

Reporte Técnico de un Sistema basado en SoC para Capturar y Almacenar Fotos en la Nube a través de una Conexión Wi-Fi

Resumen.

En este reporte se explicará el funcionamiento, alcance del proyecto, metodologías y procedimientos utilizados, esquemas, características y posibles mejoras del proyecto. Aquí se describe un sistema basado en el ESP32-CAM, programado con Arduino IDE, que permite capturar imágenes utilizando una cámara integrada y almacenarlas en la nube mediante una conexión Wi-Fi.

Introducción

El ESP32-CAM es un módulo compacto que combina un microcontrolador ESP32 y una cámara OV2640. Utilizando Arduino IDE, es posible programar el ESP32-CAM para capturar imágenes y enviarlas a la nube a través de una conexión Wi-Fi.

Componentes Utilizados

Módulo ESP32-CAM

https://articulo.mercadolibre.com.mx/MLM-773163751-esp32-cam-ov2640-wifi-bluetooth-camera-2mp-desarrollo- JM#is_advertising=true&position=2&search_layout=stack&type=pad&tracking_id=80378a45-e2b2-44b7-b786-0d559fcfed8e&is_advertising=true&ad_domain=VQCATCORE_LST&ad_position=2&ad_click_id=YTMzMzBiZTltMjE4ZS00ZWY1LTkxNzAtMGUwMzQ0ZDM2M2Y1

Cable USB tipo B para programación y alimentación

Computadora con Arduino IDE instalado

Acceso a una red Wi-Fi

Cuenta con un servicio de almacenamiento en la nube (por ejemplo, Google Drive, Dropbox).

Switch boton.

<https://www.steren.com.mx/micro-switch-de-push-con-4-terminales.html>

Interruptor.

<https://www.steren.com.mx/switch-deslizable-miniatura-de-1-polo-2-tiros-2-posiciones.html>

Cámara OV2640.

https://articulo.mercadolibre.com.mx/MLM-2238404554-modulo-de-camara-integrada-por-telefono-movil-ov2640-con- JM#position=24&search_layout=grid&type=item&tracking_id=b6b1cab-b5be-47b7-bba2-c9e784c20393

Datos Mviles 4G. (para el envío de fotos).

Programador USB a Serial para ESP32-CAM.

Procedimiento

1. Configuración del Entorno:
 - Descargue e instale Arduino IDE en su computadora.
 - Abra Arduino IDE y configure la placa ESP32 en el Administrador de tarjetas (seguir las instrucciones proporcionadas por Arduino para la instalación).
 - Conecte el ESP32-CAM a su computadora mediante el cable USB.
 - Seleccione la placa ESP32-CAM en Arduino IDE y configure el puerto correcto.
 - Instale la biblioteca "ESP32 Camera" en Arduino IDE (siguiendo las instrucciones del repositorio oficial de GitHub).
2. Programación del ESP32-CAM:
 - Cree un nuevo sketch en Arduino IDE.
 - Importe las bibliotecas necesarias, como "WiFi.h" y "CameraWebServer.h".
 - Configure los detalles de su red Wi-Fi, como SSID y contraseña.
 - Establezca las dimensiones y la calidad de las imágenes a capturar.
 - Escriba el código para inicializar la cámara y capturar imágenes.
 - Implemente el código para enviar las imágenes capturadas a la nube utilizando el servicio de almacenamiento elegido (por ejemplo, Google Drive API o Dropbox API).
3. Conexión con la Nube:
 - Registre una cuenta en el servicio de almacenamiento en la nube elegido.
 - Obtenga las credenciales o claves de API necesarias para acceder al servicio.

Implementar

+
?

Nueva implementación

Gestionar implementaciones

Implementaciones de prueba

Gestionar implementaciones

Activa

Sin título

Sin título

Archivadas

No hay implementaciones archivadas

Configuración

Versión

Versión 2 del 25 may, 21:04

Descripción

ID de implementación

AKfycbx8XnrJi48cEhJaevC8foVLp9E2cX-cW9nnzs4EFEisM077iBeOKt3JhcgZB8zDS...

Copiar

Aplicación web

URL

https://script.google.com/macros/s/AKfycbx8XnrJi48cEhJaevC8foVLp9E2cX-cW9n...

Copiar

Cancelar

Implementar

- Actualice el código del ESP32-CAM con las credenciales y configuraciones apropiadas.
- Implemente la lógica necesaria para autenticarse en el servicio de almacenamiento y enviar las imágenes capturadas.
- Pruebas y Verificación:
- Cargue el código en el ESP32-CAM.
- Conecte el ESP32-CAM a una fuente de alimentación adecuada.
- Asegúrese de que el ESP32-CAM esté conectado a la red Wi-Fi correctamente.
- Realice pruebas de captura de imágenes y verifique que se estén almacenando en la nube de manera correcta.

Resultados

- **Codigo Funcional en GitHub**

Este código es para comprobar que se está visualizando correctamente la imagen de la cámara, para lo cual se crea un sitio web donde se hace streaming y se manda la imagen recibida de la cámara y mediante la url se puede ver la calidad y funcionamiento de la cámara.

<https://github.com/D-cyber680/fotoADrive/tree/master>

Mandar fotos a Google Drive cada cierto tiempo, utilizando la configuración de la imagen de la cámara establecida en el código

<https://github.com/D-cyber680/fotoADrive/tree/Send-photos-to-Gdrive>

- **Carpeta de Google Drive con fotos capturadas**

<https://drive.google.com/drive/folders/1SjYopmfwoEuiUCsuBJACdOp4x9RsUVX-?usp=sharing>

- **Script para almacenamiento de imágenes es carpeta de google drive**

```
function doPost(e) {
    const name = Utilities.formatDate(new Date(), 'GMT-7',
    'yyyyMMdd-HH:mm:ss') + '.jpg' // Change GMT+7 with GMT in your area.
    const subFolderName = Utilities.formatDate(new Date(), 'GMT+7',
    'yyyyMMdd')
    const folderName = e.parameters.folder || 'ESP32-CAM'
    const data = Utilities.base64Decode(e.postData.contents)
    const blob = Utilities.newBlob(data, 'image/jpeg', name)

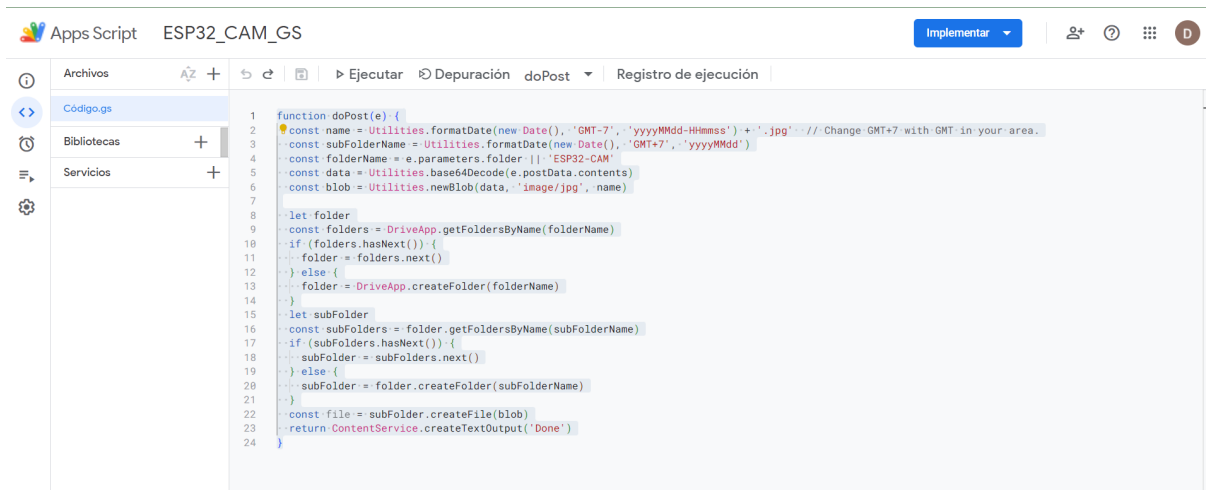
    let folder
    const folders = DriveApp.getFoldersByName(folderName)
    if (folders.hasNext()) {
        folder = folders.next()
    } else {
        folder = DriveApp.createFolder(folderName)
    }
    let subFolder
    const subFolders = folder.getFoldersByName(subFolderName)
    if (subFolders.hasNext()) {
        subFolder = subFolders.next()
    }
}
```

```

    } else {
      subFolder = folder.createFolder(subFolderName)
    }

    const file = subFolder.createFile(blob)
    return ContentService.createTextOutput('Done')
  }
}

```



Recomendaciones para refinamiento del proyecto

Fuentes de alimentación

1. Utilizar como fuente de alimentación las baterías de litio, como:

Ya sean dos baterías de litio de 1,5 V en serie o una batería de litio CR123 de 3 V, todo funciona perfectamente con baterías de litio. Estas mantienen un voltaje de 3V bastante constante, a menos de 2,7 voltios se utiliza más del 90% de la capacidad de una batería de litio, a 2,55 voltios está prácticamente vacía.

Por ejemplo, un Varta CR123 (3 V, 1700 mAh) incluso se puede usar en modo de espera durante más de 5 años, dependiendo, por supuesto, de la frecuencia con la que el ESP32 se despierta y tiene que hacer algo, o cuánto tiempo y con qué frecuencia WiFi o Se utiliza Bluetooth.

Como las baterías de litio tienen una autodescarga muy baja y también funcionan bien a -20 °C, estas son las preferidas.



2. Baterías de polímero de litio

Las baterías de polímero de litio o litio funcionan porque proporcionan suficiente energía para el ESP32. Sin embargo, el voltaje de 3,7 a 4,2 V es demasiado alto para el ESP32, dependiendo del estado de carga, y por lo tanto debe reducirse. Esto tiene la desventaja de que gran parte de la energía se utiliza

Las baterías de polímero de litio funcionan durante un día o varios días, pero durante semanas y años esto es casi imposible.



3. Baterías LiFePO 4

Las baterías modernas LiFePO 4 también funcionan de manera excelente, pero entregan aprox. 70% menos de energía que una batería de litio del mismo tamaño.

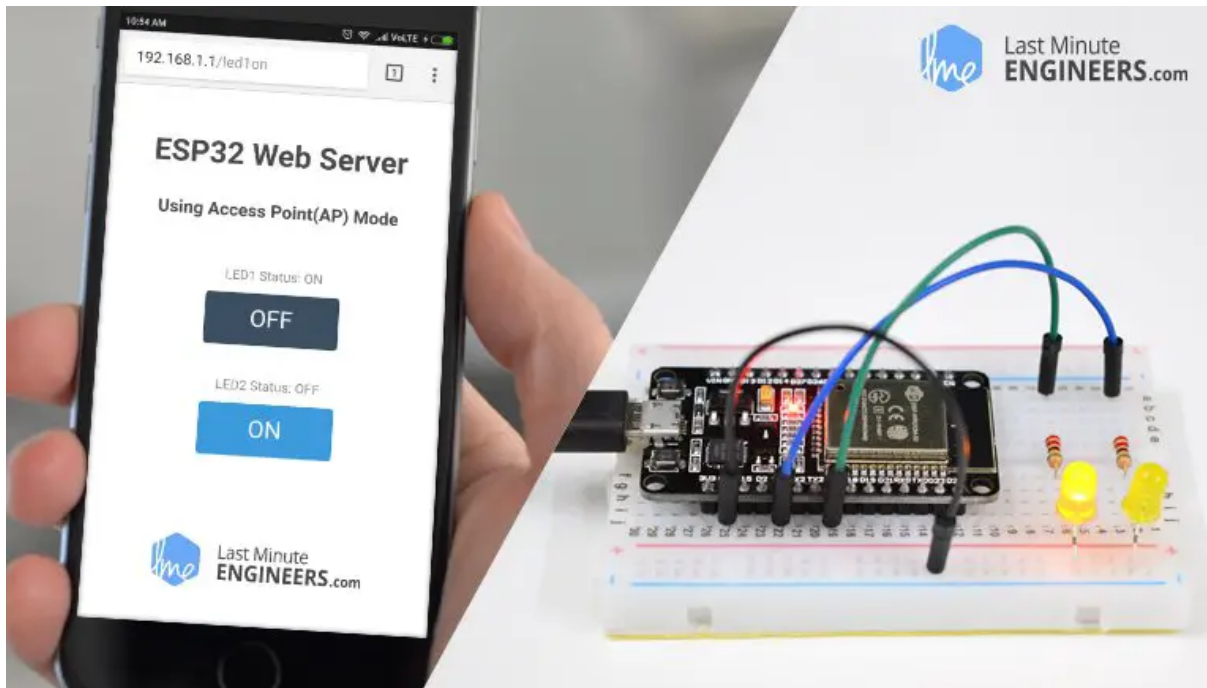
Las baterías LiFePO 4 también brindan un alto rendimiento sin problemas para la operación WiFi, pero no tienen la desventaja de las baterías de polímero de litio, que pueden incendiarse si se usan incorrectamente o si la calidad es deficiente.

Las baterías LiFePO 4 son adecuadas para un funcionamiento a corto plazo, es decir, semanas y meses. Sin embargo, es importante tener en cuenta que se requiere un cargador especial que sea adecuado para baterías LiFePO 4 3V.



Interfaz con usuario para el envío de fotos

Si se requiere enviar fotos utilizando una interfaz de usuario creo que la opción más conveniente es crear un servidor web en modo punto de acceso donde se cuenten con botones para el envío de las fotos, de esta manera no es necesario realizar una aplicación móvil para dispositivos Android o IOS. Como se muestra a continuación:



Conclusiones

En cuanto a que baterías usar para el proyecto las baterías de litio son la primera opción para una fuente de alimentación para el proyecto, ya que son las que ofrecen porcentaje de carga en mayor periodo de tiempo, lo cual lo hace ideal para este tipo de aplicación que consume muchos recursos del ESP32-CAM, ya que utiliza Wi-Fi en modo estación y el periférico de la cámara, los cuales consumen mucha energía.