

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624: LABORATORIO DE MICROCONTROLADORES

Proyecto final

Sistema de interacción digital con ESP8266 y MLX90640 para experiencias inmersivas

Prof. Marco Villalta

Estudiantes:
Sofía Fonseca Muñoz, B42634

27 de febrero de 2023

Índice

1. Introducción/Resumen	2
2. Objetivos	2
2.1. Objetivo general	2
2.2. Objetivos específicos	2
3. Alcances	2
4. Justificación	3
5. Marco teórico	4
5.1. Adafruit Feather HUZZAH ESP8266	4
5.1.1. Características eléctricas	6
5.2. MLX90640	6
5.3. Ensamblaje	7
5.3.1. Componentes electrónicos	8
5.4. Processing	8
5.5. Bibliotecas	8
6. Desarrollo y análisis de resultados	9
6.1. Programación del ESP8266	9
6.2. Comunicación por puerto serial	10
6.3. Generación de imagen en Processing	10
6.4. Resultado	12
7. Conclusiones y recomendaciones	14
Referencias	15
8. Apéndice	16

1. Introducción/Resumen

Este proyecto consiste en la creación de un dispositivo que le permita a personas interactuar con una obra de arte a través de una pantalla y un sensor. Cuando la persona se pone frente a la pantalla, podrá ver su silueta como en un espejo, sin embargo, no verá los detalles sino que verá imágenes relacionadas a una obra de arte, que desaparecerán si la persona se mueve y reaparecerán con una animación cuando la persona se acerca nuevamente.

Para eso se utilizará un ESP8266 y la cámara termal MLX90640. La elección de estos elementos se debe a que el ESP8266 es un microcontrolador de bajo costo que permite el procesamiento de datos necesario y puede enviar la información ya sea por wifi o por puerto serial a la computadora. [6] La cámara termal MLX90640 se escogió por su gran cantidad de pixeles que permiten ver una imagen más nítida de la persona. [5]

La cámara termal conectada al microcontrolador se encargan de enviar los datos por medio de una plataforma de AdaDAfruit. Una vez que los datos que forman la imagen termal de una persona se envían, estos serán procesados en la computadora para formar la imagen animada que se puede presentar en una pantalla por medio de Processing.

Este proyecto puede ser una solución de bajo costo para museos, ya que permite darle vida a obras de artes y artistas por medio de la interacción en un mundo donde nos encontramos tan expuesto a estímulos.

2. Objetivos

2.1. Objetivo general

Desarrollar un sistema de interacción digital con ESP8266 y MLX90640 para experiencias immersivas.

2.2. Objetivos específicos

1. Identificar los elementos físicos y de software necesarios para el desarrollo del proyecto y la implementación del diseño.
2. Preparar un esquema físico de la conexión de los elementos requeridos para el desarrollo del sistema.
3. Implementar el código fuente para la obtención de los datos de la cámara termal en la plataforma digital.
4. Implementar el código fuente para la obtención de las animaciones que de la imagen.

3. Alcances

Al desarrollar este proyecto, cada uno de los objetivos específicos reúne una lista de tareas particulares se requieren para su cumplimiento. En la siguiente tabla, se puede encontrar la lista de pasos que se siguieron en la ejecución del trabajo.

Objetivos específicos	Pasos y alcances
1. Identificar los elementos físicos y de software necesarios para el desarrollo del proyecto y la implementación del diseño.	1. Revisión bibliográfica sobre ESP8266, MLX90640 y otros elementos que faciliten la solución del problema. 2. Esquema de funcionalidad, donde se presenta cuál función va a realizar cada elemento, incluso la pantalla.
2. Preparar un esquema físico de la conexión de los elementos requeridos para el desarrollo del sistema.	1. Tener un circuito físico que represente la solución física del problema. 2. Probar códigos de fuente libre que se adapten al sistema físico para probar su funcionamiento.
3. Implementar el código fuente para la obtención de los datos de la cámara termal en la plataforma digital.	1. Plataforma digital en computadora donde se reciben los datos de la cámara termal a través del ESP8266.
4. Implementar el código fuente para la obtención de las animaciones que dé la imagen obtenida por la cámara termal.	1. Creación del código necesario para la obtención de las animaciones en tiempo real de la persona moviéndose.

Tabla 1: Pasos a seguir para la elaboración del proyecto.

4. Justificación

En la actualidad se encuentra en auge las experiencias inmersivas como parte de muchos museos y exposiciones. Un claro ejemplo de ello fue la exposición de **Beyond Van Gogh: the Immersive Experience** que se presenció durante el año 2022 en Costa Rica. [1]

Este tipo de experiencias han tenido mucho éxito ya que permiten a los visitantes ir más allá de ser un observador al convertirse en protagonistas del espacio en el que se encuentran. [2] En la actualidad, tenemos acceso a información rápidamente a través de internet que podemos acceder fácilmente desde nuestros celulares todo el tiempo. Esto hace que la función de los museos se vea obligada a cambiar y brinde a la experiencia algo más allá de información. Como dice la descripción de Beyond Van Gogh: "*Las obras maestras, ahora libres de marcos, cobran vida, aparecen y desaparecen, fluyen a través de múltiples superficies, la minucia de los detalles alteran nuestro sentido.*"

Por otro lado, debido a la pandemia global que estamos viviendo, esta manera de interacción nos permite distanciarnos físicamente pero conectar por medio de las obras.

Por último, es importante mencionar que este proyecto se puede lograr con un costo muy bajo ya que requiere de pocos elementos. La conexión no requiere un nivel técnico alto y se utilizan bibliotecas de código abierto. Por lo que se puede replicar fácilmente, dando acceso a museos que no tienen un presupuesto muy elevado para el desarrollo de proyectos tecnológicos.

5. Marco teórico

5.1. Adafruit Feather HUZZAH ESP8266

Para este proyecto se utilizó la placa Adafruit Feather HUZZAH ESP8266. Esta placa recibe su nombre gracias a su tamaño y peso que no supera 15 cm² y 10 g respectivamente. Fue creado con la intención de ser un microcontrolador que pueda tener funcionalidades intermedias pero que a la vez pueda ser portátil. Otro elemento que lo hace portátil, es la capacidad de tener una batería de litio de 3.7 V como fuente de poder que puede ser recargada si la placa se reconecta por medio de USB. [3]

Esta placa contiene el microcontrolador ESP8266 WiFi, el cual funciona a una frecuencia de 80 Hz y opera lógicamente a 3.3 V. Este microcontrolador puede ser programado fácilmente con Arduino pero también se puede programar por medio de MicroPython. Otra característica que beneficia su portabilidad es el módulo de WiFi, que a su vez le da un uso específico enfocado en Internet de las cosas. [3]

Algunas de sus características principales son su memoria Flash de 32 MBit, un regulador de voltaje de 3.3 V con una corriente máxima de 500 mA, un convertidor de puerto USB-Serial que permite velocidades de baudrate de hasta 921600 para rápidas transferencias, 9 pines de entrada y salida digital, 1 pin analógico con un máximo de 1V, un LED rojo y un LED azul para uso general, un pin de poder y un botón de reset.

Otra característica del microcontrolador en su diseño portátil es la tecnología que le permite funcionar con el mínimo poder. Tiene 3 modos: modo activo, modo de hibernación y modo de hibernación profunda. En este último modo consume 60 uA, y sólo el reloj y el watchdog se encuentran funcionando. El reloj puede ser programado para despertar el microcontrolador o el microcontrolador puede ser programado para que con un mínimo cambio se active.

En la siguiente imagen se puede observar el diagrama de bloques del microcontrolador donde integra todas las características que fueron mencionadas anteriormente:

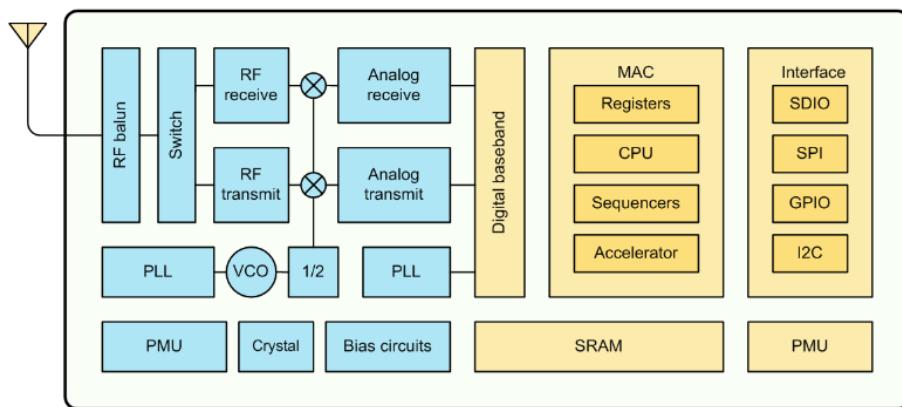


Figura 1: Diagrama de bloques del microcontrolador. Tomada de https://www.esp8266.com/wiki/doku.php?id=download_datasheets_for_esp8266

La siguiente imagen muestra el diagrama de pines de este microcontrolador. Sin embargo, en este caso este microcontrolador se encuentra integrado en la placa HUZZAH de Adafruit.

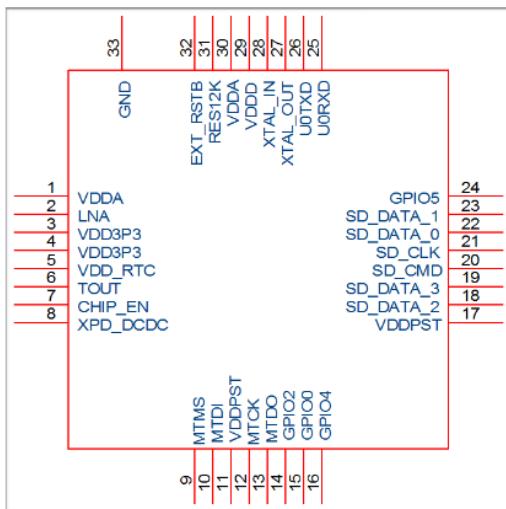


Figura 2: Diagrama de pines del microcontrolador ESP8266. Tomada de https://www.esp8266.com/wiki/lib/exe/fetch.php?media=0a-esp8266_datasheet_en_v4.3.pdf

Adafruit provee un diagrama de la placa para facilitar su uso, ya que el microcontrolador se accede a través de ella. Se muestra a continuación:

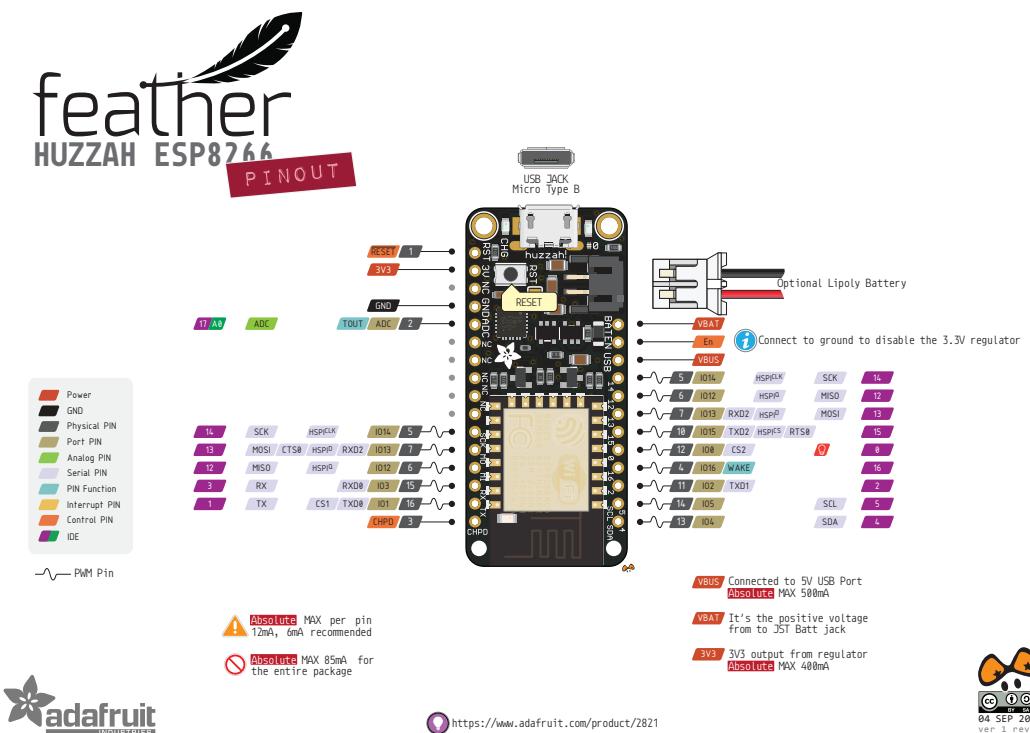


Figura 3: Diagrama de pines de la placa HUZZAH. Tomada de https://cdn-learn.adafruit.com/assets/assets/000/046/211/original/Huzzah_ESP8266_Pinout_v1.2.pdf?1504807178

5.1.1. Características eléctricas

Para utilizar este microcontrolador, es importante tomar en cuenta las características eléctricas. Esto para asegurar un comportamiento adecuado y estar seguros de que los componentes externos que se desean usar sean adecuados para el microcontrolador y viceversa. La siguiente tabla muestra las características eléctricas más importantes: [4]

Especificaciones eléctricas	Min	Típico	Max
Temperatura de almacenamiento	-40°C		+125°C
Temperatura para soldar			260°C
Voltaje de operación	3.0 V	3.3 V	3.6 V
Voltaje de entrada	-0.3/0.75V		0.25/3.6 V
Voltaje entre pines y Vss	0.8V		0.1 V
Máxima corriente por cada pin	12 mA		
Descarga electrostática a 25°C por contacto humano			2 kV
Descarga electrostática a 25°C por material conductor			0.5 kV

5.2. MLX90640

El sensor Adafruit MLX90640 es una cámara termal que permite agregar visión termal a bajo costo. Está formada por un arreglo de 24x32 sensores termales infrarrojos, lo que permite por medio de la conexión I2C a un microcontrolador, transmitir 768 lecturas de temperatura. Además es un componente bastante portátil debido a su tamaño que no supera los 2 cm².[5]

Esta cámara tiene un campo de visión de 55°x35° el cual funciona para proyectos con poco rango de detección pero existe una versión que tiene un campo de visión de 110°x70° para proyectos que requieran una visión más amplia. Puede leer temperaturas entre -40°C y 300°C con una precisión de +- 2°C en un rango de 0-100°C. [5]

Para su mejor desempeño, este sensor funciona a 16 Hz. Es compatible con plataformas como Raspberry Pi y Arduino y puede ser programado con MicroPython o con Arduino. Además requiere un poder de procesamiento de al menos 20KB de RAM. Entre las características eléctricas más importantes se encuentra su voltaje de entrada debe ser entre 3.3V y 5V, con un regulador de 3.3 V, consume menos de 23 mA y opera en temperaturas entre -40°C y 85°C. [5]

Este sensor cuenta con 5 pines que se pueden observar en la siguiente imagen:



Figura 4: Fotografía del sensor MLX90640. Tomada de <https://learn.adafruit.com/adafruit-mlx90640-ir-thermal-camera/pinouts>

Estos pines se dividen en 3 pines de poder y 2 pines de control lógico: [5]

- **Vin:** Este es el pin de poder. La cámara funciona a 3.3 V, pero gracias a su regulador de voltaje se puede conectar una fuente de hasta 5 V.
- **3V:** Este pin de salida funciona como el regulador de voltaje de 3.3 V, se puede tomar hasta 100 mA.
- **GND:** Conexión a tierra común para poder y lógica.
- **SCL:** Este es el pin de reloj de I2C. Tanto en este como en el pin de datos se puede usar lógica entre 3 y 5 V.
- **SDA:** Este es el pin de data de I2C.

5.3. Ensamblaje

Conociendo las características de ambos componentes, se comprueba que se puede realizar una conexión entre ellos ya que ambos se pueden programar por medio de Arduino. Además, el microcontrolador opera a 3.3 V al igual que la cámara termal. A continuación se muestra como se realizaron las conexiones entre ellos:

Pines ESP8266	Pines MLX90640
3V3	Vin
GND	GND
D1	SCL
D2	SDA

El pin de 3V de la cámara no se conecta a nada debido a que la tensión de entrada es de 3.3 V y no necesita regular el voltaje. Por otro lado, los pines de control lógico se conectan a los GPIOs del microcontrolador, existen 10 de ellos y se escogen el D1 y D2 debido a que la biblioteca que se va a usar con la cámara la controla desde esos pines. La conexión resultante se muestra a continuación. Es importante tener cuidado con los cables para que estos no interfieran con la imagen de la cámara termal.

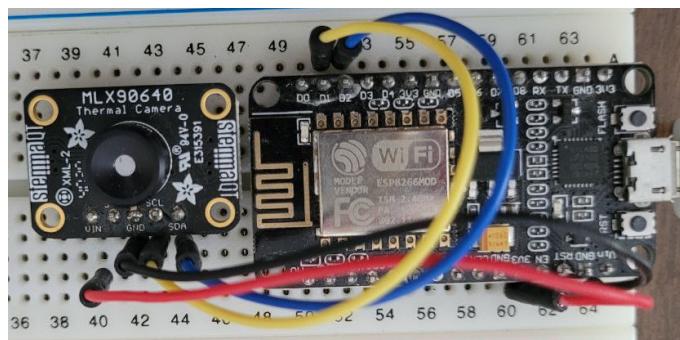


Figura 5: Fotografía de la conexión resultante para el proyecto. Figura propia.

5.3.1. Componentes electrónicos

El ensamblaje de este proyecto sólo requiere el microcontrolador y la cámara termal. Sin embargo hay que tomar en cuenta que se debe conectar a una computadora con monitor para tener la funcionalidad de la interacción. En este caso se asume que esos son elementos que se tienen. Para los componentes extra a continuación se encuentra el precio en colones tomando en cuenta un tipo de cambio de 565 colones son equilivantes a 1 USD:

Componente	Precio (colones)
Adafruit Feather HUZZAH ESP8266	8450
Cámara termal Adafruit MLX90640	42350
Precio total	50800

El precio total del proyecto son 50800 colones. El costo de los componentes es un poco alto pero la facilidad del uso permite que no se necesite un equipo con el conocimiento tecnológico sino que el usuario lo pueda hacer.

5.4. Processing

Creado en 2001, processing es una herramienta de software libre que permite crear bocetos y software enfocado en artes visuales. Processing ha resultado una buena herramienta para que personas relacionadas al arte visual se acerquen a la programación tanto como para personas relacionadas a la programación se acerquen más a artes visuales. [8]

5.5. Bibliotecas

Para el desarrollo de este proyecto se utilizaron dos lenguajes de programación. Uno de ellos fue C++ a través de Arduino para la programación del microcontrolador con la cámara termal. El otro fue para Processing, para hacer el procesamiento de los datos en la computadora al recibir una imagen animada.

Para programar el MCU se utilizaron tres librerías que serán explicadas a continuación [7]:

- **Wire.h:** es la biblioteca que necesita el microcontrolador para realizar las conexiones I2C. Se toma en cuenta tanto el establecimiento el reloj por medio de SCL y la configuración de datos por medio de SDA.
- **MLX90640_I2C_Driver.h:** contiene las funciones del sensor termal para manejar la información de la conexión I2C, estas funciones son específicas para la conexión con el ESP8266. Las funciones incluyen la configuración de frecuencia, la inicialización, la lectura y escritura de datos y reset general
- **MLX90640_API.h:** contiene las funciones que están relacionadas directamente al funcionamiento de la cámara termal. Las funciones se dividen en cuatro grupos: funciones de configuración, de preprocessamiento de datos, de adquisición de datos y de cálculos.

Por otro lado, en Processing solo fue necesario el uso de una biblioteca: [8]

- **serial:** esta biblioteca permite la lectura y escritura de datos de dispositivos externos con un byte a la vez. Esta biblioteca permite la flexibilidad de comunicarse con microcontroladores por medio del Protocolo de Comunicación Serial.

6. Desarrollo y análisis de resultados

Para entender el funcionamiento del producto esperado, es necesario tomar en cuenta el proceso que se requiere para realizarlo. A continuación se muestra un diagrama de flujos de las funcionalidades que tiene que cumplir el proyecto.

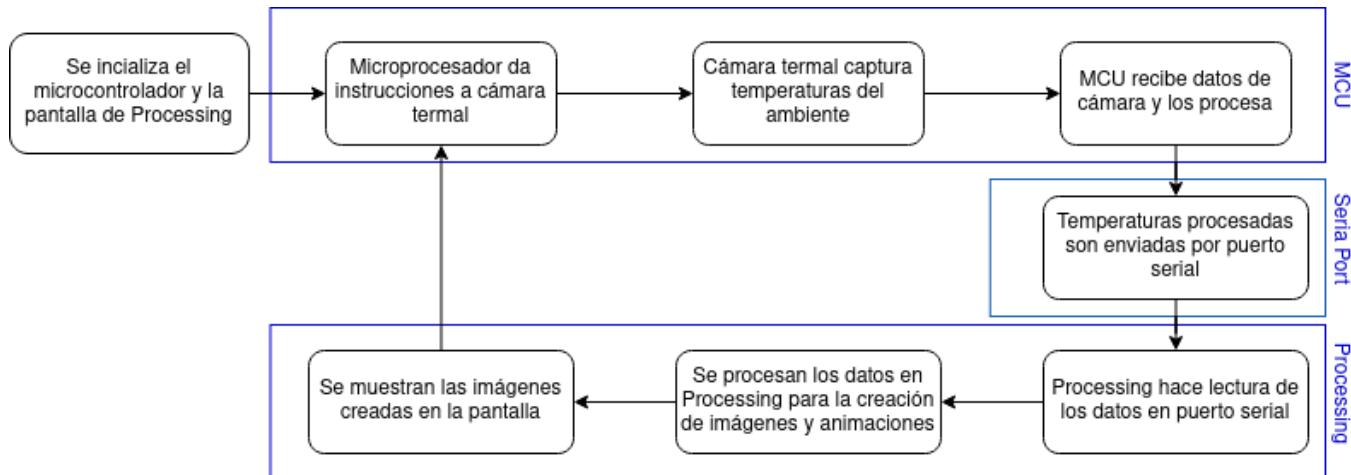


Figura 6: Diagrama de flujo del proyecto. Figura propia.

Tomando en cuenta el diagrama de flujo, se puede observar como el trabajo se divide principalmente en dos partes: una que es realizada por el microcontrolador y la otra que se logra por medio de Processing. La computadora a través del puerto serial, funciona como un intermediario para que las dos partes se puedan comunicar.

6.1. Programación del ESP8266

Las primeras tareas de este programa, se pueden lograr gracias a las funciones de la biblioteca de la cámara termal que fue explicada en la sección anterior. Estas funciones se ven reflejadas en el código como se muestra a continuación, los comentarios explican su funcionamiento:

```

1 //GetFrameData recolecta todos los datos relacionados a una captura
2 int status = MLX90640_GetFrameData(MLX90640_address, mlx90640Frame);
3
4 //GetVdd y GetTa hacen un promedio de la tensión y la temperatura de una captura
5 //respectivamente
6 float vdd = MLX90640_GetVdd(mlx90640Frame, &mlx90640);
7 float Ta = MLX90640_GetTa(mlx90640Frame, &mlx90640);
8 // Calcula la temperatura reflejada en los objetos a partir de la temperatura
9 ambiental
10 float tr = Ta - TA_SHIFT;
11 float emissivity = 0.95;
12
13 //Con la información de una captura, se procesan los datos tomando en cuenta la
14 //temperatura reflejada, la tensión y la emisión de los objetos para obtener el valor
15 //de temperatura de cada pixel
16 MLX90640_CalculateTo(mlx90640Frame, &mlx90640, emissivity, tr, mlx90640To);

```

6.2. Comunicación por puerto serial

Una vez que los datos han sido procesados estos son enviados al puerto serial para que Processing pueda acceder a ellos. Como es un arreglo de datos, se crea un bucle que permite enviar las 768 temperaturas correspondientes a cada 'pixel' de la cámara. Esto corresponde a la siguiente sección del código:

```

1   for (int x = 0 ; x < 768 ; x++)
2   {
3       Serial.print(mlx90640To[x], 2);
4       Serial.print(", ");
5   }

```

Es importante destacar que anteriormente se había configurado el puerto serial al baudrate de 115200, que es la mayor cantidad a la que se pueden transferir datos en este microcontrolador. El código de Processing se configura al mismo baudrate y una vez que está listo recibe los datos a través de la siguiente función:

```

1   if(myPort.available() > 5000){
2       myString = myPort.readStringUntil(13);}
```

6.3. Generación de imagen en Processing

Una vez que se han recibido los datos de las temperaturas sensadas por la cámara, se post-procesan los datos para generar una imagen. El primer paso en este proceso, es encontrar la temperatura más alta y la temperatura más baja de la captura específica para generar una nueva escala para la elección de las imágenes. El objetivo de este paso es evitar que si hay sólo temperaturas muy bajas o sólo temperaturas muy altas en el momento de la captura, no se muestre ninguna imagen o se muestren demasiadas imágenes. Este paso se puede ver como un filtro y se ve así dentro del código:

```

1 // For each floating point value, double check that we've acquired a number,
2 // then determine the min and max temperature values for this frame
3 for(int q = 0; q < 768; q++){
4     if(!Float.isNaN(float(splitString[q])) && float(splitString[q]) > maxTemp){
5         maxTemp = float(splitString[q]);
6     }else if (!Float.isNaN(float(splitString[q])) && float(splitString[q]) <
7     minTemp){
8         minTemp = float(splitString[q]);
9     }
10 }
11 // for each of the 768 values, map the temperatures between min and max
12 // of the space
13 for(int q = 0; q < 768; q++){
14     if(!Float.isNaN(float(splitString[q]))){
15         temps[q] = constrain(map(float(splitString[q]), minTemp, maxTemp, 180,
16         360), 160, 360);}
17     else{
18         temps[q] = 0;
19     }
}

```

Este filtro le da un valor a las temperaturas entre 180 y 360 para obtener un rango amplio de opciones. Una vez que se obtienen los nuevos valores, es ahí donde empieza la muestra de imágenes.

Como el objetivo principal del proyecto es que se detecten personas para que estas se puedan ver en la imagen reflejada e interactuar con la imagen, se toman en cuenta los valores en los que las temperaturas

corresponden a las de una persona. Después de varias pruebas, se determinó que los valores entre 280 y 360 incluyen a las personas aproximadamente. Como se quiere que se tomen en cuenta diferentes partes y se pueda ver un poco mejor la silueta, se divide ese rango de valores en cuatro grupos para que cada uno de ellos tenga una imagen con una animación diferente.

Como demostración para este proyecto se escogió la temática del juego Dungeons and Dragons, ya que es de gran interés en mi grupo social. La imagen escogida de fondo, fue una imagen de creación propia en una campaña del juego. Esta imagen corresponde a una escena del juego donde todos los jugadores se encontraban acampando y descansando antes de una pelea con uno de los más grandes adversarios. Para hacer la noche más amena, cada uno escogió una de sus habilidades para alegrar al grupo, entre ellas cantar o crear luces bailantes. La imagen se muestra a continuación:



Figura 7: Fondo de la pantalla interactiva. Figura propia.

Las luces bailantes es un hechizo en el que la criatura que lo genera puede crear una silueta humana a través de ellas. Debido a que se genera una silueta a través de la cámara termal, se tomó este como referencia. Como se mencionaba anteriormente existen cuatro grupos de valores, entonces a cada grupo de valores se le relaciona una animación con una imagen diferente. El valor más alto, es decir, el punto más caliente, tendrá luz morada ya que es la luz de más intensidad. Pasando por azul y verde, el punto menos caliente tendrá la luz amarilla.



Figura 8: Luces utilizadas como objetos interactivos dentro de la pantalla. Figura propia.

Adicionalmente, cada luz tendrá una animación diferente. La luz morada se hace cada vez más grande

si ese pixel mantiene la temperatura mayor a 340. La luz azul se mantiene de un tamaño más pequeño y si el punto ya no tiene esa temperatura desaparece. La luz verde y la amarilla tienen un efecto de 'dejar rastro' así que si ya no existe la temperatura mayor al rango correspondiente, la figura se hace cada vez más pequeña hasta desaparecer.

Para lograr estos efectos, se utilizaron arreglos que funcionan como 'memoria' de los píxeles. Este arreglo guarda el estado de la captura anterior para tomar la decisión de qué imagen mostrar en la pantalla siguiente. Un ejemplo del código de la luz amarilla se puede ver a continuación:

```
1 if(temp[ i ]>260 && temp[ i ]<280){  
2     forth_size=40;  
3     image(forth,x,y,forth_size,forth_size);  
4     still_forth[ i ] = 30;  
5 }else{  
6     if(still_forth[ i ]>0){  
7         forth_size = still_forth[ i ]/2;  
8         image(forth,x,y,forth_size,forth_size);  
9         still_forth[ i ]--;  
10    }  
11 }
```

Con esto se concluye el proceso para la creación de la imagen digital a partir de la cámara termal.

6.4. Resultado

Se puede ver una demostración del comportamiento de la experiencia immersiva en <https://youtu.be/6yyOsIjM5S8>. Cabe destacar que el video no es parte de la experiencia, pero se incluye ahí para ver el correcto funcionamiento. En el siguiente gif, se puede ver como la imagen se encuentra variando con respecto a la imagen de la persona real:

7. Conclusiones y recomendaciones

- El microcontrolador ESP8266 es ideal para este proyecto porque es de bajo costo, tiene suficiente procesamiento para la cámara, es pequeño, por lo que se puede esconder en un museo y funciona a bajo poder. Además, gracias a su módulo de Wifi la conexión se podría hacer remota si fuera necesario.
- La cámara termal MLX90640 es ideal para este proyecto porque tiene suficientes píxeles para tener una silueta nítida de una persona. Además es pequeño como el microcontrolador y funciona a bajo poder.
- La compatibilidad de la cámara y el microcontrolador es bastante alta ya que tiene el procesamiento como el poder requerido para funcionar y ambos se pueden programar en Arduino. Adicionalmente ambos son pequeños, permitiendo tener un producto portátil y fácil de disimular.
- La comunidad existente alrededor de este microcontrolador aporta una gran cantidad de documentación y ejemplos para que su uso sea más accesible. Existen bibliotecas de código abierto que se adaptan tanto al microcontrolador como al sensor.
- El software Processing permite crear animaciones a través de código, tomando en cuenta la entrada de las temperaturas sensadas por el sensor. Permite cambiar de color, tamaño, posición, entre otros, lo que lo hace ideal para esta implementación que se basa en proyectar obras de arte.
- El costo final del proyecto es bajo y la funcionalidad es accesible. Se pueden cambiar las imágenes y el fondo para tener otra obra de arte. Esta adaptabilidad permite que el proyecto sea replicable y utilizado en distintos museos con diferentes obras de arte.

Referencias

1. Beyond Van Gogh: The immersive experience Costa Rica. Recuperado de: <https://vangoghcr.com/>
2. Wiener, A. (2022) The Rise of “Immersive” Art Why are tech-centric, projection-based exhibits suddenly everywhere? Recuperado de: <https://www.newyorker.com/news/letter-from-silicon-valley/the-rise-and-rise-of-immersive-art>
3. Adafruit. (2022) Adafruit Feather HUZZAH with ESP8266. Recuperado de: <https://www.adafruit.com/product/2821>
4. ESP8266 Community Wiki. (2015) ESP8266EX Datasheet. Recuperado de: https://www.esp8266.com/wiki/lib/exe/fetch.php?media=0a-esp8266_datasheet_en_v4.3.pdf
5. Kembor, K. (2020) Adafruit MLX90640 IR Thermal Camera. Recuperado de: <https://learn.adafruit.com/adafruit-mlx90640-ir-thermal-camera>
6. Treece, T. (2016) Adafruit IO Basics: ESP8266 + Arduino. Recuperado de: <https://learn.adafruit.com/adafruit-io-basics-esp8266-arduino>
7. Melexis (2022) MLX90640 32x24 IR array. Recuperado de: <https://github.com/melexis/mlx90640-library/blob/master/MLX90640%20driver.pdf>
8. Processing (2022) Serial. Recuperado de: <https://processing.org/reference/libraries/serial/index.html>

8. Apéndice



ESP8266EX Datasheet

Version 4.3

Espressif Systems IOT Team

<http://bbs.espressif.com/>

Copyright © 2015



1. General Overview

1.1. Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

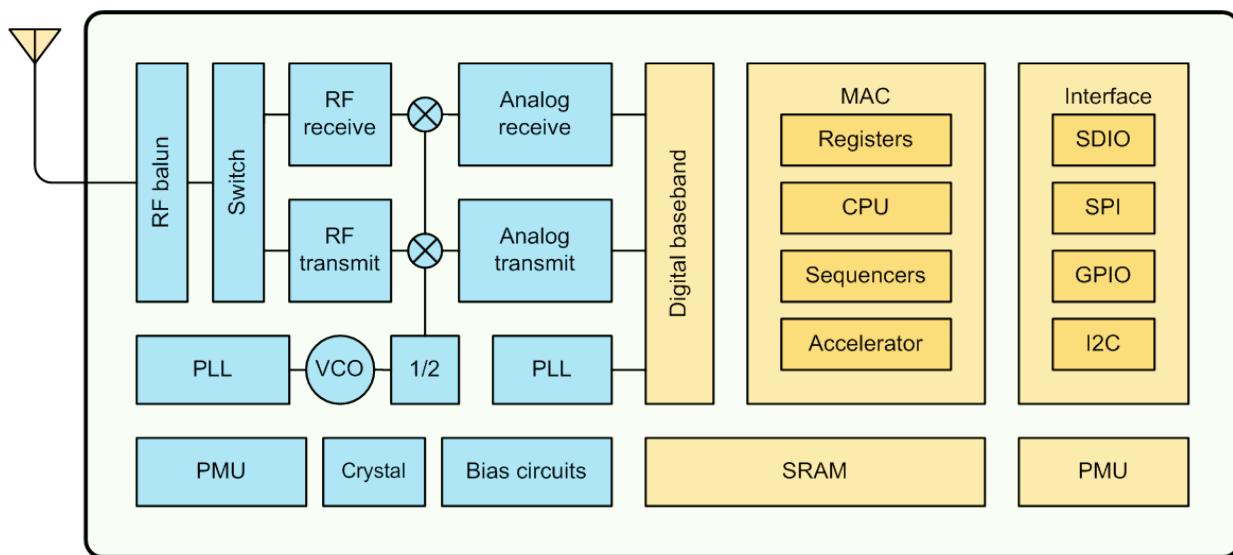


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).



Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

1.2. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

1.3. Parameters

Table 1 Parameters



Categories	Items	Values
WiFi Paramters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocles	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware Parameters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
Software Parameters	External Interface	N/A
	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Ssoftware Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP



	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App
--	--------------------	---

1.4. Ultra Low Power Technology

ESP8266EX has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode.

By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, ESP8266EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point.

When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266EX at any required interval.

The ESP8266EX can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature of the ESP8266EX can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed.

In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption.

1.5. Major Applications

Major fields of ESP8266EX applications to Internet-of-Things include:

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics



- WiFi Location-aware Devices
- Security ID Tags
- WiFi Position System Beacons



2. Hardware Overview

2.1. Pin Definitions

The pin assignments for 32-pin QFN package is illustrated in Fig.2.

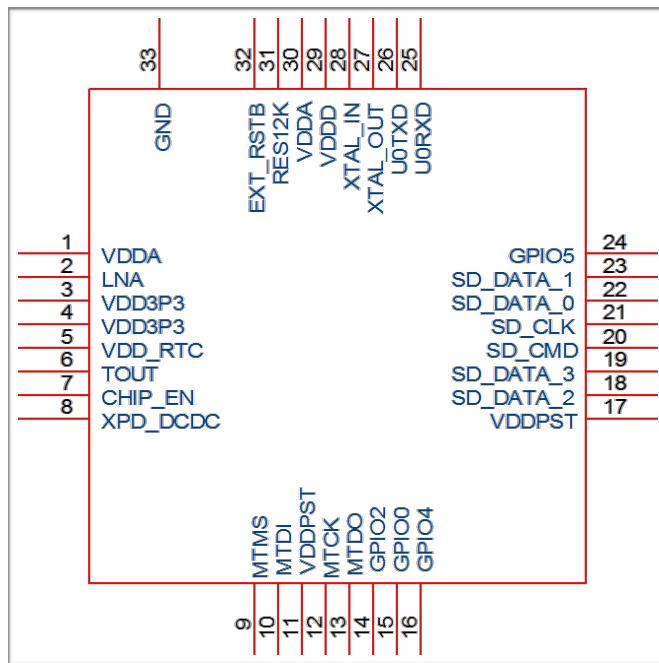


Figure 2 Pin Assignments

Table 2 below presents an overview on the general pin attributes and the functions of each pin.

Table 2 Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC (1.1V)



6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup; GPIO16
9	MTMS	I/O	GPIO14; HSPI_CLK
10	MTDI	I/O	GPIO12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MSIO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V~3.6V
30	VDDA	P	Analog Power 3.0V~3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)



Note: GPIO2, GPIO0, MTDO can be configurable as 3-bit SDIO mode.

2.2. Electrical Characteristics

Table 3 ESP8266EX Electrical Characteristics

Parameters		Conditions	Min	Typical	Max	Unit
Storage Temperature Range			-40	Normal	125	°C
Maximum Soldering Temperature		IPC/JEDEC J-STD-020			260	°C
Working Voltage Value			3.0	3.3	3.6	V
I/O	V_{IL}/V_{IH}		-0.3/0.75 V_{IO}		0.25 $V_{IO}/3.6$	V
	V_{OL}/V_{OH}		N/0.8 V_{IO}		0.1 V_{IO}/N	
	I_{MAX}				12	mA
Electrostatic Discharge (HBM)		TAMB=25°C			2	kV
Electrostatic Discharge (CDM)		TAMB=25°C			0.5	kV

2.3. Power Consumption

The following current consumption is based on 3.3V supply, and 25°C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.

Table 4 Description on Power Consumption

Parameters	Min	Typical	Max	Unit
Tx802.11b, CCK 11Mbps, P OUT=+17dBm		170		mA
Tx 802.11g, OFDM 54Mbps, P OUT =+15dBm		140		mA
Tx 802.11n, MCS7, P OUT =+13dBm		120		mA
Rx 802.11b, 1024 bytes packet length , -80dBm		50		mA
Rx 802.11g, 1024 bytes packet length, -70dBm		56		mA
Rx 802.11n, 1024 bytes packet length, -65dBm		56		mA
Modem-Sleep①		15		mA
Light-Sleep②		0.9		mA
Deep-Sleep③		10		uA
Power Off		0.5		uA