**CHAPTER 1**

**INTRODUCTION**

Smart phones are becoming a basic need in day to day life with massive storage capacities, processors, and vast communicating methodologies. Bluetooth is mainly used for exchanging data between different devices be it two smart phones or be it a robot and a smart phone. It mainly performs data transmission and even improve the characteristics of the smart phone, it was developed by telecom vendor Ericsson in 1994, shows its merits by incorporation with smart phones. It has changed the medium of how people use digital devices at home or offices and has brought wireless devices in existence. The basic element of a Bluetooth is Piconet, which is a collection several slave devices operating together with one master. Maximum of seven slaves can share a common master through a same link. Even several piconet can link together and form scatternet. It is useful in home environments, looking at its range or normal working area is 8 meters. Bluetooth has gradually increased users to prosecute smart phones, which have gingerly turned into a multipurpose portable device and are accessible to people for their use. Present day, android is widely accepted as an open source platform. Android consist of a complete package involving an operating system, middleware layer and core applications. A Smartphone is a cell phone built on a mobile computing platform, which has big number of boosted connectivity and computing ability than what a feature phone has. In this paper, we are overcoming the problem of traditional robots, which are usually handled with any remote controller. Reducing the remote work we are making the robot move by just a click on the cell phone with android operating system.

A Smartphone is a mobile phone built on a mobile computing platform, which has more advanced connectivity and computing ability than what a feature phone has. Smartphone’s are a more efficient and affordable hand held devices which can be used to support collaborative activities in a community. It is a result of a huge and remarkable advancement in the field of mobile phones technology. Human beings are anxiously working on finding new ways of interacting with machines. Vacuum cleaner is designed to make cleaning process become easier for human task. This project is a combination of hardware and software which has microcontroller, motor shield, sensor, an android application and finally a Bluetooth module via which the hardware connects the software. Smartphone, a small yet powerful device is rapidly changing its traditional ways of human-machine interaction. Nowadays, Modern smart phones are embedded with Bluetooth module, accelerometer sensor and are powered by different operating systems such as Symbian, Bada, and Android OS etc. Among all available mobile operating systems Android OS has gained significant popularity after being launched in the year 2008, overtaking all its previous competitors due to performance and open architecture Android platform brought a big revolution in the field of application development for cell phones, opening for technical exploration. This data is transmitted via Bluetooth module of Smartphone using an android app to the robot. Further, this data is processed by a microcontroller embedded on the robot to perform desirable motions. In this context, a robot is similar to any machine that is controlled by man varying from a simple toy to heavy machinery. Robots have replaced humans in performing various tasks that human are unable to perform due to physical disability, extreme environments or size limitation. Smartphone’s have proved to be of much more aid than being a device just for making calls. The large world is merging together into the palms of humans in the form of a Smartphone.

Industrial robots have designed to reduce human effort and time to improve productivity and to reduce manufacturing cost. Today human-machine interaction is moving away from mouse and pen and becoming much more pervasive and much more compatible with the physical world. Android app can control the robot motion from a long distance using Bluetooth communication to interface controller and android. Microcontroller ATMEGA328P-PU can be interfaced to the Bluetooth module though UART protocol and code is written in embedded C language. As per the commands received from android app the robot motion can be controlled. The output motion of a robotic vehicle is accurate and repeatable. Pick and Place robots can be reprogrammable and tool can be interchanged to provide for multiple applications. The purpose of this work is to design and implement an Android Controlled Bluetooth Robot which is used for Surveillance, home automation, wheelchairs, military and hostages Rescue applications.

**CHAPTER 2**

**PHYSICAL DESCRIPTION**

**2.1 BLOCK DIAGRAM**

**Driver section:**

12 V BATTERY

BLUETOOTH MODULE

ARDUINO UNO

MOTOR SHIELD

MOBILE APP

DC MOTOR

DC MOTOR

**FIG 2.1.1: BLOCK DIAGRAM OF DRIVER SECTION**

**CAMMERA SECTION:**

PC

BROWSER

IP WEBCAM APP

**FIG 2.1.2: BLOCK DIAGRAM OF CAMMERA SECTION**

**2.2 BLOCK DIAGRAM DESCRIPTION:**

**DRIVER SECTION:**

* When the power supply is given to arduino. Open the RemoteXY mobile app in android mobile.
* Connect the Bluetooth device to the mobile. By connecting the device it open a page with joystick and two switches.
* One switch is for G-Sensor which is used to tilt the phone.
* By tilting the phone the commands are moving to arduino from Bluetooth module.
* When arduino receives the commands, it encodes the code and gives the signals to the motor shield.
* The motor shield has external supply of power to give the extra power to the motors.
* The motor shield gives the signal to motor according to the directions given in mobile app.

**CAMMERA SECTION:**

* Open the IP webcam app in the mobile. It opens a settings page scroll down to the end there will be a start server click on it.
* It starts the server to browse in our pc or laptop.
* Open the internet explorer or Google chrome in a pc or laptop.
* Enter the IP address in the search box. It opens a page click on home.
* Click browser to see the video and “HTML5 WAV” to visible the audio.

**CHAPTER 3**

**HARDWARE DESCRIPTION**

**3.1 HARDWARE COMPONENTS**

**3.1.1 ARDUINO UNO:**

The Arduino Uno 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. Revision 3 of the board has the following new features:

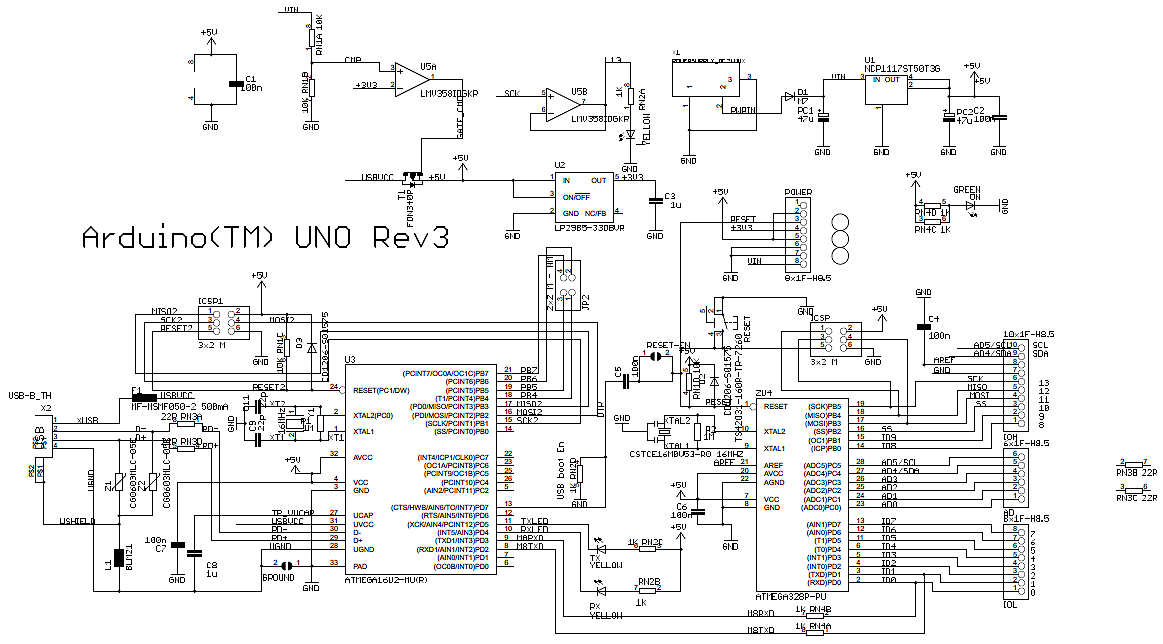
* 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and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

"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; for a comparison with previous versions, see the index of Arduino boards.

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 6DC Current per I/O Pin 40mA DC Current for 3.3V Pin 50mA Flash Memory 32KB(ATmega328) of which 0.5 KB used by bootloader SRAM 2KB (ATmega328) EEPROM 1 KB (ATmega328) Clock Speed 16MHz

**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.

**Schematic Design:**

****

**Fig 3.1:schematic diagram of arduino**

**Reference design:**



**Fig 3.2: reference design of arduino**

**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:

* VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* 5V. 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.
* 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.
* 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 40mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

* 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.
* 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.
* PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
* SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
* 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:

* 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:

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

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

**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 .inf 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). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

**Programming:**

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

* On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
* On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

**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 the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad 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; see this forum thread for details.

**USB Overcurrent Protection:**

The Arduino Uno has a resettable polyfuse 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 500mA 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 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

**3.1.2 MOTOR SHIELD:**

This motor shield allows Arduino to drive two channel DC motors. It uses a L298N chip which deliveries output current up to 2A each channel. The speed control is achieved through conventional PWM which can be obtained from Arduino’s PWM output Pin 5 and 6. The enable/disable function of the motor control is signaled by Arduino Digital Pin 4 and 7.

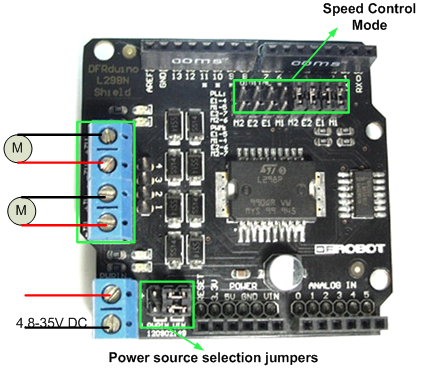
The Motor shield can be powered directly from Arduino or from external power source. It is strongly encouraged to use external power supply to power the motor shield.

* Logic Control Voltage:5V (From Arduino)
* Motor Driven Voltage:4.8~35V (From Arduino or External Power Source)
* Logic supply current Iss:=36Ma
* Motor Driven current Io:=2A
* Maximum power consumption: 25W (T=75?)
* PWM,PLL Speed control mode
* Control signal level:

High: 2.3V=Vin=5V

Low:-0.3V=Vin=1.5V

**DIAGRAM:**



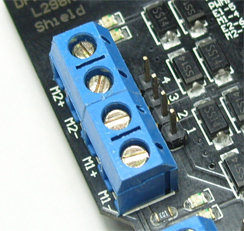
**Fig 3.3(L298N MOTOR SHIELD)**

**Control Mode Selection Jumpers:** The shield supports PWM and PLL (Phased Locked Loop) control Modes. The PWM mode uses E1 and E2 to generate PWM signal. The PLL mode uses M1 and M2 to generate phase control signal.



**Fig 3.4(Control mode selection jumpers)**

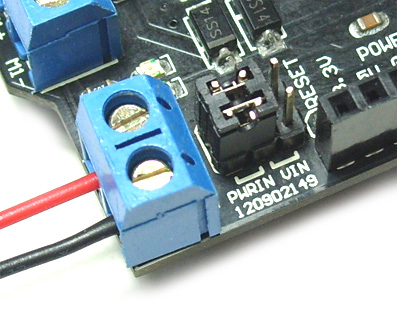
**Motor Terminal:** Two DC motors are connected to blue motor terminals. The male header beside the terminals is the same as the motor terminals.



**Fig 3.5(Motor terminals)**

**PWRIN:** The motors can be powered by external power supply when the motor current exceeds the limits provided from the Arduino. The switch between external and Arduino power is implemented by two jumpers.

* PWRIN: External Power
* VIN: Arduino Power



**Fig 3.6(External power supply to motors)**

**NOTE:** When the motor shield is powered by external power source, make sure the external power source and Arduino have the same GND.

**Control Signal Truth Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **E1** | **M1** |  | **E2** | **M2** |  |
| L | X | Motor 1 disabled | L | X | Motor 2 disabled |
| H | H | Motor 1 backward | H | H | Motor 2 backward |
| PWM | X | PWM speed control | PWM | X | PWM speed control |

**Table 3.1: control signal truth table**

**Note:** H is High level; L is Low level; PWM is Pulse Width Modulation signal; X is any voltage level.

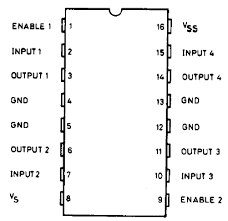
**Pin Allocation:**

|  |  |
| --- | --- |
| **Pin** | **Function** |
| Digital 4 | Motor 2 direction control |
| Digital 5 | Motor 2 PWM control |
| Digital 6 | Motor 1 PWM control |
| Digital 7 | Motor 1 direction control |

**Table 3.2(PWM control)**

|  |  |
| --- | --- |
| **Pin** | **Function** |
| Digital 4 | Motor 2 enable control |
| Digital 5 | Motor 2 direction control |
| Digital 6 | Motor 1 enable control |
| Digital 7 | Motor 1 enable control |

**Table 3.3 (PLL mode)**



**Fig 3.6 (pin diagram of L298N)**

**3.1.3 DC MOTORS:**

****

**Fig 3.8: DC Motor**

Centre Shaft Economy Series DC Motor is high quality low cost DC geared motor. It has steel gears and pinions to ensure longer life and better wear and tear properties. The gears are fixed on hardened steel spindles polished to a mirror finish. The output shaft rotates in a plastic bushing. The whole assembly is covered with a plastic ring. Gearbox is sealed and lubricated with lithium grease and require no maintenance. The motor is screwed to the gear box from inside.

Although motor gives 200 RPM at 12V but motor runs smoothly from 4V to 12V and gives wide range of RPM, and torque. Tables below gives fairly good idea of the motor's performance in terms of RPM and no load current as a function of voltage and stall torque, stall current as a function of voltage. For compatible wheels refer to Wheels and Accessories product category.

You can also mount this motor on the chassis using Motor Mount for Centre Shaft Economy Series DC Motor. For adding Position Encoder, refer to Encoder Kit for Centre Shaft Economy Series DC Motor.

**Technical Specs:**

* Metal Gears
* Operating Voltage: 6-12V
* RPM: 300
* Central Shaft
* Shaft Length: 25mm
* Shaft Diameter: 5mm
* Wire Length: 25cm approx
* Weight: 180 grams approx

**Inside view of Centre Shaft Economy Series DC Motor:**

****

**FIG 3.9:Inside view of center shaft economy series DC motor**

**Motor performance based on RPM and no load current as function of input voltage:**

|  |  |  |
| --- | --- | --- |
| **Voltage(v)** | **RPM(no load)** | **Current(A)** |
| 4 | 59 | 0.029 |
| 5 | 78 | 0.030 |
| 6 | 95 | 0.031 |
| 7 | 106 | 0.047 |
| 8 | 124 | 0.051 |
| 9 | 141 | 0.052 |
| 10 | 160 | 0.052 |

**TABLE 3.4: motor performance on RPM and no load current**

**3.1.4 BLUETOOTH MODULE:**

HC serial Bluetooth products consist of Bluetooth serial interface module and Bluetooth adapter, such as:

(1) Bluetooth serial interface module:

Industrial level: HC-03, HC-04(HC-04-M, HC-04-S)

Civil level: HC-05, HC-06(HC-06-M, HC-06-S)

HC-05-D, HC-06-D (with baseboard, for test and evaluation)

(2) Bluetooth adapter:

HC-M4

HC-M6

Bluetooth serial module is used for converting serial port to Bluetooth. These modules have two modes: master and slaver device. The device named after even number is defined to be master or slaver when out of factory and can’t be changed to the other mode. But for the device named after odd number, users can set the work mode(master or slaver) of the device by AT commands.

HC-04 specifically includes:

Master device: HC-04-M, M=master

Slave device: HC-04-S, S=slaver

The default situation of HC-04 is slave mode. If you need master mode, please state it clearly or place an order for HC-O4-M directly. The naming rule of HC-06 is same. When HC-03 and HC-05 are out of factory, one part of parameters are set for activating the device. The work mode is not set, since user can set the mode of HC-03, HC-05 as they want. The main function of Bluetooth serial module is replacing the serial port line, such as:

1. There are two MCUs want to communicate with each other. One connects to Bluetooth master device while the other one connects to slave device. Their connection can be built once the pair is made. This Bluetooth connection is equivalently liked to a serial port line connection including RXD, TXD signals. And they can use the Bluetooth serial module to communicate with each other.

2. When MCU has Bluetooth salve module, it can communicate with Bluetooth adapter of computers and smart phones. Then there is a virtual communicable serial port line between MCU and computer or smart phone.

3. The Bluetooth devices in the market mostly are salve devices, such as Bluetooth printer, Bluetooth GPS. So, we can use master module to make pair and communicate with them. Bluetooth Serial module’s operation doesn’t need drive, and can communicate with the other Bluetooth device who has the serial. But communication between two Bluetooth modules requires at least two conditions:

(1) The communication must be between master and slave.

(2) The password must be correct.

However, the two conditions are not sufficient conditions. There are also some other conditions basing on different device model. Detailed information is provided in the following chapters. In the following chapters, we will repeatedly refer to Linvor’s (Formerly known as Guangzhou HC Information Technology Co., Ltd.) material and photos.

The Bluetooth serial module named even number is compatible with each other; The salve module is also compatible with each other. In other word, the function of HC-04 and HC-06, HC-03 and HC-05 are mutually compatible with each other. HC-04 and HC-06 are former version that user can’t reset the work mode (master or slave). And only a few AT commands and functions can be used, like reset the name of Bluetooth (only the slaver), reset the password, reset the baud rate and check the version number. The command set of HC-03 and HC-05 are more flexible than HC-04 and HC-06’s. Generally, the Bluetooth of HC-03/HC-05 is recommended for the user.

Here are the main factory parameters of HC-05 and HC-06. Pay attention to the differences:

|  |  |
| --- | --- |
| **HC 05** | **HC 06** |
| Master and slave mode can be switched | Master and slave mode can’t be switched |
| Bluetooth name: HC-05 | Bluetooth name: HC-06 |
| Password:1234 | Password:1234 |
| Master role: have no function to remember the last paired salve device. It can be made paired to any slave device. In other words, just set  AT+CMODE=1 when out of factory. If you want HC-05 to remember the last paired slave device address like HC-06, you can set AT+CMODE=0after paired with the other device. Please refer the command set of HC-05 for the details. | Master role: have paired memory to remember  last slave device and only make pair with that  device unless KEY (PIN26) is triggered by high  level. The default connected PIN26 is low level. |
| Pairing: The master device can not only make pair with the specified Bluetooth address, like  cell-phone, computer adapter, slave device, but  also can search and make pair with the slave  device automatically.  Typical method: On some specific conditions,  Master device and slave device can make pair with each other automatically. (This is the default method.) | Pairing: Master device search and make pair with the slave device automatically.  Typical method: On some specific conditions,  master and slave device can make pair with each other automatically. |
| Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules. | Multi-device communication: There is only point to point communication for modules, but the adapter can communicate with multi-modules. |
| During the process of communication, the module can enter to AT mode by setting PIN34 to be high level. By releasing PIN34, the module can go back to communication mode in which user can inquire some information dynamically. For example, to inquire the pairing is finished or not. | During the communication mode, the module  can’t enter to the AT mode. |
| Default communication baud rate: 9600,  4800-1.3M are settable. | Default communication baud rate: 9600,  1200-1.3M are settable. |
| KEY: PIN34, for entering to the AT mode. | KEY: PIN26, for master abandons memory. |
| LED1: PIN31, indicator of Bluetooth mode. Slow flicker (1Hz) represents entering to the AT mode2, while fast flicker(2Hz) represents entering to the AT mode1 or during the communication pairing. Double flicker per second represents pairing is finished, the module is communicable. LED2: PIN32, before pairing is at low level, after the pairing is at high level. The using method of master and slaver’s indicator is the same.  Notice: The PIN of LED1 and LED2 are connected  with LED+. | LED: The flicker frequency of slave device is  102ms. If master device already has the memory of slave device, the flicker frequency during the pairing is 110ms/s. If not, or master has emptied the memory, then the flicker frequency is 750m/s. After pairing, no matter it’s a master or slave  device, the LED PIN is at high level.  Notice: The LED PIN connects to LED+ PIN. |
| Consumption: During the pairing, the current is fluctuant in the range of 30-40mA. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules. | Consumption: During the pairing, the current is fluctuant in the range of 30-40mA. The mean current is about 25mA. After paring, no matter processing communication or not, the current is 8mA. There is no sleep mode. This parameter is same for all the Bluetooth modules. |
| Reset: PIN11, active if it’s input low level. It can be suspended in using. | Reset: PIN11, active if it’s input low level. It can be suspended in using. |
| Level: Civil | Level: Civil |

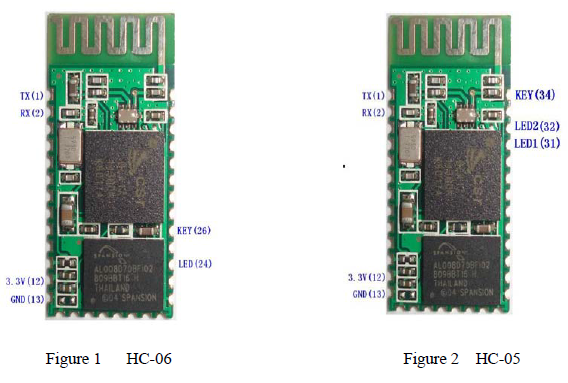
**Table 3.5: difference between HC-05& HC-06**

The table above that includes main parameters of two serial modules is a reference for user selection

HC-03/HC-05 serial product is recommended.

**Information of Package:**

The PIN definitions of HC-03, HC-04, HC-05 and HC-06 are kind of different, but the package size is the same: 28mm \* 15mm \* 2.35mm. The following figure is a picture of HC-06 and its main PINs. Figure 2 is a picture of HC-05 and its main PINs.



**Fig 3.10: BLUETOOTH MODULES OF HC-05&HC-06**

**Pairing introduction:**

HC-06 master device has no memory before the first use. If the password is correct, the mater device will make pair with the slave device automatically in the first use. In the following use, the master device will remember the Bluetooth address of the last paired device and search it. The searching won’t stop until the device is found. If master device’s PIN26 is input high level, the device will lose the memory. In that occasion, it’ll search the proper slave device like the first use. Based on this function, the master device can be set to make pair with the specified address or any address by user.

**Reset new password introduction:**

User can set a new password for the HC-06 through AT+PINxxxx command. But the new password will become active after discharged all the energy of the module. If the module still has any energy, the old one is still active. In the test, for discharging all the system energy and activating the new password, we can connect the power supply PIN with GND about 20 seconds.

**3.1.5 MOBILE APPLICATION REMOTE XY:**

RemoteXY is easy way to make and use a mobile graphical user interface for controller boards to control via smartphone or tablet. The system includes:

* Editor of mobile graphical interfaces for controller boards, located on the site remotexy.com
* Mobile app RemoteXY that allows to connect to the controller and control it via graphical interface. Download app.

**Distinctive features:**

* The interface structure is stored in the controller. When connected, there is no interaction with servers to download the interface. The interface structure is downloaded to the mobile application from the controller.
* One mobile application can manage all your devices. The number of devices is not limited.

**Connection between the controller and the mobile device using:**

* Bluetooth;
* WiFi client and access point;
* Ethernet by IP or URL;
* Internet from anywhere through the cloud server.

The source code generator have support next controllers:

* Arduino UNO, Arduino MEGA, Arduino Leonardo, Arduino Pro Mini, Arduino NANO, Arduino MICRO;
* WeMos D1, WeMos D1 R2, WeMos D1 mini;
* NodeMCU V2, NodeMCU V3;
* The AirBoard;
* ChipKIT UNO32, ChipKIT uC32, ChipKIT Max32;

Supported communication modules:

* Bluetooth HC-05, HC-06 or compatible;
* WiFi ESP8266;
* Ethernet Shield W5100;

Supported IDE:

* Arduino IDE;
* FLProg IDE;
* MPIDE;

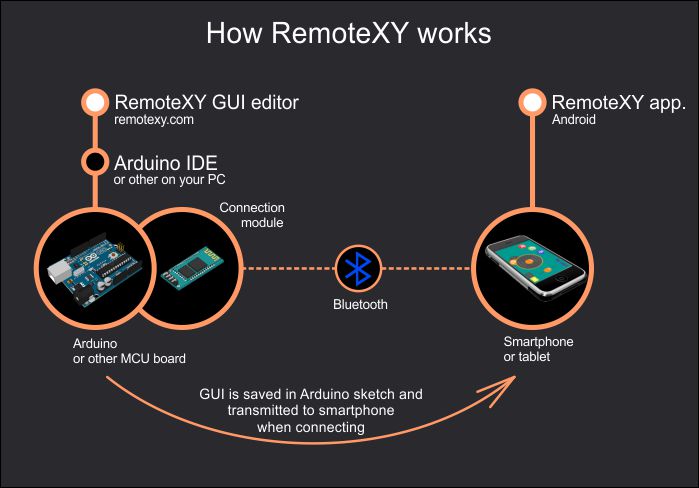
Supported mobile OS:

* Android;
* iOS;

RemoteXY is easy way to make a unique graphical interface to control microcontroller device via mobile application, Arduino for example.

RemoteXY allows:

* To develop any graphical management interface, using the control, display and decoration elements any combination thereof. You can develop the graphical interface for any task, placing the elements on the screen using the online editor. Online editor posted on the website remotexy.com.
* After the development of the graphical interface, you get the source code for the microcontroller that implements your interface. The source code provides a structure for interaction between your program with the controls and display. Thus you can easily integrate the control system into your task for which you are developing the device.
* To manage microcontroller device using your smartphone or tablet with the graphical interface. For manage used mobile application RemoteXY.
* Using one mobile application, you can manage a large number of devices with different graphical management interfaces. As the interface description is stored on board the microcontroller device.

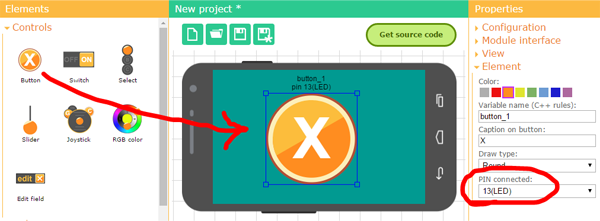


**Fig 3.11: HOW REMOTEXY APP WORKS**

**Arduino UNO + Bluetooth HC-05(06) interfacing in app:**

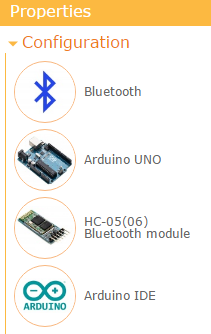
The Bluetooth module HC-05, HC-06 or compatible one must be connected to the controller. The smartphone or tablet must support Bluetooth.

* Enter the RemoteXY editor. Set one button in the smartphone field. Highlight this button, then, select the property "Snap to pin" to the value 13 (LED) in the right pane of the "Element" tab.



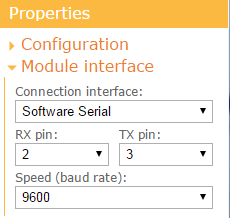
**Fig 3.12: INTERFACING BUTTONS IN APP**

* In the right pane, select the following settings under the "Configuration" tab.



**Fig 3.13: CONFIGURATION TAB**

* In the right pane set the following settings under the "Module interface" tab.

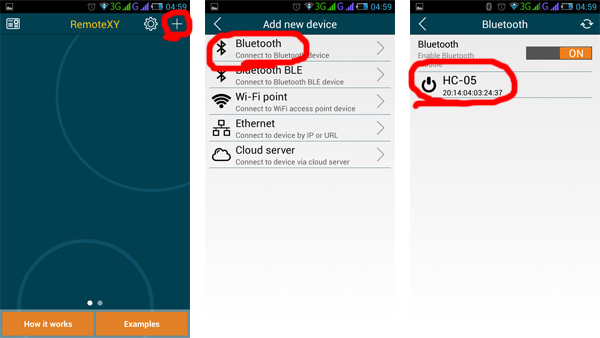


**Fig 3.14: MODULE INTERFACE TAB**

* The settings indicate that the HC-05 (06) module is connected to the Arduino via the software serial port SoftwareSerial using contacts 2 and 3 at 9600 speed. This is the standard speed for the HC-05 (06) modules.
* Press the "Get source code" button in the editor.
* In the opened page with the source code of the sketch, download it to your computer (the link "Download code") and open it in the Arduino IDE.
* Also, download the library RemoteXY (the link "Download library") from this page.
* The library being installed correctly, the source code for the sketch must be compiled without errors.

**HOW TO CONNECT BLUETOOTH TO APP:**

* Install the RemoteXY mobile application on your smartphone/tablet.
* Press the new connection "+" button in the top panel in the application. In the window that opens, select the "Bluetooth" connection. If you are using the HM-10 module, select the “Bluetooth BLE” connection.
* In the application, turn on the Bluetooth and press the button for updating the list of available devices. Your device can have one of the following names: "HC-05", "HC-06", "INVOR". Select it.
* A window for entering the Bluetooth pairing password will open. The default password for HC-05 (06) can be either 1234 or 0000. Enter the password and the connection will start.



**Fig 3.15: HOW TO CONNECT BLUETOOTH TO APP**

If there is no connection, it is likely that you will get an error:

* This means that either the request commands do not reach the controller, or the controller responses are not returned to the smartphone/tablet.
* If you receive this specific error, check yourself on the list, going to the next point only after checking the previous one:
* Your Arduino is not on;
* The required sketch has not been loaded into the Arduino;
* The RemoteXY library did not update, using old version of library;
* Power is not supplied to the Bluetooth module, the power contacts may be reversed;
* RX and TX contacts of the Bluetooth module, or one of them are not connected to the controller, a bad contact;
* The RX and TX contacts of the Bluetooth module are not connected correctly, they may be interchanged, check the scheme (step 4);
* Incorrect configuration settings or connection settings (step 2) had been chosen before generating the source code;
* The Bluetooth module is defective.

**3.1.6 IP WEB CAM APP:**

**Description:**

IP Webcam turns your phone into a network camera with multiple viewing options. View your camera on any platform with VLC player or web browser. Stream video inside WiFi network without internet access. Optional Ivideon cloud broadcasting is supported for instant global access.

Two-way audio supported in tinyCam Monitor on another android device. Use IP Webcam with third-party MJPG software, including video surveillance software, security monitors and most audio players.

**How to open app:**

Step 1: Download the IP Webcam application from the Google Play store of android mobile.

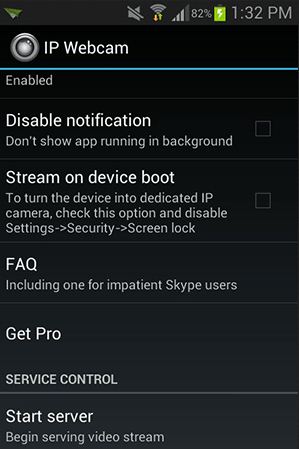
Step 2: Install the application on your android Smartphone.

Step 3: Compared with most apps you’ve come across, this particular one works with a unique type of user interface. The first screen is basically the Settings page where you could set up your entire webcam configurations. The settings options offered here are many; simply click on these to find out more about them.



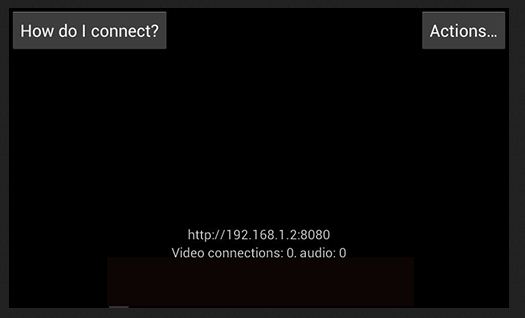
**Fig 3.16: IPWEBCAM APP**

Step 4: Scroll down towards the bottom of the page and there you will see an option indicating “Start server.” just Tap on it and wait, it is going to start your webcam server that you may gain access to starting from your browser on the laptop or computer.



**Fig 3.17: IP WEBCAM SETTINGS PAGE**

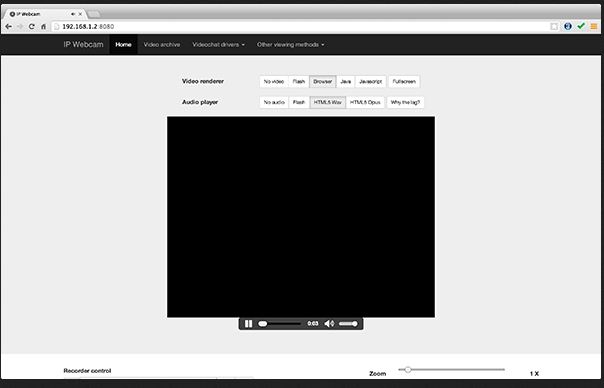
Step 5: The moment you start the server, it’ll instantly start streaming the data of your device camera. Together with that, make sure you see an IP address on the screen; merely note it down because you are gonna be working with it in the next steps of the process.



**Fig 3.18: IP ADDRESS AND VIDEO WINDOW**

Step 6: Now, Go to web browser on your computer. And then type in the IP address you noted down from the previous step and hit Enter.

Step 7: I’s takes you to next screen, where you’re going to be capable of view your camera. Within this page, select “Browser” through the Video sender section along with “HTML5 Wav” from the Audio player section.



**Fig 3.19:VIEW IN PC**

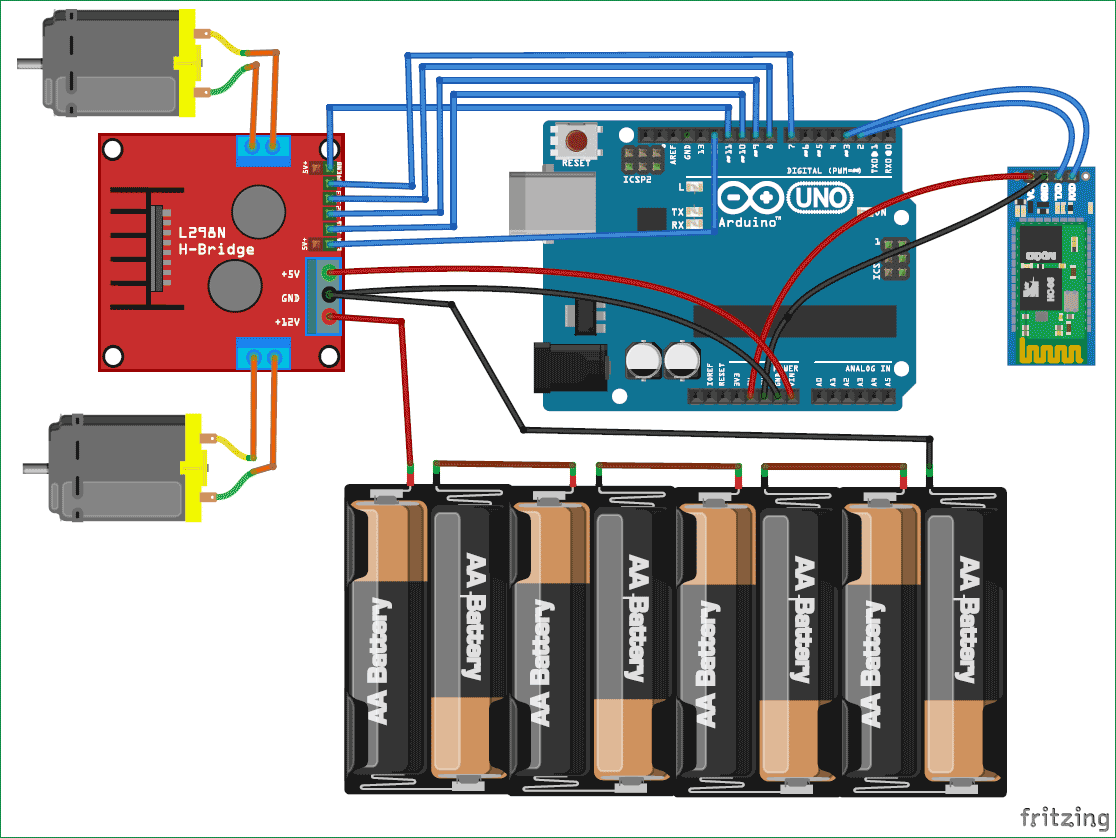
Step 8: Make sure you now be capable of seeing your camera’s footage directly in your web browser. The IP webcam application provides a some more features at the same time, such as to be able to record videos, zoom-in, focus, video quality, even more using your Android device as a webcam can help you stretch your budget that you’d usually invest in purchasing a costly webcam that does what your usual Android device does.

**Features include:**

* + Video upload to Dropbox, SFTP, FTP and Email using Filoader plug-in
  + Several web renderers to choose from: Flash, Javascript or built-in
  + Video recording in WebM, MOV, MKV or MPEG4 (on Android 4.1+)
  + Audio streaming in wav, opus and AAC (AAC requires Android 4.1+)
  + Motion detection with sound trigger, Tasker integration.
  + Date, time and battery level video overlay.
  + Sensor data acquisition with online web graphing.
  + Videochat support (video stream only for Windows and Linux via an universal MJPEG video streaming driver)
  + Cloud push notifications on motion and sound, cloud recording for motion-triggered records, online video broadcasting powered by Ivideon.
  + Extensive baby and pet monitor features: night vision, motion detection, sound detection.

**3.2 CIRCUIT DESCRIPTION**

**Circuit diagram:**

****

**Fig 3.21: CIRCUIT DIAGRAM**

**Circuit Description:**

First of all, we will interface the L298N motor controller with the Arduino. Connect the ENA and ENB pin of the motor controller to the Arduino pin 12 and 11 respectively. These two pins are for the PWM control of the motor. Using theses pins, we can increase or decrease the speed of car. Then connect the IN1, IN2, IN3 and IN4 to the Arduino pins 10, 9, 8 and 7 respectively. These pins will rotate the motors in both directions (clockwise and anti-clockwise).

To power the motor, connect the positive and negative of the battery to the 12V and the ground of the motor controller. Then connect the 5V and the ground from the motor controller to the Arduino Vin and the ground.

Then we will connect the Bluetooth module HC-06 with the arduino. If you have HC-05, then it will work too. Connect the VCC and the ground of the Bluetooth module to the 5V and the ground of the Arduino. Then connect the TX pin of Bluetooth Module to the pin of Arduino and the RX pin to the pin 3 of Arduino. Also check Bluetooth Controlled Toy Car using Arduino to learn more about using Bluetooth with Arduino.

For camera section open IP webcam app in android mobile and click start server option. It opens a camera screen with IP address. Now open your system browser and enter the IP addresses open it. It will open a screen with audio and video player option. Select “browser” in video player and “HTML 5Wav” from the audio player section.

Make sure you now be capable of seeing your camera's footage directly in your web browser. The IP webcam application provides a some more features at the same time, such as to be able to record videos, zoom-in, focus, video quality, even more using your Android device as a webcam can help you stretch your budget that you'd usually invest in purchasing a costly webcam that does what your usual Android device does.

**CHAPTER 4**

**SOFTWARE DESCRIPTION**

**4.1 SOFTWARE CODE**

**Arduino IDE:**

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consists of only two functions:

* setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
* loop(): After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

**PROGRAM CODE:**

#define REMOTEXY\_MODE\_\_SOFTWARESERIAL

#include <SoftwareSerial.h> //Including the software serial library

#include <RemoteXY.h> //Including the remotexy library

/\* RemoteXY connection settings \*/

#define REMOTEXY\_SERIAL\_RX 2 //defining the pin 2 as RX pin

#define REMOTEXY\_SERIAL\_TX 3 //defining the pin 3 as TX pin

#define REMOTEXY\_SERIAL\_SPEED 9600 //setting baudrate at 9600

unsigned char RemoteXY\_CONF[] = //remotexy configuration

{ 3,0,23,0,1,5,5,15,41,11

,43,43,1,2,0,6,5,27,11,5

, 79,78,0,79,70,70,0};

struct { //Function for declaring the variables

signed char joystick\_1\_x; //joystick x-axis

signed char joystick\_1\_y; //joystick y-axis

unsigned char switch\_1; //variables for switch

unsigned char connect\_flag;

} RemoteXY;

//defining the pins for first motor

#define IN1 10

#define IN2 9

#define ENA 12

//defining the pins for second motor

#define IN3 8

#define IN4 7

#define ENB 11

//defining the LED pin

#define ledpin 13

unsigned char first\_motor[3] =

{IN1, IN2, ENA};

unsigned char second\_motor[3] =

{IN3, IN4, ENB};

void Speed (unsigned char \* pointer, int motor\_speed)

{

if (motor\_speed>100) motor\_speed=100;

if (motor\_speed<-100) motor\_speed=-100;

if (motor\_speed>0) {

digitalWrite(pointer[0], HIGH);

digitalWrite(pointer[1], LOW);

analogWrite(pointer[2], motor\_speed\*2.55);

}

else if (motor\_speed<0) {

digitalWrite(pointer[0], LOW);

digitalWrite(pointer[1], HIGH);

analogWrite(pointer[2], (-motor\_speed)\*2.55);

}

else {

digitalWrite(pointer[0], LOW);

digitalWrite(pointer[1], LOW);

analogWrite(pointer[2], 0);

}

}

void setup()

{

//defining the motor pins as the output pins

pinMode (IN1, OUTPUT);

pinMode (IN2, OUTPUT);

pinMode (IN3, OUTPUT);

pinMode (IN4, OUTPUT);

pinMode (ledpin, OUTPUT);

RemoteXY\_Init ();

}

void loop()

{

RemoteXY\_Handler ();

digitalWrite (ledpin, (RemoteXY.switch\_1==0)?LOW:HIGH);

Speed (first\_motor, RemoteXY.joystick\_1\_y - RemoteXY.joystick\_1\_x);

Speed (second\_motor, RemoteXY.joystick\_1\_y + RemoteXY.joystick\_1\_x);

}

**4.2CODE EXPLINATION:**

First of all, we have included the libraries for Software Serial and RemoteXY. The RemoteXY library will help us in setting the app with Arduino, through which we will control the robot car. After that, we have defined the pins for the Bluetooth module, TX from the Bluetooth module is connected to the pin 2 of Arduino which is the RX pin of Arduino and the RX from the Bluetooth module is connected to the pin 3 of the Arduino which is the TX pin of Arduino and set the baud rate of the Bluetooth module at 9600.

When the joystick will be at the center, the speed will be zero and when it will be in the forward direction then the speed will increase from zero to 100. The speed will be decrease from 0 to -100 when the car will move in the reverse direction. The car can also be moved specific speed, this can be done by giving the PWM signal. The PWM signal will be given to the motors according to the rotation of the joystick.

In the following code, we have defined the function which will be called whenever we will move the joystick in the app. When we will turn on the switch in the app then the logic 1 will be given to the pin 13 of the Arduino which turns On the LED pin. While moving the robot car in the forward and backward direction, the Speed function will be called.

**4.3 OUTPUT**

G

**Fig: 4.3(A) mobile app at stop position**

**Fig: 4.3(B) joystick at right side to rotate right side of robot**

**Fig: 4.3(C) joystick at left side to rotate left side of robot**

**Fig: 4.3(D) joystick at upward direction to move robot forward**

**Fig: 4.3(E) joystick at downward direction to move robot backward**

**CHAPTER 5**

**ADVANTAGES &APPLICATIONS**

**ADVANTAGES:**

* Portable and easy to use.
* Easy to control.
* Easy to maintain and repair.
* Easy to capture the video in any areas.

**APPLICATIONS:**

* It used future as robot arm controlled by servo motors.
* Use camera to transmit frames back to Android application for display to user Bluetooth to low bandwidth.
* In the area of suspectance, the robot can be directed and if any smoke or gas is identified the robot can produce alarm and also informs the operator.
* So far the present system is designed mainly for the supervision applications.
* In the near future, productivity and competitiveness in these industries will depend in large part on flexible automation through robotics.

**CONCLUSION**

In this article, we have Control the Robot Car through the G sensor of our mobile phone and we will be able to move the Robot just by tilting the Phone. RemoteXY app is used to create the interface in the Smart Phone for controlling the Robot. We will add the joystick in the interface so that Robot can also be controlled by Joystick as well as by tilting the phone. Thus we are able to capture the video by mobile phone controlled robot using G-Sensor. We also captured the video by IP Webcam as live recording and taken the pictures in our system.

**REFERENCES**

* <https://circuitdigest.com>
* <https://geekeasier.com/how-to-use-your-android-device-as-a-webcam/1190/>
* <https://www.arduino.cc/en/Main/Software>
* <https://www.intorobotics.com/arduino-uno-setup-and-programming-tutorials/>
* [www.instructables.com/id/Arduino-Uno-Basics/](http://www.instructables.com/id/Arduino-Uno-Basics/)]
* <http://remotexy.com/>
* <http://remotexy.com/en/editor/>
* http://remotexy.com/en/library/