

Internet Controlled Techrobot using Raspberry Pi

Archit Prakash Raut¹, Aayesh Raj² and Shalvi Patil³

Department of Electronics and Telecommunication
Engineering

Vidyavardhini's College of Engineering and Technology
Vasai, India

Asst. Prof. Ashwini Katkar⁴

Department of Electronics and Telecommunication
Engineering

Vidyavardhini's College of Engineering and Technology
Vasai, India

Abstract :- A robot is a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer. Robots are playing important role in today's industrial automation and monitoring systems. As technology developed, the robots have increased their applications and functionality. Working robots will co-operate with the humans making their work effortless and less complicated. In this research project, we are working on the design and implementation of robot for Home automation. Internet technology provides a good way for us to develop an integrated computing network environment for the applications of different robotic systems. This robotic vehicle will be controlled via internet i.e. from anywhere in the world. It also describes the use of the camera mounted on the robot that can wirelessly transmit real time video fed onto the PYTHON page using Wi-Fi technology. The application allows the robot control interactions with the help of Graphical User Interface (GUI). Here we are using robotic system whose working is based on Raspberry pi, which will go into those areas and provide us with the videos of those locations. The camera mounted on the robot will keep on capturing the videos from the surroundings to keep a record of the details of the incident happened and this is readily available to user and only the authenticated users can see the recorded details. It also describes the use of obstacle detector mounted on the robot. In the presence of obstacle, the robot will stop and take turns. In presence of metal it will give buzz sounds. In darkness, it will turn on the LED'S to show path.

Keywords :- *Graphical User Interface (GUI), Light Emitting diodes (LED's), Wireless Fidelity (Wi-Fi).*

I. INTRODUCTION

The quick development and growth of industry and advancement of technology have diminished the human efforts, the fundamental reason for which being machines!! Machines play an important role in our life. A machine might be anything, be it a mobile phone or a bike or even robot. Robots have found an increasing demand in a wide range of applications in our life. Their use in defense has expanded. Robots ensemble human beings in various ways be it looks or functioning, but previously robots were not controlled by computer programs or electronic circuit. Back then they were constructed using principle of mechanics enhancing over time with the coming of electronic age. In today's world, robots find use in various places be it to detect buried bombs or in industrial applications. A robot is a mechanical or virtual artificial agent, usually an electro-mechanical machine that is guided by a PC program or electronic circuit, and thus a type of an embedded system. The branch of technology that deals with the design, construction, operation, and application of robots as well as PC systems for their control, sensory feedback, and data

processing is robotics. These technologies deal with automated machines that can take the place of humans in dangerous ambiance or manufacturing processes, or resemble humans in appearance, conduct. The complexity of computer software is based on how difficult the robot's tasks are. In this project we make use of internet to establish communication between the user and a robotic vehicle. This is a reliable connection and a live video feedback is available to control the robotic vehicle. Due to the use of internet, there is no limitation on range or separation distance between the user and the robotic vehicle. Internet of Things has opened up a completely new range of real-world applications namely tele-surgery, tele-manufacturing, tele-training, traffic control, surveillance, health care-space exploration, calamity rescue etc. and the list is supposed to increase further in the upcoming years. Surveillance is the way toward observing a circumstance, a range or a man. This generally occurs in a military scenario where surveillance on borders and territory of enemy is essential for nations well being. Human surveillance is achieved by deploying personnel near sensitive areas in order to continuously monitor for changes. But humans do have their limitations, and deployment in inaccessible places is not always feasible. There are also added risks of losing them in the event of getting caught by the enemy. With advances in technology over the years, however, it is possibly to remotely monitor areas of importance by using robots in place of humans. Apart from the advantage of not having to risk any human, terrestrial and aerial robots can also pick up details that are not obvious to humans. By furnishing them with high determination cameras and different sensors, it is conceivable to acquire data about the particular zone remotely. Satellite communication makes it possible to communicate seamlessly with the robots and obtains real-time audiovisual feedback. Thus, in recent times, surveillance has become an area of great research interest. Thus, it is our aim to build a full-featured smart surveillance robot using a Raspberry pi which can be remotely controlled over the internet. With the popularity and across the board utilization of internet, it becomes a simple and easy task for anyone to control the robotic vehicle from a remote end. Most people spend their time on internet than the average time they sleep. So by making the robotic vehicle to be controlled from internet, it becomes easy for anybody to use it from a remote end.

1.1 System Overview

In this project, control of robotic vehicle is from remote end with the use of Internet and also we are able to get the videos from the robot end for the purpose of surveillance. At the client side we will be able to see the live video and control the movement of robot and control the camera directions as well. DC motors are being used for the movement of robotic wheels and camera movement i.e. 360° rotation. Raspberry Pi 3 model B is used for video streaming and sending the streamed video to user PC with the help Internet. The use of Internet does not bring the limitation of range into consideration as if we have the internet access, we can control the robot from anywhere. In this implementation of robot, the web camera connected to the Raspberry pi keeps on recording what is going on there at the host place and saves it into the pi. This project provides or controls the surveillance robot using wireless technology. It also provides the live video obtaining from the robot. The robot is based on Raspberry pi. The robot is programmed to be controlled using the Web-page.

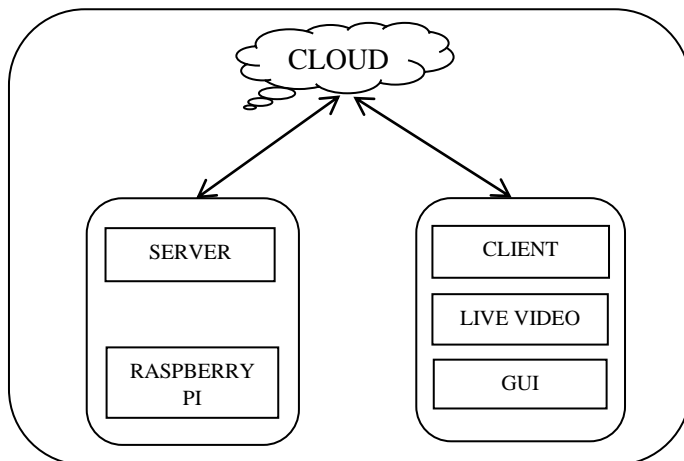


Figure 1.1. System Overview

We have created a mobile robotic platform. The project utilizes Raspberry pi 3 model B to command all the actions of the robot. A web-cam is mounted on the robot, which will give live video-streaming. An obstacle avoidance module and a metal buzzer is used to detect an obstacle and avoid it. LDR is used to give glow LED's to give light in darkness. The robot can climb up and climb down stairs. The raspberry pi hosts the cloud server. It broadcasts the information to the user and the user then controls the motion of the robot through web-page hosted on the server.

1.2 Advantages

The existing surveillance systems are stationary and cannot be used for different critical applications and situations. The controlling of the surveillance system is very tedious, time consuming and out dated.

- It can be used as a portable Surveillance System.
- It can be controlled remotely from anywhere in the world due to Internet of Things (IOT).
- It does not require Man Power.
- It can be used in critical situations like flood & bomb disposal

1.3 Applications

It can be used for following purposes:

- 1) *Home Automation*: This mini robot can be used as a home assistant. Working parents can look after their children at home from outside. This can also be used for home security as well. Also, we can control home appliances like lights, fans, A.C, etc from anywhere using internet. This would be possible due to IoT.
- 2) *Surveillance Robot*: The Robot can be used at workplaces, markets and banks for their surveillance in specific positions as it is mobile in nature and has greater adaptability than security cameras.
- 3) *Rescue scouting*: The robot can be used in any calamity disaster case for scouting of distressed regions. It can supply live video feed going to places inaccessible for humans.
- 4) *Military Spy bot*: The robot can be used for spying and surveying reported places especially in War field. This can be utilized for live feed from any region needed to be looked out for.

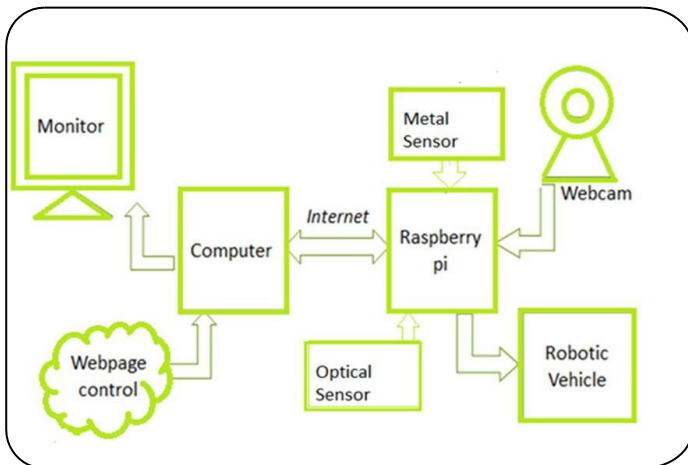
II. PROBLEM DEFINITION

This research project deals with the design and implementation of internet controlled robot using Raspberry Pi. It uses wireless technology to provide essential security using surveillance system and also object avoidance via internet from anywhere. The proposed project will capture information and transmits it via a Wi-Fi to a static IP, which is viewed using a web browser from any smart device. Raspberry Pi controls a video camera for surveillance. It streams live video and records it. This robotic vehicle will be controlled via internet from anywhere in the world. The Raspberry Pi will host the cloud server. The data will be sent to the cloud and the information is broadcast to the user. The user then controls the motion of the robot through a web-page hosted by the server. The major entities are Cloud server, GUI and client. The whole essence of the project is to establish communication between these three parties. The server hosted on the pi connects the robot to the world. The web-page i.e. GUI acts as an interface for controlling the Raspberry Pi. The user accesses the servers IP address and controls robot through various buttons on the GUI. When the buttons are clicked they send request to the server which sends control signal to robot.

II.1 METHODOLOGY

The web-cam will capture live data with regards to its surroundings and then send it to a desired device through internet. The user will be observing this data on the monitor at the user end. According to the desired movement, the user will control the robotic vehicle through the web-page available at the user end. If there is any obstacle in between it will stop. In darkness LED's will glow to show path. The input given through the web-page is then sent through the internet and the desired movement occurs at the robot end.

II.2 Block Diagram



The Robotic vehicle is constructed using Raspberry Pi, motors, motors driver L239d, optical sensor, metal buzzer, LDR, web-cam. Fig 3.2 is the block diagram of the project. With the help of server an internet connection is established between the robotic vehicle and the user. The Robotic vehicle is the server side. Raspberry Pi hosts the server. User is the client side. The robot captures the images and video using a web-cam and stores them into the memory. The next task is to capture and send live images using internet at a rate sufficient to make them seem like a live video to the human eye. DC motors are being used for the movement of robotic wheels and camera movement i.e. 360° rotation. This was initially implemented using LAN before moving to internet. Using internet, range of operation is not a limitation. We can control the robot from anywhere in the world. Also, it has obstacle sensor and metal sensor connected to Raspberry Pi. If any obstacle is present in between then the robot will stop and any detection of metal, the buzzer will be on. In darkness LED's will turn on.

III. IMPLEMENTATION

We implemented our prototype situated with three major components like on board camera, voltage regulator and a L293D motor driver. The real-time video and all other sensor data's are displayed in the web-page which can be viewed from anywhere in the world by setting a specific internet address. The Raspberry Pi doesn't have a hard disk or enough internal memory capacity, so we need to use an SD card for the data storage and for all other operations of Pi. First, an operating system has to be installed in Raspberry Pi. Linux based OS is the preferred for almost all single board computers and the scenario doesn't change with Raspberry Pi. Popular OS for Pi are Raspbian, Debian, Arch Linux, RISC OS, etc. The debian distribution based Raspbian is much preferred than any other. It's installed onto an SD card. A mouse and keyboard are required to work with it, and with the 2 USB ports available, it's possible to connect the mouse and keyboard to the system. In order to plug- in another USB device has to Pi, a powered

USB hub must be used because the device will draw power from the Pi to work and this will cause Pi to hang. The external powered hub can provide the necessary power for Pi to operate, it will be good to use an external adapter of specified rating. After the initial boot up, configurations have to be changed to suit our needs like initial boot up to the desktop, etc. The Pi is connected to a network and the IPV4-address assigned to the system need to be found using the command 'sudo ifconfig'. The address obtained will be the DHCP address, it's possible to change it to a static IP by editing the interface file present in /etc file. Then update the software and kernel as required to make the system up to date. The web page can be designed using a suitable editor like nano or leafpad. The default directory for the web server is /var/www. It can be changed later in the Lighttpd configuration file. Python is the preferred language for Raspberry Pi, but it also supports Java, C, Ruby, etc. There are 5 motors for the robotic arm, so 3 motor driver ICs are required for the control of all the joints of the robotic arm. A triple L293 driver board is designed for the control of all the motors and is interfaced with the GPIO pins of Raspberry Pi. Python programs are written for the forward and backward motion of each joint of the arm and are linked with the PHP script.

Step1: Setting up the raspberry pi. In this step go to raspberrypi.org and download any operating system you required but here we used Raspbian Wheezy.

Step 2: Install the required packages in the pi and connect the webcam to the pi, after this you will get MJPG-streamer folder

Step 3: Now design the GUI that provides a way to control our robot. This webpage is designed based on python.

Step 4: Now, we need Wi-Fi dongle to connect our Pi with Wi-Fi router. After connecting Wi-Fi dongle to PI, open WiConfig application & connect your PI with your Wi-Fi router. If it is already connected with Wi-Fi router, execute following command into terminal to know IP Address.

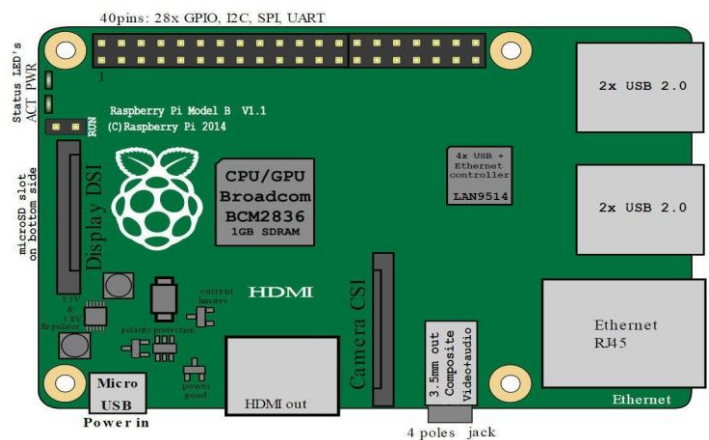


Fig.1 PCB Preview of Raspberry Pi model B

Raspberry Pi 3 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 2
29/02/2016
www.element14.com/RaspberryPi

Fig.2 Raspberry-Pi pins

Step 5: You will find IP address of Pi in output. Do remember this IP Address for further use. We will need it control your Robot.

Step 6: Now build Robot, We can use DC Motor based simple robot. To control your motors we need L293D IC. Here we connected GPIO pins 18, 4 with L293D IC to control Left Motor & 23, 24 pins with L293D IC to control Right Motor. (Refer Fig.3).

Step 7: Power Supply: Now the biggest challenge for any autonomous machine is power supply. Here we used 12v rechargeable Ni-MH battery. But as per specifications Pi will work on 5 V, So we use LM317 to regulate it to 5v. We connect battery terminals directly to motor driver IC, connect it to input of LM317 IC and regulated it to 5v by adjusting the resistance. 5 V output of this IC is connected with First GPIO of Raspberry Pi in to power up it.

Step 8: After connecting all the connections check it once again because if any wrong connections happened then definitely pi will be burn and see that you are giving power supply correctly that is in between 4.9v-5.2v

Step 9: As soon as you connect 5V supply with Pi it will turn on, you can see green LED blinking while start-up process. After some time open browser in your Laptop and write down following link: IP address / file-name.

Step 10: Connect your Raspberry pi to cloud server. The Raspberry Pi hosts the cloud server. On any remote PC, access the Raspberry Pi and now sitting in your room you can stream video and control the robot anywhere in the world.

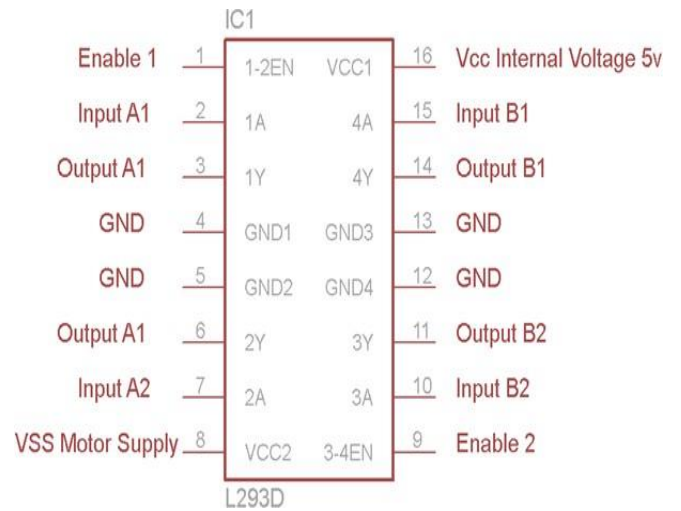


Fig.3 Pin Diagram L293D

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The Robotic vehicle has been designed and constructed. This is the final look of the robotic vehicle we designed.



Fig.4 Robotic Vehicle

The Robotic vehicle is constructed using Raspberry pi, motors, motors driver L239d, optical sensor, metal buzzer, LDR, webcam. The robot is controlled and monitored using the webpage's designed. The webpage is designed in Python language. The webpage created provides the user for controlling the robot and also the live streaming.

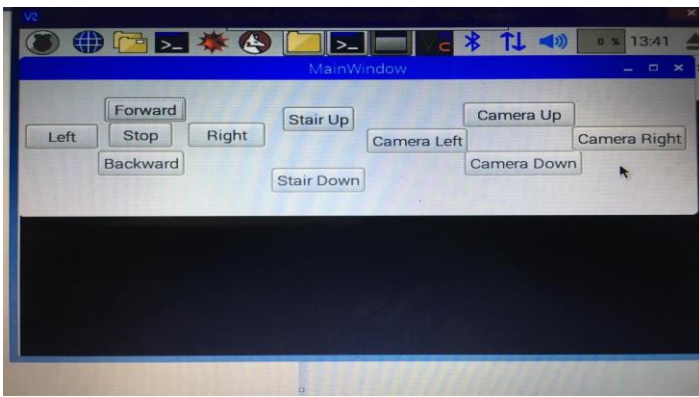


Fig.5 Graphical User Interface (GUI)

The webpage has the following buttons: Forward, Backward, Right, Left and Stop to control the motion of the robot. Stair up and Stair down for the robotic vehicle to climb up and down stairs. Camera up, Camera down, Camera right and Camera left to give camera 360° rotation.

Using the command 'sudo ./cam.exe' in command prompt of Raspberry pi, we get the live video feed.

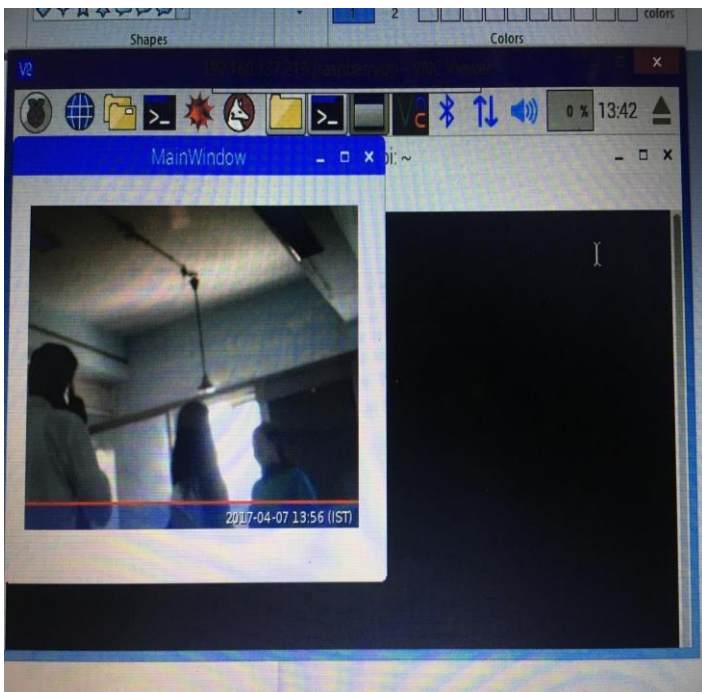


Fig.6 Live video stream

V. CONCLUSION AND FUTURE WORK

Using this robotic system a remote area can be monitored easily from remote end. One can easily monitor as well as control the activity of the robotic unit. This system can be used in any conditions and areas where it is difficult for the security forces to reach. As the communication is done with the help of internet so limitation of range of operation does not arise and thus we can monitor any area from anywhere in the world. We control the robot using GUI created in Python.

We can use this system for military applications installing suitable sensors. It can also be used in home automation. Just by changing the robotic unit design we can use it in hospitals for patient monitoring. Using some chemical sensors we can detect harmful gas leakage in the chamber the time delay which occurs in the execution of commands can be reduced and thus we can have more real time access to the robot. With reduced time delay we can have faster operation and quick response to any illegal activities in the monitored area. Also it can be used as a spy robot. The robot is very economical and mobile.

VI. ACKNOWLEDGEMENT.

I take this opportunity to express my regards and sincere thanks to my advisor and guide Prof. Ashwini Katkar, without whose support, this project would not have been possible. Her constant encouragement and moral support gave me the motivation to carry out the project successfully. I am also indebted to her for her valuable and timely guidance. The discussions with her helped a lot in developing an in-depth understanding of the topics involved. Also, I would like to thank the Lab In charge, who helped me with the lab facilities whenever I needed them.

VII. REFERENCE

- [1] Saucy and F. Mondana, "Keep On The Web: Open access to a mobile robot on the Internet", IEEE Robotics and Automation Magazine, pages 41-47, March 2000.
- [2] R.C. Luo, T.M. Chen, and C.C. Yih, "Intelligent autonomous mobile robot control through the Internet," IEEE International Symposium ISIE 2000, vol. 1, pp. 6-11, December 2000
- [3] Hu, Lixiang Yu, Pui Wo Tsui, Quan Zhou, "Internet-based Robotic Systems for Teleoperation", International Journal of Assembly Automation, Vol. 21, No. 2.
- [4] Robot using Zigbee Communication By Krishnaswamy Kannan and Gowtham S, International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462 Vol. 4 No.10 October 2012

APPENDIX
PYTHONCODE

```
#!/usr/bin/python
import RPi.GPIO as GPIO

import time

import sys

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(14,GPIO.OUT) # left
GPIO.setup(15,GPIO.OUT) # left
GPIO.setup(18,GPIO.OUT) # right
GPIO.setup(23,GPIO.OUT) # right
GPIO.setup(20,GPIO.OUT) # camera up down
GPIO.setup(21,GPIO.OUT) # camera up down
GPIO.setup(12,GPIO.OUT) # camera left right
GPIO.setup(16,GPIO.OUT) # caemra left right
GPIO.setup(24,GPIO.OUT) # stair case
GPIO.setup(25,GPIO.OUT) # stair case

GPIO.output(14,GPIO.LOW) # left
GPIO.output(15,GPIO.LOW) # left
GPIO.output(18,GPIO.LOW) # right
GPIO.output(23,GPIO.LOW) # right
GPIO.output(20,GPIO.LOW) # camera up down
GPIO.output(21,GPIO.LOW) # camera up down
GPIO.output(12,GPIO.LOW) # camera left right
GPIO.output(16,GPIO.LOW) # caemra left right
GPIO.output(24,GPIO.LOW) # stair case

GPIO.output(25,GPIO.LOW) # stair case

GPIO.setup(17,
pull_up_down=GPIO.PUD_DOWN) #obs

GPIO.IN,

GPIO.setup(27,
pull_up_down=GPIO.PUD_DOWN) #obs

GPIO.IN,

data = sys.argv[1]

print (sys.argv[1])

if (data == "m_forward"):

print("Motion: Forward")

GPIO.output(14,GPIO.HIGH) # left
GPIO.output(15,GPIO.LOW) # left
GPIO.output(18,GPIO.HIGH) # right
GPIO.output(23,GPIO.LOW) # right

elif (data == "m_backward"):

print("Motion: Backward")

GPIO.output(15,GPIO.HIGH) # left
GPIO.output(14,GPIO.LOW) # left
GPIO.output(23,GPIO.HIGH) # right
GPIO.output(18,GPIO.LOW) # right

elif (data == "m_left"):

print("Motion: Left")

GPIO.output(14,GPIO.HIGH) # left
GPIO.output(15,GPIO.LOW) # left
GPIO.output(23,GPIO.HIGH) # right
GPIO.output(18,GPIO.LOW) # right

elif (data == "m_right"):

print("Motion: Right")

P.T.O

GPIO.output(15,GPIO.HIGH) # left
GPIO.output(14,GPIO.LOW) # left
```

```
GPIO.output(18,GPIO.HIGH)    # right
GPIO.output(23,GPIO.LOW)    # right
elif (data == "m_stop"):
print("Motion: Stop")
GPIO.output(14,GPIO.LOW)    # left
GPIO.output(15,GPIO.LOW)    # left
GPIO.output(18,GPIO.LOW)    # right
GPIO.output(23,GPIO.LOW)    # right
GPIO.output(20,GPIO.LOW)    # camera up down
GPIO.output(21,GPIO.LOW)    # camera up down
GPIO.output(12,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.LOW)    # caemra left right
GPIO.output(24,GPIO.LOW)    # stair case
GPIO.output(25,GPIO.LOW)    # stair case
elif (data == "m_camup"):
print("Motion: Camera Up")
GPIO.output(20,GPIO.HIGH)    # camera up down
GPIO.output(21,GPIO.LOW)    # camera up down
time.sleep(0.3)
GPIO.output(20,GPIO.LOW)    # camera up down
GPIO.output(21,GPIO.LOW)    # camera up down
elif (data == "m_camdn"):
print("Motion: Camera Down")
GPIO.output(20,GPIO.LOW)    # camera up down
GPIO.output(21,GPIO.HIGH)    # camera up down
time.sleep(0.3)
GPIO.output(20,GPIO.LOW)    # camera up down
GPIO.output(21,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.LOW)    # caemra left right
GPIO.output(12,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.HIGH)    # caemra left right
time.sleep(0.3)
GPIO.output(12,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.HIGH)    # caemra left right
time.sleep(0.3)
GPIO.output(12,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.LOW)    # caemra left right
elif (data == "m_camright"):
print("Motion: Camera Right")
GPIO.output(12,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.HIGH)    # caemra left right
time.sleep(0.3)
GPIO.output(12,GPIO.LOW)    # camera left right
GPIO.output(16,GPIO.LOW)    # caemra left right
elif (data == "m_stair_up"):
print("Motion: Stair Up")
GPIO.output(24,GPIO.HIGH)    # stair case
GPIO.output(25,GPIO.LOW)    # stair case
time.sleep(1)
GPIO.output(24,GPIO.LOW)    # stair case
GPIO.output(25,GPIO.LOW)    # stair case
elif (data == "m_stair_dn"):
print("Motion: Stop")
GPIO.output(24,GPIO.LOW)    # stair case
GPIO.output(25,GPIO.HIGH)    # stair case
time.sleep(1)
GPIO.output(24,GPIO.LOW)    # stair case
GPIO.output(25,GPIO.LOW)    # stair case
```