TABLE OF CONTENT

| SR.NO | TITLE | PAGE NO |
| --- | --- | --- |
| 1 | PROJECT OVERVIEW  1.1 Introduction To System |  |
| 2 | REVIEW OF LITERATURE |  |
| 3 | REQUIREMENT ANALYSIS |  |
| 4 | PROJECT DESIGN |  |
| 5 | IMPLEMENTATION DETAILS |  |
| 6 | RESULTS AND DISCUSSIONS |  |
| 7 | PROJECT TIME LINE |  |
| 8 | TASK DISTRIBUTION |  |
| 9 | CONCLUSION AND FUTURE SCOPE |  |
| 10 | REFERENCES |  |

Project Overview

1.1 Introduction

Sometimes we drop small things like diamond rings, pens or valuable stuff under hidden places like under bed, under table or places that we can not reach. If we can have some mechanism that can show us where’s the lost stuff would helps a lot. That’s why I proposed this SPY robot that can explore certain places that human couldn’t reach.

SPY is a robot that is very tiny (about 5 in cube) and capable of navigating some specific places. It will explore toward pointed direction, avoid obstacles and transmit images that it “sees”. When the specified location is reached, it will return to its home. While it is exploring, it will transmit live video through its wireless (RF) video camera to home base. If the place that user assigned is in low light condition, robot will turn on its light in order to capture images. When robot found some obstacles on the path, it will turn to other direction to avoid collision and continue its work. When robot reach destination, it will show user that the task has been done by turn its head toward home direction and then blinking its LED.

The advent of new high-speed technology and the growing computer Capacity provided realistic opportunity for new robot controls and realisation of new methods of control theory. This technical improvement together with the need for high performance robots created faster, more accurate and more intelligent robots using new robots control devices, new drivers and advanced control algorithms.

This project describes a new economical solution of robot control systems .In general; the robots are controlled through wired network. The programming of the robot takes time if there is any change in the project the reprogramming has to be done. Thus they are not user friendly and worked along with the user preferences. To make a robot user-friendly and to get the multimedia tone in the control of the robot, they are designed to make user commanded work. The modern technology has to be implemented to do this.

For implementing the modern technology it should be known by all the users to make use of it. To reach and to full-fill all these needs we are using android mobile as a multimedia, user friendly device to control the robot. This idea is the motivation for this project and the main theme of the project. In this modern environment everybody uses smart phones which are a part of their day-to-day life. They use all their daily uses like newspaper reading, daily updates, social networking, and all the apps like home automation control, vehicle security, human body anatomy, health maintenance, etc has been designed in the form of applications which can be easily installed in their hand held smart phones. This project approached a robotic movement control trough the smart phones.

Hence a dedicated application is created to control an embedded robotic hardware. The application controls the movement of the robot.. A wired camera is mounted on the robot body for spying purpose.

1.1.1 Motivation

1.1.2 Advantages Over Current System

In various areas there is a need of constant surveillance. The current surveillance system includes monitoring by using CCTV cameras and other monitoring system. Mostly these systems are stationary and they can cover a limited area. These systems are mostly control manually or through a computer. They cannot be used to cover a larger area as well as they cannot be controlled using any mobile device. In short we can say that these systems which is more dynamic and can be controlled remotely.

This project is aimed at developing a surveillance system which can be controlled remotely by using an Android App. It includes a robot with a Wired Camera attach to it. This robot captures the high resolution video feed and transmits it to the connected Android device which is used to control the robot.

1.2 Proposed System Architecture

(need to add raspberry pi and how bluetooth attach to it)

Once connection is successfully established, the user can send controlling commands through the GUI of the android application. The command sent from the mobile application is received by the Bluetooth module which transmits it through serial communication to the raspberry pi

System architecture is shown in fig

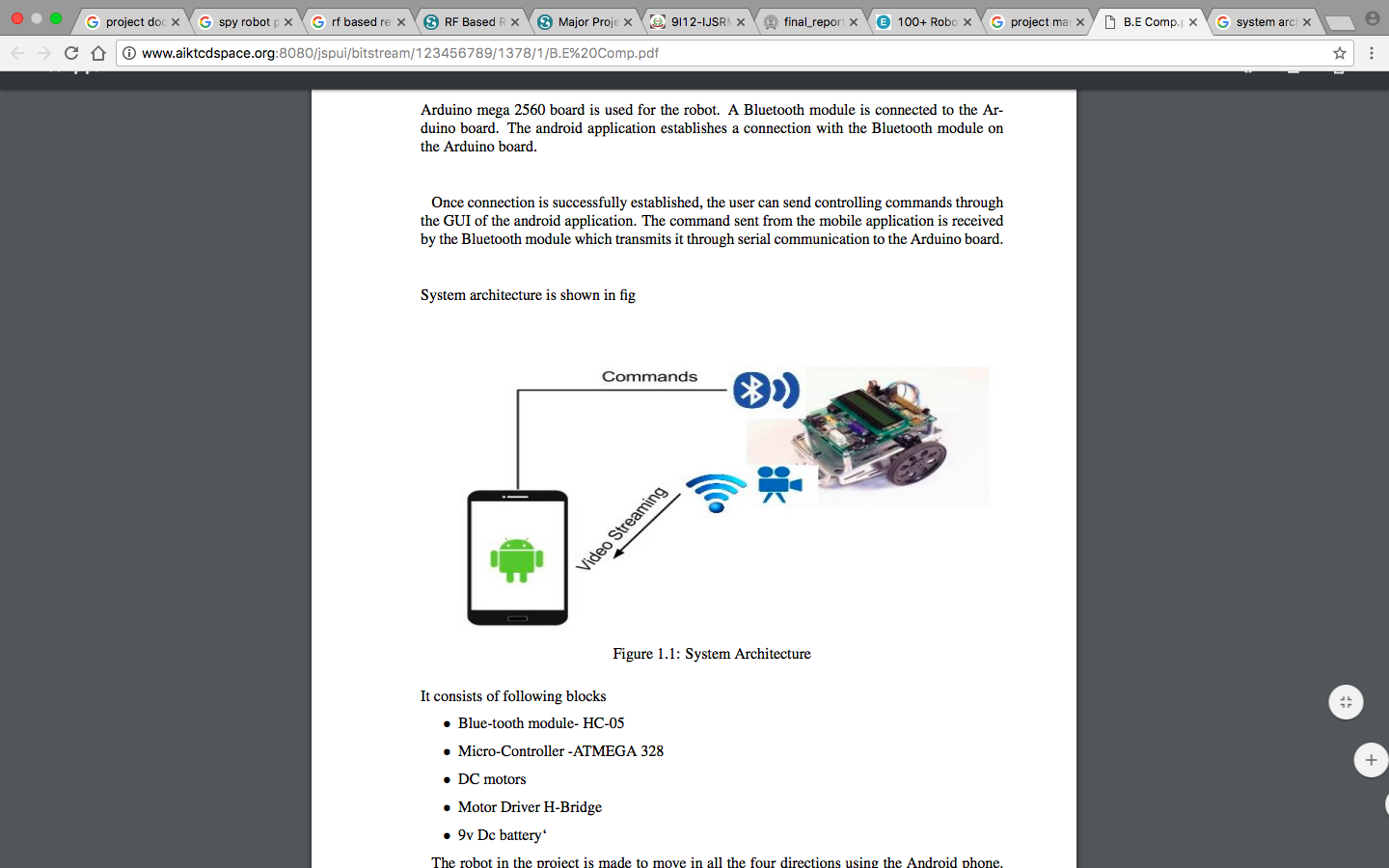


Figure 1.1: System Architecture

It consist of following blocks

* Blue-tooth module- HC-05
* Micro-Controller -(need to be filled)
* DC motors
* Motor Driver H-Bridge
* v Dc battery‘

(left to write)

1.2.1 Formulation of Problem With using Technology

1.2.1.1 Why Raspberry pi?

1.3 Organization of the Project

Chapter 2 contain Review of Literature. in this chapter we have studied and reviewed the previous work done on the topics related to our project. We did a detailed study on five such papers published in international journals. We have explained the pros, cons and the way we overcome the limitation of that project.

Chapter 3 contains Requirement analysis. In this chapter we give the detailed information on all the requirements of our project. It contains all the software and hardware related requiremets.

Chapter 4 contains the Project design. In this chapter we explain the project diagramatically. All the diagrams are included in this chapter along with the explaination.

Chapter 5 contains the Implentation details. In this chapter all the details about the implementation on the project is given. It includes the assumptions and dependencied and implementation methodology as well. It aslo includes the detailed analysis and the description of project.

Chapter 6 contains the Results and Discussion. In this chapter we give all the test cases and the related results and discussion related to in.

Chapter 7 contains the Project time line. It includes the detailed advancement of the project in the for of a matrix time line and timeline chart.

Chapter 8 contains the task distribution. It gives a detailed explaination of the distribution on the workload.

Chapter 9 contains the Conclusion and Future scope. In this chapter we give the conclusion of our project and the future scope of the project.

Review Of Literature

2.1 What is Surveillance robot ?

Surveillance robot is the robot used for the surveillance purpose. The remote areas are watched using the surveillance robots.

2.2 Wireless Controlled Surveillance Robot

2.2.1 Description

A mobile robot is a machine that is basically place or mounted on a movable platform and can be with the help of certain instructions. In todayâTMs world a lot of fields use mobile robots. Many of the complex robots that we now see have originated from the simpler mobile robots. This technology has increased many new applications in the industry.[4] The combination of mobile devices and robots are leading to new ideas in lots of fields.

The mobile devices are now being used in many of the industrial applications this is mainly

because of the reason that they are portable and have a longer battery life as compared to a laptop. Also they have a data plan through a cell phone carrier which is convenient as we can interact with the mobile robot once the connection is established.

Mobile Robots: The mobile robots can be classified into different types. The track robot is the robot that uses tracks to move around. However such robots are costly to build.[10] Also they are not as flexible as the wheeled robots. The wheeled robots are the robots which use wheels for moving. Such robots can move only on smooth flat surfaces. The third type is the legged robots which are based on human form. They have legs which helps them to move around. These robots are very difficult to design.

Proposed System

The new age of technology such as Android, GSM has redefined communication. Most people nowadays have access to mobile phones and thus the world indeed has become a global village. At any given moment, any particular individual can be contacted with the mobile phone. New innovations and ideas can be generated from it that can further enhance its capabilities.

Technologies such as Infra-red, Bluetooth, Wi-Fi which has developed in recent years goes to show the very fact that improvements are in fact possible and these improvements have eased our life and the way we live.Remote management of several home and office appliances is a subject of growing interest and in recent years we have seen many systems providing such controls.

Mobile robots are robots which have the ability to move around and interact with their environment and not just hinged to a particular place. There are many labs and research groups from various universities and industries which are completely dedicated on researching mobile robots, because of their immense potential and varied application in industry, military, security, and entertainment.

The robot is specially designed for surveillance purpose. The control mechanism is provided along with video transmission facility. The video transmission is practically achieved through high speed image transmission. Initially, the robot will be equipped with an Android smartphone which will capture the scenario in front of it will transfer the images to the server on which the user will be controlling and watching the live feed.

2.2.2 Pros

1. Infrared LED: 8pcs infrared LED, automatic operate in dark environment Resolution: VGA (640x480)/ QVGA(320x240)/ QQVGA(160x120)
2. Motion detection to trigger alarm

2.2.3 Cons

1. Limited Frequency Range: The frequency range used for typical RF communication is near about 3KHz-3GHz. The use of channel separator increases the reliability but decreases the actual usable working frequency range.
2. Limited Functions: The limited number of channels causes less number of combinations and thus there are less numbers of available functions.

3. Limited Working Range: The working range of RF circuits with transmitters and receiver is very small. It starts from a few meters to a few kilometres. The working varies from circuits to circuits, but mainly depends on the values of physical components used in the circuit. Mainly Wi-Fi and Wi-Max wireless services are used in RF transmitter and receiver circuits. The following table shows the actual working range of different wireless standards that can be used in wireless communication.

4. Reliability of Operation: The RF circuits are very prone to errors due to external conditions such as EMI (Electro-Magnetic Interference), medium saturation, absorption due to repetitive reflections from surface. Hence the output recovered is not always what is expected. This might be a serious problem when working with scientific experimental components.

5. Security reasons: This is the main disadvantage of using a RF circuit and the main reason why RF circuits are not preferred today. The RF frequency band is available for almost all the users for data communication. So there might be a scenario where more than one user is trying to accommodate channel for its own communication. In such case the frequency band may get interference from another user. Or worst case would be, some user intentionally trying to jam our communication network. The RF jammer circuits are very easy to design; hence the question of security arises when RF circuit is used in the circuit. This security loop hole can be very dangerous when the robot is being used for very confidential purposes. In areas of military these security threats can produce disastrous outcomes.

2.2.4 How we overcome Those problem in Project

1. Wifi connection is used for the operation of robot.
2. This gives the high security.
3. It also provide much more reliability of operation working range it also includes wifi.
4. N type wifi is used for higher security purpose as well as for better range.

2.6 Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor

2.6.1 Description

Raspberry pi is a credit- card sized computer .It functions almost as a computer . There are various surveillance systems such as camera ,CCTV etc., [25] In these types of surveillance systems, the person who is stationary and is located in that particular area can only view what is happening in that place. Whereas, here ,even if the user is moving from one place to another he/she can keep track of what is happening in that particular place. Also another advantage is that it offers privacy on both sides since it is being viewed by only one person. The other major advantage is that it is a simple circuit .the operating system used here is Raspbian OS.Raspbian OS has to be installed so that the image can be transmitted to the smartphone.

Functional Description

The functions of the various components are given below: A. USB Camera: USB Camera captures the image and sends it to the USB port of the functionalities of the components are given below The various components of Raspberry- Pi are functionalities of the components are given below The various components of Raspberry- Pi are[26]

• SD Card Slot is used to install OS/booting/long term storage .The total memory of the SD card is about8GB.

• Micro USB Power Port provides 700mA at 5A

* RCA Video Out is connected to display if HDMI output is not used. It is mainly used to carry audio and video signals. They are otherwise called as A/V jacks.
* Audio Out Digital audio is obtained if HDMI is used to obtain stereo audio. Here analogue RCA connection is used.
* Ethernet Port is used to connect to the Internet. It also plays a role in updating, getting new software easier.
* HDMI OUT(High Definition Multimedia Interface) is used with HDTVs and monitors with HDMI input. Also HDMI-HDMI is used here.
* BROADCOM BCM 2835: It is otherwise defined as System on chip .It is a 700 MHz Processor. It has a Video core IV GPU.
* GPIO allows us to control and interact with real world . Raspberry Pi board. The camera model used here is USB Camera model 2.0.

2.6.2 Pros

1. Fast processing
2. High range
3. Can perform various task

2.6.3 Cons

1. There is no dedicated app
2. There is no provision for video streaming
3. Control using already developed application
4. It is very expensive
5. Even the camera used, is very expensive

2.6.4 How we overcome Those problem in Project

1. We use a network camera, mounted on the robot, for providing the live video feed
2. We have developed our own app to control the robot wirelessly
3. Our app includes live video streaming
4. It can be used for various real world applications
5. We use Arduino board which is cheaper than Raspberry pi
6. We use Android phone istead of expensive camera

2.7 Technological Review

2.7.1 Android (operating system)

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, regular PCs (e.g. the HP Slate 21) and other electronics.

As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded. A developer survey conducted in Aprilâ“May 2013 found that 71 % of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013. As of 2015, Android has the largest installed base of all generalpurpose operating systems.

Android’s source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software developed and licensed by Google. Initially developed by Android,Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

2.7.2 Raspberry pi board

2.7.2.1 What is raspberry pi?

2.7.2.2 Why raspberry pi?

2.7.2.3 Bluetooth

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz[4]) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 25,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics.The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks.A manufacturer must make a device meet Bluetooth SIG standards to market it as a Bluetooth device. A network of patents apply to the technology, which are licensed to individual qualifying devices.

.

7.2.4 Wi-Fi

Wi-Fi (or WiFi) is a local area wireless technology that allows an electronic device to participate in computer networking using 2.4 GHz UHF and 5 GHz SHF ISM radio bands.

The Wi-Fi Alliance defines Wi-Fi as any "wireless local area network" (WLAN) product based on the Institute of Electrical and Electronics Engineers’ (IEEE) 802.11 standards".However, the term "Wi-Fi" is used in general English as a synonym for "WLAN" since most modern WLANs are based on these standards. "Wi-Fi" is a trademark of the Wi-Fi Alliance. The "Wi-Fi CERTIFIED" trademark can only be used by Wi-Fi products that successfully complete Wi-Fi Alliance interoperability certification testing.

Many devices can use Wi-Fi, e.g. personal computers, video-game consoles, smartphones, digital cameras, tablet computers and digital audio players. These can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.

Depiction of a device sending information wirelessly to another device, both connected to the local network, in order to print a document.

Wi-Fi can be less secure than wired connections, such as Ethernet, because an intruder does not need a physical connection. Web pages that use TLS are secure, but unencrypted internet access can easily be detected by intruders. Because of this, Wi-Fi has adopted various encryption technologies. The early encryption WEP proved easy to break. Higher quality protocols (WPA, WPA2) were added later. An optional feature added in 2007, called Wi-Fi Protected Setup (WPS), had a serious flaw that allowed an attacker to recover the router’s password.The Wi-Fi Alliance has since updated its test plan and certification program to ensure all newly certified devices resist attacks.

Requirement Analysis

3.1 Platform Requirement :

3.1.1 Supportive Operating Systems for Server :

1. Ubuntu 12.4 and above The supported Operating Systems For server include Linux. Linux is used as server operating system.
2. Window 7 and above Microsoft Windows is a series of graphical interface operating systems developed, marketed, and sold by Microsoft. Microsoft introduced an operating environment named Windows on November 20, 1985 as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs).
3. Android OS Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

3.1.2 Supportive Operating Systems for Client:

1. Android OS Android is open source operating system based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation

3.2 Software Requirement :

The Software Requirements in this project include:

3.2.1 Android studio :

(need to be written)

3.2.2 JDK :

Java Platform (JDK) 8u25 The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application.

3.2.3 JVM :

JVM Version 8 A Java virtual machine (JVM) is an abstract computing machine. There are three notions of the JVM: specification, implementation, and instance. The specification is a book that formally describes what is required of a JVM implementation. Having a single specification ensures all implementations are interoperable. A JVM implementation is a computer program that meets the requirements of the JVM specification in a compliant and preferably performant manner. An instance of the JVM is a process that executes a computer program compiled into Java bytecode.

3.3 Hardware Requirement :

The Hardware components required for our project are Min 1 GB of RAM,10 GB HDD,Dual core processor for the machine on which development will be done,Robot kit,IP Camera,Bluetooth Module etc for developing the robot.

3.3.1 Hardware Required For Project Development:

1. Raspberry pi
2. Rotor motor
3. Batteries
4. Camera module
5. Wifi module
6. Chassis

Robot Details

1. Platform design

Since SPY is designed for exploring limited places like very narrow places or height limited places. It is will be more appropriate to make it as small as possible. I did not choose it to as small as 1-inch cube size because it will increase design difficulty and because this is a prototype, the size of some parts probably not small enough, so the whole platform size is around 5-inch cube.

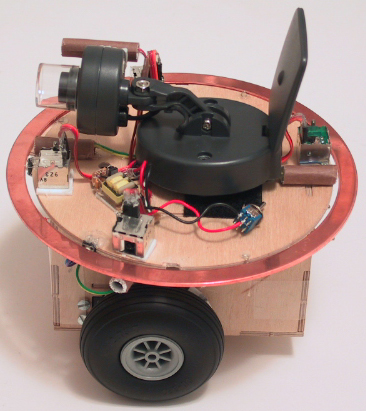


Figure 1 SPY platform design.

There are three wheels, two of them is driven by two servos. Wheel encoders is attached to two front wheels to record how many steps has been explored at each wheel at a given time. In this way, the robot can trace back to its home using these recorded data. Shaft encoder is attached at left and right side of robot against the black and white stripe wheel.

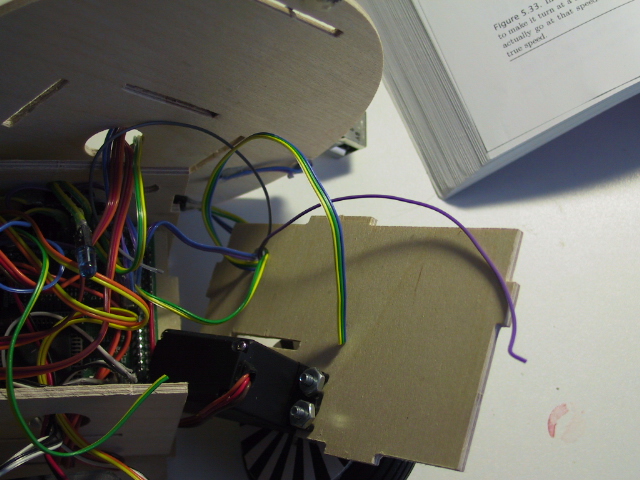
 

Figure 2. Wheel design(left) and shaft encoder(right)

The power is composed of 6 NiCd batteries to supply enough driven strength. Power pack is located at the rear side of robot. Since wireless video camera consumes lots of power for wireless communication, it would drain out lots of power from robot. To avoid burn out TJPro board, I designed another power line dedicated for video camera. And since video need 12V, I also design a 4.5V to 12V regulator for video camera power supply.

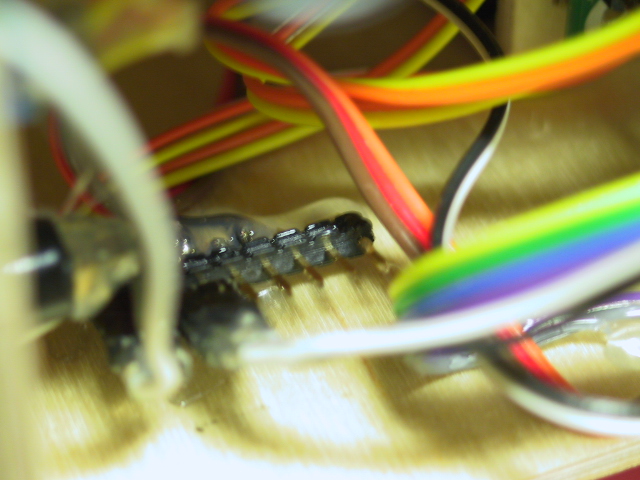


Figure 3. (Left)Power pack location (Right) Another power line dedicate for video camera

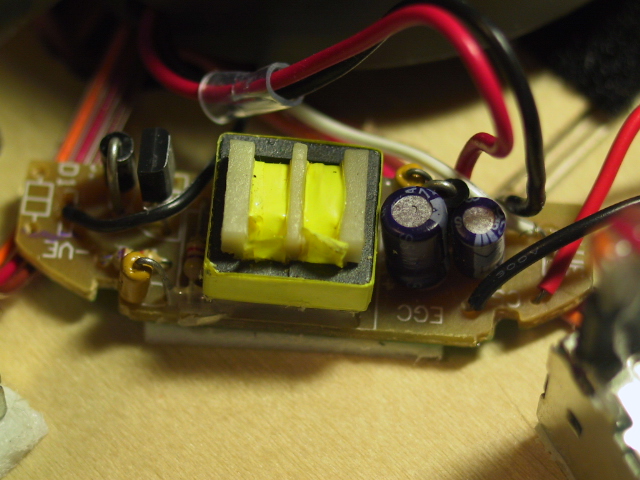


Figure 4. Video camera power regulator

Camera is located at top of robot. To avoid collision, 4 IR sensors are equipped. IR sensors sense that if any obstacle is approaching ( or robot is approaching any obstacle). IR sensors are located at front left, front right, rear left and rear right of robot. Because IR detector is very sensitive and bumper is made of metal, I use alumni to shield IR emitter such that it will be affected by bumper and the IR detector will not so sensitive to the environment. Since camera can not show image when the light is dim, I use one photocell to detect light condition, it is located beside the camera.

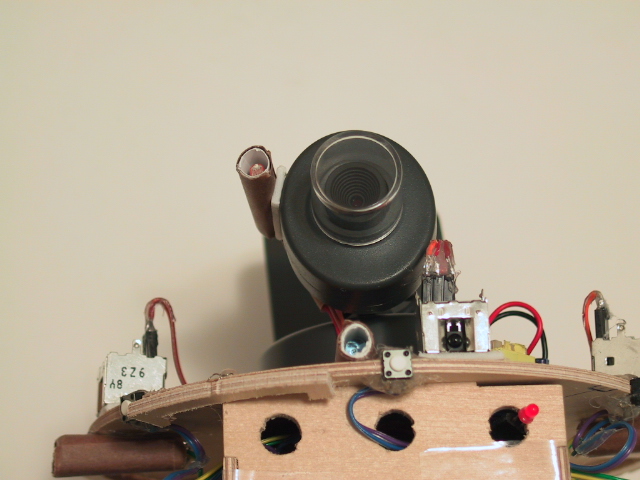


Figure5. Camera and light sensors. Below the camera are IR emitter and IR detector

Sometimes, IR sensos do not work so well, bumpers are needed to prevent collision. Since bumper should be firm enough to make bumper sensor work properly, I use copper bumper.

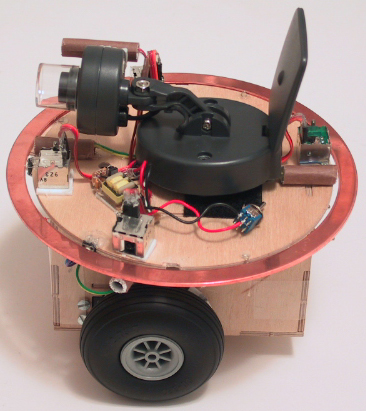


Figure 6. Bumper design

1. Actuator Design

The actuation that will be used in this project are tow hacked servos. Hacked servo is connected with driven wheel, it can forward and backward according to code used. Two wheels are being used as driven wheel.

1. Sensors Design

Sensors that will be used in this project are: 4 IR sensors, two wheel encoders, 4 micro-switches, one photoresistor and one sonar.

IR sensors are used for sense possible obstacles. Wheel encoders are used to record how many “steps” has passed in a given time at each wheel. And micro-switches are used for bumper. Photoresistor is used for sense environment light condition, if it is in low light condition, robot will turn on its own light to help camera capture images.

**Shaft encoder components**

The components in my wheel encoder are:

* Shaft encoder
* 32 segments striped wheel
* Interface circuit
* Software driver

**Shaft Encoder**

Shaft encoder is a sensor that measures the position or rotation rate of a shaft. From signal point of view, we can tell shaft encoder as: absolute encoders and incremental encoders. Absolute encoders deliver a code that corresponds to a particular orientation of the shaft while incremental encoders produce a pulse train. The pulse is generated when output voltage of shaft encoder changes from high to low(or vice versa), so the faster the shaft turns the faster the pulse generated.

From implementation point of view, we can tell shaft encoder as:photointerrupter and photoreflector. Photointerrupter includes an inferared LED emitter and a photodetector and a slotted disk. When disk spins, the light is interrupted by the moving slots thus photodetector detects the changes of the light and generate pulse at its output. If we use microprocessor the count the pulses generated by photodetector, we can know how far the wheels has rotated. The photoreflector is different, it uses the reflection of the light to detect the movement of the wheel. Inferared LED emitter emits light to striped wheel. Since light will not reflect if it meets black stripes, phototransistor then detects the changes of the light and sends out singals according to the reflection.

The encoders that I use in SPY robot is Hamamatsu P5587 photoreflector. The photoreflector is packaged with an infrared LED and a phototransistor in a very compact unit. The device is attached on the body of the robot and the distance between device and striped wheel is about 0.5cm.

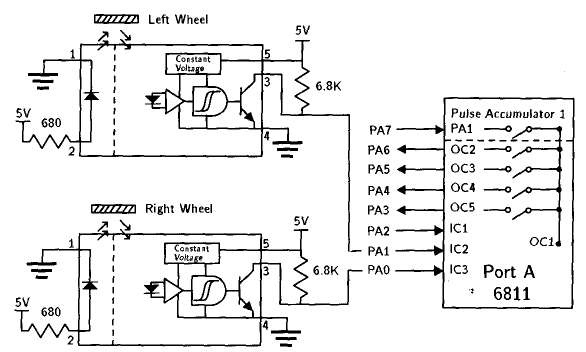
**Striped Wheel**

The width of the stripes is an important issue in designing shaft encoder. I tried 32 segments, 48 segments and 64 segments striped wheel, only 32 segments striped wheel works correctly Although more stripes give greater resolution to the output measurements, but the stripes cannot be narrower than the field of view of the photoreflector. That’s the reason why it does not work well in 48 segments and 64 segments.

Because the near-infrared energy emitted by the LED can penerate thin, white paper, so I put thick hard paper behind printed striped wheel this will make sure the light will reflect if it meets white stripes.

**Interface circuit**

Since Hamamatsu P5587s photoreflectors have circuitry integrated in the package to amplify and condition the output of the phototransistor, the only interface components required for connecting to the TJPro board are two resistors: 680Ω and 6.8KΩ resistor. 680Ω resistor is connected to the input of the LED to limit the current through LED. 6.8KΩ resistor is connected to the output of the phototransistor to pull up phototransistor's open-collector output.



The power for Hamamatsu P5587s photoreflector is coming from analog port on TJPro board. Analog port provides regulated 5V and ground connection thus meets our need.

The wires connecting TJPro board and photoreflector are: Grey for pin 1, Purple for pin 2, Blue for pin 3, Green for Pin 4, Yellow for pin 5. The pin assignment for Hamamatsu P5587s photoreflector is as below: Pin 1 is connected to LED ground. Pin 2 provides LED power. Pin 3 is for signal. Pin 4 is ground and Pin 5 is power.

MC68HC11’s port A has 8 pins, they have various input capture and output compare registers associated with them, which are able either to mark the time that events happen on those pins or to initiate events at preprogrammed times.

In the original design of interface circuits, the left wheel shaft encoder is connected to MC68HC11’s PA7. A pulse accumulator function is associated with PA7 making it easy to count the pulses produced by the shaft encoder in software, but PA7 pin is used for motor control, I choose PA1 and PA0 for right and left wheel shaft encoder. I only use pin 1 of PA1 and PA0 on TJPro board, pin2 and pin 3 which are for power and ground connection is not used since they are not regulated.

**Software Driver**

To drive wheel encoder, we need some surgery to interrupt handler. MC68HC11 port A pin PA0, PA1 and PA2 are associated with input capture registers. To count encoder clicks, we must use interrupt handlers. Since right wheel encoder is connected to PA1, IC2 register is for right wheel encoder. Same as left wheel encoder, which is connected to PA0, thus IC3 register is for left wheel encoder.

There are four actions can be assigned to input capture registers. They are: capture disable, capture on rising edge and capture on any edges. We must use TCTL2 register to set the desired response for any successful input-event detection. Since four different responses can be assigned for one input capture, it is obvious that we need 2 bits to represent all possible value. In TCTL2 register, EDG3A and EDG3B is for setting IC3 response action, EDG2A and EDG2B is for setting IC2 response action. The address for TCTL2 is $1021, in ICC11, TCTL2 is a legitimate variable, we don’t need to do anything before using this variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EDG3A | EDG3B | EDG2A | EDG2B | Configuration |
| 0 | 0 | 0 | 0 | Capture Disable |
| 0 | 1 | 0 | 1 | Capture on Rising Edge |
| 1 | 0 | 1 | 0 | Capture on Falling Edge |
| 1 | 1 | 1 | 1 | Capture on Any Edge |

We will trigger on risign edges, so four bits must be written to the TCTL2 register to configure it for risigng edge-triggered interrupts. In order not to disturb other setting, I store 0101 to the least significant 4 bits and leave other four bits not changed.

TCTL2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $1021 | 0 | 0 | EDG1B | EDG1A | EDG2B | EDG2A | EDG3B | EDG3A |
|  | x | x | x | x | 0 | 1 | 0 | 1 |

Each time an interrupt happened according to the event we assigned, an interrupt flag will be set indicates that an interrupt event happened. The flag of input capture register is contained by TFLG1 register. TFLG1 register is at address $1023, and we can use TFLG1 variable name in ICC11 directly without pre-define. When input capture event happened, associated flag bit ICxF will be set, hardware will automatically initiate an interrupt. Code in the interrupt service routine thus must clear the ICxF flag; otherwise, when an attempt is made to return from the interrupt, the hardware will think the ICx interrupt is pending and immediately service it again. Suppose IC3 interrupt happened, after service, we must write a 1 to the bit in the TFLG1 register that corresponds to that interrupt’s flag as below.

TFLG1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $1023 | OC1F | OC2F | OC3F | OC4F | OC5F | IC1F | IC2F | IC3F |
|  | x | x | x | x | x | x | X | 1 |

It is obvious that after IC2 service, we must clear IC2F flag. After above discussion, the interrupt handler routine, which automatically runs whenever a rising edge is detected, must do the following tasks: increment a counter, clear the interrupt flag and return from the interrupt.

Now we need to configure interrupt vector such that when IC event happened, hardware will jump to our service routine instead of default interrupt service. The TMSK1 register contins the bits that must be set to enable interrupts associated with events on any input capture pin. TMSk1 is at address $1022 and the bits associate to input capture 3 and input capture 2 are IC3I and IC2I.

TMSK1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| $1022 | OC1I | OC2I | OC3I | OC4I | OC5I | IC1I | IC2I | IC3I |
|  | x | x | x | x | x | x | 1 | 1 |

Before we set IC2I and IC3I bits, interrupt handler will not handle inputer capture 2 and input capture 3 events. Now we should redirect interrupt vector to our routine. The vector address for IC3 interrupt is $FFEA, and IC2 interrupt is $FFEC. The two byte address stored at this location is the address at which the user’s interrupt handler code muse begin. In ICC11, to modify interrupt handler, we must modify vectors.c file. The interrupt handler routine for IC2 and IC3 now are changed to my\_TIC3\_isr and my\_TIC2\_isr routine. The whole flow is presented in following page.

Source code for software driver is attached at the end of report.

1. Behavior

User first assign one position to robot like x=???, y=??? then robot will go to that position. On the path to destination, if robot meet obstacle, it will turn around and continue its path. While it is walking, it will record steps that it used in past few seconds to record path. Once destination is reached, robot will return to its home according to the data it recorded.

In the beginning, the coordinate that robot standing now is position (0,0). To go to destination, robot will try to increase Y direction first. If it met obstacle, it will turn 90 degree and try to increase X direction. If positive Y and X direction has obstacle, it will go to negative Y direction until X direction is free of obstacle then go X direction. If Y or X direction is meet, robot will go other direction until destination is meet.

Since this robot is not for maze explorer, I will not program it as those mice that run in maze. But reasonable obstacle avoidance will be programmed.

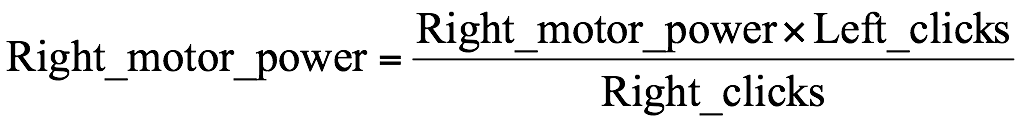
1. Experimental Layout and Results

**Wheel Encoder**

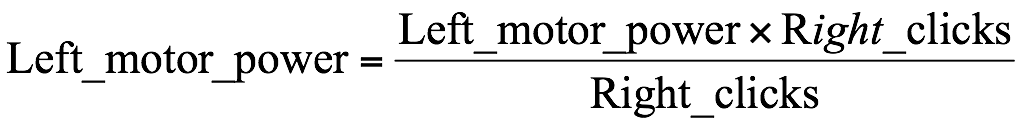
Wheel encoder can be used to measure the distance from last measured position to current position. To measure this simply reset counter for each wheel in the beginning of measure, then at the destination ,see how many clicks happened so far then we know the distance. To measure velocity of robot, we need constantly check clicks of each wheel. This can be done by checking clicks every second(or reasonable time), then we know the velocity of robot.

The most challenge application of wheel encoder is to synchronize two wheels such that robot walks as straight as possible. This includes measure wheel clicks and adjust motor power level. Since motor device is not uniform and the zero point of each servo is different and the adjustment for different servo is different too. Thus it is almost not possilbe for two servos with identical properties. The challenge now becomes how could we use software to adjust two servo such that two servo will have similar performance.

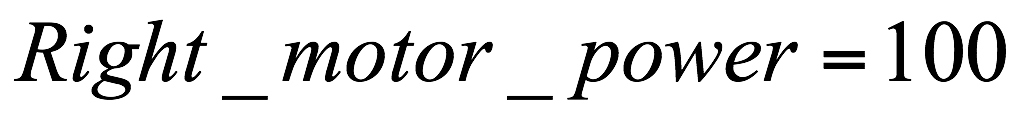
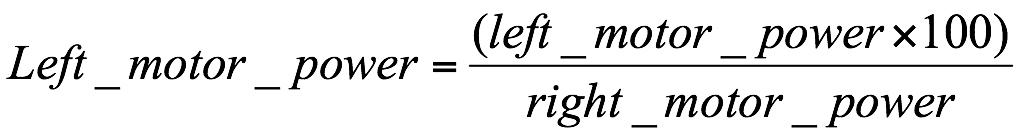
Using wheel encoder and make two servos runs at the same rate is feasible. Before robot begins, robot itself will run a self-calibrate to sync two servo. Once it got reasonable clicks sync, it will start perform its task. In the beginning of calibrate routine, it will assign full power level for each motor. Then run and wait 9000 e-clocks. Since processing interrupt takes time, after 9000 e-clock, we wait another 500 e-clock to let interrupt handler finish its work. Then we compare clicks from two wheels. The adjustment for each motor is according to the click counts. If right wheel count is more than left wheel count then right motor power is adjusted according to following equation:



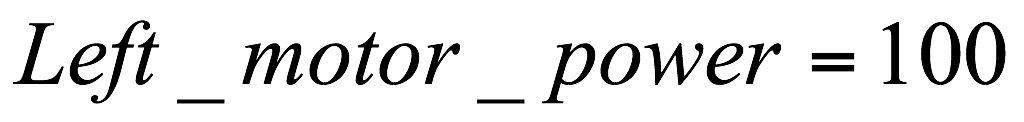
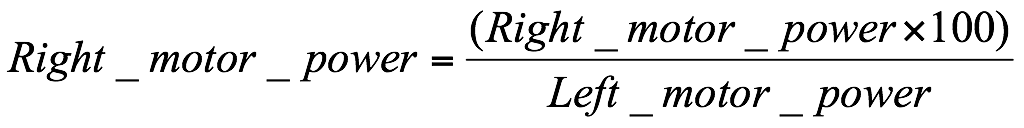
The same is done when left wheel count is more than right wheel count accroding to following equation:



But after some tries, both left motor power and right motor power will decrease, we don’t want this situation happened, so I made re-adjustment if motor power becomes too low. If right motor power level is lower than 100 and after adjustment left motor power will not exceed 100, then we do following adjustment:

Same is done when left motor power is lower than 100:

Because moto performance will decrease when battery power level decrease, this increases the difficulties of syncing both motors. So when two motors are at the same clicks, program will try same power level and see if the result clicks are the same. If they are the same, then calibrate routine ends and begins to execute regular task, otherwise it will calibrate again.

I do 20 times of experiments and found right motor power is about twice powerful than left motor power. Actually after calibrate, the average power level for left motor is always 100, and power level for right motor is about 74. The average calibrate time in each experiment is about 7\*9500=66500 e-clocks.

The experiment chart and result is included at the end of report.

**Range measure**

From the experiment I did, the range and the click counts is about:

1 feet = 26 clicks

**Rotation measure**

For rotate 90 degree, the click count should be 6 clicks.

**IR detection measure**

For correctly detect obstacle, IR detector should continuous detect bounced IR. In the experinment, if obstacle is in half feet range, IR detector will have a reading about 100. I use this as threash hold to tell that an obstacle is detected.

**IR coverage**

In the IR coverage experinment, I use hand to detect the exact IR coverage. The measured IR coverage is about 45 degree.

Project Design

4.1 Design Approach

4.1.1 Front End Designs

Fig: Front End

4.2 Software Architectural Designs

Figure 4.2: Software Architectural Designs

4.2.1 Component Diagram

Figure 4.3: Component Diagram

4.2.2 Deployment Diagram

Figure 4.4: Deployment Diagram

4.3 Work-flow Design

Figure 4.5: Work-Flow Design

4.4 Flow Graph

4.4.1 DFD LEVEL

Figure 4.6: DFD Level 0

Figure 4.7: DFD Level 1

Implementation Details

5.1 Assumptions And Dependencies

The new age of technology such as Android, GSM has redefined communication. Most people nowadays have access to mobile phones and thus the world indeed has become a global village. At any given moment, any particular individual can be contacted with the mobile phone. New innovations and ideas can be generated from it that can further enhance its capabilities. Technologies such as Infra-red, Bluetooth, Wi-Fi which has developed in recent years goes to show the very fact that improvements are in fact possible and these improvements have eased our life and the way we live. Remote management of several home and office appliances is a subject of growing interest and in recent years we have seen many systems providing such controls.

Mobile robots are robots which have the ability to move around and interact with their environment and not just hinged to a particular place. There are many labs and research groups from various universities and industries which are completely dedicated on researching mobile robots, because of their immense potential and varied application in industry, military, security, and entertainment.

The robot is specially designed for surveillance purpose. The control mechanism is provided along with video transmission facility. The video transmission is practically achieved through high speed image transmission. Initially, the robot will be equipped with an Android Smartphone which will capture the scenario in front of it will transfer the images to the server on which the user will be controlling and watching the live feed.

5.2 Implementation Methodologies

5.2.1 Modular Description of Project

5.2.1.1 Bluetooth module

The module provides a method to connect wirelessly with a PC or Bluetooth phone to transmit/receive embedded datasuch as GPS data, ADC voltage reading and other parameters. Bluetooth module JY MCU BT used in the project can be connected to any device, via built in UART interface to communicate with other Bluetooth -enabled devices such as mobile phones, handheld computers and laptops. The module runs on a 3.6V to 6V supply.

5.2.1.2 raspberry pi

5.2.1.3 L293d

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal to run four solenoids, two DC motors or one bi-polar or unipolar stepper with up to 600 mA per channel using the L293D. These are known as the drivers in the Ada fruit Motor shield.

The L293 and L293D are quadruple high-current half-H drivers. In L293 is designed to provide bidirectional drive currents up to 1 A at voltage range from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltage range from 4.5 V to 36 V. Both the devices are designed to drive inductive loads such as relays, dc and bipolar stepping motor as well as other high-current/high-voltage loads in positive-supply applications.

The Android application is basically divided into two modules i.e.

1. Video Streaming Module
2. Robot control module A network camera or an Android mobile phone is mounted on the robot, which our application fetches the live video streaming display it. This video is achieved using WIFI technology.

The second module is the control module. Our application provides a GUI to control the robot wirelessly. This control is achieved using Bluetooth technology. Button are used to control the robot in forward, backward, left or right direction.

5.3 Detailed Analysis and Description of Project

The android application will be used to search for products and view stores at which those products are available. The android application will need to communicate to a GPS application within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user. The GPS will provide the mobile application with locations of both the user and the stores and the distance between them, but it will also provide maps and the functionality to display the applicationâTMs data on the map. The functionality provided by the GPS will be embedded into the application in order for the user to be able to use the functions in the application in a seamlessly manner.

Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The android application will communicate with the database. The user will use the application to get data from the database while the vendor will also add and modify data, which will be monitored by the admin. All of the database communication will go over the Internet. The mobile application has some restrictions about the resource allocation

5.3.1 Usecase Report

|  |  |
| --- | --- |
| Title: | Surveillance Robot Controlled Using An Android App |
| Description: | In this project we can control the surveillence robot using an android app. This is done through the help of a user-friendly GUI. |
| Primary Actor: | The one who uses the mobile |
| Preconditions: | Installation of Android app on our mobile |
| Postconditions: | controlled a robot using app . |
| Main Success  Scenario: | 1. Featching a live videostreaming on android app.  2. video is taken from a mobile which is mounted on a robot and then send to our android app.  3. The Procedures starts by seeing a video then we can control the robot left,right,forward,bavckward easily |
| Frequency of Use: | User can use many times |
| System Requirement: | android app. |

Table 5.1: Usecase Report

Figure 5.1: Use Case Diagram

5.4 Class Diagram

Figure 5.2: Class Diagram

|  |  |
| --- | --- |
| Title: | Surveillance Robot Controlled Using An Android App |
| Description: | In this project we can control the surveillence robot using an android app. This is done through the help of a userfriendly GUI. We can fetch a video using two mobiles one is mounted on a robot and another one is at a user hand. |
| Primary Actor: | The one who uses the Android App |
| Preconditions: | User connect a robot using bluetooth |
| Post conditions: | watching a live video streaming and controlling it |
| Comparative Website : | 1. Data Independence 2. Controlled Redundancy 3. Accuracy 4. Security 5. Performance |
| Arduino: | Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computers. |

5.5 Class Report

Table 5.2: Class Diagram Report

Results and Discussion

6.1 Test Cases

1. When i have tested my robot i have found that the range of bluetooth is aroud 10 meter.

2. In dark situation there was no light so that i used LED flash light for better visibility.

3. Our robot is easily move on the rough area.

6.2 Results Discussion

We have two results i.e. the hardware and the software result. The hardware includes the robot which runs on DC motors. The input to the motors is provided by the L293D motor driver shield. The input to the driver shield is provided by the arduino board.

The navigational inputs are given by the user to the arduino board using the android application via Bluetooth. The arduino board, on receiving the signal, processes it and produces the appropriate output. The communication between the android application and the arduino board takes place using the Bluetooth module which is interfaced with the arduino board. It provides serial communication between the application and the Arduino.

Project Time Line

7.1 Project Time Line Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Name | Duration | Start | Finish |
| 1 |  | 1(a) Requirement Gathering |  |  |  |
| 2 |  | 1(b) Confirms Requirement |  |  |  |
|  |  |  |  |  |  |
| 3 |  | 2(a)Frontend-user interface |  |  |  |
| 4 |  | 2(b)Backend robot desining |  |  |  |
|  |  |  |  |  |  |
| 5 |  | 3(a) Frontend Coding |  |  |  |
| 6 |  | 3(b) Robot Creation |  |  |  |
| 7 |  | 3(c) Coding for GUI |  |  |  |
| 8 |  | 3(d) Creation of test case |  |  |  |
|  |  |  |  |  |  |
| 9 |  | 4(a) Unit Test |  |  |  |
| 10 |  | 4(b) System Test |  |  |  |
| 11 |  | 4(c) Functional Test |  |  |  |
|  |  |  |  |  |  |
| 12 |  | 5(a) Deployment |  |  |  |

Figure 7.1: Time Line Matrix

7.2 Project Time Line Chart

Task Distribution

8.1 Distribution of Workload

8.1.1 Scheduled Working Activities

|  |  |  |
| --- | --- | --- |
| Activity | Time Period | Comment |
| Requirement Gathering | 08 Days | Requirement gathering has took placed through searching on internet and taking the ideas, sharing the views among group members. |
| Planning | 04 Days |  |
| Design | 04 Days | Designing has done by creating UML diagram, By creating Charts, |
| Implementation | 90 Days | Implementation has done First creating the backend and then front end module by module. |
| Testing | 10 Days | Testing has done by perfoming unit testing, alpha & Beta Testing, integrated testing and system testing. |
| Deployment | 05 Days |  |

Table 8.1: Scheduled Working Activities

8.1.2 Members actvities or task

Conclusion and Future Scope

9.1 Conclusion

We have successfully implemented the working of the wireless video surveillance robot controlled using android mobile device. The robot is successfully controlled using the android application through the wireless Bluetooth technology. Even the real time video feel is successfully achieved using the Wi-Fi technology on our designed android application.

9.2 Future Scope

Surveillance is needed in almost every field. It could be a great solution to various problems or situation where wireless Surveillance is needed our project has tremendous scope as it uses the latest technology in the market.

Our application uses the android OS which is currently the most used OS and also has a great future scope. The Surveillance robot can be controlled remotely using the android application; this gives it a huge scope for future application.