Intrusion Detection and Monitoring System

Team: WatchfulEyes

1.DEVARINTI JYOTHIKA (21BLC1050)

2. AKRUTI SOMKUWAR(21BLC1024)

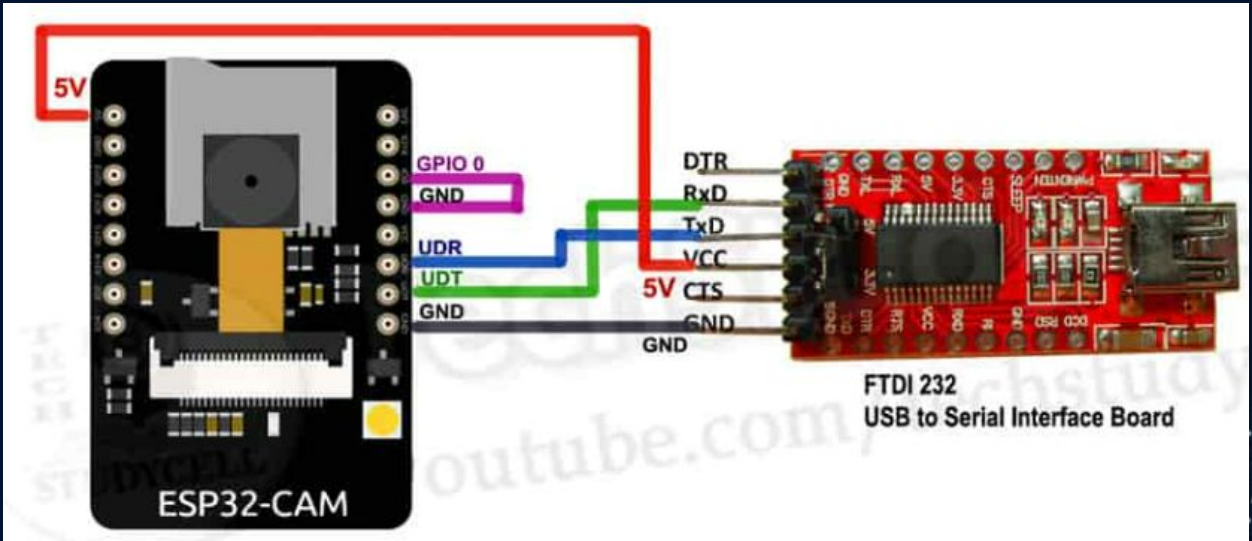
3. ADITYA UTTAM DHAVALE(21BLC1040)

Project Objective

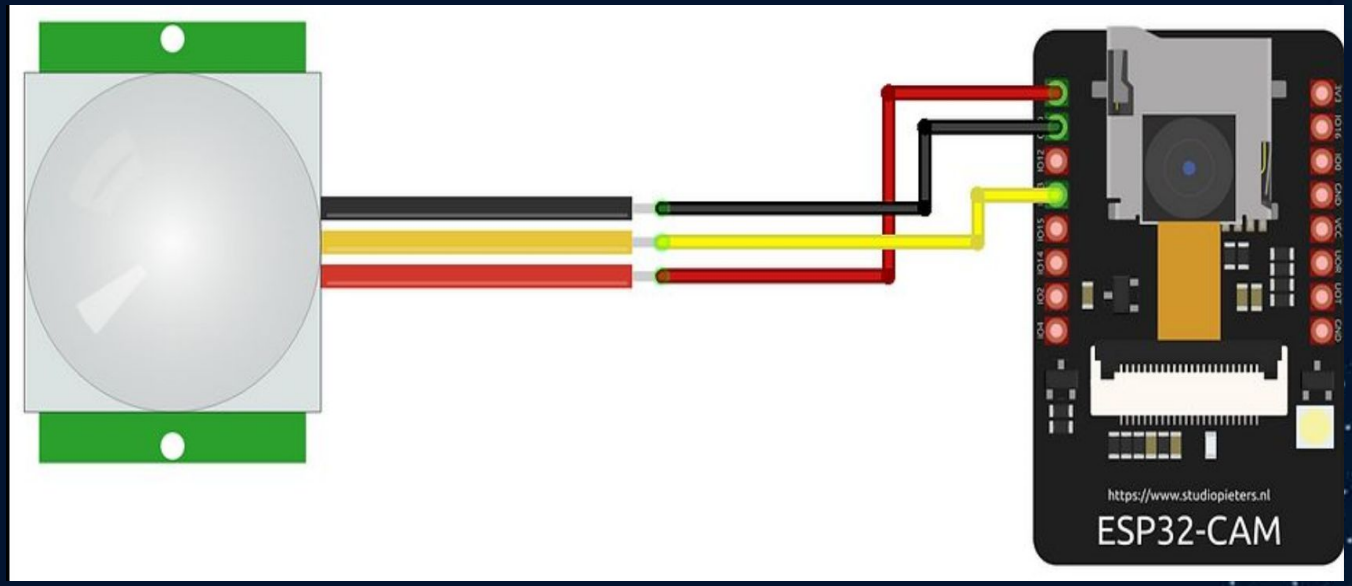
Our project aims to develop a motion-activated intruder detection system utilizing motion sensors and camera in an Internet of Things (IoT) framework. The Motion Sensor Camera project with WiFi Connectivity aims to develop a versatile surveillance system that detects motion and captures images wirelessly. The motion detector camera system will provide a reliable,efficient and user-friendly solution for monitoring and recording motion events in various settings such as homes, offices, warehouses or outdoor areas.

Circuit Diagram

- FTDI 232 Serial interface and ESP32 circuit



PIR and ESP 32 cam circuit



Code

https://codeshare.io/Watchfuleyes

Explanation of code :

- The code is an arduino sketch for an ESP32 cam motion alert system that sends an image to the telegram.
- The code contains the libraries and define the pins of ESP32 cam.
- The wifi connection is established by providing the SSID and password in the code.
- The code checks if the Wifi connection is successful,if not ,it permits for reset.
- If wifi is connected successfully , the code blinks an LED a few times.
- The PIR motion sensor state is read in loop() function.
- If motion is detected (sensor state = High) the alert telegram () function is called to send the image.
- A delay of 15 sec is added to avoid continuous alerts .

- The alerts2telegram() function establishes a secure connection to the telegram API server.
- captures an image using ESP 32 cam
- alerts2telegram() function sends telegram the image.
- It waits for the response from the server and retrieves the response.

https://codeshare.io/Watchfuleyes

code Feeding in ESP32 CAM using FTDI 232 serial Interface

Output | Serial Monitor

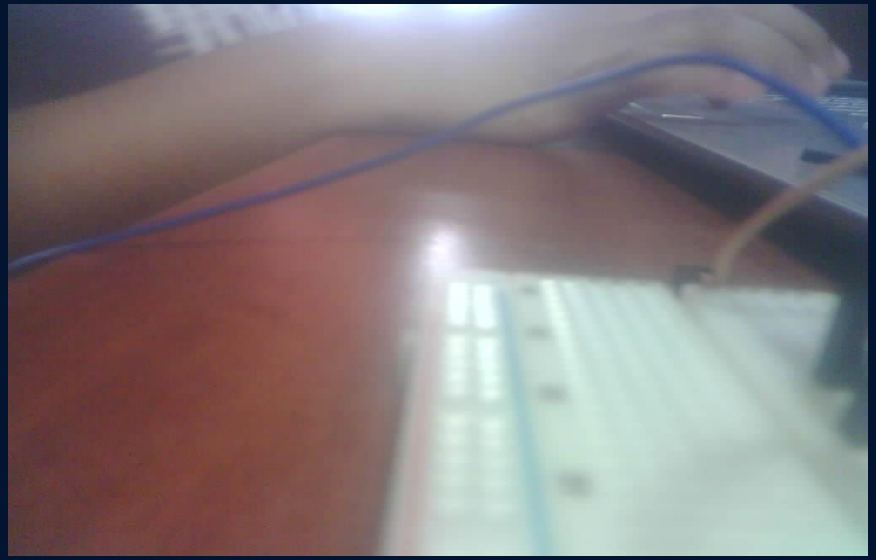
```
Writing at 0x00038000... (34 %)
Writing at 0x0003c000... (37 %)
Writing at 0x00040000... (40 %)
Writing at 0x00044000... (43 %)
Writing at 0x00048000... (46 %)
Writing at 0x0004c000... (50 %)
Writing at 0x00050000... (53 %)
Writing at 0x00054000... (56 %)
Writing at 0x00058000... (59 %)
Writing at 0x0005c000... (62 %)
Writing at 0x00060000... (65 %)
Writing at 0x00064000... (68 %)
Writing at 0x00068000... (71 %)
Writing at 0x0006c000... (75 %)
Writing at 0x00070000... (78 %)
Writing at 0x00074000... (81 %)
Writing at 0x00078000... (84 %)
Writing at 0x0007c000... (87 %)
Writing at 0x00080000... (90 %)
Writing at 0x00084000... (93 %)
Writing at 0x00088000... (96 %)
Writing at 0x0008c000... (100 %)
Wrote 950160 bytes (520978 compressed) at 0x00010000 in 8.1 seconds (effective 943.7 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1536.0 kbit/s)...
Hash of data verified.

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

https://codeshare.io/Watchfuleyes

Serial Monitoring in Arduino

```
16:50:34.944 -> Connecting to Mi A2
16:50:35.948 ->
16:50:35.948 -> STAIP address:
16:50:35.948 -> 192.168.43.233
16:50:35.948 ->
16:50:38.336 -> 1
16:50:42.549 -> Connected to api.telegram.org
16:50:47.426 -> ..........................................................................................
16:51:08.409 -> 1
16:51:12.021 -> Connected to api.telegram.org
16:51:40.701 -> ..............................................
16:51:45.716 -> {"ok":true,"result":{"message_id":2,"from":{"id":5826723062,"is_bot":true,"first_name":"camDJAS","username":"camDJAS_bot"},"chat":{"id":55837
16:51:56.763 -> 1
16:51:58.707 -> Connected to api.telegram.org
16:52:01.222 -> ..............
16:52:02.825 -> {"ok":true,"result":{"message_id":3,"from":{"id":5826723062,"is_bot":true,"first_name":"camDJAS","username":"camDJAS_bot"},"chat":{"id":55837
16:52:13.899 -> 1
16:52:15.920 -> Connected to api.telegram.org
16:52:18.380 -> ...........
16:52:19.613 -> {"ok":true,"result":{"message_id":4,"from":{"id":5826723062,"is_bot":true,"first_name":"camDJAS","username":"camDJAS_bot"},"chat":{"id":55837
16:52:30.584 -> 1
16:52:32.706 -> Connected to api.telegram.org
16:52:37.016 -> ............
16:52:38.320 -> {"ok":true,"result":{"message_id":5,"from":{"id":5826723062,"is_bot":true,"first_name":"camDJAS","username":"camDJAS_bot"},"chat":{"id":55837
16:52:49.406 -> 1
16:52:51.752 -> Connected to api.telegram.org
16:52:55.441 -> ...........
16:52:56.657 -> {"ok":true,"result":{"message_id":6,"from":{"id":5826723062,"is_bot":true,"first_name":"camDJAS","username":"camDJAS_bot"},"chat":{"id":55837
16:53:07.624 -> 1
```

Sample pictures



Mechanism of Packet transfer

1.Data Generation: The intruder detector system generates data, such as sensor readings or alarm signals, when it detects potential intrusions.

2.Packetization: The generated data is divided into small units called packets. Each packet contains a portion of the original data and additional information required for transmission, such as the source and destination addresses.

3.Network Layer Encapsulation: The packets are encapsulated at the network layer of the communication protocol stack. In the case of Wi-Fi, this would typically involve encapsulating the packets within Internet Protocol (IP) packets.

4.Transport Layer Encapsulation: The encapsulated packets are further encapsulated at the transport layer, typically using the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). This includes adding appropriate headers and checksums for error detection and correction.

5.Wi-Fi Transmission: The encapsulated packets are sent over the Wi-Fi network. Wi-Fi operates on the IEEE 802.11 standard, which defines how data is transmitted wirelessly between devices. The packets are modulated into radio waves and transmitted over the air using specific frequency channels.

6.Reception and Decapsulation: The receiving device, such as a central monitoring station or another connected device, receives the transmitted packets. It then decapsulates the packets by removing the headers added during encapsulation.

7.Processing and Analysis: The received data packets are then processed and analyzed by the receiving device. In the context of an intruder detector system, this analysis may involve comparing sensor readings against predefined thresholds or patterns to determine if an intrusion has occurred.

8. Alerts : Based on the analysis results, the receiving device can generate an alert or initiate a response.

Progress

● We assembled the components Esp32 camera , FTDI 232 USB to serial communication cable , NPN transistor , PIR sensor on the breadboard using jumping cables.

● Next we downloaded Arduino-IDE Software along with esp32 libraries and then we installed the Telegram app and used the api bots to get auth token and chat id.

● We interfaced the code mentioning the wifi info and chat id info in Esp32 camera using FTDI 232 USB to serial communication cable .

- we designed the circuit as per circuit diagram using PIR sensor,ESP32 cam and FTDI 232 .
- And tested for the output .

THANK YOU