Gesture based wireless single-armed Robot in Cartesian 3D space using Kinect

B.E. Dissertation

Submitted in partial fulfillment of the requirement of

University of Mumbai

For the Degree of

Bachelor of Engineering
(Electronics and Telecommunication Engineering)

by

Memon Md. Farhan Md. Fareed (11ET33)
Quraishi Imran Akram (11ET41)
Shaikh Bilal Anees (12ET86)

Under the guidance of

Prof. Rizwan Alvi

Department of Electronics and Telecommunication Engineering
Anjuman-I-Islam's Kalsekar Technical Campus,
Sector 16, New Panvel, Navi Mumbai
(Affiliated to University of Mumbai)

April 2015

Anjuman-I-Islam's

Kalsekar Technical Campus

(Affiliated to the University of Mumbai)

Plot 2 and 3,Sector 16, Khandagaon, Near Thana Naka, New Panvel, Navi Mumbai 410206.

Certificate

This is to certify that, the dissertation titled

"Gesture based wireless single-armed Robot in Cartesian 3D space using Kinect"

is a bonafide work done by

Memon Md. Farhan Md. Fareed (11ET33)
Quraishi Imran Akram (11ET41)
Shaikh Bilal Anees (12ET86)

and is submitted in the partial fulfillment of the requirement for the degree of

Bachelor of Engineering
in
Electronics and Telecommunication Engineering
to the
University of Mumbai.

Guide | Project Coordinator

Head of Department | Principal

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Memon Md. Farhan Md. Fareed: ______________________

11ET33

Quraishi Imran Akram: ______________________

11ET41

Shaikh Bilal Anees: ______________________

12ET86

Date:

Place:

Certificate of Approval by Examiners

This is to certify that the dissertation entitled "**Gesture based wireless single-armed Robot in Cartesian 3D space using Kinect**" is a bonafide work done by **Memon Md. Farhan Md. Fareed, Quraishi Imran Akram and Shaikh Bilal Anees** under the guidance of **Mr. Rizwan Alvi**. This dissertation has been approved for the award of **Bachelor's Degree in Electronics and Telecommunication Engineering**, University of Mumbai.

Examiners:

Name:	Name:

Acknowledgment

We have immense pleasure in expressing our thanks and deep sense of gratitude to our guide ***Mr. Rizwan Alvi, Assistant Professor, Department of Electronics and Telecommunication Engineering, AIKTC*** for his guidance throughout this project.

We would also like to express our deepest appreciation to our project coordinator ***Mr. Rizwan Alvi, Assistant Professor, Department of Electronics and Telecommunication Engineering, AIKTC*** for his technical support in our project.

We also express our sincere thanks to ***Prof. Mujib Tamboli, Head of the Department, AIKTC*** for extending his help.

We wish to express our profound sense of gratitude to ***Dr. Abdul Razak Honnutagi, Director, AIKTC*** for his encouragement, and for all facilities to complete this project.

We express our sincere gratitude to ***Mr. Awab Fakhi, Assistant Professor, Department of Electronics and Telecommunication Engineering, AIKTC*** and ***Mr. Afzal Shaikh, Assistant Professor, Department of Electronics and Telecommunication Engineering, AIKTC*** and all the members of faculty, non-teaching staff, our family and friends who contributed their valuable advice and helped to complete the project successfully.

Abstract

Human Machine Interaction (HMI) has always played an important role in everybody's life motivating research in the area of intelligent service robots. Conventional methods such as remote controllers or wearables cannot cater the high demands in some scenarios. To overcome this situation, the challenge is to develop vision-based gesture recognition techniques. This project describes our work of controlling an Arduino based wheeled, one armed robot, used as a prototype, controlled through various gestures of the arms and legs. For gesture recognition, we make use of skeletal tracing ability of Kinect – a product of Microsoft. Bluetooth is used to make the controls wireless. Since it is not line of sight operation, the robot also captures the environment video and transmits it over radio frequency in real-time and displayed on the screen. On the user end, according to the received video, the operator guides the robot and uses the arm to pick and place objects with the help of predetermined gestures.

List of Table and Figures

Contents

Chapter 1

INTRODUCTION

1.1 Overview

We are in the age were technology is upgrading with a blink of eye. We had ball controlled wired mouse an amazing invention which hit the market with a roar. People used it with lots of enthusiasm, but the maintenance of the ball proved hectic. Inventors came up with optical mouse a pioneer innovation. Luxury and comfort later became the need of the hour so this got modified in to wireless pieces, which operate from anywhere in the room.

Robotics is another highly developing field. Wired robots, controlled using remote controls, which emerged in to wireless RF remote controlled robots. These can be manipulated as RF is a broadcasting frequency band. Also user need sufficient knowledge regarding the controls along with practice. This moved the cursor towards creating better Human machine interactions (HMI).Inspired with advancements in HMI we aimed towards designing a vision based gesture controlled robot, which can be operated merely by pure human gestures. For this we made use of Microsoft's Kinect sensor. Its skeleton tracking ability is efficiently improvised by us in controlling a single armed robot using gestures. The programming done by us in C# is so core that any person standing in front of Kinect is tracked continuously for his gesture. The person can control the robot by his own arm movements effectively. In short the robot will replicate the movements. The robot performs motion on surface.

To make it portable we made it wireless which function using a Bluetooth that makes one - one communication making it secured from hacking. The robot is also equipped with wireless RF camera to capture the scenario to be controlled accurately by remote user. Collision can cause damage, ultrasonic sensors are placed on robot to prevent collisions from obstacles.

The robot provides quick response to the movements made by it user. It flawlessly operates within the range of Bluetooth.

1.2 Software Specifications

- Microsoft .Net framework 4.5 or higher
- Microsoft Visual Studio 2013
- Microsoft Kinect SDK v1.8.0
- Atmel Studio v6.2
- Arduino v1.0.5
- Eagle v7.1.0
- Multi-Sim

1.3 Hardware Specifications

- Arduino MEGA 2560 board.
- Motor Driver LM-293D
- You also need a Micro USB cable to connect your PC to the Arduino Board to upload code.
- A Bluetooth Shield HC-05
- Few mini modular bread boards
- Jumper wires – Mixed pack with M/M, F/F, M/F
- A Tiny Bluetooth dongle for your PC/Laptop, like this one, to communicate with the blue tooth shield in the robot (if you don’t have built in Bluetooth)
- A battery pack and battery holder that should supply around 12V to the Motor Driver
- Duct tapes/rubber bands.

Chapter 2
System study and analysis

2.1 Existing System

The current systems which are vastly spread over different places in the field of Robotics and Automation are the ones in which the robot or the destination machine is controlled form the source or the controller by sending the codes or instructions.

The current processing of gestures is done on software like Matlab, Wolfram, Maple, etc. these are the conventional Image Processing techniques. Also instructions are generated by the controller from devices which are physically worn or are handled by the controller like the Accelerometer, gyroscope, or the MOVE controller from SONY.

We need something other than this. Also the programming complexity of previous languages has made the advancement in automation a bit slow, time consuming and a costly affair.

2.2 Problem Description

The core problem faced by any system is its cost effectiveness, the systems developed were very expensive in the form of processing software like Matlab, Wolfram, Maple, etc. and also these were manipulating gestures in the form of wearables or hand-held sensors or devices these were not pure gestures, they are imitating it.

Also the problem faced was that the controller in gesture control has a lose connection or broken pin of the controlling device or fluctuationin the expected outcome of the results or something unexpected for proper control of the device. Also programming is a big headache in C or C++.

2.3 Proposed System

As we have read above that we need something else but what??

What we propose is that we need automation and robot control without any restrictions of the held on devices or the wearables, we have come across the sensor by Microsoft named Kinect in which we can control in our case our Robot by just standing in front of the sensor. The sensor with appropriate programming does the Image Processing at very low costs when compared to the current system which costs Lakhs of Rupees and this sensor does it in few Thousands.

We will be using the latest C# for simplicity of its programming techniques. Also we are going to make it a wirelessly controlled robot with a wireless RF camera to avoid the wiring complexities to the maximum possibility.

We have planned to completely eliminate the problems faced by the user in the area of gesture control mentioned above in the Problem Description.

So in our proposed system we are defining new technology of gesture control. Table 2.1 gives the comparative description as to why our project is superior to the existing conventional forms of the gesture controlling techniques.

Parameter	Conventional Technique	Proposed Technique
Gesture type	Manipulative i.e. using wearables or having limitations.	Pure visual-based gesture.
Processing technique	Matlab, Wolfram, Maple and other image processing software.	Visual Studio and Arduino.
Control type	Wired	Wireless
Language	C, CPP, C++, etc.	C# and Java.
Cost	Very Costly	Comparatively very cheap

Table 2.1: Comparative Study

Chapter 3

IDEOLOGY

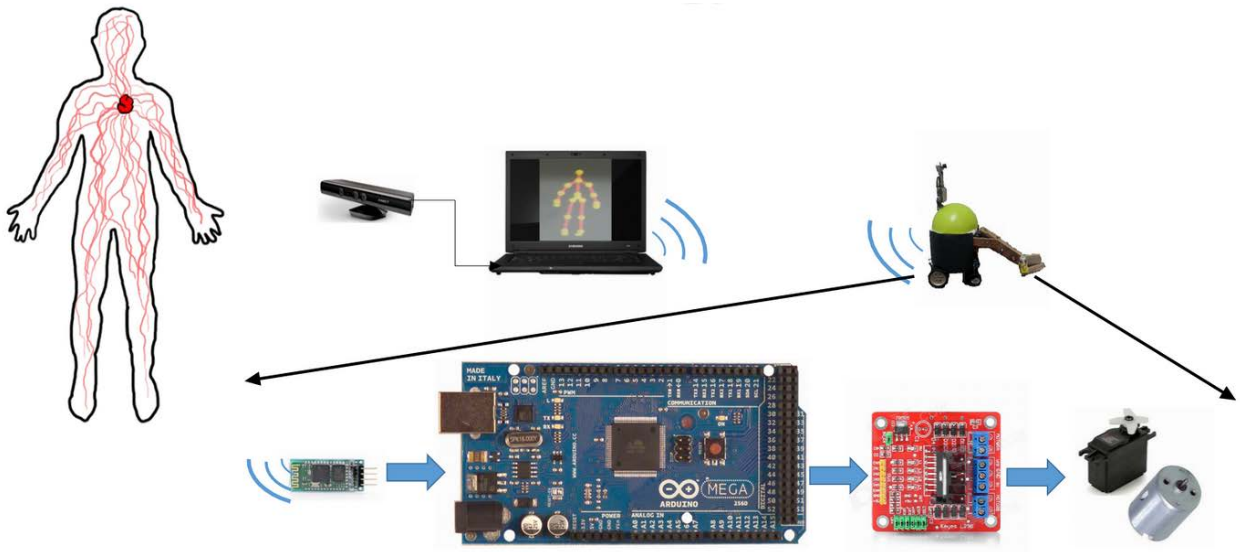


Figure 3.1: Block Diagram

Explanation:

3.1Sensor:

The sensor used over here is the Kinect sensor of Microsoft. This sensor has a Color VGA video camera, Depth sensor and Multi-array microphone we would be using the Color camera for getting the video of the person standing in front of the sensor which is basically a RGB camera and Depth sensor having an infrared projector and a monochrome CMOS (complimentary metal-oxide semiconductor) sensor work together to "see" the room in 3-D regardless of the lighting conditions, the IR projects the light to the person standing in-front of the sensor and the CMOS sensor receives the reflected signals.

Our body has different absorption and reflective properties so do the joints and bones have, the joints don't absorb much of the IR signals as compared to the bones so they reflect the signals in larger quantity than the bones, the Kinect sensor is smartly programmed to differentiate between them.

As it has differentiated between the bones and joints it makes a structure representing a total of 20 joints in our body and it joins those detected joints with the structure of bone, so the combined structure of joints and bones represents a human.

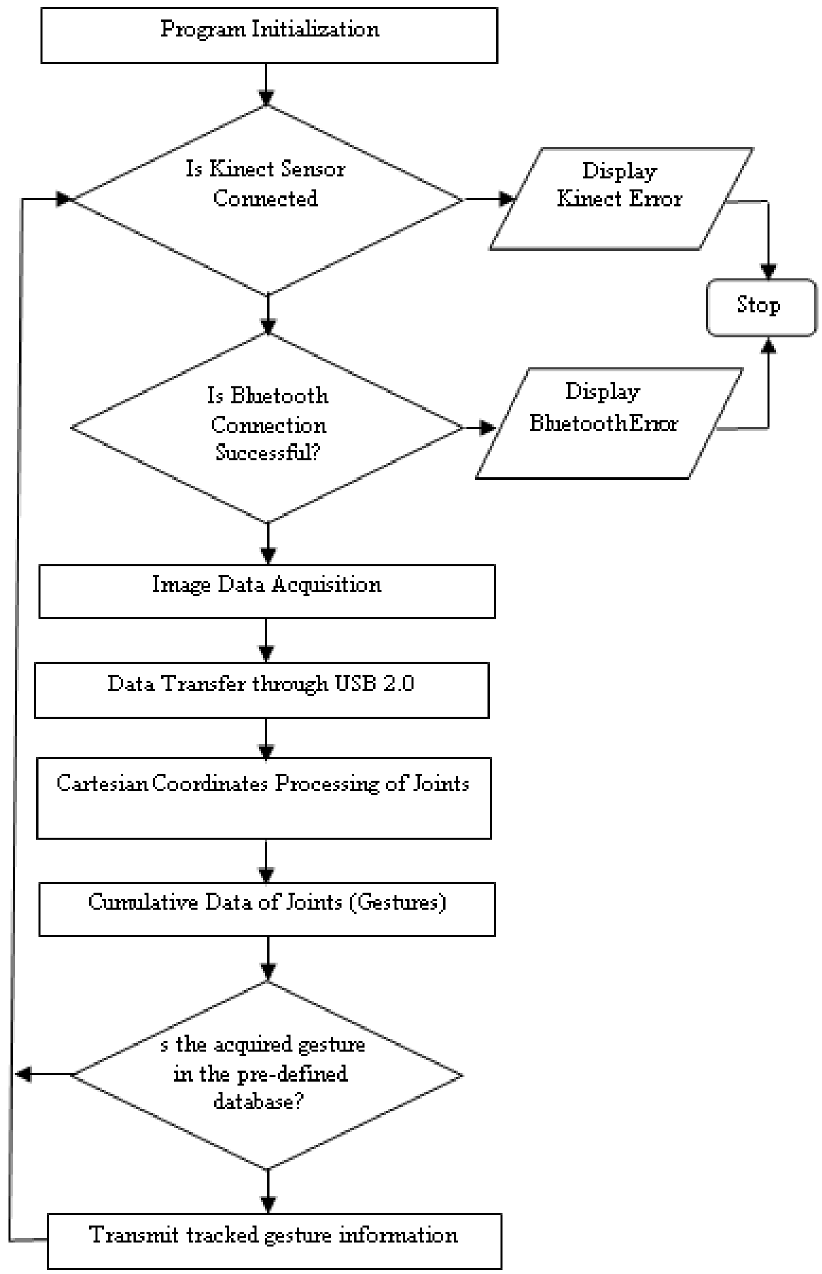


Figure 3.2: Flow chart for image data capturing

The person standing in front of the sensor is continuously tracked and as the person does some body movements these are captured using the Kinect sensor. The information is acquired in the form of positions of the various joints as explained earlier using the 3 dimensional Cartesian co-ordinate space.

3.2 Computer:

The tracked skeleton or captured image data by the Kinect sensor is transmitted serially through the USB port to a computer machine where the real time tracking of skeleton is displayed and the co-ordinates of the joints are processed. The human body movements or the gestures are required to be monitored and compared with the programmed gestures. The logical flow of the program shown above will help in understanding it in a better way.

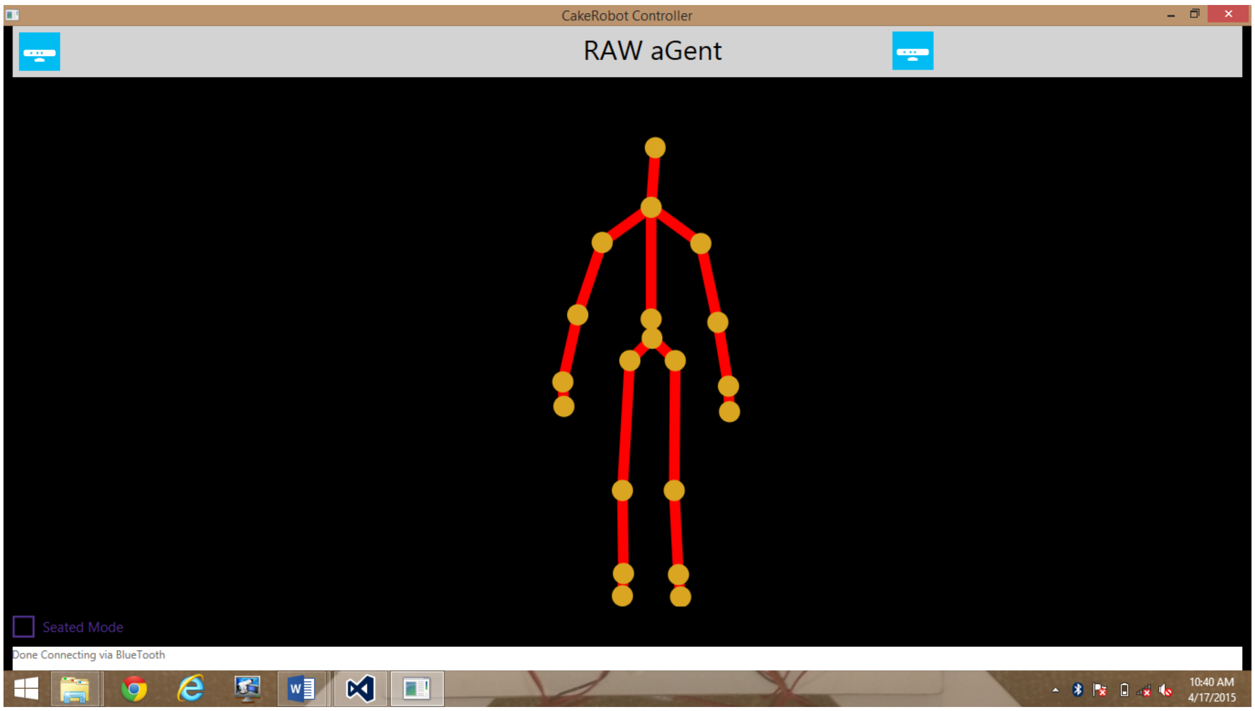


Figure 3.3: Sensor detecting the skeleton of person standing in front of Kinect

Flowchart shows the program flow for the image data acquisition. For Kinect to capture the motion of the body, we first need to initialize the sensor by creating a window environment for reproducing the captured image data on the display. Check for Kinect sensor if connected or not. If the Kinect sensor is not detected, it gives the error. Similarly, it checks for the Bluetooth and tries to connect to the robot and if robot not found, the program execution stops displaying the corresponding error.

Once the initialization code is executed, the sensor captures the images at the rate of 30 frames per second and is transferred as it is through the serial bus to the program which displays the skeleton.

The positions of the joints in the Cartesian coordinates space resulting in the gesture and comparing those gestures with the ones which are predefined in the program is processed simultaneously. If the gestures are matched, the corresponding information is transmitted via Bluetooth to the robot, else the tracking continues.

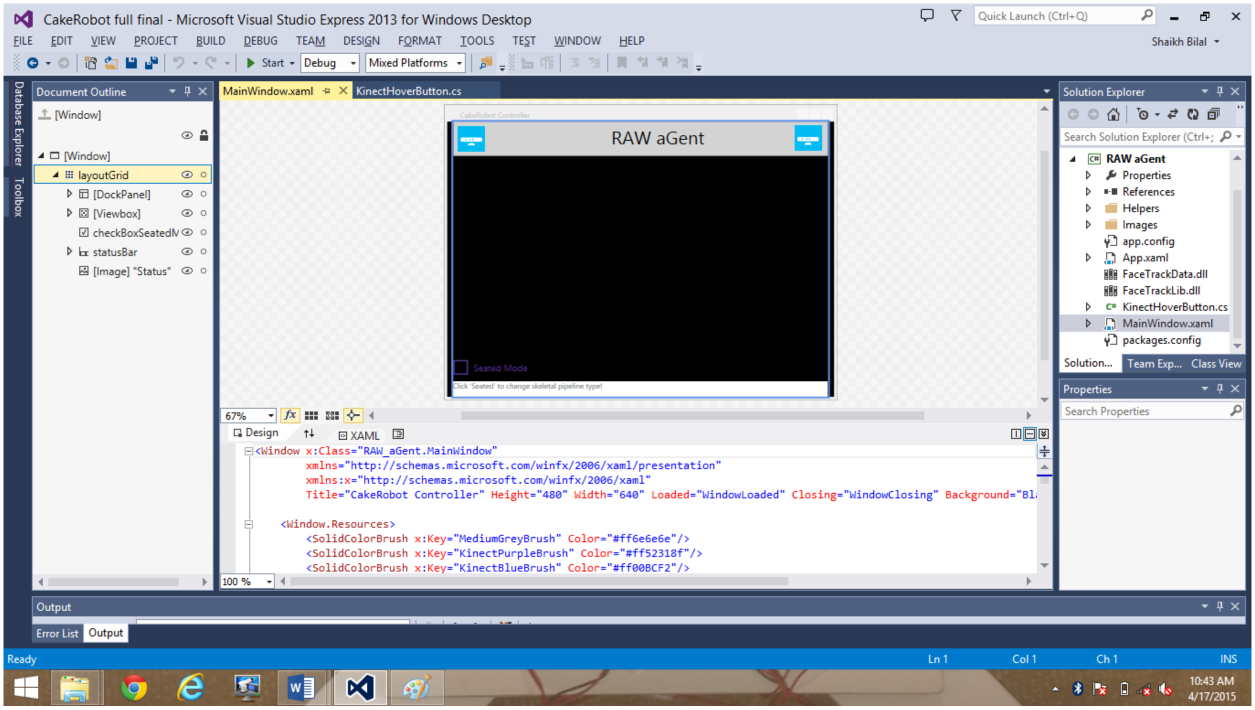


Figure 3.4: Window used for the detecting of skeleton

Processing of the gestures requires a program compatible with the Software Development Kit (SDK) of the Kinect.

3.3 Wireless Serial Transmitter:

After the processing is done in the laptop and verified it should be transmitted to the robot for further processing and real time operation for this the data is transmitted using the Bluetooth of the laptop which makes it a wireless control of robot possible

3.4 Wireless Serial Receiver:

The transmitted data from the Bluetooth of the laptop and is received by the Bluetooth module of the robot which is HC-05and is stored in buffer of the Bluetooth module connected to the robot. This stored data in the buffer is further passed on to the microcontroller board one by one for getting the sync of the operation at a baud rate of 9600.

3.5 Processing in Microcontroller:

The information about the recognized gesture is sent to the processing board in our case which is Arduino board data received via Bluetooth connected in the Arduino board is processed by the internal microcontroller of the board and various tasks are performed. The Arduino board is solely responsible for the control of both the Motion and the Arm.

To understand the concept of programming let's have a look at the flowchart which resembles the logical flow of programming.

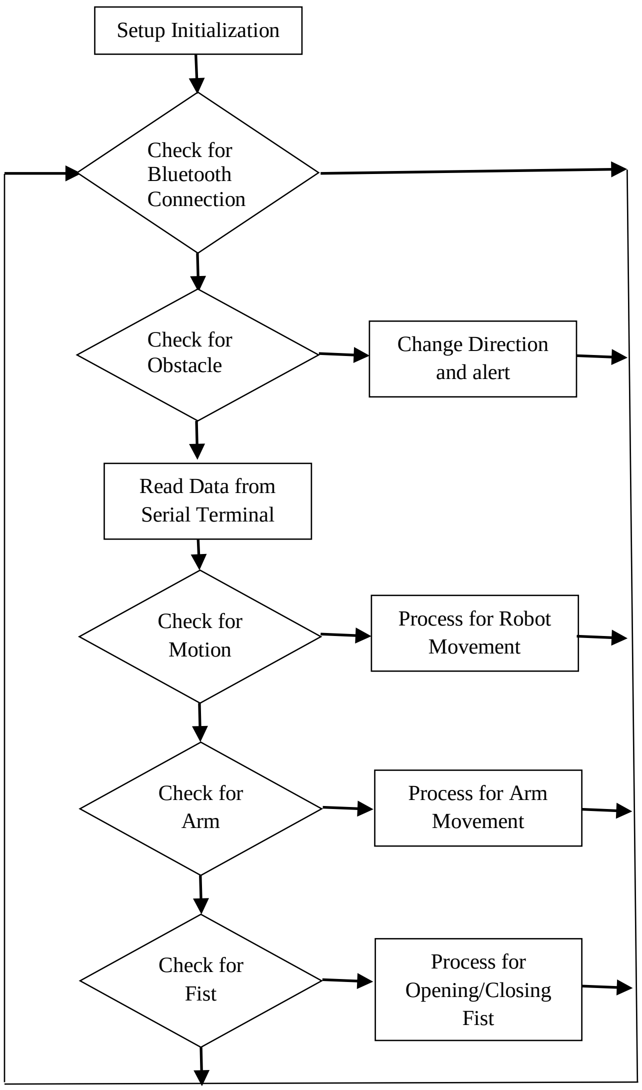


Figure 3.5: Flow chart for Robot

The programming of robot flows the similar way as that of Kinect sensor. Fig. 4 shows the flow chart of the same. The robot when initialized, the Bluetooth module scans for the devices for connection from the remote computer machine.

Once the connection between the two is established, the gesture information is received to the microcontroller. The microcontroller processes the received data and scans for the relevant operation to be performed. If the received information is for motion of the robot, the corresponding PWM signals are generated. If they belong to the arm control, the received angles are first quantized and then the axel rotation of the motors takes place and the arm replicates the arm of the user.

However, if the command is for hand, it opens and closes its jaws accordingly. There’s a provision for obstacle detection too using the ultrasonic sensor. It protects the robot from getting damaged by continuously monitoring its environment and maintaining safe distance.

We have used Arduino board for performing this operation which generates various PWM signals.

Chapter 4

ROBOT

4.1 Robot Motion

The Robot does the movement in the 4 basic directions which are:

4.1.1 Forward:

The above figure shows the movement of the robot in the forward direction. This had to be defined for allowing the Robotic arm to be made portable.

Figure 4.1: Motion Forward

4.1.2 Reverse:

The above figure shows the movement of the robot in the backward direction. This had to be defined for allowing the Robotic arm to be made portable.

Figure 4.2: Motion Reverse

4.1.3 Right:

Figure 4.3: Motion Right

The above figure shows the movement of the robot in the right direction. This had to be defined for allowing the Robotic arm to be made portable.

4.1.4 Left:

The above figure shows the movement of the robot in the left direction. This had to be defined for allowing the Robotic arm to be made portable.

Figure 4.4: Motion Left

4.2 Robotic ARM

4.2.1 Base:

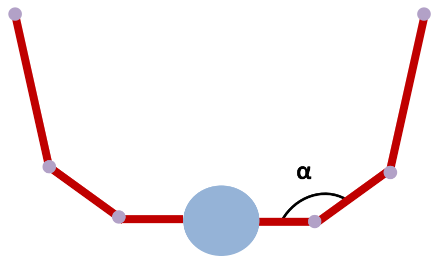


Figure 4.5: ARM Base

The above figure shows the angle between the shoulder and the elbow bone in horizontal plane, is named as the base angle denoted by α as shown. This will help in moving the arm in right or left direction.

4.2.2 Shoulder:

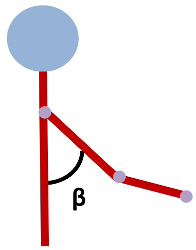


Figure 4.6: ARM Shoulder

The above figure shows the second angle denoted by β, is the one between the body center and the shoulder bone as shown. It will move the arm in vertical direction.

4.2.3 Elbow:

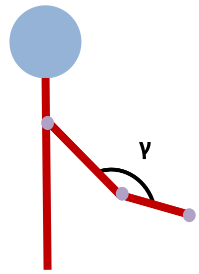


Figure 4.7: ARM Elbow

The above figure shows the angle formed by the elbow and shoulder bone, denoted by γ. This will lift the jaw up and down helping in picking up objects and placing them where you want.

4.2.4 ARM

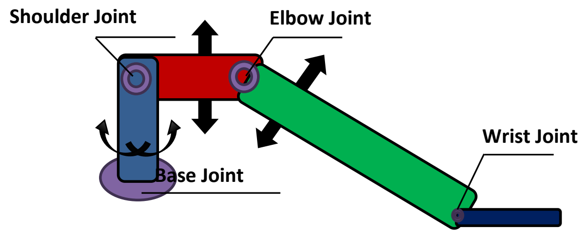
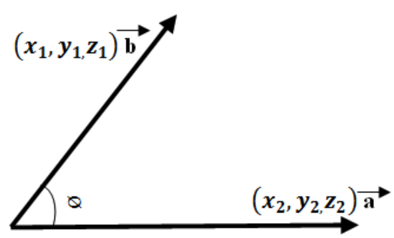


Figure 4.8: ARM

The above figure shows the complete mechanical arm after combining all the components. There is a high probability that the human arm may suddenly change the speed and direction or may go out of bounds causing the jerks in the robotic arm. This situation will not only cause damage to the robot but also there will be an unsafe feeling to the user. To avoid such scenarios, we have specified movement ranges by quantizing the angles received through the Kinect (Table I). It will assure smooth functioning and operation of the robot increasing its shelf life.



Two vectors with 3D coordinates

$$\varnothing = \cos^{-1}\left(\frac{x_1 . x_2 + y_1 . y_2 + z_1 . z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} . \sqrt{x_2^2 + y_2^2 + z_2^2}}\right)$$

Figure 4.9: ARM Vector Calculation

The above formulae shows the equation for the calculation of the angles for the arm along with the basic trigonometry structure shown, which is essential for the movement of the Robotic Arm in the 3-dimentional space

Chapter 5

Add-On

5.1 Remote Surveillance:

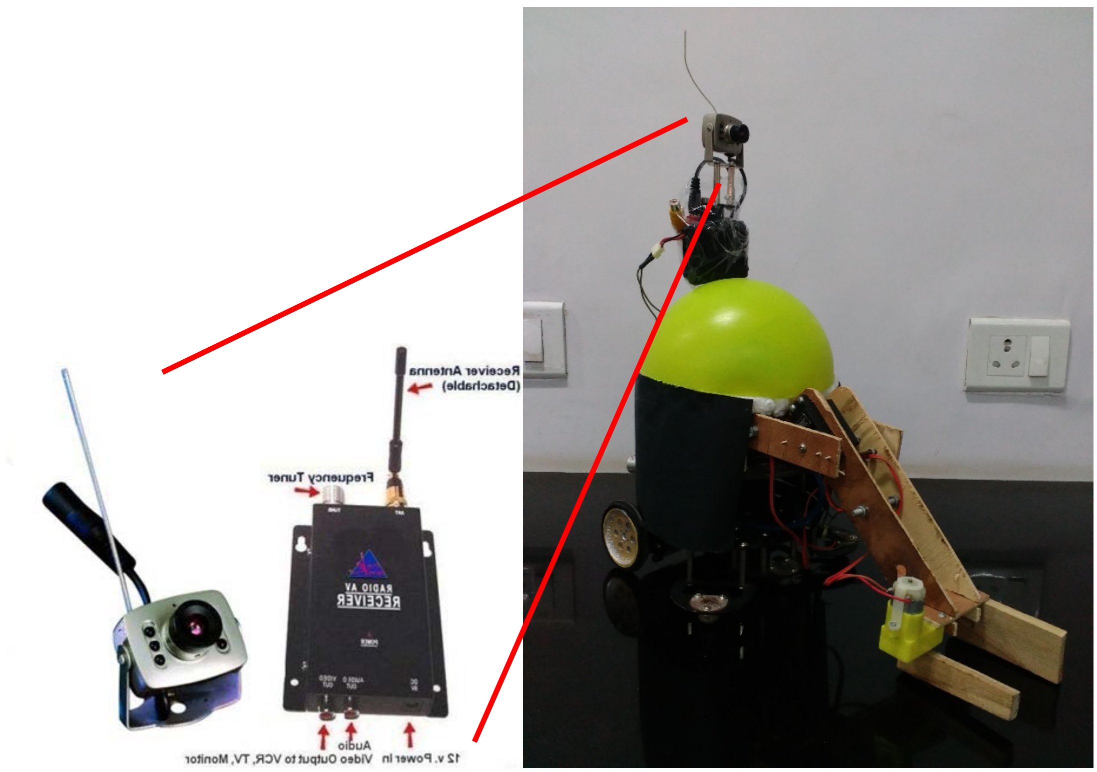


Figure 5.1: Remote Surveillance

We have made the robot with the intention that it would go on places as it is being controlled wirelessly and as it would go places there is a possibility that it is not in the vision of the controller always. Our robot can also be used for remote surveillance purpose as we have embedded a wireless Radio Frequency (RF) camera for the same purpose

5.2 Obstacle Detection:

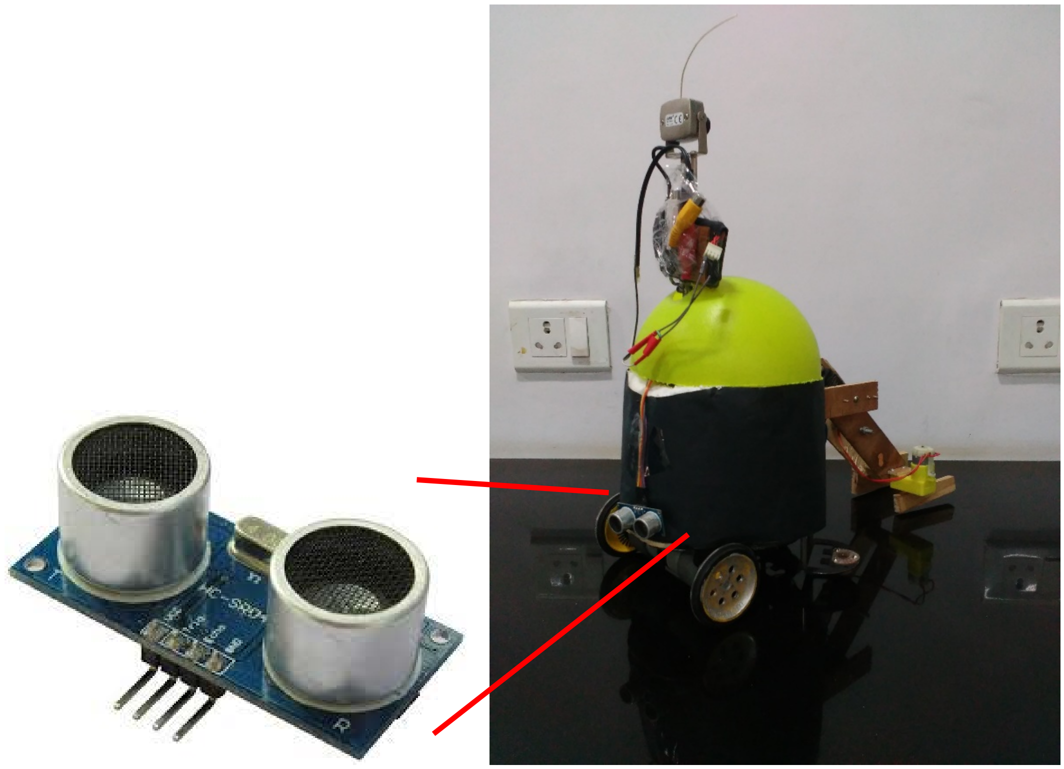


Figure 5.2: Obstacle Detection

We have made the robot with the intention that it would go on places as it is being controlled wirelessly and as it would go places there is a possibility that it is not in the vision of the controller always or may bump onto something. Our robot can also be used for remote surveillance purpose as we have used a wireless Radio Frequency (RF) camera the camera would be giving us the front look but what about the back side the robot could crash. To avoid this we have embedded an obstacle detector which is basically an Ultrasonic Sensor.

Chapter 6

SUPPLY PCB

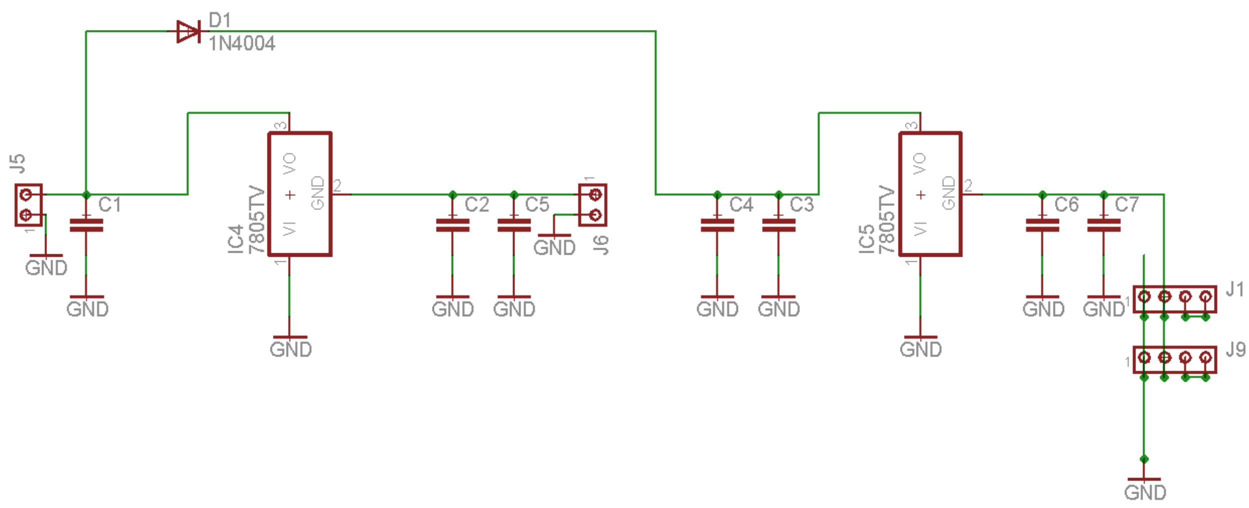


Figure 6.1: Schematic of Supply PCB

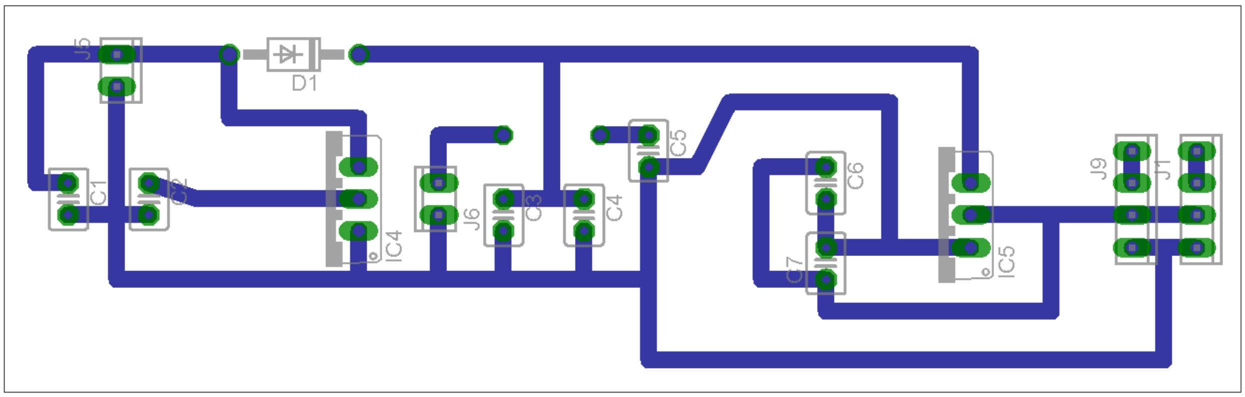


Figure 6.2: Board structure of Supply PCB

For powering of the Servos and the Arduino board we had customized the supply board depending upon the current and voltage requirements of the project.

The board is able to derive the Arduino Mega board and 2 servos with the output voltage as 5-V DC and 800-mAmps as output current.

Chapter 7

SOFTWARE DISCRIPTION

7.1 C Sharp

7.1.1 Introduction

C# or C sharp is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative.

C# is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages to be used on different computer platforms and architectures.

C# is an object-oriented programming language. In Object-Oriented Programming methodology, a program consists of various objects that interact with each other by means of actions. The actions that an object may take are called methods. Objects of the same kind are said to have the same type or, more often, are said to be in the same class.

7.1.2 Overview

C# (pronounced as *see sharp*) is a modern, general-purpose, object-oriented programming language developed by Microsoft and approved by Ecma and ISO.

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional,generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 5.0, which was released on August 15, 2012.

7.1.3 Features

- The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.
- The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.
- The language is intended for use in developing software components suitable for deployment in distributed environments.
- Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.
- Support for internationalization is very important.
- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

7.1.4 Advantages

- Modern, general-purpose programming language
- Object oriented.
- Component oriented.
- Easy to learn.
- Structured language.
- It produces efficient programs.
- It can be compiled on a variety of computer platforms.
- Part of .Net Framework.

7.2 Visual Studio

7.2.1 Introduction

As new technology are emerging day by day, so we thought of using the one which is said to be the advanced in the field of programming packages, the Visual Studio Express 2013 is being used by us cause it gives the cross platform of programming and the Windows Presentation Foundation (WPF), WPF provides developers with a unified programming model for building rich Windows smart client user experiences that incorporate UI, media, and documents. So we thought of using the Microsoft Visual Studio Express 2013 for making our project user interface (UI) more user friendly and interactive

7.2.2 Overview

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source-control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++ and C++/CLI (via Visual C++),VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010[6]). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Microsoft provides "Express" editions of its Visual Studio at no cost.

7.2.3 Features

7.2.3.1 Code editor

Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for not only variables, functions and methods but also language constructs like loops and queries. IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing web sites and web applications. Auto complete suggestions are popped up in a modeless list box, overlaid on top of the code editor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it. The code editor is used for all supported languages.

The Visual Studio code editor also supports setting bookmarks in code for quick navigation. Other navigational aids include collapsing code blocks and incremental search, in addition to normal text search and regex search. The code editor also includes a multi-item clipboard and a task list. The code editor supports code snippets, which are saved templates for repetitive code and can be inserted into code and customized for the project being worked on. A management tool for code snippets is built in as well. These tools are surfaced as floating windows which can be set to automatically hide when unused or docked to the side of the screen. The Visual Studio code editor also supports code refactoring including parameter reordering, variable and method renaming, interface extraction and encapsulation of class members inside properties, among others.

Visual Studio features background compilation (also called incremental compilation). As code is being written, Visual Studio compiles it in the background in order to provide feedback about syntax and compilation errors, which are flagged with a red wavy underline. Warnings are marked with a green underline. Background compilation does not generate executable code, since it requires a different compiler than the one used to generate executable code. Background compilation was initially introduced with Microsoft Visual Basic but has now been expanded for all included languages.

7.2.3.2 Debugger

Visual Studio includes a debugger that works both as a source-level debugger and as a machine-level debugger. It works with both managed code as well as native code and can be used for debugging applications written in any language supported by Visual Studio. In addition, it can also attach to running processes and monitor and debug those processes. If source code for the running process is available, it displays the code as it is being run. If source code is not available, it can show the disassembly. The Visual Studio debugger can also create dump as well as load them later for debugging. Multi-threaded programs are also supported. The debugger can be configured to be launched when an application running outside the Visual Studio environment crashes.

The debugger allows setting breakpoints (which allow execution to be stopped temporarily at a certain position) and watches (which monitor the values of variables as the execution progresses). Breakpoints can be conditional, meaning they get triggered when the condition is met. Code can be stepped over, i.e., run one line (of source code) at a time. It can either *step into* functions to debug inside it, or *step over* it, i.e., the execution of the function body isn't available for manual inspection. The debugger supports *Edit and Continue*, i.e., it allows code to be edited as it is being debugged (32 bit only; not supported in 64 bit). When debugging, if the mouse pointer hovers over any variable, its current value is displayed in a tooltip ("data tooltips"), where it can also be modified if desired. During coding, the Visual Studio debugger lets certain functions be invoked manually from the immediate tool window. The parameters to the method are supplied at the immediate window.

7.2.4 Designer

7.2.4.1 Windows Forms Designer

The Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box, grid view, etc.) can be bound to data sources like databases or queries. Data-bound controls can be created by dragging items from the Data Sources window onto a design surface.[31] The UI is linked with code using an event-driven programming model. The designer generates either C# or VB.NET code for the application.

7.2.4.2 WPF Designer

The WPF designer, codenamed *Cider*, was introduced with Visual Studio 2008. Like the Windows Forms designer it supports the drag and drop metaphor. It is used to author user interfaces targeting Windows Presentation Foundation. It supports all WPF functionality including data binding and automatic layout management. It generates XAML code for the UI. The generated XAML file is compatible with Microsoft Expression Design, the designer-oriented product. The XAML code is linked with code using a code-behind model.

7.2.4.3 Web designer/development

Visual Studio also includes a web-site editor and designer that allow web pages to be authored by dragging and dropping widgets. It is used for developing ASP.NET applications and supports HTML, CSS and JavaScript. It uses a code-behind model to link with ASP.NET code. From Visual Studio 2008 onwards, the layout engine used by the web designer is shared with Microsoft Expression Web. There is also ASP.NET MVC support for MVC technology as a separate download and ASP.NET Dynamic Data project available from Microsoft.

7.3 Kinect SDK

Microsoft announced that it would release a non-commercial Kinect software development kit (SDK) for Windows, which was released for Windows 7 in 12 countries. The SDK includes Windows 7 compatible PC drivers for Kinect device. It provides Kinect capabilities to developers to build applications with C++, C#, or Visual Basic by using Microsoft Visual Studio 2010 and includes following features:

1. Raw sensor streams: Access to low-level streams from the depth sensor, colour camera sensor, and four-element microphone array.
2. Skeletal tracking: The capability to track the skeleton image of one or two people moving within Kinect's field of view for gesture-driven applications.
3. Advanced audio capabilities: Audio processing capabilities include sophisticated acoustic noise suppression and echo cancellation, beam formation to identify the current sound source, and integration with Windows speech recognition API.
4. Sample code and Documentation.

Chapter 8
HARDWARE DISCRIPTION

8.1 MICROSOFT KINECT

8.1.1 Overview

The key enabling technology is human body- language understanding; the computer must first understand what a user is doing before it can respond. This has always been an active research field in computer vision, but it has proven formidably difficult with video cameras. The Kinect sensor lets the computer directly sense the third dimension (depth) of the user and the environment, making the task much easier. It also understands when users talk, knows who they are when they walk up to it, and can interpret their movements and translate them into a format that developers can use to build new experiences. With its wide availability and low cost, many researchers and practitioners in computer science, electronic engineering, and robotics are leveraging the sensing technology to develop creative new ways to interact with machines and to perform other tasks.

Figure 8.1: Kinect Sensor

8.1.2 Technology

The innovative technology behind Kinect is a combination of hardware and software contained within the Kinect sensor accessory that can be added to any existing Xbox 360. The Kinect sensor is a flat black box that sits on a small platform, placed on a table or shelf near the television you're using with your Xbox 360. Newer Xbox 360s have a Kinect port from which the device can draw power, but the Kinect sensor comes with a power supply at no additional charge for users of older Xbox 360 models. For a video game to use the features of the hardware, it must also use the proprietary layer of Kinect software that enables body and voice recognition from the Kinect sensor.

8.1.3 Structure

There's a trio of hardware innovations working together within the Kinect sensor:

- **Color VGA video camera -** This video camera aids in facial recognition and other detection features by detecting three color components: red, green and blue. Microsoft calls this an "RGB camera" referring to the color components it detects.

- **Depth sensor** - An infrared projector and a monochrome CMOS (complimentary metal-oxide semiconductor) sensor work together to "see" the room in 3-D regardless of the lighting conditions.

- **Multi-array microphone** - This is an array of four microphones that can isolate the voices of the players from the noise in the room. This allows the player to be a few feet away from the microphone and still use voice controls.

A further look at the technical specifications for Kinect reveal that both the video and depth sensor cameras have a 640 x 480-pixel resolution and run at 30 FPS (frames per second). The specifications also suggest that you should allow about 6 feet (1.8 meters) of play space between you and the Kinect sensor, though this could vary depending on where you put the sensor.

The Kinect hardware, though, would be nothing without the breakthrough software that makes use of the data it gathers. Leap forward to the next page to read about the "brain" behind the camera lens.

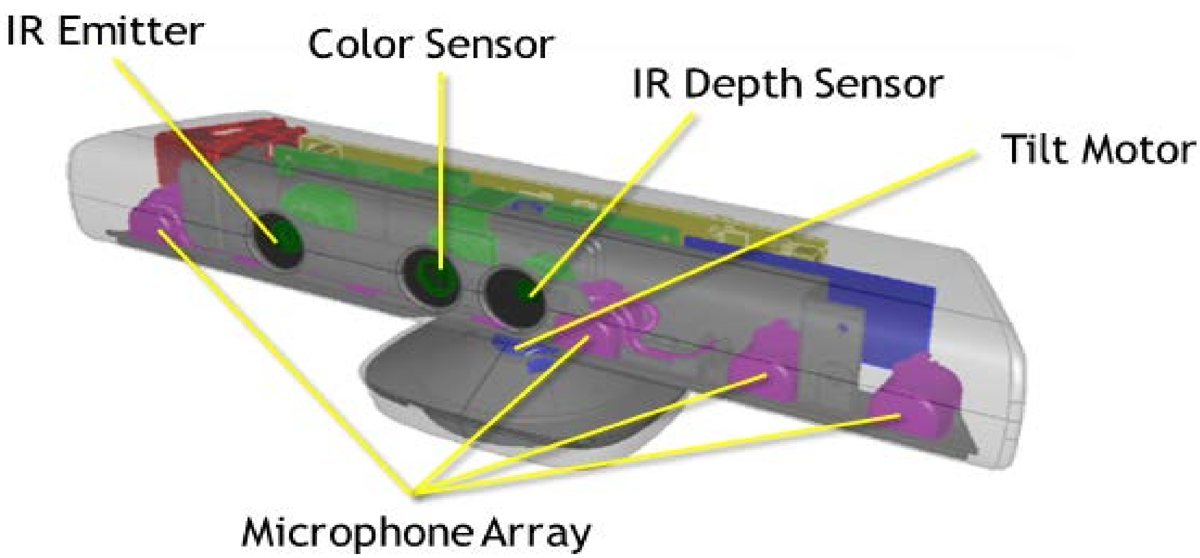


Figure 8.2: Kinect internal structure

8.1.4 Working

The Kinect sensor incorporates several advanced sensing hardware. Most notably, it contains a depth sensor, a color camera, and a four-microphone array that provide full-body 3D motion capture, facial recognition, and voice recognition capabilities .Figure 1b shows the arrangement of the infrared (IR) projector, the color camera, and the IR camera. The depth sensor consists of the IR projector combined with the IR camera, which is a monochrome complementary metal- oxide semiconductor (CMOS) sensor. The IR projector is an IR laser that passes through a diffraction grating and turns into a set of IR dots. The IR projector is an IR laser that passes through a diffraction grating and turns into a set of IR dots. The depth value is encoded with gray values; the darker a pixel, the closer the point is to the camera in space. The black pixels indicate that no depth values are available for those pixels.

What we have used in the Kinect sensor is its Skeleton tracking and the Depth Sensor.

8.1.5 Depth Sensor

The Depth sensor is used by the Kinect sensor to positionize the joints which means its 3D coordinates, these 3D coordinates will be used in the programming of the gesture specific operations which will be performed by the robot.

The basic principle behind the Kinect depth sensor is emission of an IR pattern and the simultaneous image capture of the IR image with a (traditional) CMOS camera that is fitted with an IR-pass filter. The image processor of the Kinect uses the relative positions of the dots in the pattern to calculate the depth displacement at each pixel position in the image. It should be noted that the actual depth values are distances from the camera-laser plane rather than distances from the sensor itself. As such, the depth sensor can be seen as a device that returns (x, y, z)-coordinates of 3D objects.

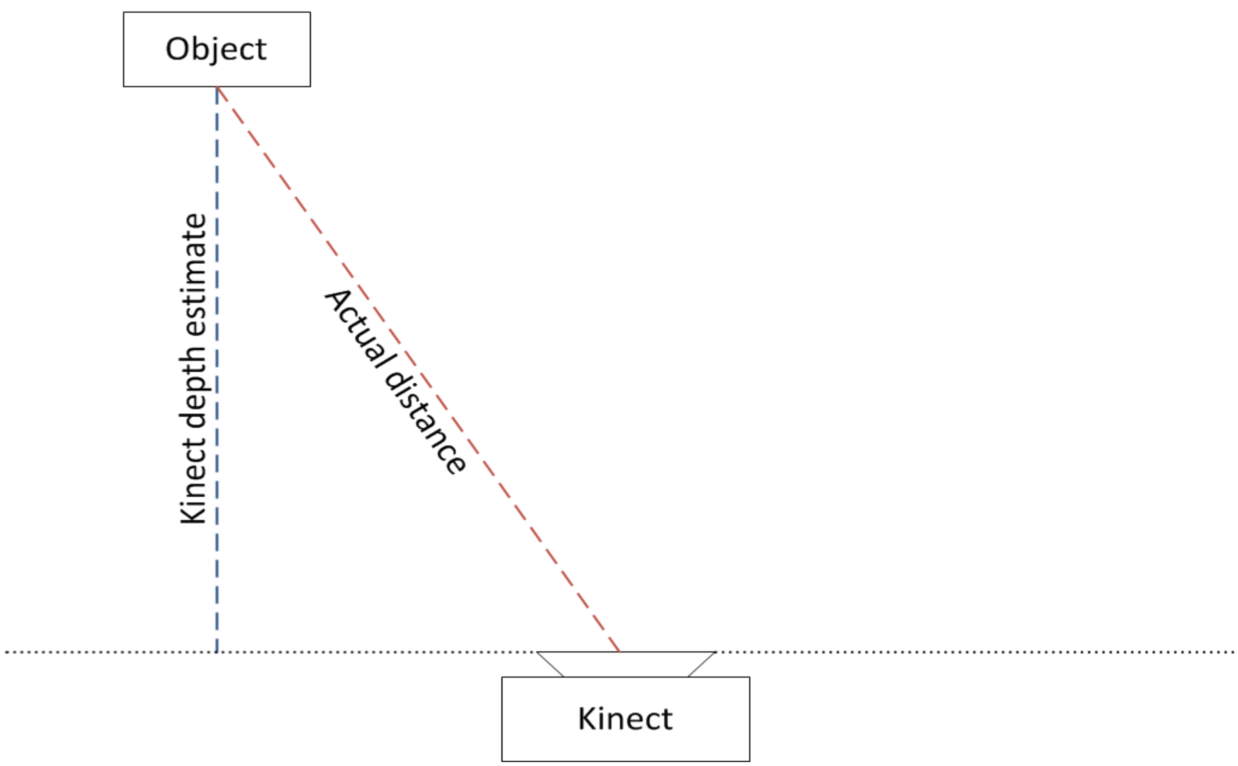


Figure 8.3: Depth view

8.1.6 Joint Tracking

We are planning to use the skeleton tracking a feature of the Kinect sensor, the sensor detects 20 joints in the human body who so ever is standing in front of it. 10 points in the upper half and the next 10 points in the lower half of the body. These points will be used as the reference for the programming of the Kinect sensor for our application.

The innovation behind Kinect hinges on advances in skeletal tracking. The operational envelope demands for commercially viable skeletal tracking are enormous. Simply put, skeletal tracking must ideally work for every person on the planet, in every house- hold, without any calibration. Entire sets of dimensions are necessary to describe unique individuals, including size, shape, hair, clothing, motions, and poses. In skeletal tracking, a human body is represented by a number of joints representing body parts such as head, neck, shoulders, and arm. Each joint is represented by its 3D coordinates. Figure illustrates the whole pipeline of Kinect skeletal tracking. The first step is to perform per-pixel, body-part classification. The second step is to hypothesize the body joints by finding a global centroid of probability mass (local modes of density) through mean shift. The final stage is to map hypothesized joints to the skeletal joints and fit a skeleton by considering both temporal continuity and prior knowledge from skeletal train data.

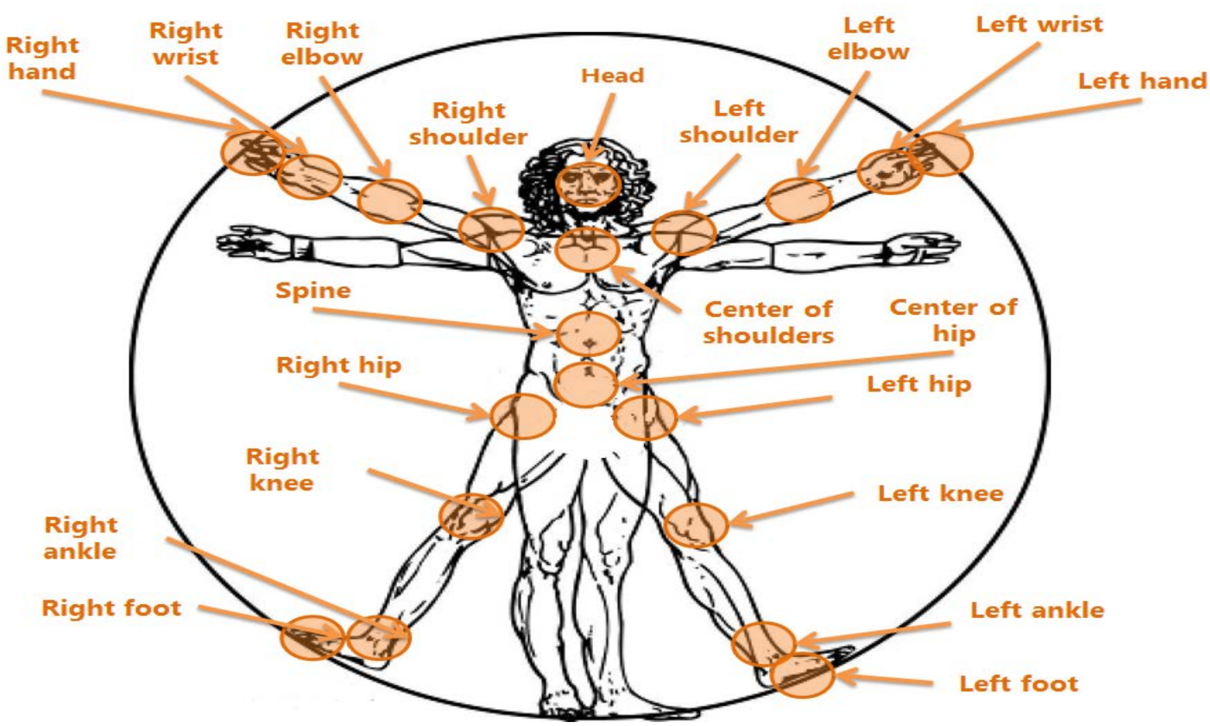


Figure 8.4: Joint Tracking

8.1.7 Skeletal Tracking

Skeletal Tracking allows Kinect to recognize people and follow their actions. Using the infrared (IR) camera, Kinect can recognize up to six users in the field of view of the sensor. Of these, up to two users can be tracked in detail. An application can locate the joints of the tracked users in space and track their movements over time.

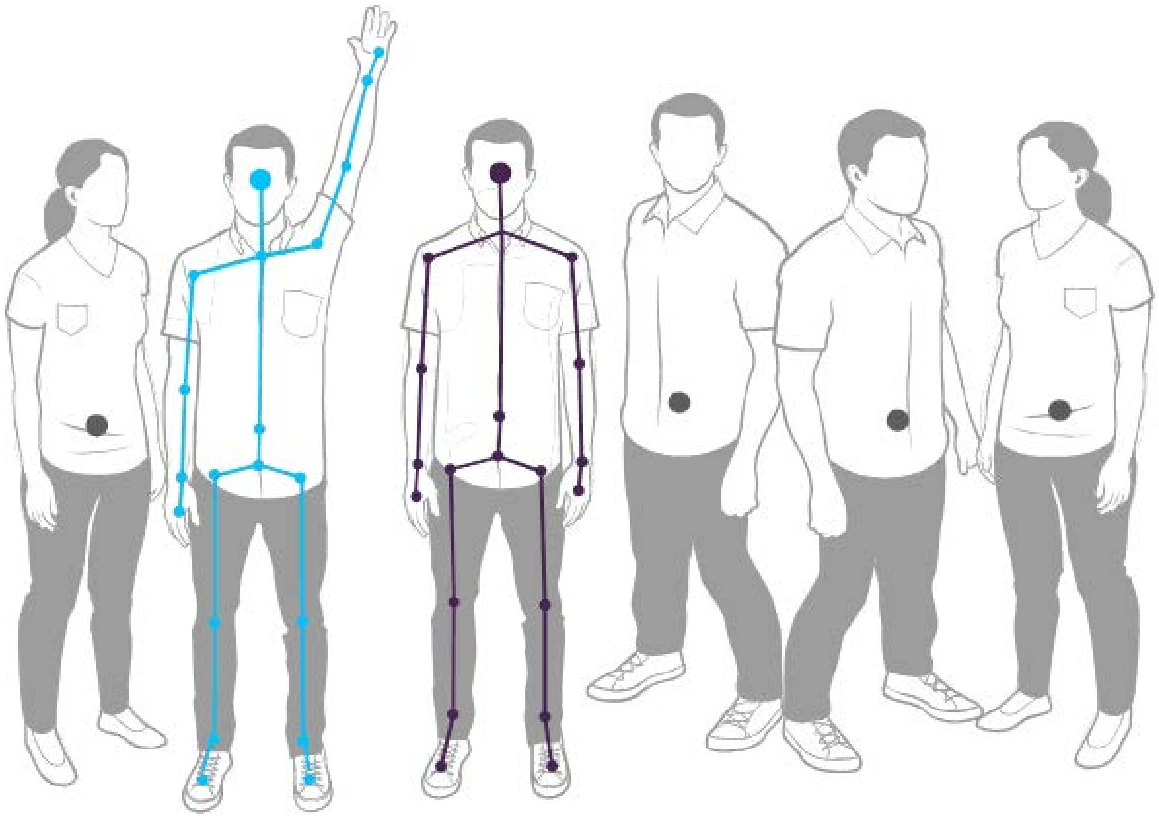


Figure 8.5: Skeletal Tracking (Kinect can recognize six people and track two)

Skeletal Tracking is optimized to recognize users standing or sitting, and facing the Kinect; sideways poses provide some challenges regarding the part of the user that is not visible to the sensor.

To be recognized, users simply need to be in front of the sensor, making sure the sensor can see their head and upper body; no specific pose or calibration action needs to be taken for a user to be tracked.

8.7.1.1 Field of View

Kinect field of view of the users is determined by the settings of the IR camera, which are set with the Depth Range Enumeration.

In default range mode, Kinect can see people standing between 0.8 meters (2.6 feet) and 4.0 meters (13.1 feet) away; users will have to be able to use their arms at that distance, suggesting a practical range of 1.2 to 3.5 meters. For more details, see the k4w_hig_main.

8.7.1.2 Skeletal Tracking Precision and Multiple Kinect Sensors

The infrared emitter of a Kinect sensor projects a pattern of infrared light. This pattern of light is used to calculate the depth of the people in the field of view allowing the recognition of different people and different body parts. If you use more than one Kinect sensor to illuminate the target area, you may notice a reduction in the accuracy and precision of skeletal tracking due to interference with the infrared light sources. To reduce the possibility of interference, it is recommended that no more than one Kinect sensor (or infrared light source) points to a field of view where skeletal tracking is being done.

In near range mode, Kinect can see people standing between 0.4 meters (1.3 feet) and 3.0 meters (9.8 feet); it has a practical range of 0.8 to 2.5 meters.

Skeleton tracking a new modality, call seated mode for tracking user skeletons.

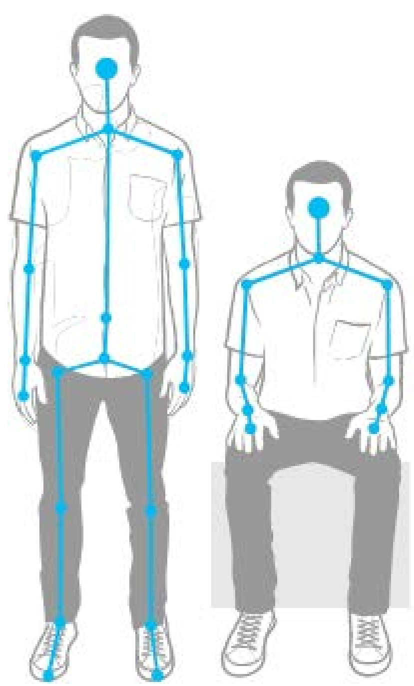


Figure 8.6: Skeletal tracking types (Default and Near Mode)

8.2 ARDUINO

8.2.1 Why Arduino

As we all know that Arduino community is Open Source which means open hardware and open software and also our Institute has taken the initiative to promote Open Source Community in collaboration with Indian Institute of Technology – Bombay (IIT-B)

Open Software we are all aware of that we can make changes in the programming of the software and use it as per our requirements or can be modified depending on the application, in other words we can tailor made a software according to the use of our application without any notification to the source company.

But what is Open Hardware, hopefully listing it first time. And how the Hardware can be open source as we are purchasing for it. We all also have the misconception that the Open Source things are free of cost. But over here the hardware is open source because the company has made the Schematic file .sch, the Board file .brd, and all the other related documents and sources available freely on the official website that also in the most common formats like EAGLE pcb Software, .pdf and all.

The out of the box thing is that we can actually modify the circuit of the board and make or generate our own Tailor made PCB application specific without any notification to the Arduino Inc.

8.2.2 Why Arduino MEGA 2560

There are many hell lot of Arduino boards available like Arduino UNO, Arduino NEO, Arduino Micro, Arduino Esplora and the list goes on and on.

But the main drawback of these above mentioned boards were that they had less number of ports and the processor size is small i.e. Flash Memory which is used to store the program.

We thought of using a board in which there would be no limitations on speed, memory and ports, so during our surfing we came across a board named Arduino MEGA 2560 and we thought it would be the best as pre what we have planned in the application of our project, as it is having 15 PWM out pins over 20 Digital out pins Flash of 256 KB and a high processing speed 16 MHz, so we thought it is going to be the best board for our project.

8.2.3 Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila. The Mega 2560 is an update to the Arduino Mega, which it replaces.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 boards) programmed as a USB-to-serial converter. Revision 2 of the Mega2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- Pin-out: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

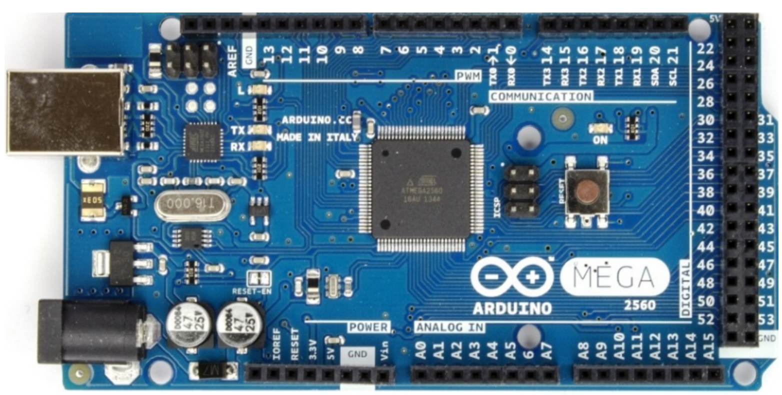


Figure 8.7: Arduino Mega 2560 R3 Front

8.2.4 Schematic, Reference Design & Pin Mapping

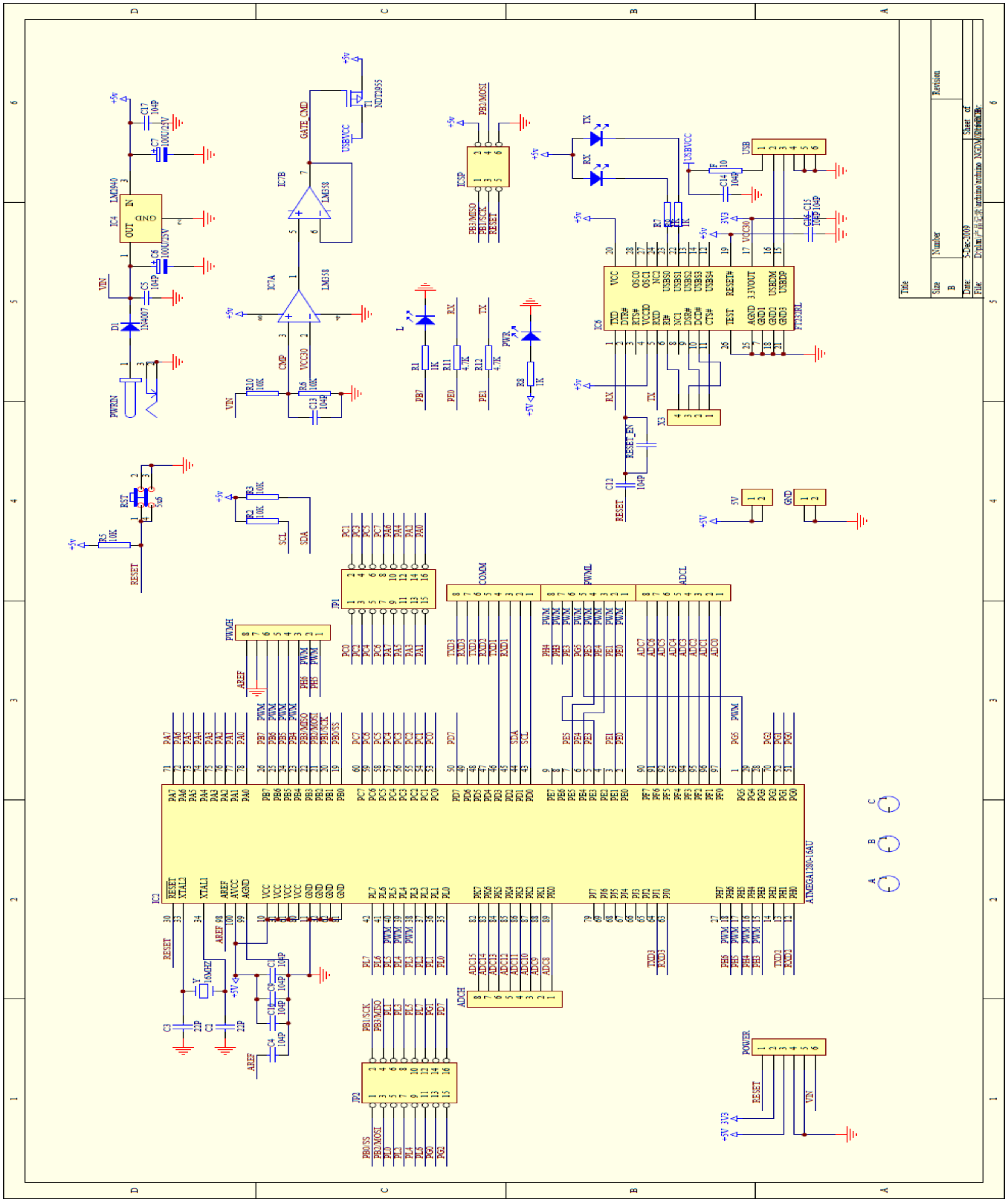


Figure 8.8: Arduino Mega schematic

8.2.5 Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

8.2.6 Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V. this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
  - 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
  - GND. Ground pins.
  - IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

8.2.7 Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

8.2.8 Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2(ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

8.3 Bluetooth Module

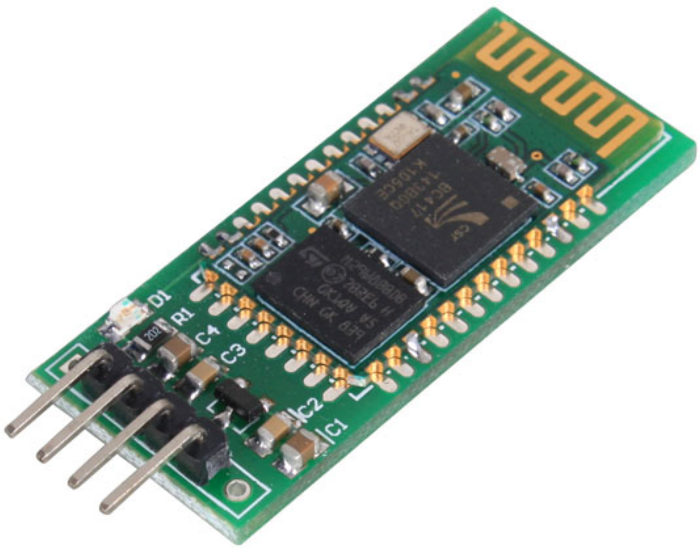


Figure 8.9: Bluetooth Module

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

HC-05 is a class-2 Bluetooth module with Serial Port Profile, which can configure as either Master or slave. A Drop-in replacement for wired serial connections, transparent usage. You can use it simply for a serial port replacement to establish connection between MCU, PC to your embedded project and etc.

8.3.1 Specification:

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm

8.3.2 Schematic

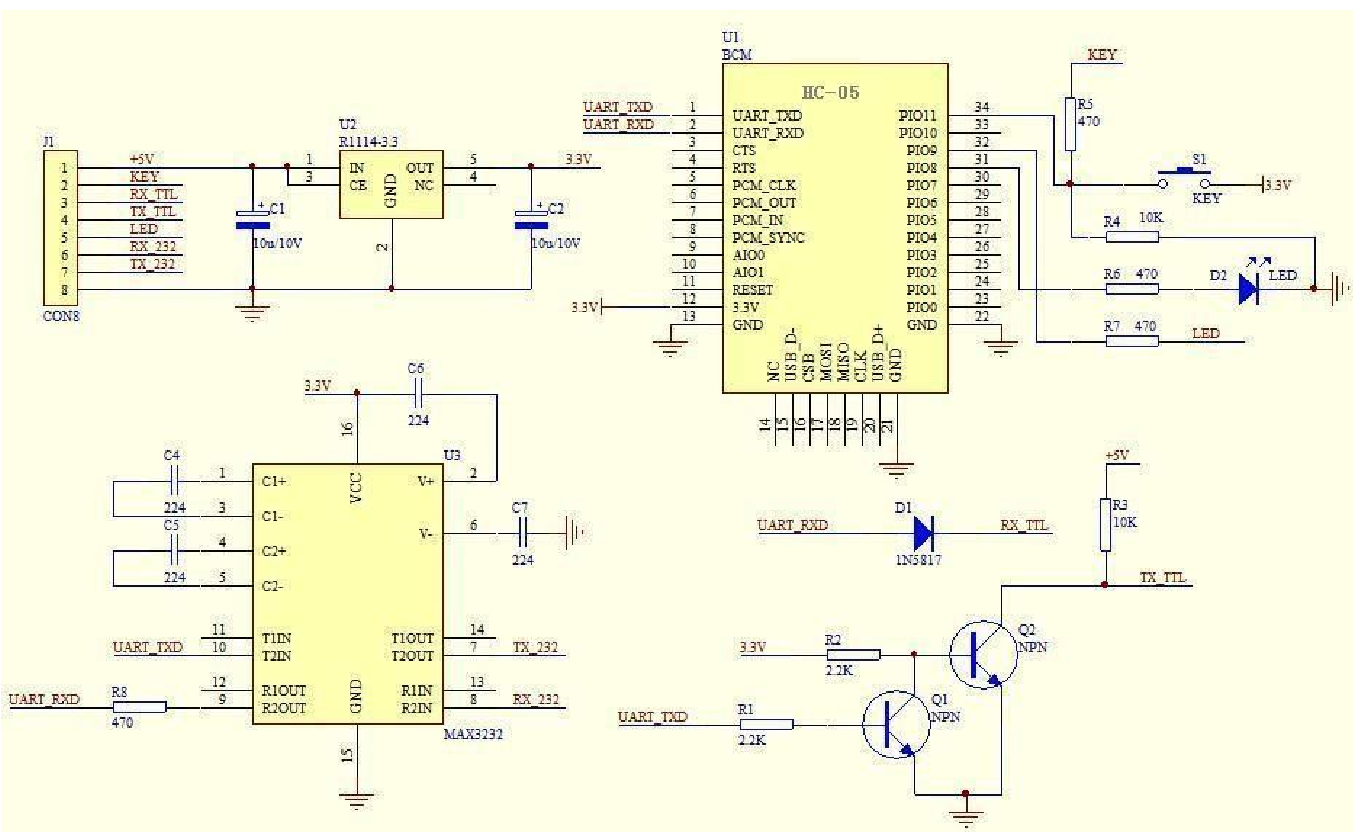


Figure 8.10: Bluetooth Module schematic

8.4 Motor Driver

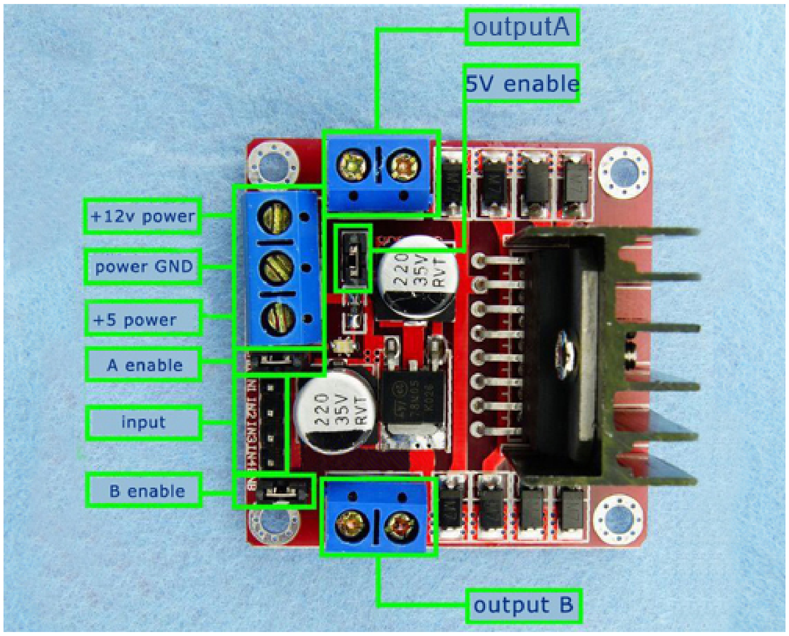


Figure 8.11: Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions.

It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc.

The circuit incorporates 4 direction LEDs (2 per motor), a heat sink, screw-terminals, as well as eight Schottky EMF-protection diodes. Two high-power current sense resistors are also incorporated which allow monitoring of the current drawn on each motor through your microcontroller.

An on-board user-accessible 5V regulator is also incorporated which can also be used to supply any additional circuits requiring a regulated 5V DC supply of up to about 1A.

The circuit also offers a bridged mode of operation allowing bidirectional control of a single motor of up to about 4A.

8.4.1 **Features:**

- Motor supply: 6 to 35 VDC
- Control Logic: Standard TTL Logic Level
- Output Power: Up to 2 A each
- Current Sense Outputs
- Onboard Power Resistors Provided for Current Limit
- Enable and Direction Control Pins
- External Diode Bridge Provided for Output
- Heat sink for IC
- Power-On LED indicator
- 4 Direction LED indicators

Dimensions: 3 x 2.5"

8.4.2 Schematic:

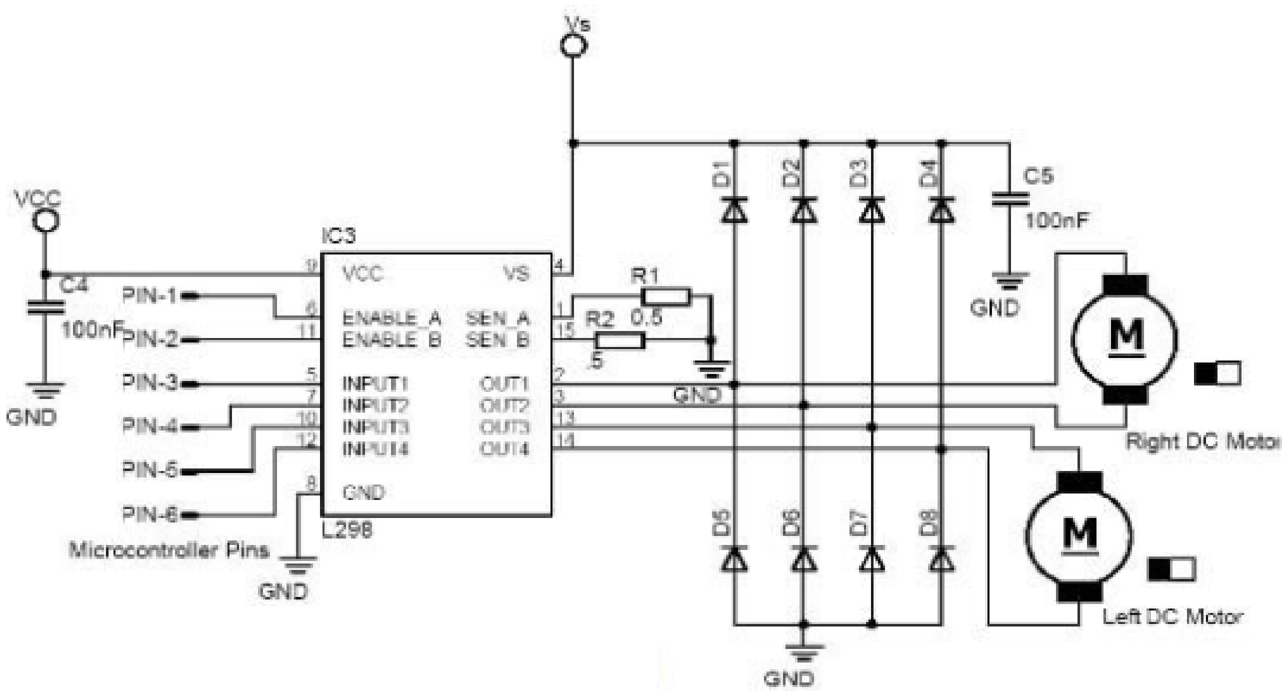


Figure 8.12: Motor Driver schematic

8.5 Ultrasonic Sensor

Figure 8.13: Ultrasonic sensor

Ultrasonic transducers are transducers that convert ultrasound waves to electrical signals or vice versa. Those that both transmit and receive may also be called ultrasound transceivers; many ultrasound sensors besides being sensors are indeed transceivers because they can both sense and transmit. These devices work on a principle similar to that of transducers used in radar and sonar systems, which evaluate attributes of a target by interpreting the echoes from radio or sound waves, respectively. Active ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object. Passive ultrasonic sensors are basically microphones that detect ultrasonic noise that is present under certain conditions, convert it to an electrical signal, and report it to a computer.

8.6 Wireless Camera

Figure 8.14: Wireless Camera

Wireless cameras are cameras that transmit a video and audio signal to a wireless receiver through a radio band. Many wireless cameras require at least one cable or wire for power; "wireless" refers to the transmission of video/audio. However, some wireless cameras are battery-powered, making the cameras truly wireless from top to bottom.

Wireless cameras are proving very popular among modern consumers due to their low installation costs (there is no need to run expensive video extension cables) and flexible mounting options; wireless cameras can be mounted/installed in locations previously unavailable to standard wired cameras. In addition to the ease of use and convenience of access, wireless camera allows users to leverage broadband wireless internet to provide seamless video streaming over-internet.

8.7 Voltage Regulator

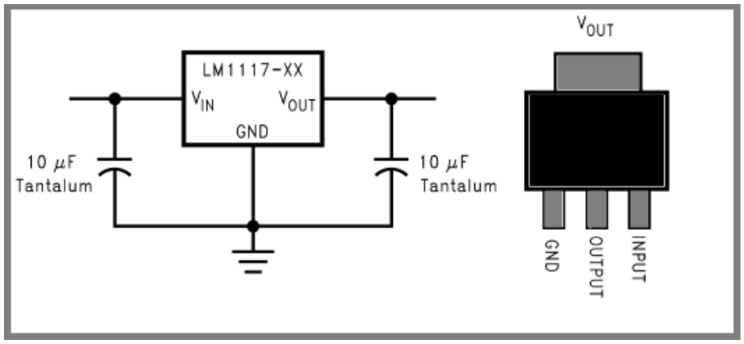
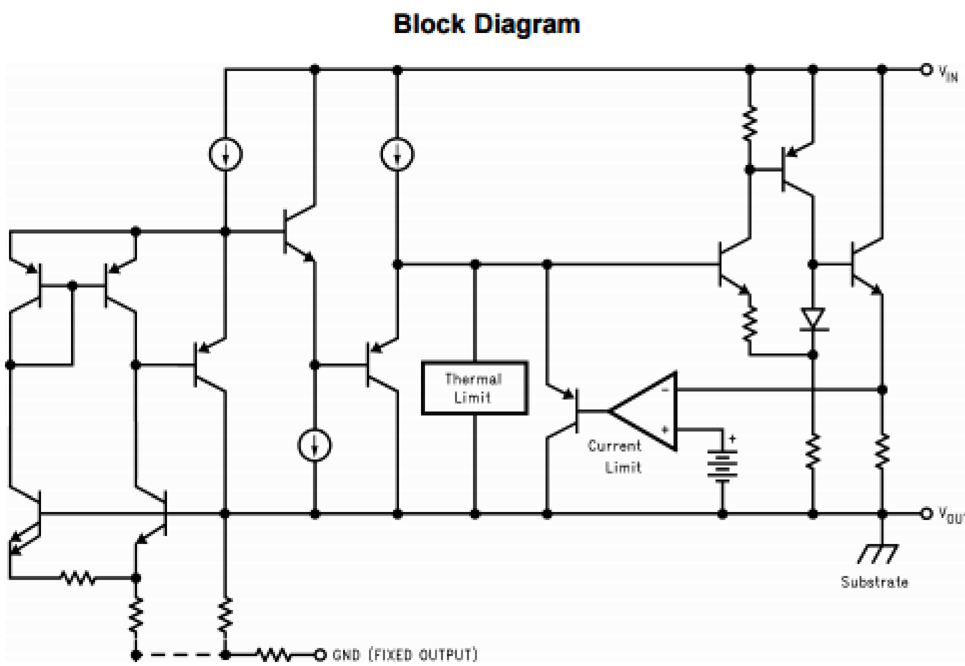


Figure 8.15: Voltage regulator connections



The LM1117-N is a series of low dropout voltage regulators with a dropout of 1.2V at 800mA of load current. It has the same pin-out as Texas Instruments' industry standard LM317. The LM1117-N is available in an adjustable version, which can set the output voltage from 1.25V to 13.8V with only two external resistors. In addition, it is also available in five fixed voltages, 1.8V, 2.5V, 2.85V, 3.3V, and 5V.

The LM1117-N offers current limiting and thermal shutdown. Its circuit includes a zener trimmed band gap reference to assure output voltage accuracy to within ±1%. The LM1117-N series is available in WSON, PFM, SOT-223, TO-220, and TO-263 DDPAK packages. A minimum of 10µF tantalum capacitor is required at the output to improve the transient response and stability.

8.7.1 Features

- Available in 1.8V, 2.5V, 2.85V, 3.3V, 5V, and Adjustable Versions
- Space Saving SOT-223 and WSON Packages
- Current Limiting and Thermal Protection
- Output Current 800mA
- Line Regulation 0.2% (Max)
- Load Regulation 0.4% (Max)
- Temperature Range
  – LM1117-N: 0°C to 125°C
  – LM1117I: −40°C to 125°C

8.7.2 Applications

- 2.85V Model for SCSI-2 Active Termination
- Post Regulator for Switching DC/DC Converter
- High Efficiency Linear Regulators
- Battery Charger
- Battery Powered Instrumentation

8.8 DC Motor

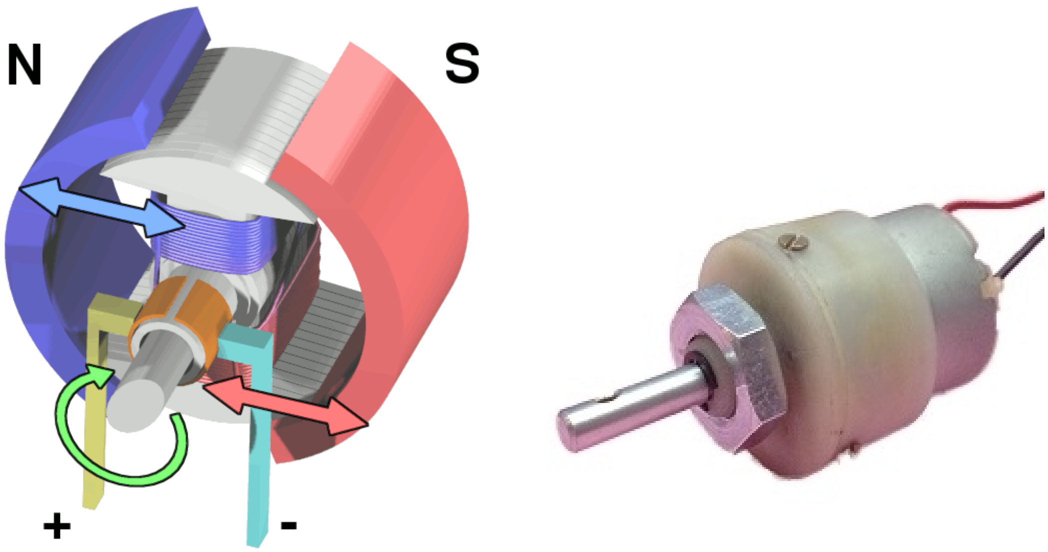


Figure 8.17: DC Motor

A DC motor is any of a class of electrical machines that converts direct current electrical power into mechanical power. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor. Most types produce rotary motion; a linear motor directly produces force and motion in a straight line.

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and hoists, or in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

8.9 Servo Motor

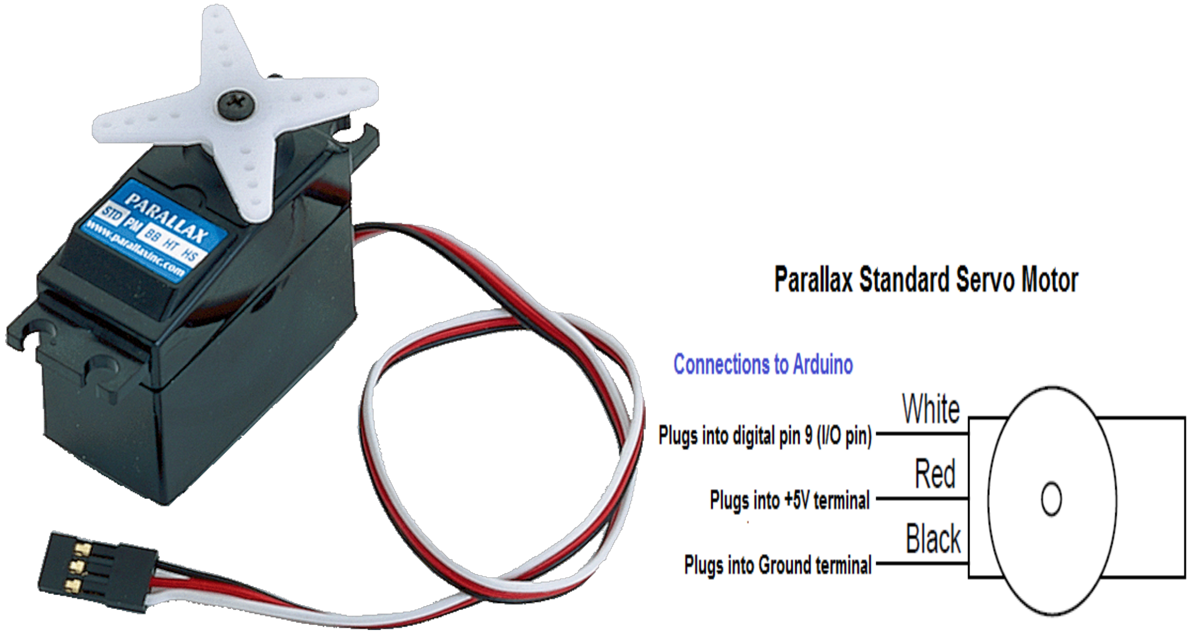


Figure 8.18: Servo Motor

A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. These are often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

As the name suggests, a servomotor is a servomechanism. More specifically, it is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is some signal, either analogue or digital, representing the position commanded for the output shaft.

The motor is paired with some type of encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in position. As the positions approach, the error signal reduces to zero and the motor stops.

8.10 Resistors

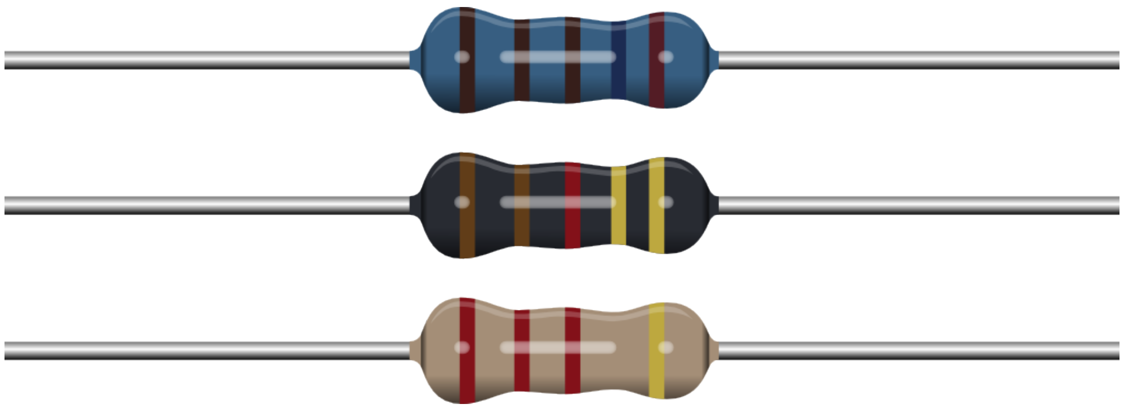


Figure 8.19: Resistors

Resistors are components that have a predetermined resistance. Resistance determines how much current will flow through a component. Resistors are used to control voltages and currents.

A very high resistance allows very little current to flow. Air has very high resistance. Current almost never flows through air. (Sparks and lightning are brief displays of current flow through air.

The light is created as the current burns parts of the air.) A low resistance allows a large amount of current to flow. Metals have very low resistance. That is why wires are made of metal.

They allow current to flow from one point to another point without any resistance. Wires are usually covered with rubber or plastic.

This keeps the wires from coming in contact with other wires and creating short circuits. High voltage power lines are covered with thick layers of plastic to make them safe, but they become very dangerous when the line breaks and the wire is exposed and is no longer separated from other things by insulation.

8.11 Capacitor

Figure 8.20: Capacitors

Capacitors usually have two legs.

One leg is the positive leg and the other is the negative leg.

The positive leg is the one that is longer.

The picture on the right is the symbol used for capacitors in circuit drawings (schematics).

When you put one in a circuit, you must make sure the positive leg and the negative leg go in the right place.

Capacitors do not always have a positive leg and a negative leg.

A capacitor is similar to a rechargeable battery in the way it works.

The difference is that a capacitor can only hold a small fraction of the energy that a battery can. (Except for really big capacitors like the ones found in old TVs. These can hold a lot of charge. Even if a TV has been disconnected from the wall for a long time, these capacitors can still make lots of sparks and hurt people.)

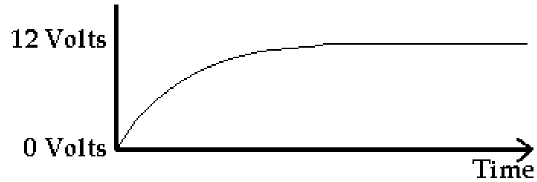
As with a rechargeable battery, it takes a while for the capacitor to charge.

So if we have a 12 volt supply and start charging the capacitor, it will start with 0 volts and go from 0 volts to 12 volts.

Below is a graph of the voltage in the capacitor while it is charging.

The same idea is true when the capacitor is discharging.

If the capacitor has been charged to 12 volts and then we connect both legs to ground, the capacitor will start discharging but it will take some time for the voltage to go to 0 volts.



8.12 Batteries

Figure 8.21: Batteries

A rechargeable battery, storage battery, secondary battery or accumulator is a type of electrical battery. It comprises one or more electrochemical cells, and is a type of energy accumulator used for electrochemical energy storage. It is also known as a secondary cell because its electrochemical reactions are electrically reversible. Rechargeable batteries come in many different shapes and sizes, ranging from button cells to megawatt systems connected to stabilize an electrical distribution network. Several different combinations of chemicals are commonly used, including lead–acid, nickel cadmium (NiCd), nickel metal hydride (NiMH), lithium ion (Li-ion), and lithium ion polymer (Li-ion polymer).

Rechargeable batteries have a lower total cost of use and environmental impact than disposable batteries. Some rechargeable battery types are available in the same sizes as common consumer disposable types. Rechargeable batteries have a higher initial cost but can be recharged inexpensively and reused many times.

The one which we are using in our project is a Lithium ion of 2000 mAH and 1000 mAH.

8.13 Electrical Connectors / Wires

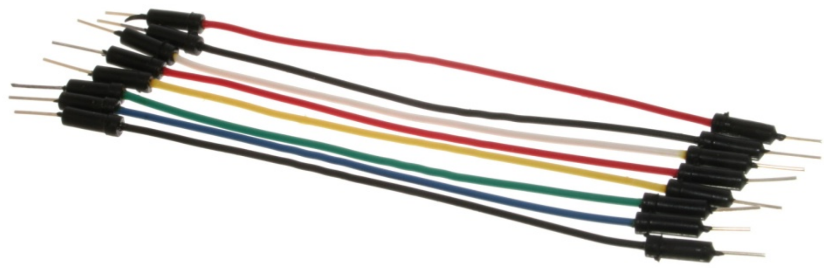


Figure 8.22: Connectors

An electrical connector or wire is an electro-mechanical device for joining electrical circuits as an interface using a mechanical assembly. Connectors consist of plugs (male-ended) and jacks (female-ended). The connection may be temporary, as for portable equipment, require a tool for assembly and removal, or serve as a permanent electrical joint between two wires or devices. An adapter can be used to effectively bring together dissimilar connectors.

There are hundreds of types of electrical connectors. Connectors may join two lengths of flexible copper wire or cable, or connect a wire or cable to an electrical terminal. In computing, an electrical connector can also be known as a physical interface.

Chapter 9

APPLICATIONS

9.1 Defense

9.1.1 Remote Spying

The robot can be driven to places for monitoring and spying. The robot is also well equipped for surveillance at place were humans cannot reach. It can also provide live feeds form the remote places to the user continuously.

9.1.2 Mine Detector

Land mines are the most hazardous weapons used during wars, they are intentionally planted to destroy the enemy forces in bulk. When the wars are settled, the mines which are left behind, pose the threat to inhabitants of that region. Since they are hidden and are activated even with minute pressure, it's very risky to search for it manually and can cost lives. This robot can be used with attached mine detector and controlled from a safe distance to find mine locations. Thus they can be diffused later safely.

9.2 Commercial and Industrial Applications

9.2.1 Quarry Mining

India – Digging up more coal has become a national priority for India as it tries to meet its electricity needs. But for the people who do it, like the miners, it's a dangerous and potentially lethal assignment. Our robot can be used to detect threat levels and intensity of harmful gases inside the quarry and thus the danger can be minimized and disasters can be prevented.

9.2.2 Radioactive Zones

Nuclear power plants have radioactive zones which have high levels of radiations, humans are forbidden to enter these regions, here the robot can be used to detect the intensity of these radiation levels and monitor the ambiance.

Chapter 10
FUTURE SCOPE

10.1 Range Improvement:

The range of the robot can be enhanced according to the various applications by using more advanced and effective communication methods like:

10.1.1 Zig-Bee

Figure 10.1: Zig-Bee

10.1.2 GSM

Figure 10.2: GSM Module

10.1.3 Internet Protocol (IP)

Figure 10.3: Internet Protocol

10.2 Access to Handicaps:

People suffering from physical disabilities or having paralyzed bodies cannot take advantage of skeleton detection. Also, many of them fail to communicate verbally. Here, another feature of Kinect sensor can prove to be boon for them i.e. Face Detection. With various facial expressions they can communicate with the robot to do their tasks.

Chapter 11

CONCLUSION

We are in the era of revolution, a revolution in technology. This revolution is bringing significant advancements in Human-Machine Interactions (HMI).From using keypad mobile sets we have upgraded to touch screen ones were human touch can operate devices effectively, which is possible using HMI. In this project we have designed and fabricated a robot which can function on vision - based gesture recognition using Kinect sensor. Human can control the robots arm by simply moving his own arm. The robot basically replicates human movements effectively. To make the robot more practically applicable we designed it to move on wheels. Moreover to make it immensely portable we made it wireless, which functions on Bluetooth so now it moves freely anywhere. The robot is also fitted with a wireless RF camera which provides live streaming of the scenario where it is moving, this makes remote access possible.

Extending it further, the robot has an ultrasonic sensor to avoid collisions with the obstacles. This robot is a revolution in itself no need of remote controls or trained personal this robot will do what human, controlling it, will do. The robot is very cheap in cost which is a boon in itself. This technology will certainly bring a revolution in robotics. With this technology it is not far when a humanoid robot will mimic human completely performing all their difficult task efficiently.

REFERENCES

Journal Paper,

1. Joshua Fabian et al.; “Integrating the Microsoft Kinect with Simulink: Real Time object tracking example”, IEEE Transactions on Mechatronics (Feb. 2014).
2. Antonio Frisoli et al.; “A new gaze-BCI-driven control of an upper limb of exoskeleton for rehabilitation in real world task”, IEEE Transactions on systems (Nov. 2012).
3. Daniel Kruse et al.; “A sensor based dual arm tele robotics system”, IEEE Transactions on Automation Science and Engineering (May 2014).
4. Dowling, A.V., Barzilay, O., Lombrozo, Y., Wolf, A.; “An adaptive home-use robotic rehabilitation system for the upper body.” IEEE Journal of Translational Engineering in Health and Medicine, March 2014.
5. Wenjun Zeng; University Missouri, “Microsoft Kinect Sensor and its effects”, published by IEEE Computer Society (2012).

PUBLICATIONS

With immense pleasure and gratitude, we would like to say that a paper in the ***"Institute of Electrical & Electronics Engineers"*** abbreviated as ***'IEEE'*** has been successfully presented and published.

The paper titled as ***"Gesture based wireless single – armed Robot in Cartesian 3D space using Kinect"*** was presented at ***"The fifth IEEE International Conference on Communication Systems and Network Technologies" (CSNT-2015)*** organized by ***'Machine Intelligence research Labs' (MIR Labs)***, Gwalior on 6th April 2015. The copy of the paper is attached hereafter.

The paper won ***'<u>Best Innovative Project</u>' and '<u>Excellent Presenter</u>'*** awards at the conference.

Gesture based wireless single-armed Robot in Cartesian 3D space using Kinect

Memon Md. Farhan Md. Fareed
Dept. Of Electronics & Telecommunication
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel, India
md.farhanmemon@gmail.com

Quraishi Imran Akram
Dept. Of Electronics & Telecommunication
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel, India
iq1604@gmail.com

Shaikh Bilal Ahmed Anees
Dept. Of Electronics & Telecommunication
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel, India
skbilal93@gmail.com

Awab Habib Fakih
Dept. Of Electronics & Telecommunication
Anjuman-I-Islam's Kalsekar Technical Campus
New Panvel, India
awaabfakih0@gmail.com

***Abstract*—Human Machine Interaction (HMI) has always played an important role in everybody's life motivating research in the area of intelligent service robots. Conventional methods such as remote controllers or wearables cannot cater the high demands in some scenarios. To overcome this situation, the challenge is to develop vision-based gesture recognition techniques. This paper describes our work of controlling an Arduino based wheeled, one armed robot, used as a prototype, controlled through various gestures of the arms and legs. For gesture recognition, we make use of skeletal tracing ability of Kinect – a product of Microsoft. Bluetooth is used to make the controls wireless. Since it is not line of sight operation, the robot also captures the environment video and transmits it over radio frequency in real-time and displayed on the screen. On the user end, according to the received video, the operator guides the robot and uses the arm to pick and place objects with the help of predetermined gestures.**

***Index Terms*—Arm control, Arduino, Gesture recognition, Kinect Sensor, Motion control, Robotic arm, Skeletal Tracking**

I. INTRODUCTION

This paper describes a prototype single-armed robot which is controlled purely using vision based gesture recognition. Fig. 1 describes the concept of the project in brief. The body movements of the user are tracked by the Kinect sensor, a product of the Microsoft. The tracked joints are scaled in Cartesian coordinate 3 dimensional space. The Kinect SDK determines this data and compares it with predefined gestures. Appropriate data are then transmitted to the robot.

Robot processes this received data and performs tasks as instructed [1]. The robot moves in four specified directions and the arm can be used to grab objects and place them as required. Our robot is self-equipped to detect obstacles and avoid collisions. With the moto to make this robot go to remote places for surveillance we have plugged in a wireless camera on the robot. The camera continuously captures and stream live scenario from the site to the user in radio frequency.

II. MOTIVATION

In present times, the technology is advancing with minute's hand. Existing methods of controlling robots with keyboards and mouse have aged. HMI which emphasized on human interaction with machines went through various advancements. Devices like accelerometers [1] and gyroscopes are used to manipulate the body gestures. However, there are limitations and hardware complexities, these devices needs to be held or worn on body (wearables). Also these wearables need to be properly calibrated to make exact gesture recognitions. Moreover most of these existing devices are wired.

Therefore we focused ourselves towards creating a pure gesture controlled robot which can actually replicate human real-time body movements and perform various tasks with ease. We also made our prototype robot wireless improving upon its applications and practicality.

III. KINECT SENSOR

The Kinect sensor revolutionized the HMI. Kinect's splendid technological advancements made the machine interaction quite easy. It can track and follow the person standing in front of it. The sensor actually can track the person's skeleton, significantly exploring the major joints and tracing them efficiently. Kinect can also judge how much closer or farther the person is with the help of its depth sensors.

The sensor is engraved with advance sensing hardware. It consist of a normal color camera (RGB camera), the depth sensor which comprises of an infrared projector (IR projector) which actually is an infrared laser that emits IR beams and the infrared camera (IR camera) which is an infrared receiver. The sensor also have an array of microphone for voice recognition [2]. The sensor can not only see you but can also hear you. The sensor provides appropriate data of the joints traced which can be processed by the computer. The sensor's 3D tracking ability is used in this prototype design.

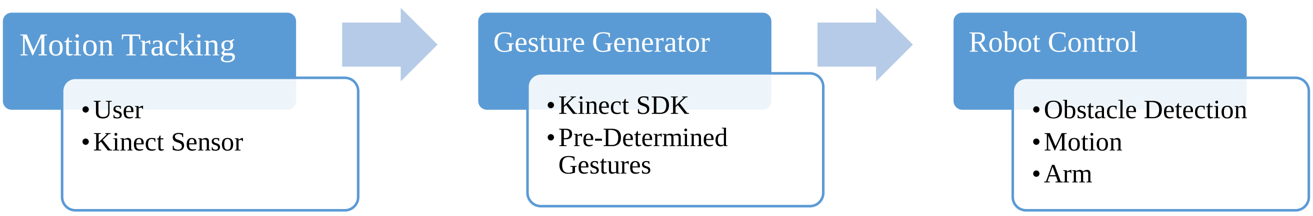


Figure 1. Concept of the project

A. Skeleton Tracking:

The significant innovation in Kinect is its skeleton tracking ability. The sensor can elegantly track the skeleton of the person standing in front of it. Locate and trace person's major joints and provide helpful content of its locations which can be processed by computer. This is done by the sensor in following manner:

1. Depth sensor:

The depth sensor is a combination of IR projector (laser) and IR camera, which is a CMOS sensor The IR projector projects a beam of infrared light which is allowed to fall on a diffraction grating which scatters the light beams into closely spaced tiny spots of light. This spots of light falls on various object and gets reflected back in its path. The light is capture by the IR camera which is a monochrome complementary metal-oxide semiconductor sensor (CMOS).

The original picture can be reconstructed with these spots. Depth values are coded in gray values the closer the object the darker will be its depth image pixels. The objects distance from the sensor is flawlessly coded into gray values depending on the time taken by the IR lights to return back to the sensor [2]. However, the objects which are too far from the sensor are coded black pixels as IR light is dispersed and no depth values are available. The similar is the case if objects are too near, these objects fall in sensors blind spot and cannot be detected hence coded as black pixel.

2. Joint Tracking:

Human body has different depths at different parts. This varying depths of human body produces different pixels in depth image. The depth image of the body is divided into various segments depending upon their pixel. Evaluation of this per pixel sub division give us the trace of joints. The joints are classified it to various categories like (head, hand, neck, shoulder etc.). The sensor in all track 20 major joints of human body. These joints are effectively located by the sensor and tracked continuously. The joints are interconnected to obtain the person's skeleton [3]. This result is optimized by tracking the joints at a rate of 5ms per frame. The sensor also provides the 3-D co-ordinates of the tracked joints.

IV. Overview

The prototype which was created consists of various components and sections. Fig. 2 describes the parts of the project are in the form of block diagram.

A. Sensor:

The body movements of the user are captured using the Kinect sensor. The information is acquired in the form of positions of the various joints as explained earlier using the 3 dimensional Cartesian co-ordinate space.

B. Computer:

The captured image data is sent serially through the USB port to a computer machine where the captured skeleton is displayed and the co-ordinates of the joints are processed. Processing of the gestures requires a program compatible with the Software Development Kit (SDK) of the Kinect.

C. Wireless Serial Transmitter:

After the recognition of the gestures is completed, the corresponding data is transmitted from the computer to the robot through any wireless transmitter. In this project we have used Bluetooth for this operation.

D. Wireless Serial Receiver:

The transmitted data is received in the buffer of the Bluetooth module connected to the robot and given to microcontroller for further processing.

E. Processing in Microcontroller:

The received information of the gesture via Bluetooth is processed in the microcontroller for various tasks to perform. We have used Arduino board for performing this operation which generates various PWM signals [4].

V. Logical Flow of the Program

As we mentioned earlier, the program is needed to operate the Kinect sensor and the robot. We coded the Kinect program in C# language and used Arduino for the motion and arm control in robot. The flow charts of the same are as described below:

A. Input Acquisition and its Transmission:

Fig. 3 shows the program flow for the image data acquisition. For Kinect to capture the motion of the body, we first need to initialize the sensor by creating a window environment for reproducing the captured image data on the display. Check for Kinect sensor if connected or not. If the Kinect sensor is not detected, it gives the error. Similarly, it checks for the Bluetooth and tries to connect to the robot and if robot not found, the program execution stops displaying the corresponding error.

Once the initialization code is executed, the sensor captures the images at the rate of 30 frames per second [2] and is transferred as it is through the serial bus to the program which displays the skeleton. The positions of the joints in the Cartesian coordinates space resulting in the gesture and comparing those gestures with the ones which are predefined in the program is processed simultaneously. If the gestures are matched, the corresponding information is transmitted via Bluetooth to the robot, else the tracking continues.

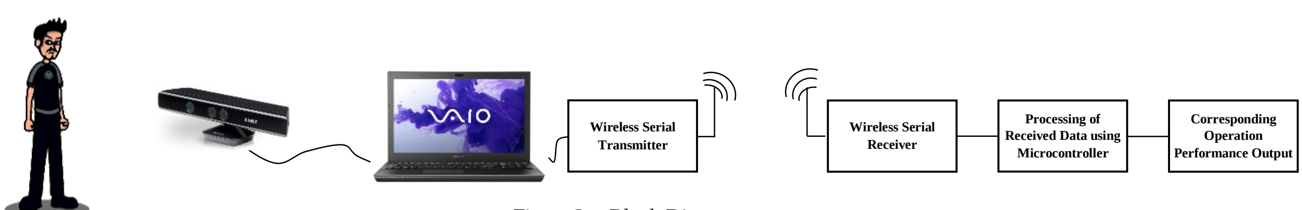


Figure 2. Block Diagram

B. Processing at the Robot End:

The programming of robot flows the similar way as that of Kinect sensor. Fig. 4 shows the flow chart of the same. The robot when initialized, the Bluetooth module scans for the devices for connection from the remote computer machine. Once the connection between the two is established, the gesture information is received to the microcontroller. The microcontroller processes the received data and scans for the relevant operation to be performed. If the received information is for motion of the robot, the corresponding PWM signals are generated. If they belong to the arm control, the received angles are first quantized and then the axel rotation of the motors takes place and the arm replicates the arm of the user.

However, if the command is for hand, it opens and closes its jaws accordingly. There's a provision for obstacle detection too using the ultrasonic sensor. It protects the robot from getting damaged by continuously monitoring its environment and maintaining safe distance.

VI. Control Operations of the Robot

Robot can be controlled by simple gestures using bare hands. The hand movements are processed by the computer in real-time and appropriate gesture is identified from the predefined feed. In this project we have used the left hand to control the motion and the right hand controls the arm of the robot. The gestures are as defined below.

A. Motion Control:

We incorporated following gestures to control the motion of the robot:

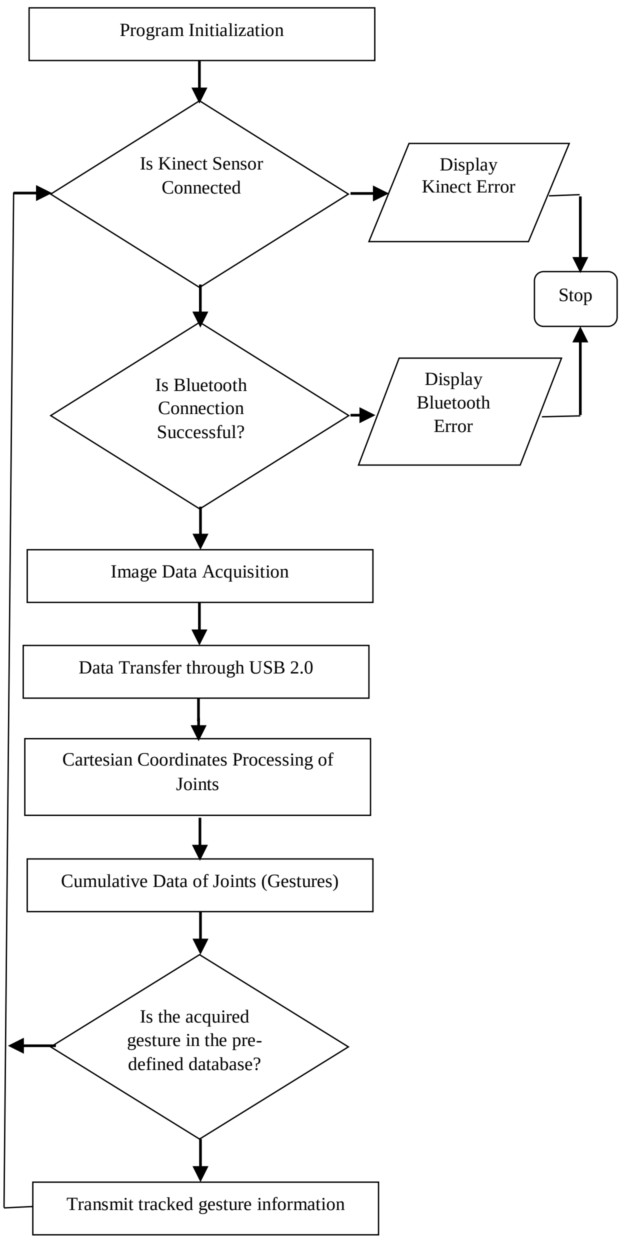


Figure 3. Flow chart for image data capturing

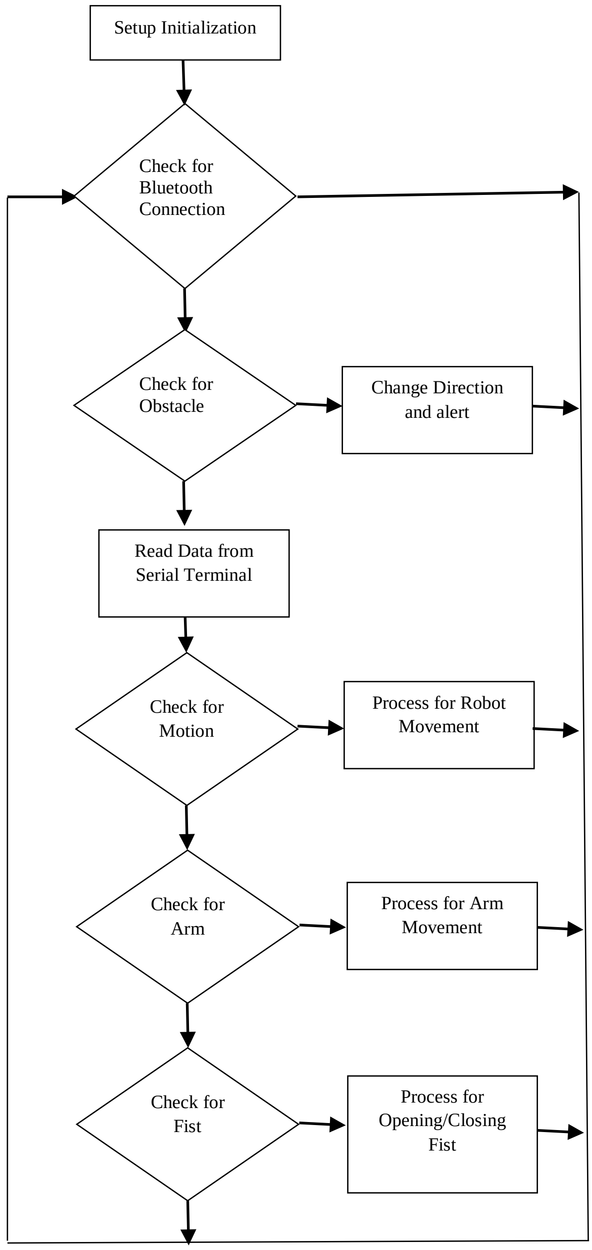


Figure 4. Flow chart for Robot

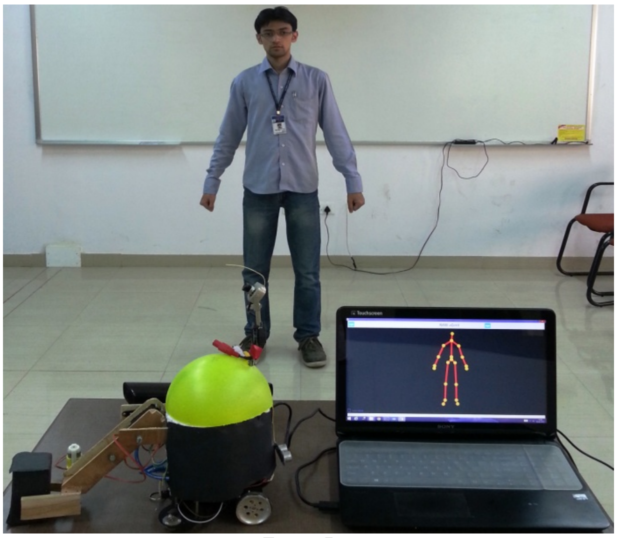


Figure 5

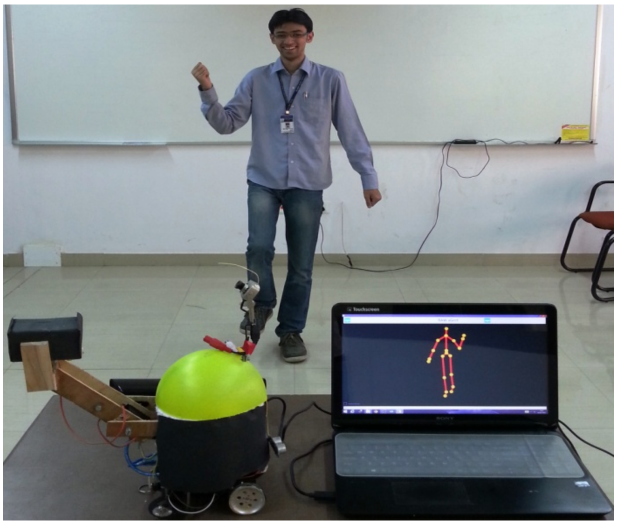


Figure 6


Figure 7


Figure 8


Figure 9


Figure 10

Fig. 5 shows the initial position of the robot, this gesture also stops the motion of the robot. Fig. 6-8 depicts how the arm and gripper is controlled, right hand has been used for this purpose. In Fig. 9 and Fig. 10, the left hand gestures makes the robot move in forward, reverse, left and right directions along with the arm control.

B. Arm Control:

The basis of movement of the arm imitating that of the user is purely dependent on calculating the angles and positions between various joints of the hand. The angle between two bones is analogous to finding angles between two vectors using the following equation.

Fig. 11 describes the angle between the two vectors in the in 3D space can be calculated, as in (1)

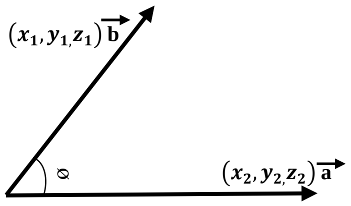


Figure 11. Two vectors with 3D coordinates

$$\theta = \cos^{-1}\left(\frac{x_1.x_2 + y_1.y_2 + z_1.z_2}{\sqrt{x_1^2+y_1^2+z_1^2}\,.\sqrt{x_2^2+y_2^2+z_2^2}}\right) \qquad (1)$$

Now let's see the joints which we considered for determining the angles between the arm bones with the help of the tracked skeleton [4].

1. Base:

Fig. 12 shows the angle between the shoulder and the elbow bone in horizontal plane, is named as the base angle denoted by α as shown. This will help in moving the arm in right or left direction.

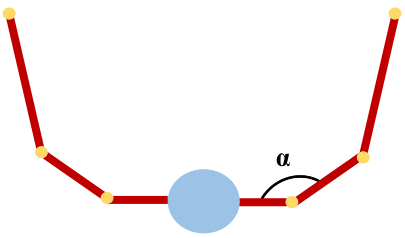


Figure 12. Base Angle

2. Shoulder:

Fig. 13 shows the second angle denoted by β, is the one between the body center and the shoulder bone as shown. It will move the arm in vertical direction.

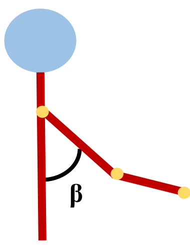


Figure 13. Shoulder Angle

3. Elbow:

Fig. 14 shows the angle formed by the elbow and shoulder bone, denoted by γ. This will lift the jaw up and down helping in picking up objects and placing them where you want.

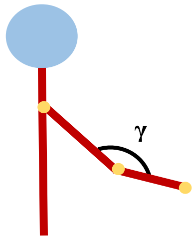


Figure 14. Elbow Angle

Fig. 15 shows the complete mechanical arm after combining all the components. There is a high probability that the human arm may suddenly change the speed and direction or may go out of bounds causing the jerks in the robotic arm. This situation will not only cause damage to the robot but also there will be an unsafe feeling to the user [4]. To avoid such scenarios, we have specified movement ranges by quantizing the angles received through the Kinect (TABLE I). It will assure smooth functioning and operation of the robot increasing its shelf life.

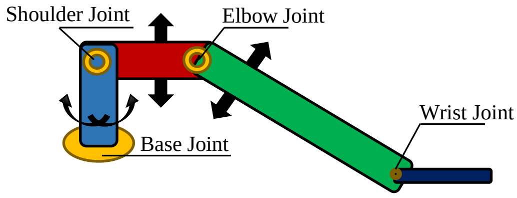


Figure 15. Robotic Arm

TABLE I. EXTREME RANGES OF THE ANGLES FOR ROBOTIC ARM MOVEMENT

Joint Name	Minimum Value of Rotation	Maximum Value of Rotation
Base Joint	-90°	+90°
Shoulder Joint	0°	+45°
Elbow Joint	0°	+90°

VII. CONCLUSION

In this paper we have designed a single-armed robot with four way motion. The robot is made self-equipped with collision detection. Furthermore we have even made it more practical and portable by implementing it wirelessly using Bluetooth. However the use of 3-Dimensional Gesture recognition used to operate the robot makes extremely easy to use. Any person can easily operate this robot on their own. We have put in efforts to make this technology more application oriented. With future work in this field could bring more fantastic results in the field of improving HMI and making Humanoid robots. In future the range of operation this robot can be enhanced by using different antenna modules. This would make the robot applicable in many industrial sectors. It can be used efficiently in the radioactive zones which are human forbidden region.

With appropriate sensors it can be used in mining to detect the level of hazard for miners as a precautionary measure. Operational coverage can be increased by modules like ZigBee, GSM module, Internet Protocol (IP).

REFERENCES

[1] Neto, Pedro, Pires, JN and Moreira, “Accelerometer-based control of an Industrial Robotic Arm Robot and Human Interactive Communication”, RO-MAN 2009.

[2] Wenjung Zeng, University of Missouri, “Microsoft Kinect Sensor and its Effects”, IEEE 2012.

[3] RizqaAfthoni, Achmad Rizal, Erwin Susanto, “Proportional Derivative Control Based Robot Arm System Using Microsoft Kinect”, International Conference of Robotics, Biometics, Intelligent Computation System, 2013.

[4] Rajesh Kannan Megalingam, Nihil Saboo, Nitin Ajithkumar, SreeramUnny, Deepansh Menon, “Kinect Based Gesture Controlled Robotic Arm: A Research work at HuT Labs”, IEEE, 2013.