Intrusion Detection System

Introduction

Microcontrollers have spread everywhere sometimes even without us realizing that. Fridges, kettles, toys, smart sockets and switches, automatic gates — all of them are operated with the help of microcontrollers. That to for a good reason. They can reduce the cost and size of the system by integrating most of the functions on a single chip. They can be programmed and reused for different applications, making them flexible and versatile. They can operate with low power and heat consumption, making them suitable for battery-powered or energy-efficient devices. They can interface with various sensors, actuators, and communication modules, making them capable of performing complex tasks. In this project we are going to creating a DIY intrusion detection system. One might ask why they need a system like that. Here are some reason that might make you interested in this project :

- Preventing theft: Theft detection systems can deter theft by making it more difficult for thieves to steal items.
- Reducing losses: Theft detection systems can also help to reduce losses by identifying and apprehending thieves.
- Protecting assets: Theft detection systems can help to protect assets by providing a layer of security.
- Enhancing security: Theft detection systems can also help to enhance security by providing information about potential threats.

Materials used

ESP32-CAM

The ESP32-CAM is a versatile and powerful camera module that can be used for a variety of applications. It is small, low-power, and easy to use. The ESP32-CAM is a great choice for anyone looking for a wireless camera module for their IoT project. Here are some of the key features of the ESP32-CAM:

- * ESP32 microcontroller: The ESP32-CAM is based on the ESP32 microcontroller, which is a powerful and versatile chip.
- * OV2640 camera: The ESP32-CAM comes with an OV2640 camera, which is a 2-megapixel camera.
- * Onboard TF card slot: The ESP32-CAM has an onboard TF card slot, which can be used to store images and videos.
- * Wi-Fi and Bluetooth connectivity: The ESP32-CAM has Wi-Fi and Bluetooth connectivity, which allows it to connect to a network or to a mobile device.
- Low power consumption: The ESP32-CAM is a low-power device, which means that it can be used for battery-powered applications.
- If you are looking for a small, low-power, and versatile camera module for your IoT project, then the ESP32-CAM is a great option.

Fig.1: ESP32-CAM

PIR sensor

A passive infrared (PIR) sensor is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications. PIR sensors work by detecting changes in infrared radiation. When an object moves in front of a PIR sensor, it blocks some of the infrared radiation that is being emitted by the object. This change in infrared radiation is detected by the PIR sensor and is used to trigger an alarm or turn on a light. PIR sensors are a type of passive sensor, which means that they do not emit any radiation. This makes them ideal for security applications, as they cannot be detected by intruders. PIR sensors are also relatively inexpensive and easy to use.

UartSBee programmer

UartSBee v5 is a USB to serial adapter that is compatible with FTDI cables. It has a 20-pin 2.0mm BEE socket and an integrated FT232RL chip. The FT232RL chip can be used for programming or communicating with MCUs. UartSBee can also be used to connect a PC to various wireless applications via a Bee compatible module. In addition, UartSBee provides breakouts for the bit-bang mode pins of the FT232RL chip. These bit-bang mode pins (8 I/O pins) can be used as a replacement for applications that involve the PC parallel port, which is becoming increasingly rare. Usb cable The usb cable is need to connect the programmer to the PC when the ESP32 cam is connected to flash the code. Later on, it is also used to power the ESP32 cam and the whole circuit.

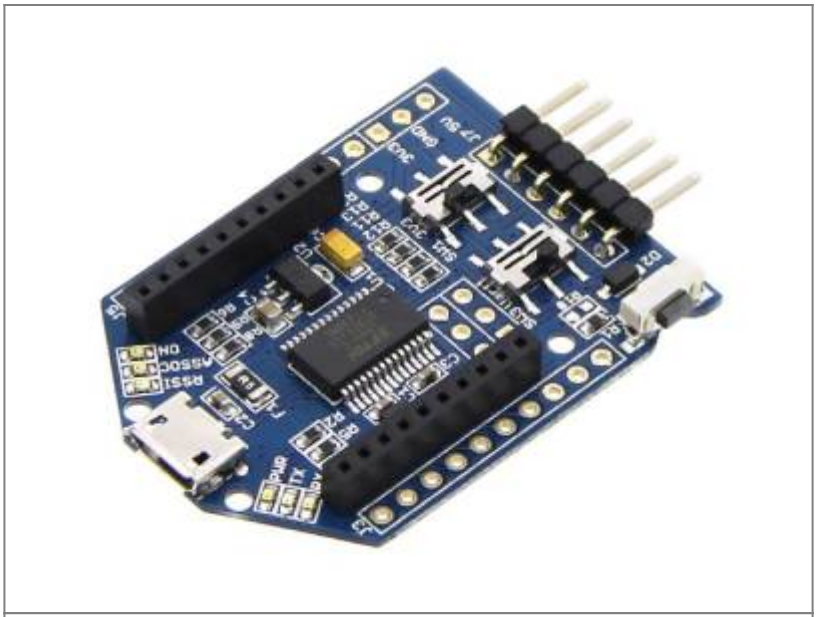


Fig.2: UartSBee V5

Breadboard

A breadboard is a prototyping tool that allows you to build electronic circuits without soldering. It is a rectangular board with a grid of holes that are connected together electrically. The holes are typically arranged in rows of five, and each row is connected to a common bus. This allows you to easily connect components together without having to solder them.

Jumper Wires

These are used to connect components with each other.

Tasks

The scenario we would like to implement is as follows:

- The ESP32-CAM module connects to Wi-Fi and falls into deep sleep
- The PIR motion sensor serves as an “alarm” that wakes the board when the motion is detected
- The board connects to flespi via MQTT, takes a shot, publishes the shot as an MQTT message and falls asleep again.
- The message is sent to the phone as well through Telegram

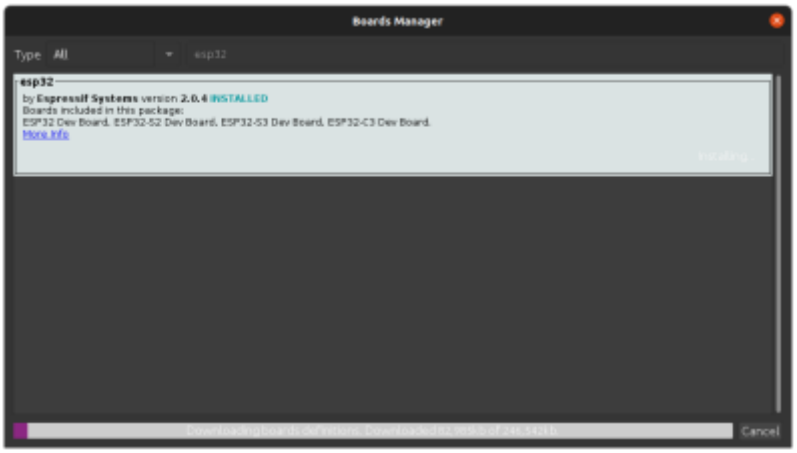
I would be using Arduino IDE for this project to flash the code

Code

Download the code from the following link https://github.com/flespi-software/ESP32-CAM

Methods

First we would install the board for the ESP32-Cam in Arduino. ^



^

Fig.3: Board Manager Arduino

To make the code work you need to specify your personalized data for some parameters:

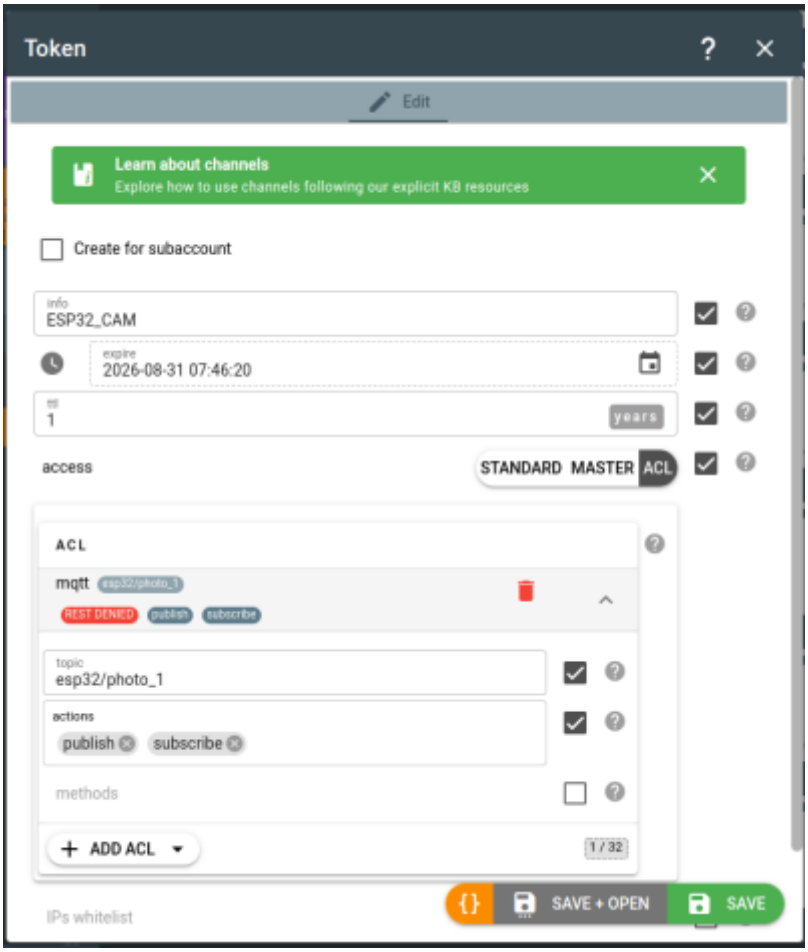
- WIFI hotspot access credentials :

```
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";
```

- Where to send the shots from the ESP32-CAM: I am using the flespi MQTT broker with the following config but you are free use whatever broker you prefer just change the details in the code as per your broker:

```
const char* mqtt_server = "mqtt.flespi.io";
const int mqtt_port = 1883;
const char* mqtt_user = "YOUR_FLESPI_TOKEN";
const char* mqtt_password = "";
```

My mqtt server settings looked something like this : ^



^

Fig.4: MQTT Server Settings

The ESP32-CAM can work in two modes in the code one is the periodic photo capture every N seconds or motion- triggered photo capture with help of the PIR sensor connected. In our case, we would prefer the capturing of photo triggered due to the PIR. In the code you have to change the value to 0 in the following code line.

```
#define SLEEP_DELAY 60000 //Delay for 60 Sec
```

Once we make the following changes we can start uploading the code to the ESP32-Cam by connecting it to the UartSBee using the connections given below.

UartSBee	**ESP32 Cam**
5V	VCC(5V)
UOR	TX
UOT	RX
GND	GND
	GPIO 0 to GND

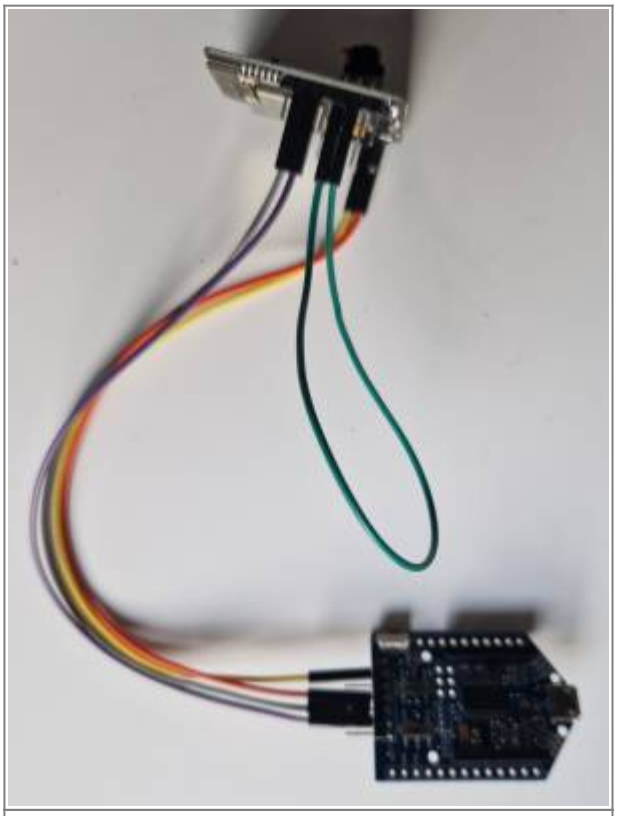


Fig.5: ESP32-CAM to UartSBee connections

After successfully flashing the ESP32-CAM we can start by making the circuit as per the given below circuit diagram :

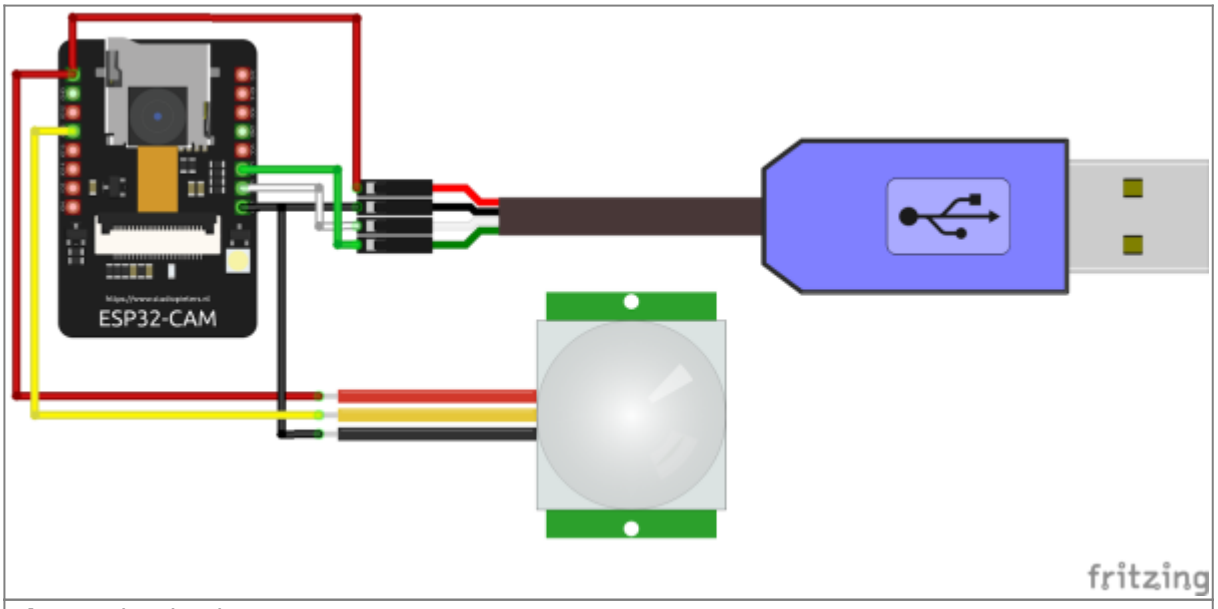


Fig.6: Circuit Diagram

The circuit looked something like this after following the circuit diagram:

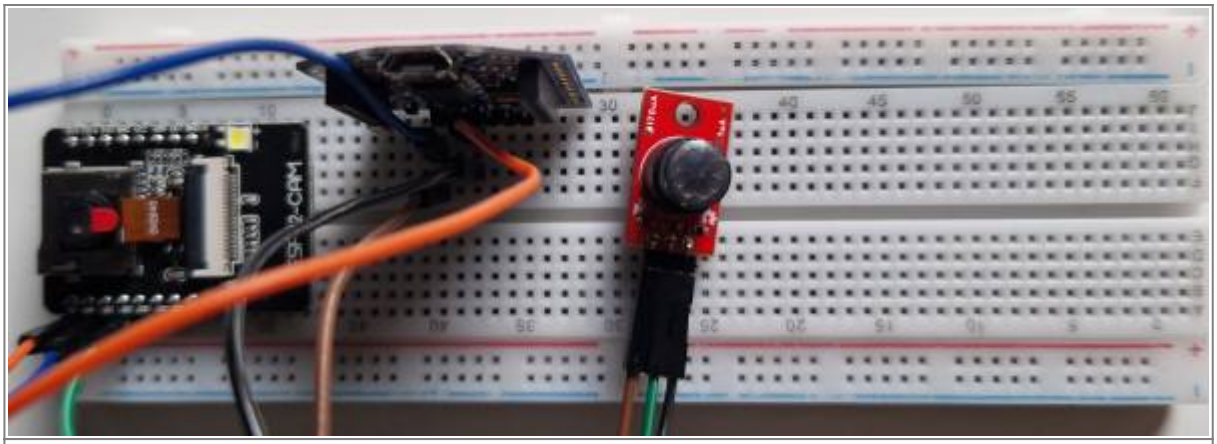


Fig.7: Circuit

Results

If the all the connections are right. The image with timestamp captured due to triggering of PIR sensor can be found in dashboard of your relevant MQTT broker:

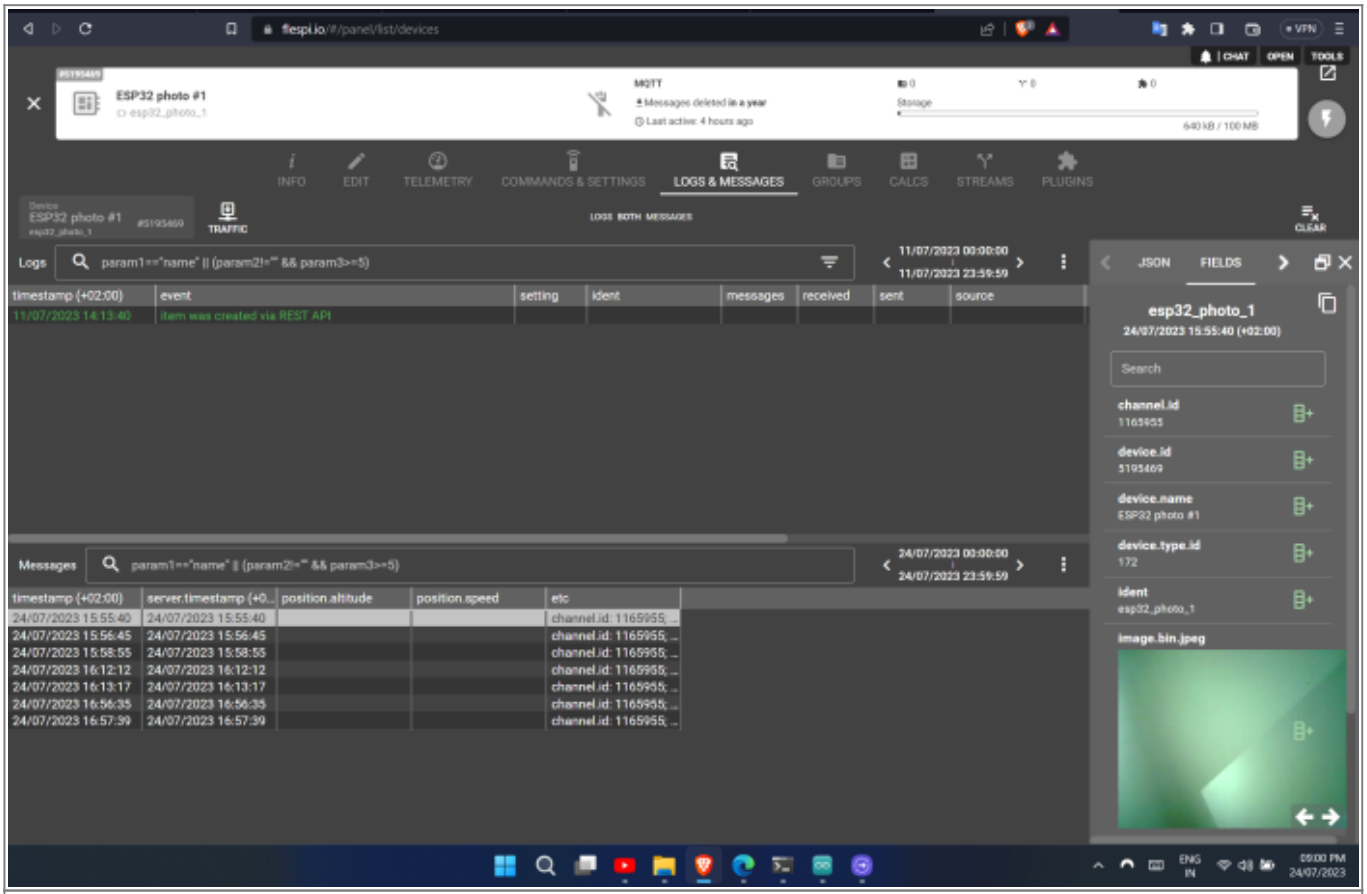


Fig.8: Image on MQTT

Smartphone Notification

To the get the notification on the phone. The following procedure is needed:

First is to download the Telegram app on your phone and then the install the application Node-Red. You would need to create a bot on the Telegram app and install the following libraries in the Node-Red

as shown in the image.

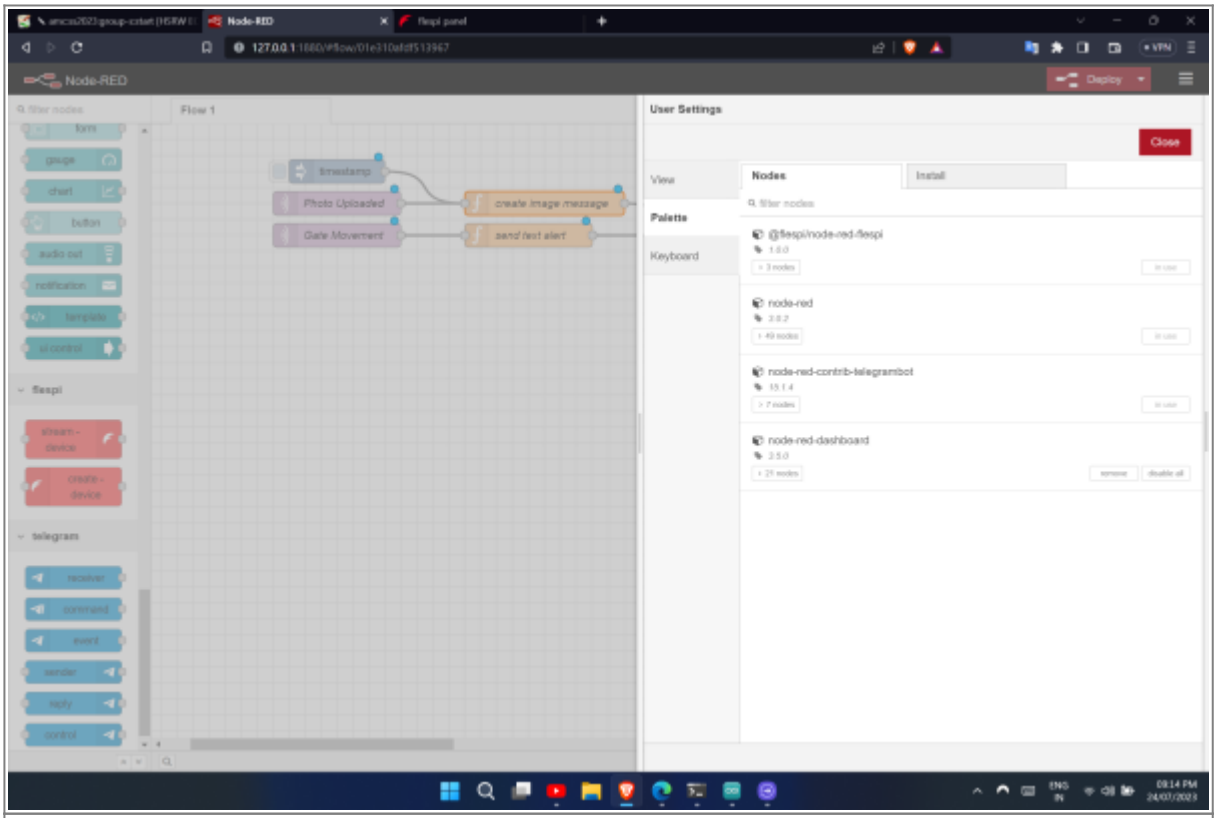


Fig.9: Libraries for Nod-Red

Once the nodes are installed we make connections with the following nodes :

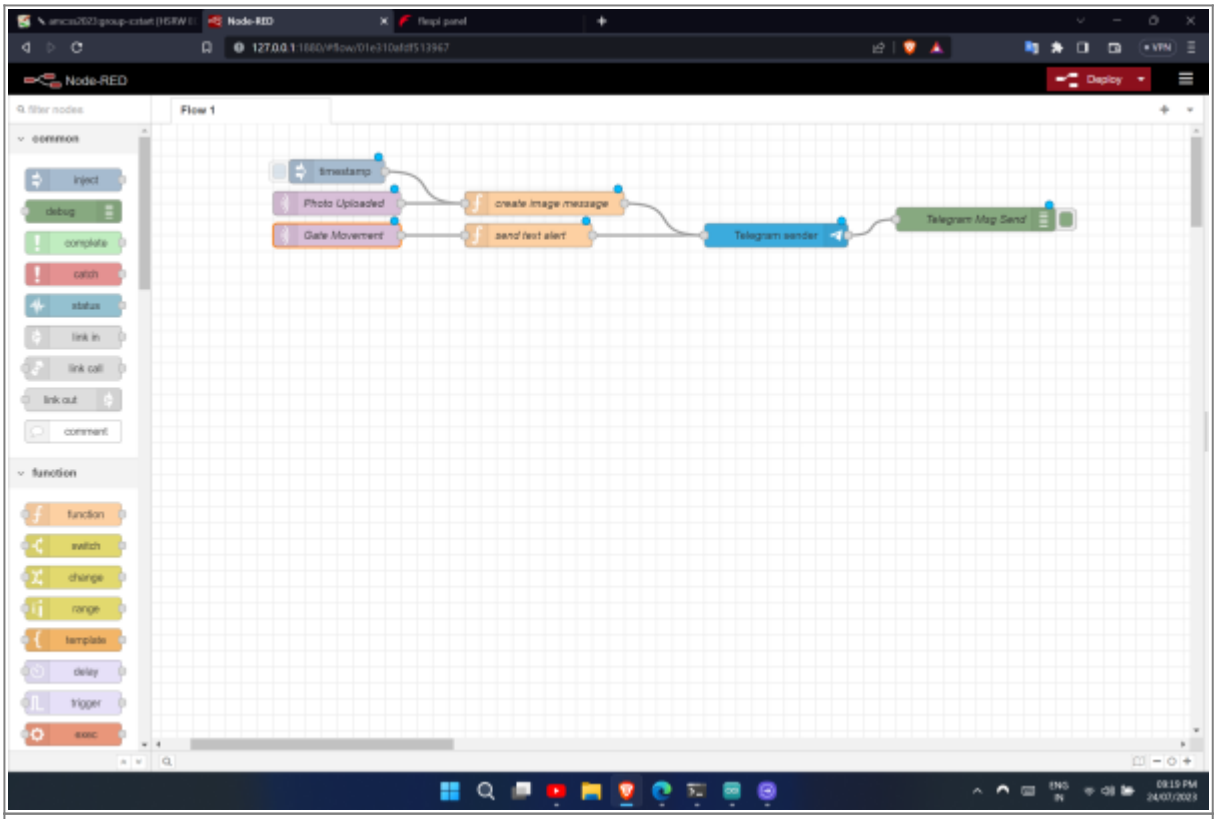


Fig.10: Node-RED connections

The JSON file for Node-RED can be found below :

[{"id":"c28d1213.3a847","type":"mqtt in","z":"01e310afdf513967","name":"Gate Movement","topic":"gate/motion/state","qos":"0","datatype":"auto","broker":"e50612ee.93d158","nl":false,"rap":false,"inputs":0,"x":360,"y":140,"wires":[["b6291d0.4869e6"]]},{"id":"e50612ee.93d158","type":"mqtt-broker","name":"MQTT_Broker","broker":"mosquitto","port":"1883","clientid":"","autoConnect":true,"usetls":false,"compatmode":false,"protocolVersion":4,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":"","credentials":{}}]

The telegram bot and MQTT needs to configured in Node-RED.

Discussion And Conclusion

The ESP32-CAM is a powerful microcontroller which can be used for DIY projects. This project is just one showcase how it can be used. It can be used in a variety of different projects. Here are some possible use cases:

1. As a surveillance camera
2. As smart water meter and gas meter
3. As a smart door lock with face recognition.

And in many other appliances

Possible problems that might occur:

1. During the flashing of the ESP32-CAM all the connections should be secured with no faulty cable
2. One would need a high quality cable to flash otherwise errors may occur.
3. The ESP32-CAM works better with 2.4ghz connection compared to 5ghz. If your router is transmitting 5ghz WIFI your connection would not be stable.
4. If ESP32-CAM is triggered a lot of times with flash it gets overheated easily in a hot environment and may short circuit.
5. The ESP32-CAM works better with 5V connection compared to 3V.

To conclude microcontroller are amazing devices which most people can afford. If one decides automize things in his life. They are an essential tool in toolbox.

Links

https://flespi.com/blog/esp32-cam-motion-triggered-photo-mqtt

https://nodered.org/

https://www.arduino.cc/

https://github.com/flespi-software/ESP32-CAM

From:
https://student-wiki.eolab.de/ - **HSRW EOLab Students Wiki**

Permanent link:
https://student-wiki.eolab.de/doku.php?id=amc:ss2023:group-s:start&rev=1690230282

Last update: **2023/07/24 22:24**