Sudan University of Sciences and Technology
Collage of Engineering
School of Electrical &Nuclear Engineering

Control of DC Motor Using Android

التحكم في محرك التيار المستمر باستخدام الأندرويد

A Project Submitted In Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) In Electrical Engineering

Prepared BY:

1. **Mohamed Abdulgafour Osman Kailani**
2. **Abdelrhman Suraj aldin Abdelrhman Osman**
3. **Aymen Mohamed El-mukhtar El-tayeb El-siddig**
4. **Mohamed Salah Mohamed Kheir Gasm Alseed**

Supervised BY:
Ust. Galal Abdelrahman Mohammed Hammad

October 2015

الآية

قال الله تعالى في محكم تنزيله:

"يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ

بِمَا تَعْمَلُونَ خَبِيرٌ"

المجادلة (11)

الإهداء

الى أمي وأبي

الى كل من علمني ، و أخذ بيدي ، وأنار لي طريق العلم والمعرفة .

الى كل من شجعني في رحلتي الى التميز و النجاح .

الى كل من ساندني ، ووقف بجانبي .

الى كل من قال لي : لا ، فكان سببا في تحفيزي .

الى كل من كان النجاح طريقه ، والتفوق هدفه ، والتميز سبيله .

إليكم جميعا أهدي هذا المشروع .

الشكر و العرفان

الى من كانوا لنا خير معين و كانوا شموعا تحترق لتضيء لنا الطريق ، و الى كل من لم يبخلوا علينا بوقتهم و جهدهم و نصائحهم في إكمال هذا العمل الذي ترونه بين أيديكم .

الى الأعزاء على قلوبنا الاستاذ جلال عبدالرحمن محمد و المهندس طارق حسين عبدالله .

الى كل من ساهم في إكمال هذا العمل بجهد أو نصح أو مشورة .

Abstract

Arduino device overcame the slow control problem in Direct Current (DC) motors, where the control process required a lot of time and effort and also it did not give accurate results. Arduino has made some advantages in control process such as high speed response and decreasing the number of laborers as we can connect group of DC motors with it and control all of them through only one Arduino device. Bluetooth has been used as a mediator between Android mobile phone and Arduino device, so the control became available easily and wireless for good distances. Android system provided us a wide range in angular control for DC motors, where the control in angles became available between (0-180) degrees.

المستخلص

جهاز الأردوينو عالج مشكلة التحكم في محركات التيار المستمر حيث كان الأمر يتطلب وقتا و جهدا كثيرين و لا يعطي نتائج دقيقة. يتميز جهاز الأردوينو بسرعة استجابة عالية كما يؤدي لتقليل الأيدي العاملة حيث يمكننا ربط مجموعة من المحركات التيار المستمر به والتحكم بها من خلال جهاز أردوينو واحد فقط .إستخدام البلوتوث كوسيط لنقل الأوامر بين الهاتف الجوال (الذي يعمل بنظام الأندرويد) و جهاز الأردوينو جعل عملية التحكم متاحة عن بعد لمسافات مقبولة بسهولة ويسر. نظام الأندرويد وفر مدى أوسع في التحكم الزاوي لمحرك التيار المستمر،حيث أصبح التحكم في الزوايا في المدى ما بين (0-180) درجة .

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

E	Supply voltage, v
Eb	Back EMF, v
P	Number of poles
φ	Flux per pole, wb
Z	Total number of conductors
N	Speed of DC motor, rpm
A	Number of parallel paths
ω	Angular speed, rad\sec
Ia	Armature current, A
Ra	Armature resistance, Ω
Pm	Mechanical power, w
Tg	Electromagnetic torque, N.m

LIST OF ABBREVIATIONS

DC	Direct Current
MIMO	Multiple Input Multiple Output
PLC	Programmable Logic Controllers
MCU	Micro Controller Unit
RAM	Random Access Memory
OTP	One Time Programmable
EPROM	Erasable Programmable Read Only Memory
EEPROM	Electrical Erasable Programmable Read Only Memory
LED	Light Electronic Display
LCD	Liquid Crystal Display
I/O	Input Output
ISR	Interrupt Service Routine
RAM	Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter
PWM	Pulse Width Modulation
MIT	Massachusetts Institute of Technology
AVRC	Audio/Video Remote Control
PDA	Personal Digital Assistants
AC	Alternative Current
EMF	Electro Motive Force
BDC	Brushes Direct Current
BLDC	Brushes Less Direct Current
OS	Operation Systems
SDK	Software Development Kit
API	Application Programming Interface
QEMU	Quick Emulator
IDE	Integrated Development Environment
ADT	Android Development Tool

NDK	Native Development Kit
HTML	Hyper Text Markup Language
APK	Android Application Package
RSS	Rich Site Summary
SMS	Short Message Service
IOS	IPhone Originally System
GPS	Global Positioning System
LTE	Long Term Evolution
UHF	Ultra High Frequency
ISM	Institute for Supply Management
PAN	Personal Area Network
EDR	Enhanced Data Rate
USB	Universal Serial Bus
AREF	Analog Reference
IC	Integrated Circuit
SRAM	Static Random Access Memory

CHAPTER ONE

INTRODUCTION

1.1 General Concepts

Electric motors impact almost every aspect of modern living. Refrigerators, vacuum cleaners, air conditioners, fans, computer hard drives, automatic car windows, and multitudes of other appliances and devices all use electric motors to convert electrical energy into useful mechanical energy. In addition to running the commonplace appliances that we use every day, electric motors are also responsible for a very large portion of industrial processes.

Electric motors are used at some point in the manufacturing process of nearly every conceivable product that is produced in modern factories. Because of the nearly unlimited number of applications for electric motors, it is not hard to imagine that there are over 700 million motors of various sizes in operation across the world. This enormous number of motors and motor drives has a significant impact on the world because of the amount of power they consume.[1]

The systems that controlled electric motors in the past suffered from very poor performance and were very inefficient and expensive. In recent decades, the demand for greater performance and precision in electric motors, combined with the development of better solid-state electronics and cheap microprocessors and microcontrollers made DC motors more accessible for control in innumerable applications. Also made control operation very ease, faster and more accurate.

1.2 Problem Statement

In the past the DC motors control process was very difficult and required a lot of equipment. This process of control were consumes time and requires the presence of the operator near the machine. Angular control in these engines was confined to certain angles. In spite of all above defects a process control was slow response and inaccurate.

1.3 Objectives

The main objective of this project are:

- Control 0 -180 degree of motor position.
- Auto control function and can adjust timing to control.
- Possibility to control motor remotely.

1.4 Methodology

Android smartphone enters command or order to Arduino this signal will be received by Bluetooth module then will be send to Arduino via Bluetooth which stores information about various jobs or tasks and programmed to activate motor.

1.5 Project Layout

This project consists of five chapters. Chapter one is an introduction of the project which includes general concepts , problem statement, objectives and methodology of the project. Chapter two is a general overview about the project. It includes introduction, control system, microcontroller, Arduino, Bluetooth, DC motor, DC servomotor and android.

Chapter three is talking about components of the circuit. It includes introduction, Bluetooth module, Arduino board and DC servomotor. Chapter four is the simulation and practical results of the project. It contains introduction, Proteus, steps of simulation, Operation of the circuit, practical work and results. Finally, chapter five includes conclusion and recommendations.

CHAPTER TWO

OVERVIEW

2.1 Introduction

Control system engineering is the branch of engineering which deals with the principles of control theory to design a system which gives desired behavior in a controlled manner. Control system engineers analyze, design, and optimize complex systems which consist of highly integrated coordination of mechanical, electrical, chemical, metallurgical, electronic or pneumatic elements. Thus control engineering deals with diverse range of dynamic systems which include human and technological interfacing.

Control system engineering focuses on analysis and design of systems to improve the speed of response, accuracy and stability of system. The two methods of control system include classical methods and modern methods. The mathematical model of system is set up as first step followed by analysis, designing and testing. Necessary conditions for the stability are checked and finally optimization follows.[2]

2.2 Control System

A control system is a system of devices or set of devices, that manages, commands, directs or regulates the behavior of other device(s) or system(s) to achieve desire results. In other words the definition of control system can be rewritten as A control system is a system, which controls other system.

As the human civilization is being modernized day by day the demand of automation is increasing accordingly. Automation highly requires control of devices. In recent years, control systems plays main role in the development and advancement of modern technology and civilization. Practically every aspects of our day-to-day life is affected less or more by some control system. A bathroom toilet tank, a refrigerator, an air conditioner, a geezer, an automatic iron, an automobile all are control system. These systems are also used in industrial process for more output. We find control system in quality control of products, weapons system, transportation systems, power system, space technology, robotics and many more. The principles of control theory is applicable to engineering and non-engineering field both.[2]

2.2.1 Feature of Control System

The main feature of control system is, there should be a clear mathematical relation between input and output of the system. When the relation between input and output of the system can be represented by a linear proportionality, the system is called linear control system. Again when the relation between input and output cannot be represented by single linear proportionality, rather the input and output are related by some non-linear relation, the system is referred as non-linear control system.[2]

2.2.2 Requirement of good control system

There are several requirements must be in control system to say that our control system is good, such:

- ❖ **Accuracy:** Accuracy is the measurement tolerance of the instrument and defines the limits of the errors made when the instrument is used in normal operating conditions. Accuracy can be improved by using feedback elements. To increase accuracy of any control system error detector should be present in control system.
- ❖ **Sensitivity:** The parameters of control system are always changing with change in surrounding conditions, internal disturbance or any other parameters. This change can be expressed in terms of sensitivity.

Any control system should be insensitive to such parameters but sensitive to input signals only.

- ❖ **Noise:** An undesired input signal is known as noise. A good control system should be able to reduce the noise effect for better performance.
- ❖ **Stability:** It is an important characteristic of control system. For the bounded input signal, the output must be bounded and if input is zero then output must be zero then such a control system is said to be stable system.
- ❖ **Speed:** It is the time taken by control system to achieve its stable output. A good control system possesses high speed. The transient

period for such system is very small.

- ❖ **Oscillation:** A small numbers of oscillation or constant oscillation of output tend to system to be stable.

2.2.3 Types of control systems

- ✓ **Open loop control system**

A control system in which the control action is totally independent of output of the system then it is called open loop control system. Manual control system is also an open loop control system. Figure 2.1 below shows the block diagram of open loop control system in which process output is totally independent of controller action.[2]

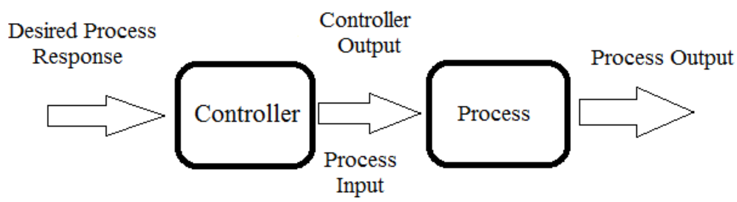


Figure 2.1: Block diagram of open loop control system

There are several advantages of open loop control system such as:

- Simple in construction and design.
- Economical.
- Easy to maintain.
- Convenient to use as output is difficult to measure.

Also there are many limitations of open loop control system such as:

- They are inaccurate.
- They are unreliable.
- Any change in output cannot be corrected automatically.
- ✓ **Closed loop control system**

Control system in which the output has an effect on the input quantity in such a manner that the input quantity will adjust itself based on the output generated is called closed loop control system. Open loop control system can be converted in to closed loop control system by providing a feedback. This feedback automatically makes the suitable changes in the output due to external disturbance. In this way closed loop control system is called automatic control system. Figure 2.2 below shows the block diagram of closed loop control system in which feedback is taken from output and fed in to input.[2]

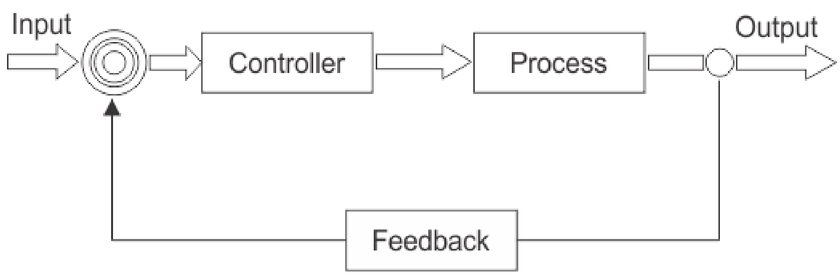


Figure 2.2: Block diagram of closed loop control system

A feedback is a common and powerful tool when designing a control system. Feedback loop is the tool which take the system output into consideration and enables the system to adjust its performance to meet a desired result of system.

In any control system, output is affected due to change in environmental

condition or any kind of disturbance. So one signal is taken from output and is fed back to the input. This signal is compared with reference input and then error signal is generated. This error signal is applied to controller and output is corrected. Such a system is called feedback system. Figure 2.3 below shows the block diagram of feedback system.

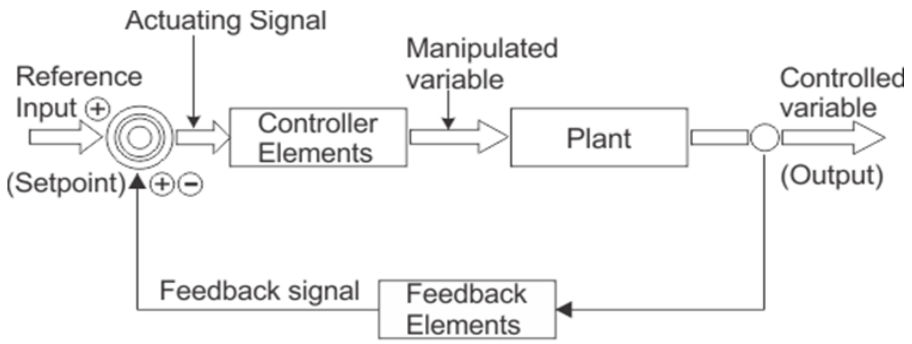


Figure 2.3: Block diagram of feedback system

When feedback signal is positive then system called positive feedback system. For positive feedback system, the error signal is the addition of reference input signal and feedback signal. When feedback signal is negative then system is called negative feedback system. For negative feedback system, the error signal is given by difference of reference input signal and feedback signal.

Adding feedback loop to the system involves the following affections such as:

- ✓ Error between system input and system output is reduced.
- ✓ System gain is reduced by a factor 1/(1±GH).

- ✓ Improvement in sensitivity.
- ✓ Stability may be affected.
- ✓ Improve the speed of response

Refer Figure 2.4 below, which represents feedback system where

- R = Input signal
- E = Error signal
- G = forward path gain
- H = Feedback
- C = Output signal
- B = Feedback signal

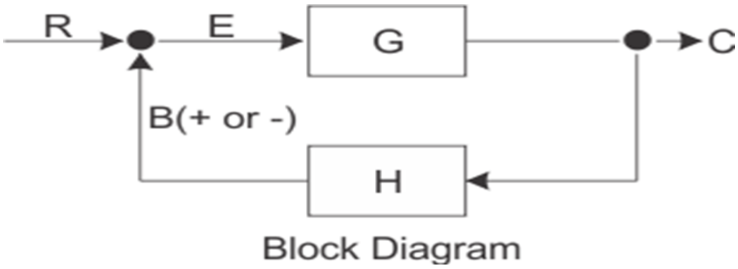


Figure 2.4: Feedback loop block diagram

Closed loop control systems have various advantages such as:

- ❖ Closed loop control systems are more accurate even in the presence of non-linearity.
- ❖ Highly accurate as any error arising is corrected due to presence of feedback signal.
- ❖ Bandwidth range is large.
- ❖ Facilitates automation.
- ❖ The sensitivity of system may be made small to make system more stable.
- ❖ This system is less affected by noise.

Also there are many disadvantages of closed loop control systems such as:

- ❖ They are costlier.
- ❖ They are complicated to design.
- ❖ Required more maintenance.
- ❖ Feedback leads to oscillatory response.
- ❖ Overall gain is reduced due to presence of feedback.

Table 2.1: Comparison between closed loop and open loop control system

No	Open Loop System	Closed Loop System
1	The feedback element is absent	The feedback element is always present
2	An error detector is not present	An error detector is always present
3	It is stable one	It may become unstable
4	Easy to construct	Complicated construction
5	It is an economical	It is costly
6	Having small bandwidth	Having large bandwidth
7	It is inaccurate.	It is accurate
8	Less maintenance	More maintenance

2.3 Microcontroller

A microcontroller (sometimes abbreviated µC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of Ferroelectric Random Access Memory (RAM), NOR flash or One Time Programmable Read only Memory (OTP ROM) is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.[3]

2.3.1 Embedded design

A microcontroller can be considered a self-contained system with a processor, memory and peripherals and can be used as an embedded system. The majority of microcontrollers in use today are embedded in other machinery, such as automobiles, telephones, appliances, and peripherals for computer systems.

While some embedded systems are very sophisticated, many have minimal requirements for memory and program length, with no operating system, and low software complexity. Typical input and output devices include switches, relays, solenoids, Light Electronic Display (LEDs), small or custom Liquid

Crystal Display (LCD) displays, radio frequency devices, and sensors for data such as temperature, humidity, light level etc. Embedded systems usually have no keyboard, screen, disks, printers, or other recognizable Input Output (I/O) devices of a personal computer, and may lack human interaction devices of any kind.[4]

2.3.2 Interrupt

Micro controllers must provide real time response to events in The

embedded system they are controlling. When certain events occur, an interrupt system can signal the processor to suspend processing the current instruction sequence and to begin an Interrupt Service Routine (ISR), or "interrupt handler". The ISR will perform any processing required based on the source of the interrupt, before returning to the original instruction sequence. Possible interrupt sources are device dependent, and often include events such as an internal timer overflow, completing an analog to digital conversion, a logic level change on an input such as from a button being pressed, and data received on a communication link. Where power consumption is important as in battery operated devices, interrupts may also wake a microcontroller from a low power sleep state where the processor is halted until required to do something by a peripheral event.

2.3.3 Programs

Typically microcontroller programs must fit in the available on-chip program memory, since it would be costly to provide a system with external, expandable, memory. Compilers and assemblers are used to convert high-level language and assembler language codes into a compact machine code for storage in the microcontroller's memory. Depending on the device, the program memory may be permanent, read-only memory that can only be programmed at the factory, or program memory that may be field-alterable flash or erasable read-only memory.

Manufacturers have often produced special versions of their microcontrollers in order to help the hardware and software development of the target system. Originally these included Electrical Programmable Read Only Memory

(EPROM) versions that have a "window" on the top of the device through which program memory can be erased by ultraviolet light, ready for reprogramming after a programming ("burn") and test cycle. Since 1998, EPROM versions are rare and have been replaced by Electrical Erasable Programmable Read Only Memory (EEPROM) and flash, which are easier to use (can be erased electronically) and cheaper to manufacture.

Other versions may be available where the ROM is accessed as an external device rather than as internal memory, however these are becoming increasingly rare due to the widespread availability of cheap microcontroller programmers.

The use of field-programmable devices on a microcontroller may allow field update of the firmware or permit late factory revisions to products that have been assembled but not yet shipped. Programmable memory also reduces the lead time required for deployment of a new product.

Where hundreds of thousands of identical devices are required, using parts programmed at the time of manufacture can be an economical option. These "mask programmed" parts have the program laid down in the same way as the logic of the chip, at the same time.

A customizable microcontroller incorporates a block of digital logic that can be personalized in order to provide additional processing capability, peripherals and interfaces that are adapted to the requirements of the application.[4]

2.3.4 Higher integration

A micro-controller is a single integrated circuit, commonly with the

Following features:

- ❖ Central processing unit - ranging from small and simple 4-bit processors to complex 32- or 64-bit processors.
- ❖ Volatile memory (RAM) for data storage.
- ❖ ROM, EPROM, EEPROM or flash memory for program and operating parameter storage.
- ❖ Discrete input and output bits, allowing control or detection of the logic state of an individual package pin.
- ❖ Serial input/output such as serial ports Universal Asynchronous Receiver/Transmitter (UARTs).
- ❖ Serial communications interfaces , serial peripheral interface and controller Area Network for system interconnect.
- ❖ Peripherals such as timers, event counters, Pulse Width Modulation (PWM) generators, and watchdog.
- ❖ Clock generator - often an oscillator for a quartz timing crystal, resonator or RC circuit.
- ❖ Many include analog-to-digital converters, some include digital-to-analog converters.

This integration drastically reduces the number of chips and the amount of wiring and circuit board space that would be needed to produce equivalent systems using separate chips. Furthermore, on low pin count devices in particular, each pin may interface to several internal peripherals, with the pin function selected by software. This allows a part to be used in a wider variety of applications than if pins had dedicated functions.

Microcontroller architectures vary widely. Some designs include general-purpose microprocessor cores, with one or more ROM, RAM, or I/O functions integrated onto the package. Other designs are purpose built for control applications. A micro-controller instruction set usually has many instructions intended for bit-wise operations to make control programs more compact.

There are many microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.[4]

2.4 Arduino

Arduino is a tool for making computers that can sense and control more of the physical world than our desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on our computer .

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

The first Arduino was introduced in 2005. The project leaders sought to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Adafruit Industries estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

The original Arduino hardware is manufactured by the Italian company Smart Projects .Some Arduino-branded boards have been designed by the American company SparkFun Electronics. Sixteen versions of the Arduino hardware have been commercially produced to date.[5]

2.4.1 Advantages of Arduino

There are many advantages of using Arduino such as:

- Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than 50$.
- The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.
- The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the audio video remote control(AVRC) programming language on which it's based. Similarly, we can add AVRC code directly into our Arduino programs if we want.
- The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even

relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

2.4.2 Disadvantages of Arduino

There are many disadvantages of using Arduino such as:

- ❖ Arduino uses its own C language with core written on java byte code which is burned to controller, A lot of times this make micro-controller consume more energy as more machine cycle are used for same arithmetic operations.
- ❖ Lot of the libraries of Arduino are needs to address issues and needs a lot of improvement.
- ❖ Although there have been some use of Arduino at commercial level still it is far from competing to industrial hardware.
- ❖ Arduino strips our hardware level access barring us from many functions.

2.5 Bluetooth

We all have experienced the inconvenience that arises when we start connecting peripherals to a computer, or when we connect other electronic devices, with a lot of cables that becomes difficult to control. Then we start to think how easy it would be if all these connections were done using a

Different way from the physical cables, like infrareds, radio or microwaves.

The companies of computer science and telecommunications needed to develop an opened, low cost interface to make easier the communication between devices without using cables. This is the origin of the technology which key name is "Bluetooth". This is a fact nowadays, but now another problem arises and is that there are a lot of standards and technologies, incompatible between them. What we need now is a universal, valid device for the connection of all kinds of peripheral, and that works in a transparent way for the user. That called Bluetooth.

Bluetooth is a standard used in links of radio of short scope, destined to replace wired connections between electronic devices like cellular telephones, Personal Digital Assistants (PDA), computers, and many other devices. Bluetooth technology can be used at home, in the office, in the car, etc. This technology allows to the users instantaneous connections of voice and information between several devices in real time. The way of transmission used assures protection against interferences and safety in the sending of information.

2.5.1 The symbol of Bluetooth

As famous as the name is the Bluetooth symbol. Everybody can recognize this symbol like the Bluetooth icon, but again few of them knows the origin. Bluetooth's logo combines the representation of the Nordic runes Hagalaz(transcribed by 'H') and Berkana (transcribed by 'B') in the same symbol. This is, HB like Harald Blåtand the king.[6]

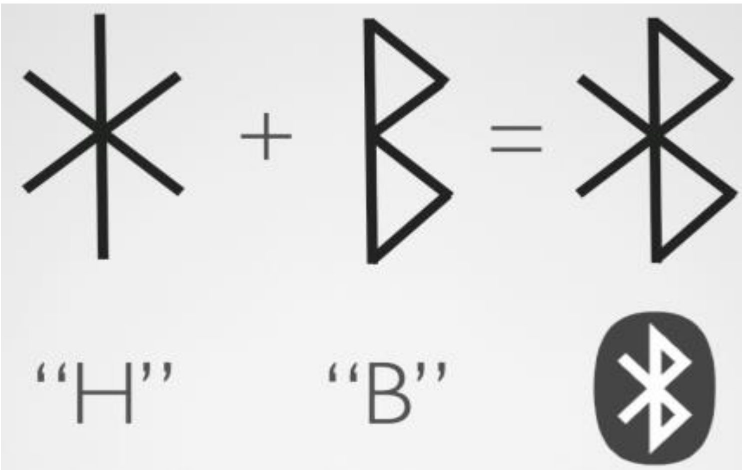


Figure2.5: The symbol of Bluetooth

2.5.2 Theory of work

Every device will have to be equipped with a microchip (transceiver) that transmits and receives in the frequency of 2.4 GHz that is available in the whole world (with some variations of bandwidth in different countries). Besides the information, there are three channels of voice available.

The information can be exchanged to speeds of up to 1 megabit for second (2 megabits for second in the Second Generation of this Technology). A scheme of “frequency hop” (jumps of frequency) allows to the devices to communicate inclusive in areas where a great electromagnetic interference exists. Besides that is provided with schemes of encryption and check.

2.5.3 Advantages of Bluetooth

There are several advantages of Bluetooth such as:

- ❖ Bluetooth does not require a clear line of sight between the synced devices. This means that the devices need not be facing each other, and it is also possible to carry out transfers when both the devices are in separate rooms.
- ❖ The maximum range that it offers is 100 meters, but this range is not the same for all similar connections. It depends on the nature of the devices and the version that they operate upon.
- ❖ The processing power and battery power that it requires in order to operate is very low. This makes it an ideal tool for so many electronic devices, as the technology can be implemented pretty much anywhere.
- ❖ One major advantage is its simplicity of use. Anyone can figure out how to set up a connection and sync two devices with ease. Moreover, the technology is completely free to use and requires no charges to be paid to any service provider.

2.5.4 Disadvantages of Bluetooth

there are many disadvantages of Bluetooth such as

- ❖ Though the transfer speeds are impressive at around 25 Mbps, certain other technologies like Wi-Fi Direct can offer speeds up to 250 Mbps. This is an area that can be improved upon in the near future.
- ❖ Even though the security is good, it is even better on Wi-Fi Direct. This can be accounted to the (comparatively) larger range of Bluetooth and also to the lack of a line of sight. Someone who knows how to hack such networks can do so eventually.
- ❖ The battery usage during a single transfer is negligible, but there are some people who leave the device switched on in their devices. This inevitably eats into the battery of these devices, and lowers the battery life considerably.

2.6 DC motor

Electrical motors are everywhere around us. Almost all the electro-mechanical movements we see around us are caused either by an A.C. or a DC motor. Here we will be exploring this kind of motors. This is a device that converts DC electrical energy to a mechanical energy.

2.6.1 Principle of DC motor

This DC or direct current motor works on the principal, when a current carrying conductor is placed in a magnetic field, it experiences a torque and has a tendency to move. This is known as motoring action. If the direction of current in the wire is reversed, the direction of rotation also reverses. When magnetic field and electric field interact they produce a mechanical force, and based on that the working principle of DC motor established.

The direction of rotation of a this motor is given by Fleming’s left hand rule which states that if the index finger, middle finger and thumb of your left hand are extended mutually perpendicular to each other and if the index finger represents the direction of magnetic field, middle finger indicates the direction of current, then the thumb represents the direction in which force is experienced by the shaft of the DC motor.[7]

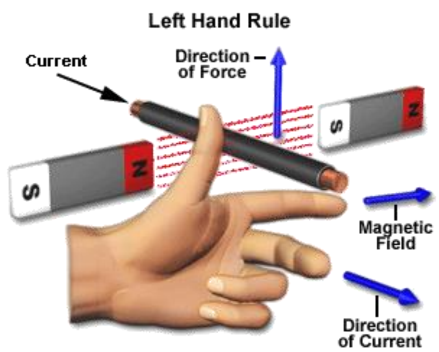


Figure 2.6: Fleming’s left hand rule

Structurally and construction wise a direct current motor is exactly similar to a DC generator, but electrically it is just the opposite. Here we unlike a generator we supply electrical energy to the input port and derive mechanical energy from the output port. We can represent it by the block diagram shown below

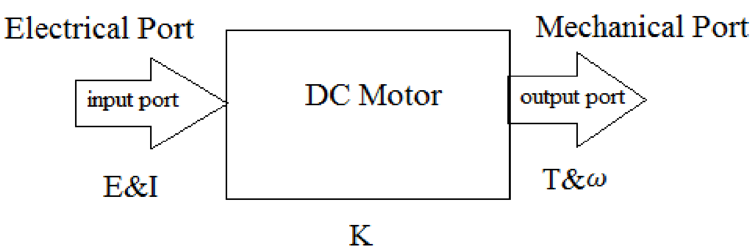


Figure 2.7: Block diagram of DC motor

Here in a DC motor, the supply voltage E and current I is given to the electrical port or the input port and we derive the mechanical output i.e. torque T and speed ω from the mechanical port or output port.

The input and output port variables of the direct current motor are related by the parameter K.

So from the Figure 2.7 above we can well understand that motor is just the opposite phenomena of a DC generator, and we can derive both motoring and generating operation from the same machine by simply reversing the ports.

2.6.2 Construction of DC motor

A DC motor like we all know is a device that deals in the conversion of electrical energy to mechanical energy and this is essentially brought about by two major parts required for the construction of DC motor, namely.

- ➢ Stator – The static part that houses the field windings and receives the supply and,
- ➢ Rotor – The rotating part that brings about the mechanical rotations.

Other than that there are several subsidiary parts namely the

- ❖ Yoke of DC motor.
- ❖ Poles of DC motor.
- ❖ Field winding of DC motor.
- ❖ Armature winding of DC motor.
- ❖ Commutator of DC motor.
- ❖ Brushes of DC motor.

Now let's do a detailed discussion about all the essential parts of DC motor.

- ✓ **Yoke of DC motor**

The magnetic frame or the yoke of DC motor made up of cast iron or steel and forms an integral part of the stator or the static part of the motor. Its main function is to form a protective covering over the inner sophisticated parts of the motor and provide support to the armature. It also supports the field system by housing the magnetic poles and field winding of the DC motor.

- **Poles of DC motor**

The magnetic poles of DC motor are structures fitted onto the inner wall of the yoke with screws. The construction of magnetic poles basically comprises of two parts namely, the pole core and the pole shoe stacked together under hydraulic pressure and then attached to the yoke. These two structures are assigned for different purposes, the pole core is of small cross sectional area and its function is to just hold the pole shoe over the yoke, whereas the pole shoe having a relatively larger cross-sectional area spreads the flux produced over the air gap between the stator and rotor to reduce the loss due to reluctance. The pole shoe also carries slots for the field windings that produce the field flux.

- **Field winding of DC motor**

The field winding of DC motor are made with field coils (copper wire) wound over the slots of the pole shoes in such a manner that when field current flows through it, then adjacent poles have opposite polarity are produced. The field winding basically form an electromagnet, that produces field flux within which the rotor armature of the DC motor rotates, and results in the effective flux cutting.

- **Armature winding of DC motor**

The armature winding of DC motor is attached to the rotor, or the rotating part of the machine, and as a result is subjected to altering magnetic field in the path of its rotation which directly results in magnetic losses. For this reason the rotor is made of armature core, that's made with several low-hysteresis silicon steel lamination, to reduce the magnetic losses like hysteresis and eddy current loss respectively. These laminated steel sheets are stacked together to form the cylindrical structure of the armature core. The armature core are provided with slots made of the same material as the core to which the armature winding made with several turns of copper wire distributed uniformly over the entire periphery of the core. The slot openings a shut with fibrous wedges to prevent the conductor from plying out due to the high centrifugal force produced during the rotation of the armature, in presence of supply current and field.

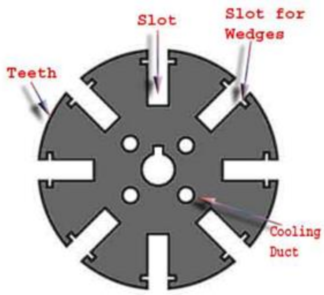


Figure 2.8: Armature core

The construction of armature winding of DC motor can be of two types:

- Lap winding: In this case the number of parallel paths between conductors A is equal to the number of poles P.
- Wave winding: Here in this case, the number of parallel paths between conductors A is always equal to 2 irrespective of the number of poles. Hence the machine designs are made accordingly.

- ✓ **Commutator of otor**

The commutator of DC motor is a cylindrical structure made up of copper segments stacked together, but insulated from each other by mica. Its main function as far as the DC motor is concerned is to commute or relay the supply current from the mains to the armature winding housed over a rotating structure through the brushes of DC motor.

- ✓ **Brushes of DC motor**

The brushes of DC motor are made with carbon or graphite structures, making sliding contact over the rotating commutator. The brushes are used to relay the current from external circuit to the rotating commutator form where it flows into the armature winding. So, the commutator and brush unit of the DC motor is concerned with transmitting the power from the static electrical circuit to the mechanically rotating region or the rotor.

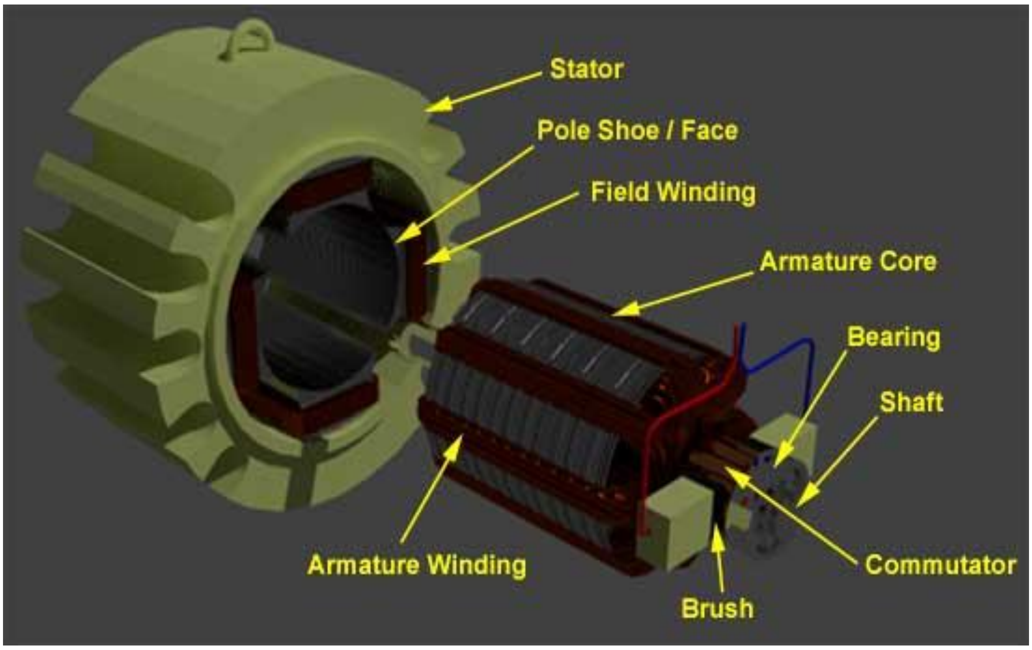


Figure 2.9: Construction of DC motor

2.6.3 Detailed description of a DC motor

The direct current motor is represented by the figure 2.10 below, on which is mounted the brushes, where we connect the external terminals, from where supply voltage is given. On the mechanical terminal we have a shaft coming out of the motor, and connected to the armature, and the armature-shaft is coupled to the mechanical load. On the supply terminals we represent the armature resistance Ra in series. Now, let the input voltage E, is applied across the brushes. Electric current which flows through the rotor armature via brushes, in presence of the magnetic field, produces a torque Tg . Due to this torque Tg the DC motor armature rotates. As the armature conductors are carrying currents and the armature rotates inside the stator magnetic field, it also produces an Electro Motive Force (EMF) Eb in the manner very similar to that of a generator. The generated EMf Eb is directed opposite to the supplied voltage and is known as the back EMF, as it counters the forward voltage.

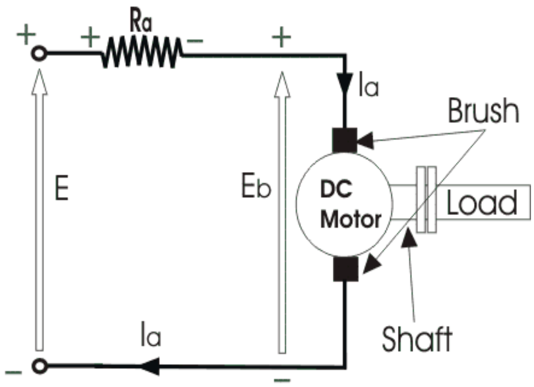


Figure 2.10: Circuit diagram of DC motor

The back emf like in case of a generator is represented by:

$$Eb = \frac{P.\varphi.Z.N}{60.A} \tag{2.1}$$

So from the above equation we can see Eb is proportional to speed ‘N’. That is whenever a direct current motor rotates, it results in the generation of back EMF. Now lets represent the rotor speed by ω in rad/sec. So Eb is proportional to ω.

So when the speed of the motor is reduced by the application of load, Eb

decreases. Thus the voltage difference between supply voltage and back emf increases that means E − Eb increases. Due to this increased voltage difference, armature current will increase and therefore torque and hence speed increases. Thus a DC motor is capable of maintaining the same speed under variable load.

Now armature current Ia is represented by

$$I_a = \frac{E - E_b}{R_a} \tag{2.2}$$

Now at starting, speed ω = 0 so at starting Eb = 0.

$$I_a = \frac{E}{R_a} \tag{2.3}$$

Now since the armature winding electrical resistance Ra is small, this motor has a very high starting current in the absence of back EMF. As a result we need to use a starter for starting a DC motor.

Now as the motor continues to rotate, the back Emf starts being generated and gradually the current decreases as the motor picks up speed.

2.6.4 Torque equation of DC motor

Referring to the Figure 2.10, we can see, that if E is the supply voltage, Eb is the back emf produced and Ia, Ra are the armature current and armature resistance respectively then the voltage equation is given by,

$$E = Eb + IaRa \tag{2.4}$$

But keeping in mind that our purpose is to derive the torque equation of DC motor we multiply both sides of equation (2.4) by Ia.

$$EIa = EbIa + Ia^2Ra \tag{2.5}$$

Now Ia^2Ra is the power loss due to heating of the armature coil, and the true effective mechanical power that is required to produce the desired torque of DC machine is given by,

$$Pm = EbIa \tag{2.6}$$

The mechanical power Pm is related to the electromagnetic torque Tg as,

$$Pm = Tg\omega \tag{2.7}$$

Now equating equation (2.6) and (2.7), we get:

$$EbIa = Tg\omega \tag{2.8}$$

Now for simplifying the torque equation of DC motor we substitute

$$Eb = \frac{P.\varphi.Z.N}{60.A}$$

$$\omega = \frac{2\pi N}{60} \tag{2.9}$$

Substituting equation (2.1) and (2.9) in equation (2.8), we get:

$$Tg = \frac{P.Z.\varphi.Ia}{2\pi A} \tag{2.10}$$

The torque we so obtain, is known as the electromagnetic torque of DC motor, and subtracting the mechanical and rotational losses from it we get the mechanical torque.

Therefore, Tm = Tg – mechanical losses.

This is the torque equation of DC motor. It can be further simplified as:

$$Tg = Ka\varphi Ia \tag{2.11}$$

Where

$$Ka = \frac{P.Z}{2\pi A} \quad (2.12)$$

Which is constant for a particular machine and therefore the torque of DC motor varies with only flux φ and armature current Ia.

2.6.5 Types of DC motor

The direct current motor or the DC motor has a lot of application in today's field of engineering and technology. Starting from an electric shaver to parts of automobiles, in all small or medium sized motoring applications DC motors come handy. And because of its wide range of application different functional types of DC motor are available in the market for specific requirements.

The DC motors are divided mainly to:

- Brush DC motors (BDC).
- Brushless DC motors (BLDC).

The different types of BDC motors are distinguished by the construction of the stator or the way the electromagnetic windings are connected to the power source. These types are:

- Permanent Magnet.
- Shunt-Wound.
- Series-Wound.
- Compound-Wound.
- Separately excited DC motor.
- Universal motor.
- DC servomotors.

2.7 DC servomotors

We execute our project using DC servomotor which are also called control motors. Unlike large industrial motors, they are not used for continuous energy conversion but only for precise speed and precise position control.

DC servomotors differ in application capabilities from large industrial motors in the following respects:

- They produce high torque at all speeds including zero speed.
- They are capable of holding a static (i.e. no motion) position.
- They do not overheat at standstill or lower speeds.
- They are able to accelerate and decelerate quickly.
- They are able to return to a given position time after time without any drift.[7]

DC Servo motor mechanism is used in a large number of applications which are critical in position control.

One of the most popular DC servomotor applications is robotic as shown in figure 2.11 below. which is used to pick an object from one position and place the object at different position.

Also DC servomotors can be used in conveyors system as shown in figure 2.12 below which are used in industrial manufacturing and assembling units to pass an object from one assembly station to another.

Also DC servomotors can be used in solar tracking system as shown in figure 2.13 below. if we attach DC servomotors to the solar panel s in such a way that we are able to precisely control its angle of movement so that it closely follows the Sun, then the overall efficiency of the system vastly increases.

Figure 2.11: DC servomotor in robotics

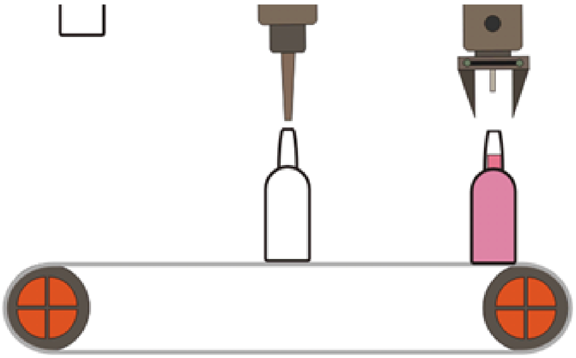


Figure 2.12: DC servomotor in conveyors

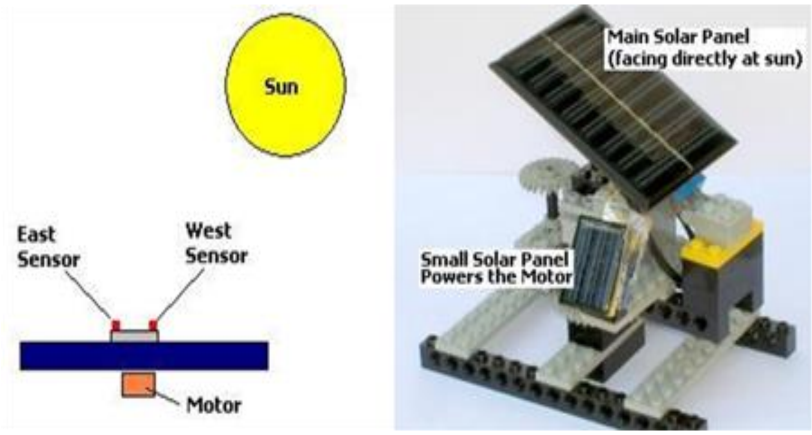


Figure 2.13: DC servomotor in solar Tracking System

However, we will explain DC servomotors construction, design and principle of operation in next chapter.

2.8 Android

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 touchscreen input, it has also been used in game consoles, digital cameras, regular PCs, and other electronics. As of 2015, Android has the largest installed base of all operating systems.

Android phones are highly customizable and as such can be altered to suit our tastes and needs with wallpapers, themes and launchers which completely change the look of our device's interface. We can download applications to do all sorts of things like check our Facebook and Twitter feeds, manage our bank account, order pizza and play games. We can plan events on from our phone's calendar and see them on our computer or browse websites on our desktop and pick them up on our phone.

2.8.1 Android Applications

Applications ("apps"), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, the Java programming language that has complete access to the Android ion programming interface (APIs). Java may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support the Go programming language is also supported since its version 1.4, which can also be used exclusively although with a restricted set of Android APIs .The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on quick emulator (QEMU), documentation, sample code, and tutorials. Initially, Google's supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on intelligent idea, as its primary IDE for Android application development. Other development tools are available, including a native development kit (NDK) for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks. In January 2014, Google unveiled an framework based on Apache Cordova for porting Chrome hypertext markup language (HTML) 5 web applications to Android, wrapped in a native application shell.

Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK (Android application package) file, or by downloading them using an application store program that allows users to install, update, and remove applications from their devices. Google Play Store is the primary application store installed on Android devices that comply with Google's compatibility requirements and license the Google Mobile Services software. Google Play Store allows users to browse, download and update applications published by Google and third-party developers; As of July 2013, there are more than one million applications available for Android in Play Store. As of May 2013, 48 billion applications have been installed from Google Play Store and in July 2013, 50 billion applications were installed. Some carriers offer direct carrier billing for Google Play application purchases, where the cost of the application is added to the user's monthly bill.

2.8.2 Updates of Android software

Operating systems which power computing devices, from large desktop workstations to pocket-sized smartphones, are targeted by updates at regular intervals which are meant to fix technical glitches or improve upon the digital services offered to users. There are few, if any, operating systems which have received more updates than the Android system.

Google provides major incremental upgrades to Android every six to nine

months, with confectionery-themed names, which most devices are capable of receiving over the air. The latest major release is Android 6.0 "Marshmallow".

2.8.3 Advantages of Android

There are several advantages of Android such as:

- **Multitasking:** Android phones can run many applications, it means you can browse Facebook while listened to the song.
- **Ease of Notification:** Any short message service (SMS), Email, or even the latest articles from an rich site summary (RSS) Reader, there will always be a notification on the Home Screen Android phone, do not miss the LED indicator is blinking, so you will not miss a single SMS, Email or even Misscall .
- Easy access to thousands of applications via the Google Android App Market.
- **Can install a modified ROM:** not satisfied with the standard view of Android, do not worry there are many custom ROM that can be used in our mobile phones Android.
- **Widget:** absolutely right, with the widgets on the home screen, You can easily access a variety of settings quickly and easily.

2.8.4 Disadvantages of Android

There are several disadvantages of Android such as:

- **Continuous Internet connection:** most Android phones require a simultaneous Internet connection alias continuously active.
- **Advertising:** Application in the Android phones can indeed be obtained easily and for free, but the consequences in each of these applications, will always be ads on display, either the top or bottom of the application.

CHAPTER THREE

CIRCUIT COMPONENTS

3.1 Introduction

In this chapter we will talk about circuit components. Taking each component separately giving definition about its construction, operation, method of connection and function.

All these components controlled by Android smartphone (mobile phone with an advanced mobile operating system) .They typically combine the features of a cell phone with those of other popular mobile devices, such as personal digital assistant (PDA), media player and GPS navigation unit. Most smartphones have a touchscreen user interface, can run third-party apps and are camera phones. Most Smartphones produced from 2012 onwards also have high-speed mobile broadband 4G LTE internet, motion sensors, and mobile payment mechanisms.

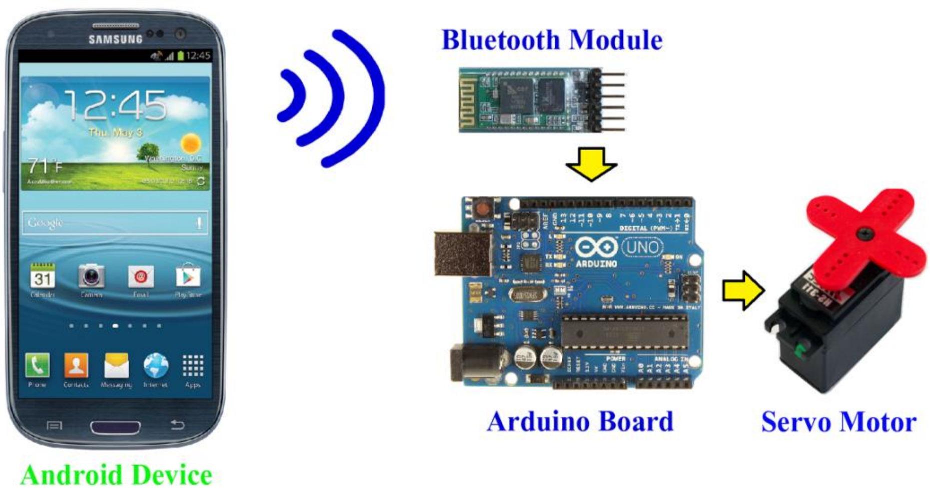


Figure 3.1: Circuit components

3.2 Bluetooth module

Bluetooth is a wireless technology standard for exchanging data over short distances using short-wavelength ultra-high frequency (UHF) radio waves in the institute for supply management (ISM) band from (2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs).

The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication .We can use it simply for a serial port replacement to establish connection between MCU and global positioning system (GPS).

Table 3.1: Specifications of HC-05 Bluetooth module

Protocol	Bluetooth Specification v2.0+EDR
Frequency	2.4GHz ISM band
Emission power	≤4dBm, Class 2
Speed of transfer data	2.1Mbps (Max)
Security	Authentication and encryption
Profiles	Bluetooth serial port
Power supply	+3.3VDC 50mA
Working temperature	-20 - +75 Centigrade
Dimension	26.9mm x 13mm x 2.2 mm

3.2.1 Description of Bluetooth module

The HC-05 Bluetooth Module has 6 pins (VCC, GND, TX, RX, Key, and LED). It comes pre-programmed as a slave, so there is no need to connect the Key pin, unless we need it change it to Master Mode.

The major difference between Master and Slave modes is that, in Slave mode the Bluetooth module cannot initiate a connection, it can however accept incoming connections.

After the connection is established the Bluetooth module can transmit and receive data regardless of the mode it is running in. If we are using a phone to connect to the Bluetooth module, we can simply use it in the Slave mode. The default data transmission rate is 9600kbps.

The range for Bluetooth communication is usually 30m or less. The module has a factory set pin of “1234” which is used while pairing the module to a phone.

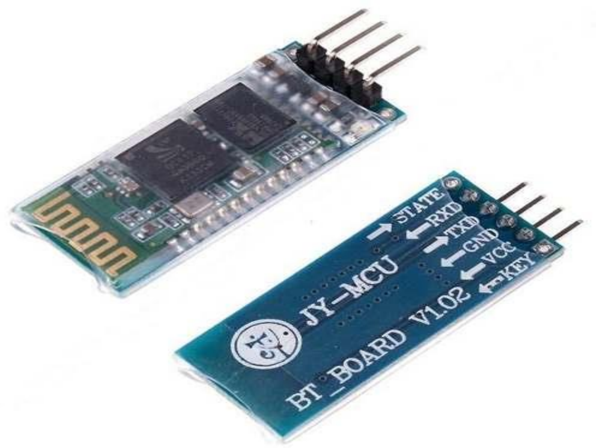


Figure 3.2: Hc-05 Bluetooth module

3.2.2 Connection of Bluetooth module

HC-05 Bluetooth module are connected through four-wire connection, one wire for a DC power supply, one for grounding and others wires for receiving and sending data.

Table 3.2: connection of Bluetooth module

HC-05	Arduino
GND	GND
VCC	3.3/5V
TX	RX
RX	TX

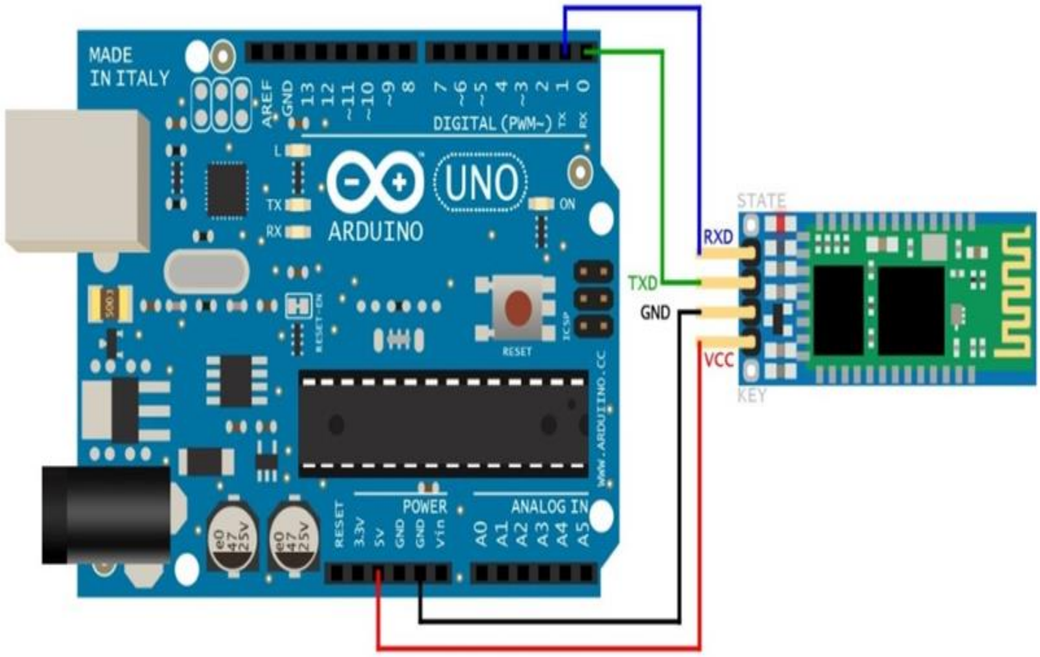


Figure 3.3: Connection of Bluetooth module

3.2.3 function of Bluetooth module

Bluetooth module operate as receiver and transmitter.it received signals from android smartphone wireless and then transmitting these signals to Arduino board.

3.3 Arduino board

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or Integrated Development Environment (IDE) that runs on our computer, used to write and upload computer code to the physical board.

3.3.1 Construction of Arduino board

There are many varieties of Arduino boards that can be used for different purposes. but most Arduinos have the majority of these components in table 3.3 below

Table 3.3: Construction of Arduino UNO board

Component	Number on board
USB connection	1
Barrel jack	2
GND Pin	3
5v Pin	4
3.3v Pin	5
Analog Pins	6
Digital Pins	7
PWM	8
AREF	9
Reset button	10
Power LED indicator	11
TX RX LEDs	12
Main IC	13
Voltage Regulator	14

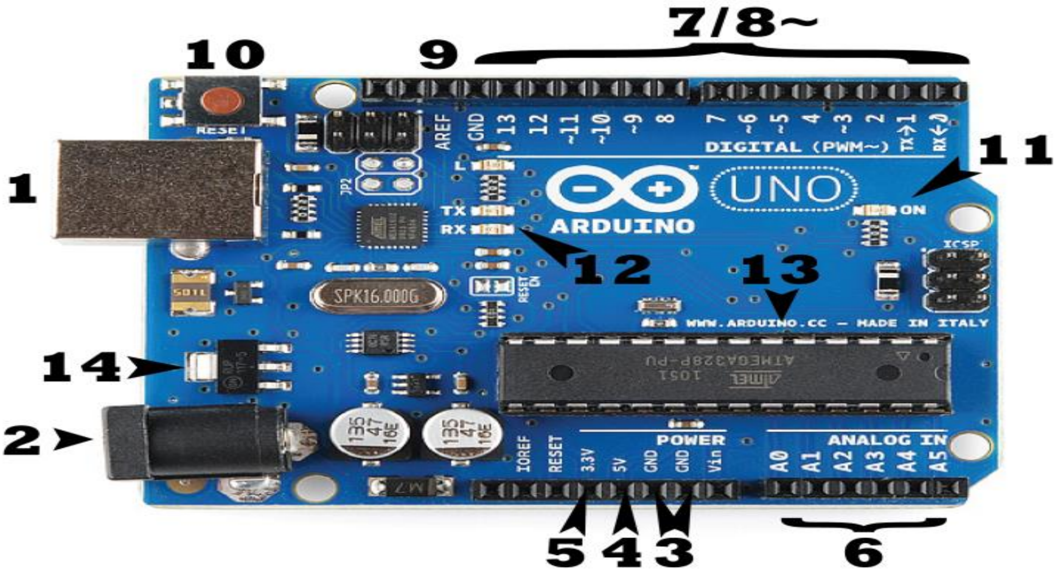


Figure 3.4: Construction of Arduino UNO board

❖ **Power (USB / Barrel Jack)**

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from our computer or a wall power supply that is terminated in a barrel jack. In the figure 3.4 above, the USB connection is labeled (1) and the barrel jack is labeled (2).

The USB connection is also used to load code onto our Arduino board.

❖ **Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)**

The pins on our Arduino are the places where we connect wires to construct a circuit (probably in conjunction with a breadboard and some wire). They usually have black plastic 'headers' that allow to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

• GND (3)

Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground our circuit.

• 5V (4) and 3.3V (5)

As we might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

• Analog (6)

The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.

• Digital (7)

Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

• PWM (8)

These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).these pins as being able to simulate analog output (like fading an LED in and out).

• **AREF (9)**

Stands for Analog Reference. Most of the time we can leave this pin alone. It is sometimes used to set an external reference voltage

(between 0 and 5 Volts) as the upper limit for the analog input pins.

❖ **Reset Button**

the Arduino has a reset button (10). Pushing it will temporarily connect

the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if our code doesn't repeat, but we want to test it multiple times.

❖ **Power LED Indicator**

Just beneath and to the right of the word "UNO" on our circuit board, there's a tiny LED next to the word 'ON'.This LED should light up whenever we plug our Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to recheck our circuit.

❖ **TX RX LEDs**

TX is short for transmit, RX is short for receive. These markings Appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs (12). These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

❖ **Main IC**

The black thing with all the metal legs is an IC, or Integrated Circuit (13). Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as we may need to know the IC type (along with our board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC.

- **Voltage Regulator**

The voltage regulator (14) is not actually something we can interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up our Arduino to anything greater than 20 volts.

Table 3.4: Specifications of Arduino UNO

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25g

3.3.2 function of Arduino board

Arduino board represents the core of our circuit, it operates to interpret the received signals from Bluetooth module and sends these signals to activate and control DC servomotor.

3.4 DC servomotor

DC servomotor is a type of motors whose output shaft can be moved to a specific angular position by sending it a coded signal. The DC servomotor will maintain the position of the shaft as long as we keep applying the coded signal. When we change the coded signal, the angular position of the shaft

will change.

3.4.1 Construction of DC servomotor

A typical servo consists of a small electric motor driving a train of reduction gears. A potentiometer is connected to the output shaft. Some simple electronics provide a closed-loop servomechanism.

- **The controller circuit:** This is the "brain" of the servo. This circuit is responsible to read the user's input signal (pulses) and translate it into a motor revolution in such a way, that the drive shaft will be rotated to the desired position.

- **The feedback potentiometer:** The shaft of the potentiometer is attached to the drive shaft of the servo. When the drive shaft rotates, so does the potentiometer. In that way, each and every rotation angle of the drive shaft, corresponds to a different resistance of the potentiometer. By reading the potentiometers' resistance, the controller is able to know the exact angle of the drive shaft of the servo.

- **The motor:** This is usually a small high speed DC motor.

▪ **The gearbox:** The gearbox will drive the motor's revolution to the drive shaft. Also, the revolution per minute will be significantly reduced and the torque will be increased. The torque is one of the main characteristics of servos.

▪ **The drive shaft:** When all of the above operate in perfect harmony, the drive shaft will be rotated with accuracy to the our requested angle.

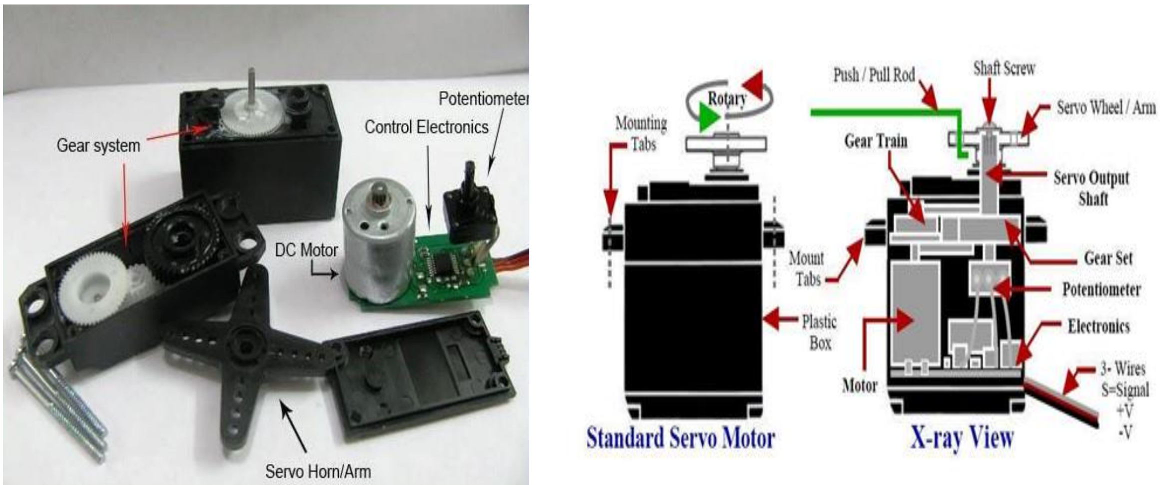


Figure 3.5: Construction of standard DC servomotor

3.4.2 Operation of DC servomotor

The position of the output, measured by the potentiometer, is continually compared to the commanded position from the control. Any difference gives rise to an error signal in the appropriate direction, which drives the electric motor either forwards or backwards, and moving the output shaft to the commanded position.

When the servo reaches this position, the error signal reduces and then becomes zero, at which point the servo stops moving.

If the servo position changes from that commanded, whether this is because

the command changes, or because the servo is mechanically pushed from its set position, the error signal will re-appear and cause the motor to restore the servo output shaft to the position needed.[8]

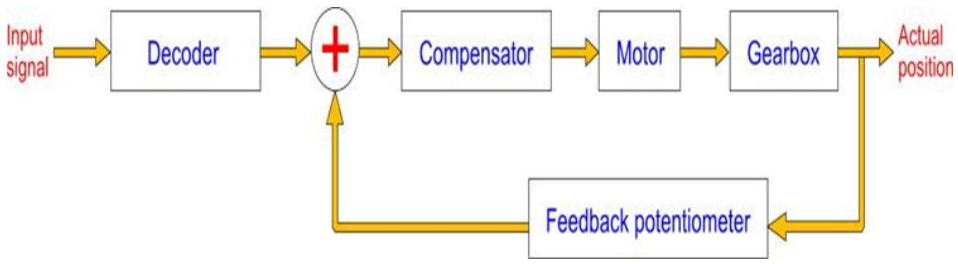


Figure 3.6: Block diagram of standard DC servomotor

3.4.3 Connection of DC servomotor

Standard DC servomotors are connected through three-wire connection, two wires for a DC power supply and one for control, carrying a PWM signal. Each servo has a separate connection and PWM signal .This signal is easily generated by simple electronics, or by microcontrollers such as the Arduino.

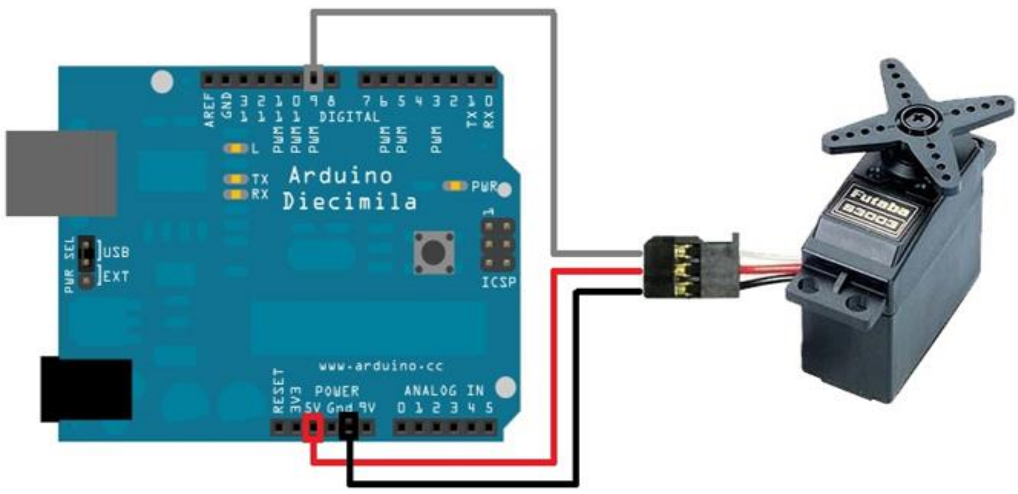


Figure 3.7: Connection of standard DC servomotor

3.4.4 function of DC servomotor

The servo move based on the pulses sent over the control wire, which set the angle of the actuator arm. The servo expects a pulse every 20 ms in order to gain correct information about the angle. The width of the servo pulse dictates the range of the servo's angular motion.

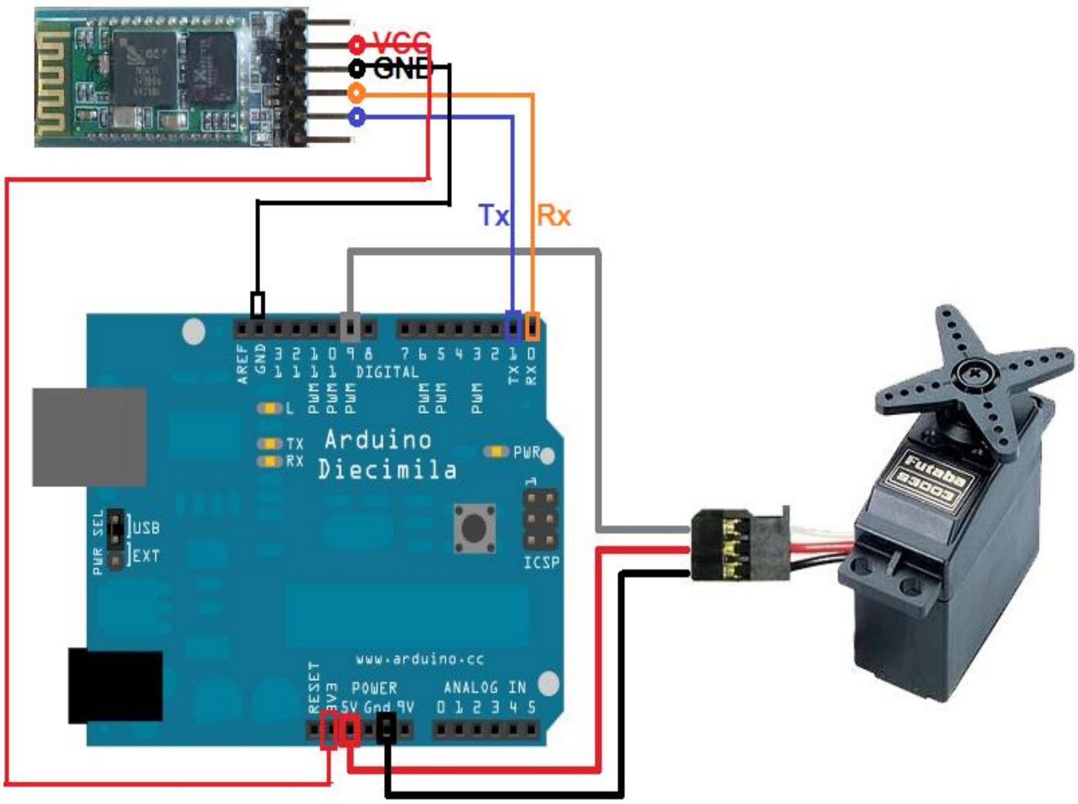


Figure 3.8: Connection of circuit components

CHAPTER FOUR

SIMULATION AND PRACTICAL RESULTS

4.1 Introduction

Simulation is the imitation of the operation of a real-world process or system over time. The act of simulating something firstly requires that a model to be developed .The model represents the system itself, whereas the simulation represents the operation of the system over time at different conditions.

Simulation is used in many contexts, such as simulation of technology for performance optimization, safety engineering, testing, training, education, and video games. Often, computer experiments are used to study simulation models. Simulation is also used with scientific modelling of natural systems or human systems to gain insight into their functioning.[9]

4.2 Proteus

There are several simulation software can be used to represent different types of systems .Proteus is one of most common Simulation software which based on the process of modeling a real phenomenon with a set of mathematical formulas. It is, essentially, a program that allows the user to observe an operation through simulation without actually performing that operation. Proteus is used widely to design equipment so that the final product will be as close to design specs as possible without expensive in process modification.

In addition to imitating processes to see how they behave under different conditions, simulations are also used to test new theories. After creating a theory of causal relationships, the theorist can codify the relationships in the form of a computer program. If the program then behaves in the same way as the real process, there is a good chance that the proposed relationships are correct.

4.3 Steps of simulation

We will use Proteus (ISIS) for designing circuit by following these steps

✓ Create a new project and give name to the project.

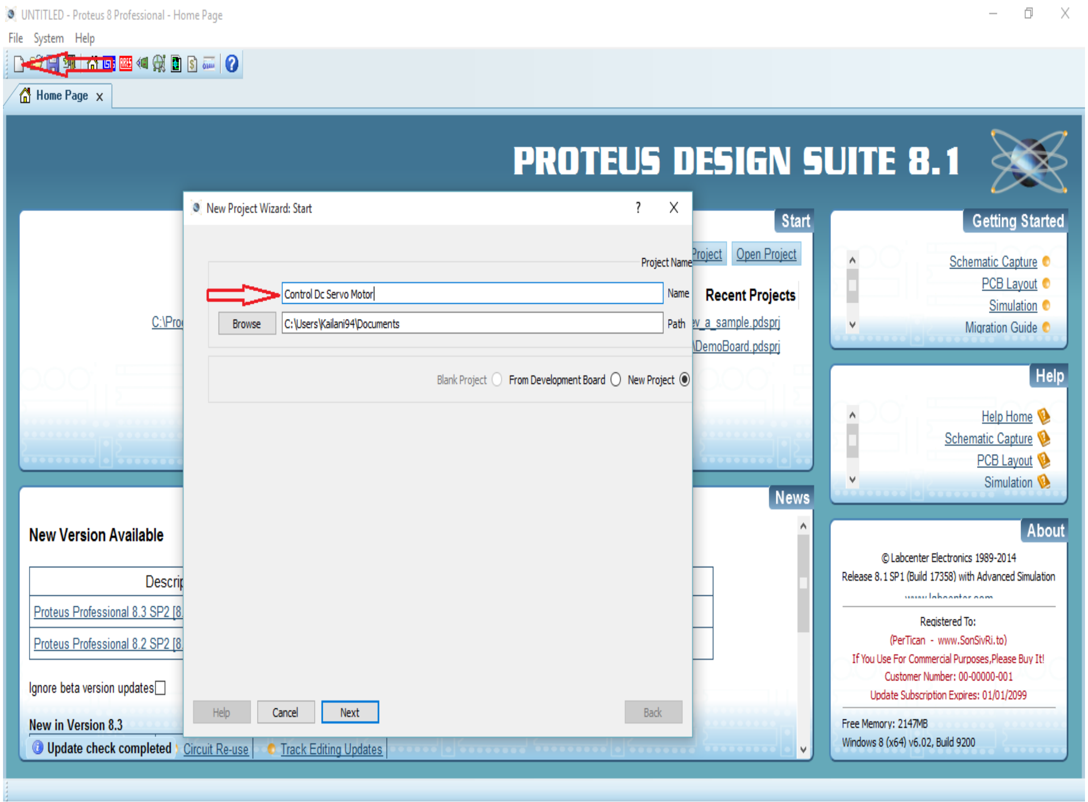


Figure 4.1: Step one of simulation

✓ Draw the circuit diagram, by click Schematic Capture button, will be directed to schematic panel.

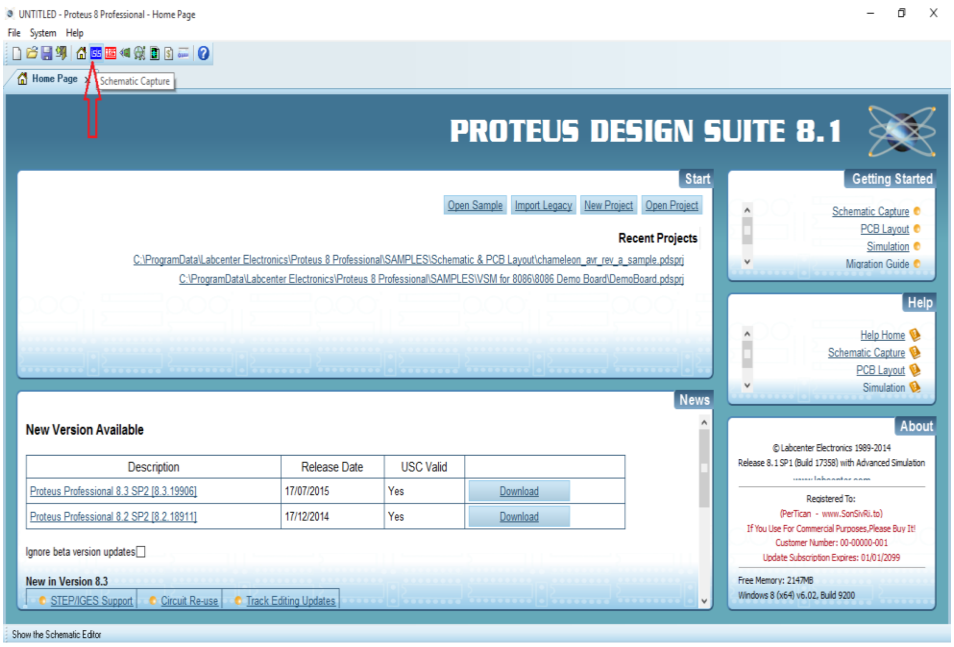


Figure 4.2: Step two of simulation

✓ Choose our component (Arduino Uno, DC servo motor and COMPIM) by typing the name at Keyword box. After selecting item we add them to workspace . and connect between components. COMPIM operates as Bluetooth. sends and receives signals.

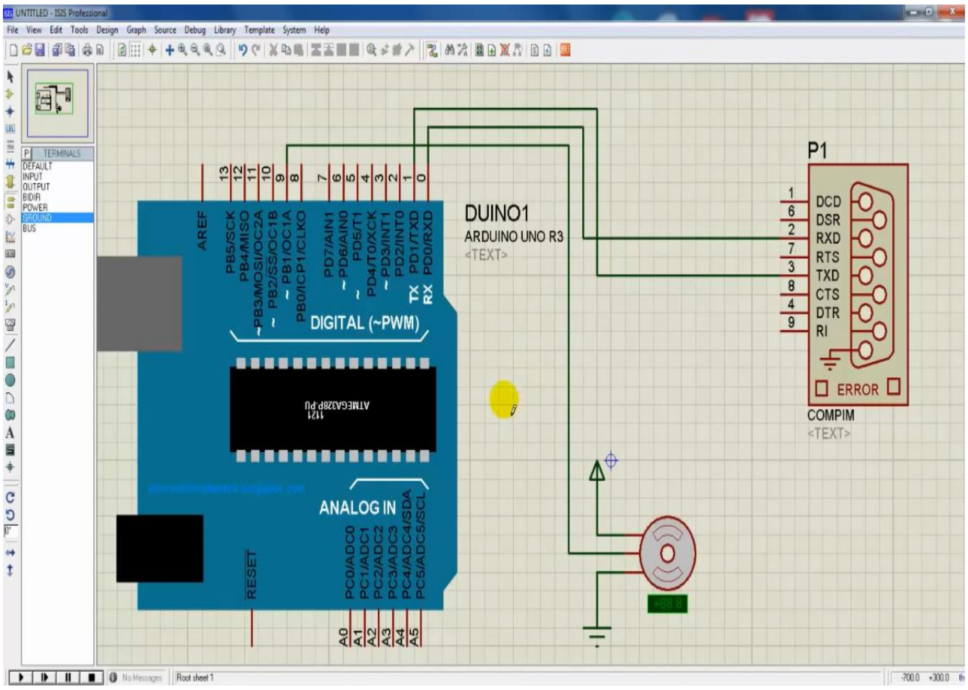


Figure 4.3: Step three of simulation

✓ Open Arduino software and write code of controlling DC Servomotor.

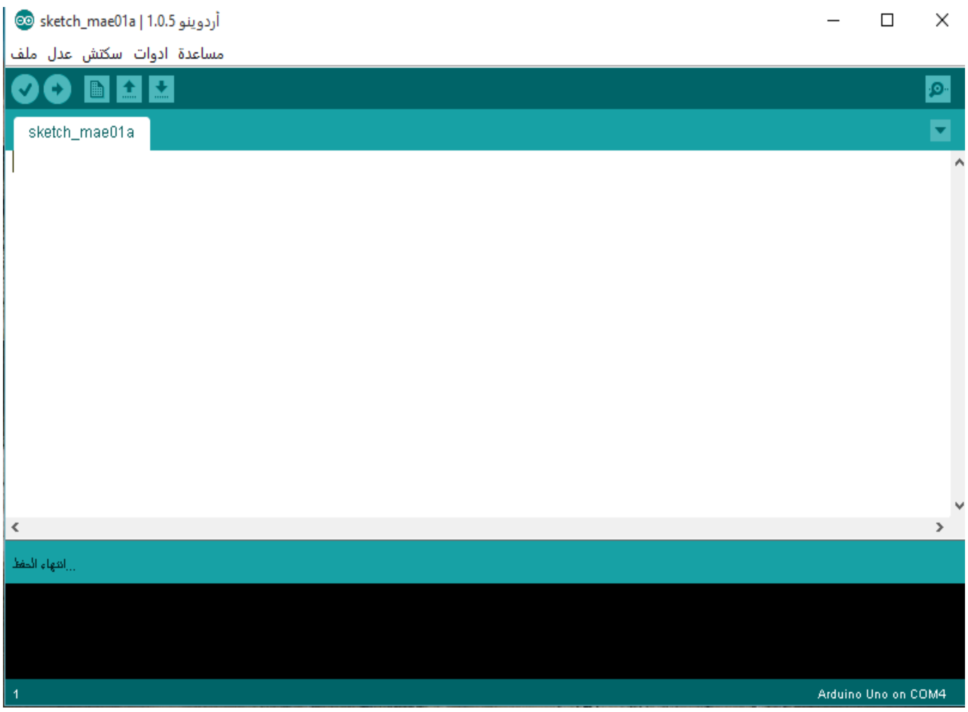


Figure 4.4: Step four of simulation

- ✓ Go to Proteus ISIS and press double click on Arduino Uno and upload the code.
- ✓ Open Visual Studio Professional software and design Track Bar which will represent Android Smartphone. and then add this Track Bar to Proteus ISIS. This Track Bar allows us Control 0 -180 Degree of motor Position by sending signals to COMPIM which sends these signals to Arduino board which interpretation signals to control motor.

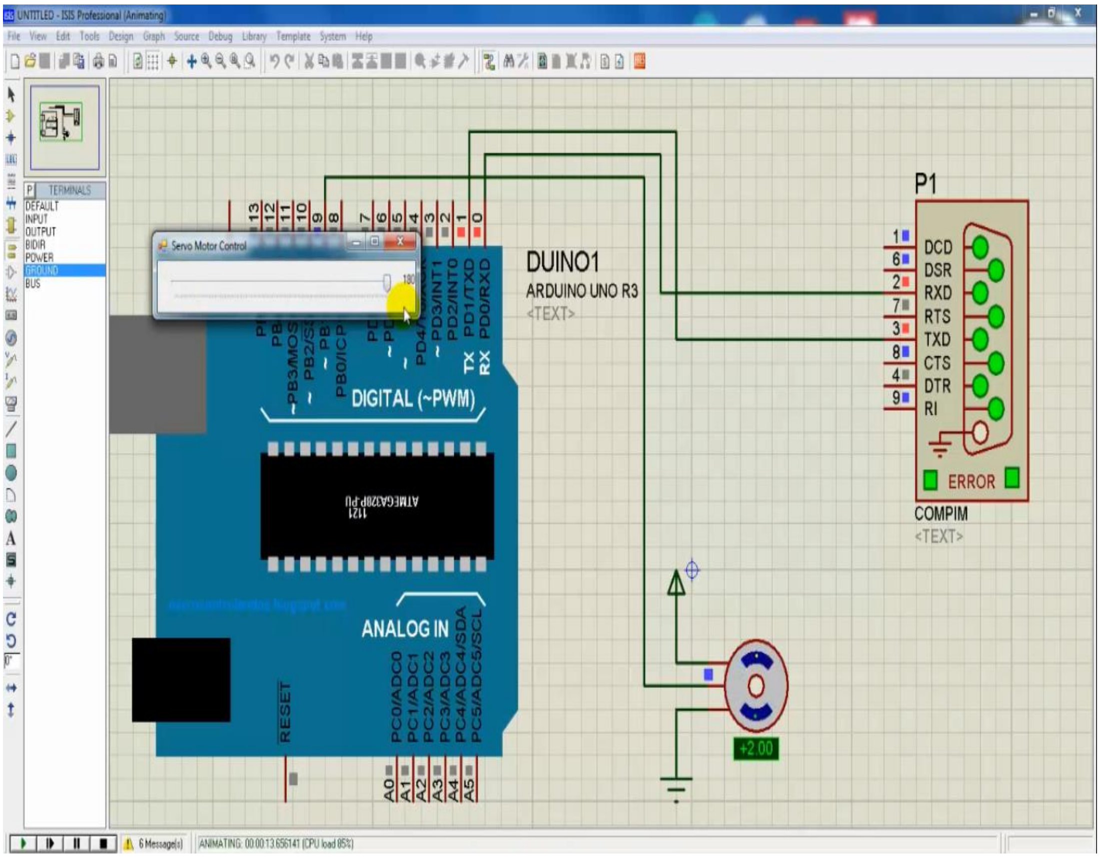


Figure 4.5: Step six of simulation

4.4 Operation of the circuit

Arduino UNO board has been installed in shingle and linked with Bluetooth Module (HC-05) , after that DC servomotor has been connected with Arduino UNO Board. Which supplied from 9V battery to provide necessary Voltage to it to work. After battery connected , All components has become standby to work. Bluetooth became ready to receiving and sending data, Arduino became ready to interpret signals and send them as commands to control DC servomotor.

When application opened from Android mobile phone ,application opened Bluetooth to search for near devices, HC-05 Bluetooth module founded and connected with it. After connection done, wireless signals sent from Android mobile phone to Bluetooth, these signals either certain values of angles or specific intervals to auto control function to obtain certain angles.

When these signals reached Bluetooth module ,Signals sent to Arduino UNO board which interpreted these signals according to saved code in it ,to order to act by DC servomotor.

4.5 Practical and results

We want to obtain different values of DC servomotor positions. we measured Arm angels of DC servomotor which we got by using two threads and protractor and tabulated results in table 4.1 below

Table 4.1: Degrees of DC servomotor positions

Position (degree)	Arm angle (degree)	Error
45º	41º	8.8%
90º	87º	3.3%
135º	131º	2.9%

Through project implementation we find that was small errors in Angle position of controlled DC servomotor . These errors happen cause DC servo motor Arm was not aligned with X-axis exactly at normal condition (zero

degree).

Also We use auto control function to obtain 90 degree angle after different periods of time. We measured these times by using stopwatch and take three results and take the average value for them. Finally we have tabulated results in table 4.2 below

Table 4.2: Results of auto control function

The Time period (Sec)	Measured (Sec)	Average (Sec)
2	1.88	1.87
	1.84	
	1.88	
4	3.84	3.79
	3.79	
	3.74	
8	7.97	7.94
	7.95	
	7.91	
16	15.67	15.62
	15.34	
	15.85	

there are small errors when auto control function periods measured. These errors result to time delayed between command and running stopwatch.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Theoretical aspects of project has been studied to achieve objectives of project and solving problems, so control circuit designed in chapter three, after that our circuit simulated and applied practically in chapter four. This circuit performed the purposes. And we have reached to results .

5.2 Recommendations

There are some recommendations apply them improve performance of project :

- ❖ Use Wi-Fi module instead Bluetooth module to increase range of controlling up to 40 meter.
- ❖ Couple Arduino board with data sim card, this sim card can receive command from any phone around the world, this will make control range an infinite.
- ❖ Provide Arduino board with display screen to display the angular position of DC servomotor .
- ❖ Use DC servomotor with a high torque to give ability to use our circuit industrial automation.

REFERENCES

[1] Ali Emadi, "Energy-Efficient Electric Motors Third Edition" , CRC Press, 2004.

[2] Katsuhiko Ogata, "Modern Control Engineering'', Prentice Hall, 2010

[3] Odd Jostein Svendsli, "Atmel's Self-Programming FlashMicrocontrollers" 2012.

[4] Steve Heath, "Embedded Systems Design" Newnes, 2002.

[5] David Kushner ,"The Making of Arduino", IEEE Spectrum 2011.

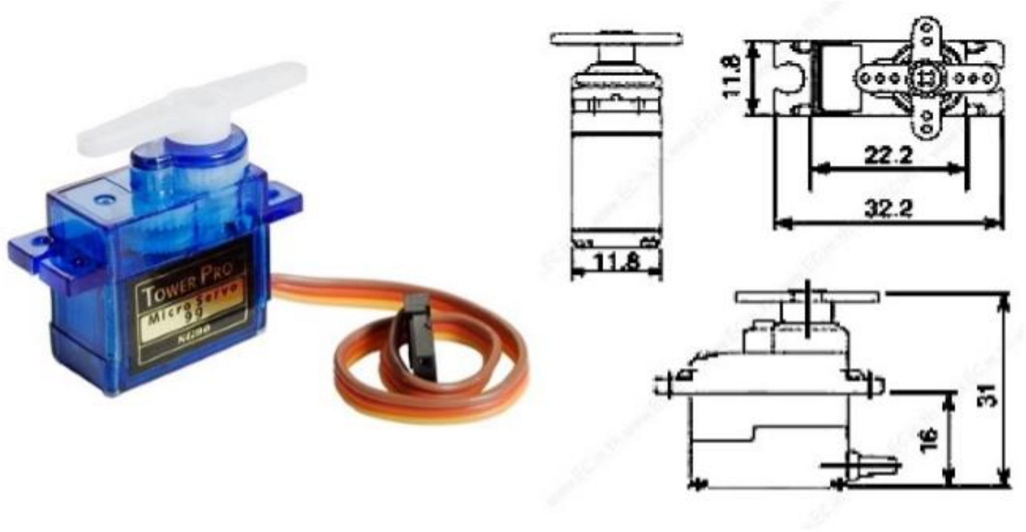
[6] Rebecca Spaker , "Bluetooth Basics Fast Facts" , 2013.

[7] A. K. Theraja, "A.T.B. Of Electrical Technology", S. Chand, 2008.

[8] Newton C. Braga," Mechatronics Sourcebook" , Thomson/Delmar Learning, 2003.

[9] J. Banks, J. Carson, B. Nelson," Discrete-Event System Simulation", D. Nicol 2001.

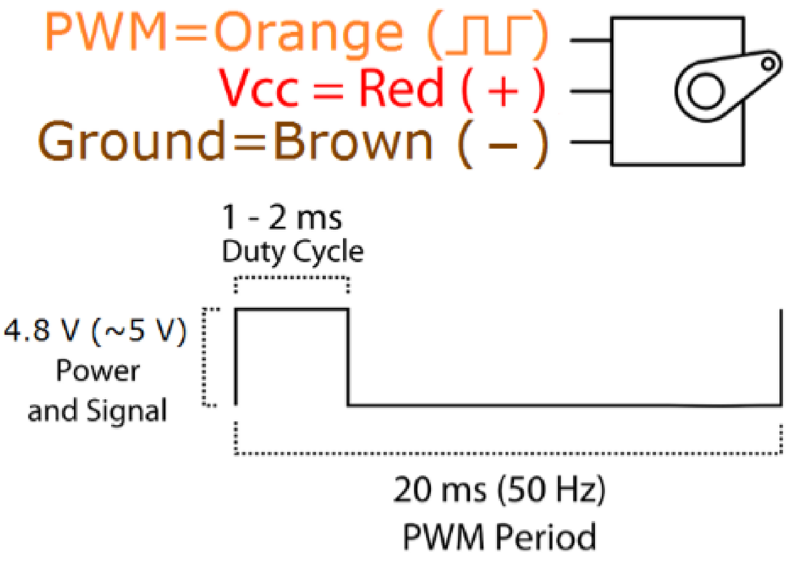
APPENDIX A



SG90 9 g Micro Servo

Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the left.   ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, " the left. ms pulse) is all the way to the right, "-90" (~1



Specifications

• **Weight:** 9 g

• **Dimension:** 22.2 x 11.8 x 31 mm approx.

• **Stall torque:** 1.8 kgf·cm

• **Operating speed:** 0.1 s/60 degree

• **Operating voltage:** 4.8 V (~5V)

• **Dead band width**: 10 µs

• **Temperature range:** 0 ºC – 55 ºC

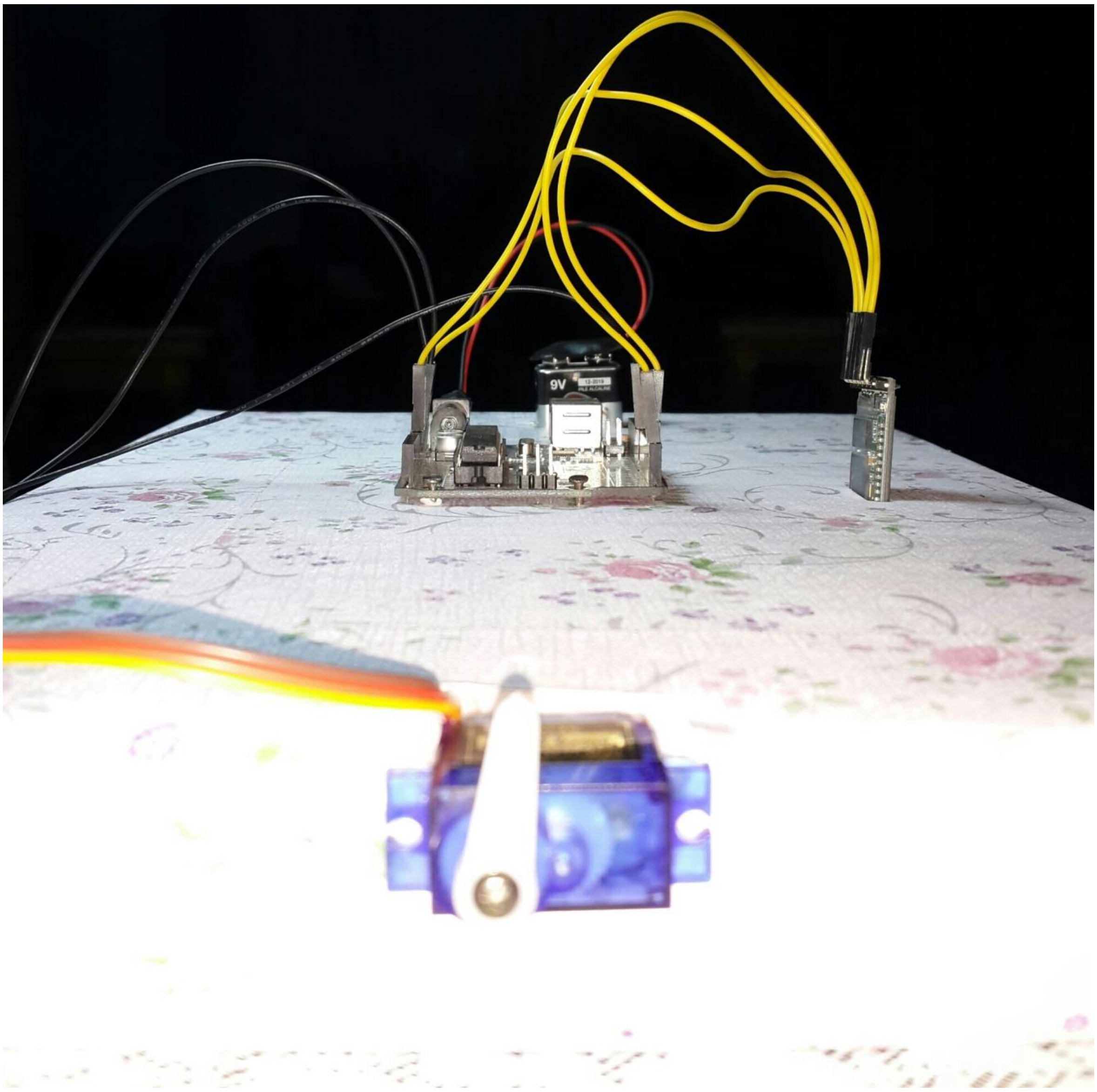
APPENDIX B

```
#include Servo.h>

 Servo myservo;  // create servo object to control a servo

                // a maximum of t servo objects can be created

int pos = 0;    // variable to store the servo position

int motor = 0;

void setup()

{

Serial.begin(9600);  // initialize serial:

myservo.attach(9);  // attaches the servo on pin 9 to the servo object

Serial.print("Arduino control Servo Motor Connected OK");

Serial.print('\n');

}

void loop()

{

// if there's any serial available, read it:

while (Serial.available() > 0) {

// look for the next valid integer in the incoming serial stream:

motor = Serial.parseInt();

// do it again:
```

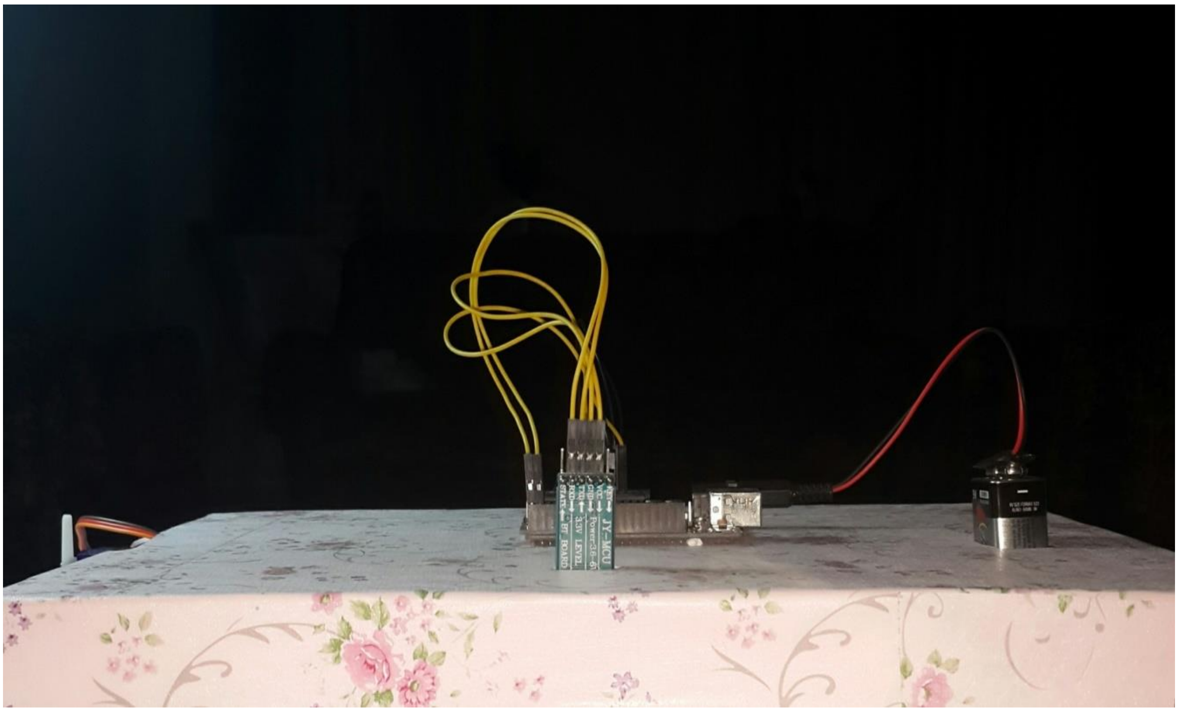
```
pos = Serial.parseInt();
// look for the newline. That's the end of your  sentence:
if (Serial.read() == '\n') {
myservo.write(pos);              // tell servo to go to position in variable 'pos'
delay(15);                       // waits 15ms for the servo to reach the position
// print the three numbers in one string as hexadecimal:
Serial.print("Data Response : ");
Serial.print(motor, HEX);
Serial.print(pos, HEX);
    }
  }
}
//for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees
 //{                                  // in steps of 1 degree
//  myservo.write(pos);              // tell servo to go to position in variable 'pos'
//  delay(15);                       // waits 15ms for the servo to reach the position
//}
//for(pos = 180; pos>=1; pos-=1)     // goes from 180 degrees to 0 degrees
//{
//  myservo.write(pos);              // tell servo to go to position in variable 'pos'
//  delay(15);                       // waits 15ms for the servo to reach the position
```

```
//}

//val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)
//val = map(val, 0, 1023, 0, 179);     // scale it to use it with the servo (value between 0 and 180)
//myservo.write(val);                  // sets the servo position according to the scaled value
//delay(15);
```

APPENDIX C



front view of our project



sidy view of our project

horizontally view of our project