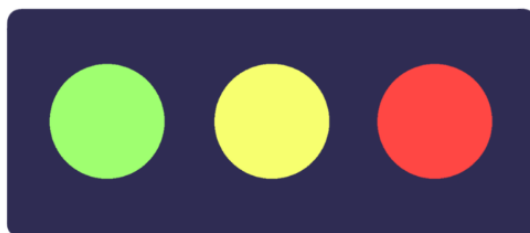


2021



TCC

Carpeta Técnica

Ambientes Controlados, Trabajo Seguro

E.E.S.T N°7 Quilmes "TRQ"

Integrantes del Grupo:

CASTIGLIONE, León Joaquín. DNI: 44595507

CORRAO, Marco Renzo. DNI: 42819481

GUZMÁN, Juan Manuel. DNI: 44339123

NUÑEZ PIEDRAS, Sebastian. DNI: 44395556

ORELLANA, Maximiliano Octavio. DNI: 44455623

(Todos los integrantes son de la división 7°1° Aviónica)

Docentes Responsables:

MEDINA, Sergio.

BIANCO, Carlos Cesar.

ARGUELLO, Gustavo.

CABALLERO, Facundo Rodrigo.

MINUCCI, Mauro.

Fecha de Inicio, Duración y Horas Dedicadas:

El proyecto empezó el 17 de Marzo de 2021, su realización duró 28

Semanas y los integrantes le dedicaron un promedio de 30 Horas por cada semana

Personas Afectadas:

BIANCO, Carlos Cesar:

Guia para el proyecto, Ayuda en la construcción de la estructura.

CARLASSARA, Fabricio:

Encargado de la inscripción, Ayuda en el software

MEDINA, Sergio:

Guía para organizar el avance del proyecto, Utilización del Sistema Scrum

CABALLERO, Facundo Rodrigo:

Ayuda en el diseño de circuitos

MORCILLO, Jacqueline Elizabeth:

Búsqueda de precios, Encargada de la compra de materiales



Equipo TCC



TCC



1. Introducción:	5
1.1. Contexto	5
1.2. Objetivo del Proyecto	5
1.3. Alcance	5
1.4. Beneficios	5
2. Características Técnicas:	6
2.1. Descripción General	6
2.2. Diagrama en Bloques del Funcionamiento	7
2.3. Electrónica	7
Alimentación	7
Sensores	9
Microcontroladores	11
Circuitos de Control	14
2.4. Programación	15
Código STM32	16
Código PIC16F88	17
3. Anexo:	19
3.1. Software Utilizado	19
3.2. Bibliografía	19



1. Introducción:

1.1. Contexto

En las fábricas, centrales, talleres y demás zonas de trabajo industrial, los trabajadores son expuestos a distintos tipos de contaminación por cortos o largos periodos de tiempo, que sin las medidas de prevención necesarias, puede llegar a afectar negativamente en su salud.

Aunque los trabajadores están obligados a vestir la indumentaria de protección, muchos de ellos omiten esta obligación debido a que muchos tipos de contaminación no se pueden percibir directamente y parecería que no están. Este juicio muchas veces es incorrecto y resulta en accidentes y enfermedades profesionales a corto o largo plazo.

1.2. Objetivo del Proyecto

El objetivo del TCC es mostrarle a los trabajadores el nivel de contaminación que se encuentra en su ambiente laboral para que usen la indumentaria necesaria (i.e: mascarilla, protección auditiva) y se protejan de la contaminación.

1.3. Alcance

El TCC fue pensado como un proyecto de seguridad y salud laboral en talleres aeronáuticos, pero se puede adaptar fácilmente a todos los ambientes industriales donde los trabajadores se exponen a gases producidos por combustión, ejemplos de esto sería:

- Talleres mecánicos de cualquier vehículo motorizado.
- Centrales de energía termoeléctrica.
- Ambientes laborales con maquinaria motorizada y exceso de ruido.

1.4. Beneficios

El TCC analiza el nivel de contaminación e indica claramente el peligro que este le supone al trabajador con el objetivo de que el mismo use la indumentaria de seguridad necesaria. Este sistema funciona como una forma de prevención de enfermedades profesionales relacionadas con el sistema respiratorio y la pérdida parcial y/o total de la audición.



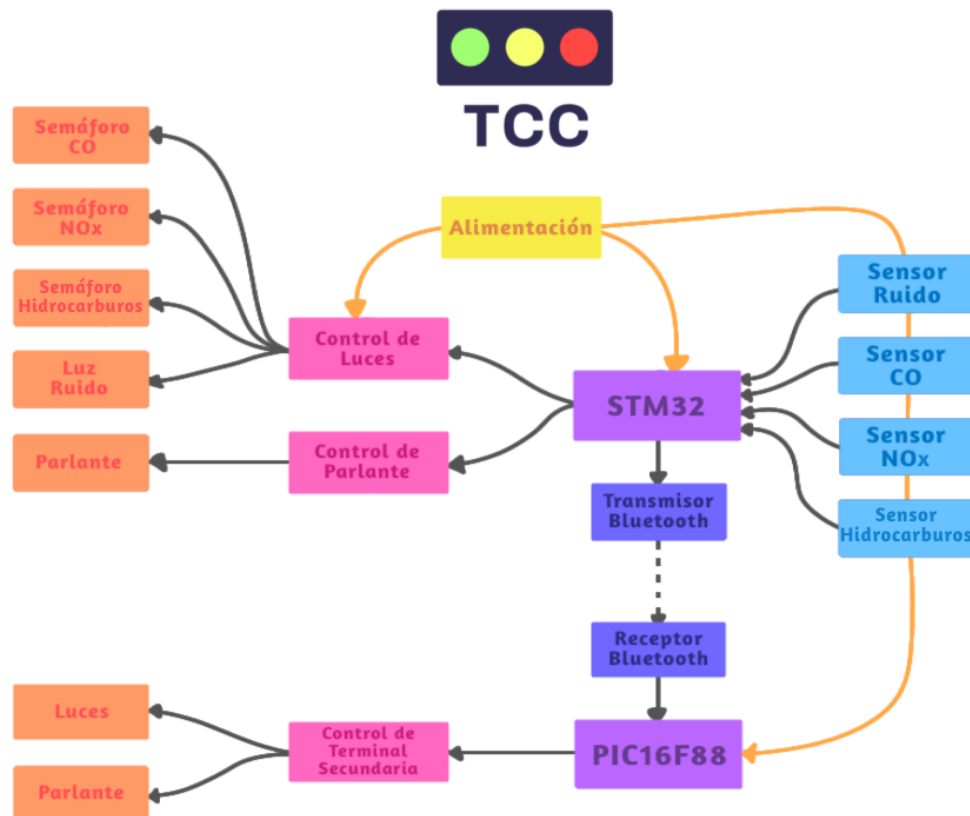
2. Características Técnicas:

2.1. Descripción General

El TCC consiste de tres etapas:

1. Etapa de sensado: Mediante componentes electroquímicos, el TCC detecta gases contaminantes en tiempo real mientras que, mediante un micrófono condensador omnidireccional detecta el nivel de presión sonora del ambiente.
2. Etapa de procesamiento de señales: Mediante microprocesadores y un sistema operativo en tiempo real, procesa las señales que recibe de la etapa de sensado para interpretarlas como partes por millón en el caso de los gases y decibelios en el caso del ruido, y las compara con niveles de contaminación de gases máximos recomendados por la ONU y los niveles de ruido máximos para los hangares aeronáuticos establecidos por el ANAC.
3. Etapa de muestreo y alarma: Mediante dos terminales distintas (una principal de mayor tamaño y detalle, y una secundaria de menor tamaño y detalle que se comunica de manera inalámbrica con la principal) muestra los niveles de contaminación presentes. Además de esto, si la presencia de gases excede los niveles máximos establecidos, sonará una alarma para llamar la atención de los trabajadores.

2.2. Diagrama en Bloques del Funcionamiento



2.3. Electrónica

El proyecto está compuesto electrónicamente por 3 cápsulas electroquímicas, dos microcontroladores, un transmisor y un receptor bluetooth, luces y parlantes, y placas de auditoría propia que cumplen distintas funciones.

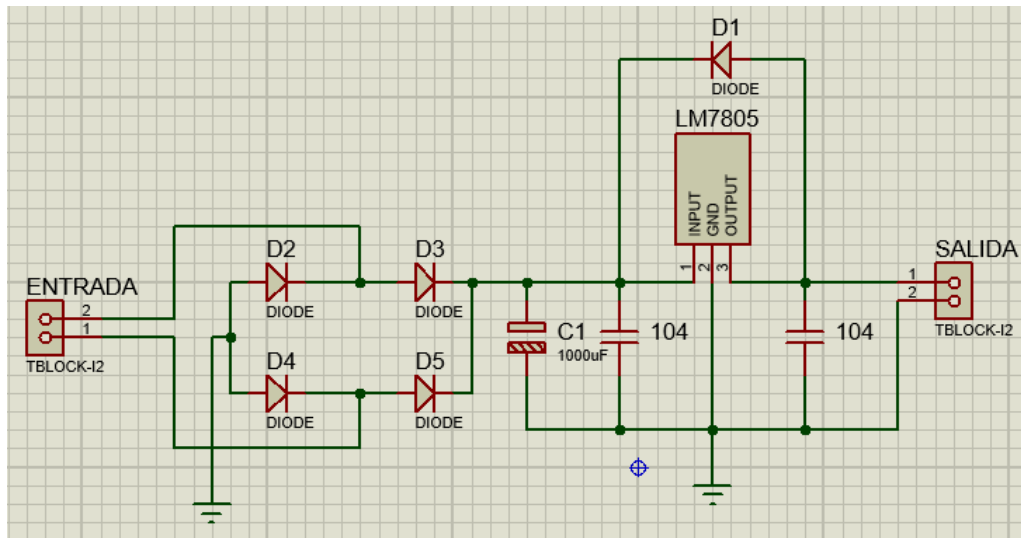
Alimentación

El circuito de la terminal secundaria y los sensores están alimentados únicamente con 5V de continua, exceptuando el sensor de sonido que se encuentra alimentado a la vez con -5V también de continua.

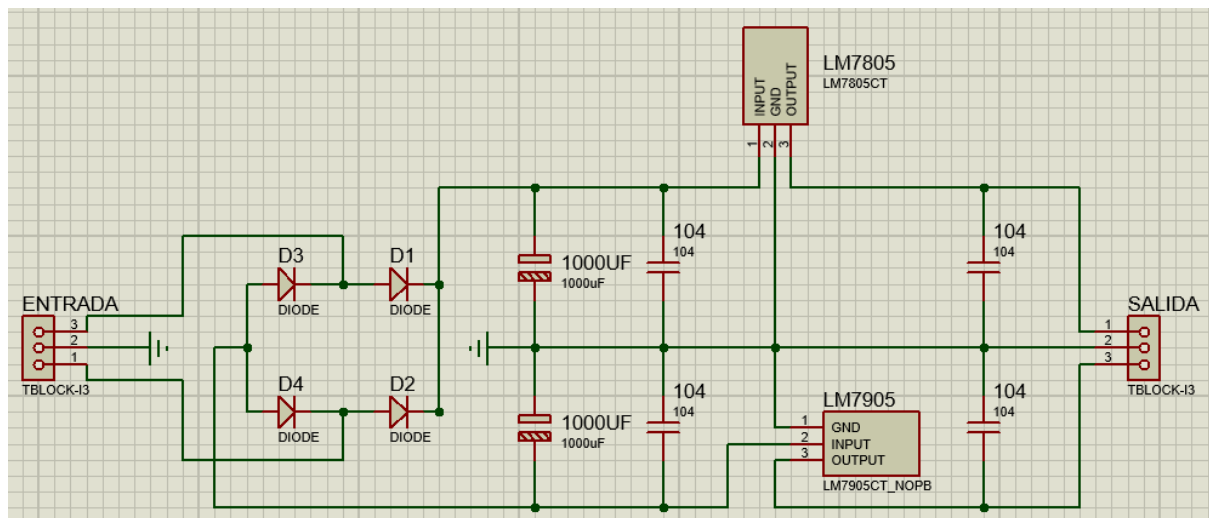
El microcontrolador STM32 se alimenta con 3.3V.

Los circuitos que controlan las luces de los semáforos están alimentados con 220V de alterna ya que es a ese voltaje en el que operan las lámparas que componen los semáforos.

El circuito que controla el parlante de la terminal principal se alimenta mediante el STM32.



Diseño Fuente 5V



Diseño de Fuente Simetrica 5V/-5V

Sensores

Sensores de Gases:

Los sensores utilizados en el TCC son las cápsulas electroquímicas MQ-9 para el monóxido de carbono, el MQ-135 para los óxidos de nitrógeno y el MQ-6 para los hidrocarburos , además de esto se diseñó y fabricó un sensor de sonido utilizando: Un circuito compuesto de un micrófono electret, un preamplificador para el mismo y amplificadores operacionales con distintas configuraciones.

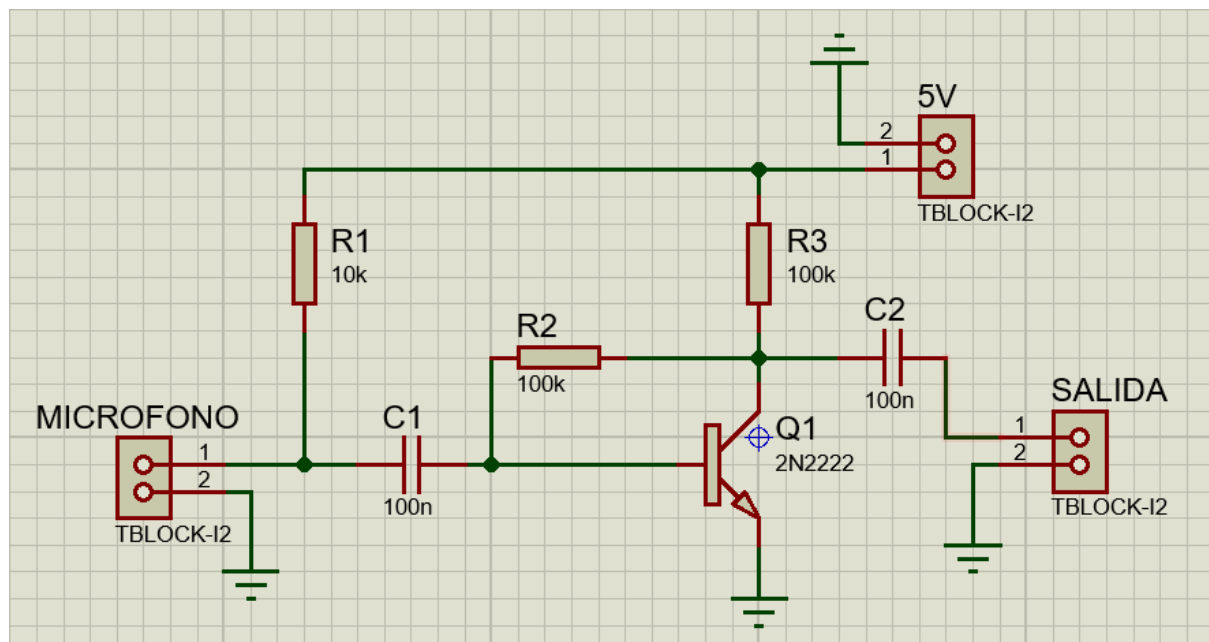


Cápsulas electroquímicas MQ-6, MQ-9 y MQ-135

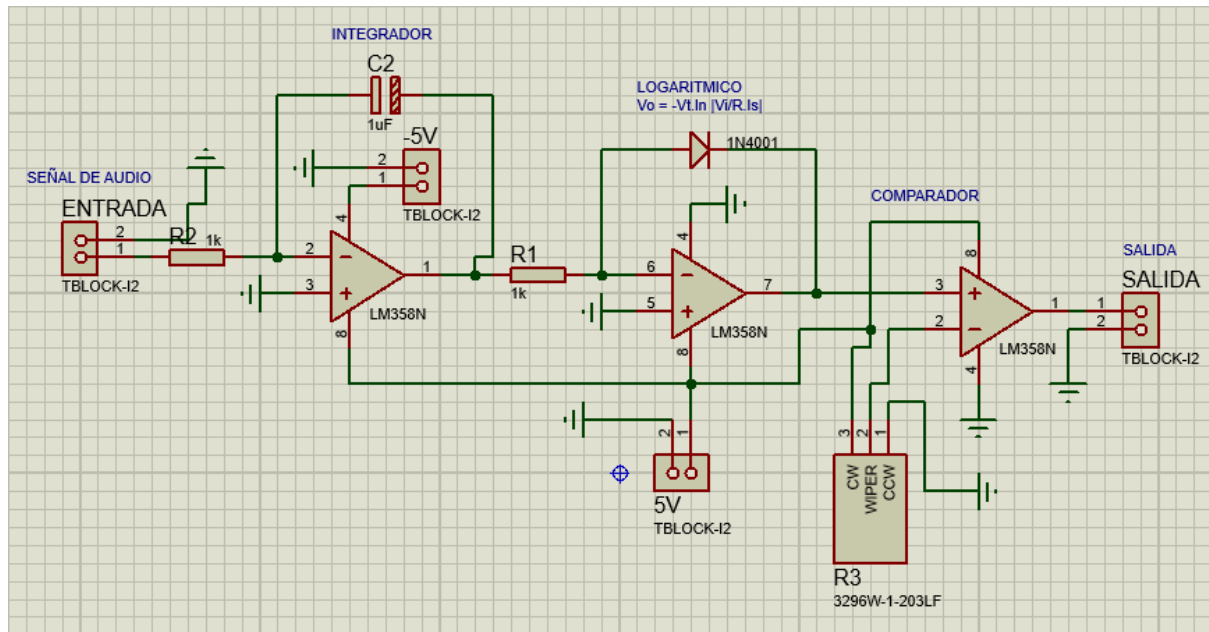
Sensor de sonido:

El sensor de sonido que diseñamos y fabricamos se basa en dos circuitos:

- Circuito pre-amplificador: Utilizamos un pre-amplificador para amplificar la señal del micrófono y así poder procesarla y utilizarla, debido a que por sí sola la señal es muy pequeña.
- Procesamiento de señal: Se procesa la señal mediante 3 configuraciones diferentes de amplificadores operacionales que cumplen funciones distintas.
 - El integrador otorga un valor en voltaje promedio a la salida que solo varía si la variación se sostiene en el tiempo.
 - El logarítmico ofrece a la salida un valor de voltaje igual al logaritmo natural de la entrada, lo cual hace que el incremento y decremento de la salida se asemeje a la forma de percibir el sonido del oído humano.
 - El comparador se utiliza para comparar dos señales, en este caso si la señal de entrada supera un valor establecido la salida será 5V.



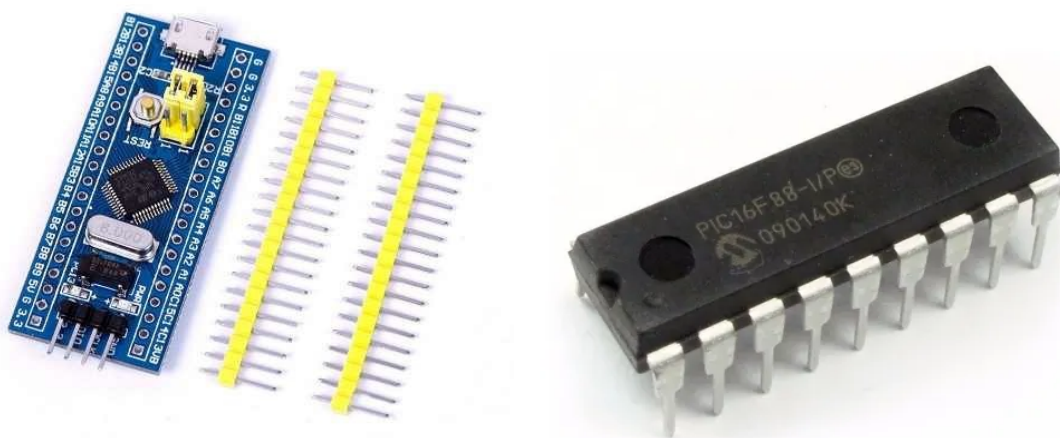
Circuito Pre-Amplificador



Procesamiento de la señal

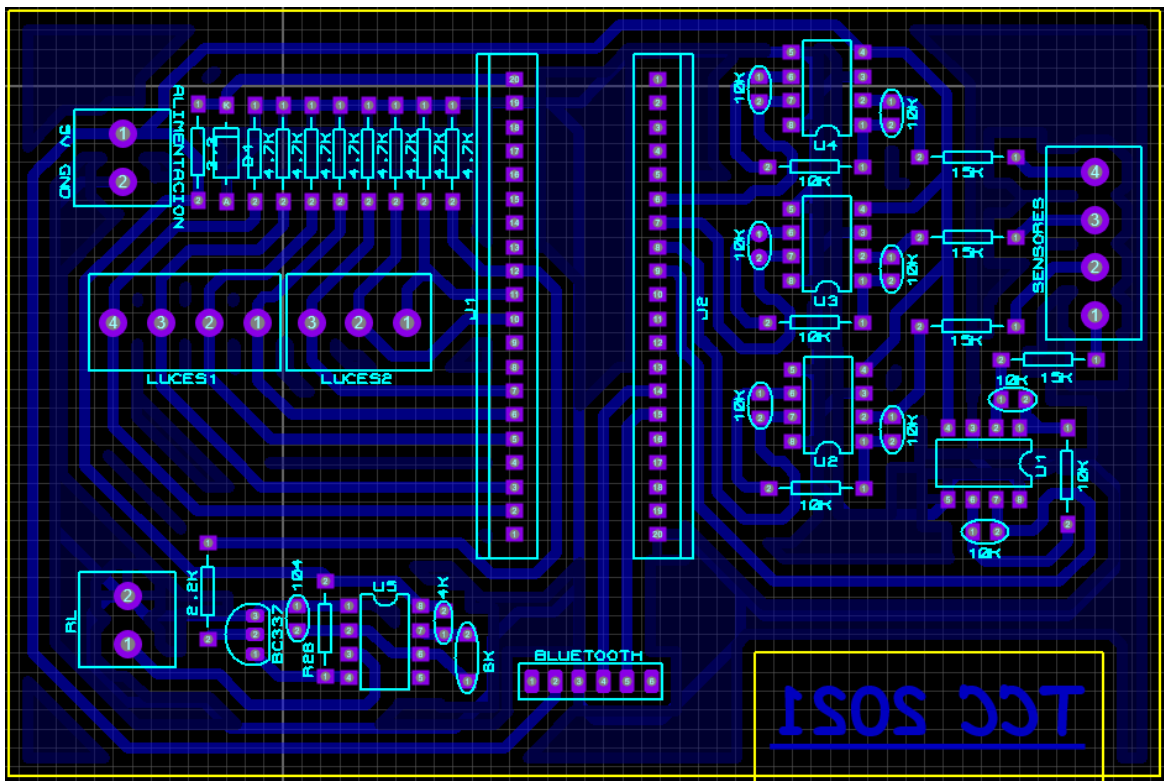
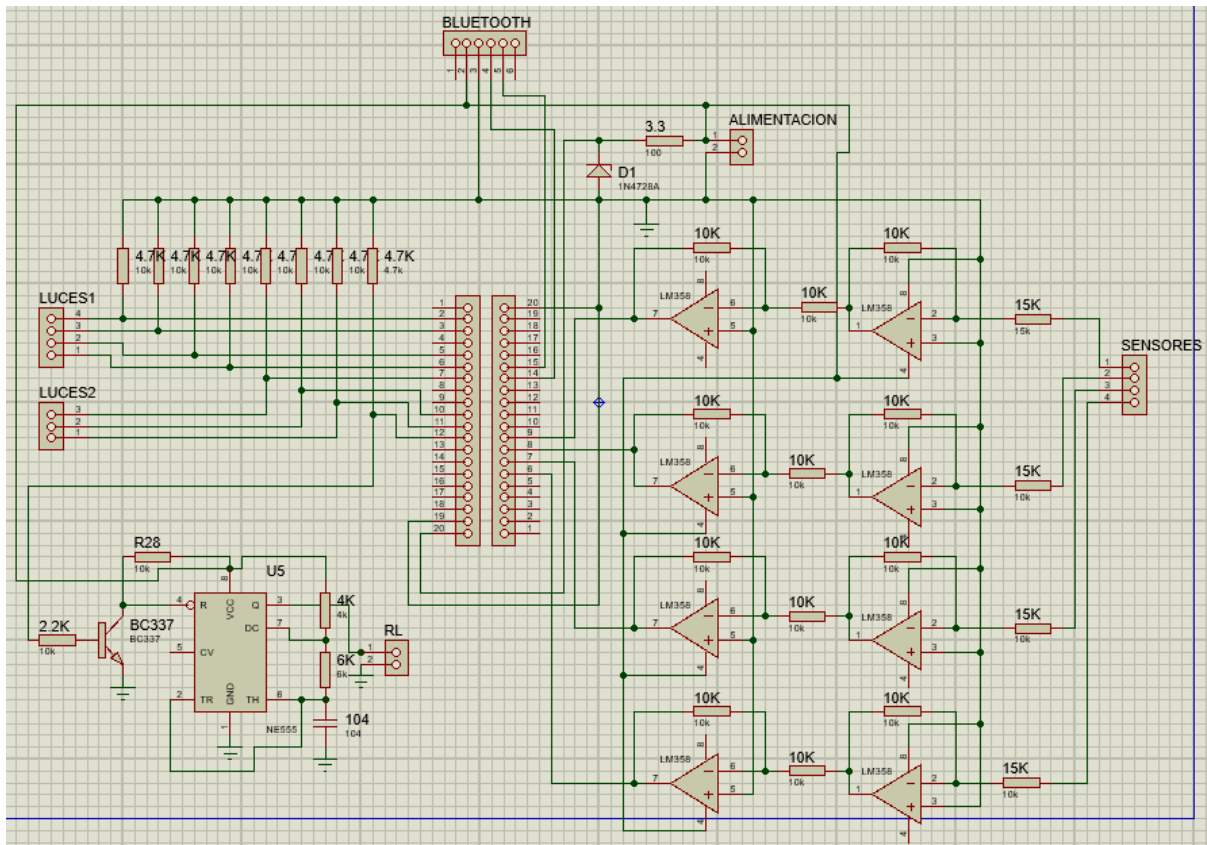
Microcontroladores

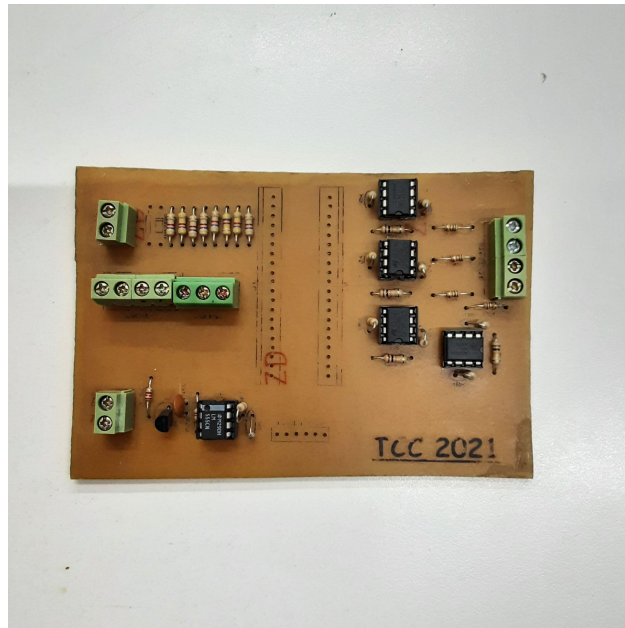
El TCC utiliza dos microcontroladores: El SMT 32, el cual está basado en la familia de procesadores Arm Cortex M3, para la recepción de datos de los sensores, salidas en la terminal principal y envío de señales a la terminal secundaria utilizando un Sistema Operativo de Tiempo Real; Y el PIC16F88 para la recepción de datos enviados por el STM mediante un receptor bluetooth y las salidas en la terminal secundaria.



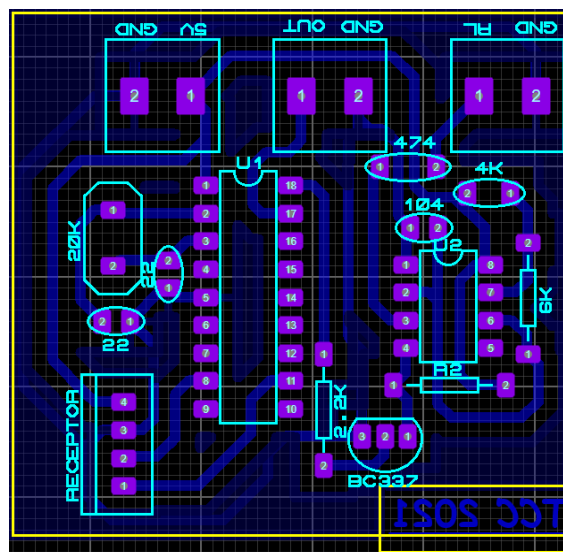
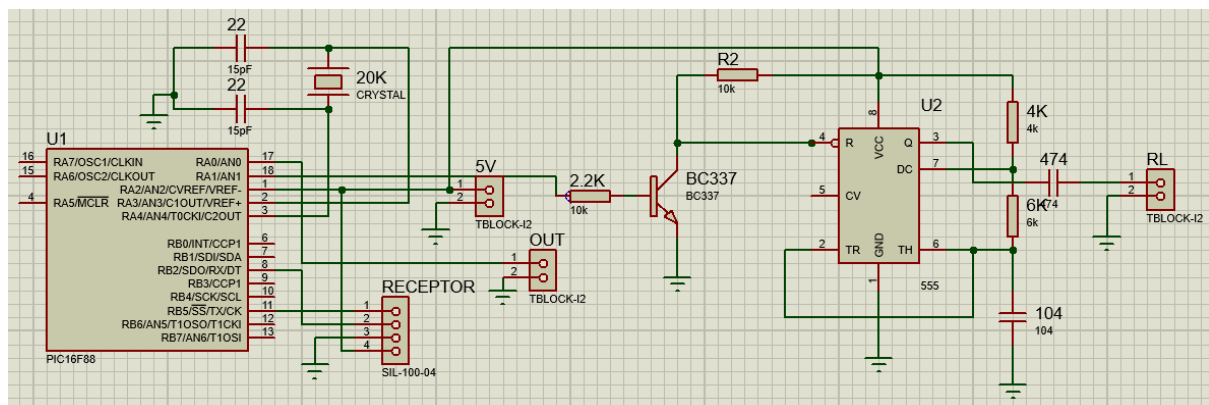
Módulo de Desarrollo STM32 "Blue Pill" y PIC16F88 de Microchip

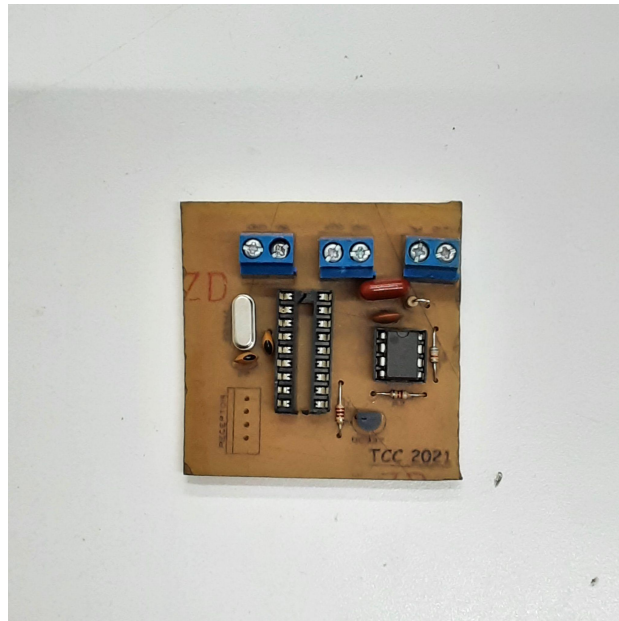
Para ambos microcontroladores se diseñaron y fabricaron circuitos.





Circuito del STM32F103C8T6

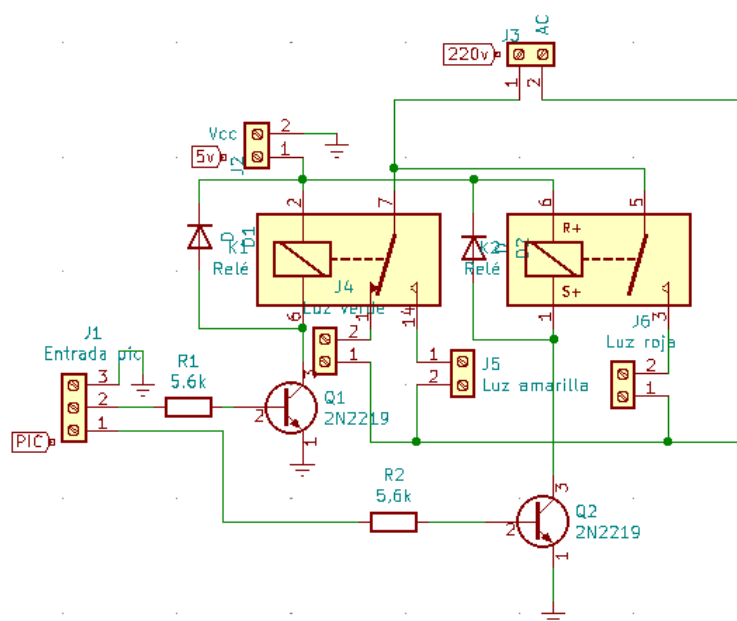


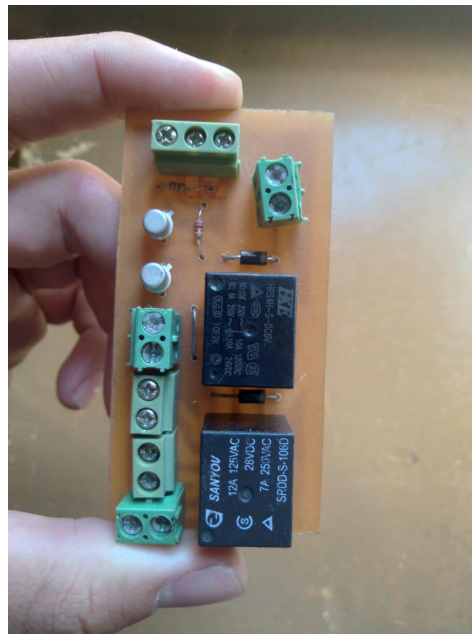
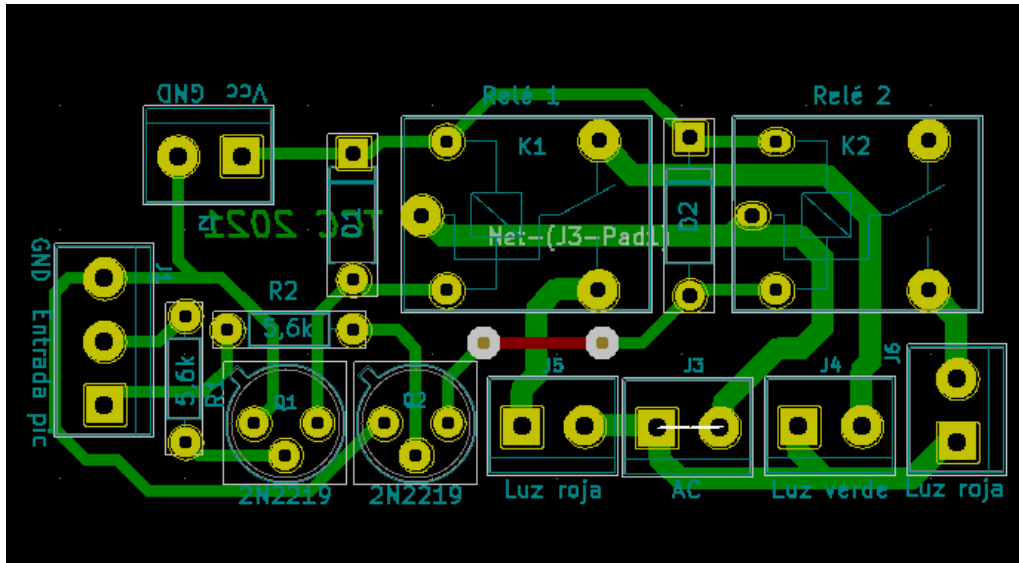


Circuito del PIC16F88

Circuitos de Control

Tanto para la terminal principal como para la secundaria, se diseñaron y fabricaron circuitos para activar las diferentes salidas. Se hizo un circuito por cada grupo de luces y un circuito para el parlante en el caso de la terminal principal, y un circuito que controla las salidas de la terminal secundaria.





2.4. Programación

15



grabado en el microprocesador mediante el dispositivo USB ST-link con STM CUBE programmer.

La programación consta de una estructura de datos donde se acoplan los datos obtenidos de los sensores, puertos, pines y valores de lectura, valores a los cuales los consideramos Límites (usados para realizar comparaciones), por último, la estructura de datos tiene el que canal por el cual se recibe el dato. La estructura es nombrada Límites.

Código STM32

Comandos por defecto:

System_clock_config

Mx_GPED_init Puertos I/O Digitales

A/D Mx_ADC_init Puerto conversor

SPI Mx_usart1_UAR_init Puerto comunicación

Get ADC value (devuelve el dato “valor”) Realiza la conversión

Estructuras:

Estructura “Límite”: Organiza y limita los datos recibidos en enteros sin signos. Los datos que organiza son:

- Luz verde
- Luz amarilla
- “Valor”, El valor que devuelve “get_ADC_value”
- Posición de pines
- N, Información sobre los nodos
- ADC channel type depth, Información sobre los canales

Funciones:

main(): Inicializa el programa. Llama la estructura “Límite” y empieza a recibir los datos. Crea los valores CO, NO y SO. Crea el dato “máximo ruido”.

while(): Hace que la estructura pase por los datos CO, NO, SO y máximo ruido. Compara los datos recibidos con los establecidos y crea el dato “dato” y lo envía por el transmisor bluetooth y las demás salidas.



```
int main(void)
{
    /* USER CODE BEGIN 1 */
    struct Limite CO,NO,SO;

    uint32_t MaximoRuido = 1000;

    CO.verde=550;
    CO.amarillo=2000;

    NO.verde=550;
    NO.amarillo=2000;

    SO.verde=550;
    SO.amarillo=2000;

    CO.sig = &NO;
    NO.sig = &SO;
    SO.sig = &CO;

    CO.n = 0;
    NO.n = 1;
    SO.n = 2;

    CO.pos[0] = CO1_Pin;
    CO.pos[1] = CO2_Pin;
    NO.pos[0] = NO1_Pin;
    NO.pos[1] = NO2_Pin;
    SO.pos[0] = SO1_Pin;
    SO.pos[1] = SO2_Pin;

    CO.ch.Channel = ADC_CHANNEL_1;
    CO.ch.Rank = ADC_REGULAR_RANK_1;
    CO.ch.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;

    NO.ch.Channel = ADC_CHANNEL_2;
    NO.ch.Rank = ADC_REGULAR_RANK_1;
    NO.ch.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;

    SO.ch.Channel = ADC_CHANNEL_3;
    SO.ch.Rank = ADC_REGULAR_RANK_1;
    SO.ch.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;

    ADC_ChannelConfTypeDef sdr_canal;
    sdr_canal.Channel = ADC_CHANNEL_4;
    sdr_canal.Rank = ADC_REGULAR_RANK_1;
    sdr_canal.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;

    uint8_t dato;

    /* USER CODE END 1 */
}
```

main()

Código PIC16F88

Estructuras:

Estructura “Struct”: Genera un dato llamado “datos_s” a partir de una secuencia de números de 1 bit.

Estructura “Union”: Genera un dato llamado “datos_u” que es la secuencia de números de la estructura “Struct” representados con un símbolo.

Funciones:

void USART_init: Inicializa los puertos USART para la lectura del receptor bluetooth.

void USART_read: Analiza si está preparado para la recepción de datos, si no lo está sigue esperando hasta que lo esté. En caso de que esté



preparado, empieza a leer los datos que recibe del receptor bluetooth. Recibe una secuencia de datos de valor “1” o “0” y retorna la secuencia al main(). Si se genera un overflow, desactiva la recepción de datos, limpia las instrucciones, y vuelve a activar la recepción de datos.

main(): Configura los pines de I/O. Crea las variables para los datos que va a utilizar. Crea una flag que se activa si uno de los datos que recibió indica que hubo un exceso de contaminación. Configura un Timer.

while(): Confirma que llegó el paquete de datos. Compara los datos recibidos por el módulo bluetooth con unas variables que representan los límites de contaminación. En el caso de que haya un exceso de los límites establecidos, se activa la flag y comienza una rutina de parpadeo de las luces en la salida utilizando el Timer configurado en el main().

```
// Main programa
void main(void){
    // Pin config
    TRISA = 0x00; // todos los pins como salida
    PORTA = 0x00; // todos los pins seteados en 0

    // Variables
    Datos_u_datos_union;
    unsigned char detecion_f = 0;

    USART_init(38400);
    TMR0 = 0;

    //configuracion de timer
    OPTION_REGbits.T0CS = 0; // el pulso depende del oscilador
    OPTION_REGbits.PSA = 0; // activa el preescaler
    OPTION_REGbits.PS = 0b101; // prescaler 1:64
    /*
    * La frecuencia del oscilador es de 20Mhz.
    * El clock tiene una frecuencia de 1:4 de la frecuencia del oscilador.
    * Lo que daría una frecuencia de 5Mhz
    * y con el ultimo el preescaler de 1:64 da una frecuencia de clock de 78,125khz.
    * En conclusion 12,8 milisegundos por iteracion
    */

    unsigned int iteraciones = TIEMPO/12.8;

    while(1){
        datos_union.puerto = USART_read();
        if(datos_union.datos.OK){
            if(datos_union.datos.CO2 || datos_union.datos.NO2 || datos_union.datos.SO2 || datos_union.datos.SdR)
                detecion_f = 1; //contaminacion detectada
            else
                detecion_f = 0; //ambiente libre de contaminacion
        }
        if(detecion_f){
            PORTA |= (1<<LUZ);

            // parpadeo
            if(TMR0>iteraciones){ // tiempo
                if(PORTA & (1<<PARLANTE))
                    PORTA |= (1<<PARLANTE);
                else
                    PORTA &= ~(1<<PARLANTE);
                TMR0 = 0; // reset timer
            }
        }
    }
    return;
}
```

main ()



3. Anexo:

3.1. Software Utilizado

- Software utilizado para la organización de trabajo: Trello, GitHub
- Software utilizado para el desarrollo de placas: Proteus 8 Professional, KiCad.
- Software utilizado para el desarrollo de software: MPLAB X IDE, STM32Cube IDE, Visual Studio Code.

3.2. Bibliografía

- a. [Definición de la Política Ambiental de la Administración Nacional de Aviación Civil](#)
- b. [Guia práctica sobre el ruido en el ambiente laboral. Superintendencia de Riesgos del Trabajo.](#)
- c. [Tipos de gases producidos en la combustión y sus consecuencias](#)
- d. [El ruido ambiental y los efectos en el hombre](#)