NIGHT VISION PATROLLING ROBOT WITH SOUND SENSOR USING COMPUTER VISION TECHNOLOGY

N.Hemavathy[1], Arun.K[2], Karthick.R[2], Srikanth.A.P[2], Venkatesh.S[2]

[1]Assistant Professor, ECE, Velammal Engineering College, Chennai.
[2]Student members, ECE, Velammal Engineering College, Chennai.

ABSTRACT: In this project, we propose a security patrolling robot that uses night vision camera for securing any premises. The robotic vehicle moves at particular intervals and is equipped with night vision camera and sound sensors. It uses a predefined line to follow its path while patrolling. It stops at particular points and moves to next points if sound are detected. The system uses IR based path following system for patrolling assigned area. It monitors each area to detect any intrusion using 360degree rotating HD camera. It has the ability to monitor sound in the premises. Any sound after company is closed and it starts moving towards the sound on its predefined path. It then scans the area using its camera to detect any human faces detected. It captures and starts transmitting the images of the situation immediately on sound or human face detection. Here we use IOT Local Area Network (LAN) for receiving transmitted images and displaying them to user with alert sounds. Thus we put forward a fully autonomous security robot that operates tirelessly and patrols large areas on its own to secure the facility.

Keywords: Night vision, patrol robot, sound sensor, IOT, security, machine learning, line follower, image processing.

1. INTRODUCTION

A significant proportion of Indian firms might have faced data theft issues recently. Around 41 per cent of these companies saw such instances, compared to the global average of 29 per cent, showed a survey by a risk management and consultancy firm, released on Monday. India's number is higher than other countries such as the United States (26 per cent), the United Kingdom (32 per cent) and Japan (27 per cent). Autonomous security robots are a new groundbreaking innovation in advanced security and surveillance technology. Guards patrolling an area with flashlights and batons never worked very well; however, today there is an immergence of smart security systems with intelligent sensors, embedded systems, autonomous control mechanism and mobile application. The first security surveillance robot was proposed by Everett, H. & Gage, D.W, 1999 in "Mobile Detection, Assessment and Response System (MDARS)". Since then security robots have become a growing interest with increasing developments in research and application. The idea behind this is to secure the whole area of a building or premises. Any small sound results in the alert and robot automatically goes to the particular area and captures the image of that area and sends it to the user. Raspberry Pi connected with the camera plays an important role in making an automatic robotic system. During the occurrence of an abnormal event like theft, the patrol robot detects the sound and captures the image of the intruder. The image of the intruder is sent to the security department via Email and Short Messaging Service (SMS). An automatic patrolling vehicle acts as a security patroller in the security system, which can monitor those dead zones of the traditional fixed surveillance system. Countless applications today are mobile service security robots, including autonomous navigation, security patrolling, housework, search-and-rescue operations, material handling, manufacturing, and automated transportation systems. Autonomous navigation remains one of the primary challenges in the mobile-robot industry; many control algorithms and techniques have been recently developed that aim to overcome this challenge. We use line follower with IR Sensors to perform night patrol navigation. The security patrol robot will utilize several sensors and motors in order to navigate indoors. It will also be able to communicate and be controlled via Wi-Fi.

2. EXISTING SYSTEM

In the existing system, robot has to be controlled remotely where the person has to monitor the area using remote control and more human resource is needed for this operation of the patrol robot. Some systems require a smart phone for controlling the night vision robot. Some use local host webpage hence the control range is limited to local network range. Lacking of few parts, autonomous operation requires work force to operate

3. PROPOSED METHODLOGY

In this proposed system, Raspberry pi is installed with the night vision camera which help the system to go for the automation and help to find the human or any problem detected using the sound sensor and according to the sound produced it automatically goes to that area and capture the image and send it to user using IOT technology.

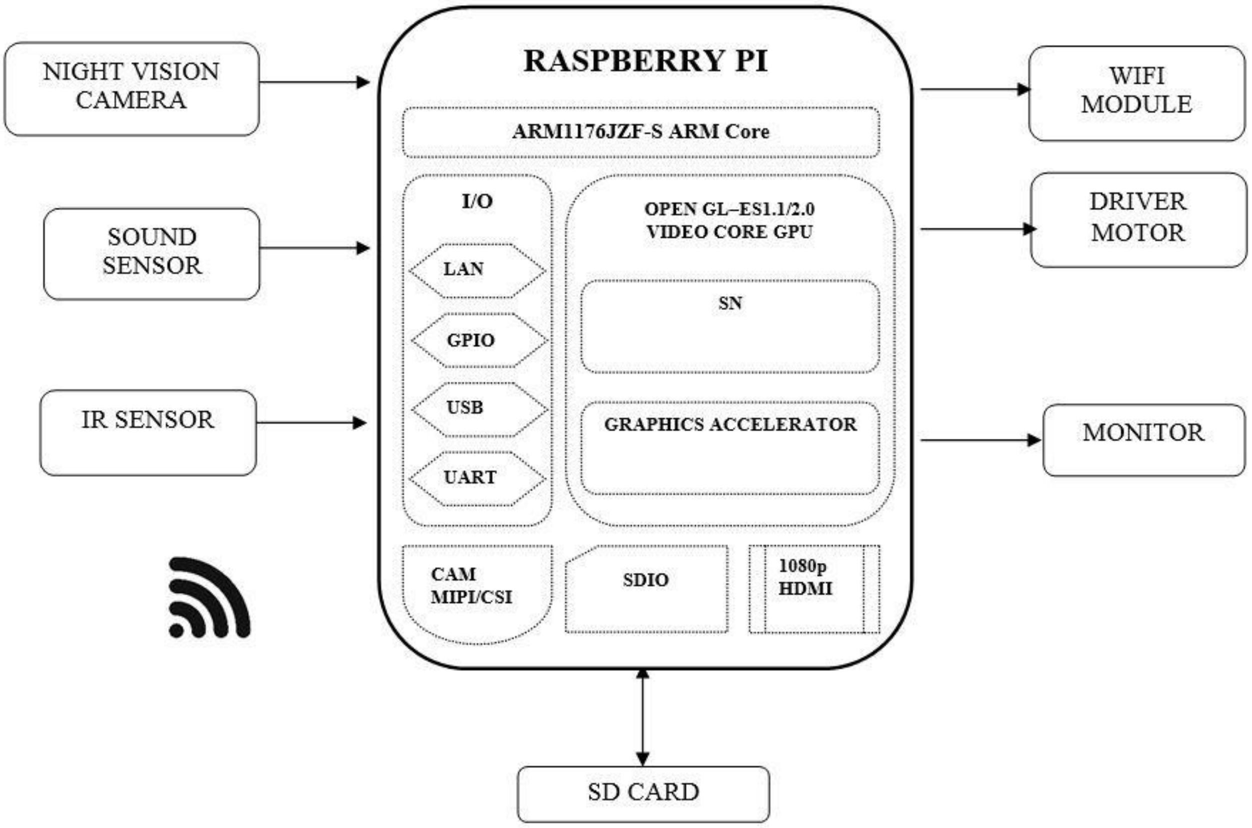


FIGURE 1: BLOCK DIAGRAM OF NIGHT VISION PATROL ROBOT

3.1 BLOCK DIAGRAM DESCRIPTION

• In this project IR Sensor is used to make the robot move automatically following a specific path.

• Sound sensor is used to know the sound in the particular area.

• IOT is used to send the capture image to the person.

• Connect USB camera with raspberry pi

• Connect power supply for Raspberry pi

• Plug the HDMI cable in Raspberry pi from the monitor using VGA to HDMI converter cable

• Connect USB Mouse and USB keyboard to the Raspberry pi

3.4 HARDWARE REQUIREMENTS

- Raspberry Pi
- Night Vision Camera
- IR Sensor
- Sound Sensor
- DC Motor (Robot module)
- SD card
- Monitor

3.5 SOFTWARE REQUIREMENTS

- Raspbian Jessie
- Language – Linux
- Python

4. HARDWARE COMPONENTS USED

4.1 Raspberry Pi 3: Broadcom BCM2835

Raspberry Pi is a small single-board Computer developed in UK by Raspberry Pi foundation to promote the teaching of computer science in schools and in developing countries. Original model become far more popular than anticipated sealing outside of its target market, for uses such as robots.

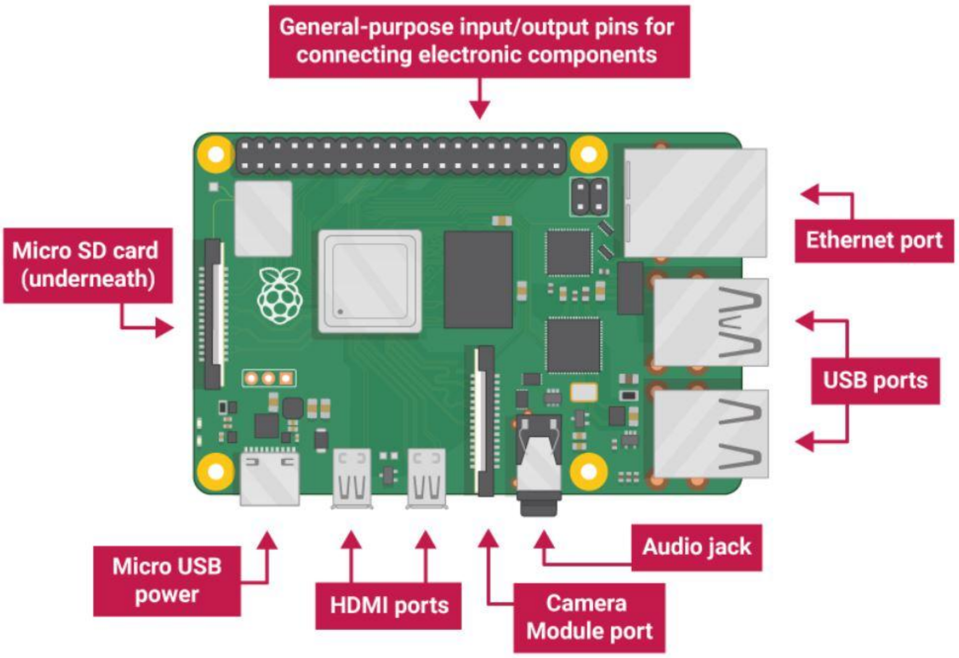
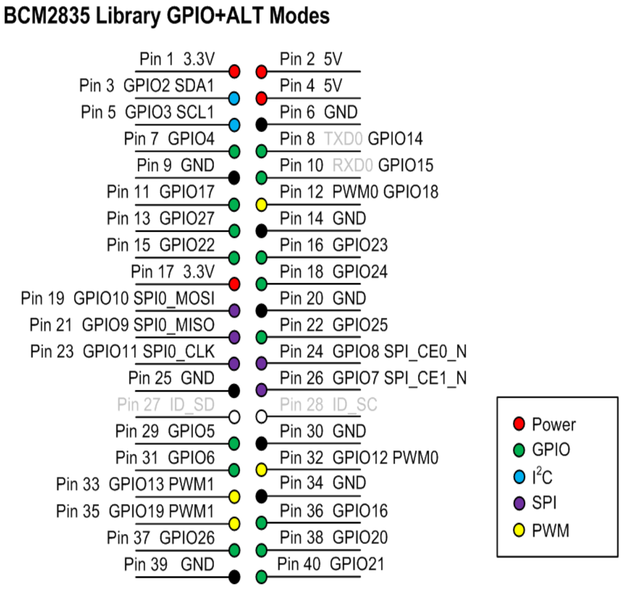

General-purpose input/output pins for connecting electronic components
Micro SD card (underneath)
Ethernet port
USB ports
Audio jack
Micro USB power
HDMI ports
Camera Module port

FIGURE 2: RASPBERRY PI DIAGRAM

4.1.1 FEATURES

- CPU speed ranges from 700 MHz to 1.2 GHz.
- On board Memory (RAM) ranges from 256 MB to 1 GB.
- USB slot differs from 1 slot to USB slots.
- HDMI, composite video output and 3.5mm phone jack.
- Low level output is provided by GPIO pins which support common protocols like I2C (inter-integrated circuit).
- Ethernet 8 Position 8 Contact (8P8C).



Pin 1 3.3V	●	●	Pin 2 5V
Pin 3 GPIO2 SDA1	●	●	Pin 4 5V
Pin 5 GPIO3 SCL1	●	●	Pin 6 GND
Pin 7 GPIO4	●	●	Pin 8 TXD0 GPIO14
Pin 9 GND	●	●	Pin 10 RXD0 GPIO15
Pin 11 GPIO17	●	●	Pin 12 PWM0 GPIO18
Pin 13 GPIO27	●	●	Pin 14 GND
Pin 15 GPIO22	●	●	Pin 16 GPIO23
Pin 17 3.3V	●	●	Pin 18 GPIO24
Pin 19 GPIO10 SPI0_MOSI	●	●	Pin 20 GND
Pin 21 GPIO9 SPI0_MISO	●	●	Pin 22 GPIO25
Pin 23 GPIO11 SPI0_CLK	●	●	Pin 24 GPIO8 SPI_CE0_N
Pin 25 GND	●	●	Pin 26 GPIO7 SPI_CE1_N
Pin 27 ID_SD	○	○	Pin 28 ID_SC
Pin 29 GPIO5	●	●	Pin 30 GND
Pin 31 GPIO6	●	●	Pin 32 GPIO12 PWM0
Pin 33 GPIO13 PWM1	●	●	Pin 34 GND
Pin 35 GPIO19 PWM1	●	●	Pin 36 GPIO16
Pin 37 GPIO26	●	●	Pin 38 GPIO20
Pin 39 GND	●	●	Pin 40 GPIO21

FIGURE 3: PIN CONFIGURATION OF RASPBERRY PI

4.2 NIGHT VISION CAMERA

A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video.

FIGURE 4: NIGHT VISION CAMERA

4.2.1 FEATURES

- Inbuilt sensitive microphone and image sensor quality CMOS sensor
- Image resolution interpolated to 25 mega pixels with 6 light sensors ; 16 MP Image Resolution ; USB Interface ; Night Vision ; Usb Cable Length: 1m ; Focus Range: 4 cm to infinity
- Image control color saturation, brightness, sharpness and brightness is adjustable
- Snap shot switch for taking still pictures. Focus Range 4cm to infinity
- Anti-flicker 50Hz, 60Hz or outdoor
- Resolution hardware: 500K pixels
- Image quality: RGB24 or I420
- Exposure: Auto or manual and angle of view: 58 Degree
- Interface: USB2.0
- Frame rate: 30 fps (max)

4.3 INFRARED SENSOR

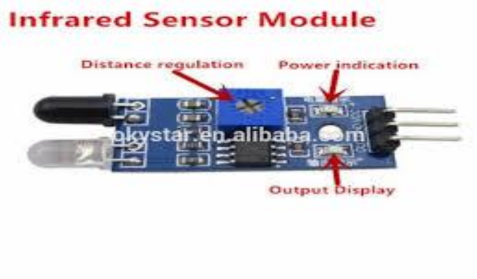


FIGURE 5: INFRAREDSENSOR

4.3.1 Working Principle

An Infrared Sensor works in the following sequence:

- IR source (transmitter) is used to emit radiation of required wavelength.
- This radiation reaches the object and is reflected back.
- The reflected radiation is detected by the IR receiver.
- The IR Receiver detected radiation is then further processed based on its intensity. Generally, IR Receiver output is small and amplifiers are used to amplify the detected signal.

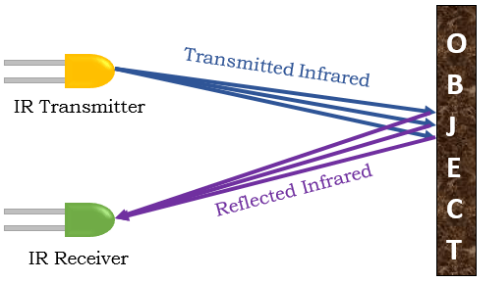


FIGURE 6:

WORKING OFINFRAREDSENSOR

4.3.2 Line following using IR Sensor

In this **project,** we have used IR Transmitters and IR receivers also called photo diodes. They are used for sending and receiving light. IR transmits infrared lights. When infrared rays falls on white surface, it's reflected back and catched by photodiodes which generates some voltage changes. When IR light falls on a black surface, light is absorb by the black surface and no rays are reflected back, thus photo diode does not receive any light or rays.

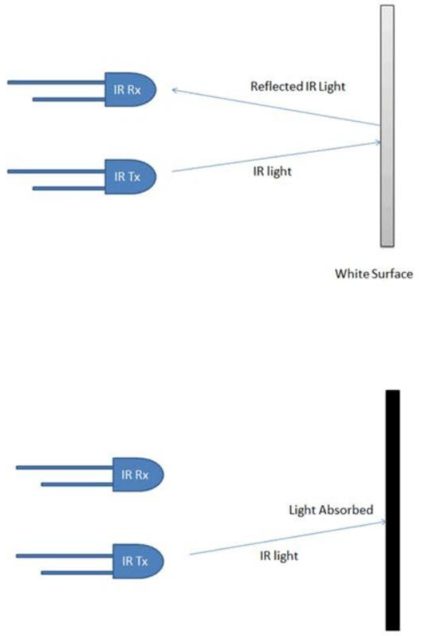


FIGURE 7: LINE FOLLOWER USING INFRAREDSENSOR

4.4 SOUND SENSOR

The sound sensor is one type of module used to notice the sound. Generally, this module is used to detect the intensity of sound. The applications of this module mainly include switch, security, as well as monitoring. The accuracy of this sensor can be changed for the ease of usage. This sensor employs a microphone to provide input to buffer, peak detector and an amplifier. This sensor notices a sound, & processes an o/p voltage signal to a microcontroller. After that, it executes required processing. This sensor is capable to determine noise levels within DB's or decibels at 3 kHz 6 kHz frequencies approximately wherever the human ear is sensitive. In smartphones, there is an android application namely decibel meter used to measure the sound level.

4.4.1 Pin Configuration

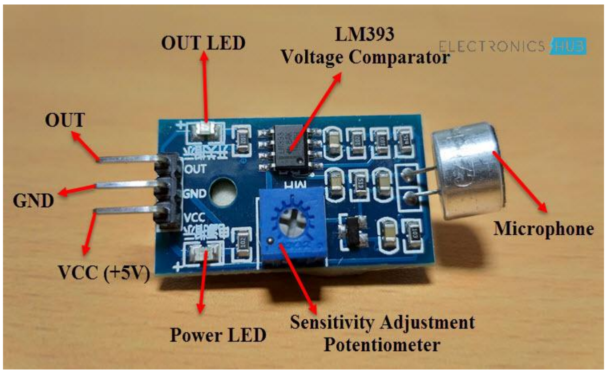


FIGURE 8: PIN CONFIGURATION OF SOUND SENSOR

- Pin1 (VCC): 3.3V DC to 5V DC
- Pin2 (GND): This is aground pin
- Pin3 (DO): This is an output pin

4.4.2 Working Principle

The working principle of this sensor is related to human ears. Because human eye includes a diaphragm and the main function of this diaphragm is, it uses the vibrations and changes into signals. Whereas in this sensor, it uses a microphone and the main function of this is, it uses the vibrations and changes into current otherwise voltage. Nowadays, a lot of security events are initiated due to some sort of sound which includes gunshots, aggressive behaviour, breaking the glass. But cameras with inbuilt sound exposure facilities can add huge value to the security system. Because they give an alert automatically when real and potential incidents occur. Then immediately they activate quick and appropriate actions to reduce the consequences.

4.4.3 Specifications

- The range of operating voltage is 3.⅗ V
- The operating current is 4~5 mA
- The voltage gain 26 dB ((V=6V, f=1kHz)
- The sensitivity of the microphone (1kHz) is 52 to 48 dB
- The impedance of the microphone is 2.2k Ohm
- The frequency of m microphone is16 to 20 kHz
- The signal to noise ratio is 54 dB

4.5 DC MOTOR and l293d DRIVER

A machine that converts DC electrical power into mechanical power is known as a Direct Current motor. DC motor working is based on the principle that when a current carrying conductor is placed in a magnetic field, the conductor experiences a mechanical force.

4.5.1 Working principle of a DC motor

An electric motor is an electrical machine which converts electrical energy into mechanical energy. The basic working principle of a DC motor is: "whenever a current carrying conductor is placed in a magnetic field, it experiences a mechanical force". The direction of this force is given by Fleming's left-hand rule and its magnitude is given by:

F = BIL

B = magnetic flux density,

I = current,

L = length of the conductor within the magnetic field.

4.5.2 L293D

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC.Dual H-bridge *Motor Driver integrated circuit (IC).*The l293d can drive small and quiet big motors as well, check the Voltage Specification at the end. It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. As you know voltage need to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, hence H-bridge IC are ideal for driving a DC motor. In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. Given below is the pin diagram of a L293D motor controller.

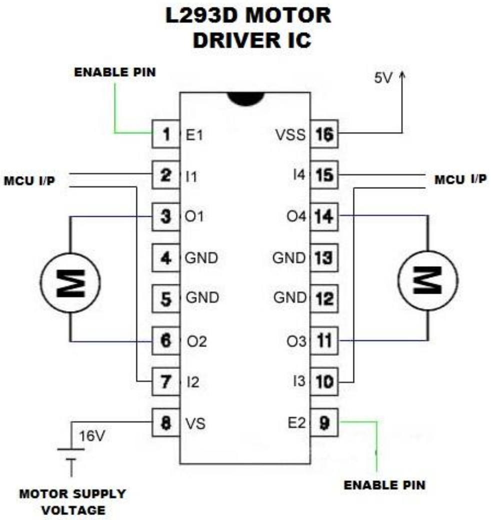


FIGURE 9: PIN CONFIGURATION OF L293d IC

4.5.2.1 Working of L293D IC

- L293D motor driver IC contains two H-bridge circuit inside it, which may also use Darlington transistor some times for current amplification.

- Let we understand this H-bridge circuit, in case 1, when logic '1' apply to transistor T1 and T4, motor starts rotating in clockwise direction due to circuit complete and current flows through it as shown by blue indication.

- In case 2, we apply logic '1' to transistor T2 and T3, so motor starts rotating in anti-clockwise direction due to circuit complete and direction of current flows through it is shown by green indication.

- As I mentioned that logic '1' apply to T1 & T4 or T2 & T3 is just for simplicity cause this transistor is hard wired inside IC so you just need to take care of pin-out and apply logic to IC's pin directly and carried out your work.

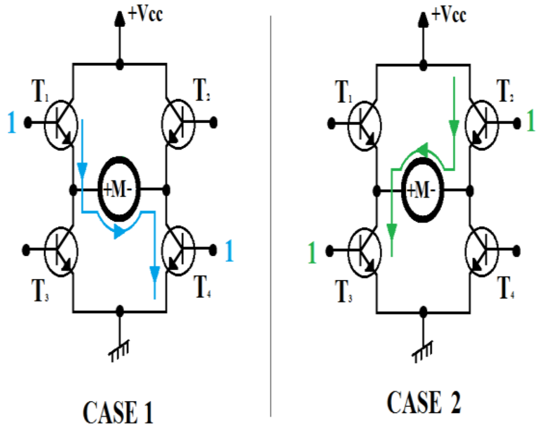


FIGURE 10: WORKING OF L293D IC

5. SOFTWARE USED

5.1 MQTT (Message Queuing Telemetry Transport)

5.1.1 Brief history

MQTT (**M**essage **Q**ueuing **T**elemetry **T**ransport) is a publish/subscribe messaging protocol that works on top of the TCP/IP protocol. The first version of the protocol was developed by Andy Stanford-Clark of IBM and Arlen Nipper of Cirrus Link in 1999. What makes MQTT faster than say sending HTTP requests with your IoT device is MQTT messages can be as small as 2 bytes, whereas HTTP requires headers which contains a lot of information that other devices might not care about. Also, if you have multiple devices waiting for a request with HTTP, you'll need to send a POST action to each client. With MQTT, when a server receives information from one client, it will automatically distribute that information to each of the interested clients.

5.1.2 Basics terms

- **Broker -** The broker is the server that distributes the information to the interested clients connected to the server.

- **Client** - The device that connects to broker to send or receive information.
- **Topic** - The name that the message is about. Clients publish, subscribe, or do both to a topic.
- **Publish** - Clients that send information to the broker to distribute to interested clients based on the topic name.
- **Subscribe** - Clients tell the broker which topic(s) they're interested in. When a client subscribes to a topic, any message published to the broker is distributed to the subscribers of that topic. Clients can also **unsubscribe** to stop receiving messages from the broker about that topic.
- **QoS** - Quality of Service. Each connection can specify a quality of service to the broker with an integer value ranging from 0-2. The QoS does not affect the handling of the TCP data transmissions, only between the MQTT clients. ***Note:*** *In the examples later on, we'll only be using QoS 0.*
- **0** specifies at most once, or once and only once without requiring an acknowledgment of delivery. This is often referred to as fire and forgets.
- **1** specifies at least once. The message is sent multiple times until an acknowledgment is received, known otherwise as acknowledged delivery.
- **2** specifies exactly once. The sender and receiver clients use a two level handshake to ensure only one copy of the message is received, known as assured delivery.

5.1.3 Working of MQTT

MQTT is a publish/subscribe messaging protocol. Clients will connect to the network, which can subscribe or publish to a topic. When a client publishes to a topic, the data is sent to the broker, which then is distributed to all the clients that are subscribed to that topic.

Topics are arranged in a directory-like structure. A topic might be **"LivingRoom"**, or **"LivingRoom/Light"** if you have multiple clients within that parent topic. The subscriber client will listen for incoming messages from the subscribed topic and react to what was published to that topic, such as **"on"** or **"off"**. Clients can subscribe to one topic and publish to another as well. If the client subscribes to **"LivingRoom/Light"**, it might also want to publish to another topic like **"LivingRoom/Light/State"** so that other clients can monitor the state of that light.

Now that we understand the theory of how MQTT works, lets build a quick and easy example with a Raspberry Pi and ESP32 Thing boards to see it working in action. We'll start by setting up the broker and running a quick test to make sure it's working correctly.

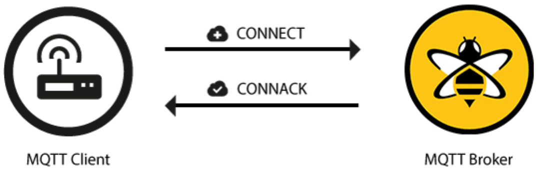


FIGURE 11: WORKING OF MQTT

5.1.4 MQTT protocol applications and use cases

Facebook currently uses MQTT for its Messenger app, not only because the protocol conserves battery power during mobile phone-to-phone messaging, but also because the protocol enables messages to be delivered efficiently in milliseconds (ms), despite inconsistent internet connections across the globe. Most major cloud services providers, including Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft Azure, support MQTT. MQTT is well suited to applications using M2M and IoT devices for purposes such as real-time analytics, preventative maintenance and monitoring in environments, including smart homes, healthcare, logistics, industry and manufacturing.

5.1.5 MQTT in IoT

MQTT is one of the most commonly used protocols concerning IoT. MQTT enables resource-constrained IoT devices to send, or publish, information about a given topic to a server that functions as an MQTT message broker. The broker then pushes the information out to those clients that have previously subscribed to the topic. To a human, a topic looks like a hierarchical file path. Clients can subscribe to a specific level of a topic's hierarchy or use a wild-card character to subscribe to multiple levels.

5.2 Raspberry pi software

5.2.1 Linux

Linux is a free open source operating system and it belongs to the Unix operating systems. Actually Linux means the kernel itself which is the heart of the operating system and handles the communication between the user and hardware. Normally Linux is used to refer to the whole Linux distribution. Linux distribution is a collection of software based on the Linux Kernel. It consists of the GNU-project's components and applications. Because Linux is an open source project, anyone can modify and distribute it. That is the reason why there are many variations of Linux distributions. Most popular distributions are Ubuntu, Red Hat Linux, Debian GNU/Linux and SuSe Linux.

5.2.2 Raspbian Wheezy

Raspbian Wheezy is a free operating system based on Debian distribution. It is created by a small team of developers who are fans of Raspberry Pi. Raspbian is optimized for the Raspberry Pi's hardware and it comes with over 35000 packages and pre-compiled software. Raspbian is still under active development and it aims to improve the stability and performance of the Debian packages.

Raspbian is officially recommended for beginners and it includes the graphical desktop environment called LXDE. Raspbian Wheezy is one of the fastest ways to setup and get the RasPi running.

5.2.3 Programming languages

There are considerable numbers of programming languages which have been adapted for Raspberry Pi. Python programming language is recommended by The Raspberry Pi foundation especially for the beginners. Basically any programming language which can be compiled for ARMv6 can run on the Raspberry Pi. Therefore the users are not restricted to use only the Python. On the Raspberry Pi there are preinstalled several languages for example C, C++, Java, Scratch and Ruby.

5.3 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is interpreted:** Python is processed at runtime by the interpreter.You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

5.3.1 Python Features

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearlydefined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable andcross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter. Thesemodules enable programmers to add to or customize their tools to be more efficient.
- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created andported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programsthan shell scripting.

5.3.2 Python's standard library

- Pandas
- Numpy
- Sklearn
- Seaborn
- Matplotlib
- Importing Datasets
- OpenCV

5.4 IMAGE PROCESSING USING OPENCV IN PYTHON

Image processing is the process of manipulating pixel data in order to make it suitable for computer vision applications or to make it suitable to present it to humans. For example, changing brightness or contrast is an image processing task which make the image visually pleasing for humans or suitable for further processing for a certain computer vision application.

5.4.1 Software and Hardware Requirements:

Hardware:

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB

Software:

- Python

5.4.2 Open CV

OpenCV is Open Computer Vision Library. It was initially launched in 1999 by Intel. With more updates, it has been modified since then to aim for the real-time computer vision. This library has been written under programming languages like C and C+. It can be easily runon operating systems Windows and Linux. This library can be easily interface with programing languages like Python, MATLAB, Ruby and others as well. Along with Numpy and Python image processing (shape &colour detection) can be performed at ease.

5.4.3 Read an Image

First, a sampleimage in which processing is to be applied is to be read. It's done usinga pre-defined Python function: CV2.imread (). The sample image should be available in current folder or the full location of the image is to be mentioned as an argument. For reading an image, we can use functions like imread_color, imread_grayscale, imread_unchanged.

5.4.4 PANDAS

Pandas is quite a game changer when it comes to analysing data with Python and it is one of the most preferred and widely used tools in data munging/wrangling if not THE most used one. Pandas is an open source. What's cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

Installation and Getting Started

- import pandas as pd
- import numpy as np

5.4.5 Viewing and Inspecting Data

Now that you've loaded your data, it's time to take a look. How does the data frame look? Running the name of the data frame would give you the entire table, but you can also get the first n rows with df.head(n) or the last n rows with df.tail(n). df.shape would give you the number of rows and columns. df.info() would give you the index, datatype and memory information. The command s.value_counts(dropna=False) would allow you to view unique values and counts for a series (like a column or a few columns). A very useful command is df.describe() which inputs summary statistics for numerical columns.

It is also possible to get statistics on the entire data frame or a series (a column etc):

- df.mean() Returns the mean of all columns
- df.corr() Returns the correlation between columns in a data frame
- df.count() Returns the number of non-null values in each data frame column
- df.max()Returns the highest value in each column
- df.min()Returns the lowest value in each column
- df.median()Returns the median of each column
- df.std()Returns the standard deviation of each column

5.4.6 Selection of Data

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary. You can select a column (df[col]) and return column with label col as Series or a few columns (df[[col1, col2]]) and returns columns as a new DataFrame. You can select by position (s.iloc[0]), or by index (s.loc['index_one']) .

5.4.7 Filter, Sort and Group by

You can use different conditions to filter columns. For example, df[df[year] > 1984] would give you only the column year is greater than 1984. You can use & (and) or

| (or) to add different conditions to your filtering. This is also called boolean filtering.

It is possible to sort values in a certain column in an ascending order using df.sort_values(col1) ; and also in a descending order using df.sort_values(col2,ascending=False). Furthermore, it's possible to sort values by col1 in ascending order then col2 in descending order by using df.sort_values([col1,col2],ascending=[True,False]).

The last command in this section is groupby. It involves splitting the data into groups based on some criteria, applying a function to each group independently and combining the results into a data structure. df.groupby(col) returns a groupby object for values from one column while df.groupby([col1,col2]) returns a groupby object for values from multiple columns.

5.4.8 Data Visualization in Python

50-58, 2016 Data visualization is the discipline of trying to understand data by placing it in a visual context, so that patterns, trends and correlations that might not otherwise be detected can be exposed. Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has a excellent library for you.

To get a little overview here are a few popular plotting libraries:

- **Matplotlib:** low level, provides lots of freedom
- **Pandas Visualization:** easy to use interface, built on Matplotlib
- **Seaborn:** high-level interface, great default styles
- **ggplot:** based on R's ggplot2

6. CONCLUSION

According to this system, whole area surveillance is done using the night vision camera and also automatic system when the sound is detected robot will follow the particular path and go to that area and capture the area and send using IOT. This system is an automatic smart way for night vision patrolling

REFERENCES

[1] "Review of human detection techniques in night vision" in 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)

[2] Eun Som Jeon et al., "Human detection based on the generation of a background image by using a far-infrared light camera", *Sensors*, vol. 15, no. 3, pp. 6763-6788, 2015.

[3] Dushyant Kumar Singh, Dharmender Singh Kushwaha, "automatic intruder combat system: a way to smart border surveillance", *Defence Science Journal*, vol. 67, no. 1, pp