

PROYECTO: CAMARA CON SERIVOR WEB

Para la elección de este proyecto opté por programar e implementar una cámara web en el ESP32, queriendo así comprobar a que nivel se podía llegar con un microprocesador tan simple y hasta que punto es funcional este mismo.

Materiales

ESP32-CAM: Para la realización de este proyecto he tenido que cambiar el microprocesador por otro idéntico, pero con la cámara incorporada, ya que el ESP32 con el que hemos trabajado en las practicas no tenía esa función. El chip en si es el ESP32-CAM.

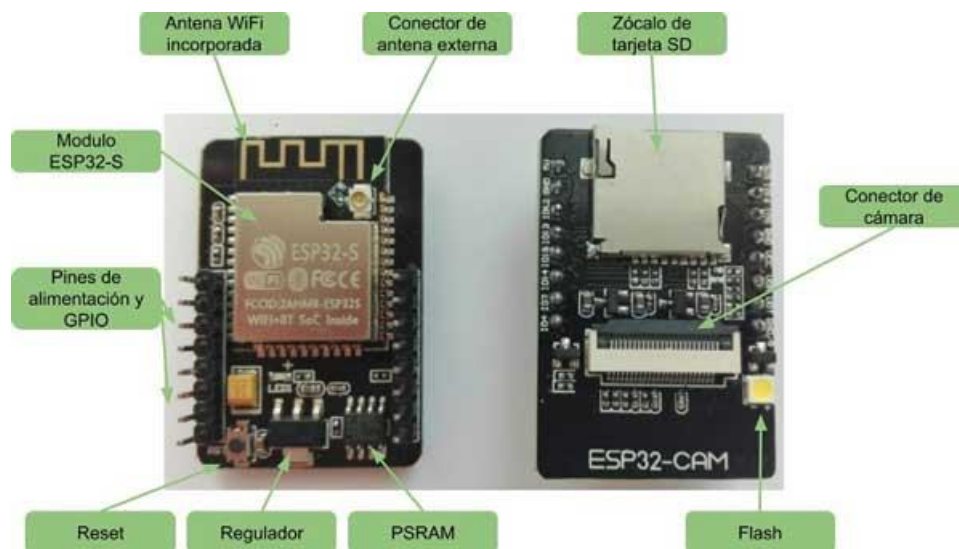


Imagen 1: ESP32-CAM

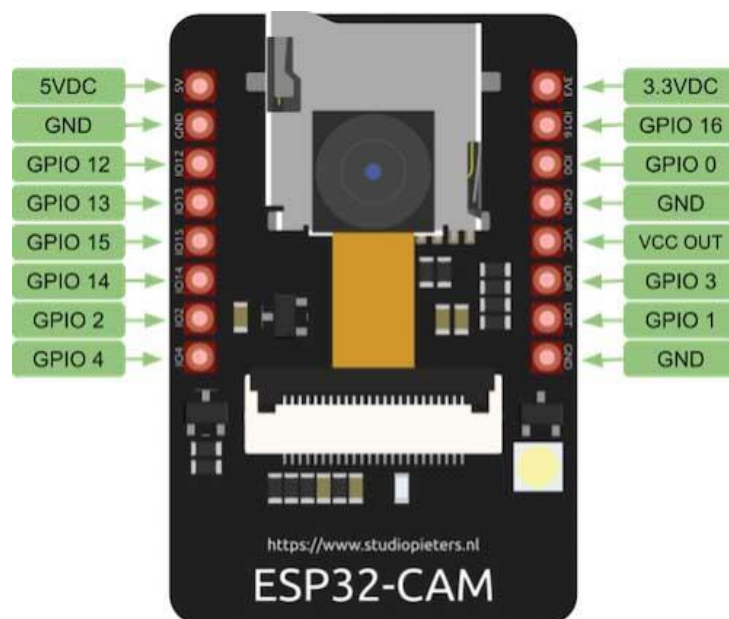


Imagen 2: Diagrama ESP32-CAM

Convertidor: Después de esto, tuve que mirar cómo subir el código al microprocesador, ya que el ESP32-CAM no cuenta con conexión Micro-USB a USB para conectarlo al ordenador directamente. Para esto escogí un convertidor USB a TTL, el cual mediante de cables hembra-hembra se conecta con los pines del microprocesador.

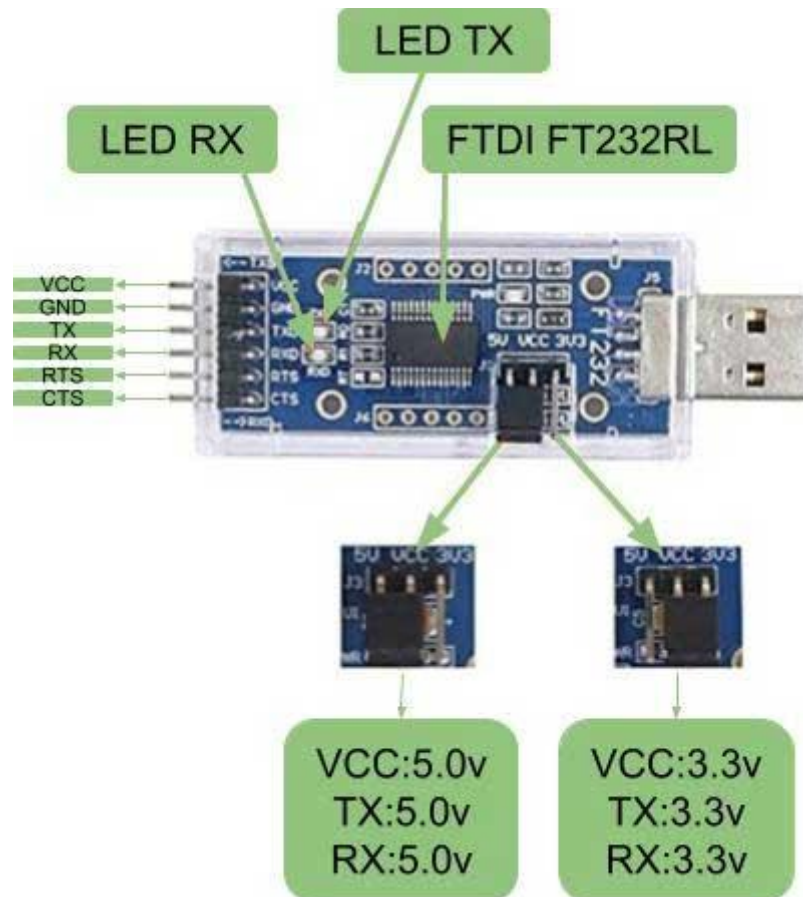


Imagen 3: Convertidor USB a TTL

Disipador: al acabar de montar todo, subir el código y comprobar que todo funciona correctamente, surgió el problema del sobrecalentamiento de la parte trasera del chip, lo cual podía suponer un problema a largo plazo debido a que podía dejar de funcionar. Para ello le instalé un pequeño disipador en la parte posterior para que la temperatura disminuya.



Imagen 4: Disipador

Montaje

El montaje, principalmente consiste en conectar el convertidor con el microprocesador siguiendo el esquema que hay en la siguiente imagen y posteriormente conectarlo al ordenador.

Por otro lado, el disipador solamente se le ha de retirar una banda para destapar la cara donde se encuentra el pegamento y pegarlo en el sitio donde se produce el sobrecalentamiento.

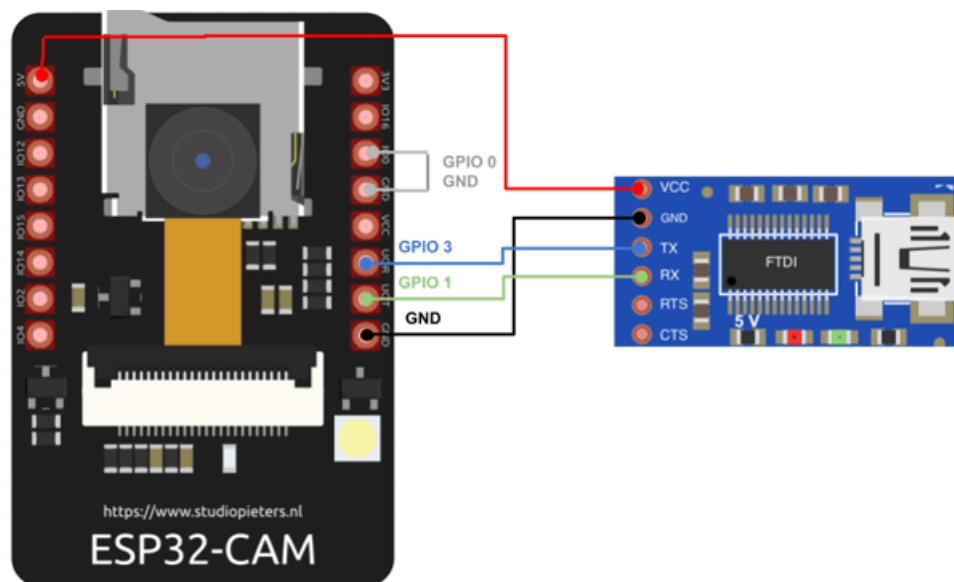
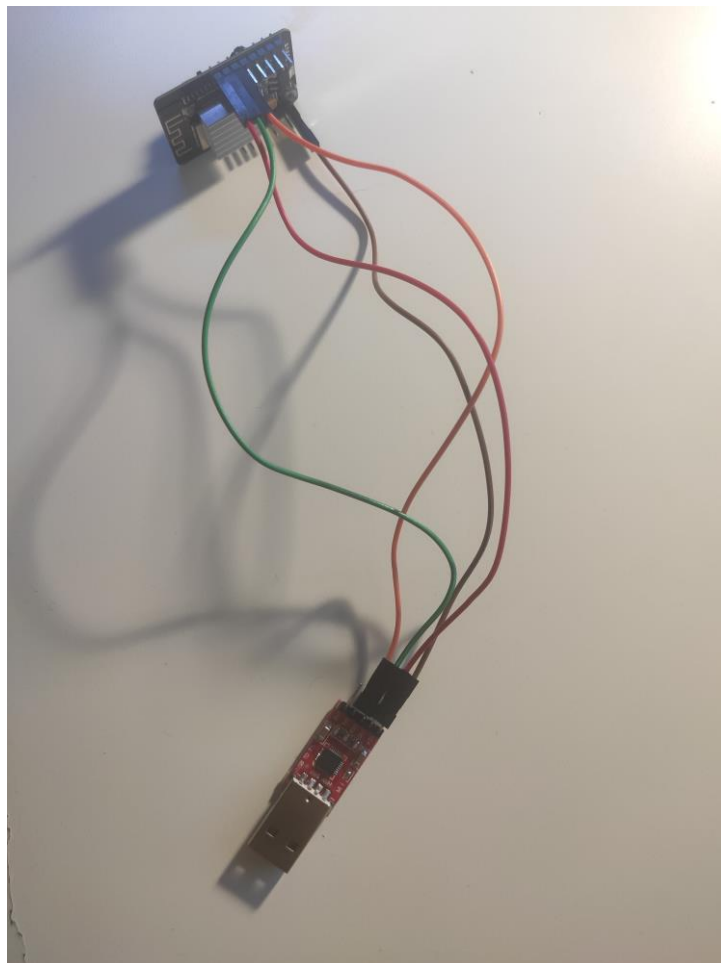


Imagen 5: Esquema de montaje

Foto del montaje:



Código

Para programar el ESP32-CAM he tenido que utilizar el entorno de Arduino, ya que el Visual Studio Code no es del todo compatible y tiene una serie de problemas que hace que sea mucho más complicado escribir el código.

Utilizando un programa de prueba de Arduino y adaptándolo a las necesidades del proyecto, he podido programar el chip para poder crear un servidor web donde se pueda ver la imagen que capte la cámara.

Antes de todo, se tienen que instalar las librerías de ESP32 en el programa de Arduino para poder configurar correctamente el espacio de trabajo. Seguidamente, en el apartado de “Herramientas” se selecciona la placa que vamos a utilizar, la “AI Thinker ESP32-CAM” así como el puerto en el cual vamos a conectar el convertidor a nuestro PC.

Una vez este configurado el programa, en el código generado de prueba, lo primero es definir el tipo de cámara que va a utilizar nuestro chip. En este caso se comenta la línea correspondiente a la cámara del AI Thinker correspondiente al ESP32-CAM y se dejan comentadas todas las demás.

```
10 // Select camera model
11 //#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
12 //#define CAMERA_MODEL_ESP_EYE // Has PSRAM
13 //#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
14 //#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
15 //#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
16 //#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
17 #define CAMERA_MODEL_AI_THINKER // Has PSRAM
18 //#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
```

Seguidamente, introducimos nuestros parámetros tanto de la red a la que se va a conectar el chip cómo la contraseña de ésta.

```
22 const char* ssid = "MiCamara";
23 const char* password = "accesocamara";
24
```

Nota: para poder visualizar el servidor web en el navegador hace falta que el dispositivo desde el cual se quiere visualizar esté conectado a la misma red que el ESP32-CAM.

Después de hacer lo mencionado anteriormente, se tiene que subir el código al chip para poder visualizar el servidor web. Para esto, el GPIO 0 y el GND tienen que estar puenteados entre sí para que el ESP32-CAM se ponga en modo programación.

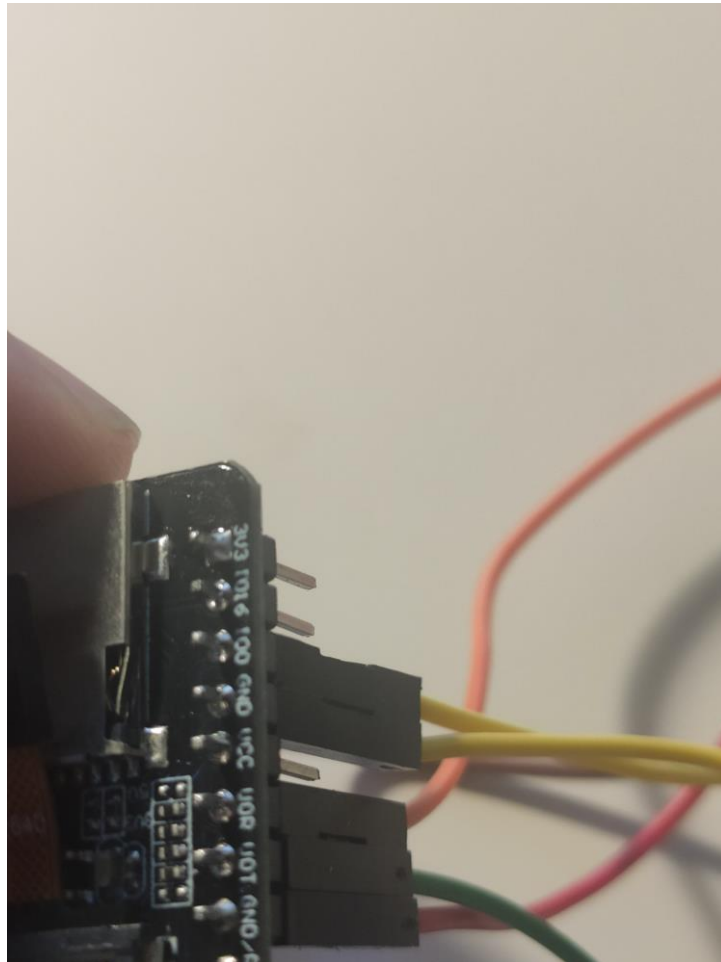


Imagen 6: Puente GPIO 0 y GND

Una vez se haya subido el código correctamente, el puente para que se haya podido subir el código se desconecta y se abre el serial monitor del Arduino situado en la esquina superior derecha, con un icono de una lupa.

Una vez abierto, hacemos un reset de nuestro ESP32-CAM y esperamos hasta que nos confirme que se ha conectado correctamente a la red. Cuando esto ocurra, nos proporcionará el ID para que podamos visualizar el servidor web.



Imagen 7: Serial Monitor

Nota: para poder visualizar el contenido de la imagen anterior es importante seleccionar correctamente el “Monitor Speed” según lo que se haya puesto en el código. Se puede cambiar en el icono de la parte inferior derecha donde pone “115200 baudio”.

Servidor Web

Una vez hayamos obtenido la dirección IP del serial monitor al subir el código, solo tendríamos que buscar ese IP en nuestro navegador para poder visualizar la página web.

Una vez hecho, el resultado sería tal que así:

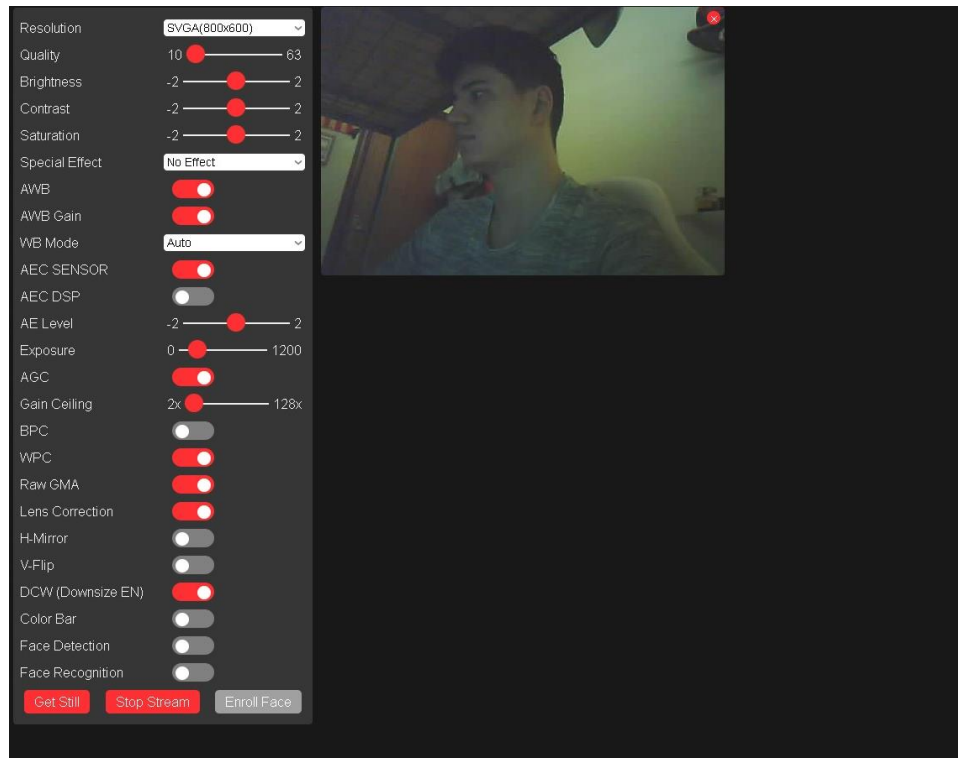


Imagen 8: Servidor Web

Por defecto, al entrar en la dirección IP no se mostrará ninguna imagen ya que aún no se ha empezado a hacer “Stream”. Para ello, solo hace falta seleccionar la opción de “Start Stream” y la cámara empezará a mostrar la imagen que esté captando. Se puede parar en cualquier momento apretando el mismo botón que ahora pone “Stop Stream”.

De entre las funcionalidades más comunes están el poder editar los niveles de calidad, brillo, contraste, saturación y algunos efectos para la cámara. También se puede cambiar la resolución de la pantalla desde un mínimo de 160x120 hasta un máximo de 1600x1200.