

Diseño de sistema de monitoreo de temperatura y posición de personas en lugares públicos.

Instituto Tecnológico de Costa Rica
Diseño de Sistemas Digitales

Leonardo Guillén Fernandez -2019031688
José Julián Camacho Hernández - 2019201459
Juan Pablo Carrillo Salazar - 2019380111
Ethan Flores Montero - 2017152644

Resumen—The purpose of this project is to design an electronic monitoring system that keeps track of body temperature in a non-invasive and automated way, in order to detect possible symptoms of COVID-19. The idea is to place the system in a specific crowded area, such as a bank in this case, and in this way control and make sure that the people who are inside have sanitary measures, without exposing others. The document will guide you from modeling the solution, from use and borderline cases, to implementation with off-the-shelf hardware components, a budget for that project is also displayed.

Index Terms—Monitoreo, Temperatura, Sensor, COVID-19, ARM, Cortex M3.

I. INTRODUCCIÓN

Debido a la presente pandemia de COVID-19 se han tenido que tomar varias medidas sanitarias y varias medidas de salud para poder contener esta enfermedad. Hasta el día de hoy gracias a las vacunaciones y a diferentes medidas sanitarias tomadas, se ha controlado la pandemia, dando ciertas libertades a las personas como mascarilla no obligatoria en algunos lugares y que estas personas vuelvan a lugares concurridos normalmente. Sin embargo, no ha desaparecido el COVID-19 y hay que seguir previniendo contagios, para esta situación es necesario buscar la forma de dar seguridad a las personas cuando van a lugares con bastantes personas cercas y garantizar su bienestar lo mayor posible y con esto entra una implementación de un sistema de monitoreo de temperatura corporal no invasiva y automatizada en tiempo real, donde detectara patrones de temperatura en las personas dentro del local y pueda distinguir posibles casos de COVID-19. Con esto se busca ayudar al sector de la salud y así volver a una vida más fluida y cotidiana sin dejar de lado el cuidado de la pandemia.

II. SOLUCIÓN PLANTEADA

A continuación, se explicará una propuesta de prototipo del sistema de monitoreo para poder medir la temperatura de las personas en un escenario específico, con ciertas condiciones determinadas por el grupo.

II-A. Descripción del área de aplicación

Para este modelo se limitó un área, en la cual se procederá a utilizar el dispositivo. Dicha área de aplicación es un banco, ya que se consideró este como un lugar donde las personas acuden

comúnmente y en grandes cantidades. Este lugar fue elegido para poder tener control de las personas y estas no generen un contagio si ingresan al local con síntomas de COVID-19.

En la figura 1, se muestra una vista aérea de un banco, en la cual se presentan los espacios que integran al mismo. Este cuenta con un área total de 120 metros cuadrados, con dimensiones de 10 x 12 metros. Se compone de dos bloques principales de 10 x 6 metros, donde el primero, a la derecha, corresponde a la sala donde se encuentran los administrativos y los que atienden en las ventanillas, así como sus respectivos espacios de reuniones y el baño del banco.

En la segunda, a la izquierda, se encuentran espacios donde los clientes esperan su turno y son atendidos. Para el caso que se determina, el aforo máximo de esta parte del banco es de 24 personas sentadas.

Se establece este máximo para garantizar un distanciamiento social, donde teóricamente cada persona pueda tener un área personal mayor a 2 metros cuadrados de los 60 metros totales de esta sección.



Figura 1. Vista aérea del banco.

II-B. Posicionamiento de los componentes del sistema

En la figura 2 se presenta el área que se va a utilizar para la propuesta del dispositivo de monitoreo de temperatura en tiempo real. Como se muestra, el bloque administrativo no será tomado en consideración para el sistema, debido a que se asume que para entrar al mismo se debe pasar por el área

de clientes y el número de funcionarios es muy reducido, por lo que no se considera necesario dar seguimiento a sus respectivas temperaturas.

El dispositivo del sistema estará compuesto inicialmente por cuatro cámaras térmicas ubicadas en las esquinas de la sala para clientes. Estas se ubican de tal manera que se cubra el área total del recinto, para así lograr monitorear a todas las personas que se encuentren en el mismo.

Como se muestra en la figura, las cámaras térmicas deben tener un ángulo de visión horizontal de al menos 90° y un alcance de 5 metros como mínimo para conseguir el objetivo de cubrir la totalidad del espacio. Estas condiciones implican una superposición en las áreas alcanzadas por las cámaras, tal como se presenta en la imagen. Este factor será beneficioso para hacer una confirmación de las mediciones en dichas áreas superpuestas.

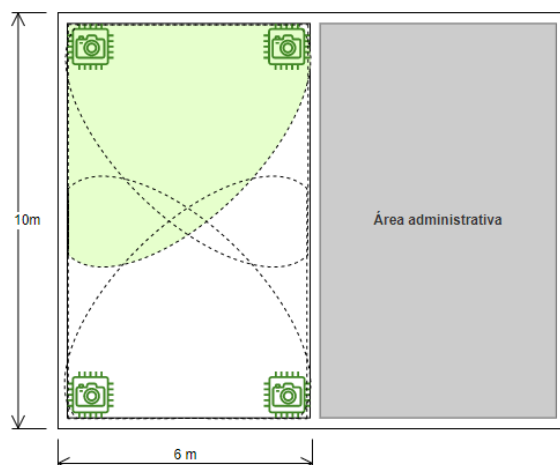


Figura 2. Zonas del banco delimitadas y posicionamiento de las cámaras térmicas. La zona verde representa el campo de visión horizontal de una cámara.

A continuación, en la figura 3, se presenta una vista en tres dimensiones del edificio donde se distinguen ambas secciones del banco. En el área de clientes se muestra la ubicación de las cámaras térmicas, así como de los demás componentes del sistema. El dispositivo que se observa en el centro del techo del edificio es el controlador y un parlante que se encargará de alertar si las cámaras detectaron alguna anomalía. Al lado de cada cámara se coloca un LED que se encenderá si la cámara mide una temperatura mayor a 38 grados, que es la que se considera como elevada para el ser humano.

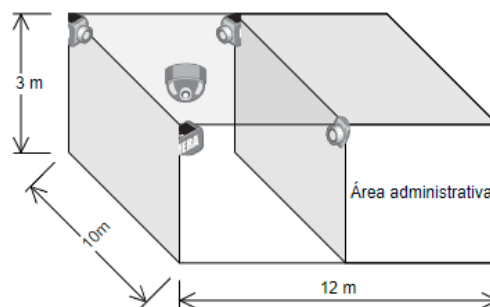


Figura 3. Vista tridimensional del edificio del banco con la ubicación de los componentes.

Seguidamente, en la figura 4 se observa una vista lateral de la visión de las cámaras un ángulo. Se presenta un campo de visión vertical de 75° , ya que ese es el ángulo de visibilidad que tiene la cámara que se utilizará en el prototipo. Se consideró este campo debido a que, como se muestra en la imagen, es posible medir la temperatura de toda la persona a partir de menos de un metro del sensor, y con un alcance suficiente como se explicó anteriormente.

Con el fin de maximizar el aspecto de que el sistema sea no invasivo, como se presenta en la imagen 3 y 4, se sitúan las cámaras térmicas en las esquinas superiores que conforman las paredes del edificio. Se propone que se deje un ángulo pequeño de margen con el techo, tal como se presenta en la siguiente figura, para obtener un campo de visión óptimo.

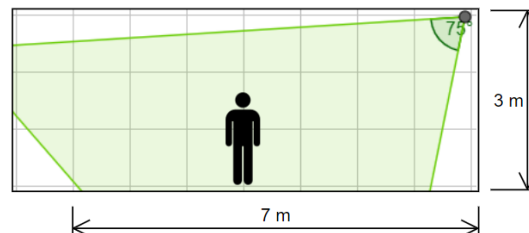


Figura 4. Zona del banco delimitada

II-C. Casos extremos

Para el diseño planteado, en resumen se tiene los siguientes casos extremos:

- Como caso extremo esta la detección de calores externos a las personas, tanto como comidas o algún dispositivo que produzca calor, esto producirá falsas alarmas sobre el COVID-19.
- De la misma manera, las personas que recién hacen ejercicios y vienen con un alto calor corporal, activan la alarma sin la necesidad de estar contagiados.
- Otra característica principal, es que las cámaras no reconocen entre humanos y animales, y si bien es cierto cualquier lugar público es propenso a que ingresen animales y esto también produciría falsas alarmas, por lo que se recomienda tener fuerte control de paso de ellos.
- Como punto importante es una la ubicación de las personas con temperatura, no dirá donde está ubicado el punto de calor, haciendo que este paso de reconocimiento sea de manera manual.

III. COMPONENTES DEL DISEÑO

Para elaborar el diseño anterior, se debe realizar la escogencia de los elementos que mejor se ajusten entre sí y a nivel de rendimiento y costo.

III-A. Elección del microcontrolador

Para el microcontrolador se ha decidido usar el Cortex-M3 de ARM, consideramos que es el que mejor se adapta para este diseño, en el cual el microcontrolador procederá a manejar y monitorear los sensores térmicos en alta frecuencia, además de los altavoces, luces LEDs y el reproductor de tipo MP3. Las características del microcontrolador son las siguientes:

- Presenta un arquitectura ARMV7-M de 32 bits.
- Múltiples buses de alto rendimiento para distintos I/O.
- Posee un set amplio que facilita la programación para el diseño.
- Instrucciones integradas de procesamiento de bits y bandas de bits a nivel de bus.

Analizando las plataformas electrónicas para la construcción del diseño, se procedió a escoger un Arduino DUE. Debido a que el diseño esta presente en una arquitectura ARM y este justamente esta desarrollada en la misma.



Figura 5. Arduino DUE

Características del Arduino DUE:

- EL Arduino Due funciona a 3.3 V.
- Tarjeta de microcontrolador basada en la CPU Atmel SAM3X8E ARM Cortex-M3.
- 12 entradas analógicas.
- Tiene 54 pines de entrada/salida digitales.
- 4 UARTs (puertos serie por hardware).
- 2 puertos para interfaz I2C

III-B. Elección del sensor de medición de temperatura

Para la medición de la temperatura dentro del recinto, se ha procedido a tomar en elección una cámara termográfica, debido a que estas presentan un gran rango de detección en comparación a los sensores por láser. Las características de la cámara MLX90640BAA son:

- Para distancias de hasta 7 m.
- Posee un pin de interrupción configurable que se puede disparar cuando un píxel individual supera o sobrepasa el umbral establecido.

- Matriz IR de 32x24 pixeles de pequeño tamaño y bajo coste.
- Calibrado de fábrica.
- Tensión de alimentación de 3.3 V/5 V.
- Consumo de corriente inferior a 23mA.
- Campo de visión 110°x75°.
- Protocolo de comunicación I2C.
- Temperatura de funcionamiento -40°C a 85°C.
- Frecuencia de refresco programable de 0.5 Hz a 64 Hz.
- Temperatura objetivo -40°C a 300°C.
- Resolución de $\pm 1^\circ\text{C}$.

III-C. Elección del dispositivo para alertar

En la circunstancia que la cámara detecte a una persona con una temperatura mayor a la especificada, se deberá proceder a emitir una alerta para poder tratar esta circunstancia. Por esta razón, se procede a utilizar un parlante que alerte a la seguridad por medio de un sonido si hay un sospechoso y puedan atender a este. Además del parlante se procederá a conectar un LED luminoso para la ayuda del proceso de alerta para tener un aspecto visual tomando en cuenta la posibilidad de alguna circunstancia por discapacidad u/o otros inconvenientes. El parlante escogido para este diseño es el CQRR0508 de la marca CQRobot, sus características son las siguientes:

- Compatible con Arduino, Robot, Raspberry Pi, máquina de publicidad, máquina de juegos, altavoz especial de máquina integrada, entre otros.
- Impedancia: 8 ohmios $\pm 15\%$.
- Potencia: potencia nominal / potencia máxima 3,0W/3,5W.
- Banda de frecuencias de efecto: hasta 20 KHZ MAX:20dB.



Figura 6. Altavoz

Adicional al parlante, se procede a agregar un módulo para la reproducción de dicho sonido alertante, para esto se hace uso del DFR0768, sus características se muestran a continuación:

- Soporta cuatro modos de control: Arduino, comando AT, botones integrados y ADKEY.
- Voltaje de operación 3.3V/5V.
- Protocolo de comunicación UART.

- Salida estéreo de doble canal.

III-D. Estimación de costos

Para este diseño se procedió a realizar el cálculo de los costos por cada componente, así como de adicionales para la instalación del diseño, como lo serían los cables, canaletas, entre otros.

Cuadro I
ESTIMACIÓN DE COSTOS DE LOS COMPONENTES DEL SISTEMA

Componentes	Cantidad	Precio (\$)	Total (\$)
Cámara Térmica	4	59,32	237,28
Arduino DUE	1	61,95	69,95
Carcasa para el Arduino	1	3,5	3,5
Fuente de Alimentación 120V @60Hz a 7 V	1	9	9
Metro Cable AWG Calibre 20	106.5	0,21	22,5
LED 10mm	4	0,55	2,2
Módulo MP3	1	9	9
Altavoz	1	7,99	7,99
Metro Canaletas plásticas	26.6	3,27	87

El total de los costos en componentes es de 448,42 dólares por unidad construida por lo que, para un presupuesto de 10 sistemas desarrollados sería por un valor de 4484,2 dólares. Para nuestro diseño, justificamos la cantidad de cable debido a que se consideran 4 líneas por cámara. En este aspecto tomamos en cuenta la distancia vertical de la cámara hasta el techo y por consiguiente la distancia diagonal del techo al centro, donde se ubica el Arduino. De la misma forma, se hace uso de estas distancias para calcular los metros de canaleta. (Los metros de cable son mayores a los de las canaletas debido a las 4 líneas por cámara).

Por consiguiente, para completar el costo de las 10 unidades, se debe de contemplar las horas invertidas por ingeniero, en Costa Rica el costo por hora es de aproximadamente 40 dólares. A continuación, se procederá a representar las horas invertidas por cada uno en la etapa designada del diseño.

Cuadro II
HORAS INVERTIDAS

Etapas	Ingeniero(s)	Horas
Diseño del prototipo	Guillen, Juan Pablo, Julian e Ethan	6
Diseño de software	Ethan, Julian	2
Parámetros de Manufactura	Guillen, Juan Pablo, Julian e Ethan	2
Diseño Ingenieril	Guillen, Juan Pablo	4
Reuniones y recolección de información para el potencial industrial	Guillen, Juan Pablo, Julian e Ethan	2
Pruebas de funcionamiento	Ethan	1
Ajustes electrónicos	Julian	2
Elaboración del documento IEEE	Guillen, Juan Pablo, Julian e Ethan	6

En total se utilizaron 28 horas para todo el proceso del diseño el cual fue dividido entre 4 ingenieros en las distintas

tareas a realizar. Cabe destacar que cada ingeniero participó en tareas específicas, por lo que no estuvo presente para todo, con el fin de tener una mejor distribución del tiempo y así alcanzar la mayor eficiencia para el desarrollo del diseño, por lo que el costo total de la mano de obra es aproximadamente 1000 dólares. Además de que este costo de mano de obra solo se procede a pagar una vez por los 10 prototipos. Adicional a esto habrá un costo de 15 dólares por unidad para su ensamblaje adicionando los materiales.

El proyecto completo tendrá un costo aproximado de 5634 dólares contemplando el costo de 10 prototipos.

IV. DIAGRAMAS

En esta sección, se presentará una serie de diagramas en los cuales será posible visualizar el funcionamiento del sistema, al mostrar la interacción existente entre los diferentes componentes que lo integran.

IV-A. Diagrama de primer nivel

En este primer diagrama, se observa como el dispositivo recibirá 4 entradas, referentes a las cuatro cámaras térmicas, las cuales serán controladas y administradas por un sistema de control de temperaturas que determina cuando se activará la alarma y cuando se muestran los resultados en la pantalla.

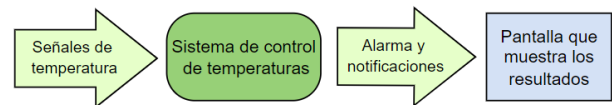


Figura 7. Diagrama de primer nivel

IV-B. Diagrama de segundo nivel

En el siguiente diagrama se observa el mismo funcionamiento anterior, pero con entradas detalladas conectadas a un sensor de temperatura y sus salidas se componen de la alarma y de la pantalla.

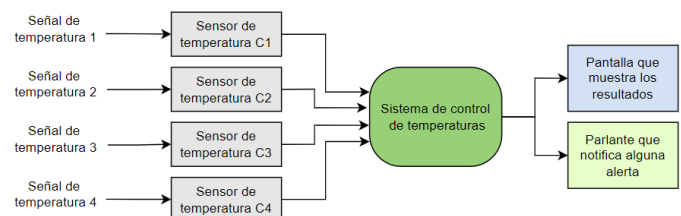


Figura 8. Diagrama de segundo nivel

IV-C. Máquina de estados

Para el análisis del código se procedió a realizar una máquina de estados de Mealy, para poder entender el proyecto en términos de código y variables lógicas. Se observan varias variables como ON, la cual indica si el sistema está encendido o apagado, representado con 1 y 0 respectivamente. Si se encuentra apagado, el sistema se mantiene en cero, pero si este pasa a 1, entra en el estado en espera de señal de los sensores y dependiendo de cuál sensor se active, así se activará su respectiva variable. Finaliza activando la alarma y la pantalla cuando las alarmas se activan. Pasado un tiempo, estas se desactivan volviendo al estado inicial.

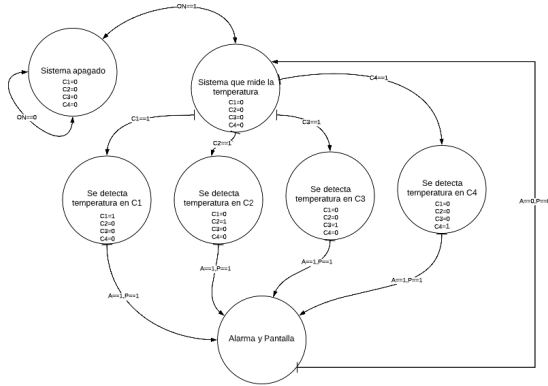


Figura 9. Máquina de estados de Mealy

V. IMPLEMENTACIÓN DE HARDWARE

En la presente sección, se presenta la implementación de hardware, en la cual se muestran las conexiones de los diversos periféricos con el procesador, así como la explicación de la manera en que estos se comunican.

En la figura 10, se muestra la imagen del procesador con cada uno de sus puertos y partes que lo integran. Conectado a algunos de estos, se presentan los componentes del sistema que fueron descritos en secciones anteriores.

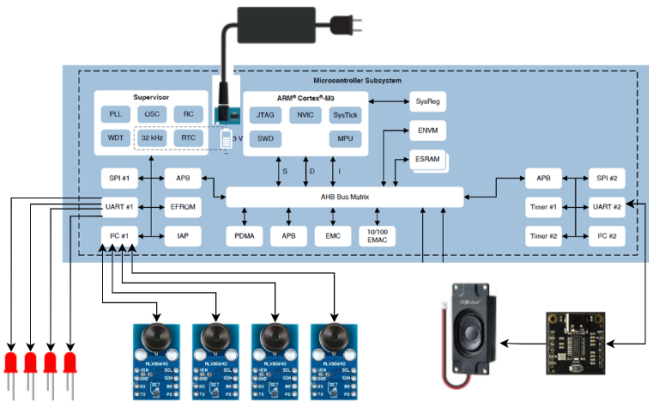


Figura 10. Conexiones en la implementación del hardware.

V-A. Alimentación

En la parte superior, se encuentra la alimentación que debe tener el Arduino. Según su hoja de datos, esta puede ser

suministrada por medio de una conexión USB o una fuente de alimentación externa. Esta alimentación externa puede ser suministrada por un adaptador AC/DC o por una batería. Para este sistema se utilizará la opción del adaptador, que según las especificaciones puede conectarse mediante un conector de alimentación de 2,1 mm con un contacto positivo en el centro.

Adicionalmente, respecto a este tipo de alimentación, la hoja de datos determina que el Arduino puede funcionar con una tensión límite de 6 a 20 voltios. Sin embargo, el rango recomendado de tensión de entrada para el funcionamiento óptimo de la placa es de 7 a 12 V. Por esta razón, se propone el uso de una fuente de alimentación que convierta los 120 V en AC de los tomacorrientes comunes, a 7 V en DC.

V-B. Conexión de las cámaras térmicas

En el puerto I2C #1 que se presenta en el procesador, se conectarán las cuatro cámaras termográficas necesarias en el sistema. Esto con el fin de utilizar la característica de este bus que permite la inclusión de varios dispositivos periféricos para el mismo controlador, que este caso será el Arduino DUE.

A pesar de que el Arduino seleccionado cuenta con puertos tanto UART como I2C, se decide utilizar el segundo, ya que este por lo general es más rápido que el UART. Este factor es de gran importancia para tener una comunicación efectiva entre los dispositivos controlador y periféricos, en la cual los datos captados por las cámaras sean transmitidos de manera veloz hacia el procesador, con el fin de que el sistema sea capaz de monitorear de manera óptima. Para esta interacción, se consideró que la característica half-duplex propia del protocolo I2C es suficiente, debido a que únicamente se necesita que el controlador reciba los datos de las cámaras.

Adicionalmente, este bus cuenta con la ventaja de que únicamente necesita la conexión en paralelo de los dispositivos en las líneas de datos y de reloj, que corresponden a la SDA y SCL respectivamente, así como las respectivas tensiones V_{CC} y GND. Dicha característica es también beneficiosa para reducir el tamaño físico del sistema.

El protocolo de comunicación será el I2C, que permitirá que el controlador envíe a los periféricos la señal de inicio, identifique a cada uno de ellos a través de sus direcciones para posteriormente funcionar como receptor de los datos que envían dichos periféricos en palabras de 8 bits.

V-C. Conexión del dispositivo de audio

El módulo para la reproducción del sonido de alerta ante la detección de una persona con posibles síntomas, se conectará en el puerto UART #2, tal como se muestra en la figura 10. Este componente se conecta al procesador de esta manera debido a que utiliza dicho protocolo de comunicación.

Este bus tiene la ventaja de poder funcionar como half-duplex o full-duplex, que permitirá la transmisión de datos en ambos sentidos si es necesaria en algún punto del funcionamiento del sistema. Para esto, se conectan los tres cables indispensables para este tipo de comunicación, como lo son el Tx, Rx y tierra.

Respecto a esta conexión, resulta importante apuntar que como este protocolo es asíncrono, los dos dispositivos conectados en este puerto se deben sincronizar al mismo baud rate.

El módulo de reproducción será el responsable de controlar la emisión de sonido a través del parlante cuando el estado del sistema lo requiera.

V-D. Conexión de los LEDs

Los cuatro LEDs se conectan al UART #1 debido a que este se encuentra disponible y estos no requieren de características como una gran velocidad de transmisión de datos o algunas de las propiedades de algún puerto en específico.

V-E. Conexión de los periféricos con el procesador

Cada uno de los buses a los que se conectan los periféricos que componen el sistema se conectarán a un bus APB. Esto se debe a que este tipo de bus se encuentra optimizado para la conexión de dispositivos de entrada y salida (I/O) que no requieran de un rendimiento muy elevado, como lo son los elementos que se utilizan en el proyecto.

Este bus, a su vez se conectará a un AHB, que se encargará de establecer la comunicación con el procesador. Dicha asignación es conveniente para este tipo de bus, ya que posee un gran rendimiento.

VI. PROGRAMACIÓN EN ENSAMBLADOR ARM CORTEX-M3

Para diseñar la programación básica para el sistema detector de temperatura, se hizo uso del programa llamado Keil µVision. De primero, se procede a definir los registros para el desarrollo de la programación.

- R0: representa si el sistema de detección esta encendido o apagado.
- R1: representa si el sensor 1 ha detectado un punto de alta temperatura.
- R2: representa si el sensor 2 ha detectado un punto de alta temperatura.
- R3: representa si el sensor 3 ha detectado un punto de alta temperatura.
- R4: representa si el sensor 4 ha detectado un punto de alta temperatura.
- R5: representa cual LED debe encenderse, 1 para el LED1, 2 para el LED2 y así respectivamente.
- R6: representa si la alarma de detección de alta temperatura esta activa.
- R7: representa el estado del temporizador de la alarma, en este caso el 0 determina que falta tiempo con la alarma encendida y 1 cuando ya ha pasado el tiempo.

A nivel general de cada registro, se define que el valor de 0 es cuando esta apagado o es negativa la lectura y el 1 corresponde al encendido o una lectura positivo como convencionalmente se usa.

Para la programación se procede a utilizar el módulo heatDetection para el uso de la sintaxis general de un Cortex-M3.

```
AREA heatDetection, CODE, READONLY
ENTRY
EXPORT __main
```

Figura 11. Encabezado de la programación.

El main es quien dirige el curso de la programación y en este se procede a definir los valores para cada uno de los LEDs que se usan en el diseño.

```
__main

LED1 EQU 1
LED2 EQU 2
LED3 EQU 3
LED4 EQU 4
```

Figura 12. Definición dentro del main de los LEDs.

Por consiguiente la subrutina MAIN es la que se ejecuta de primero. En esta subrutina se hace un llamado al waitSystemBoot. Este se encarga del registro R0 hasta el momento que se encuentre en 1, básicamente hasta que el sistema de detección de temperatura se encuentre en 1, cuando esto suceda se procede a realizar un redefinición de todos los valores de los registros a 0. Por consiguiente, cuando finaliza el proceso de redefinición de los registros se procede a llamar a la subrutina measuringHeat para la medición de la temperatura.

```
MAIN      B waitSystemBoot

waitSystemBoot    CMP R0, #1
                  BNE waitSystemBoot
                  B rebootRegisters

rebootRegisters   MOV R0, #0
                  MOV R1, #0
                  MOV R2, #0
                  MOV R3, #0
                  MOV R4, #0
                  MOV R5, #0
                  MOV R6, #0
                  MOV R7, #0
                  B measuringHeat
```

Figura 13. Reseteo de los registros después de el waitSystemBoot.

En la figura 14, se muestra la subrutina measuringHeat, esta subrutina lo que hace es verificar si alguno de los registros R1, R2, R3 o R4 se encuentran en 1, lo cual indica que se ha detectado un punto de temperatura mayor a los 37 grados centígrados. Basándonos en el diagrama de la figura 2, estos serían equivalentes a:

- 1) R1: El sensor ubicado en la esquina superior izquierda ha detectado una temperatura alta.
- 2) R2: El sensor ubicado en la esquina inferior izquierda ha detectado una temperatura alta.
- 3) R3: El sensor ubicado en la esquina inferior derecha ha detectado una temperatura alta.

4) R4: El sensor ubicado en la esquina superior derecha ha detectado una temperatura alta.

Si sucede que alguno de los sensores coloca un 1 en los registros mencionados, lo siguiente que se hace es llamar a las subrutinas isAorD, isAorB, isBorC o isCorD, las cuales se explicarán a continuación.

```
measuringHeat    CMP R1, #1
                 BEQ isAorD
                 CMP R2, #1
                 BEQ isAorB
                 CMP R3, #1
                 BEQ isBorC
                 CMP R4, #1
                 BEQ isCorD
                 B measuringHeat
```

Figura 14. Subrutina para indicar que hay un punto con temperatura alta.

Acá básicamente lo que tenemos son cuatro subrutinas, las cuales se encargan de identificar con mayor precisión en cuál de las cuatro zonas se ha identificado el punto de alta temperatura.

Entonces, primero tenemos la rutina isAorD, la cual verifica si existe una detección de alta temperatura según el registro R2. En caso de que esto sea correcto, se prosigue con la subrutina heatInA, puesto que se ha determinado que el punto de alta temperatura procede del sector A. En el caso que esto sea falso, se concluye que la temperatura proviene del sector D. Debido a este funcionamiento, es que las subrutinas reciben el nombre de isAorD, isAorB, isBorC y isCorD.

En la Figura 15 se muestran el resto de estas subrutinas, que se rigen bajo la misma lógica, asociando las mediciones con los cuatro posibles sectores identificados para las ubicaciones físicas dentro del caso planteado.

```
isAorD          CMP R2, #1
                 BEQ heatInA
                 BNE heatInD

isAorB          CMP R1, #1
                 BEQ heatInA
                 BNE heatInB

isBorC          CMP R2, #1
                 BEQ heatInB
                 BNE heatInC

isCorD          CMP R3, #1
                 BEQ heatInC
                 BNE heatInD
```

Figura 15. Subrutina para identificar la zona en la que hay un punto con temperatura alta.

Ahora ya por último, tenemos las subrutinas heatIn, donde si por ejemplo, se llama a la subrutina heatInA (debido a que R1 y R2 están en 1). Se coloca a la constante LED1 en el

registro R5 (ya que es registro asignado para indicar el LED que se debe encender) y se coloca un 1 en el registro R6 (el cual indica que se debe encender la alarma).

Luego, se valida si el registro R7 está en 1 (el cual indica que ya han pasado los 3 segundos de duración de la alarma) y en el momento en que se hayan cumplido los 3 segundos, se termina por llamar a la subrutina rebootRegisters mostrada anteriormente, volviendo a comenzar nuevamente con todo el proceso de detección de temperatura.

Se tiene una subrutina heatIn trabajando con esta misma lógica para cada uno de los cuatro sectores físicos en los cuales se divide el área de detección de temperatura. La codificación de estas se muestra en la figura 16.

```
heatInA         MOV R5, #LED1
                 MOV R6, #1
                 CMP R7, #1
                 BNE heatInA
                 B rebootRegisters

heatInB         MOV R5, #LED2
                 MOV R6, #1
                 CMP R7, #3
                 BNE heatInB
                 B rebootRegisters

heatInC         MOV R5, #LED3
                 MOV R6, #1
                 CMP R7, #3
                 BNE heatInC
                 B rebootRegisters

heatInD         MOV R5, #LED4
                 MOV R6, #1
                 CMP R7, #3
                 BNE heatInC
                 B rebootRegisters

END
```

Figura 16. Subrutinas heatIn.

A partir de esto, se garantiza una mayor efectividad en la detección de los puntos de alta temperatura e identificación de los sectores correspondientes planteados en nuestro caso de trabajo.

VII. ANEXOS

VII-A. Participación de los integrantes del grupo

En la presente sección se detalla la participación en horas por todos los integrantes del grupo.

Nombre del colaborador - ID	Escuela	Horas
José Julián Camacho - 2019201459	Computadores	20
Leonardo Guillén - 2019031688	Computadores	20
Juan Pablo Carrillo - 2019380111	Computadores	20
Ethan Flores Montero - 2017152644	Electrónica	20

REFERENCIAS

- [1] Farnell. Electronic Component Distributors. <https://www.farnell.com/datasheets/1682211.pdf> (accedido el 18 de mayo de 2022).
- [2] "ÅµVision User's Guide: ÅµVision User's Guide". Keil Embedded Development Tools for Arm, Cortex-M, Cortex-R4, 8051, C166, and 251 processor families. <https://www.keil.com/support/man/docs/uv4/> (accedido el 20 de mayo de 2022).
- [3] "Due — Arduino Documentation — Arduino Documentation". Arduino Docs — Arduino Documentation — Arduino Documentation. <https://docs.arduino.cc/hardware/due> (accedido el 20 de mayo de 2022).
- [4] Cortex-M3". Arm Developer. <https://developer.arm.com/Processors/Cortex-M3> (accedido el 20 de mayo de 2022).