



Universidad Veracruzana

Universidad Veracruzana

Experiencia Educativa:

Sistemas embebidos



FACULTAD
DE
INSTRUMENTACIÓN
ELECTRÓNICA

Reconocimiento facial con ESP 32

Profesor:

Hernández Machuca Sergio Francisco

Alumno:

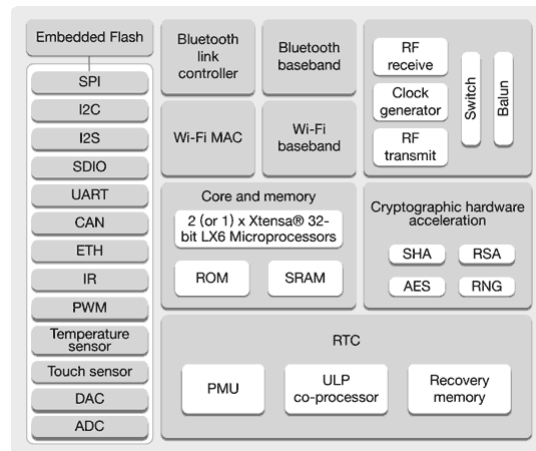
Abad Dolores Lázaro

Rodríguez Hernández Erick Abimael

03 enero de 2021

¿QUÉ ES EL ESP32?

El ESP32 es un SoC (System on Chip) diseñado por la compañía china Espressif y fabricado por TSMC. Integra en un único chip un procesador Tensilica Xtensa de doble núcleo de 32bits a 160Mhz (con posibilidad de hasta 240Mhz), conectividad WiFi y Bluetooth.



El módulo ESP32

El módulo ESP32 es una solución de Wi-Fi/Bluetooth todo en uno, integrada y certificada que proporciona no solo la radio inalámbrica, sino también un procesador integrado con interfaces para conectarse con varios periféricos. El procesador en realidad tiene dos núcleos de procesamiento cuyas frecuencias operativas pueden controlarse independientemente entre 80 megahercios (MHz) y 240 MHz. Los periféricos del procesador facilitan la conexión a una variedad de interfaces externas como:

- Interfaz periférica serial (SPI)
- I²C
- Transmisor receptor asíncrono universal (UART)
- I²S
- Ethernet
- Tarjetas SD
- Interfaces táctiles y capacitivas



Además, los chips ESP32 más nuevos son compatibles con funciones de seguridad como arranque seguro y encriptación flash necesarios para implementar dispositivos inteligentes de Internet of Things (IoT) reales y listos para la producción, y curiosamente, ahora también tienen soporte para aprendizaje automático con el framework ESP-OMS.

¿Qué es el reconocimiento facial?

El reconocimiento facial es una forma de autenticación biométrica que utiliza medidas corporales para verificar tu identidad. El reconocimiento facial es un subconjunto de datos biométricos que identifica a las personas mediante la medición de la forma y estructura únicas de sus rostros. Los diferentes sistemas existentes utilizan técnicas distintas, pero en lo fundamental, el reconocimiento facial utiliza los mismos principios que otras técnicas de autenticación biométrica, como los escáneres de huellas digitales y el reconocimiento de voz.



¿Cómo funciona el reconocimiento facial?

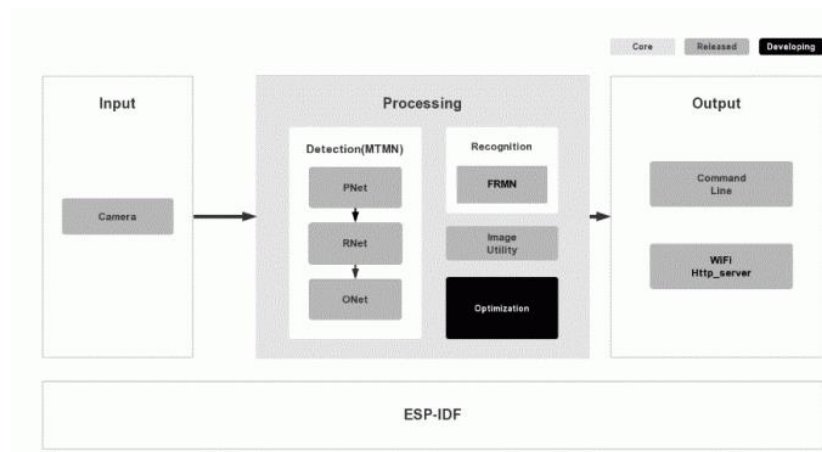
Todos los sistemas de reconocimiento facial capturan una imagen bidimensional o tridimensional de la cara de una persona, y luego comparan la información clave de esa imagen con una base de datos de imágenes conocidas. En el caso de las fuerzas policiales, esa base de datos puede ser recopilada de fotografías de identificación policial. En el caso de las cámaras para hogares inteligentes, es probable que los datos provengan de imágenes de personas que has identificado como familiares o amigos a través de la aplicación de la cámara.

Reconocimiento facial con ESP32

Esp32 cuenta con cualidades de gran versatilidad, entre ellas esta la capacidad de implementar sistemas con reconocimiento facial apoyándonos de librerías y placas de desarrollo dotadas del hardware necesario.

Frameworks

el framework ESP-WHO es un framework de referencia para la implementación del reconocimiento facial en la ESP. está construido sobre el framework de desarrollo de IoT (ESP-IDF), un framework de desarrollo oficial para el chip de Espressif, que está destinado a la detección y reconocimiento facial.



ESP-WHO es pues la utilidad de detección, reconocimiento e imagen son el núcleo de la plataforma constando en realidad de los siguientes módulos:

- **Image Utility** ofrece API de procesamiento de imágenes fundamentales.
- **La detección** toma imágenes como entrada y proporciona la posición de la cara si hay una cara. Se implementa con el modelo MTMN, que se refiere a MTCNN y MobileNets .
- **El reconocimiento** es identificar a la persona en particular y necesita los resultados de la detección. Se implementa con el modelo MobileFace.
- **La optimización** consiste principalmente en aumentar la precisión de la inferencia y acelerar todo el proceso. Pero también podría cambiar la estructura de la red, actualizar los coeficientes, refactorizar el código, etc.

Tanto la entrada como la salida son flexibles porque las fuentes de imagen pueden **ingresarse** a través de la cámara (aunque sin embargo, no proporcionan muchos controladores en este momento, ya que los de otros módulos de cámara se lanzarán en el futuro) y porque los resultados se pueden **mostrar** a través de la línea de comando, un LCD o incluso el sitio web a través del servicio de Wi-Fi http.

Para saber cómo configurar el módulo ESP32 para sus aplicaciones, puede consultar el archivo README.md de cada ejemplo del repositorio de github (<https://github.com/esp8266/esp-who>)

Preparación de software

La resolución recomendada de la imagen de entrada es QVGA (320×240) . En cuanto a elegir la cámara si no usa el citado modulo recomendado [ESP-EYE](#) el módulo ESP32 usado deberá tener libre los pines específicos según la cámara que use Por ahora el sw proporcionado soporta el controlador de OV2640 y OV3660 , que son muy recomendables para comenzar.

ESP-IDF

Consulte las guías de configuración para obtener instrucciones detalladas para configurar el ESP-IDF:[Guía de inicio para la versión estable de ESP-IDF](#)

En este caso, tomamos ESP-IDF v3.2 como la versión de prueba. Si ya ha configurado ESP-IDF antes y no desea cambiar el existente, puede señalar IDF_PATH al ESP-IDF que se encuentra en ESP-WHO.

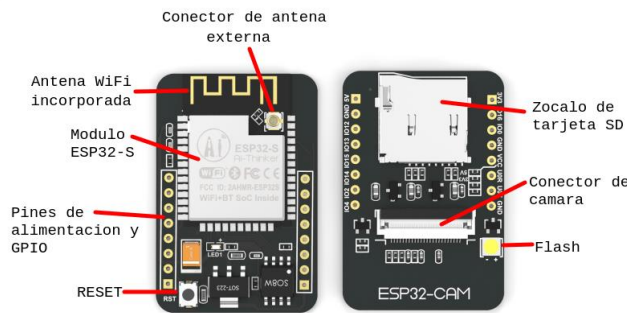
Otros componentes sw

Otros componentes sw necesarios lo constituye el marco principal del SDK, con algunos controladores y algunos algoritmos dentro.

- Cámara : El componente de cámara contiene controladores para dispositivos de cámara de ESP32.
- Face de esp: El componente esp-face contiene las API de las redes neuronales ESP-WHO, incluido el marco de detección y reconocimiento de rostros.

ESP32-CAM

El ESP32-CAM es un módulo fabricado por la empresa Ai Thinker (aunque existen otros similares) basado en un SoC ESP32-S de Espressif con 520 KB de memoria interna y 4 MB externos. Cuenta con una antena Wifi incorporada y un conector tipo IPEX para conectar una antena externa y lograr así un mayor alcance. También incluye un LED para emplear como flash, un zócalo para alojar una tarjeta SD de hasta 4 GB, un conector para una cámara como la OV2640 o la OV7670 y 16 pines con señales de alimentación y entradas/salidas (GPIO)



Aspecto del módulo ESP32-CAM (Fuente: Ai Thinker)

Características

- Alimentación: 5V
- SoC: ESP-32S
- Memoria flash: 32 Mbit
- RAM: 520 KB interna + 4 MB externa (PSRAM – Pseudostatic RAM)
- Antena: PCB incorporada + conector IPEX
- Protocolos WiFi: IEEE 802.11 b/g/n/e/i
- Bluetooth: Bluetooth 4.2 BR/EDR y BLE
- Modos WIFI: Station / SoftAP / SoftAP+Station
- Seguridad: WPA/WPA2/WPA2-Enterprise/WPS
- Formatos de imagen: JPEG (OV2640), BMP, escala de grises
- Tarjeta SD: 4GB máx
- Interface: UART/SPI/I2C/PWM
- GPIO: 9
- Velocidad UART: 115200 bps (por omisión)
- Dimensiones: 40.5mm x 27mm x 4.5mm

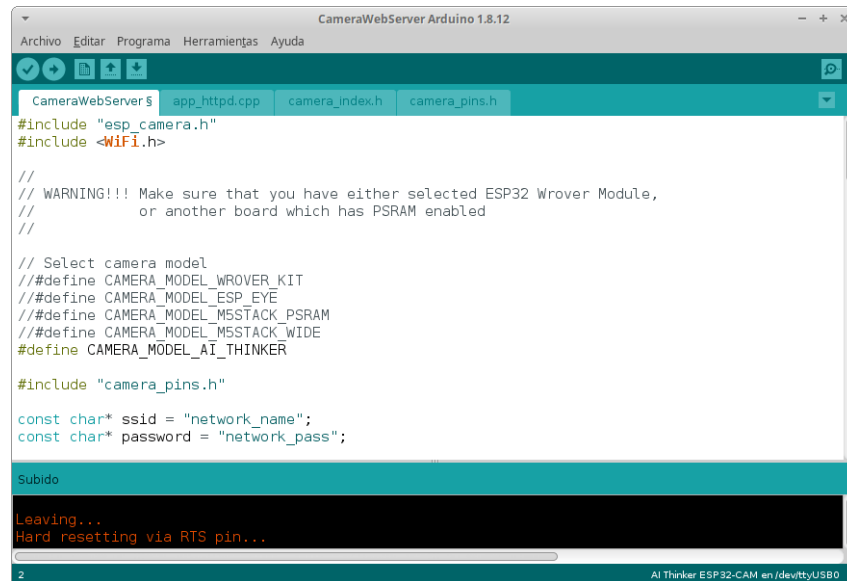
Programación

Al contener un ESP-32S, este módulo puede ser programado empleando distintos lenguajes, como C/C++, Python o hasta BASIC, usando distintos entornos de desarrollo, como el ESP-IDF de Espressif o el conocido IDE de Arduino. En esta revisión usaremos este último, con un programa que se incluye de ejemplo. Si no tienes configurado el IDE de Arduino para programar el ESP-32, puedes encontrar todas las instrucciones [en este otro artículo](#).

El ESP32-CAM lamentablemente no tiene una interfaz USB que permita su conexión directa a una PC, como es el caso de otros módulos ESP32 o ESP8266. Para programarlo deberemos utilizar un adaptador FTDI entre los niveles de 5V del puerto USB y los 3V de los pines de programación del ESP32. La siguiente imagen muestra los pines del ESP32-CAM con su función.

El último paso es indicarle al programa específicamente que módulo estamos usando (puede utilizarse con distintos modelos). Comentamos los no usados y descomentamos el apropiado (AI Thinker en mi caso)

También debemos cargar las credenciales de la red Wifi, esto es, el nombre de la red (SSID) y su contraseña, para que el módulo se pueda conectar.



```
CameraWebServer Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda

CameraWebServer app_httpd.cpp camera_index.h camera_pins.h

#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
// or another board which has PSRAM enabled
//

// Select camera model
// #define CAMERA_MODEL_WROVER_KIT
// #define CAMERA_MODEL_ESP_EYE
// #define CAMERA_MODEL_M5STACK_PSRAM
// #define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

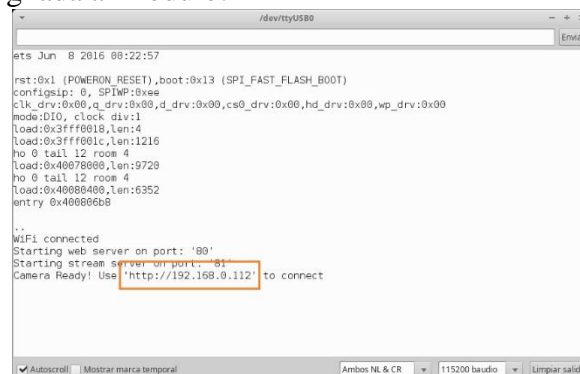
const char* ssid = "network_name";
const char* password = "network_pass";

Subido

Leaving...
Hard resetting via RTS pin...

2 AI Thinker ESP32-CAM en /dev/ttyUSB0
```

Cuando todo está completo, podemos subir el programa al módulo. Una vez terminado sin errores, debemos quitar el cable que une GPIO0 con GND, resetear el módulo y abrir el monitor serie del IDE. En el monitor el programa nos muestra el estado de funcionamiento y debemos buscar la IP asignada al módulo:



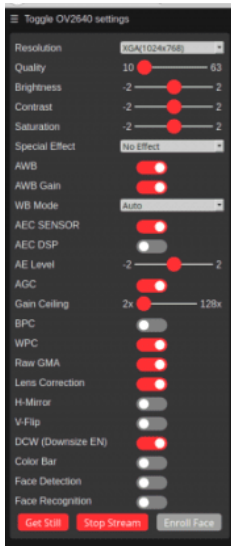
```
/dev/ttyUSB0
ets Jun  8 2016 08:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9728
ho 0 tail 12 room 4
load:0x40086400,len:6352
entry 0x400806b8

..
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.0.112' to connect

Autoscroll Mostrar marca temporal Ambos NL & CR 115200 baudio Limpiar salida
```

Si abrimos esa dirección desde un navegador en una PC o un teléfono conectados a la misma red, podremos ver el streaming de video del módulo.



En el lateral izquierdo contamos con un panel de configuración y control de las cualidades de la cámara incluyendo el reconocimiento facial. Entre las cosas que podemos configurar tenemos:

- Resolución
- Contraste
- Detección de rostros
- Reconocimiento facial
- Indexación de rostros adicionales

Activando el reconocimiento facial el sistema nos mandara una alerta como saludo de rostros reconocidos o de rostros intrusos



Fuentes

<https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/>

<https://www.naylampmechatronics.com/img/cms/Datasheets/ESP32-CAM%20Product%20Specification.pdf>

<https://robotzero.one/esp32-face-door-entry/>