de la investigación [39], se realizó el diseño del circuito impreso de la placa. Se han seguido una serie de pautas para su diseño, con el fin de evitar posibles problemas de diseño o fabricación:

- Plantear el diseño en papel o en ordenador, y si es posible simularlo para ver si todo está correcto.
- Estudio intensivo de los circuitos del *datasheet* con el fin de encontrar la configuración que se adapta mejor a nuestra aplicación.
- Cumplir las normas básicas de ruteado de circuitos (distancia entre pistas, tamaño de pistas...)
- Fijar unas dimensiones de base sobre las que trabajar.
- Una buena elección de componentes, documentada, para evitar problemas futuros.

Como resultado de seguir todas estas premisas y el desarrollo de los circuitos necesarios para la adaptación se han obtenido los circuitos necesarios para trabajar con todos los sensores y con otros dispositivos y tecnologías. En la Figura 70 podemos observar su evolución. Todos los circuitos finales los podemos observar en el esquemático de la Versión 2 en la Figura 71.

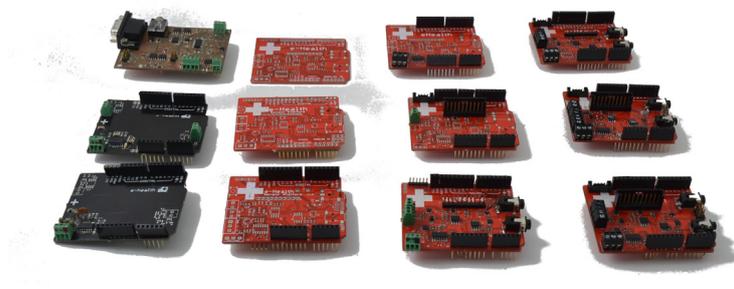


Figura 70 Evolución de las distintas placas de prototipado en todo el desarrollo

eHealth sensor platform schematic

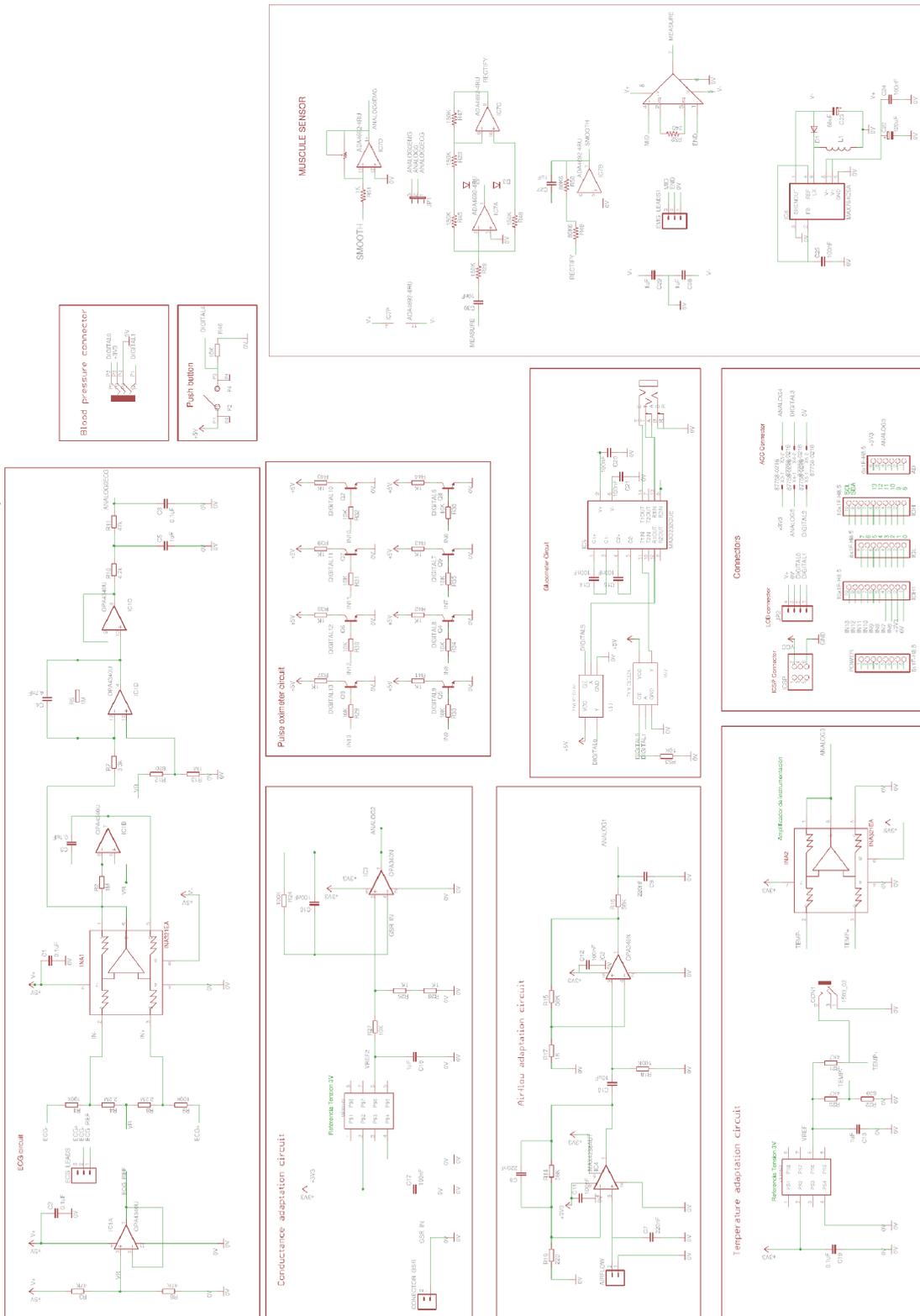


Figura 71 Esquemático definitivo de la placa e-Health Versión 2

4.4.1 Placa e-Health Versión 1

Finalmente fue diseñada una única placa para integrar todos los sensores biométricos testeados que podemos observar en la Figura 72.

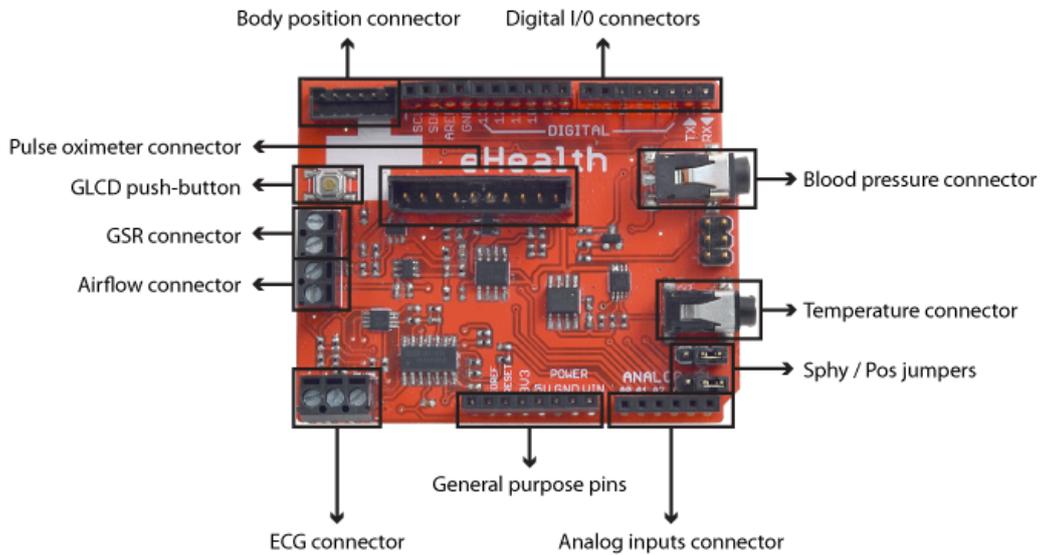


Figura 72 Esquema y conexionado de e-Health Versión 1

La primera versión final de la placa se hizo compatible tanto con Arduino directamente, como con Raspberry Pi por medio de una placa adaptadora que permite integrar placas para Arduino sobre Raspberry Pi (Figura 73).

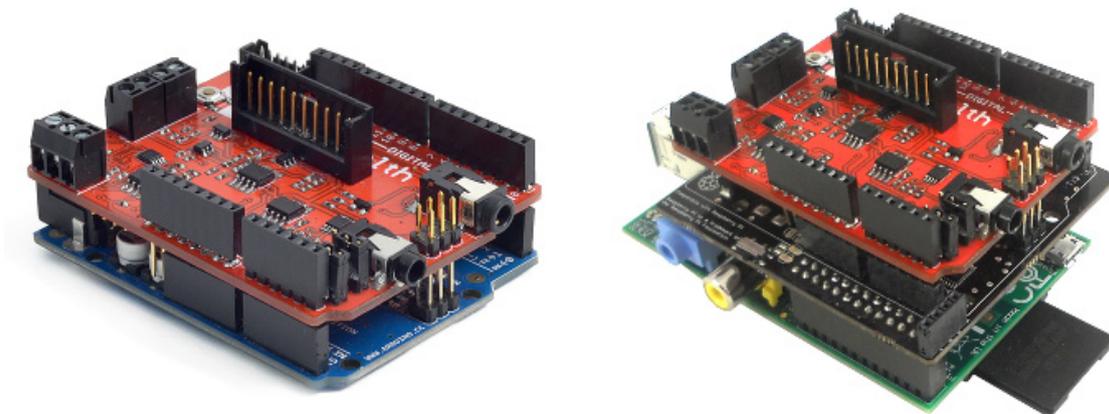


Figura 73 e-Health Versión 1 sobre Arduino y Raspberry Pi

Finalmente las características de esta primera versión fueron (Figura 74):

- 7 sensores no invasivos y 1 invasivo.
- Almacenamiento y uso de los datos de glucosa.
- Monitorización de la señal ECG.
- Control del flujo respiratorio del paciente.
- Datos de temperatura corporal.
- Medición de respuesta galvánica de la piel.
- Detección de la posición corporal.
- Funciones de pulso y oxígeno en sangre.
- Control de tensión sanguínea..
- Múltiples posibilidades de visualización de datos.
- Compatibles con dispositivos que funcionen por UART.



Figura 74 Conexión final de todos los sensores sobre e-Health Versión 1

4.4.2 Placa e-Health Versión 2

En agosto de 2013 se lanzó la nueva versión de la placa gracias a mejoras del producto y a opiniones de usuarios de la comunidad que han utilizado este dispositivo. Las principales mejoras integradas son:

- Sensor de musculatura EMG.
- Nuevo sensor de tensión sanguínea, debido a problemas con el anterior suministrador.
- Actualización del sensor de glucosa, ya que se integra el circuito de adaptación en la propia placa.
- Mejora de las posibilidades de conexión.

Todas estas mejoras se pueden observar en los distintos archivos de diseño y esquema de la plataforma en la Figura 75, la Figura 76 y la Figura 77.



Figura 75 Conexión final de todos los sensores sobre e-Health Versión 2

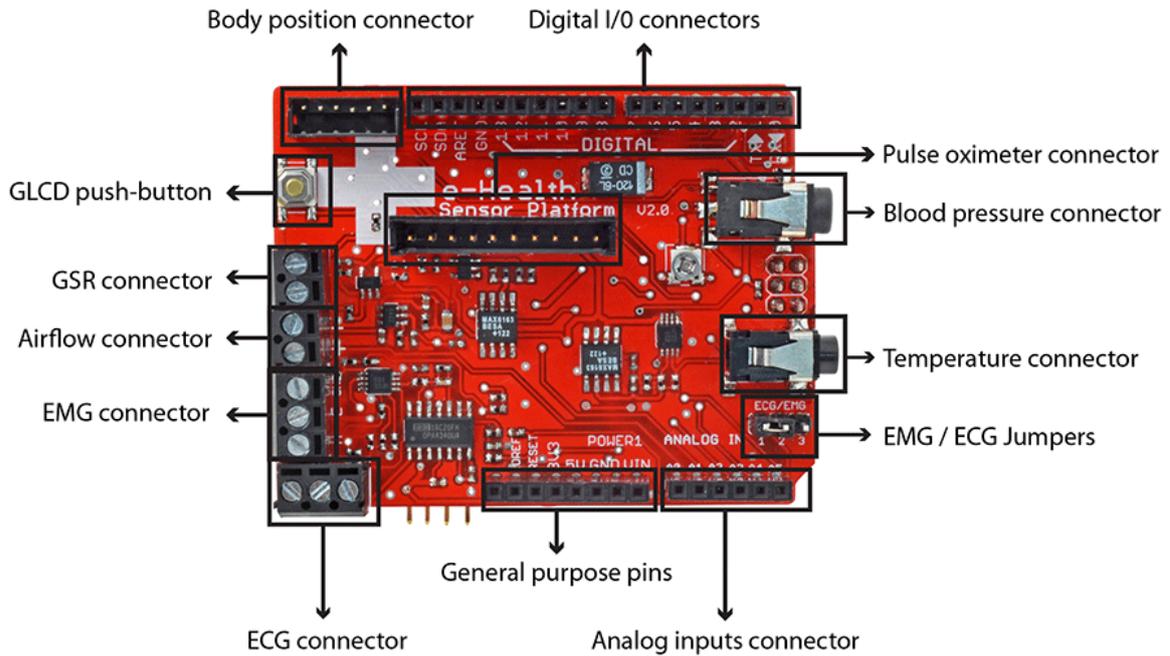


Figura 76 Esquema y conexionado de e-Health Versión 2

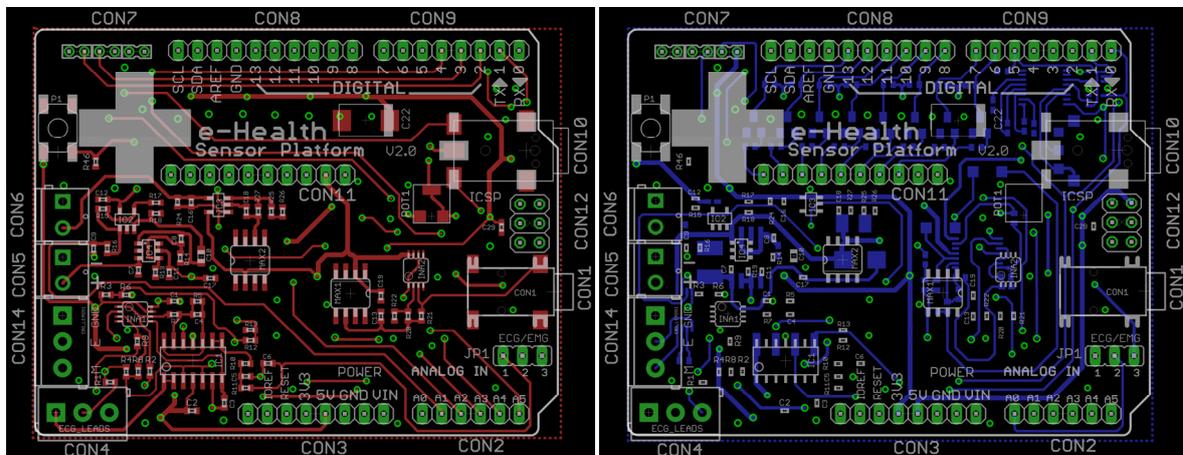


Figura 77 Layout definitivo de la placa e-Health Versión 2

5 Diseño de software

Para la implementación del firmware, Libelium pone a disposición de los usuarios multitud de librerías que ayudan a programar de forma bastante cómoda y una interfaz donde se explica cada función y variable.

Para cada placa, ya fuera de prototipado, placa intermedia o placa definitiva, se ha desarrollado el código apropiado, que ha sido comprobado y depurado hasta conseguir el definitivo. La implementación del código ha supuesto una parte muy importante del presente proyecto, puesto que se invirtió alrededor del 30% del tiempo en esta labor.

Sin embargo, no se explicará en esta memoria cada código ya que es muy extenso: el código para el módulo tiene múltiples líneas y centenas de variables definidas.

Los códigos relevantes se encuentran en el documento anexos, junto a distintos códigos de ejemplo. Bajo normativa de la empresa cada variable y línea están explicadas en inglés, por lo que resulta bastante sencillo entender la función que realiza cada una de ellas.

Todo el código generado tiene licencia GPL (general public license) por lo que cualquier persona tiene derecho a acceder al código, estudiarlo y modificarlo.

Para la implementación del código se han seguido los siguientes criterios:

- Hacer un código rápido. Se evitan bucles o instrucciones innecesarias.
- Limitar el número de variables, ya que la memoria del microcontrolador no es mucha.
- Conseguir una gran precisión de medición y funciones autónomas.

5.1 Librería e-Health

La librería para esta plataforma está desarrollada buscando resultar de fácil utilización al usuario final, sin necesidad de conocer los componentes con los que ha sido diseñada. Está implementada en lenguaje C-C++ íntegramente, y en una arquitectura orientada a objetos, siguiendo el modelo de las demás librerías de Arduino.

La librería engloba el conjunto de subprogramas que sirven para el funcionamiento correcto del software. Está compuesta por dos archivos: ehealth.h, la cabecera de la librería, en donde se definen todas las variables y funciones a utilizar, y sus propiedades (públicas, privadas...), y ehealth.cpp, que es el objeto de la librería, en la que se programan cada una de las funciones definidas en el otro archivo.

Esta librería tiene funciones implementadas para facilitar la lectura de todos los sensores, con el fin de poder enviarla por medio de cualquier protocolo de comunicaciones inalámbricas. Ofrece un sistema libre y sencillo de uso.

Para asegurar la compatibilidad entre ambas plataformas utilizando prácticamente los mismos códigos se utiliza la librería ArduPi para Raspberry Pi que permite a los desarrolladores implementar códigos en el mismo lenguaje que para Arduino. Para hacerlo totalmente compatible tanto en software como en hardware, utilizaremos la placa “puente” adaptadora, desarrollada por Libelium, que permite la interconexión directa de placas diseñadas para Arduino, sobre Raspberry Pi.

5.1.1 Arduino

La librería básica ha sido implementada para trabajar con Arduino e incorpora todas las funciones necesarias para controlar la plataforma e-Health. Antes de empezar a utilizar la placa es necesario descargar la librería e incluirla en el IDE de programación. La placas necesarias para trabajar con Arduino se observan en la Figura 78.

En los siguientes apartados estudiaremos las funciones y los códigos de ejemplo implementados para cada uno de los sensores sobre Arduino, igualmente aplicables para Raspberry Pi.

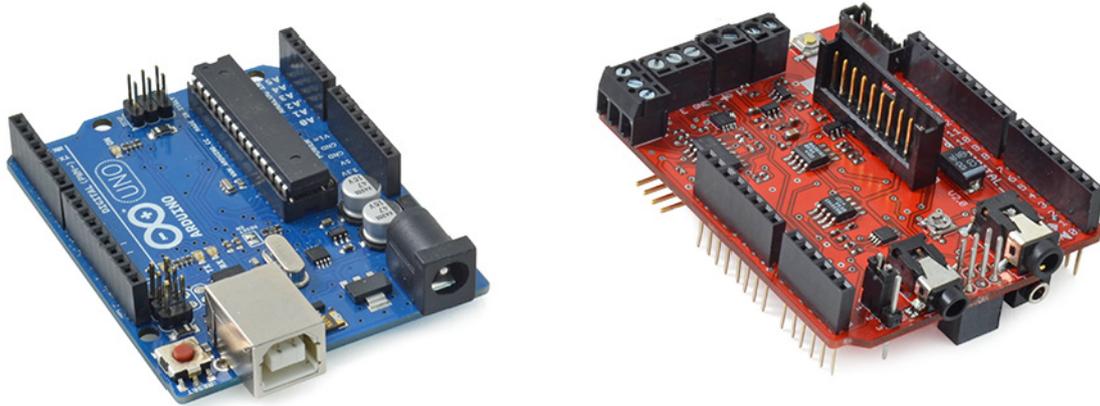


Figura 78 Placa Arduino y placa e-Health

5.1.2 Raspberry Pi

En el caso de Raspberry Pi se han adaptado las funciones básicas de la librería a ArduPi, para permitir la compatibilidad de las plataformas. Debemos incluir la librería en la misma carpeta que la librería ArduPi. La librería nos permite escribir programas similares a los de Arduino, habilitando diversas funciones básicas en Raspberry Pi: I2C, UART, SPI, GPIOs... La placas necesarias para trabajar con Raspberry Pi se observan en la Figura 79.

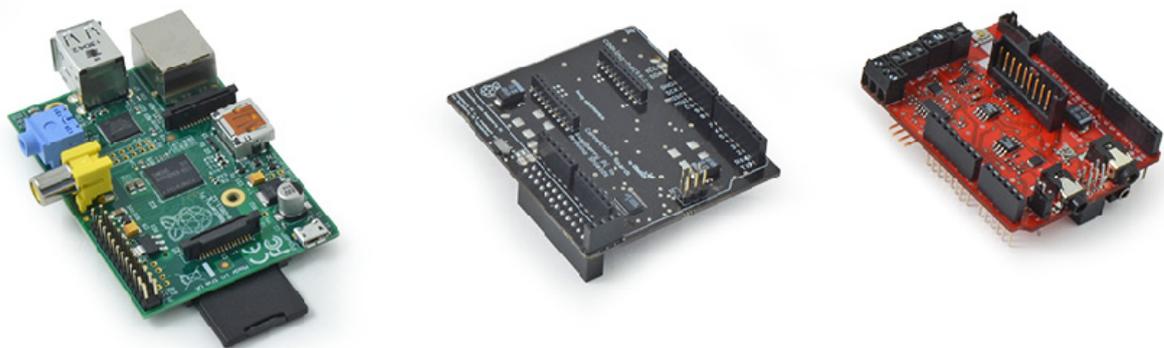


Figura 79 Placa Raspberry Pi, placa adaptadora y placa e-Health

arduPi es una librería en C++ que permite escribir programas para Raspberry Pi como si estuvieras trabajando con Arduino. Tiene disponibles todas las funciones de comunicación básicas como: UART, I2C, SPI. Además incluye GPIOs controlables.

Una vez creamos el programa básico, deberemos utilizar esta estructura incluyendo la librería de eHealth, y programar en la sección indicada:

```

/Include eHealth library (it includes arduPi)
#include "eHealth.h"

/*****
 * IF YOUR ARDUINO CODE HAS OTHER FUNCTIONS APART FROM *
 * setup() AND loop() YOU MUST DECLARE THEM HERE *
 * *****/

/*****
 * YOUR ARDUINO CODE HERE *
 * *****/

int main (){
    setup();
    while(1){
        loop();
    }
    return (0);
}

```

Tras esto solo es necesario compilar el código en Raspberry Pi desde el terminal por medio de comandos, y ejecutarlo.

```

g++ -c arduPi.cpp -o arduPi.o

g++ -c eHealth.cpp -o eHealth.o

g++ -lpthread -lrt user-e-health-app.cpp arduPi.o eHealth.o -o user-e-health-app

sudo ./user-e-health-app

```

5.2 Firmware asociado a los sensores

Todos los programas y librerías implementadas para Arduino se dividen en 3 partes: estructura, valores (variables y constantes), y funciones.

Se han implementado una serie de programas de ejemplo sencillos con el fin de permitir mostrar el uso de las distintas funciones de la librería al usuario, permitiéndole así usar los programas o modificarlos para sus propias experiencias. En este apartado voy a mostrar el diagrama de flujo de alguno de ellos, las funciones que rigen cada sensor y extractos de los distintos códigos. En el Anexo B pueden consultarse los códigos en más profundidad.

5.2.1 Funciones

Pulso y oxígeno en sangre

Estas son las funciones implementadas para este sensor:

```
initPulsioximeter() // It initialize the pulsioximeter sensor.  
readPulsioximeter() // It reads a value from pulsioximeter sensor.  
getBPM() // Returns the heart beats per minute.  
getOxygenSaturation() // Returns the oxygen saturation in blood in percent.
```

Este sensor funciona por interrupciones y es necesaria una librería especial ya implementada para que funcione correctamente. Esto nos evita tener que programar partes de código que regulen las interrupciones.

```
#include <PinChangeInt.h >
```

Tras incluir esto, debemos habilitar las interrupciones en nuestro código para que se puedan actualizar los valores medidos.

```
PCintPort::attachInterrupt(6, readPulsioximeter, RISING);
```

El pin digital 6 será el encargado de enviarnos la señal de que se está realizando la medida correctamente.

```
void readPulsioximeter(){  
    cont ++;  
    if (cont == 50) { //Get only one 50 measures to reduce the latency
```

```
eHealth.readPulsioximeter();
cont = 0;
}
}
```

El sensor dispone de un botón que debe ser presionado para comenzar a realizar la medida. Para inicializar las variables del sensor y realizar mediciones tenemos las siguientes funciones presentadas en un ejemplo de aplicación:

```
{
int SPO2 = eHealth.getOxygenSaturation()
int BPM = eHealth.getBPM()
}
```

En la Figura 80 se puede observar la salida del monitor serie en el ejemplo básico.

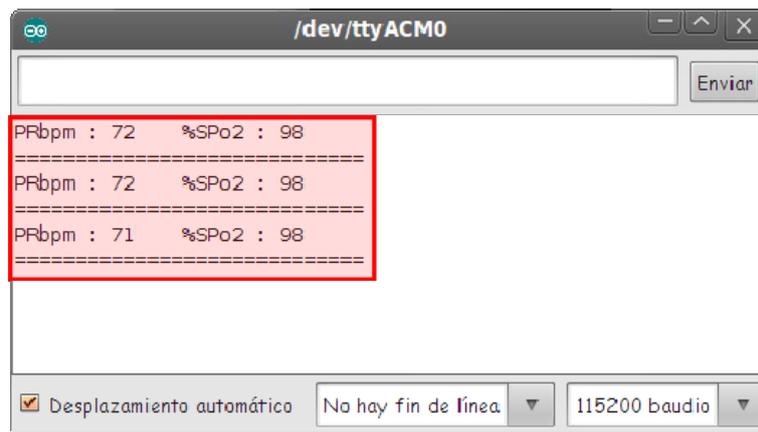


Figura 80 Visualización de la medida de SpO2 por medio del monitor serie de Arduino

Electrocardiograma

Estas son las funciones implementadas para este sensor:

```
getECG() // Returns an analogic value to represent the Electrocardiography.
```

El sensor de ECG devuelve un valor analógico de entre 0 y 5 voltios que puede ser representado en una onda. Únicamente es necesario leer el pin analógico al que se encuentra conectado.

```
{
float ECGvolt = eHealth.getECG();
}
```

Respiración

Estas son las funciones implementadas para este sensor:

`getAirFlow()` // Returns an analogic value to represent the air flow.

`airFlowWave()` // Prints air flow wave form in the serial monitor.

El sensor de respiración se conecta también a una entrada analógica devolviendo un valor de 0 a 1024 en función de la temperatura del aire expulsado e inspirado por el paciente. Se ha implementado una función que representa por el monitor serie disponible en Arduino, estos valores en formato gráfica.

```
{  
  int airFlow = eHealth.getAirFlow();  
  eHealth.airFlowWave(air);  
}
```

En la Figura 81 se puede observar la salida del monitor serie en el ejemplo básico.

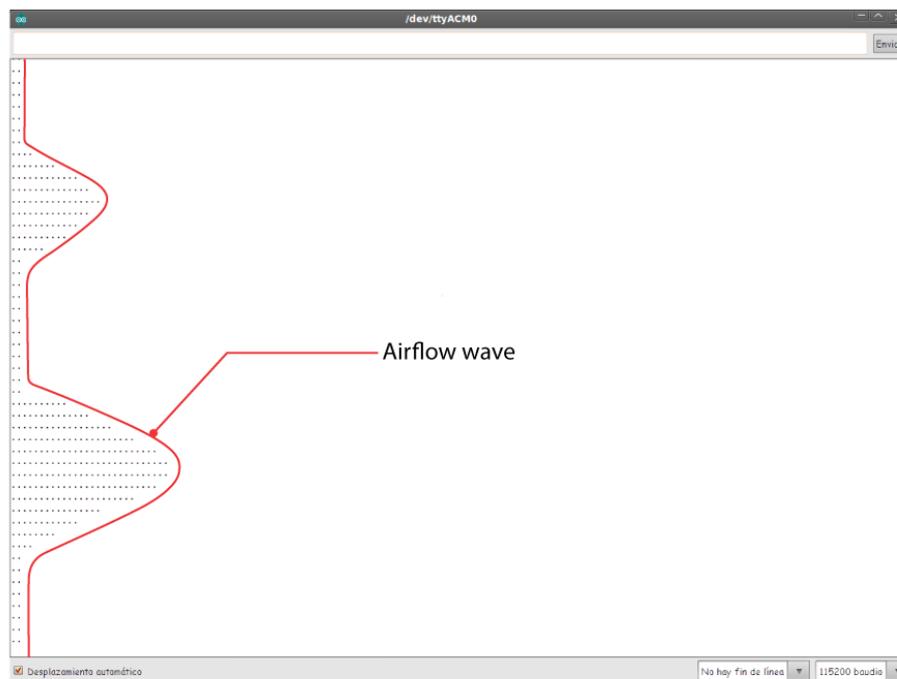


Figura 81 Visualización de la medida de respiración por medio del monitor serie de Arduino

Temperatura corporal

Estas son las funciones implementadas para este sensor:

```
getTemperature() // Returns the corporal temperature.
```

El sensor de temperatura corporal solo dispone de una función muy sencilla encargada de una medida analógica. A partir de esa tensión, es posible calcular la resistencia mostrada por la sonda de temperatura según el circuito diseñado. Por medio de una regresión matemática relacionada con los valores del datasheet del dispositivo se devuelve el valor de temperatura.

```
{  
    float temperature = eHealth.getTemperature();  
}
```

En la Figura 82 se puede observar la salida del monitor serie en el ejemplo básico.



Figura 82 Visualización de la medida de temperatura por medio del monitor serie de Arduino

Tensión sanguínea Versión 1

Este sensor fue implementado para la Versión 1 de la plataforma, aunque tuvo que ser retirado por falta de suministro. Ambos modelos tienen protocolos de lectura e integración muy distintos.

Algunos de los parámetros deben ser inicializados. En este caso únicamente utilizamos la medida directa del sensor sin adaptar, y realizamos una calibración por software para obtener la medida. Una vez realizada la adaptación se puede correlacionar los valores obtenidos en la pantalla con los obtenidos por el sensor para calibrar el sensor.

Tensión sanguínea Versión 2

Estas son las funciones implementadas para este sensor:

```
initBloodPressureSensor() // It initialize and measure the blood pressure sensor.  
getBloodPressureSensor() // Returns the number of data stored in the blood pressure sensor.  
getSystolicPressure(i) // Returns the value of the systolic pressure number i.  
getDiastolicPressure(i) // Returns the value of the diastolic pressure number i.
```

Para actualizar algunos parámetros básicos para la medida de este sensor se ha implementado una función de inicialización. La misma función que lo inicializa, establece conexión con el dispositivo y solicita los datos disponibles en la memoria. Al comunicarse con él por protocolo UART, necesita estar conectado por medio de los cables pertinentes y con la señal de conexión correcta en el dispositivo.

```
{  
    eHealth.readBloodPressureSensor();  
    Serial.begin(115200);  
}
```

Se ha implementado una función que devuelve la cantidad de datos disponibles en el dispositivo:

```
{  
    uint8_t numberOfData = eHealth.getBloodPressureLength();  
    Serial.print(F("Number of measures : "));  
    Serial.println(numberOfData, DEC);  
}
```

```
    delay(100);
}
```

Las siguientes funciones devuelven los valores de tensión sistólica y diastólica almacenados en variables privadas de la clase ehealth.

```
{
    int systolic = eHealth.getSystolicPressure(1);
    int diastolic =eHealth.getDiastolicPressure(1);
}
```

En la Figura 83 se puede observar la salida del monitor serie en el ejemplo básico.

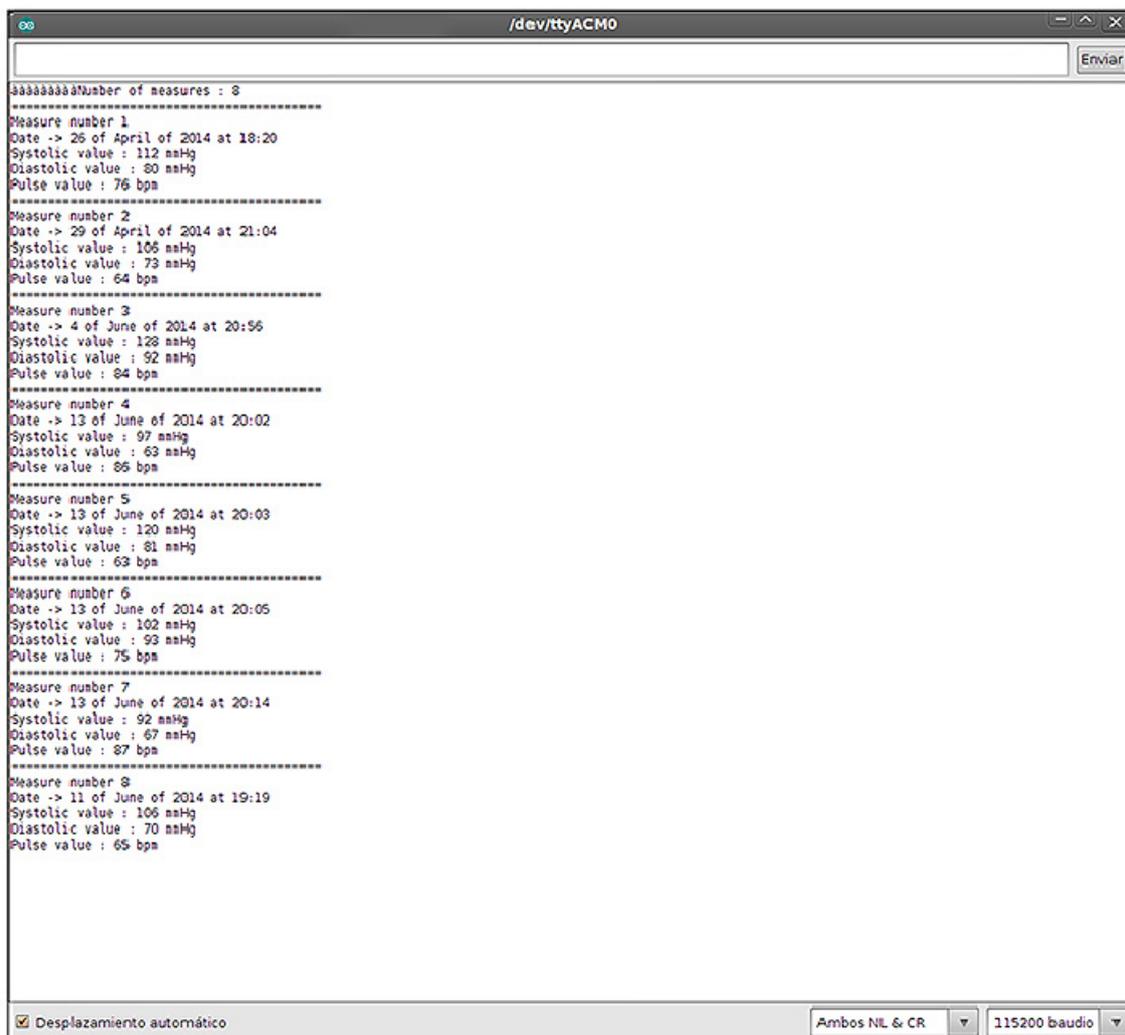


Figura 83 Visualización de la medida de tensión sanguínea por medio del monitor serie de Arduino

Posición del paciente

Estas son las funciones implementadas para este sensor:

```
initPositionSensor() // It initialize the position sensor.  
getBodyPosition() // Returns the body position.  
printPosition() // Prints the current body position.
```

La librería dispone de unas funciones sencillas implementadas para el acelerómetro que únicamente nos devuelve la posición del paciente en función de la posición de los 3 ejes. Antes de realizar la medición debemos inicializar la conexión con el sensor vía I2C y configurar algunos parámetros.

```
{  
    eHealth.initPositionSensor();  
}
```

Para obtener los datos disponemos de una función directa que almacena la posición en variables privadas de la clase ehealth.

```
{  
    uint8_t position = eHealth.getBodyPosition();  
}
```

Pudiendo obtener 5 posiciones distintas, todas ellas determinadas de manera interna por la librería en función de la distribución de aceleraciones en los distintos ejes. Para ello determina el nivel de aceleración de los distintos ejes X, Y, Z, y en función de cómo se distribuye la aceleración de la gravedad podemos distinguir entre 5 casos representados en la Figura 84 (todo ello al estar posicionado el acelerómetro en el pecho):

1 == Supine position.

2 == Left lateral decubitus.

3 == Right lateral decubitus.

4 == Prone position.

5 == Stand or sit position.

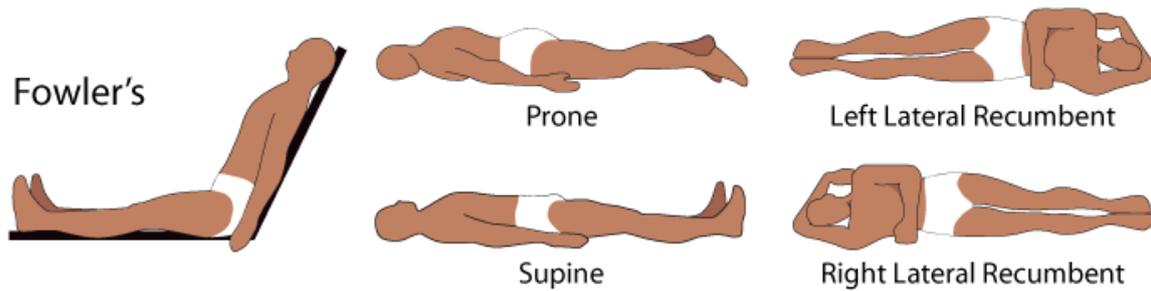


Figura 84 Posiciones detectadas por el sensor de posición

Para representar la posición únicamente debemos imprimirla por el monitor serie. Para ello también se ha implementado una función directa.

```
{
  Serial.print("Current position : ");
  uint8_t position = eHealth.getBodyPosition();
  eHealth.printPosition(position);
}
```

En la Figura 85 se puede observar la salida del monitor serie en el ejemplo básico.

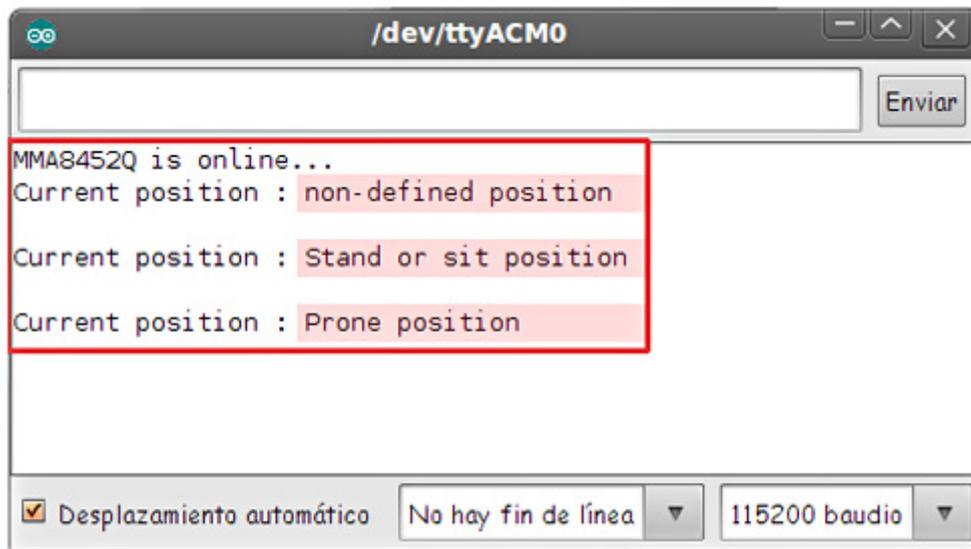


Figura 85 Visualización de la medida de posición por medio del monitor serie de Arduino

Conductividad de piel

Estas son las funciones implementadas para este sensor:

```
getSkinConductance() // Returns the value of skin conductance.
getSkinResistance() // Returns the value of skin resistance.
getSkinConductanceVoltage() // Returns the value of skin conductance in voltage.
```

El sensor de sudoración nos devuelve un valor analógico, únicamente es necesario hacer los cálculos de resistencia conociendo los datos del circuito de adaptación, y de su inversa obtener la conductancia.

```
{
  float conductance = eHealth.getSkinConductance();
  float resistance = eHealth.getSkinResistance();
  float conductanceVol = eHealth.getSkinConductanceVoltage();
}
```

En la Figura 86 se puede observar la salida del monitor serie en el ejemplo básico.



Figura 86 Visualización de la medida de conductividad por medio del monitor serie de Arduino

Glucosa en sangre

Estas son las funciones implementadas para este sensor:

```
readGlucometer() // Read the values stored in the glucometer.  
GetGlucometerLength() // Returns the number of data stored in the glucometer.  
numberToMonth() // Convert month variable from numeric to character.
```

Trabaja de manera muy similar al sensor de tensión sanguínea de la Versión 2. Establece primero conexión por medio de comunicación UART (el hardware adapta esta señal al protocolo 232 que utiliza el sensor) aunque en este caso la función habilita su lectura, ya que dispone de un interruptor que lo aísla del circuito para no provocar interferencias. Previamente hemos debido de realizar la medida de glucosa.

```
{  
    eHealth.readGlucometer();  
    Serial.begin(115200);  
}
```

La cantidad de datos está disponible por medio de una función pública.

```
{  
    uint8_t numberOfData eHealthClass.getGlucometerLength()  
}
```

Una vez obtenidos los datos podemos representarlo por pantalla de manera sencilla.

```
{  
    Serial.print(F("Glucose value : "));  
    Serial.print(eHealth.glucoseDataVector[i].glucose);  
    Serial.println(F(" mg/dL"));  
}
```

En la Figura 87 se puede observar la salida del monitor serie en el ejemplo básico.

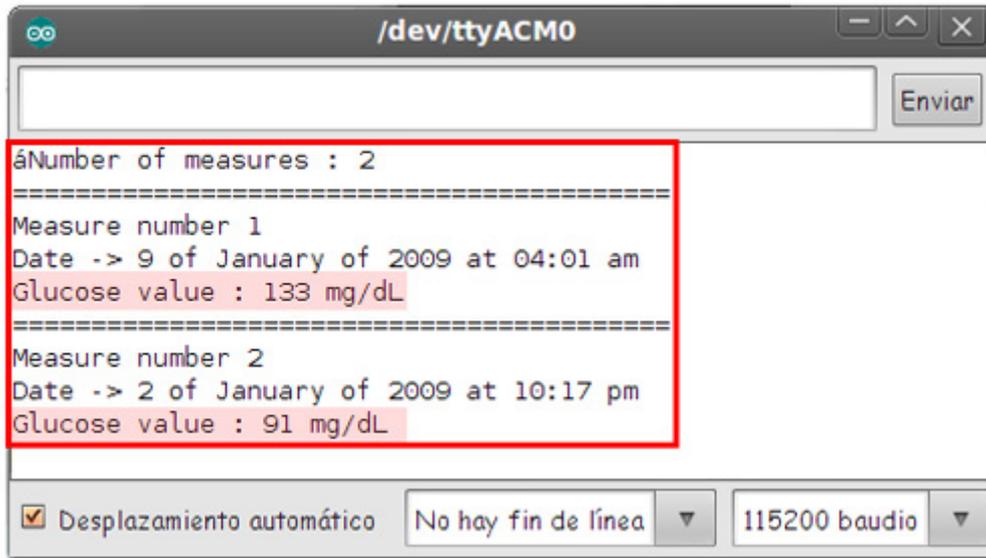


Figura 87 Visualización de la medida de glucosa por medio del monitor serie de Arduino

Electromiograma

Estas son las funciones implementadas para este sensor:

`getEMG()` // Returns an analogic value to represent the Electromyography.

Al igual que el sensor de ECG, utiliza una señal analógica de entrada para obtener un valor de 0 a 5 voltios que representa el impulso muscular.

```
{
  float EMG = eHealth.getEMG();
}
```

En la Figura 88 se puede observar la salida del monitor serie en el ejemplo básico.

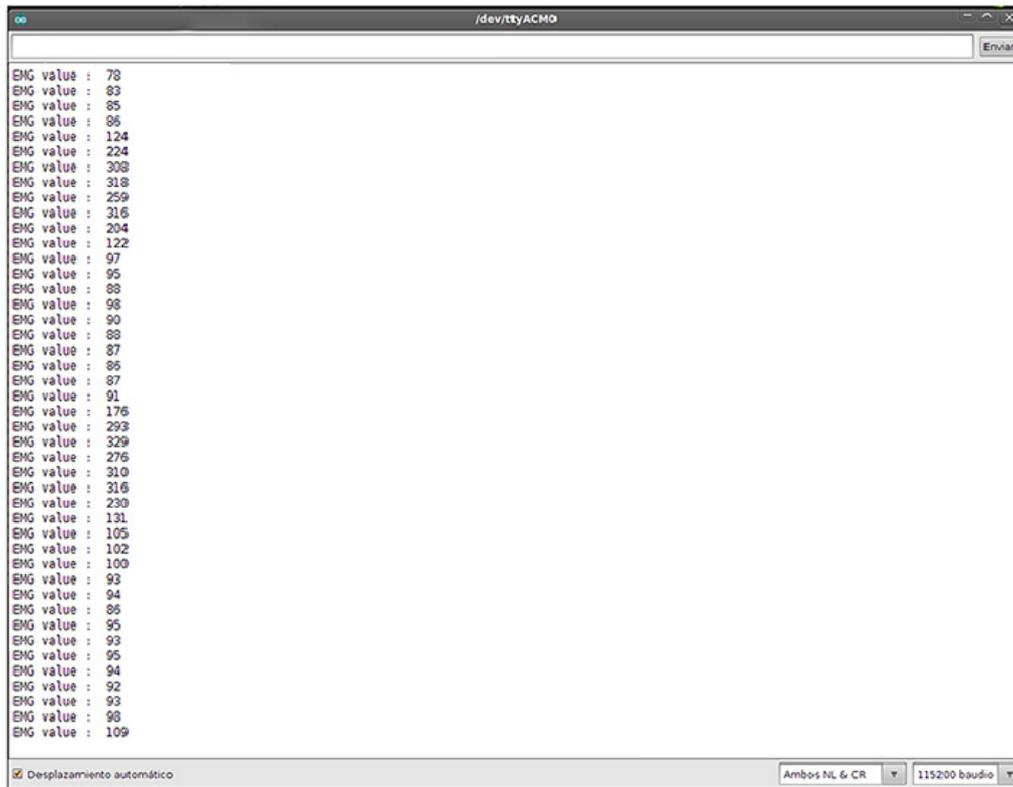


Figura 88 Visualización de la medida de electromiograma por medio del monitor serie de Arduino

5.2.2 Programas de ejemplo

Existen programas de ejemplo básicos para cada uno de los sensores, cuya misión es representar los valores medidos por el monitor serie disponible en Arduino.

Existen numerosos códigos de ejemplo básicos para los diferentes sensores. En la Figura 89 podemos observar dos modelos básicos en función del proceso de adquisición de datos.

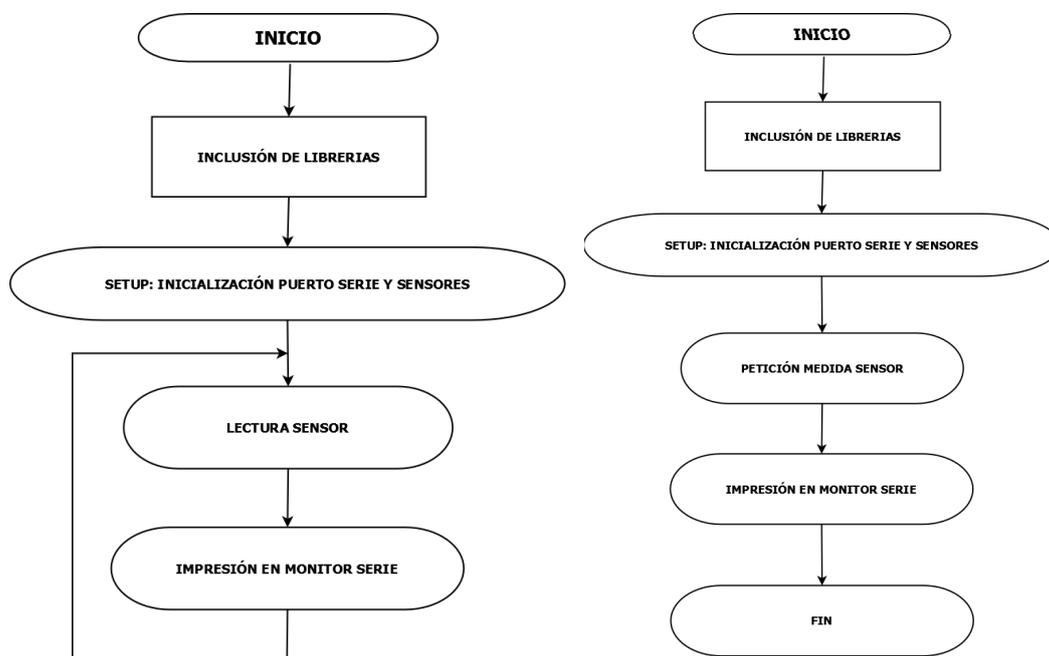


Figura 89 Diagrama de flujo de los códigos básicos de lectura de sensores

Interfaz monitor serie

Se han implementado otros muchos ejemplos combinados, pero destacaremos uno en el que se conectan varios sensores, implementando una interfaz en el monitor serie para elegir con que sensor trabajar. A continuación en la Figura 91 podemos observar el diagrama de flujo del código.

Todos los datos se pueden mostrar de manera sencilla en el monitor como en la captura de la Figura 90. En este ejemplo únicamente leemos diversos sensores, y los inicializamos. Una vez cargado el código y abierto el monitor obtendremos un menú de selección. En el menú podemos navegar escribiendo en la zona de entrada de caracteres las letras indicadas.

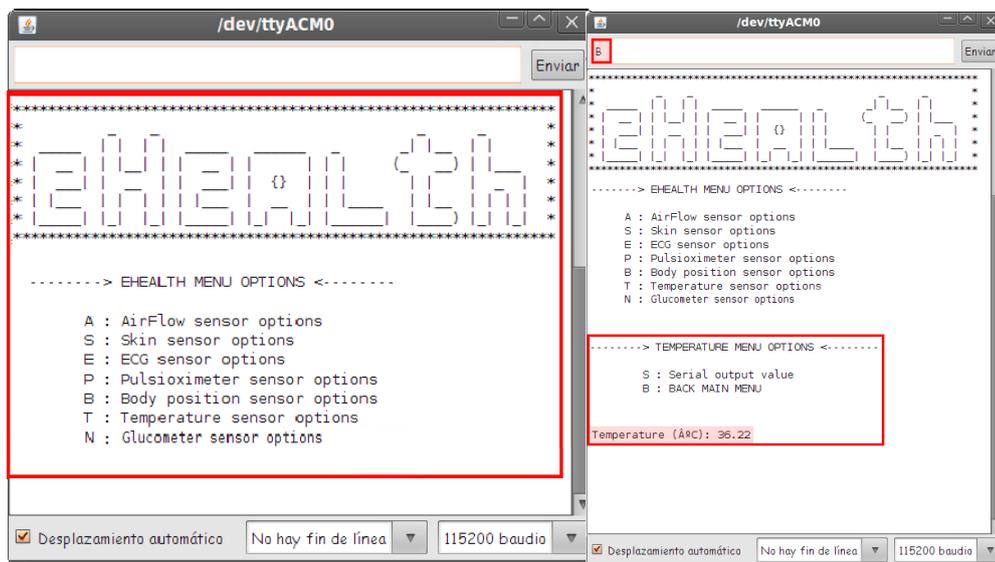


Figura 90 Interfaz del ejemplo en el monitor serie de Arduino

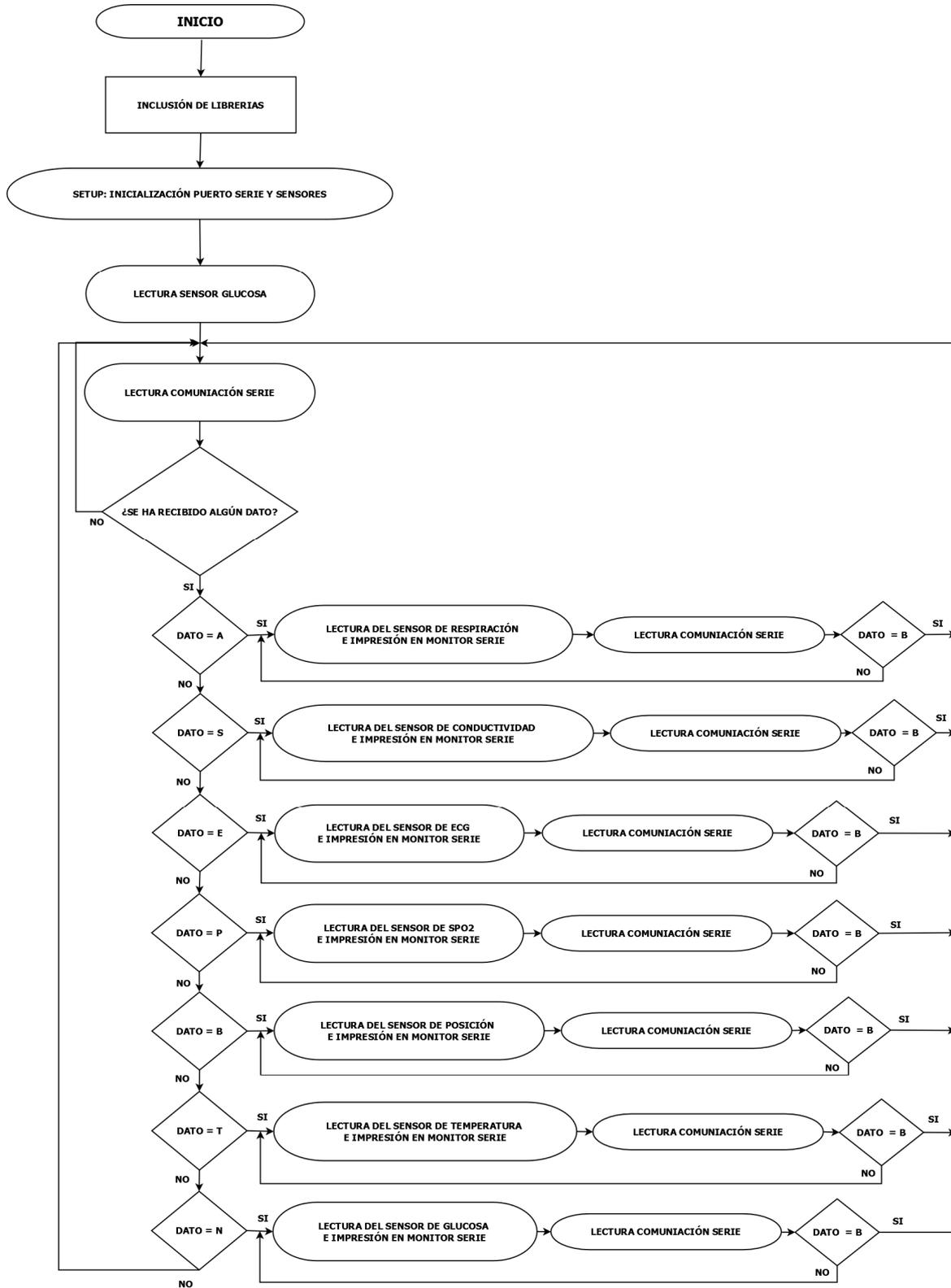


Figura 91 Diagrama de flujo del código de interfaz monitor serie

Las medidas pueden ser representadas de distintas maneras, mandando las letras correspondientes. Podemos utilizar una “B” para volver atrás en los menús.

```
#include <PinChangeInt.h>

char serialByte;
uint8_t state = 0;
uint8_t cont = 0;

void setup()
{
  eHealth.readGlucometer();
  Serial.begin(115200);
  initScreen();
  delay(100);
}

void loop()
{
  state = 0;

  while (Serial.available()>0){
    serialByte=Serial.read();

    if ((serialByte=='A') || (serialByte == 'a')) { //Airflow sensor
      airFlowMenu();
      airFlowBucle();
    } else if ((serialByte=='S') || (serialByte == 's')){
      skinSensorMenu();
      skinSensorBucle();
    } else if ((serialByte=='E') || (serialByte == 'e')){
      ECGMenu();
      ECGBucle();
    } else if ((serialByte=='P') || (serialByte == 'p')){
      pulsioximeterMenu();
      pulsioximeterBucle();
    } else if ((serialByte=='B') || (serialByte == 'b')){
      bodyPositionMenu();
    }
  }
}
```

```

bodyPositionBucle();

} else if((serialByte=='T') || (serialByte == 't')){
temperatureMenu();
temperatureBucle();

} else if((serialByte=='N') || (serialByte == 'n')){
GlucometerMenu();
GlucometerBucle();
}
}
}
}

```

5.3 Firmware asociado a otras funcionalidades

Para conseguir una completa versatilidad de la plataforma se han implementado ejemplos de distintas aplicaciones finales. Esto debe servir de base para los desarrolladores y que no se sientan perdidos ante el uso de la placa y la librería asociada.

5.3.1 Funciones

LCD Gráfico

Un ejemplo de aplicación final ha sido implementado mediante el uso de una pantalla gráfica. Se trata de un LCD gráfico de 128x64 que dispone de comunicación serial UART como se observa en la Figura 92.

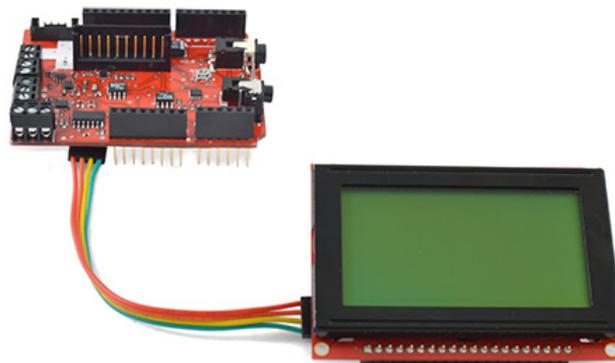


Figura 92 Display LCD gráfico conectado a Arduino

Para ello deberemos incluir en el código una librería especialmente diseñada para regular los mensajes impresos por pantalla.

```
#include < eHealthDisplay.h >
```

Debemos inicializar valores básicos sobre el LCD para poder configurarlo correctamente.

```
{  
    eHealthDisplay.init();  
}
```

Se definen funciones para distintas pantallas con valores predefinidos. En una primera se representarán valores básicos numéricos (Figura 93): posición, temperatura, SPO2, conductividad... Todas las pantallas deben ser inicializadas:

```
{  
    eHealthDisplay.initValuesScreen();  
}  
  
{  
    eHealthDisplay.initAirFlowScreen();  
}  
  
{  
    eHealthDisplay.initECGScreen();  
}
```

Y una vez inicializadas se pueden imprimir los valores correctamente. Se representan diversas ondas en la Figura 94 y la Figura 95.

```
{  
    eHealthDisplay.printValuesScreen();  
}  
  
{  
    eHealthDisplay.printAirFlowScreen();  
}  
  
{  
    eHealthDisplay.printECGScreen();  
}
```

La plataforma incluye un pulsador programable encargado de pasar entre las distintas pantallas de este ejemplo.

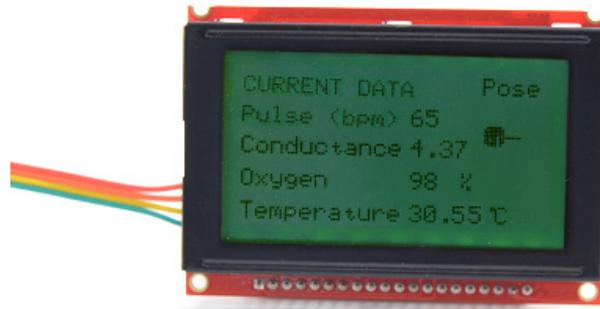


Figura 93 Primera pantalla con parámetros básicos del ejemplo del GLCD en e-Health

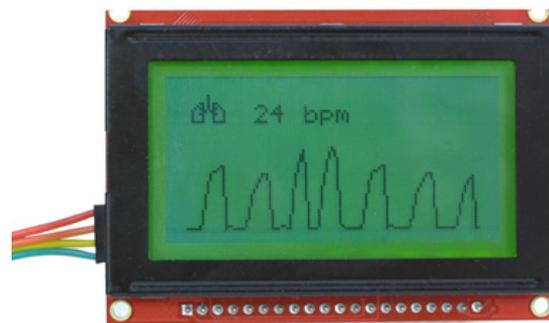


Figura 94 Segunda pantalla con medición respiratoria del ejemplo del GLCD en e-Health

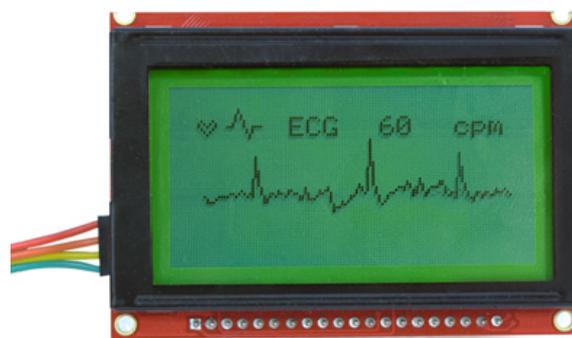


Figura 95 Tercera pantalla con medición ECG del ejemplo del GLCD en e-Health

5.3.2 Programas de ejemplo

En este caso, explicaremos un ejemplo de aplicación final en la que registramos los valores obtenidos en tiempo real en nuestro ordenador y los representamos en una gráfica, como indica el diagrama de flujo de la Figura 96.

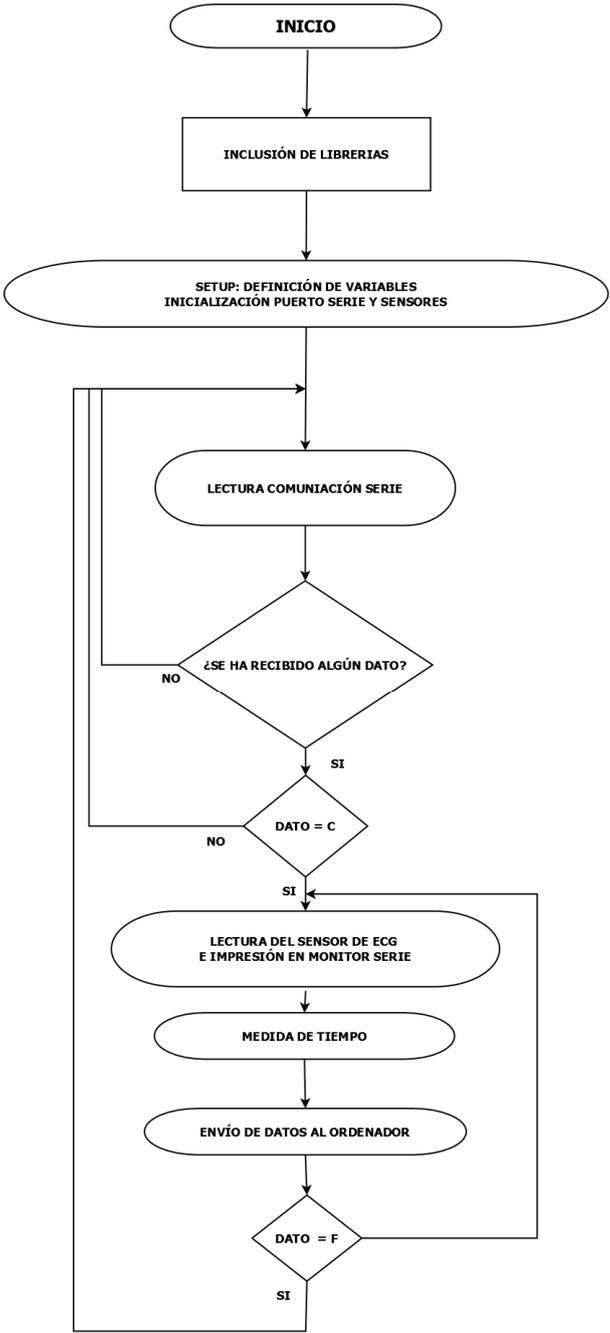


Figura 96 Diagrama de flujo del código de realización de gráficas en tiempo real

En este ejemplo, implementado para Linux, configuramos por consola que todos los datos que sean escritos en el USB al que se conecta Arduino se almacene en un archivo .txt.

Una vez esto activado, con el siguiente código, realizamos la medida de un sensor y lo imprimimos junto a una referencia de tiempo.

```
#include <eHealth.h>

extern volatile unsigned long timer0_overflow_count;

float fanalog0;
int analog0;
unsigned long time;
byte serialByte;

void setup() {
  Serial.begin(9600);
  Serial.println("Starting...");
}

void loop() {
  while (Serial.available()>0){
    serialByte=Serial.read();
    if (serialByte=='C'){
      while(1){
        fanalog0=eHealth.getECG();
        // Use the timer0 => 1 tick every 4 us
        time=(timer0_overflow_count << 8) + TCNT0;
        // Microseconds conversion.
        time=(time*4);
        //Print in a file for simulation
        Serial.print(time);
        Serial.print(",");
        Serial.println(fanalog0,5);
        if (Serial.available()>0){
          serialByte=Serial.read();
          if (serialByte=='F') break;
        }
      }
    }
  }
}
```

En este caso lo utilizaremos con el ECG como en la Figura 97, pero podríamos utilizar sudoración o respiración para otros ejemplos.

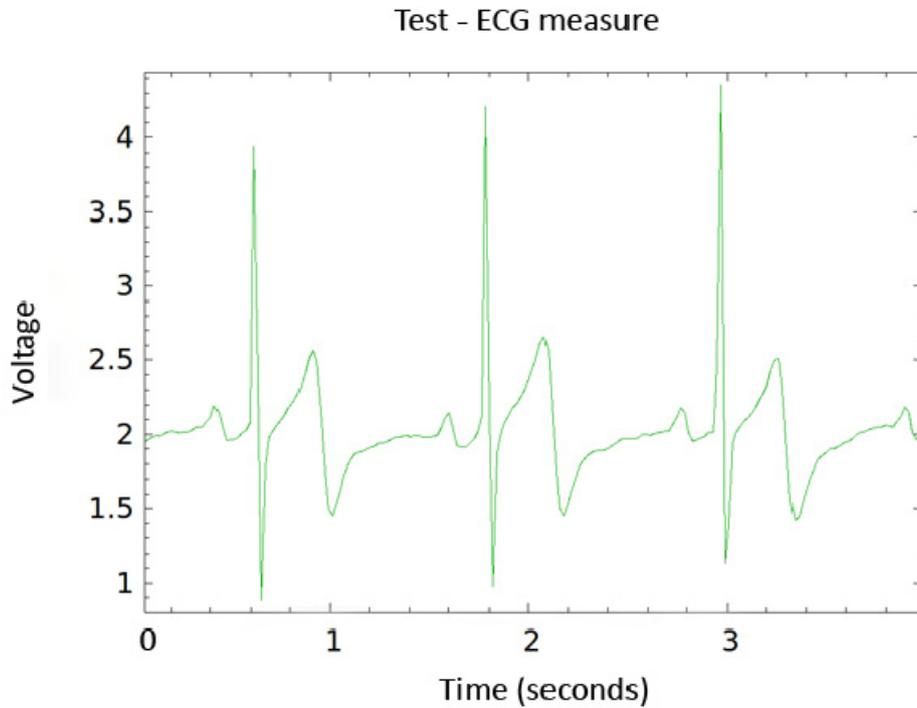


Figura 97 Representación gráfica de la señal ECG

En este momento estamos actualizando constantemente los valores de ECG en el archivo de texto, y ahora se puede utilizar un programa de representación gráfica en tiempo real como es el KST (Figura 98), para utilizar esos datos.

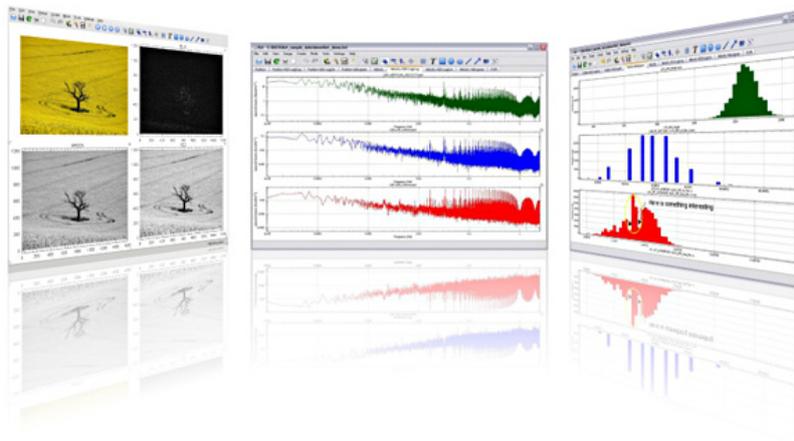


Figura 98 Interfaz básica del programa KST utilizado para graficar las señales de e-Health

6 Pruebas de laboratorio

Una parte relevante del proyecto es el de las distintas experiencias vividas que han servido para llegar a comprender los fundamentos de la telemedicina en la actualidad.

Tras esta primera toma de contacto más bien meramente informativa, se decidió la implementación de diversos ejemplos combinados con distintas tecnologías inalámbricas que sirvieran como pruebas de concepto, y pruebas reales de medición. El cómo utilizar esta plataforma para llevar a cabo un producto, es parte de la imaginación del desarrollador final. El producto deja el acceso abierto a todas las posibilidades, no como otros dispositivos del mercado que son completamente cerrados.

Esto sirvió también para observar que ciertos sensores con el tiempo o al principio de su funcionamiento necesitaban de un proceso de calibración, ya que en algunos casos las desviaciones podían ser críticas. Las calibraciones se explicarán en profundidad en el Anexo A.

6.1 Calibración sensores

Temperatura corporal

Al adaptarse el sensor por medio de un puente resistivo podemos ajustar los valores del circuito en la librería. Para ello deberemos utilizar un multímetro en distintas zonas del circuito.

Conductividad de piel

La precisión del sensor de conductividad puede ser crítica en determinadas aplicaciones. Al tratarse de una cuenta propia de circuitos la exactitud de los valores de resistencia y valores de tensión y corriente son muy importantes cuando se quiere precisión.

Electromiograma

Este sensor dispone de una ganancia ajustable por medio de un potenciómetro disponible en la parte superior de la placa.

6.2 Pruebas de medición

Representación numérica

Utilizando la comunicación serie que Arduino tiene implementada, y el monitor serie del IDE de programación, como se observa en la Figura 99, se han probado los valores de todos los sensores y comparado con los valores de dispositivos médicos certificados.

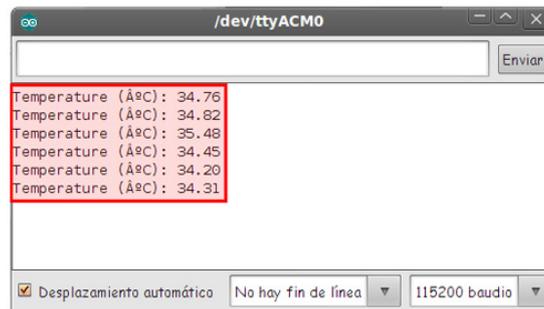


Figura 99 Salida de datos genérica por el monitor serie de Arduino

Representación gráfica

Como se observa en la Figura 100, podemos utilizar un programa para representar los valores en tiempo real de los sensores. Este proceso se ha realizado con varios de los sensores, especialmente con aquellos que permiten representación mediante gráficas.

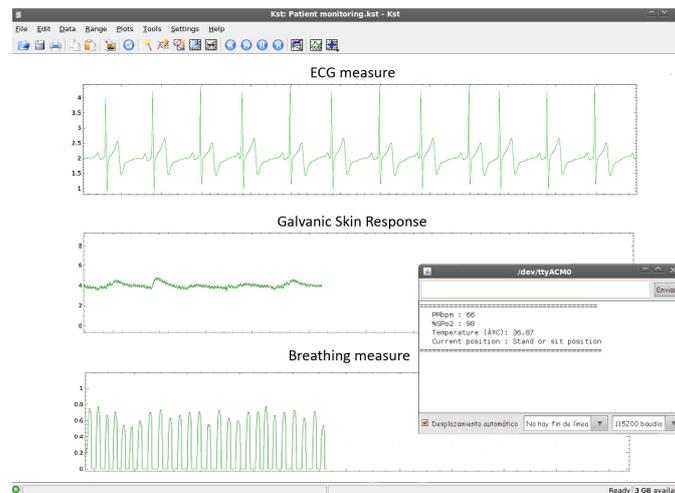


Figura 100 Representación gráfica de los valores biométricos medidos con e-Health

6.3 Pruebas de compatibilidad

La telemedicina es el uso de las telecomunicaciones y las tecnologías de la información para obtener apoyo médico a distancia. Esto elimina muchas barreras y ofrece soporte a comunidades rurales. e-Health gracias a estos ejemplos nos permite compartir los datos a la nube y realizar diagnóstico en tiempo real.

APP para móviles

Con conectividad WiFi se ha desarrollado una aplicación de móvil APP, gratuita tanto para Android como para Iphone, que permite monitorizar todos los parámetros de e-Health desde el móvil. Para este ejemplo como para muchos otros en este apartado utilizaremos la placa XBee desarrollada por Libelium, que permite conectar módulos de comunicaciones inalámbricas directamente sobre Arduino. Para este ejemplo utilizaremos el módulo WiFi Roving RN-171 como en la Figura 101.

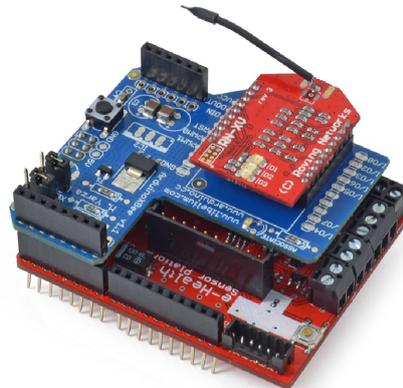


Figura 101 Módulo WiFi conectado sobre e-Health y Arduino utilizando el módulo de comunicaciones

El departamento de sistemas de la empresa ha desarrollado aplicaciones para ambos sistemas operativos de móviles, y se han creado ejemplos de aplicación para Arduino en la librería e-Health. Se establecen conexión vía WiFi de una manera determinada para cada uno de los sistemas operativos, y se envían los valores en una trama. Estos se representan en diversas pantallas separando valores numéricos, gráficas o ejemplos visuales.

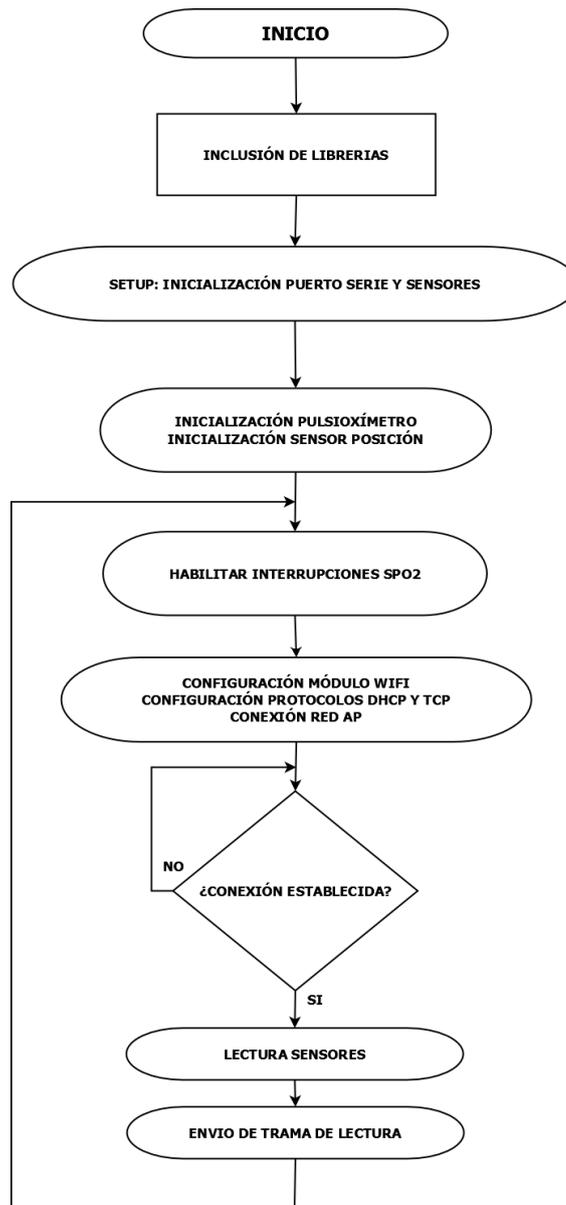


Figura 102 Diagrama de flujo del código de comunicación WiFi con un Smartphone

Aquí podemos ver el código, el diagrama de flujo para la aplicación de Android en la Figura 102 y en la Figura 103 capturas de pantalla de la aplicación:

```
#include < PinChangeInt.h >
#include < eHealth.h >
```

```

char recv[128];
float parameter = 0.098;
uint8_t cont = 0;

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

  eHealth.initPulsioximeter();
  eHealth.initPositionSensor();

  //Attach the intruptions for using the pulsioximeter.
  PCintPort::attachInterrupt(6, readPulsioximeter, RISING);
  delay(1000);
}

void loop()
{
  while (Serial.available()>0) {}
  // Enters in command mode
  Serial.print("$$$"); check();
  // Sets DHCP and TCP protocol
  Serial.print("set ip dhcp 1\r"); check();
  Serial.print("set ip protocol 1\r"); check();
  // Configures the way to join the network AP
  Serial.print("set wlan join 0\r"); check();
  Serial.print("join ANDROID\r"); check();
  Serial.print("set i h 255.255.255.255\r"); delay(1000);
  Serial.print("set i r 12345\r"); check();
  Serial.print("set i l 2000\r"); check();
  Serial.print("exit\r"); check();
  while(1){

    eHealth.initBloodPressureSensor(0.098);

    //1. Read from eHealth.
    int airFlow = eHealth.getAirFlow();
    float temperature = eHealth.getTemperature();
    float conductance = eHealth.getSkinConductance();
    float resistance = eHealth.getSkinResistance();
    float conductanceVol = eHealth.getSkinConductanceVoltage();
    int BPM = eHealth.getBPM();
  }
}

```

```

int SPO2 = eHealth.getOxygenSaturation();
uint8_t pos = eHealth.getBodyPosition();
int syst = eHealth.getSystolicPressure();
int diast = eHealth.getDiastolicPressure();
float ECG = eHealth.getECG();
//uint8_t glucose = eHealth.glucoseDataVector[0].glucose;

//Data sensor must be sent in this order to mobile android application
Serial.print(int(airFlow)); Serial.print("#");
Serial.print(ECG); Serial.print("#");
Serial.print(syst); Serial.print("#");
Serial.print(diast); Serial.print("#");
Serial.print(int(0)); Serial.print("#"); //Glucose is not implemented yet
Serial.print(temperature); Serial.print("#");
Serial.print(int(BPM)); Serial.print("#");
Serial.print(int(SPO2)); Serial.print("#");
Serial.print(conductance); Serial.print("#");
Serial.print(int(resistance)); Serial.print("#");
Serial.print(int(airFlow)); Serial.print("#");
Serial.print(int(pos)); Serial.print("#");
Serial.print("\n");

// Reduce this delay for more data rate
delay(250);
}
}

```

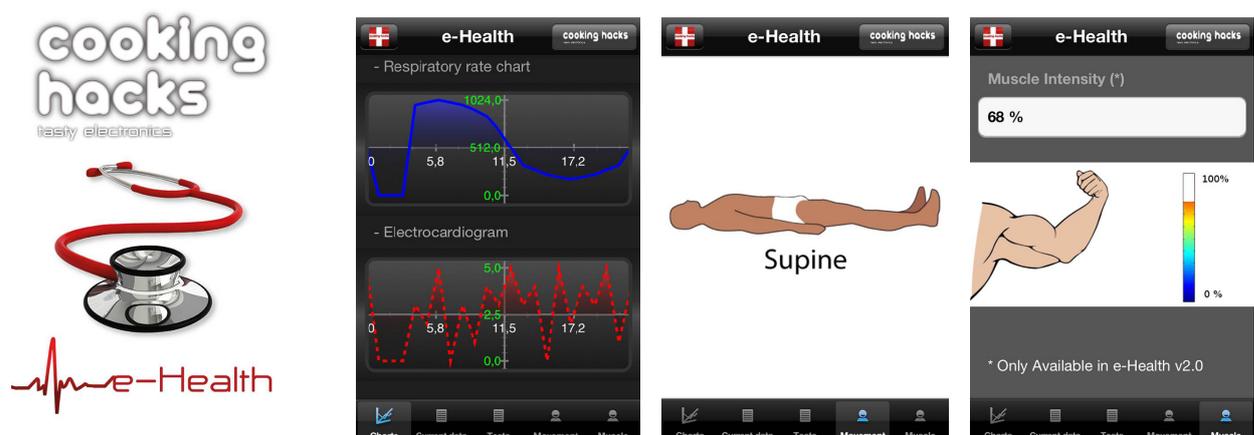


Figura 103 Capturas de pantalla de la aplicación de e-Health para smartphones

Bluetooth

También se ha implementado un ejemplo de envío y recepción de datos utilizando un módulo BT con el montaje de la Figura 104.



Figura 104 Ejemplo combinado de e-Health con tecnología bluetooth

Zigbee

Otro ejemplo es el de envío y recepción de datos por medio de comunicación inalámbrica XBee, específico para redes sensoriales de bajo consumo, con el montaje de la Figura 105.



Figura 105 Ejemplo combinado de e-Health con tecnología XBee

GPRS

Por medio de un módulo GPRS se ha programado una aplicación que envía los datos vía SMS, con el montaje de la Figura 106.

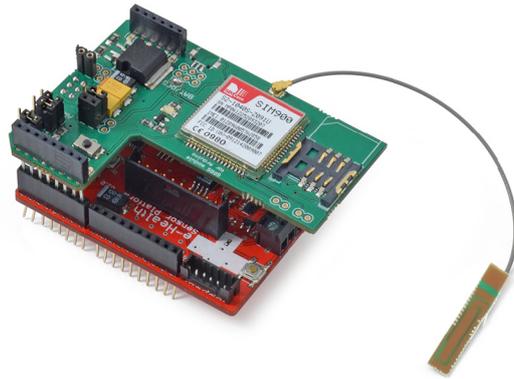


Figura 106 Ejemplo combinado de e-Health con tecnología GPRS

3G

Utilizando un módulo 3G se han subido datos a internet directamente a un servidor. El módulo 3G dispone de una cámara que permite realizar y diagnosticar imágenes a distancia (Figura 107).



Figura 107 Fotografía realizada con Arduino y tecnología 3G simulando una picadura de serpiente

7 Conclusiones

Se ha cumplido con éxito el objetivo del proyecto: la construcción y puesta a punto de un prototipo electrónico de un sistema de medición biométrico con fines médicos y de diagnóstico de pacientes en tiempo real.

Se dispone de un sistema que incorpora la electrónica junto a los circuitos necesarios, y el software utilizado para su control y programación.

También se han realizado las pruebas pertinentes para verificar su correcto funcionamiento.

La consecución del objetivo se hizo en orden, lo que ha permitido identificar una serie de hitos parciales:

- Se completó un exhaustivo proceso de investigación inicial, analizando todo lo relacionado con la telemedicina, reuniendo información general del tema.
- Se realizó un estudio básico de la tecnología con la que se iba a trabajar: Arduino, Waspote y Raspberry Pi.
- Se consultaron las necesidades de los profesionales y usuarios del entorno de la telemedicina.
- Se realizó un estudio de los distintos dispositivos más relevantes de diagnóstico del mercado y otro sobre los componentes y sensores para utilizar en el diseño.
- Se hizo un estudio de las diferentes aplicaciones de la telemedicina basado en una serie de experiencias reales.
- Para el diseño final del módulo, se realizó una serie de prototipos, con un *firmware* básico para posibilitar ejemplos de aplicación básicos. Tras diversas pruebas de funcionamiento, se diseñó un prototipo final funcional, con su correspondiente *firmware* definitivo.
- Se realizaron las pruebas sobre la placa definitiva, que constataron que se cumplían las expectativas y objetivos preestablecidos.

- En paralelo se ha llevado una documentación diaria en la que se anotaba todos los avances realizados a lo largo de la investigación y desarrollo.

El módulo final trabajando con Arduino y Raspberry Pi forma un sistema de altas prestaciones, muy completo, fiable y flexible, lo que junto al precio competitivo y su facilidad de uso permitirán llegar a un gran número de usuarios.

Como el acceso al *firmware* final es sencillo y está disponible, cualquier persona con algún conocimiento de programación puede usar o personalizar su sistema, más allá del original.

7.1 Conclusiones personales

Durante los meses de trabajo en Libelium, he aprendido multitud de cosas gracias a su excelente plantilla además de haber obtenido una grata experiencia. He aprendido cosas de múltiples ramas de la ingeniería, desde producción y fabricación (diseño, elección de componentes, soldadura y puesta a punto de una PCB), hasta diseño e investigación.

Tras la consecución de mi proyecto, he comenzado a formar parte de la plantilla de Libelium como ingeniero en I+D. Bajo mi punto de vista he realizado un buen trabajo con posibilidades de futuro y salida comercial.

En estos meses trabajando en el proyecto, he adquirido una visión más cercada y real del sector de la electrónica. El esfuerzo que he invertido me ha servido para aprender a tratar con dificultades, y la experiencia de trabajo en una empresa dentro de un departamento de I+D me ha resultado muy gratificante y constructiva. En este tiempo he madurado en diversos aspectos, y he aprendido a aplicar conocimientos que poseía.

He descubierto la relevancia que tiene el conocimiento del inglés en el mundo de la ingeniería, ya que diariamente te encuentras situaciones en las que debes desenvolverte de manera fluida con este idioma: como la lectura de documentación técnica o conversaciones con empleados de distintas empresas con el fin de búsqueda de información.

En cuanto a mi paso por la Universidad en la Ingeniería Industrial me he dado cuenta que los conocimientos adquiridos me han sido de gran utilidad. Aunque al trabajar en un empleo, te das cuenta de que has estado varios años sentado enfrente de una pizarra, sin ser consciente en las dificultades de tener que abordar un proyecto electrónico. Por esto creo que debería tratarse en mayor medida en un plan de estudio universitario de ingeniería, aspectos prácticos como son el desarrollo de productos, para abrimos las puertas al mundo laboral.

7.2 Líneas futuras

Actualmente e-Health es una plataforma de prototipado. Ha sido desarrollada para ayudar a investigadores, desarrolladores, artistas... con propósitos experimentales, lúdicos o de investigación. Desde para un artista que quiere que su obra interactúe con las señales biométricas emitidas por una persona (si su corazón se acelera al verla, si su temperatura cambia...), hasta para controlar los valores de glucosa de una persona mayor sin necesidad de moverse de casa o un investigador que no tiene acceso a los caros dispositivos de medición de ECG (electrocardiograma) de los hospitales, y recurre a esta plataforma para sus pruebas.

La principal idea de futuro es la de conseguir un mini hospital de bajo coste para países en vías de desarrollo o subdesarrollados, con un precio muy bajo, capaz de salvar muchas vidas, pero el producto tiene muchas otras posibilidades de aplicación: enfermos crónicos, ancianos, discapacitados... [40] [41] [42]

Dado que se ha llevado a cabo una segunda versión de la plataforma, mejorando algunos de los sensores e integrado algunos nuevos se hace una diferencia entre las líneas futuras ya implementadas y las no implementadas.

Algunos de los posibles trabajos futuros con el objetivo de mejorar este sistema que ya han sido implementados son:

- Integración de nuevos sensores biométricos: Electromiograma (movimiento muscular)
- Mejora de los sensores, para evitar un menor proceso de fabricación.

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

Algunos de los posibles trabajos futuros con el objetivo de mejorar este sistema que ya no han sido implementados son:

- Integración de nuevos sensores biométricos: Electroencefalograma (actividad cerebral), Electrooculograma (movimiento ocular), Sensor de ronquidos...
- Integración de módulos de recepción de datos mediante Bluetooth. De esta manera, serían integrables pulsioxímetros, así como aparatos de medición de presión arterial que soporten el envío de datos inalámbricamente. Por otro lado, este módulo de Bluetooth sería también capaz de enviar la información al móvil del posible usuario.
- Realización de pruebas de campo, bajo la supervisión de un médico, con el fin de comenzar con un posible trabajo de certificación, para poder ser utilizado en un futuro como sistema de diagnóstico real.

8 Bibliografía

Los libros, artículos, apuntes, proyectos y fuentes consultados a lo largo de este proyecto son los listados a continuación.

- [1] <http://www.libelium.com>
- [2] <http://www.libelium.com/products/waspmote/>
- [3] <http://www.libelium.com/products/meshlium/>
- [4] Victor Shnayder, Borrong Chen, Konrad Lorincz, Thaddeus R. F. Fulford Jones and Matt Welsh. Sensor Networks for Medical Care. 2005.
- [5] Yuri Carmona. “Diseño y evaluación de un dispositivo sensorial médico biométrico autónomo para escenarios con monitorización en tiempo real”. Universidad de Zaragoza. 2009.
- [6] <http://www.arduino.cc>
- [7] <http://www.raspberrypi.org/>
- [8] Medical Applications, users guide. Freescale. 2011.
- [9] Medical Connectivity Library, users guide. Freescale. 2012.
- [10] <http://www.polar.com/>
- [11] <http://store.neurosky.com/products/mindwave-1>
- [12] <http://www.goodday.es/>
- [13] Steven Rotolante, Suzanne Pittner, Kenneth McCarthy and Jonathan Campbell. The Wireless Vital Signs Monitoring System.
- [14] <http://embedded-lab.com/>
- [15] <http://www.seeedstudio.com/>
- [16] <http://www.advancertechnologies.com/>

- [17] <http://medicarduino.net/>
- [18] <http://www.arduino.cc/en/Main/BioSensorsBoard>
- [19] <http://rethinkmedical.com/tricorder.html>
- [20] Guang-Zhong Yang. Springer-Verlag. Body Sensor Networks. 2006.
- [21] David Prutchi and Michael Norris. Design and Development of Medical Electronic Instrumentation. 2005.
- [22] Ramon Pallàs-Areny and John G. Webster. Sensors and signal conditioning. 2001.
- [23] <http://es.farnell.com>
- [24] <http://es.rs-online.com>
- [25] <http://www.digikey.com>
- [26] Leif Sörnmo and Pablo Laguna. Bioelectrical Signal Processing in Cardiac and Neurological Applications. 2005.
- [27] Thaddeus R. F. Fulford-Jones, Gu-Yeon Wei and Matt Welsh. A Portable, Low-Power, Wireless Two-Lead EKG System. September 2004.
- [28] G. Arthur C. Guyton, M.D. & John E. Hall, Ph.D. Textbook of Medical Physiology. Chapter 73: Body Temperature, Temperature Regulation, and Fever. 2006.
- [29] Sistema termo regulatorio, sensores de temperatura, variabilidad de la frecuencia cardíaca. Dalcame.
- [30] K.W. Chan, K. Hung, Y. T. Zhang. Noninvasive and cuffless measurements of blood pressure for telemedicine. 2001.
- [31] Song Chung. Universal Serial Bus Device Class Definition for Personal Healthcare Devices. 2007.

- [32] <http://www.sparkfun.com>
- [33] Marc D. Strauss. HandWave: Design and Manufacture of a Wearable Wireless Skin Conductance Sensor and Housing. Massachusetts Institute of Technology.2005.
- [34] <http://www.datasheets.org.uk>
- [35] <http://www.alldatasheet.com>
- [36] <http://www.datasheetcatalog.org>
- [37] B. Martín del Brío, A. Bono Nuez. Transductores y sistemas de instrumentación.
- [38] B. Martín del Brío, A. Bono Nuez. Microprocesadores (Freescale 9S08).
- [39] C. Bernal. Notas sobre utilización Eagle 4.01.
- [40] Branko G. Celler, Jim Basilakis, Marc Budge, Nigel H. Lovell. A Clinical Monitoring and Management System for Residential Aged Care Facilities. 2006.
- [41] Robert Steele, Chris Secombe, Wayne Brookes. Using Wireless Sensor Networks for Aged Care: The Patient's Perspective. 2007.
- [42] Hongwei Huo, Youzhi Xu, Hairong Yan, Saad Mubeen, Hongke Zhang. An Elderly Health Care System Using Wireless Sensor Networks at Home. 2009.
- [43] Roger G. Mark. Principles of cardiac electrophysiology. MIT Division of Health Sciences and Technology. 2004.

Anexo A Información técnica

A.1. Parámetros médicos de interés

A.1.1. Pulso y oxígeno en sangre

La pulsioximetría es la medición no invasiva del oxígeno transportado por la hemoglobina en el interior de los vasos sanguíneos. Se realiza con un aparato llamado pulsioxímetro, que tiene un emisor de luz y un fotodetector, mediante los cuales calcula la absorción de oxígeno por parte de la sangre.

La medición de este parámetro es interesante para pacientes con medidas de oxígeno inestables. En la Tabla 1 es posible observar la peligrosidad de estos descensos de la saturación de oxígeno en sangre.

% Saturación	Actuación
> 95 %	No actuación inmediata.
95-90 %	Tratamiento inmediato y monitorización de la respuesta al mismo, según ésta, valorar derivación al hospital. Los pacientes con enfermedad respiratoria crónica toleran bien saturaciones en torno a estos valores.
< 90 %	Enfermo grave. Hipoxia severa. Oxigenoterapia + tratamiento y traslado al hospital.
< 80 %	Valorar intubación y ventilación mecánica.

Tabla 1 Análisis clínico de los valores de SpO₂

El punto crítico que debe dar la señal de alarma es el de saturaciones inferiores al 95% (inferiores al 90 ó 92% cuando existe patología pulmonar crónica previa) estos pacientes deben recibir tratamiento inmediato.

De este mismo sensor podemos obtener la medida de pulso. El pulso son los “saltos” palpables del flujo sanguíneo que se aprecian en diversos puntos del cuerpo. Es obtenido por lo general en partes del cuerpo donde las arterias se encuentran más próximas a la piel, como en las muñecas o el cuello. Es un indicador del estado circulatorio.

El ritmo cardiaco es medido en pulsos por minuto. En la Tabla 2 aparecen los valores normales según la edad. Edad Ritmo cardiaco (latidos por minuto).

Edad	Ritmo cardiaco (latidos por minuto)
Lactantes	120-160
Niños pequeños	90-140
Preescolares	80-110
Niños en edad escolar	75-100
Adolescentes	60-90
Adultos	60-100

Tabla 2 Ritmo cardiaco según la edad

A.1.2. Electrocardiograma

El electrocardiograma o ECG es uno de las mediciones más comunes en la medicina moderna. Sirve para el diagnóstico de patologías cardíacas y muchas otras mediciones. La medida depende mucho de las condiciones de monitorización y de la complejidad del sistema utilizado. En la Figura 108 observamos la onda típica.

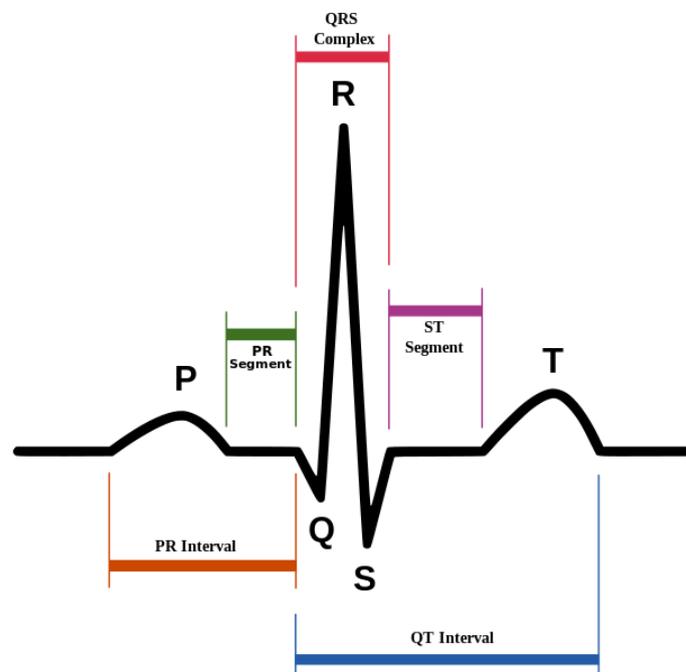


Figura 108 Señal típica de electrocardiograma

También denominado: Registro de actividad eléctrica cardiaca, ECG, EKG, Registro electrocardiográfico. Se trata de una prueba no invasiva de la actividad eléctrica del corazón. Para ello se emplean unos electrodos en la piel que detectan los distintos impulsos eléctricos del corazón en sus diferentes partes.

Los ECG se realizan con diversos electrodos pero en el caso de requerir una monitorización continua del ritmo, pueden ser usados tres electrodos solamente.

A.1.3. Respiración

El sistema respiratorio realiza diversas funciones (regulación del pH y temperatura, metabolismo de diversas sustancias, etc.), pero su función principal es la respiración, definida como todos aquellos procesos involucrados en el intercambio de gases entre el medio ambiente y el organismo. Una respiración inestable puede ser síntoma de un fallo fisiológico del organismo o en otros casos puede ser un indicador de un futuro fallo. Por eso y otras razones, monitorizar el ritmo respiratorio es otro de los principales parámetros a monitorizar en un paciente.

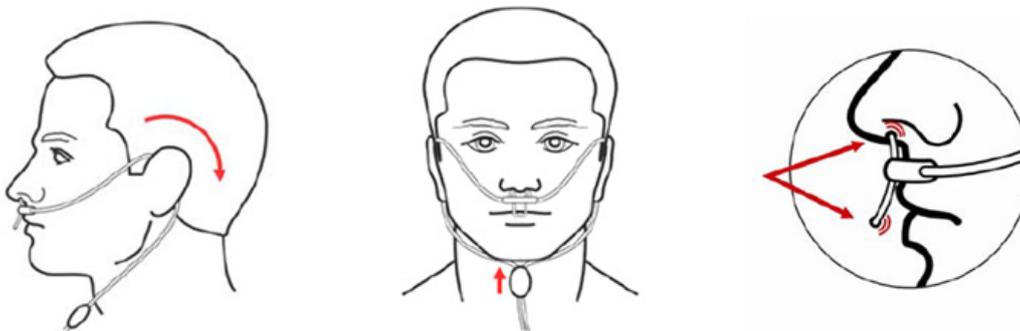


Figura 109 Colocación básica del sensor de respiración

Normalmente, el ritmo respiratorio es medido cuando una persona se encuentra en estado de reposo y simplemente se cuenta el número de veces por minuto que se produce el acto de inspirar y espirar. Este ritmo puede aumentar o disminuir en estados de fiebre, enfermedad u otras condiciones médicas. Es posible establecer un valor normal en aproximadamente 12 respiraciones por minuto en reposo para personas adultas. En la Figura 109 se indica cómo se coloca un sensor respiratorio básico.

A.1.4. Temperatura corporal

El ser humano, en condiciones fisiológicas normales, mantiene una temperatura corporal constante alrededor de 36.6°C. Ésta permanece dentro de unos límites gracias a una serie de mecanismos que favorecen el equilibrio térmico.



Figura 110 Colocación básica del sensor de temperatura en extremidades

En general la temperatura oral es un buen indicador de la temperatura corporal central y suele oscilar entre 36 y 37.8 °C. La temperatura rectal es aproximadamente 0.3-0.6 °C superior a la bucal. Su control es preferible en niños pequeños, enfermos graves y cuando la temperatura bucal no es fiable. La temperatura de la piel, como en la Figura 110, es menos representativa de la temperatura interna y varía con el estado de vasoconstricción y la distancia a los grandes vasos. En la Tabla 3 es posible ver las reacciones del ser humano frente a su temperatura corporal:

Temperatura	Reacciones en el ser humano a las diferentes temperaturas corporales
<26°C	Límite inferior de supervivencia
De 29 a 34°C	Malfuncionamiento de la termorregulación corporal
De 36 a 38°C	Temperatura normal del cuerpo.
De 38 a 40°C	Enfermedad (fiebre) o ejercicio físico de gran esfuerzo.
De 41 a 44°C	Golpe de calor. Lesiones cerebrales
> 44°C	Límite superior de supervivencia

Tabla 3 Reacciones en el ser humano frente a la temperatura corporal

Es muy importante monitorizar este parámetro, ya que la mayor parte de los fallos o patologías vienen acompañadas de cambios sustanciales de temperatura corporal.

A.1.5. Tensión Sanguínea

La tensión sanguínea está referida a la presión de la sangre en las arterias y venas, circulando gracias a la presión ejercida por nuestro corazón. Con cada latido, se impulsa sangre a lo largo de todo el organismo. Para ello se designan dos tensiones sanguíneas la sistólica, referida a los latidos y la diastólica, referida a los distintos tiempos de relajación entre los latidos. En la Figura 111 observamos la posición básica de medida.

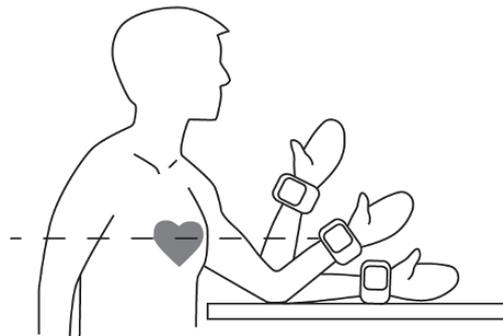


Figura 111 Posicionamiento del brazo para realizar la medida de tensión sanguínea

Una tensión elevada puede suponer un problema serio como un ataque al corazón, y sin darnos cuenta ya que no presenta síntomas aparentes. En la siguiente Tabla 4 podemos encontrar los valores normales de tensión sanguínea:

	Sistólica	Diastólica
Deseada	90–119	60–79
Pre Hipertensión	120–139	80–89
Hipertensión Estado 1	140–159	90–99
Hipertensión Estado 2	160–179	100–109
Crisis Hipertensión	≥ 180	≥ 110
Hipotensión	< 90	< 60

Tabla 4 Niveles de tensión sistólica y diastólica de los distintos estados

A.1.6. Posición del paciente

Es muchas ocasiones es necesario monitorizar la posición del paciente o que movimiento está realizando (Figura 112). Según cuál es su enfermedad habrá posiciones que no debe llegar a realizar para evitar peligros.

Si tenemos monitorizados los movimientos del cuerpo, podemos estudiarlos durante el sueño o durante una enfermedad y nos puede ayudar en la detección de caídas o malas posturas.

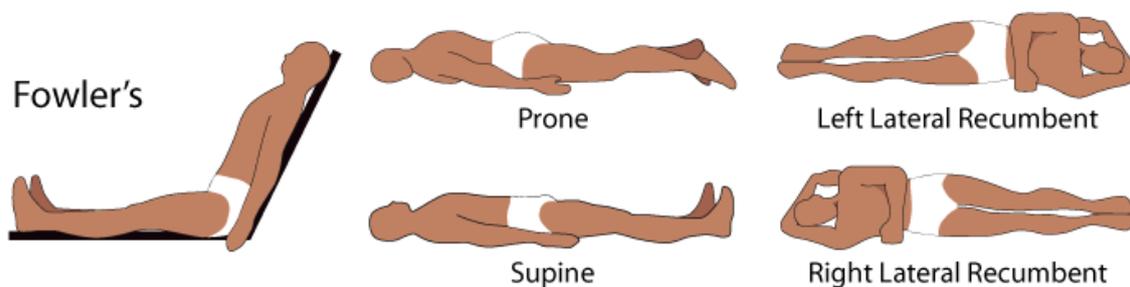


Figura 112 Posiciones básicas de un paciente

A.1.7. Conductividad de la piel

Los cambios emocionales, mentales y diferentes excitaciones físicas son reflejados en la piel de un individuo a través de las glándulas sudoríparas. Esta respuesta es denominada actividad eléctrica de la piel EDA.

El interés que tiene la conductancia de la piel, radica en la relación que tienen las excitaciones emocionales con las glándulas sudoríparas. Éstas están controladas por el sistema nervioso simpático, por lo que esta medida ha sido considerada como un buen reflejo de la actividad del sistema nervioso autónomo. Durante la excitación, de acuerdo con la respuesta simpática, las glándulas sudoríparas expulsan sudor sobre la piel, el cual es un buen conductor. Esto implica una reducción de la resistencia y por lo tanto un incremento de la conductividad de la piel. El proceso contrario, iniciado a través de la relajación disminuye el nivel de conductancia.

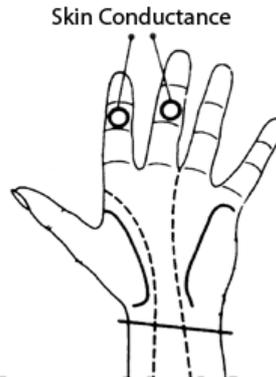


Figura 113 Colocación básica del sensor de conductividad

Convencionalmente, son aplicados 0.5 voltios en la superficie de la piel, y gracias a la medición de los cambios correspondientes de conductancia, es posible inferir el estado emocional del sujeto. En la Figura 113 se observa la zona de medida básica.

A.1.8. Glucosa en sangre

La medición de glucosa sirve para la detección aproximada de la concentración de este parámetro en la sangre. Se trata de una medida invasiva por la que utilizando una pequeña gota de sangre, gracias a la utilización de una lanceta extractora, podemos calcular el nivel de glucosa sanguíneo.

Se trata de un valor muy relevante en determinadas patologías, por lo que debe estar constantemente monitorizado y tener un registro de las medidas. Estas medidas pueden depender mucho de los alimentos previamente ingeridos, por lo que está especialmente recomendado para pacientes diabéticos.

A.1.9. Electromiografía

La electromiografía o EMG, estudia los impulsos eléctricos que mueven los músculos durante una contracción. Se trata de una técnica de evaluación de impulsos musculares, por la que se detectan diferencias de potencial en las zonas musculares, con el fin de detectar anomalías o actividades musculares extrañas.

Estas señales pueden ser utilizadas en numerosas aplicaciones médicas, a la vez que pueden servir para detectar problemas neuromusculares u otros problemas.

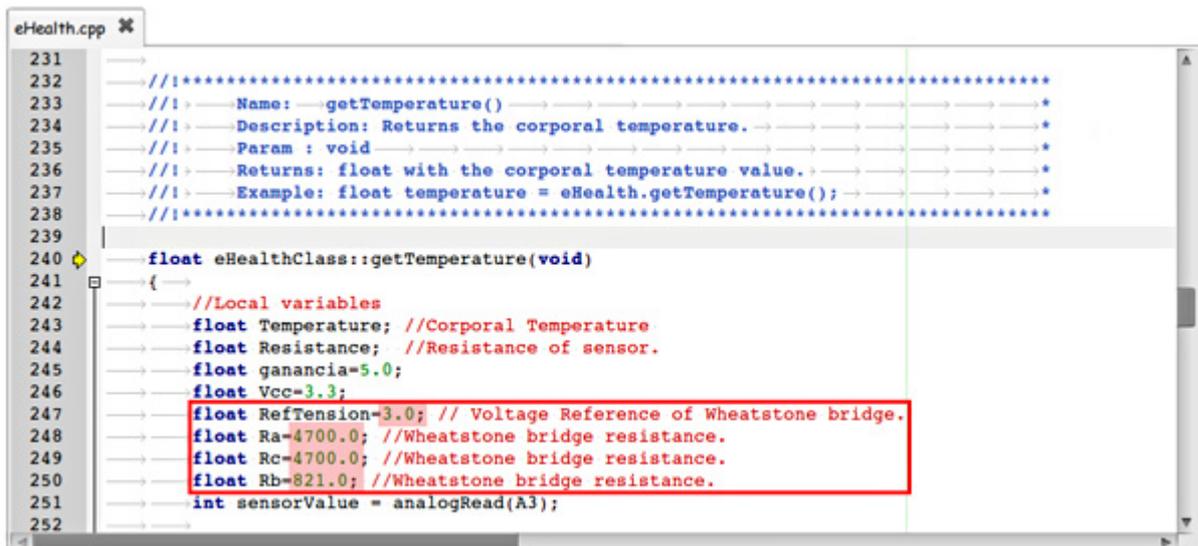
A.2 Calibración

Como se ha comentado en la memoria algunos sensores necesitan de procesos de calibración con el fin de mejorar su precisión. Son procesos muy diversos y dependen de la adaptación de cada uno de los sensores. Los procesos de calibración se detallan a continuación.

Temperatura corporal

La precisión del sensor de temperatura es suficiente para la mayoría de aplicaciones básicas. Si se quiere mejorar la precisión es necesario realizar un proceso de calibración.

Para obtener el valor de temperatura se mide un valor de tensión correlacionado con el valor de temperatura. El proceso de calibración se basa en la medición de los valores resistivos y de tensión utilizado en los cálculos necesarios para obtener la medida. Para ello únicamente necesitamos medir utilizando un multímetro en unos puntos específicos. Una vez anotados podemos modificarlos en la función de obtención de temperatura en el código ehealth.cpp de la librería como se observa en la Figura 114.



```

231
232 //!*****
233 //! Name: getTemperature()
234 //! Description: Returns the corporal temperature.
235 //! Param : void
236 //! Returns: float with the corporal temperature value.
237 //! Example: float temperature = eHealth.getTemperature();
238 //!*****
239
240 float eHealthClass::getTemperature(void)
241 {
242 //Local variables
243 float Temperature; //Corporal Temperature
244 float Resistance; //Resistance of sensor.
245 float ganancia=5.0;
246 float Vcc=3.3;
247 float RefTension=3.0; // Voltage Reference of Wheatstone bridge.
248 float Ra=4700.0; //Wheatstone bridge resistance.
249 float Rc=4700.0; //Wheatstone bridge resistance.
250 float Rb=821.0; //Wheatstone bridge resistance.
251 int sensorValue = analogRead(A3);
252

```

Figura 114 Valores que hay que calibrar en el sensor de temperatura corporal

Para medir los valores de la resistencia podemos medir entre sus extremos como se muestra en la Figura 115. Repetiremos el mismo proceso con la señal de 3 voltios de referencia. Al sustituir estos valores obtendremos una mayor precisión.

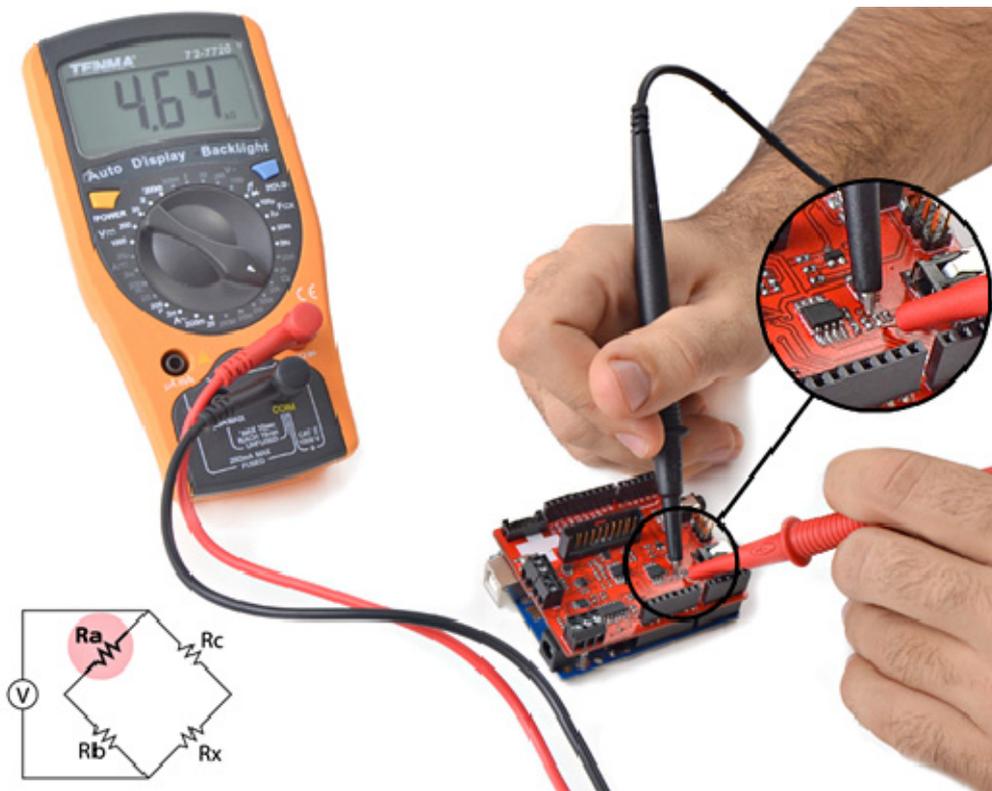


Figura 115 Utilización del multímetro para la calibración del sensor de temperatura

Tensión sanguínea y glucómetros

El proceso de calibración del primer sensor de tensión de la Versión 1, al igual que la propia medida, era demasiado impreciso, lo que sirvió de razón junto a problemas con el distribuidor para buscar un sensor que lo sustituyera.

En cuanto al sensor de la Versión 2 al igual que el sensor de glucosa, debido a trabajar ambos por UART, necesita de una modificación de software si queremos trabajar con grandes números de medidas.

El buffer definido en Arduino es demasiado pequeño para recibir grandes tramas, por lo que podemos incrementar este tamaño modificando algunas líneas de código del archivo HardwareSerial.cpp de Arduino.

Las líneas que debemos modificar son:

```
#if (RAMEND < 1000)
    #define SERIAL_BUFFER_SIZE 32
#else
    #define SERIAL_BUFFER_SIZE 256
#endif
```

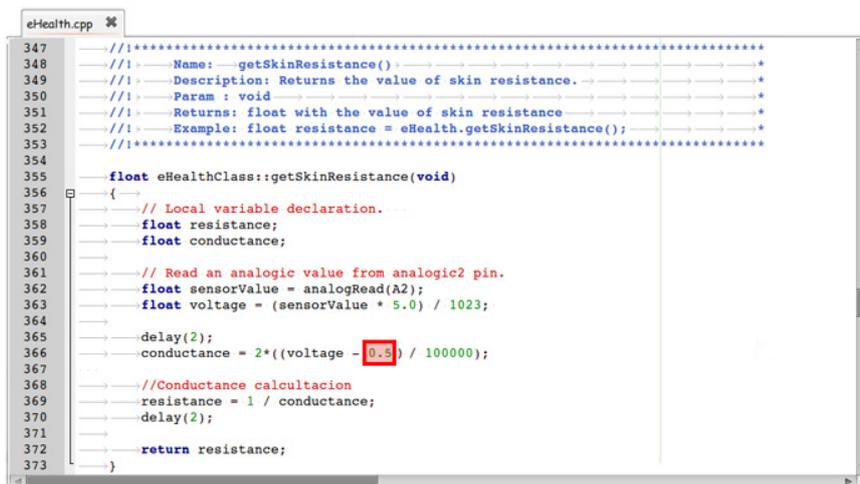
Por:

```
#if (RAMEND < 1000)
    #define SERIAL_BUFFER_SIZE 256
#else
    #define SERIAL_BUFFER_SIZE 256
#endif
```

Conductividad de piel

La precisión del sensor de conductividad puede ser mejorada para aplicaciones críticas siguiendo el proceso de medición de valores que ya hemos explicado

Utilizando un multímetro podemos medir en la zona del circuito la tensión de referencia que sirve para la medida, y se puede sustituir el valor exacto en la librería como se indica en la Figura 116.



```
eHealth.cpp
347 //!-----Name: getSkinResistance()-----*
348 //!-----Description: Returns the value of skin resistance.-----*
349 //!-----Param : void-----*
350 //!-----Returns: float with the value of skin resistance-----*
351 //!-----Example: float resistance = eHealth.getSkinResistance();-----*
352 //!-----*
353 //!-----*
354
355 float eHealthClass::getSkinResistance(void)
356 {
357     // Local variable declaration.
358     float resistance;
359     float conductance;
360
361     // Read an analogic value from analogic2 pin.
362     float sensorValue = analogRead(A2);
363     float voltage = (sensorValue * 5.0) / 1023;
364
365     delay(2);
366     conductance = 2*((voltage - 0.5) / 100000);
367
368     //Conductance calculacion
369     resistance = 1 / conductance;
370     delay(2);
371
372     return resistance;
373 }
```

Figura 116 Valores que hay que calibrar en el sensor conductividad

Electromiograma

Este sensor dispone de una ganancia ajustable por medio de un potenciómetro disponible en la parte superior de la placa como se observa en la Figura 117. Permite ajustar el valor de referencia por medio de un destornillador, ya que no todo los impulsos musculares son de la misma magnitud. En la Figura 118 observamos un ejemplo de medida.

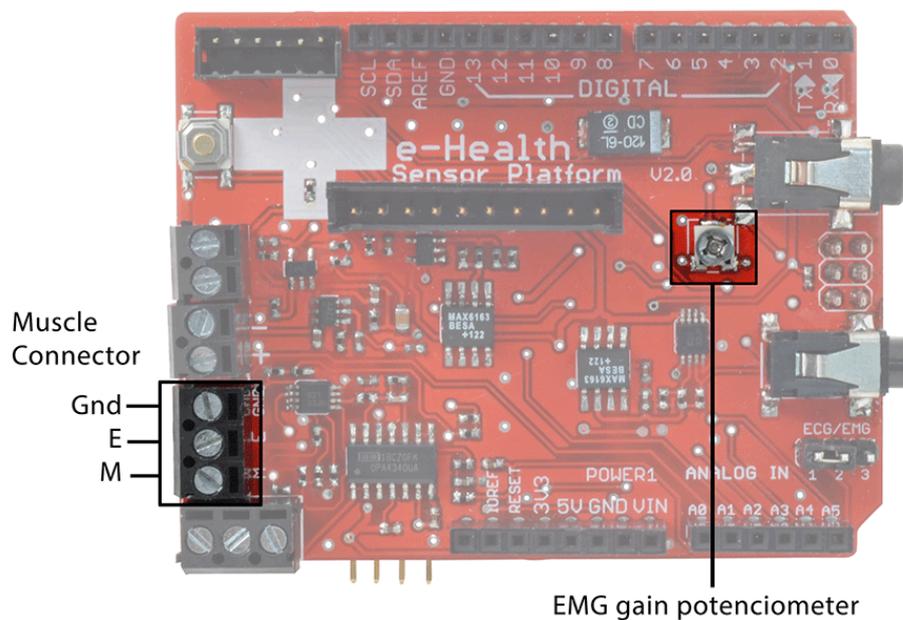


Figura 117 Localización del potenciómetro de calibración del sensor EMG

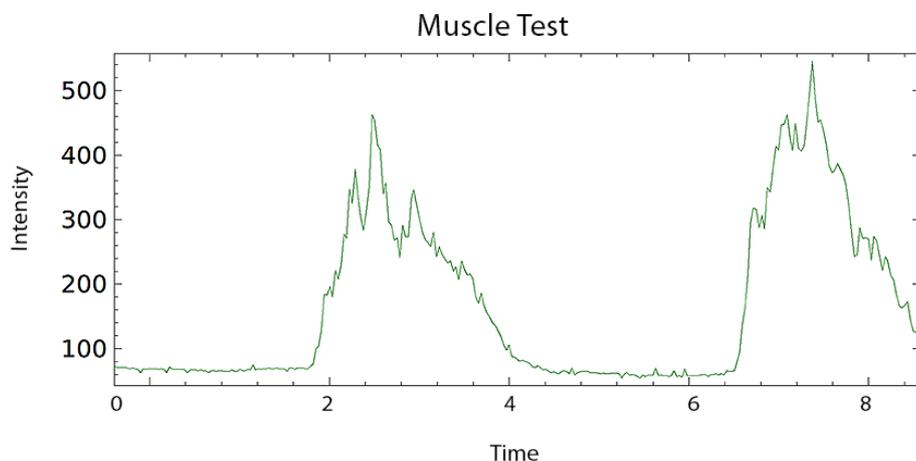


Figura 118 Gráfica obtenida con los valores EMG

Anexo B Códigos de programación

B.1 Software de base: Arduino IDE

Para realizar cualquier tipo de programación con la tecnología de Arduino, es necesario instalar el IDE. Se puede encontrar en la página web de Arduino y está disponible para descargar gratuitamente por cualquier usuario. En la Figura 119 se presenta una captura de pantalla de la interfaz básica.



```
tensionometerExample 5
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program. If not, see <http://www.gnu.org/licenses/> .
*
* Version 0.1
* Author: Ahmad Saad & Luis Martin
*/

#include <eHealth.h>
float parameter = 0.118;

void setup() {
  Serial.begin(115200);
  Serial.println("Press On/Off button please...");
}

void loop() {
  eHealth.initBloodPressureSensor(parameter);

  Serial.println("*****");
  Serial.print("Systolic blood pressure value : ");
  Serial.println(eHealth.getSystolicPressure());
  delay(10);

  Serial.println("*****");
  Serial.print("Diastolic blood pressure value : ");
  Serial.println(eHealth.getDiastolicPressure());
  delay(10);

  delay(2000);
}
```

Figura 119 Captura de pantalla del IDE de Arduino

Es importante usar la última versión que se encuentra en la web. El entorno de programación incluye todas las librerías de API necesarias para compilar los programas. Una vez tenemos un programa cargado en el microcontrolador, el funcionamiento de

Arduino se basa en el código cargado. La estructura de códigos se divide en dos partes fundamentales: *setup* y *loop*. Ambas partes del código tienen un comportamiento secuencial, ejecutándose las instrucciones en el orden establecido. El *setup* es la primera parte del código que se ejecuta, haciéndolo solo una vez al iniciar el código. En esta parte es recomendable incluir la inicialización de los módulos, alimentación, entre otros, que se vayan a utilizar. El *loop* es un bucle que se ejecuta continuamente, formando un bucle infinito

En el entorno existen una serie de botones que sirven para un manejo rápido del código:

- Verificar: Chequea el código en busca de errores.
- Cargar: Compila el código y lo vuelca en la placa E/S de Arduino.
- Nuevo: Crea un nuevo sketch.
- Abrir: Presenta un menú de todos los programas sketch de su "*sketchbook*" (librería de sketch). Un clic sobre uno de ellos lo abrirá en la ventana actual.
- Guardar: Salva el programa sketch.
- Monitor Serial: Inicia la monitorización serie

B.2 Firmware

La librería engloba el conjunto de subprogramas que sirven para el funcionamiento correcto del software. Está compuesta por dos archivos principales: *eHealth.h*, la cabecera de la librería, en donde se definen todas las variables y funciones a utilizar, y sus propiedades (públicas, privadas...), y *eHealth.cpp*, que es el objeto de la librería, en la que se programan cada una de las funciones definidas en el otro archivo.

Todas las funciones públicas definidas se encuentran explicadas en la Memoria del proyecto. En esta sección incluiremos el archivo *.h* ya que el archivo *.cpp* tiene demasiada extensión. Se puede consultar en el CD adjunto con el proyecto, o en la sección Bibliografía de la Memoria.

eHealth.h

/*

* Description: "The e-Health Sensor Shield allows Arduino and Raspberry Pi
 * users to perform biometric and medical applications by using 9 different
 * sensors: Pulse and Oxygen in Blood Sensor (SPO2), Airflow Sensor (Breathing),
 * Body Temperature, Electrocardiogram Sensor (ECG), Glucometer, Galvanic Skin
 * Response Sensor (GSR - Sweating), Blood Pressure (Sphygmomanometer) and
 * Patient Position (Accelerometer)."
 * Version 2.0
 * Author: Luis Martín & Ahmad Saad

*/

*/

*/

// Ensure this library description is only included once

#ifndef eHealthClass_h

#define eHealthClass_h

#include "Arduino.h"

// Library interface description

class eHealthClass {

public:

/**

// Constructor of the class

*

/**

//! Class constructor.

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

\return float : The corporal temperature value.

```
*/ float getTemperature( void );
```

//! Returns the oxygen saturation in blood in percent.

```
/*!
```

```
\param void
```

\return int : The oxygen saturation value. Normal values between 95-99%

```
*/ int getOxygenSaturation(void);
```

//! Returns the heart beats per minute.

```
/*!
```

```
\param void
```

\return int : The beats per minute.

```
*/ int getBPM(void);
```

//! Returns the value of skin conductance.

```
/*!
```

```
\param void
```

\return float : The skin conductance value.

```
*/ float getSkinConductance(void);
```

//! Returns the value of skin resistance.

```
/*!
```

```
\param void
```

\return float : The skin resistance value.

```
*/ float getSkinResistance (void);
```

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

//! Returns the value of skin conductance in voltage.

/*!

\param void

\return float : The skin conductance value in voltage (0-5v).

*/ float getSkinConductanceVoltage(void);

//! Returns an analogic value to represent the Electrocardiography.

/*!

\param void

\return float : The analogic value (0-5V).

*/ float getECG(void);

//! Returns an analogic value to represent the Electromyography.

/*!

\param void

\return float : The analogic value (0-5V).

*/ int getEMG(void);

//! Returns the body position.

/*!

\param void

\return uint8_t : the position of the patient.

* 1 == Supine position.

* 2 == Left lateral decubitus.

* 3 == Right lateral decubitus.

* 4 == Prone position.

* 5 == Stand or sit position

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

```
*/ uint8_t getBodyPosition(void);

//! Returns the value of the systolic pressure.
/*!
\param void
\return int : The systolic pressure.
*/ int getSystolicPressure(int i);

//! Returns the value of the diastolic pressure.
/*!
\param void
\return int : The diastolic pressure.
*/ int getDiastolicPressure(int i);

//! Returns an analogic value to represent the air flow.
/*!
\param void
\return int : The value (0-1023) read from the analogic in.
*/ int getAirFlow(void);

//! Prints the current body position
/*!
\param uint8_t position : the current body position.
\return void
*/ void printPosition( uint8_t position );

//! It reads a value from pulsioximeter sensor.
```

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

```
/*!  
  
\param void  
  
\return void  
  
*/    void readPulsioximeter(void);  
  
//! Prints air flow wave form in the serial monitor  
/*!  
  
\param int air : analogic value to print.  
  
\return void  
  
*/    void airFlowWave(int air);  
  
//! Read the values stored in the glucometer.  
/*!  
  
\param void  
  
\return void  
  
*/    void readGlucometer(void);  
  
//!Returns the number of data stored in the glucometer.  
/*!  
  
\param void  
  
\return int : length of data  
  
*/    uint8_t getGlucometerLength(void);  
  
//!Returns the number of data stored in the blood pressure sensor.  
/*!  
  
\param void  
  
\return int : length of data
```

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

```
*/      uint8_t getBloodPressureLength(void);

//! Returns the library version
/*!

\param void
\return int : The library version.

*/      int version(void);

//! Convert month variable from numeric to character.
/*!

\param int month in numerical format.
\return String with the month characters (January, February...).

*/      String numberToMonth(int month);

//!Struct to store data of the glucometer.
struct glucoseData {
    uint8_t year;
    uint8_t month;
    uint8_t day;
    uint8_t hour;
    uint8_t minutes;
    uint8_t glucose;
    uint8_t meridian;
};

//!Vector to store the glucometer measures and dates.
glucoseData glucoseDataVector[8];
```

//!Struct to store data of the blood pressure sensor.

```

struct bloodPressureData {
    uint8_t year;
    uint8_t month;
    uint8_t day;
    uint8_t hour;
    uint8_t minutes;
    uint8_t systolic;
    uint8_t diastolic;
    uint8_t pulse;
};
  
```

//!Vector to store the blood pressure measures and dates.

```
bloodPressureData bloodPressureDataVector[8];
```

private:

```

//*****
  
```

```

// Private Methods
  
```

```

    *
  
```

```

//*****
  
```

//! Initialize the MMA8452 registers

```
void initMMA8452(byte fsr, byte dataRate);
```

//! Sets the MMA8452 to standby mode. It must be in standby to change most register settings.

```

void MMA8452Standby();

data.
    //! Sets the MMA8452 to active mode. Needs to be in this mode to output
    array

void MMA8452Active();

    //! Read i registers sequentially, starting at address into the dest byte
    array

void readRegisters(byte address, int i, byte * dest);

    //! Read a single byte from address and return it as a byte.
    byte readRegister(uint8_t address);

    //! Writes a single byte (data) into address.
    void writeRegister(unsigned char address, unsigned char data);

    //! This function will read the p/l source register and
    //! print what direction the sensor is now facing.
    void portraitLandscapeHandler();

    //! Assigns a value depending on body position.
    void bodyPosition(void);

    //! Converts from 7 segments to number.
    uint8_t segToNumber(uint8_t A,
                        uint8_t B,
                        uint8_t C,
                        uint8_t D,
  
```

```

uint8_t E,
uint8_t F,
uint8_t G );

```

//! Assigns a value depending on body position.

```
char swap(char _data);
```

```
//*****
```

```
// Private Variables
```

```
*
```

```
//*****
```

//! It stores the systolic pressure value

```
int systolic;
```

//! It stores the diastolic pressure value

```
int diastolic;
```

//! It stores the beats per minute value.

```
int BPM;
```

//! It stores blood oxygen saturation value.

```
int SPO2;
```

//! It stores current body position.

```
uint8_t bodyPos;
```

//! x/y/z accel register data store here.

<i>e-Health Sensor Platform: Diseño e implementación de una plataforma sensorial biométrica para diagnóstico de pacientes en tiempo real y otras aplicaciones médicas</i>	MEMORIA
Revisión n°:3	Fecha: 13/02/2014

```
byte data[6];

//! Stores the 12-bit signed value.
int accelCount[3];

//! Stores the real accel value in g's.
float accel[3];

//! Stores the body position in vector value.
uint8_t position[];

//!It stores the number of data of the glucometer.
uint8_t length;

};

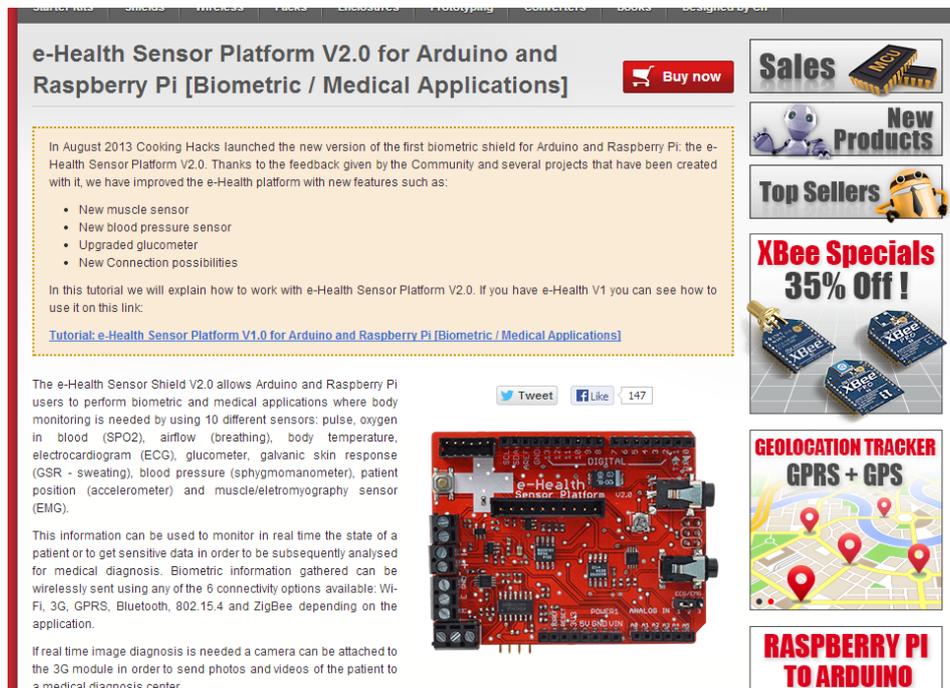
extern eHealthClass eHealth;

#endif
```

Anexo C Documentación online generada

C.1 Guía de producto

Se ha llevado una documentación online del proyecto, como un método para compartir los avances realizados con todos los usuarios del mundo, fomentando el *hardware* y *software* libre, y a su vez obtener la ayuda de las personas interesadas por la información, intercambiando emails y compartiendo experiencias, obteniendo conocimientos de manera recíproca. Dicha documentación se basa en un tutorial en ingles de uso para los usuarios del producto como se puede observar en la captura de pantalla de la Figura 120.



e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi [Biometric / Medical Applications] [Buy now](#)

In August 2013 Cooking Hacks launched the new version of the first biometric shield for Arduino and Raspberry Pi: the e-Health Sensor Platform V2.0. Thanks to the feedback given by the Community and several projects that have been created with it, we have improved the e-Health platform with new features such as:

- New muscle sensor
- New blood pressure sensor
- Upgraded glucometer
- New Connection possibilities

In this tutorial we will explain how to work with e-Health Sensor Platform V2.0. If you have e-Health V1 you can see how to use it on this link:

[Tutorial: e-Health Sensor Platform V1.0 for Arduino and Raspberry Pi \[Biometric / Medical Applications\]](#)

The e-Health Sensor Shield V2.0 allows Arduino and Raspberry Pi users to perform biometric and medical applications where body monitoring is needed by using 10 different sensors: pulse, oxygen in blood (SPO2), airflow (breathing), body temperature, electrocardiogram (ECG), glucometer, galvanic skin response (GSR - sweating), blood pressure (sphygmomanometer), patient position (accelerometer) and muscle/electromyography sensor (EMG).

This information can be used to monitor in real time the state of a patient or to get sensitive data in order to be subsequently analysed for medical diagnosis. Biometric information gathered can be wirelessly sent using any of the 6 connectivity options available: Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 and ZigBee depending on the application.

If real time image diagnosis is needed a camera can be attached to the 3G module in order to send photos and videos of the patient to a medical diagnosis center.

Sales 

New Products 

Top Sellers 

XBee Specials 35% Off! 

GEOLOCATION TRACKER GPRS + GPS 

RASPBERRY PI TO ARDUINO

Figura 120 Impresión de pantalla del tutorial online

C.2 Vídeos y presentaciones

Con el fin de dar a conocer el proyecto al mayor número de gente posible se han realizado videos (Figura 121) explicativos en inglés, con diversos ejemplos de aplicación.



Figura 121 Video presentación/demostración de la plataforma e-Health

Para los numerosos eventos en los que se ha presentado la plataforma, se han preparado presentaciones (Figura 122) de ordenador, como material complementario a demostraciones reales.

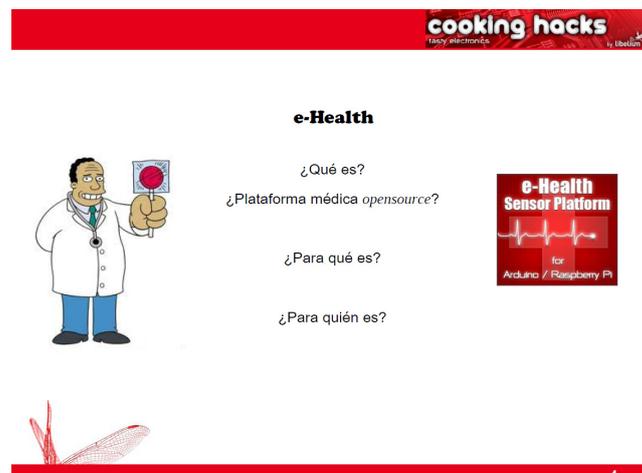


Figura 122 Presentación PowerPoint de la plataforma e-Health

Anexo D Eventos y prensa

D.1 Premios y eventos

Una vez terminado el proyecto se ha buscado dar la mayor difusión internacional posible por lo que se ha presentado a diversos eventos y premios. A continuación se comentarán los más relevantes.

D.1.1 Premios

James Dyson Award 2013

Se trata de un concurso internacional que busca inspirar a la nueva generación de diseñadores e ingenieros. En él se buscan las mejores ideas que solucionen una necesidad.

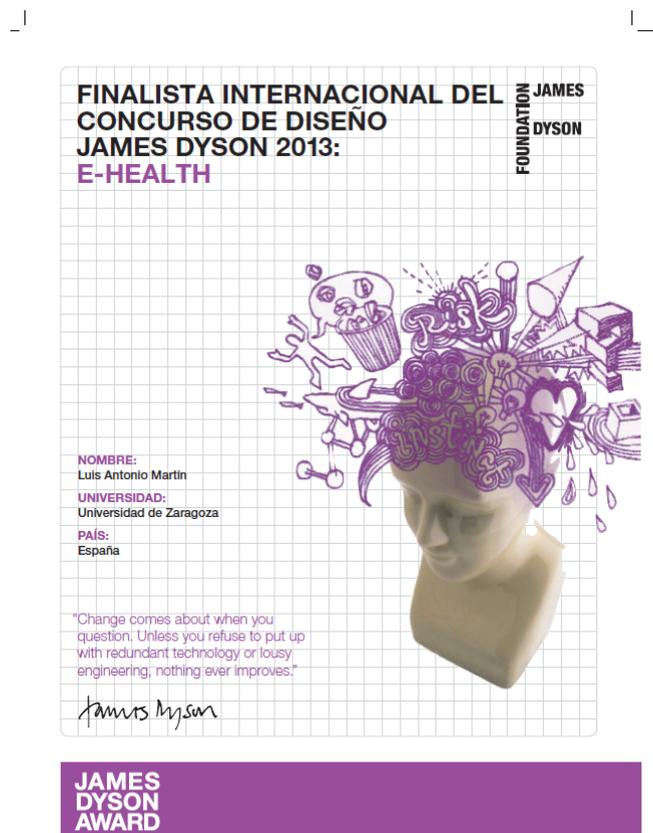


Figura 123 Certificado de finalista internacional del concurso James Dyson Award 2013

Tras una primera fase nacional, e-Health fue seleccionado como uno de los 10 mejores proyectos de España. Esto le abrió las puertas a la fase internacional. En esta fase e-Health fue elegido semifinalista, indicado en la Figura 123, como uno de los 20 mejores proyectos del 2013 de entre más de 500 proyectos, y fue calificado por numerosos miembros de la organización como uno de los 20 mejores inventos del 2013.

IoT Awards 2013 y 2014

e-Health ha sido nominado por dos años consecutivos a los premios IoT Awards en la categoría Open Source Product y Body Connected Product.

X-Project

Ganador del concurso de la empresa asiática DFrobot, que suministra y fabrica placas para Arduino, premiado con una impresora 3D Ultimaker.

D.1.2 Eventos

Maker Faire Roma

La edición europea del fenómeno de las “Ferias para gente que hace cosas”. Lugares en el que se presentan los mejores proyectos DIY y Open Source. Un punto de encuentro para gente que prefiere hacer las cosas que comprarlas. En Octubre de 2013 e-Health fue presentado y se realizaron talleres para dar a conocer sus posibles aplicaciones.

Este viaje sirvió para obtener una muestra de la nueva placa de prototipado Intel Galileo, que sirvió para adaptar el software de e-Health para hacerlo compatible.

Campus Party Londres

En Septiembre de 2013, e-Health fue una de las temáticas en las distintas ponencias que se impartían sobre nuevas tecnologías. Se presentó en un auditorio a modo de demostración.

Internet of things and 3D printing Summer Camp

Evento organizado en Julio de 2013 por Libelium, para fomentar el movimiento Maker en Zaragoza. Se presentó la plataforma a los asistentes en una demostración.

Congreso Universitario Móvil Telcel 2013

Invitados para hablar sobre la tecnología de la empresa Libelium en unas jornadas organizadas en la Universidad de México (Figura 124).



Figura 124 Charla impartida en la Universidad de México

D.2 Prensa

El ser semifinalista en los James Dyson Award, fue un paso muy importante para la difusión del proyecto lo que permitió llegar a mucha gente con la plataforma y explicar su función.

Varios medios contactaron con el fin de preguntar con el proyecto:

- Teknautas (Figura 125), sección tecnológica de ElConfidencial, realizó una entrevista online sobre el proyecto y cuáles son sus objetivos.



Figura 125 Captura de pantalla de la entrevista realizada para Teknautas

- Varios medios de radio se interesaron como: RNE, “La brújula” de Onda Cero, Aragón Radio, COPE...
- Medios impresos como: El Heraldo, El periódico de Aragón...
- El proyecto también se presentó en el programa “Sin ir más lejos” en Aragón TV.

Anexo E Kit de producto

Fruto de la colaboración junto con la empresa Libelium la realización de este proyecto culmina con la comercialización del producto e-Health Sensor Platform.

Dicho producto se venderá en un kit general, o por separado, lo que permite a los usuarios comprar lo que mejor se adapte a su producto o investigación. En la Figura 126 podemos encontrar la imagen corporativa del kit.

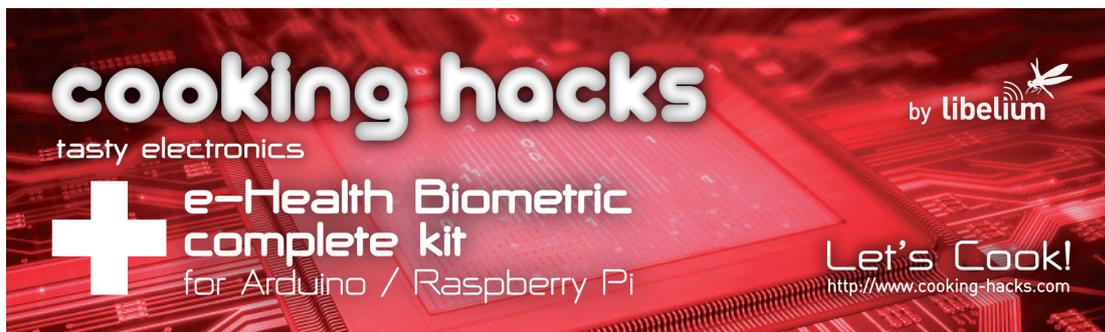


Figura 126 Pegatina para el kit de e-Health Sensor Platform

E.1 Kit completo

A continuación se muestra una imagen de la composición final del kit e-Health, que incluye principalmente:

- 1 x Placa básica e-Health
- 1 x Sensor de pulso y SPO2
- 1 x Sensor de respiración
- 1 x Sensor de temperatura
- 1 x Sensor ECG
- 1 x Sensor de glucosa
- 1 x Sensor de respuesta galvánica de la piel
- 1 x Sensor de tensión sanguínea

- 1 x Sensor de posición del paciente
- 1 x Sensor EMG

Este kit se comercializa a un precio a un precio de 450 euros, incluyendo de manera gratuita el tutorial online de montaje y el software de control (Figura 127).



Figura 127 Caja del kit completo de e-Health Sensor Platform

Anexo F Casos de estudio

F.1 Proyectos terminados

Numerosos proyectos han partido de esta plataforma como base. Entre ellos destacaremos los más interesantes, y de aquellos que hemos recibido información:

OpenPhysio

Se trata de un sistema para el análisis de la interacción afectiva y emocional entre humanos y robots. Incluyendo los sensores de GSR y ECG, permitiéndoles monitorizar el estado de una persona, o el grado de estrés. Realizaron un análisis intensivo de las ondas obtenidas (Figura 128).



Figura 128 Producto final del proyecto OpenPhysio desarrollado en el Fablab Pisa, Italia

Monitoring Physical Reactions

Se trata de un sistema que realiza un análisis de las reacciones físicas de una persona utilizando varios de los sensores de e-Health. Han implementado comunicación vía Ethernet para transmitir los datos a la nube.

F.2 Proyectos en desarrollo

Actualmente se están desarrollando diversos proyectos fin de carrera con esta plataforma, y muchos otros proyectos final de carrera de los que no hay información disponible. En este apartado voy a destacar un proyecto de colaboración que actualmente se está llevando a cabo.

3D Viewing with e-Health

Se trata de un proyecto en colaboración con Pierre Chicoine de California, ingeniero informático especializado en sistemas de visualización gráfica de sistemas en 3D. Ha realizado numerosos proyectos de monitorización de plantas energéticas e hidráulicas.

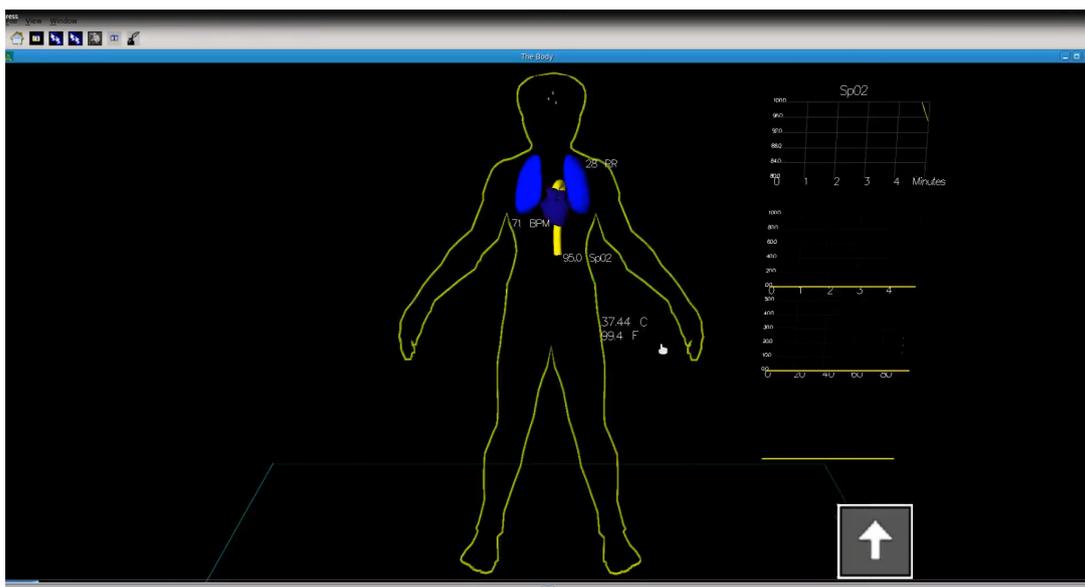


Figura 129 Captura de pantalla del programa de visualización 3D para e-Health en actual desarrollo

El proyecto busca obtener un sistema de representación gráfica de los principales parámetros básicos que monitoriza e-Health, como se observa en la Figura 129.

La colaboración empezó en enero de 2014, y se prevé disponer de un sistema funcional en un par de meses.