“Design & Implementation of an Android Phone Controlled Fire Fighting Robot”

This project paper has been submitted to the department of ME, Faculty of Engineering and Applied Science, Bangladesh University of Sonargaon University (SU) in partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Mechanical Engineering.

A thesis

by

Md. Majaharul Islam	**ID:** BME1901017121
K.M. Shamsur Rahmann	**ID:** BME1901017641
Md. Nur Alam	**ID:** BME1901017103
MD. Asadujjaman Khan	**ID:** BME 1901017144
Md. Murad Hossain	**ID:** BME1901017182

Department Of Mechanical Engineering
SONARGAON UNIVERSITY (SU)

SEPTEMBER 2022

"Design & Implementation of an Android Phone Controlled Fire Fighting Robot"

A Thesis

by

Semester: Summer 2022

In Cooperation

Md. Majaharul Islam	**ID:** BME1901017121	
K.M. Shamsur Rahmann	**ID:** BME1901017641	
Md. Nur Alam	**ID:** BME1901017103	
MD. Asadujjaman Khan	**ID:** BME 1901017144	Asad
Md. Murad Hossain	**ID:** BME1901017182	Murad

Supervisor

NiloySarkar

NILOY SARKAR

(Lecturer)

Submitted to the

DEPARTMENT OF MECHANICAL ENGINEERING

SONARGAON UNIVERSITY (SU)

In partial fulfillment of the requirements for the award of the degree

of

Bachelor of Science in Mechanical Engineering

September 2022

DECLARATION

We hereby, declare that the work presented in this project is the outcome of the investigation and research work performed by us under the supervision of Mr. Niloy Sarkar, lecturer-department of mechanical engineering, Sonargaon University (SU).

We also declare that no part of this project and thereof has been or is being submitted elsewhere for the award of any degree.

Md. Majaharul Islam	**ID:** BME1901017121
K.M. Shamsur Rahmann	**ID:** BME1901017641
Md. Nur Alam	**ID:** BME1901017103
MD. Asadujjaman Khan	**ID:** BME 1901017144
Md. Murad Hossain	**ID:** BME1901017182

In my capacity as supervisor of the candidates thesis, I certify that the above statements are true to the best of my knowledge.

....

Niloy Sarkar

Lecturer of Mechanical Engineering

Sonargaon University (SU)

TABLE OF CONTENTS

Topics		Page No
ACKNOWLEDGEMENT		I
ABSTRACT		II

CHAPTER I: INTRODUCTION

TOPICS		Page No.
1.1	Introduction	01
1.2	Background Study	01
1.3	Objectives	02
1.4	Project Organization	03

CHAPTER II: THEORY

2.1	Introduction	04
2.2	Theory	04
2.2.1	Microcontroller (PIC16F887)	05
2.2.2	Atmega328P Pin diagram	08
2.2.3	Bluetooth module (HC05)	10
2.2.4	Motor Driver (L293D)	10
2.2.5	DC Geared Motor (wheel motor)	11
2.2.6	Mini Water Pump	12
2.2.7	Servo Motor	13
2.2.8	Transformer 220/12V	14
2.2.9	LM7805 (Voltage Regulator IC)	16
2.2.10	Crystal	16
2.2.11	Resistor	17
2.2.12	Diode (IN4007)	18
2.2.13	Variable Resistor	18
2.2.14	Vero Board	20
2.2.15	Connecting Wire	20
2.2.16	LED indicator	21

CHAPTER III: DESIGN AND FABRICATION

3.1 Introduction 22

3.2 Block Diagram 22

3.3 Circuit Diagram & Description 23

3.4 Motor gear step picture outlook and casing 24

3.5 Working Principle 24

3.6 Image of Project 26

3.7 Flow Chart 27

CHAPTER IV: DATA COLLECTION

4.1 Mechanical Efficiency 28

4.2 Volumetric Flow Rate 28

4.3 Parabolic Distance 28

4.4 Maximum Height of Discharge 28

4.5 Storage Water Discharge Time 28

CHAPTER V: RESULTS AND DISCUSSIONS

5.1 Introduction 29

5.2 Result 29

5.3 Advantages 29

5.4 Limitations 30

5.5 Cost Estimation 30

5.6 Application 30

Appendix 31~49

References 50

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
2.2.1.1	ARDUINO UNO R3	05
2.2.2.1	ATmega328P Pin Mapping	06
2.2.3.1	HC05 Bluetooth module	07
2.2.4.1	L293D Motor Driver	07
2.2.4.2	L293D Pin Diagram	07
2.2.5.1	DC Geared Motor with wheel	08
2.2.6.1	Mini Water Pump	09
2.2.7.1	Servo Motors	10
2.2.7.2	PWM Cycle of Servo motor Working	10
2.2.8.1	Transformer	11
2.2.8.2	Output Waveforms of 220V/12V rectifier	11
2.2.9.1	7805 Voltage Regulator IC	12
2.2.10.1	Crystal	13
2.2.11.1	Resistor Color Code	13
2.2.12.1	Diode (IN 4007)	14
2.2.13.1	Variable Resistor/Potentiometer	15
2.2.13.2	Different Types of Potentiometer	15
2.2.14.1	Vero Board	16
2.2.15.1	Connecting Wire	17
2.2.16.1	Indicator LED	17
3.2.1	Block Diagram of the project	18
3.3.1	Circuit Diagram of the Project	19
3.4.1	Construction of the chassis	20
3.5.1	“RoboRemo” Apps screen shot	21
3.6.1	Real Image of the robot Painting car	21
3.7.1	Working Flow Chart of the Project	22

ACKNOWLEDGEMENT

In performing our assignment, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude Mr. Nioy Sarkar, lecturer of mechanical engineering, Sonargaon University for giving us a good guideline for assignment throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

The author are also grateful to Mr. Niloy Sarkar, lecturer department of mechanical engineering and all the respected teachers of the department of mechanical engineering for their cooperation and significant help for completing the thesis work successfully.

“Authors”

ABSTRACT

The project is to develop a robotic vehicle capable of detecting the presence of fire and extinguishing it automatically. It is a movable robot that consists of camera to detecting the fire, gear motor and motor driver for the movement of the robot, relay driver for pump control and bluetooth module which are used for the detecting and extinguishing the fire. Usually, the robot moves at a steady speed. When the camera detects the fire in the robot controlled area , the signal indicating the presence of fire will be sent to the Arduino through which the extinguishing is done. In the extinguishing process , whenever the detection of fire is positive the robot will stop at the place of fire occurred and starts the pump and sprinkle water through a sprinkler until the smoke is put off. The entire control is achieved using Arduino which is interfaced with the android mobile via bluetooth module, so that the control of the robot can be made from an android mobile as well.

CHAPTER I

INTRODUCTION

1.1 Introduction

Now a day's mobile robots are very useful in construction sites, warehouses and manufacturing plants. Mobile robots can also be used in material handling applications which applications are growing day by day. For analyzing different items and for handling materials mobile robots can be used. Wireless navigation is also possible for movements of mobile robot, can be controlled through android. Fuzzy logic control mechanism is used to control robot. That model does not need any mathematical model controlling. Previously Fire Fighting Robots were controlled by using different electronics devices. But this reduces the scope of control of firefighting robot. However, with the advanced techniques we can build the same robot by using android application to control the actions of the robot. With the help of such robots, fireman's work really decreased and movements of robot are so much effective.

By using an android app fireman man detect the fire and can able to extinguish it. At the same time robot can detect the obstacles and can avoid them by using ultrasonic sensors. Our project is designed to build an android application which can control operations of the firefighting robot. Fireman can send commands to robot through Bluetooth module which is mounted on robot itself. Smart phones have facility of Bluetooth, through that Bluetooth fireman can control the movement of firefighting robot. For fire detection it is using two sensors. One is temperature sensor and second is smoke detector. Fire extinguishing system will be get activated when fire detection system detects fire. Sprinkler will start sprinkling water when it detects fire. At the transmitting end android application is used and at receiving end two motors are interface to micro-controller.

1.2 Background Study

The development of service robots became popular recently due to the fact that the society needs robots to relax humans from tedious and dangerous jobs. In Bangladesh as well as other developing countries, the increasing population stimulates the construction-related activities such as interior finishing and painting. Painting is classically done by humans and generally requires exhaustive physical efforts and involves exposure to dangerous chemicals. Chemicals can seriously impair the vision, respiratory system and general health of the human painter. These factors make painting an ideal candidate process for automation. More than 100,000 apartments are built annually Bangladesh, with an average painting area of 40 million square meters (based on an average 100 m^2) apartment area with 400 m2 painting area). The surface area of painting is more due to the renovation work and expected population increase in the future. This demand imposes challenges that will hardly be met using human painters only in the next decade.

Therefore, development of a painting machine that can perform the painting task with minimum human intervention is needed and will improve the quality of painting. The need for an autonomous painting robotics both clear and strong. Automated painting had been realized successfully in the automotive industry to paint millions of cars in the assembly lines. This industry uses spray painting and the robotic system is fixed in the assembly line. The domestic painting robots should be different in the sense that robots should have mobility so that it can move to paint the fixed walls. Also, the domestic painter robots should use roller instead of spray which is the common practice in the market to attain customer satisfaction.

The robot can move between workstations and deploys four stabilizing legs at each site. This robot cannot be used in residential buildings due to its 500 kg heavy weight. A reduction in

painting time of about 70% was reported for wall painting compared to human painting, and claimed to increase extra 20% if ceil painting is included. A scaled down model for interior wall painting using multi-color spray were implemented by Naticchiaetal. [4,5,6].

The design modifications are made for both the painting arm and the mobile base to satisfy the system requirement.

1.3 Objectives:

The main goal of this project is to design and build a robotic painting car that can be interfaced by Bluetooth protocol and can be controlled by an android based cell phone. Beside the relevant objectives to be achieved in this project are:

- To learn how to configure Bluetooth module with microcontroller.
- To learn how to develop a robotic car with gear motor by microcontroller programming.
- To learn how to use and configure the Bluetooth android app to control a robot or machine.

1.4 Report Organization

The report is organized as follows. Chapter I covers introduction, motivation, objective. Chapter II represents theory, description of the water mover, Bluetooth module, Water level sensor, Temperature Sensor, motor driver, dc geared motor, microcontroller ATmega328p, transformer, voltage regulator IC, diode, resistors and other parts. Chapter III describes and working principle of the project. In chapter IV we have discussed result and discussion, cost estimation. Future modification and conclusion are drawn in chapter V.

CHAPTER II

Review

2.1. Introduction

In this particular chapter we discuss the theoretical aspects of this projects. This is simple robotic device which is controlled by a microcontroller programming in AVR platform. Here we have used servo to move the painting brush and gear wheel motors to move the cars from one place to another. There is painting brush fixed with the servo-controlled motor to move the water dispenser up and down. This is the basic structure of this device. In the later part we will discuss the components we have used to develop this system.

2.2 Theory

This remote-controlled firefighting robot is not designed using complicated components. This robot is simple and portable. The robot is designed using few plastic frames, geared wheel, Bluetooth module, temperature sensor, water level sensor, servo motor and a controller unit to control the entire operation of the robot. This robot is compact because of high speed and pressure capabilities they have. They also have a very small weight to power output ratio and predictable performance *i.e.*, losses are minimum due to less number of moving parts and so gives expected performance. It has longer life, flexibility and it is efficient and dependable, and the installation is simple and the maintenance is also easy.

The design and implementation of this painting robot car can be sectionalized in two major ways.

A. **Hardware, will be consists of as follows:**
   1. Water Storage Dram
   2. Servo control water flow robotic arm
   3. 4 Wheel car
   4. Temperature sensor
   5. Gas sensor
   6. Bluetooth Module

B. **Software, will required as follows:**
   1. Microcontroller programing language Arduino for AVR
   2. Robo-Remo smart phone application with Android phone.
   3. V380 Pro for Wi-Fi camera viewing android apps.

2.2.1 Microcontroller Module (Arduino Uno R3)

The **Arduino Uno R3** is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, 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 Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Fig:2.2.1.1 ARDUINO UNO R3

Revision 3 of the board has the following new features:

a. 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.
b. Stronger RESET circuit.
c. 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	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) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Power

The Arduino Uno 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:

I. **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.

II. **5V.** the regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

III. **3V3.** A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

IV. **GND.** Ground pins.

Memory:

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output:

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead()functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition.

Some pins have specialized functions:

I. **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip

II. **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details

III. **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

IV. **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library

V. **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally

Some pins have specialized functionality:

I. **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

A. **AREF.** Reference voltage for the analog inputs. Used with analog Reference().

B. **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.2.2 Atmega328P Pin diagram :

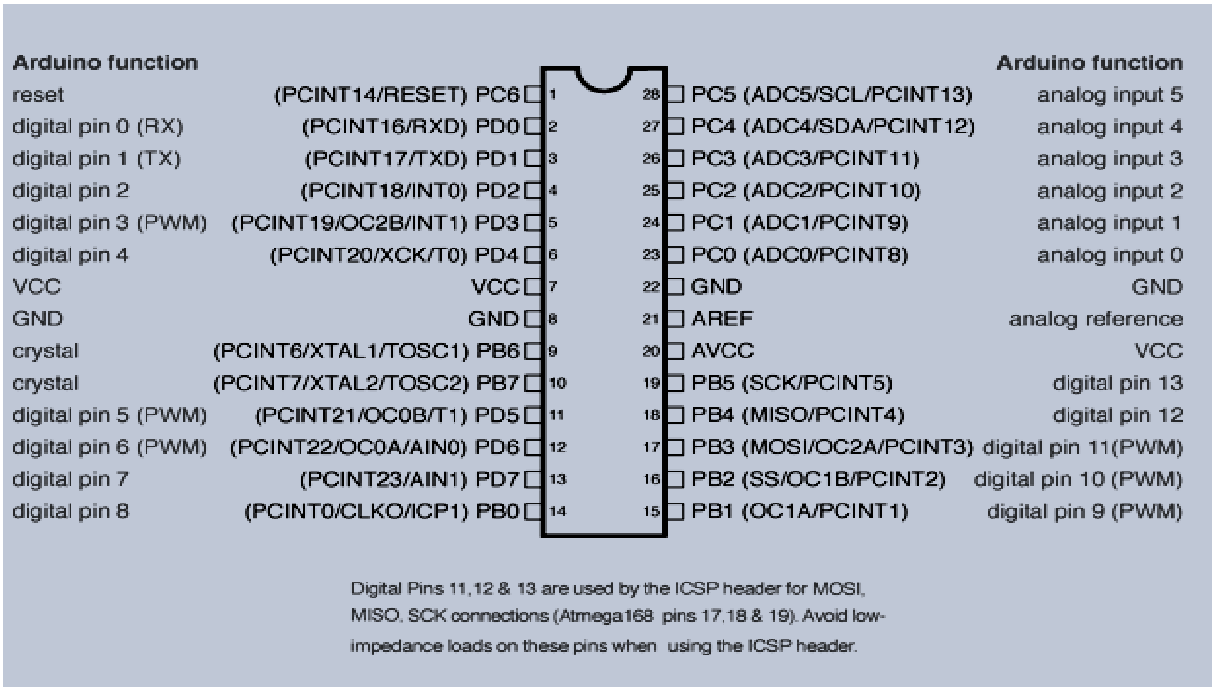

Arduino function
Arduino function
reset (PCINT14/RESET) PC6 1
28 PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX) (PCINT16/RXD) PD0 2
27 PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX) (PCINT17/TXD) PD1 3
26 PC3 (ADC3/PCINT11) analog input 3
digital pin 2 (PCINT18/INT0) PD2 4
25 PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM) (PCINT19/OC2B/INT1) PD3 5
24 PC1 (ADC1/PCINT9) analog input 1
digital pin 4 (PCINT20/XCK/T0) PD4 6
23 PC0 (ADC0/PCINT8) analog input 0
VCC VCC 7
22 GND GND
GND GND 8
21 AREF analog reference
crystal (PCINT6/XTAL1/TOSC1) PB6 9
20 AVCC VCC
crystal (PCINT7/XTAL2/TOSC2) PB7 10
19 PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM) (PCINT21/OC0B/T1) PD5 11
18 PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM) (PCINT22/OC0A/AIN0) PD6 12
17 PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7 (PCINT23/AIN1) PD7 13
16 PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8 (PCINT0/CLKO/ICP1) PB0 14
15 PB1 (OC1A/PCINT1) digital pin 9 (PWM)
Digital Pins 11,12 & 13 are used by the ICSP header for MOSI,
MISO, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-
impedance loads on these pins when using the ICSP header.

Fig: 2.2.2.1 ATmega328P Pin Mapping

Communication:

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .in file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line.

USB Overcurrent Protection:

The Arduino Uno has a resettable polypus that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics:

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mils (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Note:

The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

2.2.3 Bluetooth module (HC05)

Bluetooth transmission module, bottom board included, HC - 05 module can enable you to get rid of using cable for a serial port device and realize wireless serial communication within 10 meters. Using the module doesn't require knowing the complex Bluetooth bottom protocol. A few simple steps can get you to enjoy the convenience of wireless communication. Bluetooth transmission module has only 4 AT command. Respectively are communication testing, name changing, baud rate changing, matching password changing. AT command must be set from TXD, RXD signal pins, not via Bluetooth channel. Device sending the AT command can be various types of MCU (for example, 51, AVR, PIC, msp430, arm, etc.). It can also be a computer sending via serial port (PC serial port connected to MAX232 or USB serial port).

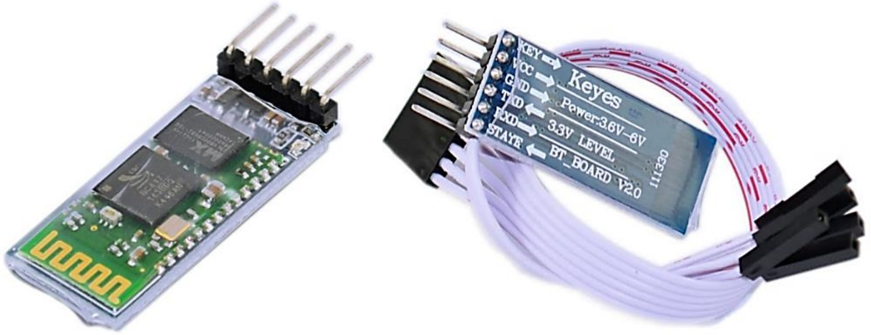
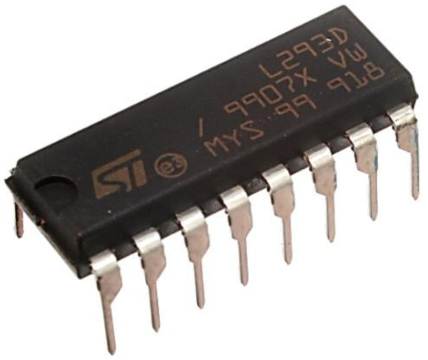


Fig: 2.2.3.1: HC05 Bluetooth module

2.2.4Motor Driver (L293D)

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.



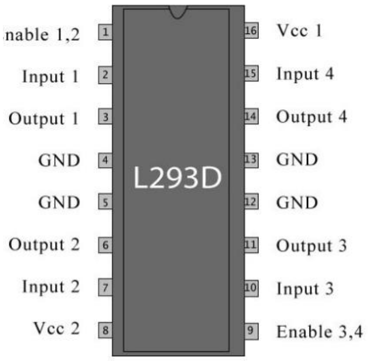


Fig: 2.2.4.1 L293D Motor Driver

Fig: 2.2.4.2 L293D Pin Diagram

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

2.2.5DC Geared Motor (wheel motor)

This is a DC motor and wheel set for making robots. These motors are light weight, high torque and low RPM. They can climb hills and have excellent traction, plus you can mount the wheel on either side of the motor with its double-sided output shaft

Fig: 2.2.5.1: DC Geared Motor with wheel

Specifications

- Motor Voltage: 3 - 12 V
- Motor Current: 70 mA (typical) - 250 mA (max)
- Speed: up to 170 RPM
- Torque: up to 0.8 Kg
- Gear Ration: 1:48
- Wheel Diameter: 66 mm
- Wheel Width: 27 mm

2.2.6 Mini Water Pump

This pump operates at 3V~9V 3W and has enough suction for most small projects. It's totally Submersible and Water-cooled. Use this pump to make a fountain or waterfall, even water your plants.

Fig: 2.2.6.1: Mini Water Pump

Specifications:

Power consumption	0.5W~3.5W
Rated voltage	9Vdc
Voltage used	4.5Vdc ~ 12Vdc
Max rated current	350mA
Max flow rate	4L/MIN(1.06G/MIN)
Max Static Head	3m(9.8ft)
Noise	<35db
Life span	More than 30000hrs

2.2.7 Servo Motor

Servo motors have been around for a long time and are utilized in many applications. They are small in size but pack a big punch and are very energy-efficient. These features allow them to be used to operate remote-controlled or radio-controlled toy cars, robots and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing, pharmaceutics and food services.

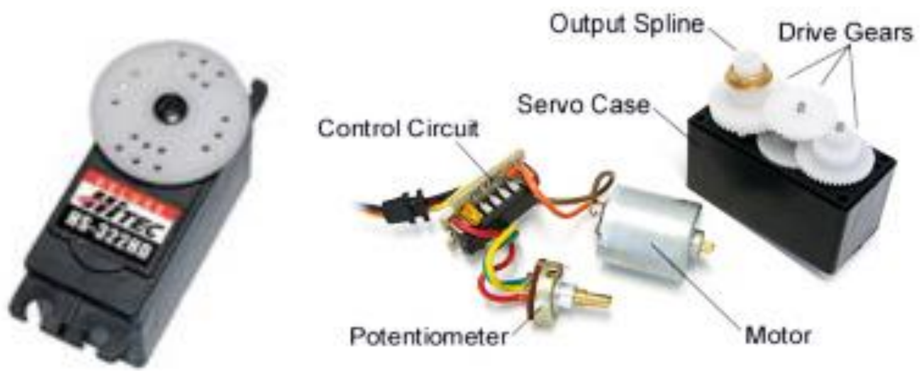


Fig: 2.2.7.1: Servo Motors

- How is the servo controlled?

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90 degrees in either direction for a total of 180-degree movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90-degree position. Shorter than 1.5ms moves it to 0 degrees, and any longer than 1.5ms will turn the servo to 180 degrees.

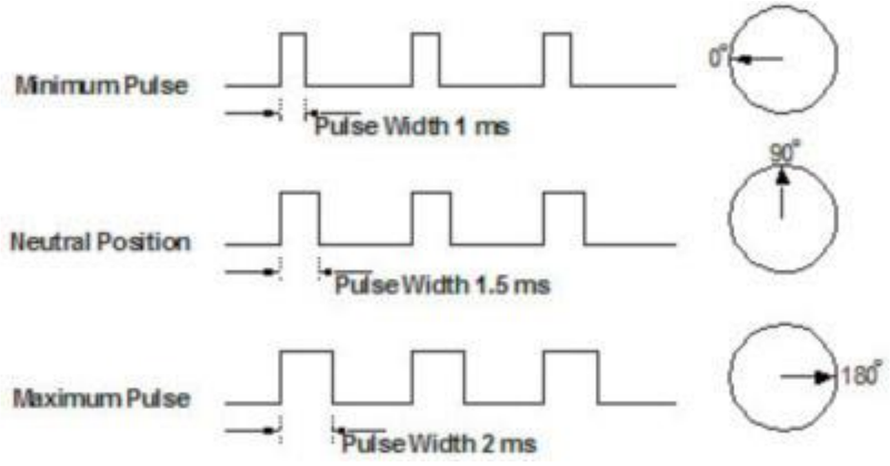


Fig: 2.2.7.2: PWM cycle of servo motor working.

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

2.2.8 12V DC Power supply Transformer (Rectifier)

A bridge rectifier circuit is a common part of the electronic power supplies. Many electronic circuits require rectified DC power supply for powering the various electronic basic components from available AC mains supply. We can find this rectifier in a wide variety of electronic AC power devices like home appliances, motor controllers, modulation process, welding applications, etc.

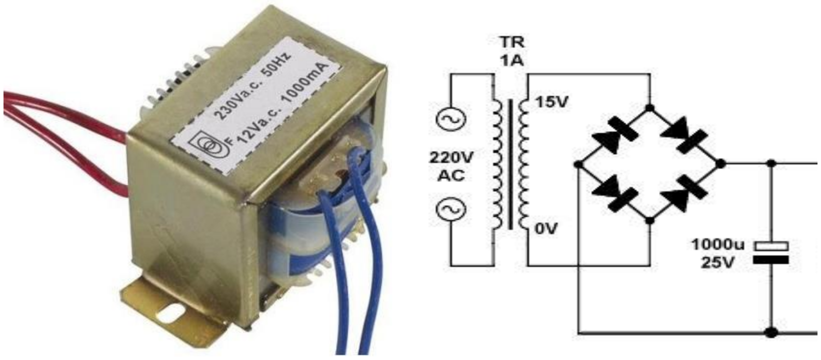


Fig: 2.2.8.1: 220/12V rectifier

The bridge rectifier circuit diagram consists of various stages of devices like transformer, Diode Bridge, filtering and regulators. The first stage of the circuit is a transformer which is a step-down type that changes the amplitude of the input voltage. Most of the electronic projects uses 230/12V transformer to step-down the AC mains 230V to 12V AC supply. Next stage is a diode-bridge rectifier which uses four or more diodes depending on the type of bridge rectifier. Choosing a particular diode or any other switching device for a corresponding rectifier needs some considerations of the device like Peak Inverse Voltage (PIV), forward current If, voltage ratings, etc. It is responsible for producing unidirectional or DC current at the load by conducting a set of diodes for every half cycle of the input signal.

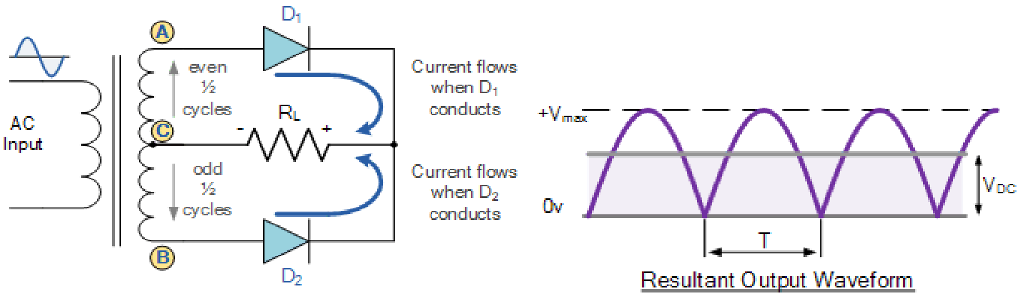


Fig: 2.2.8.2: Output Waveforms of 220V/12V rectifier

Since the output after the diode bridge rectifiers is of pulsating nature, and for producing it as a pure DC, filtering is necessary. Filtering is normally performed with one or more capacitors attached across the load, as you can observe in the below figure wherein smoothing of wave is performed. This capacitor rating also depends on the output voltage.

2.2.9 LM7805 (Voltage Regulator IC):

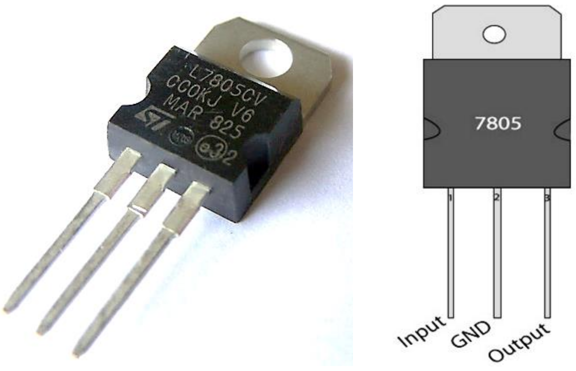


Fig: 2.2.9.1: 7805 (Voltage Regulator IC)

7805 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

Pin No	Function	Name
1	Input voltage (5V-18V)	Input
2	Ground (0V)	Ground
3	Regulated output; 5V (4.8V-5.2V)	Output

2.2.10 Crystal

A crystal or crystalline solid is a solid material whose constituent atoms, molecules, or ions are arranged in an ordered pattern extending in all three spatial dimensions. In addition to their microscopic structure, large crystals are usually identifiable by their macroscopic geometrical shape, consisting of flat faces with specific, characteristic orientations.

Fig: 2.2.10.1 Crystal

The scientific definition of a "crystal" is based on the microscopic arrangement of atoms inside it, called the crystal structure. A crystal is a solid where the atoms form a periodic arrangement.

2.2.11 Resistor

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.

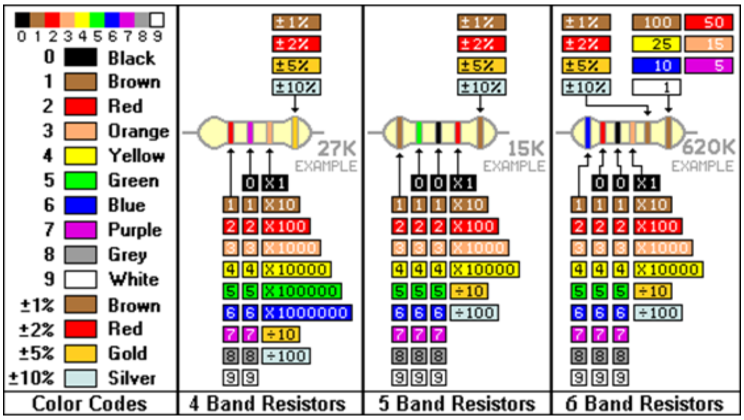


Fig: 2.2.11.1: Resistor Color Code

The current through a resistor is in direct proportion to the voltage across the resistor's terminals. This relationship is represented by Ohm's law:

$$V=IR$$

Where *I*is the current through the conductor in units of amperes, *V* is the potential difference measured across the conductor in units of volts, and *R* is the resistance of the conductor in units is (ohms).

The ratio of the voltage applied across a resistor's terminals to the intensity of current in the circuit is called its resistance, and this can be assumed to be a constant (independent of the voltage) for ordinary resistors working within their ratings.

Resistor are widely used in two ways-

Series resistor- Rt = R1+R2+R3+……+Rn

Parallel resistor,

$$Rtotal = \frac{1}{\frac{1}{R1} + \frac{1}{R2} + \cdots + \frac{1}{Rn}}$$

2.2.12 Diode (IN 4007)

In electronics, a diode is a two-terminal electronic component with asymmetric conductance, it has low (ideally zero) resistance to current flow in one direction, and high (ideally infinite) resistance in the other. A semiconductor diode, the most common type today, is a crystalline piece of semiconductor material with a p–n junction connected to two electrical terminals. A vacuum tube diode has two electrodes, a plate (anode) and a heated.

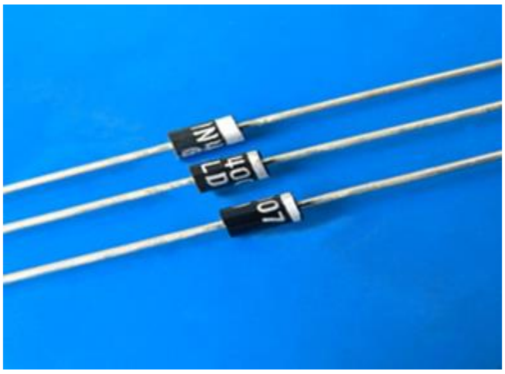


Fig: 2.2.12.1: Diode (IN 4007)

The most common function of a diode is to allow an electric current to pass in one direction (called the diode's forward direction), while blocking current in the opposite direction (the reverse direction). Thus, the diode can be viewed as an electronic version of a check valve. This unidirectional behavior is called rectification, and is used to convert alternating current to direct current, including extraction of modulation from radio signals in radio receivers—these diodes are forms of rectifiers.

We used IN 4007 which is a simple, very common rectifier diode. Often used for reverse voltage protection, the 1N4007 is a staple for many powers, DC to DC step up, and breadboard projects. 1N4007 is rated for up to 1A/1000V.

2.2.13 Variable Resistor/Potentiometer:

A potentiometer /variable resistor is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

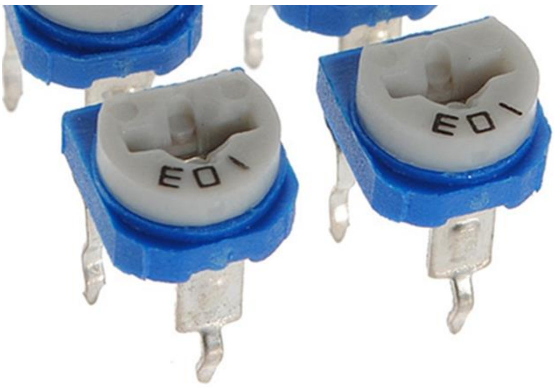


Fig:2.2.13.1: Variable Resistor/Potentiometer

The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Many inexpensive potentiometers are constructed with a resistive element formed into an arc of a circle usually a little less than a full turn and a wiper sliding on this element when rotated, making electrical contact. The resistive element, with a terminal at each end, is flat or angled. The wiper is connected to a third terminal, usually between the other two types

Fig: 2.2.13.2: Different Types of Potentiometer

Potentiometers are categorized by,

Linear taper potentiometer: A linear taper potentiometer (linear describes the electrical characteristic of the device, not the geometry of the resistive element) has a resistive element of constant cross-section, resulting in a device where the resistance between the contact (wiper) and one end terminal is proportional to the distance between them. Linear taper potentiometers [3] are used when the division ratio of the potentiometer must be proportional to the angle of shaft rotation (or slider position), for example, controls used for adjusting the centering of the display on an analog cathode-ray oscilloscope. Precision potentiometers have an accurate relationship between resistance and slider position.

2.2.14 Vero Board

Strip Board is a widely used type of electronics prototyping board characterized by a 0.1-inch (2.54 mm) regular (rectangular) grid of holes, with wide parallel strips of copper cladding running in one direction all the way across one side of the board. It is commonly known by the name Vero Board. With care, it is possible to break between holes to allow for components that have two pin rows only one position apart such as twin row headers.

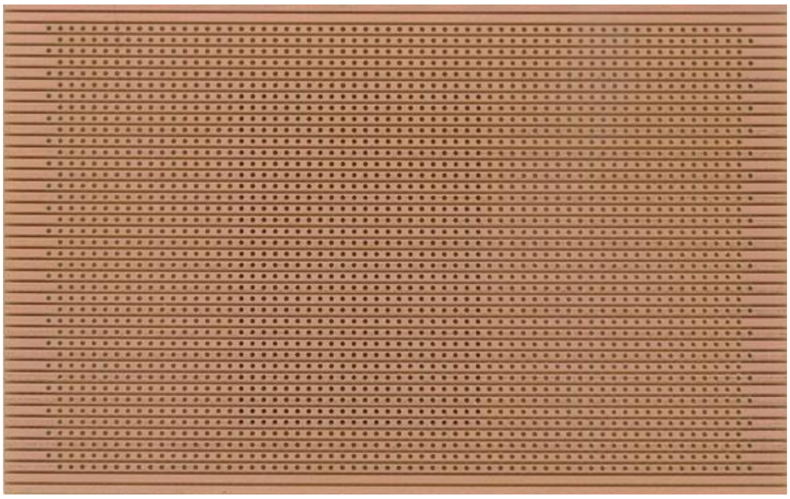


Fig. 2.2.14.1: Vero Board

2.2.15 Connecting Wire:

A wire is a single, usually cylindrical, flexible strand. Wires are used to bear carry electricity and telecommunications signals. Standard sizes are determined by various wire gauges. The term wire is also used more loosely to refer to a bundle of such strands, as in 'multi stranded wire', which is more correctly termed a wire rope in mechanics, or a cable in electricity.

Fig: 2.2.15.1: Connecting Wire

Although usually circular in cross-section, wire can be made in square, hexagonal, flattened rectangular or other cross-sections, either for decorative purposes, or for technical purposes such as high-efficiency voice coils in loudspeakers. Edge-wound coil springs, such as the Slinky toy, are made of special flattened wire.

2.2.16 LED Indicator

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a p–n junction diode, which emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.

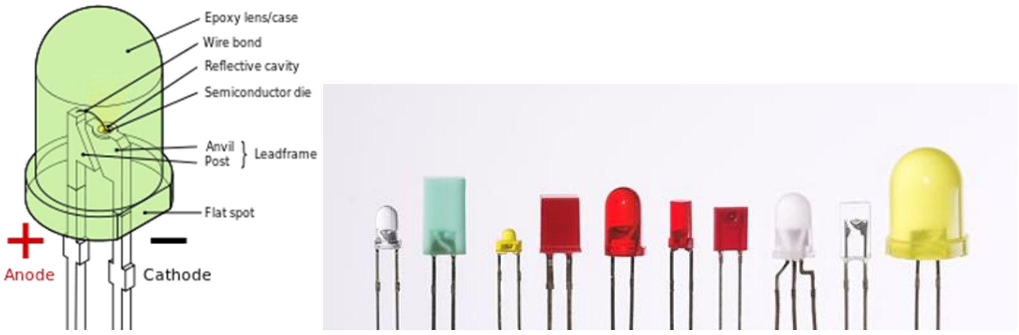


Fig: 2.2.16.1: Indicator LED

In this chapter we have briefly described the description of the robotic arm, BC05 Bluetooth module, shift resister (74HC595), motor driver (LD293), dc geared motor (car wheel), microcontroller IC ATmega328p, transformer (220/12V), voltage regulator IC 7805, diode (IN4007), resistors and other parts. And especially we will widely know about the main component of this project Microcontroller IC.

Chapter III

DESING & FABRICATION

3.1 Introduction

In this chapter fully cover with discuss design and fabrication of this project. Here we will discuss about developed block diagram and briefly describe about the circuit description and also learn about working principle. Total project flow chart is also available in this chapter.

3.2 Block Diagram

A smart phone Android operated robot. Here we have controlled our robot car using Bluetooth module HC-05 and ATmega328P microcontroller with your android Smartphone device. The controlling devices of the whole system is the microcontroller. Another very important component is servo motor which is a very effective device to control very fast and precisely. We have used this servo as the painting hand to hold the painting brush. The L293 motor driver is used to gear up the wheels of the robot car. There is a battery bank storage to power up the car and a rectifier circuit to charge if required.

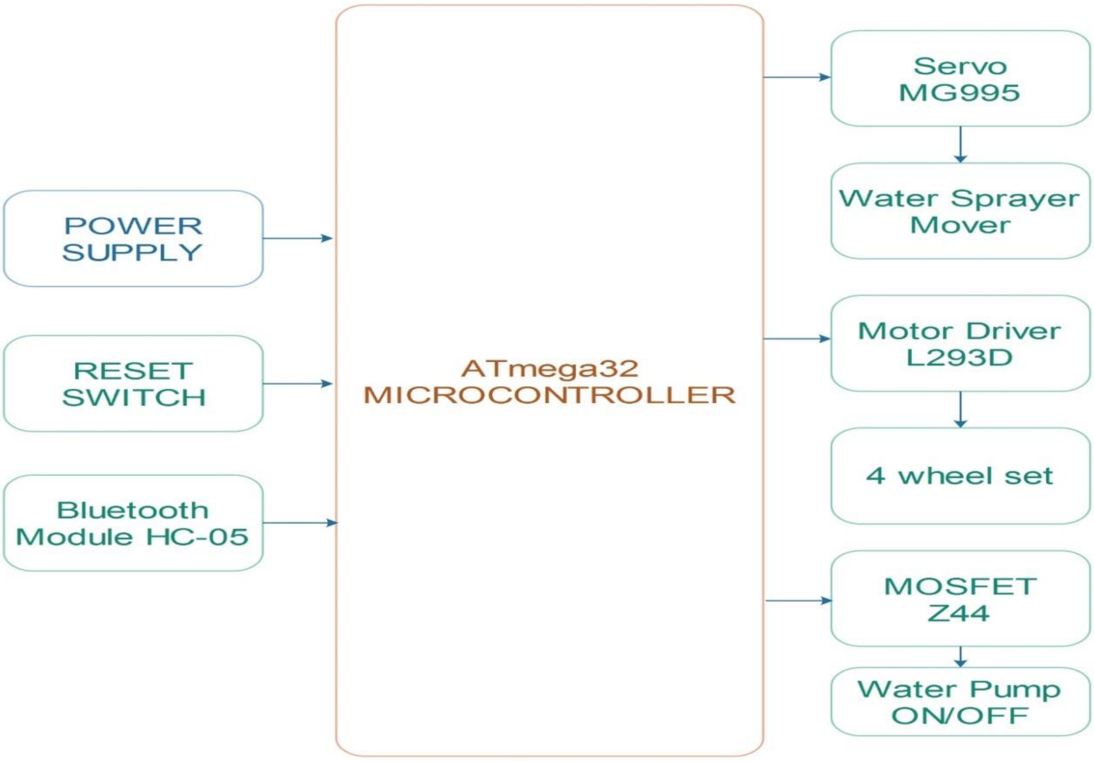


Fig: 3.2.1: Block Diagram of the project

3.3 Circuit Diagram & Description

The main controlling part is the microcontroller ATmega328p IC. Pin 6,7,8,9 this four are connected with the motor driver ic as it's input signal. This IC is used to rotate the wheel (4nos.) of the car which are controlled by two L293D driver IC. The TxD (0,1no. pin) is connected with the Bluetooth module to be connected with the cell phone. Pin 2 of ATmega328p is connected to the servo motor which is designed to rotate 0° to 45° in both directions. Pin A0 is connected with a temperature sensor LM35. Pin A1 is connected with a water level sensor to read the water level data inside of water tank.The circuit- a L293D dual H-Bridge motor driver, The H-Bridges for bidirectional driving of the motors. The output of the microcontroller is not sufficient to drive a motor; the H-Bridge is able to give max 2A peak at its both output. The Two enable pins are used to input the PWM pulse and control the speed of the motor. The other 2 pins are used for direction control of the motor.

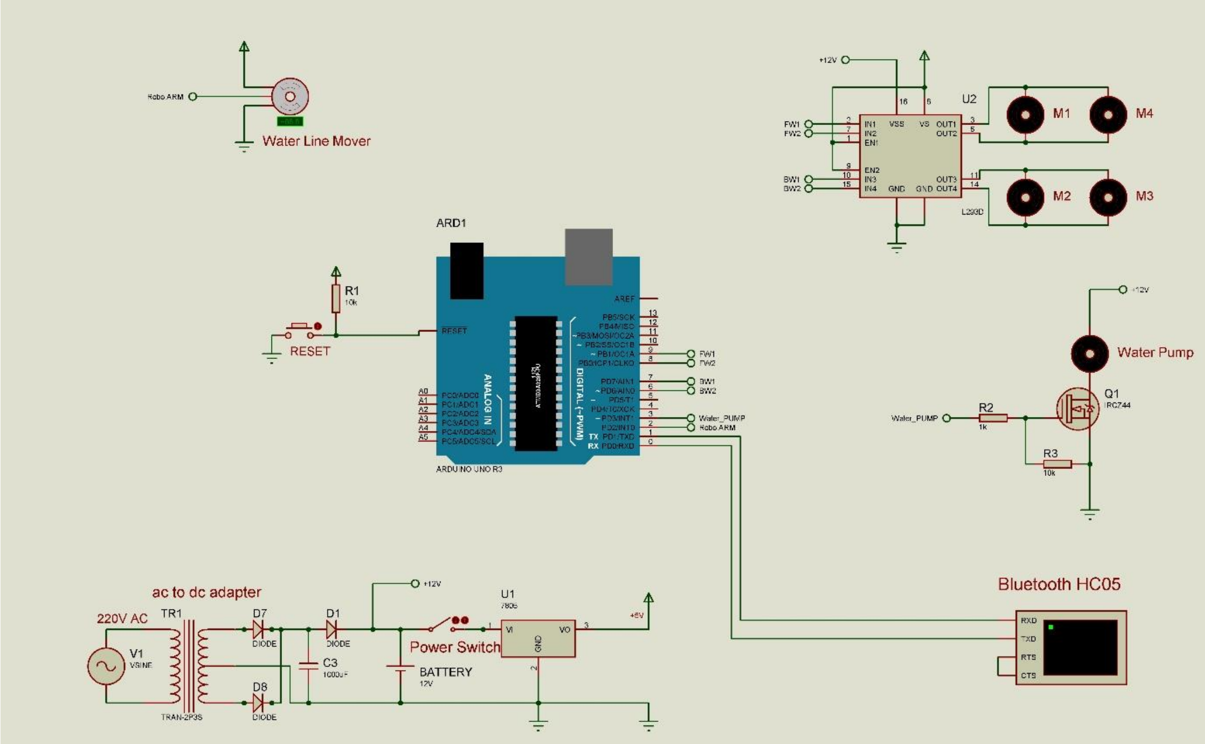


Fig: 3.3.1: Circuit Diagram of the Project

3.4 Motor gear step picture outlook and casing

The chassis is mainly divided into two parts with two independent motors. We can name the two parts as front side and back side. The front motor has a gearbox with a spring powered retaining mechanism and it controls steering and angular movement of the front wheels. The spring retaining mechanism moves the wheels back to its original position (toward the front direction when the motor is underpowered or turned off. The back motor also has a gearbox which reduces the rpm of the back motor and it increases torque of the back motor.

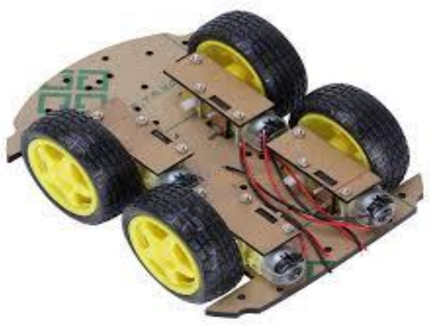


Fig: 3.4.1: Construction of the chassis

There is three rechargeable lead acid sealed cells are attached on the chassis. Each cell gives an output voltage equal to 4Vwhen it is charged. The 3 cells are attached in series to get an output voltage of 12V which is directly used by the motor driver IC to control the motors at that voltage.

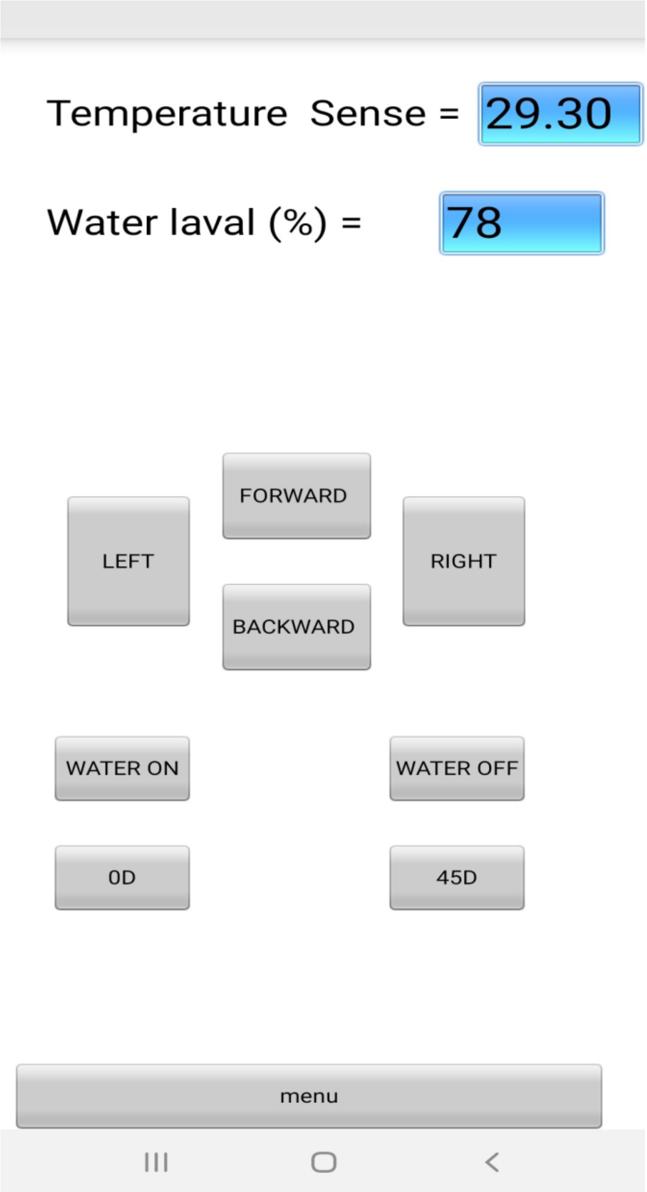
3.5 Working Principle

The system is designed to control with an android operated cell phone. We use the Roboremo android based software to interface between cell phone and Bluetooth module. This enables us to control the car easily by any android cell phone. Click on “SELECT DEVICE” icon to select paired Bluetooth module. When press “up arrow” it sends the data “A” to Bluetooth module connected with the circuit. When microcontroller detects “A” the robot car moves FORWORD. When press “DOWN ARROW” it sends the data “B” to Bluetooth module connected with the circuit. When microcontroller detects “B” the robot car moves REVERSE. When press “LEFT ARROW” it sends the data “C” to Bluetooth module connected with the circuit. When microcontroller defects “C” the robot car turns LEFT.

When press “RIGHT ARROW” it sends the data “D” to Bluetooth module connected with the circuit. When microcontroller defects “D” the robot car turns RIGHT.

The software also has some button to control the arm. By pressing the “ARM UP” and “ARM DOWN” it sends the signal “0” and signal “1” to Bluetooth module connected with the circuit. When microcontroller detects “0” and “1” the robot arm moves up and down respectively up 45°.

It has two tow sensor detection windows for show the temperature reading value and water level parcen cane value.



Temperature Sense = 29.30

Water laval (%) = 78

Fig: 3.5.1 “RoboRemo” Apps screen shot

When we press “ARM OUT” and “ARM IN” it sends the data “T” and “U” to the Bluetooth module to pass the signal into microcontroller. As soon as microcontroller received signal “T” or “U” then the arm goes “OPEN” and “CLOSE” respectively.

3.6 Image of the Project:

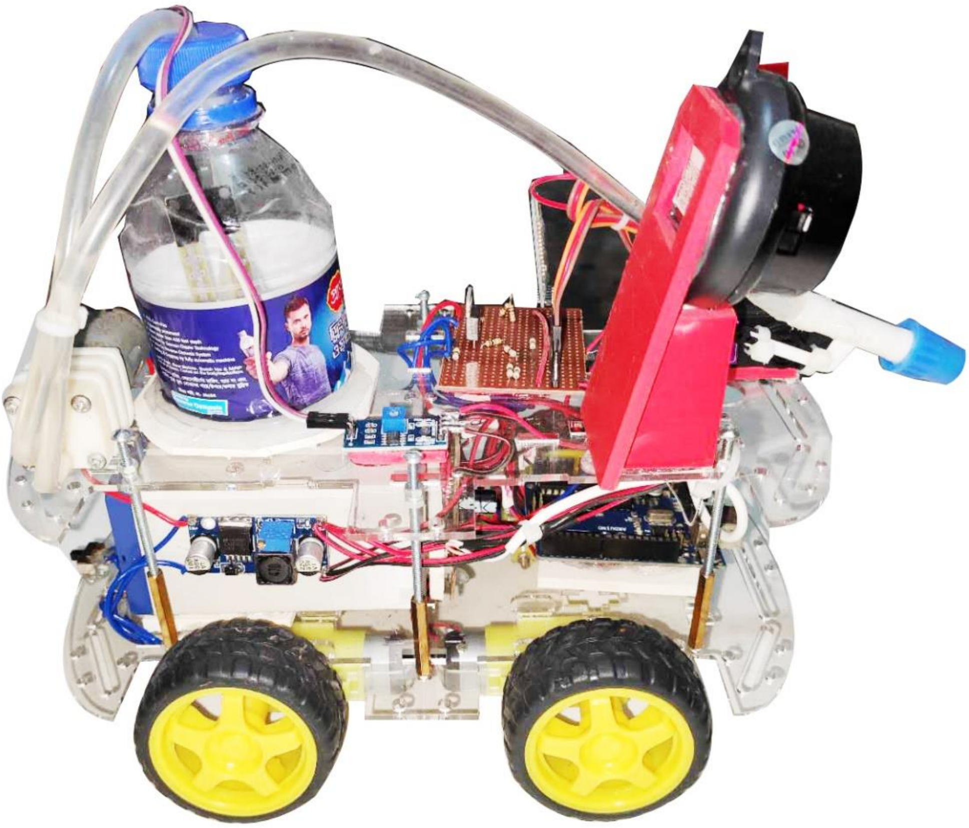


Fig. 3.6.1: Image of the firefighting robotic car

3.7 Flow Chart:

The objective of the paper is to realize the control system using Bluetooth Technology. Robot and smartphones are a perfect match. As mobile phones are widely used device, so our main objective is to use a smart phone as a controller unit. The HC05 Bluetooth module provides the facility of wireless connectivity between mobile phone and microcontroller unit. Rather by developing this system we can reduces human efforts.

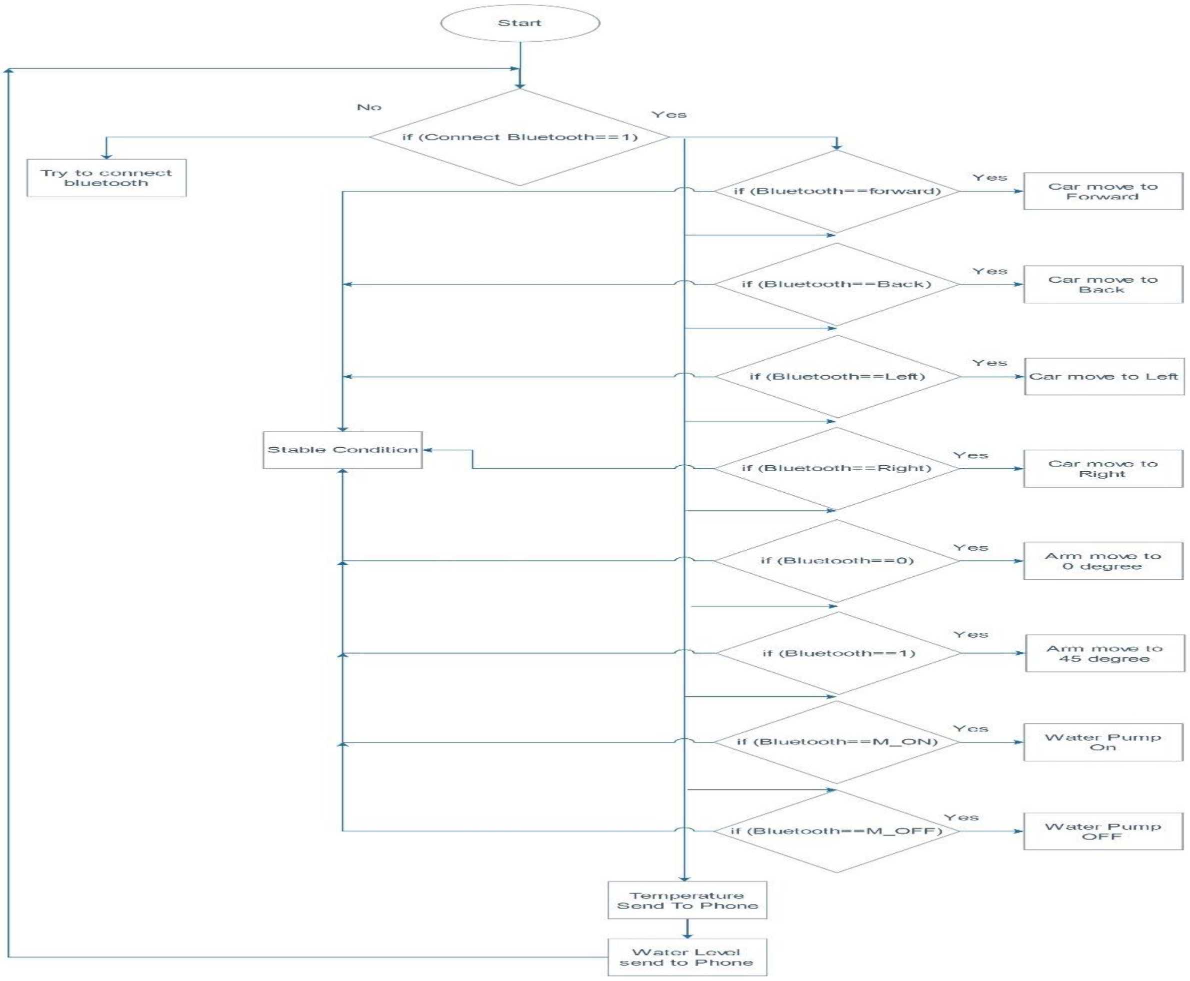


Fig: 3.7.1: Working Flow Chart of the Project

CHAPTER IV

Data collection

4.1 Mechanical Efficiency:

To determine the mechanical/hydraulic efficiency of a pump, the actual drive torque required and its theoretical torque are compared. Mechanical/hydraulic efficiency of 100% means the hydraulic pump needs no torque to provide flow at zero pressure. In reality, mechanical and fluid frictions prevent this from happening.

***ηm* = (*Actual power delivered to pump / Pump output power assuming no leakage*)× 100%**

4.2 Volumetric flow Rate:

The volumetric efficiency of a pump is determined by the ratio of the actual flow delivered by it to its theoretical flow at a given pressure. Pump displacement per revolution and driven speed are multiplied to determine the theoretical flow

***ηv* =*Theoretical flow rate / Actual flow rate by the pump*×100%.**

4.3 Parabolic Distance: We have got the parabolic distance (1.38m) of this demo.

4.4 Storage Water Discharge Time: To complete the water required time is 93 sec.

CHAPTER V

RESULTS

5.1 Introduction

This chapter contains the results obtained and discussion about the project. We have also covered discussions about advantages, disadvantages and limitation of current version of the protection system.

5.2Result

This project covers only implementation of a remotely operated car from an android mobile phone by Bluetooth communication method. This project also covers design a microcontroller circuit with peripherals. Bluetooth communication has become popular on various types of mobile communication between them in a short range. The laptop or desktop computer can be connected with the HC-05 Bluetooth module. The Bluetooth module is connected with the microcontroller. The robot is made activated after the switch on the activity board is turned on the microcontroller gets its power from the 7805 5V voltage regulator which gets power from the rechargeable battery. The microcontroller gets its command from the Bluetooth module and controls the motor driver with its I/O port.

We have tested the approaching of the robotic car, after connecting with the Bluetooth module and RoboRemo android apps, we were able to move the car in forward, backward, left side and right side direction by just pressing the “forward”, “backward”, “left” and “right” arrow of the apps. And found the car was moving according to the desired design and command within the range of the Bluetooth coverage. Also, we were able to move robot brush and pumping the liquefied paint as per our desired area.

5.3 Advantages

- In this particular project we used very simple design thus required less components.
- We have also incorporated Bluetooth remote control design which can facilitate the remote controlling by a smart phone app. This will be one of the major incorporations that anyone can easily use this having no advance knowledge even in residence.
- Will be used self-developed programming so that customized solution will be able provide as per demand.
- Since this will our locally developed device the production cost will be lower than that of imported one with better and reliable after sales service.

5.4. Limitation:

- Communication mostly used similar frequency so interference occurs. If address is not specifying.
- It cannot work beyond limit(100M)
- Tough it can pass through solids, solids like wall, roof can decrease the range.

5.5Cost Estimation

#	Particulars	Specification	Qty.	Unit Price	Total Price
1	Microcontroller IC	ATmega328p	1no.	200	200
2	Crystal	8MHz	1no.	10	10
3	IC Base	28pin	1no.	10	10
4	Robotic arm gripper	5V, DC	1no.	500	500
5	Bluetooth Module	BC05	1no.	300	300
6	Motor Shield & Driver	L293D	3nos.	200	200
7	Temperature Sensor	LM-35	1no.	100	100
8	Dc Geared Motor	12V, DC (car wheel)	4nos.	60	240
9	Servo Motor	12V, DC	1no.	500	500
10	Robot gripper holder	Steel/Aluminum	1ser	200	200
11	Robot Car Structure	PVC structure	1no.	500	500
12	Transformer	220/12V, 2amps	1no.	350	350
13	Voltage Regulator	LM 7805	2 nos.	10	20
14	Diode	IN4007	2 nos.	5	30
15	Capacitor	1000μf, 25V	1no.	10	10
16	Vero Board	As per design	1no.	30	30
17	Resistor	Various	20	1	20
18	Soldering cost				30
19	Others				30
				Total	**3,280/-**

5.6 Applications:

- Residential or Commercial Building construction:
  This machine is perfect to paint the wall of any construction site, this can reduce human effort and saves a lot of time and money.
- Vehicle, Aircraft or any other vehicle Painting factory:
  Vehicle and Aircraft painting works is a very precise job to do as well as hazardous for human; in this particular job our painting robot can be the best solution.
- Plastic and Metal industries:

These two sectors are very emerging sector for our country where painting and finishing is a very important steps of production so this painting robot is a very good solution for painting and finishing.

- Washing Tools:

  Instead of paint if we use water in the reservoir tank this could be very good washing tools (i.e. car wash, building glass wash etc.) to use.

APPENDIX

- **What is Bluetooth?**

Bluetooth is a **standardized protocol** for sending and receiving data via a 2.4GHz wireless link. It's a secure protocol, and it's perfect for short-range, low-power, low-cost, wireless transmissions between electronic devices.

These days it feels like *everything* is wireless, and Bluetooth is a big part of that wireless revolution. You'll find Bluetooth embedded into a great variety of consumer products, like headsets, video game controllers, or (of course) livestock trackers.

In our world of embedded electronics hacker, Bluetooth serves as an excellent protocol for wirelessly transmitting relatively small amounts of data over a short range (<100m). It's perfectly suited as a wireless replacement for serial communication interfaces. Or you can use it to create a DIY HID Computer Keyboard. Or, with the right module, it can be used to build a homebrew, wireless MP3-playing speaker.

This tutorial aims to provide a quick overview of the Bluetooth protocol. We'll examine the specifications and profiles that form its foundation, and we'll go over how Bluetooth compares to other wireless protocols.

- **How Bluetooth Works?**

The Bluetooth protocol operates at 2.4GHz in the same unlicensed ISM frequency band where RF protocols like ZigBee and Wi-Fi also exist. There is a standardized set of rules and specifications that differentiates it from other protocols. If you have a few hours to kill and want to learn every nook and cranny of Bluetooth, check out the published specifications, otherwise here's a quick overview of what makes Bluetooth special.

- **Masters, Slaves, and Pico nets**

Bluetooth networks (commonly referred to as **Pico nets**) use a master/slave model to control when and where devices can send data. In this model, a single master device can be connected to up to seven different slave devices. Any slave device in the Pico net can only be connected to a single master.

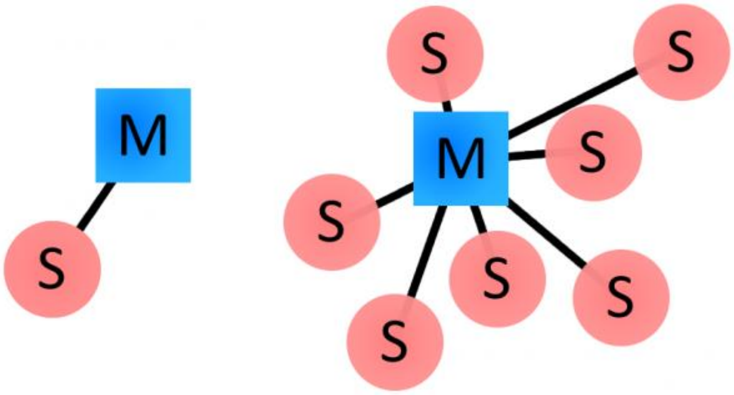


Fig: Examples of Bluetooth master/slave Pico net topologies.

The master coordinates communication throughout the Piconet. It can send data to any of its slaves and request data from them as well. Slaves are only allowed to transmit to and receive from their master. They can’t talk to other slaves in the Piconet.

- **Bluetooth Module HC05**

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 Blue core 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.

- **Specifications**

➢ **Hardware Features**

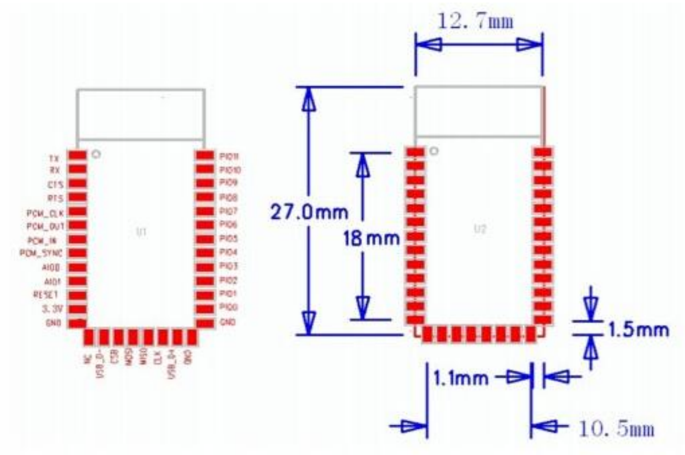
- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

➢ **Software Features**

- Default Baud rate: 38400, Data bits:8, Stop bit:1, Parity: No parity, Data control: has.

- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:”0000” as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	PAD Type	Description
GND	13,21,22	VSS	Ground Pot
3.3 VCC	12	3.3V	Integrated 3.3V(+) supply with On-chip linear regulator output within 3.15-3.3V
AIO0	9	Bi-directional	Programmable input/output line
AIO1	10	Bi-directional	Programmable input/output line
AIO0	23	Bi-directional RX EN	Programmable input/output line, control output for LNA (if fitted)
AIO1	24	Bi-directional TX EN	Programmable input/output line, control output for PA (if fitted)
PIO2	25	Bi-directional	Programmable input/output line
PIO3	26	Bi-directional	Programmable input/output line

PIO4	27	Bi-directional	Programmable input/output line
PIO5	28	Bi-directional	Programmable input/output line
PIO6	29	Bi-directional	Programmable input/output line
PIO7	30	Bi-directional	Programmable input/output line
PIO8	31	Bi-directional	Programmable input/output line
PIO9	32	Bi-directional	Programmable input/output line
PIO10	33	Bi-directional	Programmable input/output line
PIO11	34	Bi-directional	Programmable input/output line
RESETB	11	CMOS input with weak internal pull-up	Reset of low Inputdenounced so must be low for >5MS to cause a reset
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low
UART_RX	2	CMOS input with weak internal pull-down	UART Data input
UART_TX	1	CMOS output, tri-stable with weak internal pull-up	UART Data output
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input
SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low
SPI_CLK	19	CMOS input with weak internal pull-	Serial peripheral interface clock

		down	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data output
USB_-	15	Bi-directional	
USB_+	20	Bi-directional	
NC	14		
PCM_CLK	5	Bi-directional	Synchronous PCM data clock
PCM_OUT	6	CMOS output	Synchronous PCM data output
PCM_IN	7	CMOS input	Synchronous PCM data input
PCM_SYNC	8	Bi-directional	Synchronous PCM data strobe

- **AT Command Default:**

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<PARAM> OK	PARAM : firmware version

Example:

AT+VERSION? \r\n

+VERSION:2.0-20100601 OK

4. Restore default

Command	Respond	Parameter

AT+ORGL	OK	-

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01, Baud 38400bits/s.

5. Get module address

Command	Respond	Parameter
AT+ADDR?	+ADDR:<PARAM> OK	PARAM: address of Bluetooth module

Bluetooth address: NAP: UAP: LAP

Example:

AT+ADDR? \r\n

+ADDR: 1234:56: abcdef

OK

6. Set/Check module name:

Command	Respond	Parameter
AT+NAME=<PARAM>	OK	PARAM: Bluetooth module name (Default :HC-05)
AT+NAME?	+NAME:<PARAM> OK (/FAIL)	PARAM: Bluetooth module name (Default :HC-05)

Example:

AT+NAME=HC-05\r\n set the module name to HC-05

OK

AT+NAME=ITeadStudio\r\n

OK

AT+NAME? \r\n

+NAME: ITeadStudio

OK

7. Get the Bluetooth device name:

Command	Respond	Parameter
AT+RNAME?<Param1>	1. +NAME:<Param2> OK 2. FAIL	Param1,Param 2 : the address of Bluetooth device

Example: (Device address 00:02:72:od:22:24， name：ITead)

AT+RNAME? 0002， 72， od2224\r\n

+RNAME:ITead

OK

8. Set/Check module mode:

Command	Respond	Parameter
AT+ROLE=<PARAM>	OK	PARAM: 0- Slave 1-Master 2-Slave-Loop
AT+ ROLE?	+ROLE:<PARAM> OK	PARAM: 0- Slave 1-Master 2-Slave-Loop

9. Set/Check device class

Command	Respond	Parameter
AT+CLASS=<PARAM>	OK	PARAM: Device Class
AT+ CLASS?	1. +CLASS:<PARAM> OK 2. FAIL	PARAM: Device Class

10. Set/Check GIAC (General Inquire Access Code)

Command	Respond	Parameter
AT+IAC=<PARAM>	1.OK 2. FAIL	PARAM: GIAC (Default : 9e8b33)
AT+IAC	+IAC:<PARAM> OK	PARAM: GIAC (Default : 9e8b33)

Example:

AT+IAC=9e8b3f\r\n

OK

AT+IAC? \r\n

+IAC: 9e8b3f

OK

11. Set/Check -- Query access patterns

Command	Respond	Parameter
AT+INQM=<PARAM>,<Param2>,<Param3>	1.OK 2. FAIL	PARAM: 0—— inquiry_mode_standard 1—— inquiry_mode_rssi Param2: Maximum number of Bluetooth devices to respond to Param3: Timeout (1-48 : 1.28s to 61.44s)
AT+ INQM?	+INQM ：<PARAM>,<Param2>,<Param3> OK	PARAM: 0—— inquiry_mode_standard 1—— inquiry_mode_rssi Param2: Maximum number of Bluetooth devices to respond to Param3: Timeout (1-48 : 1.28s to 61.44s)

Example:

AT+INQM=1,9,48\r\n
OK
AT+INQM\r\n
+INQM:1, 9, 48
OK

12. Set/Check PIN code:

Command	Respond	Parameter
AT+PSWD=<Param>	OK	Param: PIN code (Default 1234)
AT+ PSWD?	+ PSWD ：<Param> OK	Param: PIN code (Default 1234)

13. Set/Check serial parameter:

Command	Respond	Parameter
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: Baud Param2: Stop bit Param3: Parity
AT+UART?	+UART=<Param>,<Param2>, <Param3> OK	Param1: Baud Param2: Stop bit Param3: Parity

Example:

AT+UART=115200， 1,2,\r\n

OK

AT+UART?

+UART:115200,1,2

OK

14. Set/Check connect mode:

Command	Respond	Parameter
AT+CMODE=<Param>	OK	Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop
AT+ CMODE?	+ CMODE:<Param> OK	Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop

15. Set/Check fixed address:

Command	Respond	Parameter
AT+BIND=<Param>	OK	Param: Fixed address (Default 00:00:00:00:00:00)
AT+ BIND?	+ BIND:<Param> OK	Param: Fixed address (Default 00:00:00:00:00:00)

Example:

AT+BIND=1234， 56， abcdef\r\n

OK

AT+BIND?\r\n

+BIND:1234:56:abcdef

OK

16. Set/Check LED I/O

Command	Respond	Parameter

AT+POLAR=<Param1,<Param2>	OK	Param1: 0- PIO8 low drive LED 1- PIO8 high drive LED Param2: 0- PIO9 low drive LED 1- PIO9 high drive LED
AT+ POLAR?	+ POLAR=<Param1>,<Param2> OK	Param1: 0- PIO8 low drive LED 1- PIO8 high drive LED Param2: 0- PIO9 low drive LED 1- PIO9 high drive LED

17. Set PIO output

Command	Respond	Parameter
AT+PIO=<Param1>,<Param2>	OK	Param1: PIO number Param2: PIO level 0- low 1- high

Example:

1. PIO10 output high level

AT+PI0=10，1\r\n

OK

18. Set/Check – scan parameter

Command	Respond	Parameter
AT+IPSCAN=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Query time interval Param2 ： Query duration Param3 ： Paging interval Param4 ： Call duration
AT+IPSCAN?	+IPSCAN:<Param1>,<Param2>,<Param3>,<Param4> OK	Param1: Query time interval Param2 ： Query duration Param3 ： Paging interval Param4 ： Call

		duration

Example:

AT+IPSCAN =1234,500,1200,250\r\n

OK

AT+IPSCAN?

+IPSCAN:1234,500,1200,250

19. Set/Check – SHIFF parameter

Command	Respond	Parameter
AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4>	OK	Param1: Max time Param2: Min time Param3: Retry time Param4: Time out
AT+ SNIFF?	+SNIFF:<Param1>,<Param2>,<Param3>,<Param4> OK	Param1: Max time Param2: Min time Param3: Retry time Param4: Time out

20. Set/Check security mode

Command	Respond	Parameter
AT+SENM=<Param1>,<Param2>	1. OK 2. FAIL	Param1: 0——sec_mode0+off 1——sec_mode1+ non_secure 2——sec_mode2_ service 3——sec_mode3_l ink 4——sec_mode_u nknown Param2: 0——hci_enc_mod e_off 1——hci_enc_mod e_pt_to_pt 2——hci_enc_mod e_pt_to_pt_and_bcast
AT+ SENM?	+ SENM:<Param1>,<Param2> OK	Param1: 0——sec_mode0+off 1——sec_mode1+ non_secure

		2——sec_mode2_ service 3——sec_mode3_l ink 4——sec_mode_u nknown Param2: 0——hci_enc_mod e_off 1——hci_enc_mod e_pt_to_pt 2——hci_enc_mod e_pt_to_pt_and_b cast

21. Delete Authenticated Device

Command	Respond	Parameter
AT+PMSAD=<Param>	OK	Param: Authenticated Device Address

Example:

AT+PMSAD =1234,56,abcdef\r\n

OK

22. Delete All Authenticated Device

Command	Respond	Parameter
AT+ RMAAD	OK	-

23. Search Authenticated Device

Command	Respond	Parameter
AT+FSAD=<Param>	1. OK 2. FAIL	Param: Device address

24. Get Authenticated Device Count

Command	Respond	Parameter
AT+ADCN?	+ADCN：<Param> OK	Param: Device Count

25. Most Recently Used Authenticated Device

Command	Respond	Parameter
AT+MRAD?	+ MRAD：<Param> OK	Param: Recently Authenticated Device Address

26. Get the module working state

Command	Respond	Parameter
AT+ STATE?	+ STATE：<Param> OK	Param: "INITIALIZED" "READY" "PAIRABLE" "PAIRED" "INQUIRING" "CONNECTING" "CONNECTED""DISCONNECTED" "NUKNOW"

27. Initialize the SPP profile lib

Command	Respond	Parameter
AT+INIT	1. OK 2. FAIL	-

28. Inquiry Bluetooth Device

Command	Respond	Parameter
AT+INQ	+INQ: <Param1>, <Param2>, <Param3> …. OK	Param1：Address Param2 ：Device Class Param3 ：RSSI Signal strength

Example:

```
AT+INIT\r\n
OK
AT+IAC=9e8b33\r\n
OK
AT+CLASS=0\r\n
AT+INQM=1,9,48\r\n
At+INQ\r\n
+INQ:2:72:D2224,3E0104,FFBC
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC1
+INQ:2:72:D2224,3F0104,FFAD
+INQ:1234:56:0,1F1F,FFBE
```

```
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFBE
+INQ:2:72:D2224,3F0104,FFBC
OK
```

29. Cancel Inquiring Bluetooth Device

Command	Respond	Parameter
AT+ INQC	OK	-

30. Equipment Matching

Command	Respond	Parameter
AT+PAIR=<Param1>,<Param2>	1. OK 2. FAIL	Param1 : Device Address Param2 : Time out

31. Connect Device

Command	Respond	Parameter
AT+LINK=<Param>	1. OK 2. FAIL	Param : Device Address

Example:

```
AT+FSAD=1234,56,abcdef\r\n
OK
AT+LINK=1234,56,abcdef\r\n
OK
```

32. Disconnect

Command	Respond	Parameter
AT+DISC	1. +DISC:SUCCESS OK 2. +DISC:LINK_LOSS OK 3. +DISC:NO_SLC OK 4. +DISC:TIMEOUT OK 5. +DISC:ERROR OK	Param : Device Address

33. Energy-saving mode

Command	Respond	Parameter

AT+ENSNIFF=<Param>	OK	Param : Device Address

Microcontroller Coding:

```
#include <Servo.h> //library for servo motor
//______________________(Care Wheel)
#define FW_Wheel_1 9
#define FW_Wheel_2 8
#define BW_Wheel_1 6
#define BW_Wheel_2 7

//______________________(Gas Sensor MQ2)
#define Gas_Sensor 5
//______________________(Gas Sensor MQ2)
#define Water_Pump 3



long duration;
int distance;

Servo myservo;  // servo motor object for its control

int ang = 0;    // a variable to store the servo angle
int OT_serv_Move = 0;

//______________________(Temperature)
const int sensor=A0; // Assigning analog pin A1 to variable 'sensor'
const int sensor_pin=A1; //Water Sensor Data Receive.
float tempc;  //variable to store temperature in degree Celsius
float tempf;  //variable to store temperature in Fahreinheit
float vout;  //temporary variable to hold sensor reading
int Upload_SensorDATA=0;
int output_value= 0;
```

```
void setup() {
  Serial.begin(9600);
  myservo.attach(2);  // servo motor is attached to pin no 8 og Arduino
  myservo.write(0);
  pinMode(FW_Wheel_1,OUTPUT);
  pinMode(FW_Wheel_2,OUTPUT);
  pinMode(BW_Wheel_1,OUTPUT);
  pinMode(BW_Wheel_2,OUTPUT);
  digitalWrite(FW_Wheel_1,LOW);
  digitalWrite(FW_Wheel_2,LOW);
  digitalWrite(BW_Wheel_1,LOW);
  digitalWrite(BW_Wheel_2,LOW);
  pinMode(Water_Pump,OUTPUT);
  digitalWrite(Water_Pump,LOW);



  pinMode(sensor,INPUT); // Configuring pin A0 as input
  pinMode(Gas_Sensor,INPUT); // Gas Sensor 7 as input

  Serial.print("Water ");
  Serial.println(output_value);
  Serial.println("\n");

  Serial.print("text ");  //("text 0%\n") RoboRemo ID
  Serial.println(tempc);
  Serial.println("\n");

}

void loop() {

//________________________________________________(Soile Sensor)
output_value= analogRead(sensor_pin);
output_value = map(output_value,550,0,0,100);
```

```
//________________________________________________(Temp LM35)
vout=analogRead(sensor);
vout=(vout/1024.0)*5000;     //( val/1024.0)*5000;
tempc=vout/10; // Storing value in Degree Celsius

 Upload_SensorDATA++;
if(Upload_SensorDATA > 1000){
  Serial.print("Water ");
  Serial.println(output_value);
  Serial.println("\n");

  Serial.print("text ");  //("text 0%\n") RoboRemo ID
  Serial.println(tempc);
  Serial.println("\n");
    Upload_SensorDATA=0; //RESET COUNT
}

//________________________________________________(Gas Sensor Status)

   //Receive Data From User
 if(Serial.available()>0)
   {
      char data= Serial.read(); // reading the data received from the bluetooth module
      if(data == 'A'){
          if(OT_serv_Move == 0){ //On Time Rotetion
          Clock_Rotation();
          OT_serv_Move =1;    //Reset
          }
        }
      else if(data == 'C'){
          if(OT_serv_Move == 1){
```

```
          Anty_Clock_Rotation();
          OT_serv_Move =0;      //Reset
          }
        }
      else if(data == 'F'){
          CareMove_FW();
       }
      else if(data == 'B'){
          CareMove_BW();
       }
      else if(data == 'L'){
          CareMove_LW();
       }
      else if(data == 'R'){
          CareMove_RW();
       }
      else if(data == 'S'){
          CareMove_Stop();
       }

      else if(data == 'T'){


       }
      else if(data == 'U'){


       }
      else if(data == 'V'){


       }
//__________________________________________Water PUMP CONTOLL
       else if(data == 'W'){
         digitalWrite(Water_Pump ,HIGH);
       }
      else if(data == 'X'){
         digitalWrite(Water_Pump ,LOW);
       }
      else if(data == 'Y'){


       }


   }


}





Servo Clock_Rotation()
 {
    Serial.println("Rotation from 0 to 180 degress");
    for (ang = 0; ang <= 35; ang += 1) // goes from 0 degrees to 180 degrees with a step og 5 degree
      {
        myservo.write(ang);              // rotates the servo to rotate at specific angle
        delay(10);     // adding delay of 50 msec
       }
 }
Servo Anty_Clock_Rotation()
{
    Serial.println("Rotation from 180 to 0 degress");
    for (ang = 35; ang >= 0; ang -= 1) // goes from 180 degrees to 0 degrees with a step of 5 degree
      {
        myservo.write(ang);              // rotates the servo to rotate at specific angle
        delay(10);                    // adding delay of 50 msec
       }
  }

void CareMove_FW()
  {
    digitalWrite(FW_Wheel_1,LOW);
    digitalWrite(FW_Wheel_2,HIGH);
    digitalWrite(BW_Wheel_1,HIGH);
    digitalWrite(BW_Wheel_2,LOW);
  }
void CareMove_BW()
  {
    digitalWrite(FW_Wheel_1,HIGH);
    digitalWrite(FW_Wheel_2,LOW);
    digitalWrite(BW_Wheel_1,LOW);
    digitalWrite(BW_Wheel_2,HIGH);
  }
void CareMove_LW()
  {
    digitalWrite(FW_Wheel_1,HIGH);
    digitalWrite(FW_Wheel_2,LOW);
    digitalWrite(BW_Wheel_1,HIGH);
    digitalWrite(BW_Wheel_2,LOW);
  }
void CareMove_RW()
  {
     digitalWrite(FW_Wheel_1,LOW);
    digitalWrite(FW_Wheel_2,HIGH);
    digitalWrite(BW_Wheel_1,LOW);
    digitalWrite(BW_Wheel_2,HIGH);
  }
void CareMove_Stop()
  {
    digitalWrite(FW_Wheel_1,LOW);
    digitalWrite(FW_Wheel_2,LOW);
    digitalWrite(BW_Wheel_1,LOW);
    digitalWrite(BW_Wheel_2,LOW);
  }
```

References

[1] I. Aris, A. K. Parvez Iqbal, A. R. Ramli and S. Shamsuddin. “Design and development of a programmable painting robot for houses and buildings.,” Journal Technology, Universiti Technology Malaysia, vol. 42(A), pp. 27-48,2005.

[2] B. Kahane, Y. Rosenfeld: “Balancing human-and-robot integration in building task, “Computer-Aided Civil and Infrastructure Engineering, vol.19, pp. 393-410, 2004.

[3] Warszawsky, Y. Rosenfeld: “Robot for interior finishing works in building: feasibility analysis,” ASCE Journal of Construction Engineering and Management, vol.120 (1), pp.132-151, 1994.

[4] Naticchia, A. Giretti, A. Carbonari. “Set up of an automated multi-color system for interior wall painting,” International Journal of Advanced Robotic systems, vol. 4, No. 4, pp.407-416, 2007.

[5] Naticchia, A. Giretti, A. Carbonari,“Set up of a robotized system for interior wall painting,” In the 23rd International Symposium on Automation and Robotics in Construction, Tokyo, Japan, pp.194-199, 2006.

[6] M. De Grassi, B. Naticchia, A. Giretti & Carbonaro, “Development of an automatic four-color spraying device carried by a robot,” In the24th International Symposium on Automation and Robotics in Construction, Sendai, Japan, pp.235-240, 2007.

[7] M. Sorour, M. Abdellatif, A. Ramadan and A. AboIsmail, “Development of Roller-Based Interior Wall Painting Robot”, Proceeding of the ICAM ,Venice, Italy,pp:1785-1792, 2011.

[8] Kyung-Seok Byun, Sung-Jae Kim; Jae-Bok Song, Design of a four-wheel edomni-directional mobile robot with variable wheel arrangement mechanism, ICRA pp:720-725, 2002.

[9] B. E. Teoh and S. V. Ragavan, “PAINT Bot-FPGA Based Wall Painting Service Robot Prototype,” in the Recent Advances in Intelligent Computational Systems, Trivandrum, India, pp:777-782,2011.

[10] M. Udengaard and K. Iagnemma, Analysis Design and Control of an Omni-directional Mobile Robot on Rough Terrain, Journal of Mechanical Design, Vol. 131, ASME Transactions, 2009.