

Índice.

1.	Previo.	2
2.	Introducción.	2
3.	Material necesario.	3
3.1.	Red Wifi.	3
4.	Vagón.	3
4.1.	Cámara.	4
4.2.	Placa programación.	5
4.3.	Integrado L7805.	5
4.4.	Bornas.	6
4.5.	Baterías.	6
4.6.	Cables.	6
5.	Software	7
6.	Conexión de la placa ESP32 al programador.	7
6.1.	Con la placa de programación FTDI.	7
6.2.	Con la placa Arduino Uno.	8
7.	IDE Arduino, instalación.	8
7.1.	Descargar.	8
8.	Actualizar el programa.	9
8.1.	Carga.	9
8.2.	Actualización.	9
8.3.	Compilación.	10
8.4.	Subida, carga en la placa ESP32.	11
8.5.	Prueba.	11
9.	Montaje.	15
9.1.	Esquema de conexión.	15
9.2.	Integrado L7805	15
9.3.	Bornas.	15
9.4.	Conectar pilas 9 V.	15
9.5.	Alimentar la cámara.	17
9.6.	Montaje en el vagón.	17
10.	Funcionamiento del programa.	20

1. Previo.

El uso de lo aquí existente, se hace bajo la responsabilidad individual de cada uno.

Este montaje ha sido probado suficientemente como para subirlo al foro con garantía, y no me ha generado ningún tipo de problemas en su utilización.

El software está publicado en los ejemplos de Arduino.

2. Introducción.

He montado esta cámara en un vagón, para ver cómo ruedan de verdad las composiciones por el trazado de la maqueta, a simple vista, es difícil de apreciar los desniveles y los saltos en la vía, y con esto he conseguido mejorar muchas zonas en las que no se rodaba bien.

La cámara se ha colocado en un vagón en la parte delantera, -o trasera, según se mire-, a la que se le ha cargado un programa que permite visualizar el recorrido visualizándolo en un navegador a través de una dirección IP que se genera en el momento de la grabación del programa y su ejecución por primera vez en el entorno de programación de Arduino.

Para la grabación del programa se necesita el entorno de programación de Arduino para la obtención del ejecutable que luego se grabará en la placa donde está la cámara a través de un programador, que puede ser una placa de Arduino Uno, o una placa FTDI.

No se puede exportar el hexadecimal, pues hay que actualizarlo con el nombre de la Wifi y su contraseña, en cada copia.

La cámara funciona con 5V de corriente continua, y se alimenta con dos pilas en paralelo de 9V, no en serie, para conseguir los 5V estabilizados y el amperaje necesario para mantener el led de la placa encendido y proporcionar luz a la imagen.

Con 4 pilas de 1,5 V en serie, se obtiene 6V, pero no se si la tolerancia de la placa permite ese voltaje directamente, y si se regulan con el integrado, sale menos de 5V y no funciona, al menos en mi caso.

Los 5V se obtienen a través del integrado L7805 que regula el voltaje obtenido de las pilas de 9V a los 5V de forma estabilizada. No es necesario ningún elemento más, pues las pilas ya dan la corriente en continua, pero yo no tengo suficientes conocimientos de electrónica como para ser una afirmación inapelable, solo se que después de bastantes horas, funciona y no me ha dado problemas.

Conviene añadirle un disipador, pues se calienta bastante, funciona, pero mejor más fresquito.

Se puede obtener, al menos en las pruebas sí lo he obtenido, más de dos horas de autonomía con el led encendido.

Se puede captar la alimentación de la vía, con un puente rectificador antes del integrado, para cubrir el funcionamiento en DCC, o en analógico, funciona correctamente, pero el problema son los micro cortes en la alimentación que provocan reinicios del programa, no se si con un condensador se podría salvar el problema.

La cobertura de la cámara con la wifi es buena con el punto de acceso a más de 6 m. de distancia en una habitación distinta a la de la maqueta, en mi caso es de casi 8 m.

A la placa de la cámara se le puede incorporar una antena para mejorar la conexión a la wifi.

La calidad de la imagen es buena en una resolución de 640x480 píxel y en 800 x 600 píxel, con una buena velocidad de la locomotora no se perciben saltos en la imagen, la cámara dispone de más resolución, pero se pierde velocidad en la visualización de la imagen, dependerá supongo de la calidad de la señal wifi.

Yo he visualizado en 1280 x 720, y es bastante aceptable.

Es cuestión de hacer pruebas.

No realiza grabación, pero esta se puede obtener a través de la imagen que se reproduce en el navegador, con el programa OBS Studio, o similar.

El programa se ha obtenido de los ejemplos de Arduino.

Archivo→Ejemplos→Esp32→Camera→CameraWebServer

Los pasos son:

Descarga e instalación del software del IDE de Arduino.

Compilación del programa.

Grabación del programa en la placa.

Ejecución por primera vez en el IDE para saber la dirección IP.

Montaje de la cámara en el vagón.

La descarga está explicada en documento adjunto.

3. Material necesario.

3.1. Red Wifi.

EN mi caso no he necesitado nada especial, si vuestro router está muy lejos, a lo mejor es necesario un repetidor, todo depende de la calidad de la señal.

4. Vagón.

Vagón bordes de altura media, con buena estabilidad.



En uno de este tipo, en principio las pilas caben dentro del vagón, y la placa con la cámara también.

En uno de bordes bajos también pienso que puede servir, solo que habría que ajustar bien la placa para que no diera bandazos.

4.1. Cámara.

Cámara ESP32, chip ESP32-S, conviene que se compre una cámara con un cable de interface a la placa largo, en función de los cambios que cada uno desee realizar, pero el clásico corto no es práctico, en función del vagón a utilizar.

La cámara es referencia o modelo OV2640.



Esta versión de cámara es incómoda, no permite usar el led de iluminación bien en el vagón de borde alto, en uno bajo supongo que no habría pega, buscando la forma de sujetar bien la placa, la cámara podría ir al borde del vagón creo.

El borde es muy bajo y no molestaría al led ni a la cámara, pero no lo he probado.



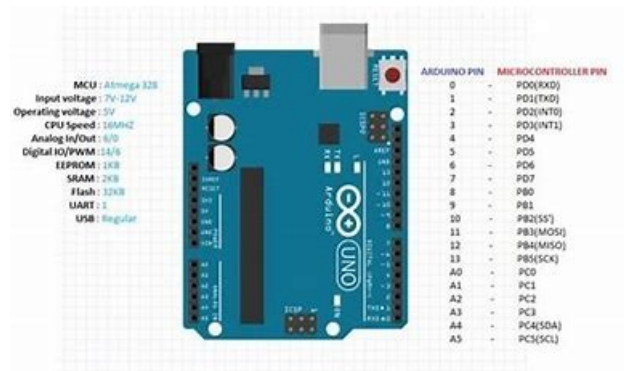
Esta es mejor, porque permite colocar la placa al contrario verticalmente y asomar el led por encima del borde del vagón.

Hay modelos que se venden con esta cámara.

En Internet es muy fácil de localizar, es cuestión de escribir en un buscador "ESP32 Cam" o "Cam ESP32".

4.2. Placa programación.

Placa Arduino Uno, o la FTDI, una de las dos.



Programador para grabar el programa en la cámara, o la placa Arduino UNO, una de las dos.



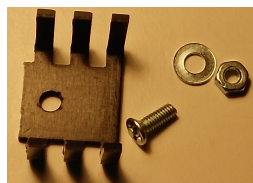
4.3. Integrado L7805.

Integrado L7805, regulador a 5V en corriente continua, y un disipador.

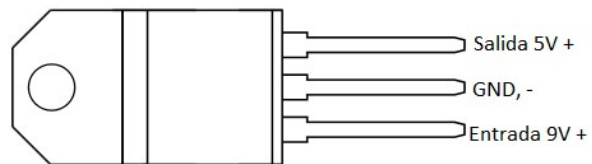


El disipador es de aluminio, no es de plástico, aunque lo parece.

Los tornillos pueden ser de 3 o 4 mm de diámetro, los de la foto son de tres, pero hay espacio suficiente para que sean de 4 mm.



El conexionado del integrado debe realizarse como se ve en el dibujo.



El dibujo se ha hecho mirando de frente el integrado.

4.4. Bornas.

Bornas de empalme, prescindibles.



4.5. Baterías.

Pilas 9V, normales o recargables.



4.6. Cables.

Cables de conexión para conectar la placa de la cámara al programador.



Macho hembra cuatro, y hembra hembra uno.
Placa de circuito para montaje.

5. Software

IDE Arduino.

<https://www.arduino.cc/en/software>

Programa CamaraWebServer

En el archivo comprimido.

Si se desea grabar la imagen en el ordenador, el programa OBS Studio va muy bien.

6. Conexión de la placa ESP32 al programador.

Seguir las indicaciones que se ven en la imagen.

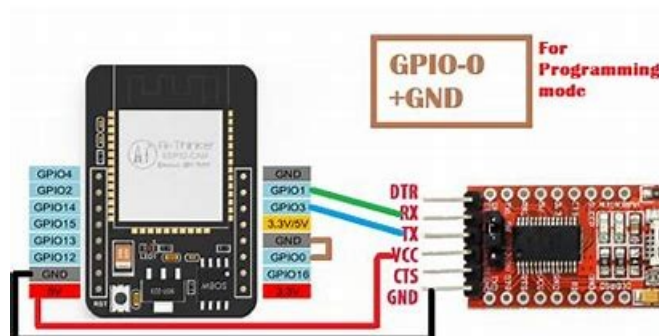
6.1. Con la placa de programación FTDI.

Solo una de las dos.

Esta placa pequeña permite conectar el programador por el puerto USB y subir el programa a la cámara.



El conexionado es como se ve en la imagen.



En modo programación hay que puentear los pins GPIO0 y GND

En la ejecución hay que quitar el puente.

No puedo añadir más, porque no lo he usado.

Pero en esta placa la transmisión va a la recepción y la recepción a la transmisión, que es lo lógico.

TX → RX

RX → TX

6.2. Con la placa Arduino Uno.

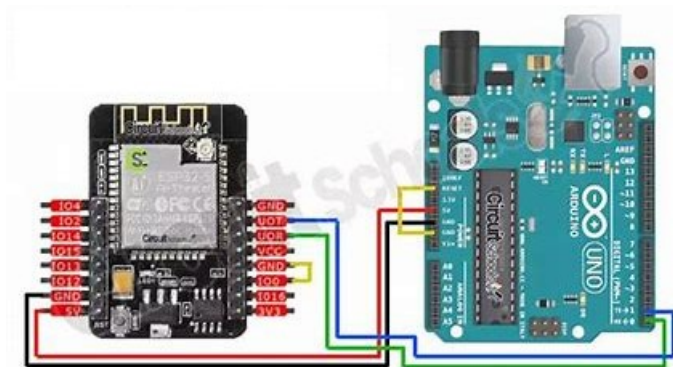
Solo una de las dos.

La conexión de la cámara a Arduino Uno es con cables macho hembra.

La conexión de los cables en Arduino es cómo se indica, no hay error, el cable de transmisión de Arduino va al pin de transmisión de la cámara y el de recepción al de recepción.

RX → RX

TX → TX



Durante la grabación del programa entre Arduino y la cámara han de estar puenteados el pin I00 y GND en la placa ESP32, con un cable hembra.

El puente amarillo en Arduino Uno no es necesario, en la placa ESP32 sí.

Esta es la que he usado yo, porque ya tenía un Arduino Uno, y no me hacía falta comprar el FTDI.

7. IDE Arduino, instalación.

7.1. Descargar.

El ejemplo es para Windows 10, con la versión de Arduino 2.3.0

Link a la web de Arduino.

<https://www.arduino.cc/en/software>

Pulsáis en Just Download, o se hace una donación, en las dos pantallas que siguen, y luego se inicia la descarga.

Se descarga la versión 2.3.0.

Los pasos están en un archivo adjunto, IDE Arduino.

8. Actualizar el programa

8.1. Carga.

Cargar el programa en el IDE de Arduino.

Para ello el directorio que figura en el archivo comprimido, la copiáis en el directorio donde deseáis tener el original del programa.

El directorio, carpeta, lo copiáis tal como viene.

Nombre	Fecha	Tipo	Tamaño
CameraWebServer	10/02/2024 19:21	Carpeta de archivos	
IDE Arduino.pdf	16/02/2024 17:22	Documento Adobe Acrobat	2 KB
TutorialCamara.pdf	16/02/2024 17:20	Documento Adobe Acrobat	5 KB
VolcadoEjecucionPr...	10/02/2024 19:24	Documento de texto	

Y dentro de CameraWebServer tendréis

Nombre	Fecha de modificación	Tipo
.skip.esp32c3	18/12/2023 12:04	Archivo ESP32C3
++ app_httpd.cpp	18/12/2023 12:04	C++ Source
camera_index.h	18/12/2023 12:04	C/C++ Header
camera_pins.h	18/12/2023 12:04	C/C++ Header
CameraWebServer.ino	10/02/2024 19:20	Archivo INO
partitions.csv	18/12/2023 12:04	Archivo de valores...

Y si hacéis doble clic en el archivo CameraWebServer.ino, se cargará automáticamente en el IDE de Arduino.

8.2. Actualización.

Modificar el programa para adecuarlo a vuestro entorno.

Primero definir el modelo de cámara adecuado, puede que sea la línea 24, para ello hay que dejar la línea que contiene el modelo Camera Model AI Thinker como se ve en la imagen, hay que quitarle las dos “//” que lleva delante.

También hay que buscar si en esa zona del programa, hay alguna otra línea, puede que sea la línea 18 la que esté sin esas dos “//”, solo debe haber un tipo de cámara definida, sin las “//”, las demás deben estar con “//”, para que no surtan efecto esas definiciones.

```
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

Lo siguiente es actualizar el contenido de las variables en las que se indica:

Nombre de vuestra wifi.

Contraseña para acceder a ella.

Buscar estas líneas, 38 y 39.

```
// =====  
// Enter your WiFi credentials  
// =====  
const char* ssid = "NombredemiWifi";  
const char* password = "ContraseñademiWifi";
```

Y actualizar su contenido al adecuado a vuestros datos.

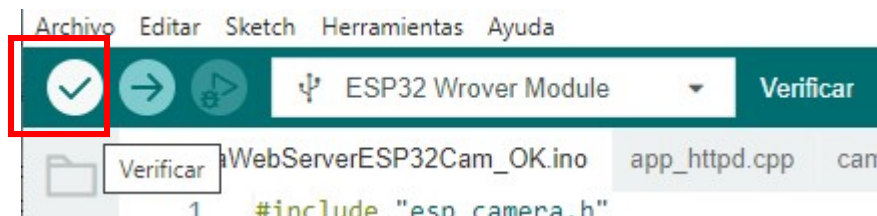
Debe acabar en “,” y estar entre comillas dobles los datos mencionados.

En programación no hay termino medio, o está bien o está mal, si se escribe mal dará un error, o en la compilación o en la ejecución.

Guardar el programa.

Compilar.

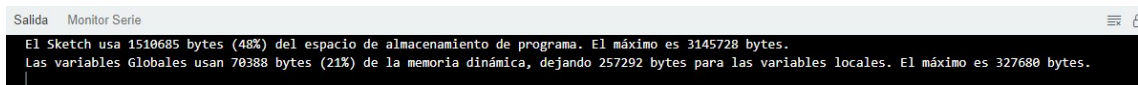
8.3. Compilación.



Arduino pone verificar.



El resultado del programa será algo parecido a esto:



El Sketch usa 1510685 bytes (48%) del espacio de almacenamiento de programa. El máximo es 3145728 bytes.
Las variables Globales usan 70388 bytes (21%) de la memoria dinámica, dejando 257292 bytes para las variables locales. El máximo es 327680 bytes.

8.4. Subida, carga en la placa ESP32.

Subir a la placa.

Para ello hay que tener conectada la placa de la cámara al programador.

```
Salida Monitor Serie
Compressed 3072 bytes to 137...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (137 compressed) at 0x00008000 in 0.1 seconds (effective 396.8 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 47...
Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 666.5 kbit/s)...
Hash of data verified.
Compressed 1516976 bytes to 990590...
Writing at 0x00010000... (1 %)
Writing at 0x00014323... (3 %)
Writing at 0x0001a947... (4 %)
Writing at 0x000266cb... (6 %)
Actualizando...
```

```
Salida Monitor Serie
Writing at 0x000b15d7... (43 %)
Writing at 0x000b6986... (47 %)
Writing at 0x000bbcd0... (49 %)
Writing at 0x000c0f68... (50 %)
Writing at 0x000c6427... (52 %)
Writing at 0x000cd1e8... (54 %)
Writing at 0x000d3d7e... (55 %)
Writing at 0x000db223... (57 %)
Writing at 0x000e1a1d... (59 %)
Writing at 0x000e72d7... (60 %)
Writing at 0x000ec6af... (62 %)
Writing at 0x000f1ac9... (63 %)
Writing at 0x000f7798... (65 %)
```

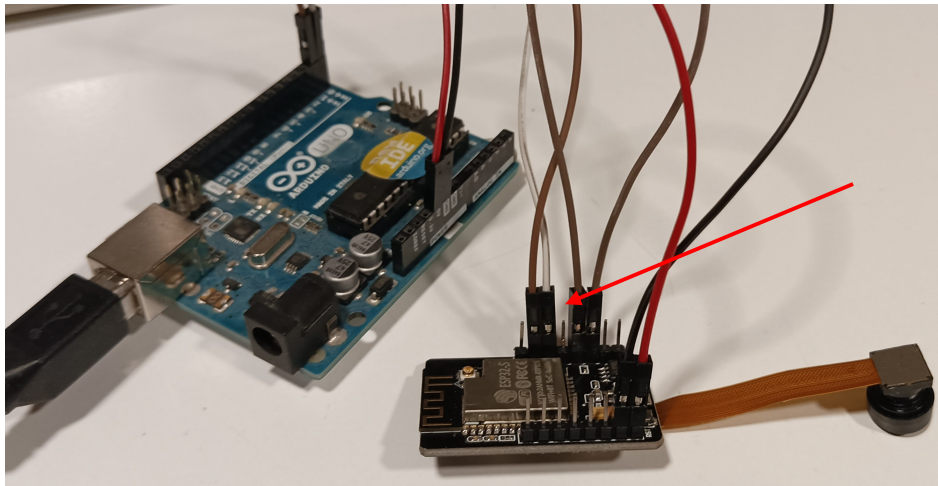
```
Salida Monitor Serie
Writing at 0x000f15d7... (68 %)
Writing at 0x0015ac95... (90 %)
Writing at 0x00162c76... (91 %)
Writing at 0x00167d50... (93 %)
Writing at 0x0016d71f... (95 %)
Writing at 0x001730b8... (96 %)
Writing at 0x0017a314... (98 %)
Writing at 0x0017fce0... (100 %)
Wrote 1516976 bytes (990590 compressed) at 0x00010000 in 85.8 seconds (effective 141.4 kbit/s)..
Hash of data verified.

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

8.5. Prueba.

Ahora probar.

La prueba se realiza con la cámara conectada al Arduino Uno, que suministra 5V sin problema a la cámara.

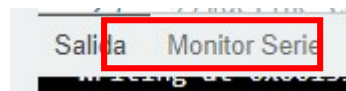


Se quita el puente que indica la flecha, es el que une IO0 con GND.

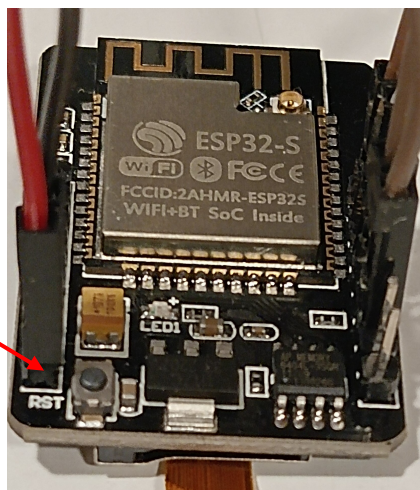
Pulsamos en la parte superior izquierda de la ventana, monitor serie.



O aquí.



Se activa la ventana del monitor serie de Arduino, se pulsa en el botón de reset.



Y se apunta la dirección IP de la cámara.

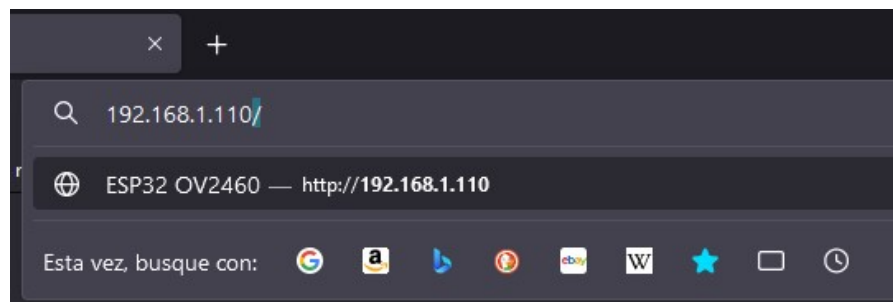
```
Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'ESP32 Wrover Module' a 'COM4')

..
WiFi connected
Camera Ready! Use 'http://192.168.1.110' to connect
ets Jun  8 2016 00:22:57

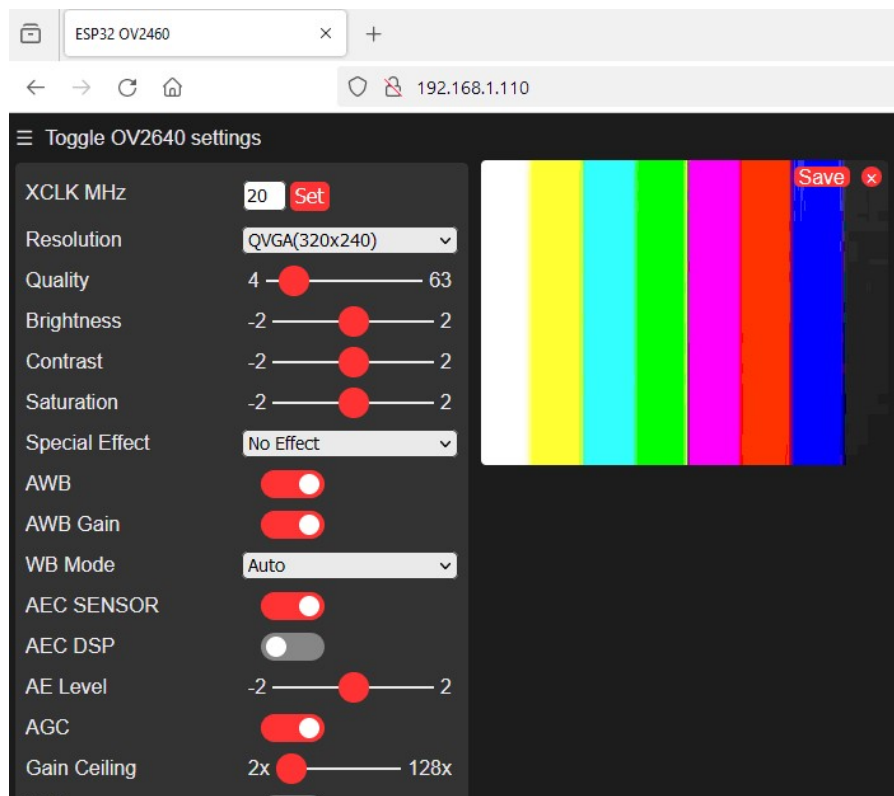
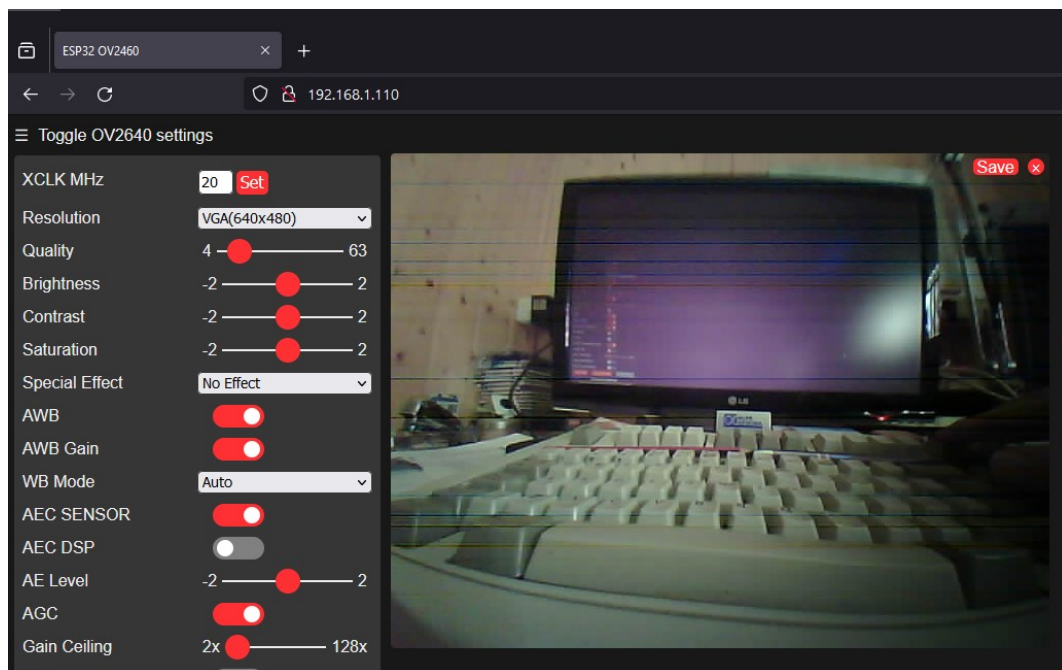
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0

.
WiFi connected
Camera Ready! Use 'http://192.168.1.110' to connect
```

Ahora vamos al navegador y ponemos esa IP en la barra de direcciones:



Y si no se os suelta ningún cable, la wifi esta próxima, y Murphi está de acuerdo, pues ...

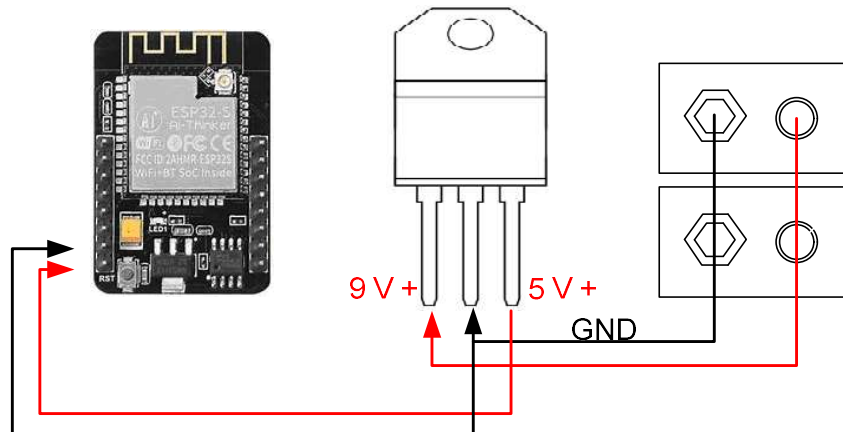


Ahora probáis todo lo que se os ocurra y el siguiente paso es el montaje en el vagón.

Información sobre el uso del programa, lo que se, al final del tutorial.

9. Montaje.

9.1. Esquema de conexión.



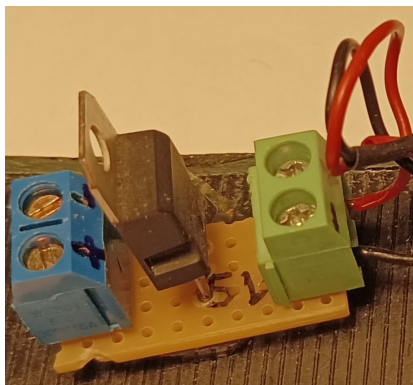
La batería de 9V alimenta el integrado L7805.
El integrado proporciona 5V a la cámara.
El cable rojo es el positivo, el negro GND, negativo.

9.2. Integrado L7805

Tiene tres patillas,
En una recibe +9V
En otra devuelve +5V
En la del centro recibe el negativo, GND.

9.3. Bornas.

Preparar la placa de circuito con las bornas soldadas y el integrado.
Se puede utilizar las bornas o se puede soldar los cables directamente en la placa del circuito.

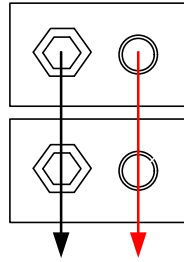


Utilizar cables identificados por el color o con un poco de cinta aislante de color rojo para tener claro cual es el cable que alimenta el positivo, como mínimo, si el cable es negro el otro no necesita identificación, ya es negro.

9.4. Conectar pilas 9 V.

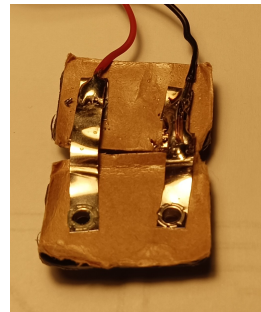
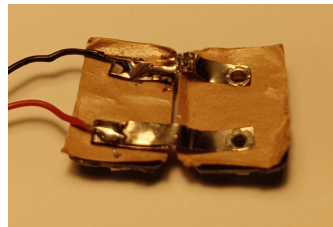
Conectar las pilas en paralelo, no en serie, en serie es 18V y no es necesario ese voltaje. En paralelo son los 9V que se rectifican a los 5V en el integrado.

Esto no son las pilas, son los conectores para conectar las pilas al integrado.



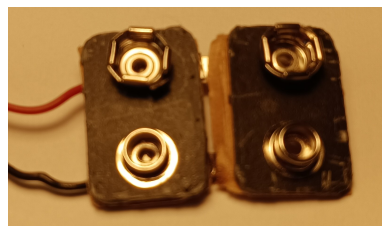
Los cables están colocados al revés, porque esto es donde conectaremos las pilas, así el rojo irá al conector “+” y el negro al “-”.

Para poner las pilas en paralelo, si tenéis alguna vieja de 9V, con unos alicates se puede desmontar los conectores y reutilizarlos para unir las dos pilas que vamos a utilizar.



Se dobla hacia fuera los bordes de la pila en su parte superior y se libera el conector de la pila. Este conector lleva unas laminas de cobre, se cortan lo más largos posible.

Luego se sueldan de forma que se unan los dos conectores iguales de cada soporte.



Con los conectores iguales soldados, el que antes era la salida del negativo de la pila, ahora se puede usar para conectar al positivo de una de las pilas de 9V, y el que era la salida del positivo ahora para conectar al negativo de la pila de 9V.

Si os entran con demasiada holgura, apretar un poquito los conectores y resuelto.

Después se utilizan para conectar la alimentación a las pilas y llevarla al integrado L7805.

En las tiendas de electrónica venden conectores para pilas de 9V con cable rojo y negro.

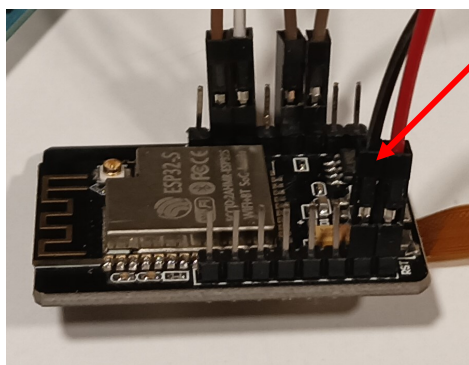


9.5. Alimentar la cámara.

Precaución con la polaridad, es mejor comprobar dos veces y conectar una.

Alimentar desde la salida del positivo del integrado a la patilla de 5V de la cámara, marcada en rojo y serigrafiada con 5V en la placa.

Alimentar desde la salida de la patilla GND del integrado a la patilla de GND de la cámara, marcada en negro y serigrafiada en la placa como GND.



Precaución con la polaridad, es mejor comprobar dos veces y conectar una.

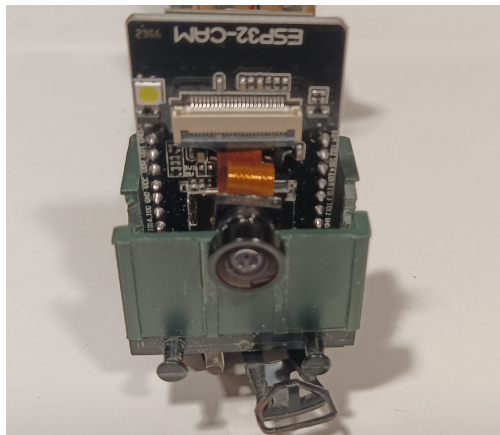
9.6. Montaje en el vagón.

Antes de montar todo en el vagón probarlo fuera, y comprobar que todo funciona correctamente, una vez probado es cuando se debe montar el conjunto en el vagón.

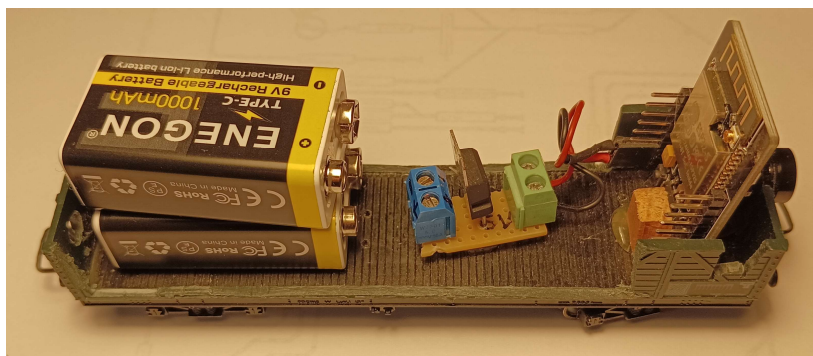
Hay vagones que tienen el bogie sujeto por el centro con un tornillo, eso le quita estabilidad al moverse de forma lateral.

En mi caso he usado un vagón de ese tipo porque era el que peor tenía y lo he sacrificado para probar esto.

Pero conviene que no sean así para que se gane en estabilidad, tampoco se va a hacer un rally, por lo que no es un problema importante, pero si hay que tenerlo presente.



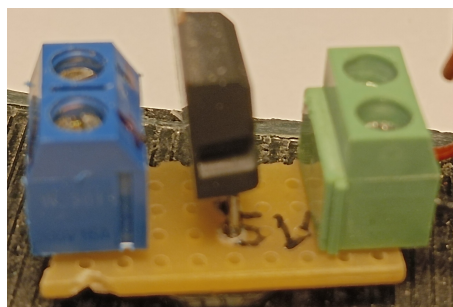
La altura de la parte más alta en mi caso es de 58 mm., cumple justo con el galibo, y debería pasar por todas las zonas, el galibo es de 59 mm. en la normativa NEM



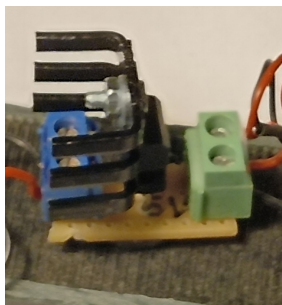
Las pilas caben en el vagón, en horizontal, en vertical, exceden la altura máxima.

Los mismos bordes del vagón las mantienen en su sitio, deben estar sueltas pues o bien para recargarlas o para cambiarlas hay que sacarlas de su sitio.

La plaquita con el integrado y las bornas en el centro del vagón, puede estar sujeta con cola termo fusible al vagón.



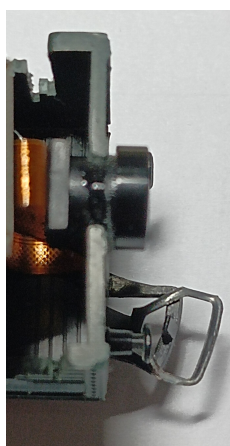
Si tuvierais que quitarla, se le echa un poco de alcohol por donde se ha puesto la cola, se espera un poco y se suelta con facilidad.



En la foto se ve el disipador añadido al integrado, es mejor, pues baja la temperatura de trabajo, y eso siempre es bueno para cualquier componente electrónico.

La cámara lleva una zona más delgada, que permite que se use para encajarla en el borde del vagón.

Para ello hay que hacerle al vagón en uno de los bordes una muesca de media circunferencia para que se sujete la cámara por esa zona más estrecha de la misma.



Avanzar muy poco a poco y realizar pruebas continuamente hasta llegar al punto en que la cámara entre haciendo una ligera presión, no debe estar suelta porque sino la cámara se moverá y la imagen será inestable.

Después probar la placa de la cámara colocada verticalmente, de forma que el led de iluminación asome por encima del borde del vagón.

Comprobar la distancia necesaria al borde y con cola termo fusible le ponéis un pequeño tope en el vagón, para evitar que se desplace hacia atrás, el cable plano de la cámara a la placa ayudará a mantener la placa verticalmente, junto con el tope.

Revisar todo, y probarlo.

Comprobar bien la polaridad antes de alimentar la cámara.

10. Funcionamiento del programa.

El programa arranca nada más conectar las baterías.
Se puede reiniciar pulsando el botón de reset en la placa.

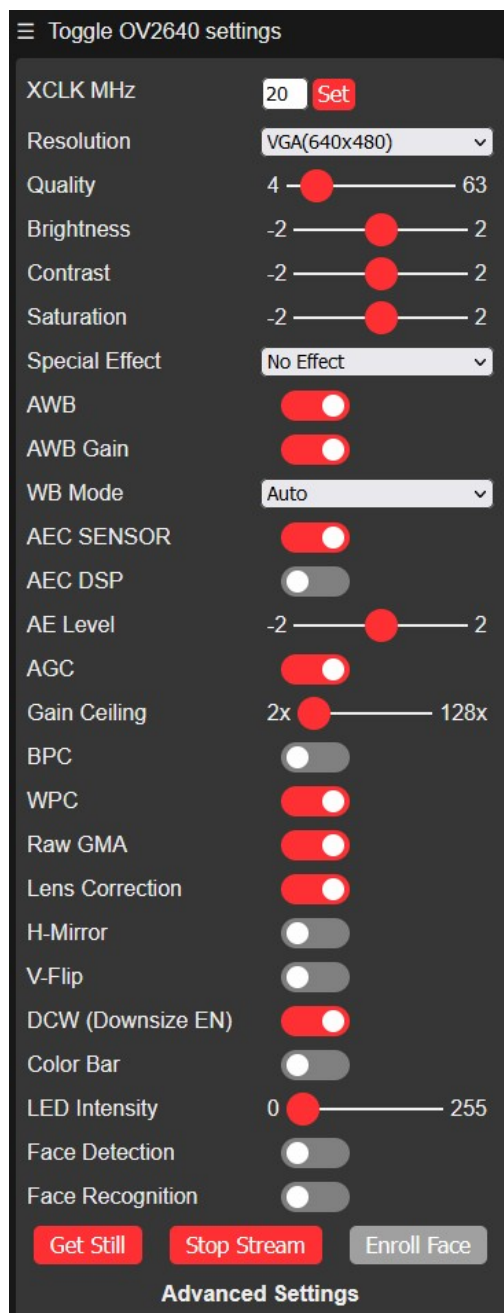
Hay que recordar la dirección IP que se genera en el primer inicio cuando se prueba en Arduino en el monitor serie.

Esta es la pantalla del programa.

Si se pulsa sobre

≡ Toggle OV2640 settings

Se esconde toda la zona de configuración.



Si no disponemos del voltaje adecuado por agotamiento de las pilas, el programa no se inicia.

Para iniciar la visualización de la imagen debemos pulsar en el botón Start Stream. Este cambiará su texto por Stop Stream en ese momento.



En resolución, seleccionáis la más adecuada a vuestro gusto, y que permita una adecuada visualización.

Con estos campos podéis regular, ajustar, la visualización de la imagen.



Es cuestión de probar y ajustar a nuestro gusto.

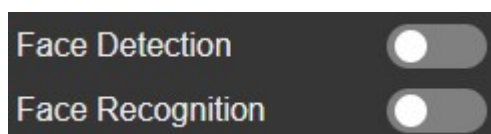


Si la imagen sale volteada, se pulsa en V-Flip para girarla verticalmente. Lo mismo si sucede en horizontal, se pulsa H-Mirror.

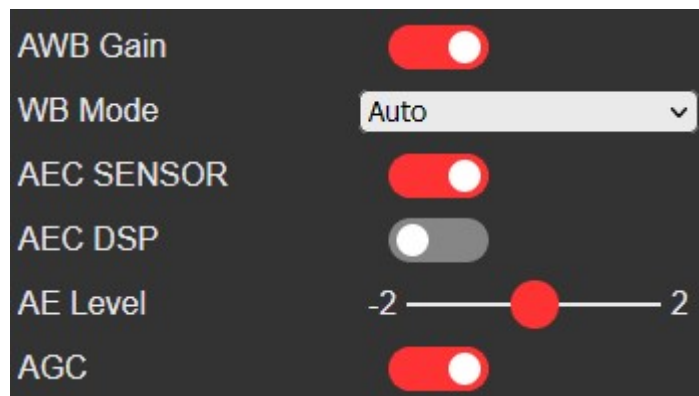
Podéis utilizar Color Bar, para ajustar los tonos, es un código de barras de colores.



Led Intensity, para regular la luz de la cámara en la reproducción, para las zonas de poca o nula iluminación.



Estos no los usamos, activar detección y reconocimiento facial.



AWB = Balance blanco automático, auto white balance.

AEC = Control de exposición automático, auto exposure control.

AGC = Control de ganancia automático, auto gain control.