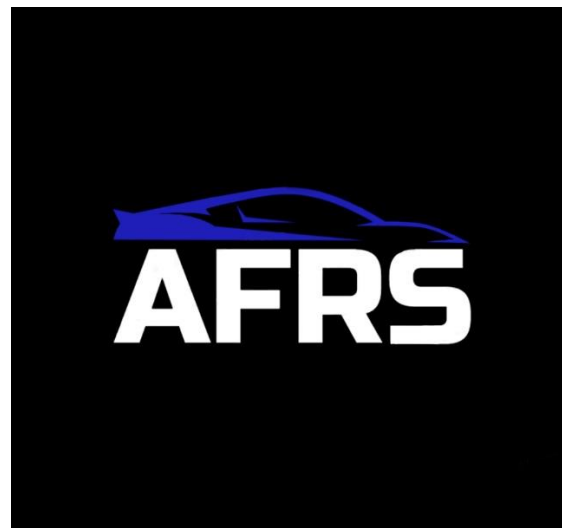




Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica

Informe descriptivo (Carpeta técnica):

Proyecto ARFS



Alumnos:

**Aristimuño Thomas
Canteros Gonzalo Yoel
Rivero Pablo Martín
Rodríguez Guillet Matías Ariel**

**Curso: 7mo 2da Aviónica
Comisión: A Grupo: 2**



Índice:

Capítulo 1: Bibliografía	Pág. 3.
1.1 Objetivo	
1.2 Utilidades	
1.3 Solución a la problemática y su funcionamiento.	
1.4 Cómo prevé el accidente	
Capítulo 2: Sensores y su funcionamiento	Pág. 5.
2.1 Ecosistema del KIT AFRS	
2.2 Sensor de alcohol MQ3	
2.3 Sensor de ultrasonido HC-SR04	
2.4 Cámara web	
Capítulo 3: Rendimiento del Sistema	Pág. 10.
3.1 ¿Como funciona el sistema entero?	
Capítulo 4: Programas, Lenguaje y Sistemas Utilizados	Pág. 11.
4.1.1 TinkerCad	
4.1.2 ¿Qué es y para qué sirve TinkerCad?	
4.1.3 ¿Cuáles son las principales características de TinkerCad?	
4.2.1 Proteus	
4.2.2 ¿Qué es y para qué sirve Proteus?	
4.2.3 ¿Cómo es el trabajo con Proteus?	
4.3.1 ¿Qué es y para qué sirve Python?	
4.3.2 ¿Dónde se utiliza Python?	
4.4.1 ¿Qué es y para qué sirve OpenCV?	
4.4.2 ¿Cuáles son las ventajas de OpenCV?	
4.5.1 ¿Qué es la función de Haars Cascade?	
Capítulo 5: Códigos y sus Características	Pág. 29.
4.1 Código ultrasonido	
4.2 Código MQ3 (Sensor de alcoholemia)	
4.3 Código completo (Sensor ultrasónico, sensor MQ3 y alarmas)	
4.4 Raspberry pi 3	
Capítulo 6: PCB	Pág. 35.
5.1 PCB de Alimentación.	
5.2 PCB del Buzzer, LEDs y RTC	
Links.	Pág. 38.



Capítulo 1: Bibliografía

1.1 Objetivo:

Nuestro objetivo con el sistema que estamos presentando es la reducción de los accidentes que surgen diariamente en la vía pública. Esto lo podemos hacer mediante un kit que se instala en el transporte, el cual mediante sensores como el de alcoholemia, de movimiento y un reconocimiento facial, pueda determinar si el conductor del vehículo se encuentra en condiciones de seguir manejando.

1.2 Utilidades:

Las utilidades de nuestro sistema serían ampliamente aplicables a diferentes entornos automovilísticos, por ejemplo, una empresa necesita transportar un cargamento mediante camiones, este dispositivo se aplicará al transporte que utilizan en el trabajo, en este caso, si el camionero está con sueño, el detector facial lo reconoce y alerta sobre la situación. El equipo puede ser usado en taxis, remises, transporte público, camiones, empresas que se dediquen al transporte de pasajeros como líneas de colectivos o colectivos de larga distancia, hasta en un auto personal, pero todas cumplen la misma función y es la de resguardar la vida del conductor como también, el peatón de la vía pública.

1.3 Solución a la Problemática y su Funcionamiento:

El funcionamiento del equipo se basa en una serie de sensores que estarán conectados en sincronía para descartar posibilidades. Nuestro principal sistema sería el reconocimiento facial. Este programa nos permite detectar diferentes gestos del conductor con una cámara que estaría instalada en el interior de la cabina o donde permita visualizar al conductor, dándonos paso para determinar si está apto para la conducción del vehículo. Su funcionamiento se basa en realizar capturas de imagen mediante la cámara para luego analizarlas con el reconocimiento facial (todas estas mediciones serían con el fin de testear las reacciones del individuo). Otra función que tendría el programa sería el detectar los gestos de la persona, con esto nos referimos a que sí está bien posicionada en el asiento, maneja con los ojos cerrados, etc. Cuenta con sensores de movimientos en el asiento colocados en la reposera de la cabeza para detectar algún cabeceo de la persona (en el caso de que se duerma) o bien que no está prestando atención al manejo. Otro sensor que queremos implementar es el de detector de alcohol en el ambiente, es un módulo que mide la cantidad de alcohol en el oxígeno de la cabina del automotor. Si el valor que recibe el módulo sensor es elevado o fuera de los límites activará la alarma.

Todas estas advertencias se mostrarán en un display que mostrará valores, estadísticas, entre otras cosas. Como somos conscientes de que los accidentes pueden seguir ocurriendo, lo que tenemos pensado implementar es una “caja negra” como un sistema auxiliar que facilita la investigación en el caso de un siniestro. El funcionamiento de esta caja negra es que archive todos los valores, alteraciones, datos, en sí, que deje respaldado todo lo que pudo verificar e



interpretar el sistema del trayecto o trayectos que hizo el vehículo. Por ende, las grabaciones de la cámara del exterior como la del interior de la cabina del automóvil quedarán registradas, datos como el recorrido, la velocidad a la que tránsito por los mismo también.

1.4 Como prevé el accidente:

Cuando el sistema detecta una anomalía en el manejo del conductor, activará una alarma sonora de advertencia, con el fin de que se reposicione el conductor, si el conductor no responde a la advertencia, se aumentará el nivel de sonido de la alarma, haciéndola más notoria en el ambiente, además de mostrar otra advertencia en la pantalla con el error que se detectó. Si el conductor no cede a frenar el vehículo se activarán las balizas del auto para señalar a los demás vehículos de la vía pública que puede haber un movimiento extraño, el sonido de la alarma sería más fuerte todavía.

...Como función más avanzada sobre la idea, investigaremos sobre cómo podemos tomar el control de la dirección y de los frenos (de ser posible) para reducir más todavía los riesgos, además de incorporarle al sistema la función de ser controlado mediante una pantalla táctil que también servirá como un sistema de alerta visual para el conductor...



Capítulo 2: Sensores y Funcionamiento

2.1 Ecosistema del kit AFRS:

Nuestro kit está conformado de sensores ultrasónicos, un sensor de alcoholemia, una cámara y como computadora de abordo una Raspberry Pi 3.

A continuación, pasaremos a explicar cómo funcionan los sensores, su función dentro del proyecto.

2.2 Sensor de Alcohol MQ-3:

El sensor detector de alcohol MQ-3 es capaz de medir la concentración de alcohol en el aliento de una persona, como en el alcoholímetro, tiene una gran respuesta y buena sensibilidad, la cual es ajustable por medio del potenciómetro.

El Módulo MQ-3 tiene una conductividad baja cuando el aire es limpio. cuando el sensor detecta gases como es el alcohol la conductividad del sensor es más alta junto el aumento de la concentración de gas. Se utiliza un electro circuito para que convierte el cambio de conductividad para que corresponda con la señal de salida de la concentración de gas.

Utilización que le damos:

Implementamos el sensor MQ-3 de forma que nos pueda decir cuando una tiene una alta concentración de alcohol en su aliento. De esta forma, regulando con el potenciómetro que tiene incorporado, cuando calculamos, le damos un límite de sensibilidad en el cual se basa para poder tener la respuesta deseada.

Especificaciones y Características:

- Voltaje de operación: 5V DC.
- Consumo: < 750mW.
- Salida Analógica (variable de acuerdo a la cantidad de alcohol detectada)
- 2 pines de salida (Salida analógica y Salida de nivel TTL).
- Salida de nivel TTL válida de bajo nivel, se puede conectar directamente al microcontrolador.
- Salida analógica de 0 a 5 V (el voltaje más alto equivale a una concentración más alta).
- Condiciones de trabajo: Temperatura ambiente: -10°C to 65°C, Humedad: ≤ 95% RH.
- Salida analógica (Variable de acuerdo a la cantidad de alcohol detectada).



2.3 Sensor de Ultrasonido HC-SR04:

El sensor HC-SR04 es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 50 cm (puede variar según el fabricante). Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web. De igual forma es el más empleado en proyectos de robótica como robots laberinto o sumo, y en proyectos de automatización como sistemas de medición de nivel o distancia.

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación. El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido (40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar).

La distancia se puede calcular utilizando la siguiente formula:

$$\text{Distancia (m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s}) / 2$$

Utilización que le damos:

Utilizaremos dos sensores ultrasónicos, los cuales se ubicarán en la butaca del vehículo. Uno estará en la altura del cuello y otro en la cabecera, la función de estos sería reconocer patrones de somnolencia o fatiga en el conductor (tambaleos, movimientos bruscos o anormales).

Especificaciones y Características:

- Voltaje de Operación: 5V DC.



- Corriente de reposo: < 2mA.
- Corriente de trabajo: 15mA.
- Rango de medición: 2cm a 450cm.
- Precision: +- 3mm.
- Ángulo de apertura: 15°
- Frecuencia de ultrasonido: 40KHz.
- Duración mínima del pulso de disparo TRIG (nivel TTL): 10 μ S.
- Duración del pulso ECO de salida (nivel TTL): 100-25000 μ S
- Dimensiones: 45mm x 20mm x 15mm
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)



2.4 Cámara Web:

La podemos identificar como webcam o cámara web. En esencia, es un tipo de cámara que está conectada a una computadora, bien sea por vía directa o inalámbrica, la cual está condicionada para captar imágenes que van a ser visualizadas de forma remota.

Especificaciones técnicas:

- Videoconferencias HD (1280 x 720 pixeles) con el sistema recomendado.
- Captura de video hasta 1280 x 720 pixeles.
- La velocidad de toma de frames es de 30FPS.
- Fotos hasta 3 megapíxeles.
- Micrófono integrado con tecnología Logitech RightSound.

Utilización que le damos:

La cámara web será utilizada para poder tener una mayor eficiencia y precisión en nuestro sistema de somnolencia, se basa en un reconocimiento facial que creamos a base de un programa de edición artificial llamado "OpenCV" que nos permite identificar parámetros en las imágenes además utilizamos unos códigos llamados "Haarcascade" para poder reconocer tanto el rostro y los ojos, de esta forma podemos medir los parámetros de las personas mediante los gestos del mismo.



2.5 Raspberry Pi 3:

La Raspberry Pi 3 es un microcontrolador que mantiene 1gb de memoria RAM, utiliza el procesador BCM2837 ARMv8, igualmente de 4 núcleos, pero que funciona a 1.2 GHz. Este incluye Wifi y bluetooth 4.1 de base, sin necesidad de adaptadores, ampliando los horizontes de su usabilidad mucho más allá.

Es un ordenador simple compuesto por un SoC, CPU, memoria RAM, puertos de entrada y salida de audio y vídeo conectividad de red, ranura SD para almacenamiento, reloj, una toma para la alimentación, conexiones para periféricos de bajo nivel, reloj...prácticamente lo mismo que si miras la parte de atrás de la torre de un ordenador, porque la Raspberry es un ordenador

Utilización que le damos:

La Raspberry Pi 3, es el ordenador de nuestro kit AFRS. Es la computadora de abordo la cual hace que nuestro sistema funcione y podamos trabajar con todos los sensores y la cámara web al mismo tiempo.

Especificaciones técnicas:

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
- RAM: 1GB LPDDR2 SDRAM.
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE.
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps).
- GPIO de 40 pines.
- HDMI.
- 4 puertos USB 2.0
- Puerto CSI para conectar una cámara.



Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica

- Puerto DSI para conectar una pantalla táctil.
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE)





Capítulo 3: Rendimiento del Sistema

3.1 ¿Como funciona el sistema entero?

El funcionamiento en conjunto va a seguir un patrón desde el momento que el conductor ingresa al vehículo:

Primero tendremos el sistema de alcohol, donde el conductor tiene que soplar por la boquilla donde se encuentra el sensor MQ-3. Este último sensor se va a encargar de analizar el alcohol en sangre, según el resultado va a proceder al siguiente paso:

- **Positivo:** Si se detecta que la persona a consumido alcohol sonará una alarma de advertencia para que no maneje en la vía pública, y el sistema se detendrá.
- **Negativo:** Si no se detecta alcohol en sangre de la persona, va a proceder a activar los sensores de distancia y la cámara.

Los sensores ultrasónicos que están ubicados a la altura del cuello y a la altura de la cabecera, nos servirán para tomar la distancia entre la cabeza del conductor y el asiento. Solamente alertara cuando este detecte patrones de somnolencia del conductor, como puede ser tambaleo, movimientos bruscos y cabeceos.

Al detectar patrones de somnolencia este activará una alarma sonora para que el conductor se orille y tome un descanso hasta que este apto para manejar.

Si no detecta ninguno de los parámetros mencionados el sistema seguirá y pasará a la cámara.

La cámara estará especialmente enfocada al rostro del conductor, la cual aportará una mayor eficiencia al sistema de somnolencia ya que comparara el rostro de una persona despierta con la del conductor, si detecta parámetros erróneos como pueden ser bostezos, ojos cerrados o que el conductor se empieza a molestar con la mano activara la alarma sonora de que se está durmiendo.

En cambio, si no se muestran estos tipos de parámetros o patrones se volverá al sensor ultrasónico y así, en un constante bucle entre el sensor de medición y la cámara, solo se detendrá cuando algún patrón o parámetro active la alarma.



Capítulo 4: Programas, Lenguaje y Sistemas Utilizados

4.1 TinkerCad:

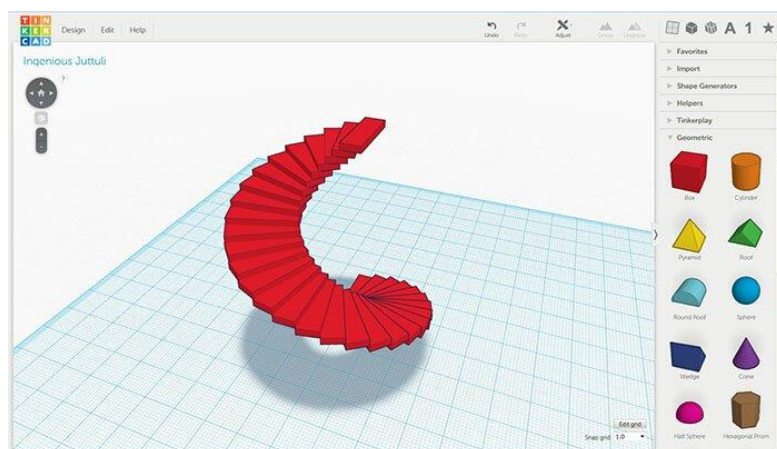
Nosotros utilizamos este programa con el fin de poder proyectar y diseñar las piezas que nos servirá para sostener nuestros sensores, mantener seguras nuestras placas y a la Raspberry

4.2 ¿Qué es y para qué sirve TinkerCad?

TinkerCAD es una herramienta online ofrecida por Autodesk la cual se utiliza de forma gratuita y solo requiere crearse una cuenta de usuario. De entre sus utilidades, probablemente la más conocida es la de diseñar piezas en 3D. Sin embargo, ofrece también una posibilidad realmente interesante y es la de montar, programar y simular circuitos con Arduino.

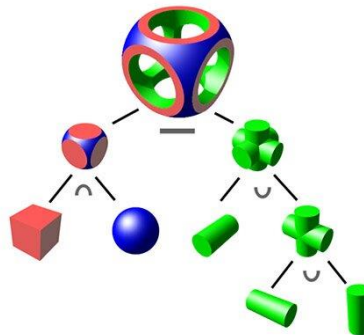
TinkerCad es una colección online que incluye herramientas de software CAD se basa en la geometría solida constructiva (CSG), que permite a los usuarios crear modelados 3D complejos mediante la combinación de objetos más simples, este es un software fácil de usar y actualmente es disfrutado por muchos, particularmente estudiantes, profesores y personas que se dediquen a la materia.

TinkerCAD es una buena alternativa a otro software de modelado 3D como Sketchup o Fusion360, otra solución de Autodesk, si no necesita las opciones más opciones más avanzadas de estas soluciones. En realidad, la compañía líder de software adquirió TinkerCAD en 2013, dos años después de su lanzamiento por el ex ingeniero de Google Kai Backman y su co-fundador Mikko Mononen. La principal ventaja sobre estos dos programas es que es gratuito y sin embargo ofrece más libertad de modelado de lo que parece.



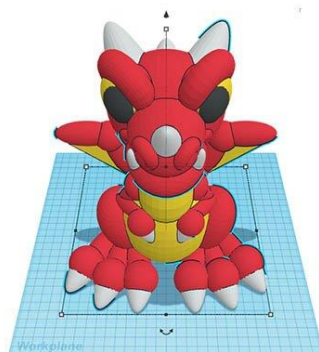
4.3 ¿Cuáles son las principales características de TinkerCad?

Aunque TinkerCAD es perfecto para principiante no significa que aquellos que tienen más experiencia con el modelado 3D no apreciaran este Software, Dado que se basa en CSG para crear modelos sólidos, siempre podemos dificultar el modelo agregando más formas. Además, el software permite agregar circuitos electrónicos a los diseños 3D para crear objetos con luz y movimiento. El resultado final incluso se puede simular en el software para verificar como responderán los componentes en la vida real. Otra capacidad de TinkerCAD es transformar un diseño 3D en modelos de ladrillo construibles, similar a la creación de legos. Finalmente, para aquellos que aman Minecraft también incluye la capacidad de crear diseños compatibles con la aplicación.



Por lo tanto, TinkerCAD se puede utilizar para una gran variedad de aplicaciones, incluida la impresión 3D. Los modelos 3D se pueden guardar en tres formatos diferentes, STL, OBJ y SVG. Una vez que tengamos el archivo STL del modelo, continuaremos utilizando un software laminado. El software de corte convierte el modelado 3D en una serie de capas delgadas y produce un archivo de código G que contiene instrucciones adaptadas a un tipo específico de impresora. En otras palabras, está dividiendo el objeto en una pila de capas planas y describe estas capas como movimientos lineales del extrusor de la impresora 3D.

Los usuarios a menudo complementan lo intuitivo que es este software CAD. La transformación, la duplicación y la modificación de la forma son fáciles de entender. Además, AutoDesk ha puesto a disposición de su comunidad muchos recursos. Entre otras cosas encontraras inspiración, consejos y trucos para comenzar, en el blog de la empresa.

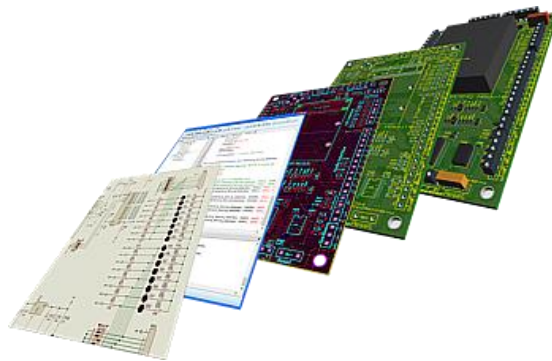


4.4 Proteus:

La implementación del programa es con el fin de que podamos hacer los esquemáticos, y el PCB de nuestras placas. De esta forma, podemos tener (una vez que se termine el proceso).

4.5 ¿Qué es y Para qué sirve Proteus?

Proteus es una aplicación para la ejecución de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño del esquema electrónico, programación del software, construcción de la placa de circuito impreso, simulación de todo el conjunto, depuración de errores, documentación y construcción.



Sin la utilización de la suite Proteus, el proceso para construir un equipo electrónico basado en un microprocesador se compone de cinco etapas. Sólo al final del proceso somos capaces de detectar los errores y cualquier problema exige volver a ejecutar el ciclo completo:





Escuela de Educación Técnica Nº7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica

El depurado de errores puede convertirse en una labor ardua en tiempo y recursos, lo que conlleva un alto coste económico. Sin embargo, con la herramienta Proteus el proceso queda definido de la siguiente manera:

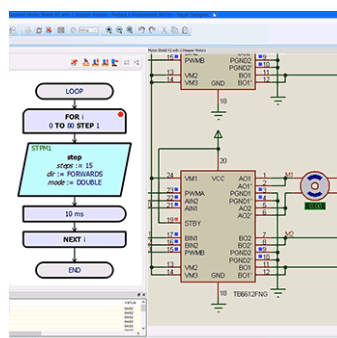


Las ventajas saltan a la vista. Con Proteus las fases de prueba no suponen la necesidad de volver a construir nuevos prototipos, con el ahorro de costos y tiempo que ello supone.

Los diferentes módulos que componen Proteus se pueden adquirir de forma independiente añadiendo nuevas funcionalidades a medida que aumentan nuestras necesidades de desarrollo y producción. Además, la capacidad de simular cada una de las familias de microprocesadores también es objeto de adquisición por separado. De esta manera podemos empezar adquiriendo unas funcionalidades básicas e ir adquiriendo progresivamente nuevas características aprovechando al máximo nuestras inversiones en la herramienta y asegurar al máximo los costes de inversión en el software.

En el mundo de la formación, Proteus se muestra como una herramienta magnífica porque permite al alumno realizar modificaciones tanto en el circuito como en el programa, experimentando y comprobando de forma inmediata los resultados y permitiéndole de esta forma aprender de forma práctica y sin riesgos de estropear materiales de elevado coste.

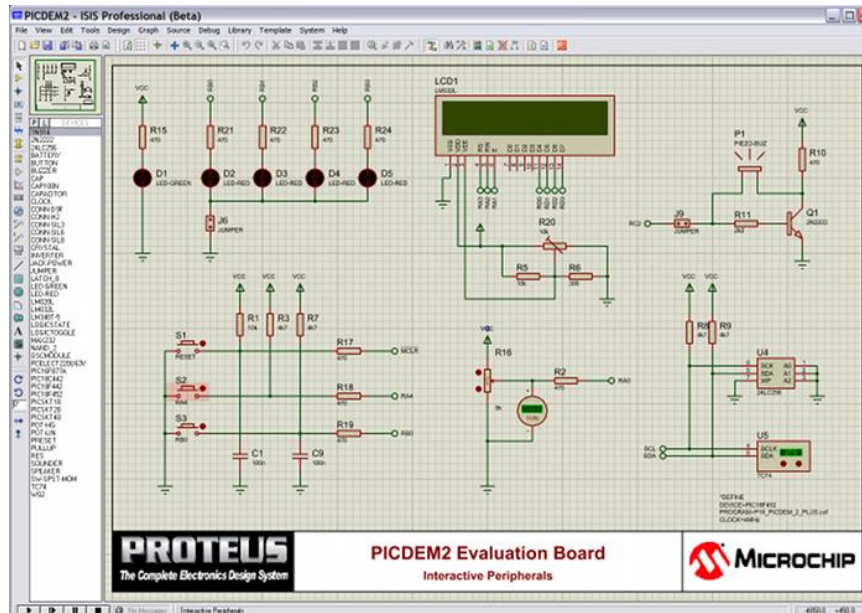
Si se desea simular el funcionamiento electrónico del circuito, el funcionamiento lógico del programa cargado en el microprocesador, construir la placa de circuito impreso, documentar todo el proceso y obtener vistas en tres dimensiones, Proteus le ofrece una herramienta completa a un precio competitivo.



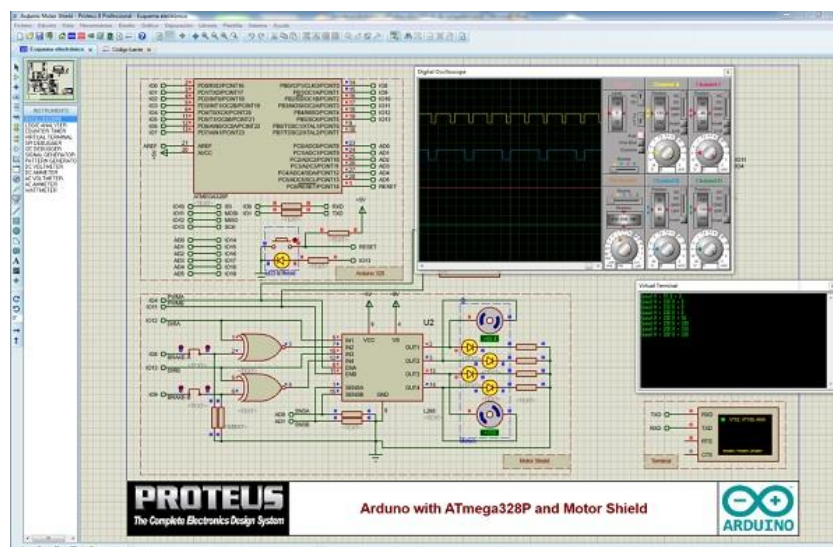


4.6 ¿Como es el trabajo con Proteus?

El primer paso en nuestro trabajo de construcción de un equipo electrónico consiste en dibujar el esquema utilizando la pestaña 'esquema electrónico'



Durante la elaboración del esquema se pueden llevar a cabo simulaciones avanzadas que nos ayudan a comprobar el correcto funcionamiento de nuestro proyecto, mediante el uso de las herramientas ProSPICE (motor de simulación de circuitos electrónicos) desde la propia pestaña 'Esquema electrónico'.



Podemos utilizar en este proceso todas las ayudas que Proteus pone a nuestra disposición: instrumentos virtuales, inspectores de buses de datos, modelos animados de dispositivos electrónicos, generadores, sondas, etc.



AVR Studio IDE showing the assembly code for the 'avr1602c' program. The code is for an AVR 1602C LCD module. It includes comments in Spanish and C++ code for initializing the LCD and displaying 'Temperatura'. The assembly code is shown in the main window, with the corresponding C++ code visible in the left margin. The right margin shows the AVR 1602C LCD module pinout and the 'LCD PCB Board'.

```

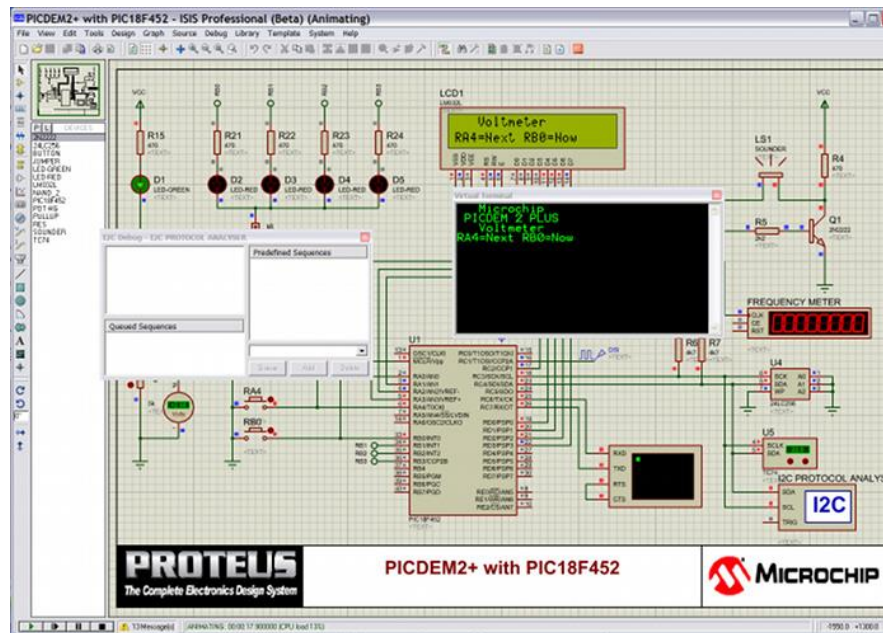
;-----
; AVR 1602C - LCD
;
; Este programa muestra la inicialización y el uso de un módulo de pantalla LCD 1602C.
;
; Autor: [Nombre]
; Fecha: [Fecha]
;
;-----
; C++ Code
;
; #include <avr/io.h>
; #include <avr/interrupt.h>
; #include "avr1602c.h"
;
; void main()
; {
;     // Inicialización de la pantalla LCD
;     lcd_init();
;
;     // Muestra la temperatura
;     lcd_puts("Temperatura");
; }
;
;-----
; Assembly Code
;
; .include "avr1602c.inc"
;
; .def _main
; .code
;
;     ; Inicialización de la pantalla LCD
;     lcd_init
;
;     ; Muestra la temperatura
;     lcd_puts "Temperatura"
;
; .end
;
;-----

```

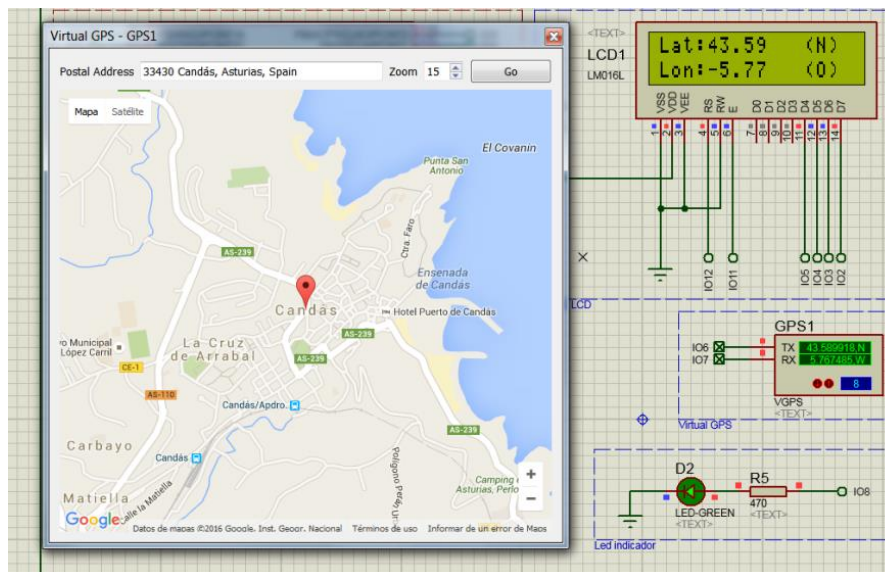
16



**Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica**

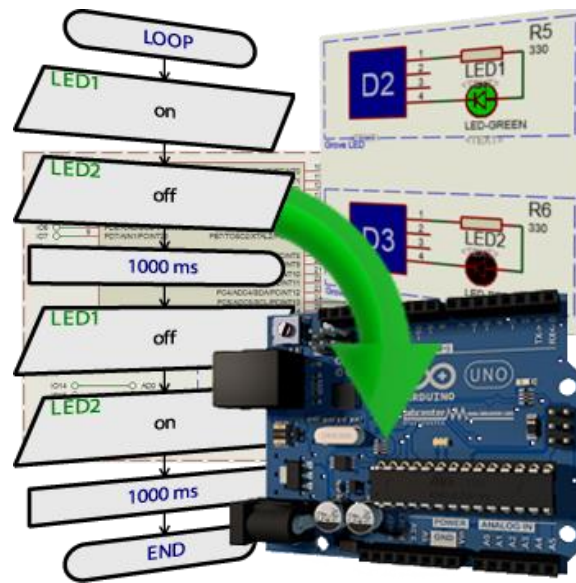


Proteus nos permite usar potentes entornos virtuales (espacio de trabajo para el desplazamiento de robots-tortuga, mapas de situación GPS, etc.) para que nuestra experiencia de simulación sea lo más parecida a su comportamiento en la vida real en la que debe funcionar nuestro equipo.

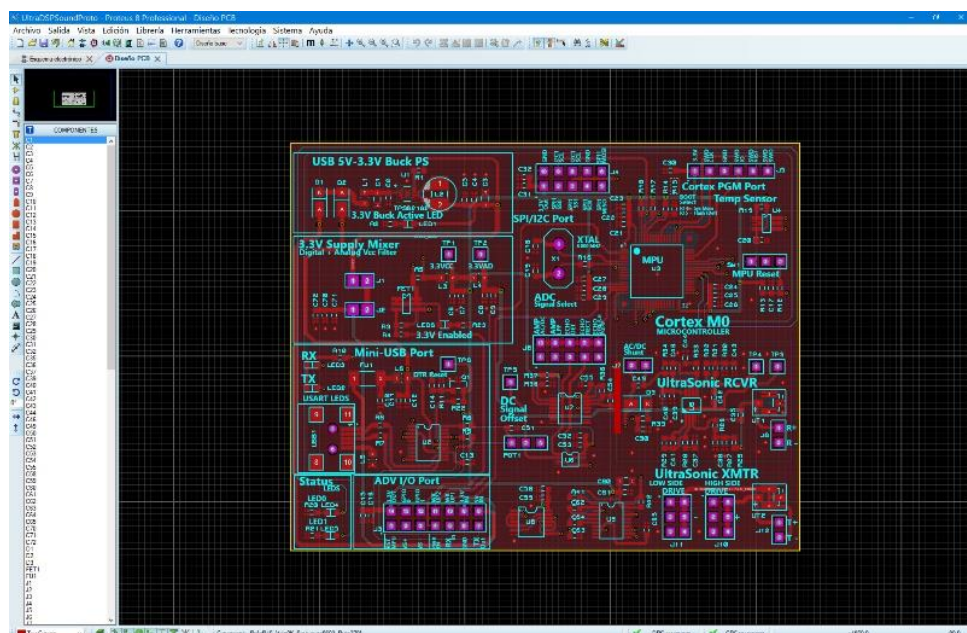




Escuela de Educación Técnica N°7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica

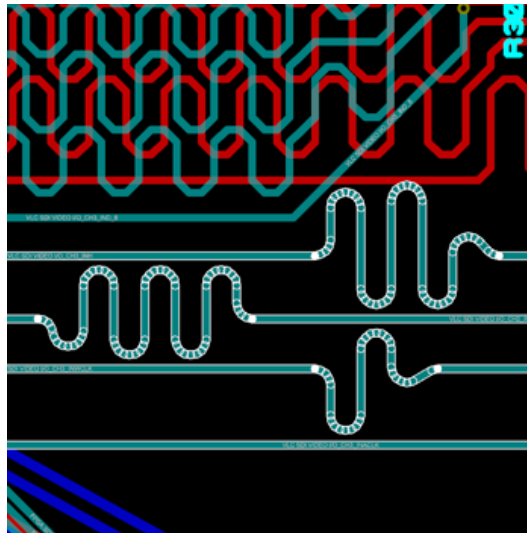


Cuando el diseño electrónico y el software funcionan a nuestra satisfacción podemos pasar a la etapa de construir el circuito impreso. Proteus genera de forma automática la lista de redes (NETLIST) a partir del diseño electrónico. Una red es un grupo de pines interconectados entre sí y la lista de redes es un listado que contiene la información completa de todas las redes que forman nuestro diseño. Desde la pestaña 'Diseño PCB' podemos proyectar nuestra placa de circuito impreso partiendo de la información de la lista de redes derivada de nuestro diseño electrónico. De esta forma, nos aseguramos que la placa tendrá unidos entre sí, con las pistas correspondientes, los pines de los distintos componentes electrónicos de forma idéntica a como los hemos definido en nuestro esquema electrónico. Podemos crear las pistas de nuestra placa de forma manual o aprovechar las potentes capacidades de enrutado automático que ofrece Proteus.

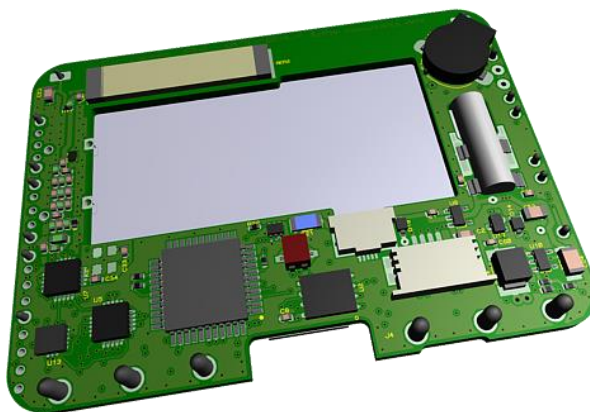




Proteus hace muy sencilla la tarea de diseñar el circuito impreso con potentes ayudas: auto-enrutado, pistas en serpentina, conexión de pistas con huellas usando la técnica de lágrima, generación automática de superficies de disipación, definición de placas multicapas y profundidad de taladros, etc.



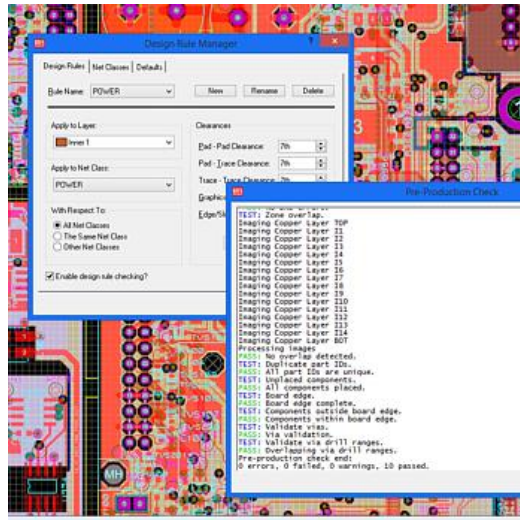
Al terminar el diseño de nuestra placa de circuito impreso, podemos obtener una imagen en tres dimensiones de nuestro diseño utilizando la pestaña 'visor 3D', que nos resultará muy útil para presentar a nuestros clientes una visión preliminar del resultado de nuestro trabajo.



Podemos comprobar que todas las reglas del diseño (las propiedades eléctricas, espacios de separación, etc. de cada uno de los diferentes tipos de pista) que hemos definido para construir nuestra placa se han ejecutado y cumplido correctamente utilizando la pestaña 'Visor Gerber'



Escuela de Educación Técnica N°7 Taller Regional Quilmes Prácticas Profesionalizantes: Especialidad Aviónica



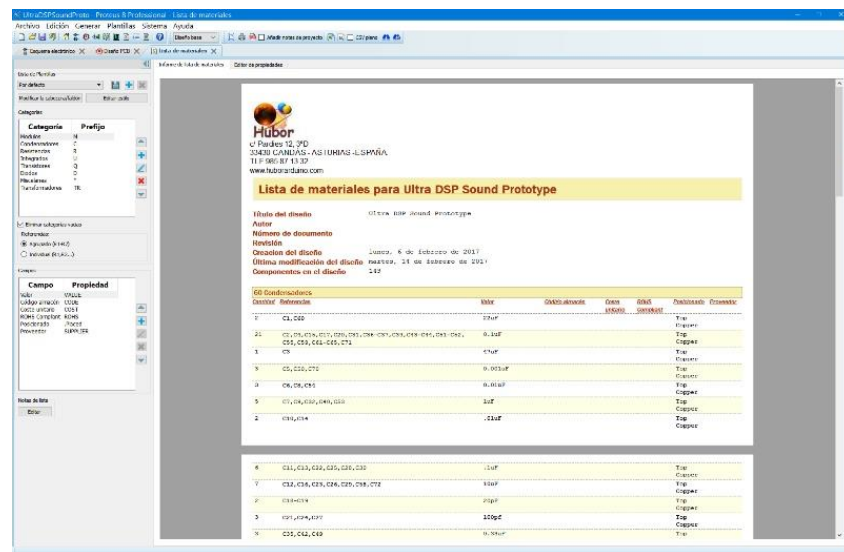
Y llevar a cabo la generación de los ficheros de fabricación de los circuitos impresos en diferentes formatos ampliamente estandarizados y utilizados por las compañías del sector.



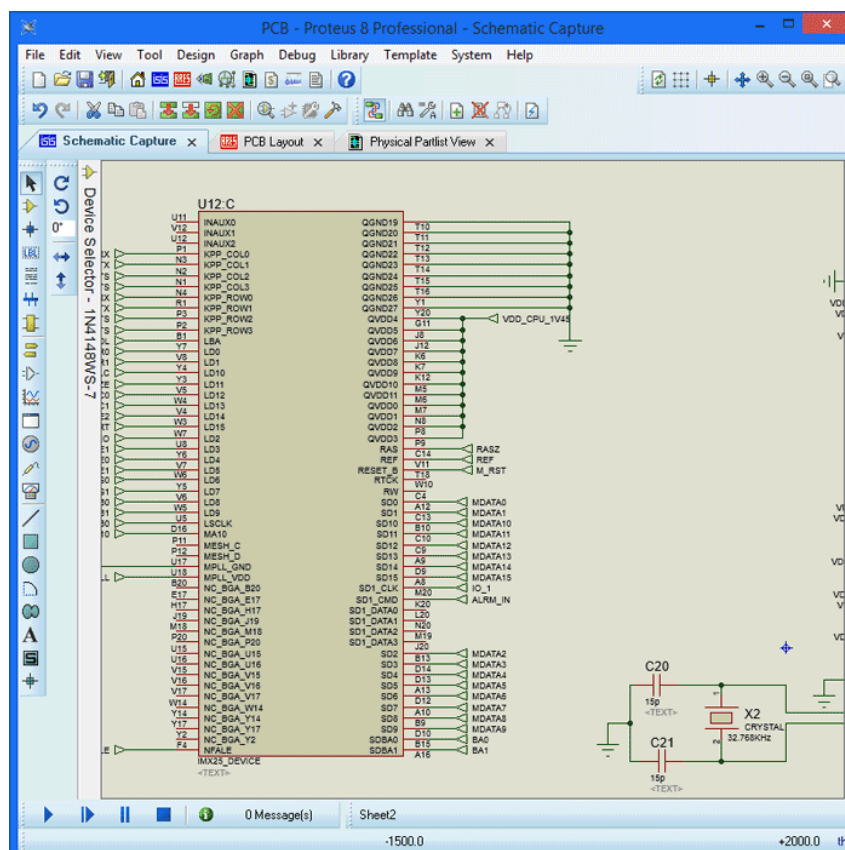
Desde la pestaña 'Lista de materiales' podremos obtener la lista de materiales a partir del diseño electrónico.



Escuela de Educación Técnica N°7 Taller Regional Quilmes Prácticas Profesionalizantes: Especialidad Aviónica



Desde el módulo 'Explorador del diseño' podemos gestionar todos los componentes que forman nuestro diseño y ubicarlos rápidamente en cualquiera de las pestañas que componen el conjunto de herramientas Proteus.



Este módulo también nos permite gestionar y controlar diferentes versiones de nuestro diseño. Por ejemplo, gestionar versiones diferentes con uno, dos o tres puertos de salida. También podemos indicar a nuestro departamento de

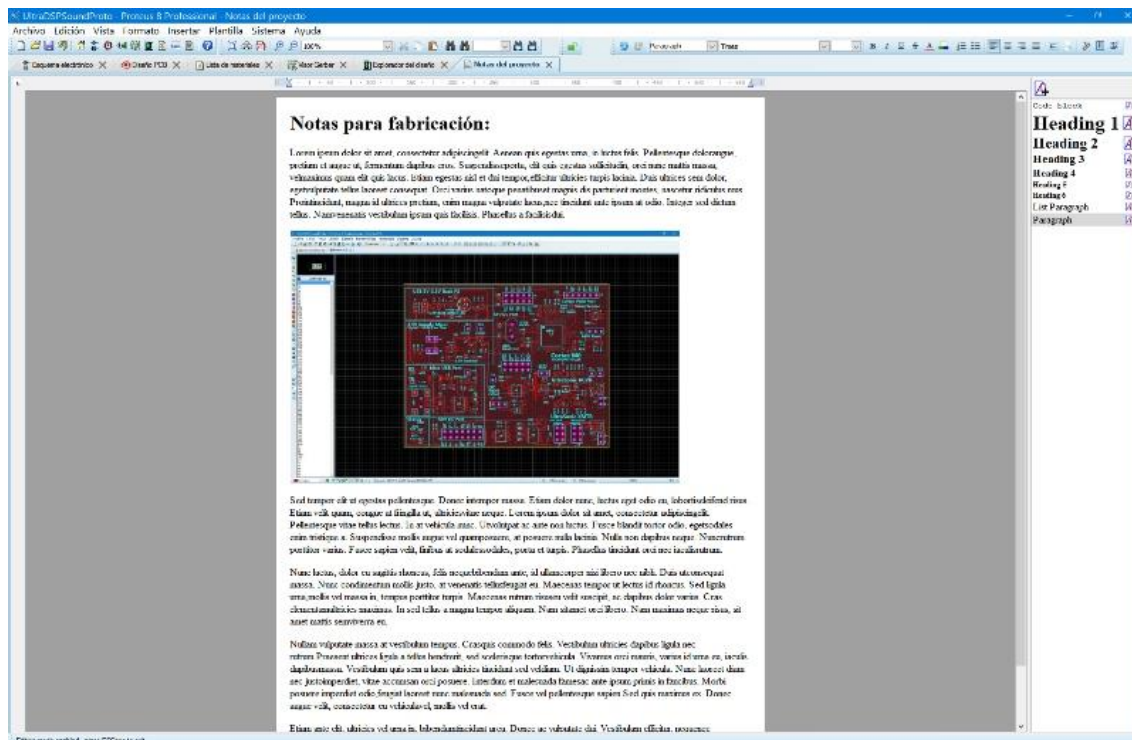


Escuela de Educación Técnica N°7 Taller Regional Quilmes Prácticas Profesionalizantes: Especialidad Aviónica

producción los materiales que deben proveerse según la versión que se vaya a construir desde el módulo 'Lista de materiales'.

Referencia	Tipo	Valor	Empaquetado
C1 (22uF)	CAP	22uF	0603
C2 (10uF)	CAP	10uF	0603
C3 (47uF)	CAP	47uF	0603
C4 (100uF)	CAP	100uF	0603
C5 (0.01uF)	CAP	0.01uF	0603
C6 (0.01uF)	CAP	0.01uF	0603
C7 (1uF)	CAP	1uF	0603
C8 (0.01uF)	CAP	0.01uF	0603
C9 (1uF)	CAP	1uF	0603
C10 (0.01uF)	CAP	0.01uF	0603
C11 (1uF)	CAP	1uF	0603
C12 (10uF)	CAP	10uF	0603
C13 (1uF)	CAP	1uF	0603
C14 (0.01uF)	CAP	0.01uF	0603
C15 (0.01uF)	CAP	0.01uF	0603
C16 (0.01uF)	CAP	0.01uF	0603
C17 (0.01uF)	CAP	0.01uF	0603
C18 (0.01uF)	CAP	0.01uF	0603
C19 (0.01uF)	CAP	0.01uF	0603
C20 (0.01uF)	CAP	0.01uF	0603
C21 (0.01uF)	CAP	0.01uF	0603
C22 (1uF)	CAP	1uF	0603
C23 (0.01uF)	CAP	0.01uF	0603
C24 (0.01uF)	CAP	0.01uF	0603
C25 (0.01uF)	CAP	0.01uF	0603
C26 (0.01uF)	CAP	0.01uF	0603
C27 (0.01uF)	CAP	0.01uF	0603
C28 (0.01uF)	CAP	0.01uF	0603
C29 (0.01uF)	CAP	0.01uF	0603
C30 (0.01uF)	CAP	0.01uF	0603
C31 (0.01uF)	CAP	0.01uF	0603
C32 (0.01uF)	CAP	0.01uF	0603
C33 (0.01uF)	CAP	0.01uF	0603
C34 (0.01uF)	CAP	0.01uF	0603
C35 (0.01uF)	CAP	0.01uF	0603
C36 (0.01uF)	CAP	0.01uF	0603
C37 (0.01uF)	CAP	0.01uF	0603
C38 (0.01uF)	CAP	0.01uF	0603
C39 (0.01uF)	CAP	0.01uF	0603
C40 (0.01uF)	CAP	0.01uF	0603
C41 (0.01uF)	CAP	0.01uF	0603
C42 (0.01uF)	CAP	0.01uF	0603
C43 (0.01uF)	CAP	0.01uF	0603
C44 (0.01uF)	CAP	0.01uF	0603
C45 (0.01uF)	CAP	0.01uF	0603

La gestión de toda la documentación asociada al proceso de construcción de nuestro equipo electrónico puede ser gestionada desde la pestaña 'Notas del proyecto'. Proteus pone a nuestra disposición un potente editor de texto que permite realizar una completa documentación de nuestros proyectos con información para los diseñadores, los encargados de la producción de los circuitos impresos, los responsables del montaje de los equipos, el departamento de compra de material o los servicios de control de calidad.



4.7 ¿Qué es y para qué sirve Python?

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina.

Python es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. Con el paso del tiempo, Python ha ido ganando adeptos gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge.

Para que puedas ver lo sencillo que es Python, a continuación, tienes un simple programa escrito en este lenguaje, que podrás entender fácilmente incluso si no sabes nada de programación. Se trata de un pequeño aplicativo para calcular un sueldo por horas, algo muy sencillo pero que podría ser perfectamente funcional:

```
horas = float (input ("Introduce tus horas de trabajo: "))  
coste = float (input ("Introduce lo que cobras por hora: "))  
sueldo = horas * coste  
print ("Tu sueldo es", sueldo)
```

En las dos primeras líneas se le pide al usuario que introduzca cuántas horas ha trabajado y lo que cobra por hora. En la tercera se hace la operación para calcular el sueldo total y se guarda en una variable. En la última línea de código se imprime por pantalla el resultado. Si, por ejemplo, ponemos que hemos trabajado 8 horas a 15 €, en pantalla se imprimirá “Tu sueldo es de 120”.





Ahora que ya sabes qué es Python, puede que también quieras saber cómo surgió este increíble lenguaje de programación. A pesar de que pueda parecer algo muy nuevo, Python remonta su origen a principios de los años 90, cuando Guido Van Rossum, un trabajador del Centrum Wiskunde & Informatica (CWI), un centro de investigación holandés, tuvo la idea de desarrollar un nuevo lenguaje basándose en un proyecto anterior, el lenguaje de programación “ABC”, que él mismo había desarrollado junto a sus compañeros.

Su filosofía fue la misma desde el primer momento: crear un lenguaje de programación que fuera muy fácil de aprender, escribir y entender, sin que esto frenara su potencial para crear cualquier tipo de aplicación. En aquellos años, el hardware que había no permitía tal cosa, y es por eso por lo que Python ha resurgido durante los últimos años, porque el avance de la tecnología ha permitido alcanzar el objetivo inicial de este lenguaje de programación adelantado a su tiempo.

4.8 ¿Dónde se utiliza Python?

Como ya has visto, Python es un lenguaje de programación multiplataforma, algo que permite desarrollar aplicaciones en cualquier sistema operativo con una facilidad asombrosa. Una gran cantidad de tecnologías se llevan muy bien con Python debido a su sencillez y a su gran potencia para el tratamiento de datos, algo que sin duda ha hecho resurgir este lenguaje a nivel laboral, donde cada vez son más las empresas que solicitan expertos en Python.

Data analytics y big data: El uso de Python está muy extendido en dos áreas que han estado, y estarán, en boca de todos: el análisis de datos y el big data. Su simplicidad y su gran número de bibliotecas de procesamiento de datos hacen que Python sea ideal a la hora de analizar y gestionar una gran cantidad de datos en tiempo real.

Python está siendo utilizado en la actualidad por muchísimas empresas, tanto de forma directa, como indirecta, ya que detrás de los distintos softwares de data analytics, muchas veces está este lenguaje interpretado de alto nivel. Analizar una gran cantidad de datos para transformarlos en información útil para el big data es una de las especialidades de Python.

Data mining: La minería de datos o data mining es un proceso que permite analizar grandes bases de datos con el objetivo de predecir futuras tendencias. Se trata de un proceso complejo al que Python puede arrojar luz a través de la limpieza y organización de datos y del uso de algoritmos de aprendizaje automático que simplifica el análisis de datos.

Data science: Tras la creación de los motores numéricos como “Pandas” o “NumPy”, Python está desbancando MATLAB, un lenguaje utilizado por científicos a la hora de trabajar con un gran número de datos. La razón es la misma que en los anteriores apartados; la sencillez y la potencia para trabajar con un gran número de datos, unidos al gran número de bibliotecas existentes, hacen que Python sea ideal para este tipo de tareas.

Inteligencia artificial: Seguro que durante los últimos años has oído hablar muchísimo de la inteligencia artificial (IA). Gran parte de su avance se debe a Python. Su facilidad de escritura y su robustez han convertido a Python en el aliado perfecto de la IA. Su capacidad de plasmar ideas complejas en pocas



líneas, unidas al gran número de frameworks existentes, han hecho que Python sea uno de los lenguajes de programación que están impulsando a la IA.

Blockchain: La base de datos distribuida Blockchain, conocida mundialmente por ser la base sobre la que se sustentan las criptomonedas, también funciona muy bien junto a Python. Como lenguaje versátil, seguro y rápido, es muy útil para formar cadenas de bloques, e incluso permite a los desarrolladores crear una cadena de bloques sencilla en menos de 50 líneas de código, haciendo sencillo algo muy complejo.

Machine learning: El machine learning o aprendizaje automático es otra de las tecnologías que está cambiando el mundo tal y como lo conocemos. La robótica y la IA son ahora capaces de aprender por sí mismas a medida que van procesando más y más datos. De esta forma, obtienen información cada vez más relevante que les permite tomar las decisiones adecuadas. Por supuesto, Python es también muy eficaz en este campo, en el tratamiento de datos eficaz es esencial.

Desarrollo web: Python también permite desarrollar webs complejas en menos líneas de código, lo que permite que estas sean más ligeras y optimizadas. Django es uno de los frameworks de Python más populares de la actualidad, que puede ser utilizado para crear webs dinámicas y muy seguras. Python es también muy utilizado para hacer scraping, es decir, para obtener información de todo tipo de webs, tal y como lo hacen Netflix, Instagram o Pinterest.

Juegos y gráficos 3D: Python también posee una gran capacidad para manejar gráficos 3D gracias a la gran cantidad de marcos de trabajo y herramientas existentes. PyGame, Blender o Arcade son algunos de los más conocidos. Uno de los juegos más populares desarrollado con Python es Battlefield 2, un juego de acción bélica lanzado en 2005 en el que el motor gráfico, las animaciones y sus distintas funcionalidades fueron desarrolladas con Python.

4.9 ¿Qué es y para qué sirve OpenCV?

¿Qué es OpenCV? Una biblioteca de código abierto que contiene implementaciones que abarcan más de 2500 algoritmos. Además, está especializada en el sistema de visión artificial y machine learning.

No exageramos al decir que OpenCV es la biblioteca de visión artificial más grande del mundo en cuanto a las funciones que posee, lo que explica su popularidad. Aunado al hecho de que, para fines académicos y comerciales, ¡es gratuita!

Es utilizada por:

- ✓ Empresas.
- ✓ Universidades.
- ✓ Miembros del movimiento maker.



Con sus más de 2500 algoritmos a cuestas, debes saber que OpenCV es capaz de acciones como las siguientes:

- ✓ Identificar objetos o caras (reconocimiento facial).
- ✓ Encontrar imágenes similares.
- ✓ Eliminar los ojos rojos de las fotografías.
- ✓ Reconocer escenarios.
- ✓ Seguir los movimientos de los ojos.
- ✓ Clasificar acciones humanas que estén en videos.
- ✓ Extraer modelos 3D.
- ✓ Útil en campos como la robótica y la realidad aumentada.

Con esta herramienta tenemos a la mano una biblioteca para aplicaciones de visión artificial, por lo que su potencial podría acelerar su utilización para fines comerciales próximamente.

4.9.1 ¿Cuáles son las ventajas de OpenCV?

Si hablamos de lo que es OpenCV para el avance de la tecnología, por eso, vamos a destacar sus beneficios:

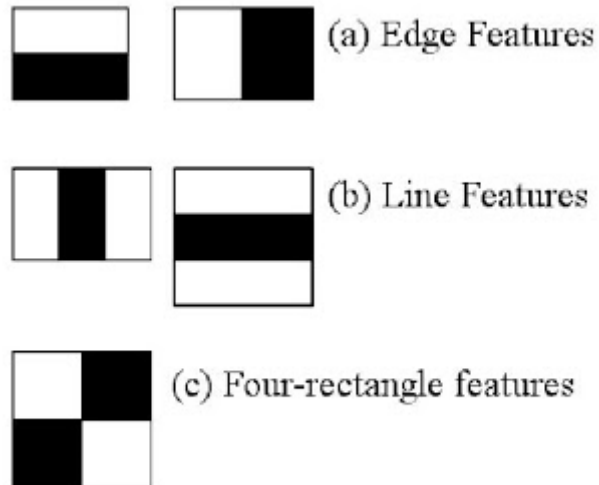
Sirve de proveedor de infraestructura para las aplicaciones relacionadas con la visión artificial.

- ✓ Es multiplataforma, capaz de ejecutarse en los principales sistemas operativos de tus dispositivos (Windows, Mac OS X, Linux, Android o iOS).
- ✓ Aunque este proyecto fue escrito en C y C++, se pueden utilizar en otros lenguajes de programación como Java, C# y uno de los más importantes, Python.
- ✓ Sólo ha contado con 3 versiones, lo que habla de la eficiencia de esta biblioteca desde su lanzamiento hace más de 21 años.
- ✓ Su desarrollo es permanente gracias al voluntariado, aunque muchas compañías importantes apoyan su desarrollo, como es el caso de Google.
- ✓ Su implementación es completamente gratuita, por ende, cualquier persona lo puede utilizar.

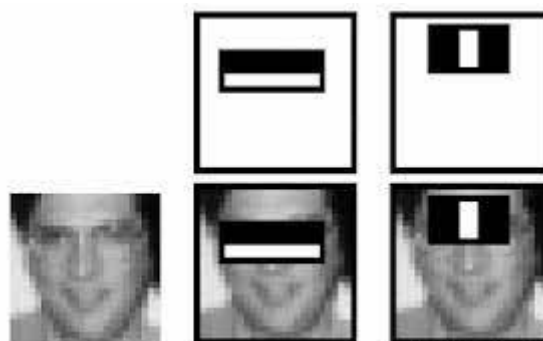
4.9.2 ¿Que es Función de Haars Cascade?

Es un detector de objetos utilizando los clasificadores en cascada basados en características de Haar es un método eficaz de detección de objetos propuesto por Paul Viola y Michael Jones en su documento, Detección rápida de objetos usando una cascada potenciada de características simples en 2001. Es un enfoque de aprendizaje automático donde la función de cascada está entrenada a partir de muchas imágenes positivas y negativas. Luego se usa para detectar objetos en otras imágenes. Aquí trabajaremos con detección de rostros. Inicialmente, el algoritmo necesita muchas imágenes positivas (imágenes de caras) e imágenes negativas (imágenes sin caras) para entrenar al clasificador. Entonces necesitamos extraer características de él. Para esto, se utilizan las características de haar que se muestran en la imagen a continuación. Cada

característica es un valor único que se obtiene al restar la suma de píxeles en el rectángulo blanco de la suma de píxeles en el rectángulo negro
Ahora todos los tamaños y ubicaciones posibles de cada kernel se usan para calcular muchas características. (Imagínese cuánto cálculo necesita, incluso una ventana de 24x24 da como resultado más de 160000 características).



Para cada cálculo de características, necesitamos encontrar la suma de píxeles en rectángulos blanco y negro. Para resolver esto, introdujeron las imágenes integrales. Simplifica el cálculo de la suma de píxeles, qué tan grande puede ser el número de píxeles, a una operación que involucra solo cuatro píxeles. Hace que las cosas sean súper rápidas. Pero entre todas estas características que calculamos, la mayoría de ellas son irrelevantes. Por ejemplo, considere la imagen a continuación. La fila superior muestra dos buenas características. La primera característica seleccionada parece enfocarse en la propiedad de que la región de los ojos a menudo es más oscura que la región de la nariz y las mejillas. La segunda característica seleccionada se basa en la propiedad de que los ojos son más oscuros que el puente de la nariz. Pero las mismas ventanas que se aplican en las mejillas o en cualquier otro lugar son irrelevantes. Entonces, ¿cómo seleccionamos las mejores características de 160000+ características?



Para esto, aplicamos todas y cada una de las características en todas las imágenes. Para cada característica, encuentra el mejor umbral que clasificará



las caras en positivas y negativas. Pero, obviamente, habrá errores o clasificaciones erróneas. Seleccionamos las características con una tasa de error mínima, lo que significa que son las características que mejor clasifican la cara y las imágenes que no son caras. (El proceso no es tan simple como esto. Cada imagen recibe el mismo peso al principio. Después de cada clasificación, se aumentan los pesos de las imágenes mal clasificadas. Luego, se repite el mismo proceso. Se calculan nuevas tasas de error. el proceso continúa hasta que se alcanza la precisión requerida o la tasa de error o se encuentra el número requerido de características). En una imagen, la mayor parte de la región de la imagen es una región sin cara. Entonces, es una mejor idea tener un método simple para verificar si una ventana no es una región de la cara. Si no es así, deséchelo en una sola toma. No lo proceses de nuevo. En lugar de enfocarse en la región donde puede haber una cara. De esta forma, podemos encontrar más tiempo para verificar una posible región de la cara. Para esto, introdujeron el concepto de Cascade Classifiers. En lugar de aplicar todas las características de 6000 en una ventana, agrupe las funciones en diferentes etapas de clasificadores y aplíquelas una a una. (Normalmente, las primeras etapas contendrán una cantidad de funciones muy inferior). Si una ventana falla la primera etapa, deséchela. No consideramos las características restantes en él. Si pasa, aplique la segunda etapa de características y continúe el proceso. La ventana que pasa todas las etapas es una región de la cara



Capítulo 5: Códigos y sus Características

5.1 Código del Sensor de Ultrasonido:

Detectar la distancia entre el sensor y el objeto más cercano que haga rebotar el pulso de onda que emite, de esta forma nos dirá a la distancia (en centímetros) del recorrido que hizo la onda mediante una cuenta específica que se la establece en la programación. Cabe aclarar que estas líneas de código se repiten ya que nosotros usamos 2 sensores de ultrasonido (lo único que cambian son los pines que se seleccionan para conectar el TRIG y el ECHO a la Raspberry).

```
import RPi.GPIO as GPIO
import time

TRIG = 2
ECHO = 3

# Indicar que se usa el esquema de numeración de pines
# de BCM (Broadcom SOC channel), es decir los números de
# pines GPIO (General-Purpose Input/Output).
GPIO.setmode(GPIO.BCM)

GPIO.setup(TRIG, GPIO.OUT)

# Establecer que ECHO es un canal de entrada.
GPIO.setup(ECHO, GPIO.IN)

print ("Medición de distancias en progreso")

try:
    while True:

        # Apagar el pin activador y permitir un par de
        # segundos para que se estabilice.
        GPIO.output(TRIG, GPIO.LOW)
        print ("Esperando a que el sensor se estabilice")
        time.sleep(2)

        GPIO.output(TRIG, GPIO.HIGH)
        time.sleep(0.00001)
        GPIO.output(TRIG, GPIO.LOW)

        print ("Iniciando eco")
        while True:
            pulso_inicio = time.time()
            if GPIO.input(ECHO) == GPIO.HIGH:
                break
        while True:
            pulso_fin = time.time()
            if GPIO.input(ECHO) == GPIO.LOW:
                break

        # La medición del tiempo es en segundos.
        duracion = pulso_fin - pulso_inicio

        # Calcular la distancia usando la velocidad del
```

Para abrirlo en una ventana nueva, darle clic dos veces sobre la imagen.

Links para el repositorio de GitHub:

- [Código Sensor HC-SR04](#)



5.2 Código MQ3:

La función principal es detectar las partículas evaporizadas de alcohol en el aire de la persona que sopla por la boquilla, para verificar si esta alcoholizada o no.

MQ3 SENSOR CODE

"""

If you have not installed the Wiring Pi Library,
install it using below command :

```
sudo pip install wiringpi
```

"""

#Copy and Paste the below code and save it as a ".py" file

```
import wiringpi as wiringpi
from time import sleep

wiringpi.wiringPiSetupGpio()
wiringpi.pinMode(10, 0)
count=0
while(count<20):
    my_input=wiringpi.digitalRead(10)
    if(my_input):
        print("Not Detected !")
    else:
        print("Alcohol Detected")
    count=count+1
    sleep(1)
```

Para abrirlo en una ventana nueva, darle clic dos veces sobre la imagen.

Links para el repositorio de GitHub:

- [Código MQ3](#)



5.3 Código Webcam Ojos:

La función principal es reconocer los ojos de la persona y hacer un marco del color que nosotros queramos sobre lo que está reconociendo.

```
import cv2

haar_file = (cv2.data.harcascades + 'haarcascade_eye_tree_eyeglasses.xml')

face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

while True:
    (_, im) = webcam.read()
    grey = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(grey)
    for (x, y, w, h) in faces:
        cv2.rectangle(im, (x,y), (x+w, y+h), (0, 0, 255), 1)

    cv2.imshow('OpenCV', im)

    key = cv2.waitKey(10)
    if key == 27:
        break
```

Para abrirlo en una ventana nueva, darle clic dos veces sobre la imagen.

Links para el repositorio de GitHub:

- [Código Webcam Ojos](#)
- [Haarcascade de Ojos](#)



5.4 Código Webcam Rostro:

La función principal es reconocer el rostro de la persona y como el anterior código, también dibuja un marco sobre lo que está reconociendo.

```
import cv2

haar_file = (cv2.data.harcascades + 'haarcascade_frontalface_default.xml')

face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)

while True:
    (_, im) = webcam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)
    for (x, y, w, h) in faces:
        cv2.rectangle(im, (x,y), (x+w, y+h), (0, 255, 0), 1)

    cv2.imshow('OpenCV', im)

    key = cv2.waitKey(10)
    if key == 27:
        break
```

Para abrirlo en una ventana nueva, darle clic dos veces sobre la imagen.

Links para el repositorio de GitHub:

- [Código Webcam Rostro](#)
- [Haarcascade de Rostro](#)



5.5 Código Cámara más las Alarmas:

Lo que hicimos en esta prueba de código de la webcam fue mezclar los datos que recibíamos de cada cámara para ponerles un contador, entonces si por ejemplo, el programa no detectaba ningún parámetro buscado, al repetirse esa función por 4 veces mostraría la alarma.

```
import cv2
import time

haar_file = (cv2.data.haarcascades + 'haarcascade_eye_tree_eyeglasses.xml')

face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(1)
contador = 0

while True:
    (_, im) = webcam.read()
    grey = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(grey)
    print(len(faces))
    if (len(faces) == 0):
        time.sleep(3)
        contador += 1
    if (contador == 4):
        print("Esto es una alerta")
        print(contador)
        contador = 0
    for (x, y, w, h) in faces:
        cv2.rectangle(im, (x,y), (x+w, y+h), (0, 0, 255), 1)

    cv2.imshow('OpenCV', im)

    key = cv2.waitKey(10)
    if key == 27:
        break

haar_file = (cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(1)
while True:
    (_, im) = webcam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)
    print(len(faces))

    if (len(faces) == 0):
        time.sleep(3)
        contador += 1
    if (contador == 4):
        print("Esto es una alerta")
```

Para abrirlo en una ventana nueva, darle clic dos veces sobre la imagen.

Links para el repositorio de GitHub:

- [Código Webcam Completo](#)
- [Haarcascade de Rostro](#)
- [Haarcascade de Ojos](#)



5.6 Código Completo (Sensores Ultrasónicos, Sensor MQ3, Cámara y Alarmas):

La función principal es activar primero el MQ3 para tomar el alcohol en sangre, si da positivo se detiene el programa y suena una alarma y si da negativo se activan los sensores ultrasónicos en un bucle constante, si este detecta parámetros o patrones incorrectos suena una alarma y detiene el programa, sino sigue en un bucle. Cabe aclarar que todavía estamos trabajando sobre este código por un error en la medición que nos dan los sensores HC-SR04.

```
import wiringpi as wiringpi
from time import sleep
import RPi.GPIO as GPIO
import time
import cv2

GPIO.setmode(GPIO.BCM)
GPIO.setup(11, GPIO.IN)
GPIO.setup(5, GPIO.OUT)

TRIG = 2
ECHO = 3
TRIG2 = 4
ECHO2 = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.setup(TRIG2, GPIO.OUT)
GPIO.setup(ECHO2, GPIO.IN)

def alcohol():
    try:
        while True:
            if GPIO.input(11):
                time.sleep(0.1)
                dato = 0
            if GPIO.input(11)!=1:
                dato = 1
                GPIO.output(5, True)
                time.sleep(0.1)
                GPIO.output(5, False)
            return dato
    finally:
        GPIO.cleanup()

def sensor1():
    #print ("Medicion de distancias progreso del primer sensor")
    try:
        while True:
            GPIO.output(TRIG, GPIO.LOW)
            #print ("Esperando a que el primer sensor se estabilice")
            time.sleep(1)
            GPIO.output(TRIG, GPIO.HIGH)
            time.sleep(0.00001)
```

Para abrirlo en una ventana nueva, darle clic dos veces sobre la imagen.

Links para el repositorio de GitHub:

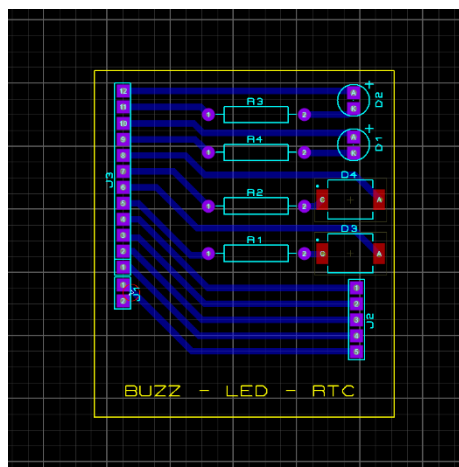
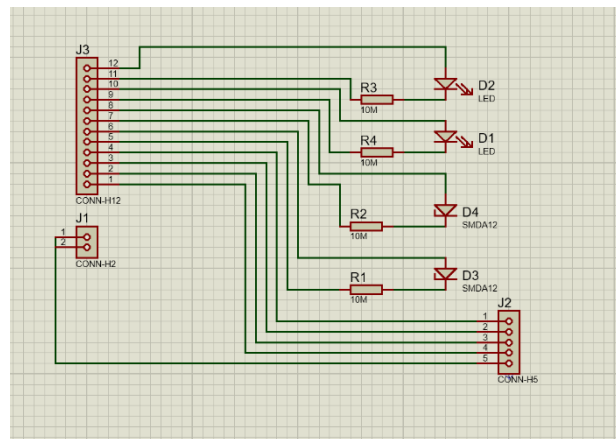
- [Código Completo](#)

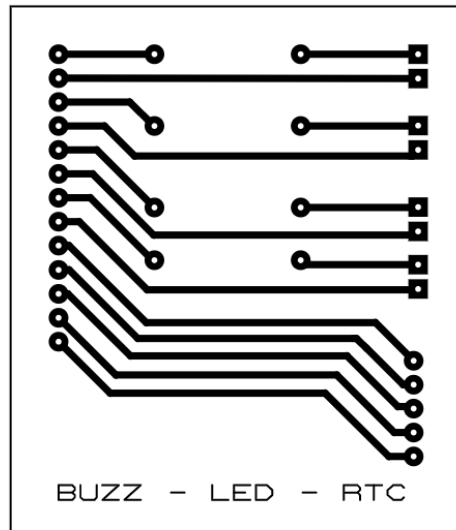


Capítulo 6: PCB

6.1 PCB de Buzzer, LED y RTC:

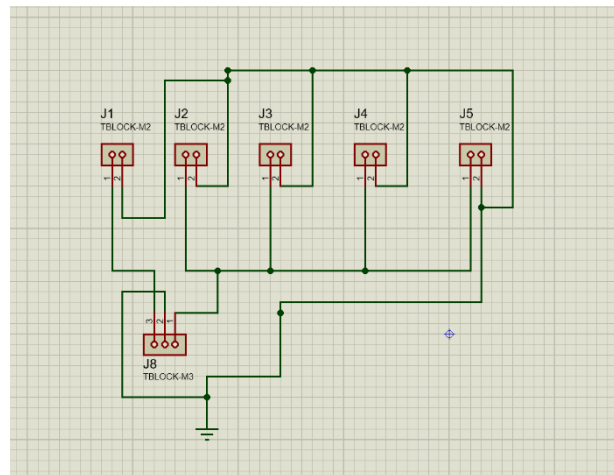
El siguiente PCB va a ser el que reciba todos los cables de salida de la Raspberry Pi para que podamos manipular los buzzer, los LEDs y un módulo de tiempo real. Entonces podríamos decir que esta placa es la encargada de reproducir el sistema de alarma que incorporamos a AFRS.





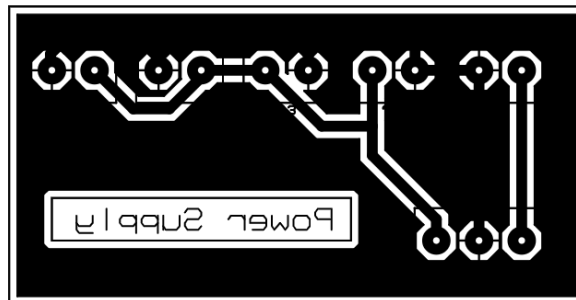
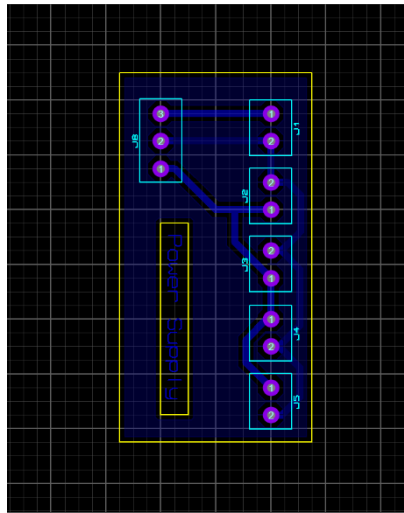
6.2 PCB de alimentación:

Nosotros diseñamos este esquemático con el fin de que podamos tener una salida en paralelo de la Raspberry de los pines de GND, 3.3v y 5v. Entonces, los únicos cables que salen de la computadora de abordo son solo 3 y después nosotros los separamos por las pistas. La ventaja que nos da las borneras es que podemos reconocer mucho más fácil los cables de alimentación además de que quede más prolijo.





Escuela de Educación Técnica Nº7
Taller Regional Quilmes
Prácticas Profesionalizantes: Especialidad Aviónica





Links de acceso:

- GitHub:
<https://github.com/MatiasRG94/AFRS>
- Drive:
<https://drive.google.com/drive/folders/1pjuKjb85txcpShlABBJAqa75F9WfbclW?usp=sharing>
- Instagram:
https://www.instagram.com/proyecto_afrs/
- YouTube:
<https://www.youtube.com/channel/UCBnhB07afu-79kdqT3G7ePQ>