**1. Introduction**

Android phones have a huge potential as low cost robot controllers. Android is a comprehensive open source platform designed for mobile devices. They have built in GPS compasses and cameras, Bluetooth and high end processors running at an average of 500Mhz.Hence the project establishes a bridge between the processor Android in the Smartphone and the microcontroller in the robot

The internet combined with smart phones and its peripherals open the doors to limitless mobile possibilities. One of these possibilities is explored and exploited by capturing video by the camera in real-time, and transferring to the mobile. Currently, streaming live video from a mobile phone is limited to commercial products such as Bambuser and Qik, while an open source solution has not yet surfaced. This project develops an android application capable of transferring the live video on the phone Users will have the ability to broadcast news and events live using only an Android-enabled mobile devices and an internet connection via the cellular network or Wifi.

The project consists of developing android application which will communicate with the robot through network connectivity. User sends command to the robot and a robot decodes same and takes action accordingly. At the same time camera will capture video at the robot side and it will send it to the mobile. The external interface is totally controlled by means of the mobile phone. By making use of android phone as a controller for the robots we form a cost efficient, powerful and extensible basis robot. This is even useful to give a richer and cohesive experience to the user.

The main aim of this project is to make new way to control totally standalone system from remote side. In this project robot is standalone system which is control by the remote side android mobile through the network connectivity. Instead of robot any particular task performing standalone system can be control by mobile. With this project it will easy to make consumer efficient product which anyone can use for their own purpose such as security, surveillance, spying or for other use.

**2. Literature Survey**

The emergence of service robots in early 90’s (Helpmate Robots and Robo-Caddy) followed by the development of natural language interface through keyboard has been given by Torrance in 1994[1].

Speech recognition evolved as an up gradation of the past work to communicate with machines but it lacked the standardization of commands due to varying languages, pitch and accent of different users. Hence, researchers [1]-[2] proposed vision-based interface that included gesture recognition through camera to provide geometrical information to the robots. They developed mobile robot systems that were instructed through arm positions but those robot systems couldn’t recognize gestures defined through specific temporal patterns. Other limitation faced by the cameras was the poor illuminations at night and in foggy weather [3]-[4].

Motion technology facilitates humans to interact with machines naturally without any interventions caused by the drawbacks of mechanical devices. Using the concept of gesture recognition, it is possible to move a robot accordingly [5].

Gyroscope and Accelerometers are the main technologies used for human machine interaction that offer very reasonable motion sensitivity, hence, are used in large array of different applications [6]. A lot of work has been done on motion technology using accelerometers [7].

In 2008, Chinese traffic police system used two 3-axis accelerometers fixed on the back of their arms that were synchronized with traffic lights. However, data could only be extracted while the arms would be steady [3]. In 2010, Sauvik Das et al have used an accelerometer as a potential spying device to show locations and activities of user without one’s knowledge [9].

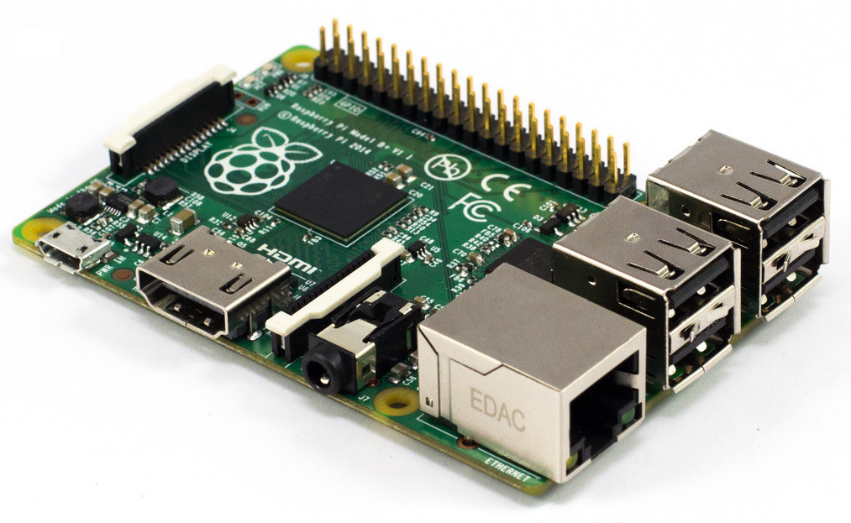
One of the limitations was that inbuilt accelerometer smartphone would have to be in the same place as was in the training mode to make accurate predictions [7].

In late 90’s the smartphones started gaining popularity. The usage of mechanical accelerometer was cumbersome as it possessed the complexity of connections and also portability was a major challenge. With the emergence of smartphone, the technology became lucid as it was equipped with several accessories in concise form [8].

In 2010, Smartphones were used to control Universal Robot Control System by the students of Kyungpook National University, Korea, to design a real time robot control system in ubiquitous environment. However, gestures involved were complex and an extra robot control manager unit was required [9]. Similar work has been done using Symbian [10] and iOS [11] platforms. However, parallel work on Android OS by Google became more popular because of its powerful capabilities and open architecture [12]. Also, it has a large community of developers writing applications that could enhance the functionality of device, written primarily in a customized version of JAVA [13].

Android OS based Smartphone was being interfaced with LEGO Mindstorm RCX and later with NXT to control robot. The usage of NXT eased the integration of sensors [12].Based on Android OS, Craige J. Mouser et al [14] built an application to control and view a live video stream from the remote robot. Smartphones have an inbuilt Bluetooth module that is a wireless technology in a short range communication system that aims to replace cables connecting portable devices. Keeping the above features in view, Nasereddin and Abdelkarim proposed controlling of robots through Bluetooth using Direct Drive and Map-based models [15]. The present work envisages the use of Android Smartphone for the controlling of robot using Bluetooth Wireless Technology.

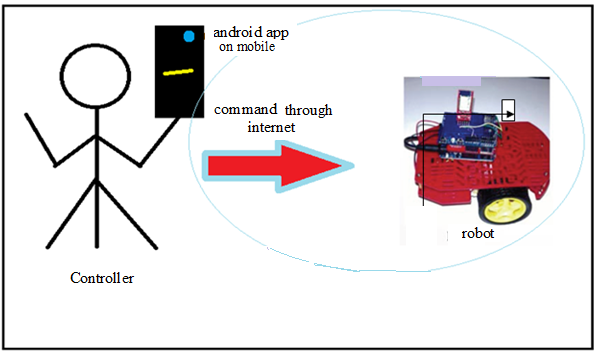
**3. Project Specification**

* ****Raspberry pi B+
* **Specifications**
* **Chip** Broadcom BCM2835 SoC
* **Core architecture** ARM11
* **CPU** 700 MHz Low Power ARM1176JZFS Applications Processor
* **GPU** Dual Core VideoCore IV® Multimedia Co-Processor
* Provides Open GL ES 2.0, hardware-accelerated OpenVG, and
* 1080p30 H.264 high-profile decode
* Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
* **Memory** 512MB SDRAM
* **Operating System Dimensions** 85 x 56 x 17mm
* **Power** Micro USB socket 5V, 2A
* Boots from Micro SD card, running a version of the Linux operating system
* **Connectors:**
* **Ethernet** 10/100 BaseT Ethernet socket
* **Video Output** HDMI (rev 1.3 & 1.4)
* Composite RCA (PAL and NTSC)
* **Audio Output** 3.5mm jack, HDMI
* **USB** 4 x USB 2.0 Connector
* **GPIO Connector** 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip
* Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
* **Camera Connector** 15-pin MIPI Camera Serial Interface (CSI-2)
* **JTAG** Not populated
* **Display Connector** Display Serial Interface (DSI) 15 way flat flex cable connector
* with two data lanes and a clock lane
* **Memory Card Slot** SDIO
* USB Camera
* Support up to 8 Mega pixels
* Frame rate up to 30fps
* True plug and play easy USB interface
* High quality CMOS sensor
* Clear and sharp motion video capturing
* Whole Project Runs on 12VDC, 1.2Ah lead acid battery.
* Android Mobile with Android OS jellybean 4.4.4.
* L293D Motor Driver an integrated IC provides peak current of 600mA for each motor
* 12V, 200 RPM DC gear motor.
* Wifi adapter
* Gives max speed up to 150Mbps
* Plug and play support
* Less power consumption
* Support WPA PSK
* Supported auto connect features

**4. Block Diagram and Description**

**4.1 General Project Diagram**

Project using Raspberry Pi B+ development board:

**Fig: 4.1 general diagram**

Almost more 50% user in the world having android mobile. Using this advantage we make interactive application, where one can interact with the remote side robot. For having perfect interaction we required hardware which should powerful in such a way that it can create internet connection using USB modem and also take and send the images coming from the video camera. To achieve this we required hardware which can run embedded OS, which makes easy way to create required things. Finally considering all aspects we chose Raspberry Pi B+ board.

The Raspberry Pi B+ is the newest member of the Raspberry Beagle Board family. It is a lower-cost, high-expansion focused Raspberry board using a low cost 700 MHz Low Power ARM1176JZFS ARM11 processor. It runs debian wheezy linux OS name Raspbian from micro SD card inserted in it.

**4.2 Block Diagram**

System on Robot side

Android mobile at user side

Wifi network

Power supply unit

USB Modem

USB camera

DC Gear Motor

Motor driver

Android based video streaming application

Raspberry Pi Development board

**Fig: 4.2 block diagram**

1. **Android based application:**

It acts as interface between robot and the mobile phone (android OS). By accepting communication permissions, wifi network connection is establish. It sends commands to the robot by arrow keys and appropriate action is performing on the robot simultaneously the robot sends a live video to the mobile which achieve two way communication.

1. **Raspberry Pi board:**

To get starting with project following things are essential to do.

* To get connected with wifi we have to edit file at location

/etc/network/interfaces file in Raspbian OS available on board as:

auto wlan0

iface wlon0 inetdncp

wpa-ssid ”myhome” ;user name of wifi network

wpa-psk”itsme4545” ; password

Wifi adapter is used to get connected with internet on the board.

* To understand General Purpose Input Output(GPIO) accessing on board small python program is executed first on Raspberry Pias :-

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM) # set board mode to Broadcom

GPIO.setup(17, GPIO.OUT) # set up pin 17

GPIO.setup(18, GPIO.OUT) # set up pin 18

GPIO.output(17, 1) # turn on pin 17

GPIO.output(18, 1) # turn on pin 18

Above example shows simple pin high program. Here pin 17 and 18 are used as output. Using above program pin 17 and 18 is controlled by giving high and low logic. Above logic is given to motor driver circuit to control movement of the robot.

* For video first connect the camera to board and check plug and play is detected or not by using following command:

ls –al /dev/

If video0 is in the list then USB Camera is recognize by OS and can be used for plug and play purpose. To get video streaming, we required to install motion software on the board. After that edit motion.config stored at location in Raspbian OS

/etc/motion/motion.config

And also make change in motion file stored at location in Raspbian OS

/etc/default/motion

Set yes to enable motion daemon start\_motion\_daemon=yes

To start video streaming use command: sudo start service motion

Interfaced camera start acquiring frames and send them on port of robot by then further send to mobile by wifi network.

**Steps wise working of project:**

1. Video Streaming application running at user side android mobile.
2. At robot side Raspbian operating system is installed on Raspberry Pi B+ board and wifi adapter and camera is interfaced with it.
3. User side application is connected to python program running on raspberry pi B+ development board through wifi connectivity via wifi adapter.
4. User gives command to robot through android application and simultaneously video can be view on mobile using USB camera of Robot side.
5. A python program running on raspberry pi B+ receives this command and decodes it. According to particular command respected direction bits of motor driver are made high and robot is control.

**4.3 Raspberry Pi and Motor Driver Interfacing Block diagram**

Motor

Logic Input from Raspberry Pi GPIO

DC Battery

Voltage

Regulator

Power Supply

Motor

Driver

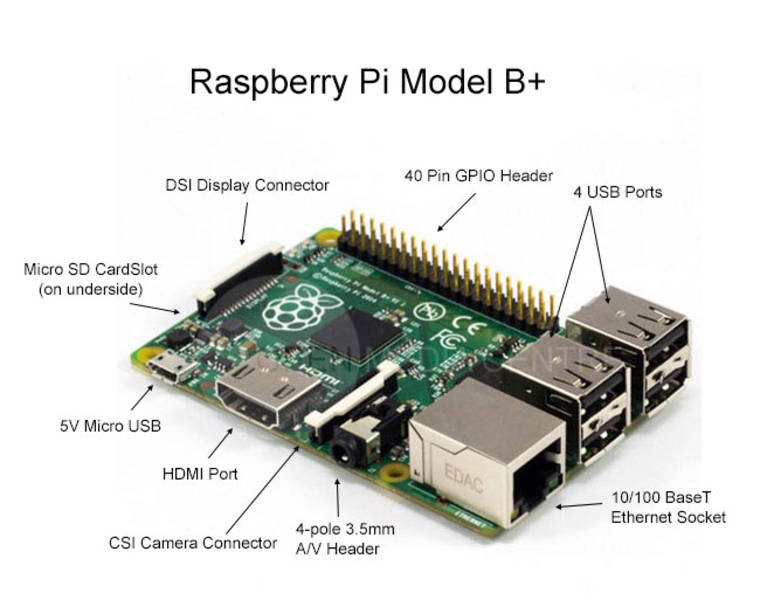
**4.3 Motor Driver Block Diagram**

* Raspberry Pi GPIO pin when high gives 3.3V logic and when low gives 0V.
* We need some circuit which accept logic input from Raspberry pi and drives motor accordingly.
* Motor driver is simple H bridge connection required to drive motor in both directions.
* To provide sufficient current to whole circuit we use battery and to get regulated voltage we use voltage regulator.

**5. Hardware Design**

* 1. **Raspberry Pi Features**

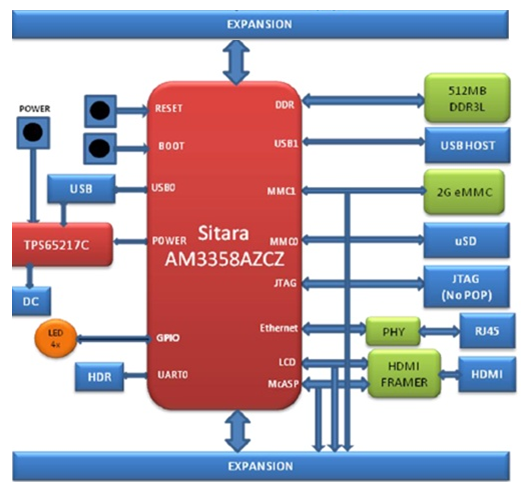
On Robot Side the raspbian OS boot on Raspberry Pi B+ development board.



* 5V Micro USB input that accepts 5V power from adapter or laptop USB port.
* 10/100 Ethernet is the connection to the LAN.
* Micro SD slot is where a micro SD card can be installed.
* HDMI connector is where the display is connected to.
* 4 USB port for USB device interface.
* A/V connector to connect Raspberry pi to TV out display to view GUI.
* HDMI port to connect HDMI display.
* DSI display connector to connect 4” display to board.
* CSI camera connector to connect portable small camera.
* 40 pin GPIO header to connect Raspberry pi to external hardware using different protocols.

**5.2 a3 size complete diagram**

**5.3 Beagle Bone Black key components:**

****

**Fig: 5.2 Beagle bone black key components**

**5.2.1 Processor**

For the initial release, the board uses the Sitara XAM3359AZCZ processor in the 15x15 package. This is basically the same processor as used on the original Beagle Bone. It does use the updated 2.0 revision with several fixes on this new processor as opposed to the original Beagle Bone. A couple of important features from this new processor include:

* 1GHZ Operation
* RTC fix

Eventually the board will move to the Sitara AM3358**B**ZCZ100 device once released and readily available from TI. At this time we do not have a date when this will happen. We do not expect any benefit from moving to this device and there should be no impact seen as a result of making this move,

**5.3.2 Memory**

Described in the following sections are the three memory devices found on the board.

**5.3.3 512MB DDR3L**

A single 256Mb x16 DDR3L 4 GB (512MB) memory device is used. The memory used is the MT41K256M16HA-125 from Micron. It will operate at a clock frequency of 400MHz yielding an effective rate of 800MHZ on the DDR3L bus allowing for 1.6GB/S of DDR3L memory bandwidth.

**5.3.4 32KB EEPROM**

A single 32KB EEPROM is provided on I2C0 that holds the board information. This information includes board name, serial number, and revision information. This is the same as found on the original Beagle Bone. It has a test point to allow the device to be programmed and otherwise to provide write protection when not grounded.

**5.3.5 2GB Embedded MMC**

A single 2GB embedded MMC (eMMC) device is on the board. The device connects to the MMC1 port of the processor, allowing for 8bit wide access. Default boot mode for the board will be MMC1 with an option to change it to MMC0, the SD card slot, for booting from the SD card as a result of removing and reapplying the power to the board. Simply pressing the reset button will not change the boot mode. MMC0 cannot be used in 8Bit mode because the lower data pins are located on the pins used by the Ethernet port. This does not interfere with SD card operation but it does make it unsuitable for use as an eMMC port if the 8 bit feature is needed.

**5.3.6 Micro SD Connector**

The board is equipped with a single micro SD connector to act as the secondary boot source for the board and, if selected as such, can be the primary boot source. The connector will support larger capacity micro SD cards. The micro SD card is not provided with the board. Booting from MMC0 will be used to flash the eMMC in the production environment or can be used by the user to update the SW as needed.

**5.3.7 Boot Modes**

As mentioned earlier, there are four boot modes:

* **eMMC Boot…**This is the default boot mode and will allow for the fastest boot time and will enable the board to boot out of the box using the pre-flashed OS image without having to purchase an micro SD card or an micro SD card writer.
* **SD Boot…**This mode will boot from the micro SD slot. This mode can be used to override what is on the eMMC device and can be used to program the eMMC when used in the manufacturing process or for field updates.
* **Serial Boot…**This mode will use the serial port to allow downloading of the software direct. A separate USB to serial cable is required to use this port.
* **USB Boot…**This mode supports booting over the USB port. A switch is provided to allow switching between the modes.
* Holding the boot switch down during a removal and reapplication of power without a micro SD card inserted will force the boot source to be the USB port and if nothing is detected on the USB client port, it will go to the serial port for download.
* Without holding the switch, the board will boot try to boot from the eMMC. If it is empty, then it will try booting from the micro SD slot, followed by the serial port, and then the USB port.
* If you hold the boot switch down during the removal and reapplication of power to the board, and you have a micro SD card inserted with a bootable image, the board will boot from the micro SD card.

**5.3.8 Power Management**

The **TPS65217C** power management device is used along with a separate LDO to provide power to the system. The **TPS65217C** version provides for the proper voltages required for the DDR3L. This is the same device as used on the original Beagle Bone with the exception of the power rail configuration settings which will be changed in the internal EEPROM to the **TPS65217C** to support the new voltages.

DDR3L requires 1.5V instead of 1.8V on the DDR2 as is the case on the original Beagle Bone. The 1.8V regulator setting has been changed to 1.5V for the DDR3L. The LDO3 3.3V rail has been changed to 1.8V to support those rails on the processor. LDO4 is still 3.3V for the 3.3V rails on the processor. An external **LDOTLV70233** provides the 3.3V rail for the rest of the board.

**5.3.9 PC USB Interface**

The board has a mini USB connector that connects the USB0 port to the processor. This is the same connector as used on the original Beagle Bone.

**5.3.10 Serial Debug Port**

Serial debug is provided via UART0 on the processor via a single 1x6 pin header. In order to use the interface a USB to TTL adapter will be required. The header is compatible with the one provided by FTDI and can be purchased for about $12 to $20 from various sources. Signals supported are TX and RX. None of the handshake signals are supported.

**5.3.11 USB1 Host Port**

On the board is a single USB Type a female connector with full LS/FS/HS Host support that connects to USB1 on the processor. The port can provide power on/off control and up to 500mA of current at 5V. Under USB power, the board will not be able to supply the full 500mA, but should be sufficient to supply enough current for a lower power USB device supplying power between 50 to 100mA.

You can use a wireless keyboard/mouse configuration or you can add a HUB for standard keyboard and mouse interfacing.

**5.3.12Power Sources**

The board can be powered from four different sources:

* A USB port on a PC
* A 5VDC 1A power supply plugged into the DC connector.
* A power supply with a USB connector.
* Expansion connectors

The USB cable is shipped with each board. This port is limited to 500mA by the Power Management IC. It is possible to change the settings in the **TPS65217C** to increase this current, but only after the initial boot. And, at that point the PC most likely will complain, but you can also use a dual connector USB cable to the PC to get to 1A.

The power supply is not provided with the board but can be easily obtained from numerous sources. A 1A supply is sufficient to power the board, but if there is a cape plugged into the board or you have a power hungry device or hub plugged into the host port, then more current may needed from the DC supply.

Power routed to the board via the expansion header could be provided from power derived on a cape. The DC supply should be well regulated and 5V +/-.25V.

**5.3.13 Reset Button**

When pressed and released, causes a reset of the board. The reset button used on the Beagle Bone Black is a little larger than the one used on the original Beagle Bone. It has also been moved out to the edge of the board so that it is more accessible.

**5.3.14 Power Button**

A power button is provided near the reset button close to the Ethernet connector. This button takes advantage of the input to the PMIC for power down features. While a lot of capes have a button, it was decided to add this feature to the board to ensure everyone had access to some new features. These features include:

* Interrupt is sent to the processor to facilitate an orderly shutdown to save files and to un-mount drives.
* Provides ability to let processor put board into a sleep mode to save power.
* Can alert processor to wake up from sleep mode and restore state before sleep was entered.
* Allows board to enter the sleep mode, preserving the RTC clock

If you hold the button down longer than 8 seconds, the board will power off if you release the button when the power LED turns off. If you continue to hold it, the board will power back up completing a power cycle.

**5.3.15 Indicators**

There are a total of five blue LEDs on the board.

* One blue power LED indicates that power is applied and the power management IC is up. If this LED flashes when applying power, it means that an excess current flow was detected and the PMIC has shut down.
* Four blue LEDs that can be controlled via the SW by setting GPIO pins.

**6. Software Design**

**6.1 For these project we mainly using following software**

1. Software install on Ubuntu OS in Raspberry Pi B+ Board

* Python 2.7
* Motion

1. Software used on Windows OS
   * Eclipse, android SDK, android emulator
   * Proteus for Simulation
   * Eagle PCB design.
   * Putty for using SSH protocol to access Ubuntu from Raspberry Pi B+ Board on Windows OS
   * PSFTP for file transfer between Raspberry Pi B+ Board and laptop

**6.1.1 Python**

Python is a widely used [general-purpose](http://en.wikipedia.org/wiki/General-purpose_programming_language), [high-level programming language](http://en.wikipedia.org/wiki/High-level_programming_language). Its design philosophy emphasizes code [readability](http://en.wikipedia.org/wiki/Readability), and its syntax allows programmers to express concepts in fewer [lines of code](http://en.wikipedia.org/wiki/Lines_of_code) than would be possible in languages such as [C](http://en.wikipedia.org/wiki/C_(programming_language)). The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple [programming paradigms](http://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](http://en.wikipedia.org/wiki/Imperative_programming) and [functional programming](http://en.wikipedia.org/wiki/Functional_programming) or [procedure](http://en.wikipedia.org/wiki/Procedural_programming) styles. It features a [dynamic type](http://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](http://en.wikipedia.org/wiki/Memory_management) and has a large and comprehensive [standard library](http://en.wikipedia.org/wiki/Standard_library).

**6.1.2 Java Eclipse**

 Eclipse is a multi-language [Integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) comprising a base [workspace](http://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) system for customizing the environment. It is written mostly in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)).

It can be used develop applications in Java & by means of various plug-in, other programming languages including Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Peal, Groovy, Scheme, and Erlang.

It can also be used to develop packages for the software [Mathematical](http://en.wikipedia.org/wiki/Mathematica). Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

**6.1.3 Putty software**

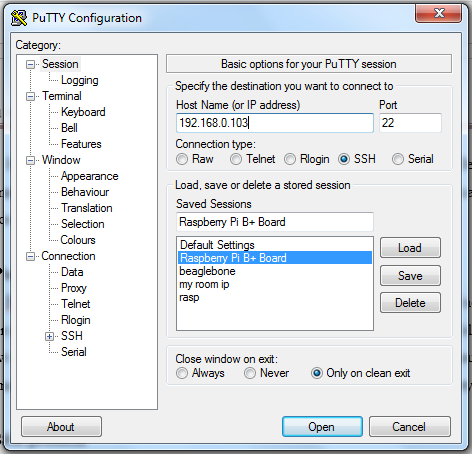
PuTTY is a [free and open-source](http://en.wikipedia.org/wiki/Free_and_open-source) [terminal emulator](http://en.wikipedia.org/wiki/Terminal_emulator), [serial console](http://en.wikipedia.org/wiki/Serial_console) and network file transfer application. It supports several [network protocols](http://en.wikipedia.org/wiki/Network_protocol), such as [SCP](http://en.wikipedia.org/wiki/Secure_copy), [SSH](http://en.wikipedia.org/wiki/Secure_Shell), [Telnet](http://en.wikipedia.org/wiki/Telnet), [rlogin](http://en.wikipedia.org/wiki/Rlogin), and raw socket connection. The name "PuTTY" has no definitive meaning, though "tty" is the name for a terminal in the [UNIX](http://en.wikipedia.org/wiki/Unix) tradition, usually held to be short for [Teletype](http://en.wikipedia.org/wiki/Teletype_Corporation).

**6.1.4 SSH protocol**

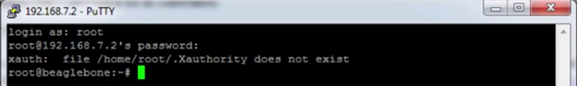
**Secure Shell** (**SSH**) is a cryptographic [network protocol](http://en.wikipedia.org/wiki/Network_protocol) for secure [data communication](http://en.wikipedia.org/wiki/Data_communication), remote [command-line](http://en.wikipedia.org/wiki/Command-line_interface) [login](http://en.wikipedia.org/wiki/Login), remote command execution, and other secure [network services](http://en.wikipedia.org/wiki/Network_service) between two networked computers. It connects, via a [secure channel](http://en.wikipedia.org/wiki/Secure_channel) over an insecure network, a server and a client running [SSH server](http://en.wikipedia.org/wiki/SSH_server) and [SSH client](http://en.wikipedia.org/wiki/SSH_client) programs, respectively. The protocol specification distinguishes between two major versions that are referred to as SSH-1 and SSH-2.

The best-known application of the protocol is for access to [shell accounts](http://en.wikipedia.org/wiki/Shell_account) on [Unix-like](http://en.wikipedia.org/wiki/Unix-like) operating systems, but it can also be used in a similar fashion for accounts on [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows). It was designed as a replacement for [Telnet](http://en.wikipedia.org/wiki/Telnet) and other [insecure](http://en.wikipedia.org/wiki/Computer_security) remote [shell](http://en.wikipedia.org/wiki/Unix_shell) protocols such as the Berkeley [rsh](http://en.wikipedia.org/wiki/Remote_Shell) and [rexec](http://en.wikipedia.org/wiki/Remote_Process_Execution) protocols, which send information, notably [passwords](http://en.wikipedia.org/wiki/Password), in [plaintext](http://en.wikipedia.org/wiki/Plaintext), rendering them susceptible to interception and disclosure using [packet analysis](http://en.wikipedia.org/wiki/Packet_analyzer).[[2]](http://en.wikipedia.org/wiki/Secure_Shell#cite_note-2) The [encryption](http://en.wikipedia.org/wiki/Encryption) used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the [Internet](http://en.wikipedia.org/wiki/Internet).

**6.1.5 BBB log in using Putty and SSH protocol**



**Fig 6.1 putty software window**

* BBB come with static IP address 192.168.7.2
* To login on OS install on BBB we required to enter this static IP when connected through USB cable.
* Connection type is to SSH and port address is 22 by default.

**Fig 6.2 SSH login**

* Then it will ask for user name and password, for Armstrong Linux username and password is “root” and for Ubuntu username is “ubuntu” and password is “temppwd”
* After validating user name and password we can execute any command using SSH protocol, like we do in terminal in case of linux.

**6.2 Android Activity Life cycle**

**Fig: 6.3 Android Activity Life cycle**

**6.3 Flowchart**

**6.3.1 Flowchart for android Program:**

As you connect to the server IP and port do….

End

Print TCP Error

Not Connected

Connect

to server

Connected

Connecting…….

Connect to Server

Socket

Print

Give Server IP address and Server port to socket so Client Server connection gets established.

Get IP address as input to connect sever

Declare require java variables and android variables to access android tools.

Start

Perform action on Android Button press

Up

Send text “Up” to server

Press

If Button

Down

Send text “Down” to server

Press

If Button

After connecting to the IP address and port

Read and show video until the android app get close.

Show live video in surface view created by MjpegView class of java.

Capture live streaming form server by using read MjpegFrame function of MjpegInputStream class of java

Create surface view to show live streaming

**6.3.2 Flowchart for Python Program:**

Connecting…….

Print

Variable: server port, python file path.

If connect to server

Not Connected

Print Error in connection

Print Receiving….

Connected

Start

Declare require variables.

Try to Connect to Server Socket

End

True

True

False

Call GPIO for Down Direction

Call GPIO for up Direction

Else if

if

Down

Up

Receive text from connected server

**6.4 Proteus Motor Driver Simulation:**

****

**Fig: 6.4 Proteus Motor Driver Simulation**

Description:

* This circuit takes input from the Raspberry at input of motor driver. To make isolation between the BBB and motor driver.
* To provide sufficient current to l298 input logic 2n2222 transitor used which is low power high frequency transistor.
* As input of optocoupler changes, input to motor driver changes and motor start rotation according to the logic given.
* Diode are used for protection at the output of motor driver.

**7. PCB Layout**

For PCB designing we used Eagle Software

**7.1 PCB layout of L298 Motor driver**



**Fig 7.2 PCB layout**

**8. Result and Performance Evolution**

**8.1. GPIO programing using python**

To program GPIO of beagle bone we used python language.

**Fig 8.1 BBB GPIO programing**

* In above image LED through resistor are mounted on breadboard. Black wire connected to GND of BBB and Red wire connected to P8 header 11th pin.
* For making pin logic high we have to import BBB python library first. After that we have to decide which pin is to be used as input or output.
* Writing command for making High on logic will glow the led connect to that pin..
* Likewise we can program any pin using python language.

**8.2. Led Controlling on BBB using Android Application**

**Fig 8.2 Android mobile and BBB communication**

* After getting started with python, we execute python file using Java program.
* For each arrow on android application there is separate python file indicating different logics for motor driver to drive motor.
* When up button pressed, up character is send over wifi network using TCP IP protocol, at BBB it detect which character send. After that respected python file executed for short duration. Which make respected GPIO pins HIGH or LOW, which drives logic of motor driver.
* For wifi connectivity LAN cable is connected to the BBB from modem. And mobile is connected through wifi protocols.
* On the terminal window on laptop we can check which buttons is pressed. For that we have to login Ubuntu using putty and SSH protocol.

**8.3. Video transmission from BBB to android**

**Fig 8.3 video streaming from BBB to android mobile**

* For video transmission we connect USB webcam to BBB through USB port. Lan cable is connected to Ethernet port and to the modem.
* For video communication we use motion software on Ubuntu, which takes JPEG images from camera and send them over network.
* On android all images are grabbed and displayed sequentially.
* For supply 5V 2A DC adapter is connected to BBB.

**8.4 Complete video and command communication between Android Mobile and BBB**

**Fig 8.4 video and control final application**

* We have already done video streaming and sending command to BBB individually, it was time to do them simultaneously. Since Ubuntu running on BBB, it was able to run multiple program simultaneously.
* We made application on android which combined the code of individual task.
* To make BBB to communicate wirelessly USB adapter connected through USB hub to the BBB, which allows BBB to connect with wifi network of mobile.
* On the same USB hub web cam is connected which used for video streaming.
* Video of 320x240 resolution stream continuously with simultaneously command transfer.
* Since both adapter and camera are connected through usb it is necessary to provide more current to the board, which in above image given by DC adapter.

**9. Applications and Future Scope**

* 1. **Advantage**

1. Streaming video use less memory and instantaneous launching as compared to download and watch player.
2. It is standalone system.
3. It can modified so that it acts as a human machine interface for any system.
4. It is cost effective project.
5. It is user friendly anyone can access it.
6. It is compatible with all latest android OS.
7. It consumes less power.

**9.2 Applications:**

1. [Sky Drone FPV: Long Range Digital HD Video FPV Solution](http://beagleboard.org/project/Sky+Drone+FPV/).
2. This project is used for surveillance application.
3. It mainly used for general spying robot in any area where mobile internet works.
4. It can be used as security system in domestic or in bank where one can observe assets via android application using live video streaming.
5. In this system camera can track the particular things. Like one can track person face such that person will not be out from the streaming frame.
6. We can use microcontroller on robot side to control the robotic arm so that it can act as bomb diffusing robots.
7. It can be used as attacking bot on war field controlled wirelessly through internet.
8. In future we can process on the incoming video frames and do the particular task.

**9.3 Future Scope:**

There were some areas we felt we did not address

1. Interfacing of temperature sensor to give the temperature update of robot side area.
2. We can increase the resolution and frames per seconds to get High Quality video.
3. We can improve the performance by improving battery backup and making it autonomous if client connection lost.
4. Using server programming we can access the system from anywhere in the world through internet.
5. Using GPS module we track the Robot on Google map.
6. Using different i2c or SPI protocol we can interface it with microcontroller to make system compatible with all TTL logics as well as sensors.

**10. Conclusion:**

We can control any standalone system at remote side using the embedded hardware and software installed there using internet protocols.

It is a real time portable system, since we are using internet connectivity on android mobile. The application running on android mobile is user friendly.

We can survey the particular area using the video streaming principle and can be used for spying purposes.

This project can work in the area where the signal strength of respected mobile operator is sufficient. The cost of video streaming will be cost of internet plan provided by that mobile operator.

Android application is generally compatible with the android OS 4.1.2 and can be designed for the lower version of Android OS.

**11. Cost of material**

|  |  |  |  |
| --- | --- | --- | --- |
| Component Used | Quantity | Rate | Cost |
| Beaglebone black processor  Sitara XAM3359AZCZ | 1 | 4200/- | 4200/- |
| Web camera | 1 | 800/- | 800/- |
| Wi-fi module | 1 | 500/- | 500/- |
| Battery 12V ,1.2A | 2 | 500/- | 1000/- |
| USB hub | 1 | 100/- | 100/- |
| Acryolic plate | 1 | 80/- | 80/- |
| PCB layout | 1 | 180/- | 180/- |
| Power connector(motor) | 2 | 20/- | 20/- |
| Gear motor | 2 | 150/- | 300/- |
| Adapter connector | 1 | 15/- | 15/- |
| IC L298 | 1 | 150/- | 150/- |
| IC 7805 | 1 | 15/- | 15/- |
| Male connector strip 40 pins | 1 | 10/- | 10/- |
| 2N2222A transistor | 4 | 3.50/- | 15/- |

|  |  |  |  |
| --- | --- | --- | --- |
| Component Used | Quantity | Rate | Cost |
| MCT2E | 4 | 3.50/- | 15/- |
| Wheel | 2 | 25/- | 50/- |
| 3mm Screw and Nut | 15 | 5/- | 80/- |
| Caster | 1 | 30/- | 30/- |
| 3mm LED | 4 | 3.50/- | 15/- |
| Resistor- 10k | 15 | 1/- | 3/- |
| Resistor- 68 ohm | 4 | 1/- | 1/- |
| Total Cost | | | 7579/- |

**12. References**

[1] Waldherr, S., Thrun, S., and Romero, R., “A Gesture based interface for Human-Robot Interaction”, Kluwer Academic Publishers, Netherland, 2000.

[2] Liu, T., Guo, H., and Wang, Y., “A new approach for color-based object recognition with fusion of color models”, Congress on Image and Signal Processing Conference, Sanya-China, vol. 3, pp. 456-460, May 2008.

[3] Wang, B., and Yuan, T., “Traffic Police Gesture Recognition using Accelerometer”, IEEE SENSORS Conference, Lecce-Italy,pp. 1080-1083,

Oct. 2008.

[4] Lalanne, T., and Lempereur, C., “Color recognition with a camera: a supervised algorithm for classification”, IEEE Southwest Symposium on

Image Analysis and Interpretation, Tucson-Arizona, pp. 198- 204, April 1998.

[5] Available: http://en.wikipedia.org/wiki/Gesture\_recognition.

[6] Cannan, J. and Hu, H., “Human-Machine Interaction (HMI): A Survey”.[Online]. Available: http://www.

dces.essex.ac.uk/staff/hhu/Papers/CES-508%20HMI-Survey.pdf

[7] Das, S., Toya, L., Green, Perez, B., and Murphy, M. M., “Detecting User Activities using the Accelerometer on the Smartphone”, Team for Research in Ubiquitous Secure Technology REU Research Program, July 2010.

[8] Aroca, R. V., AntônioPéricles B. S. de Oliveira, and Gonçalves, L. M. G., “Towards Smarter Robots With Smartphone”, 5th workshop in Applied Robotics and Automation, Bauru-Brazil, June 2012.

[9] Song, M., Kim, B., Ryu, Y., Kim, Y., and Kim, S., “A design of real time control robot system using android Smartphone” The 7th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Busan- Korea, Nov. 2010.

[10] Shilpa, V., Pradeep, H. S., and Kurian, M. Z., “The Symbian Robot,” International Journal of Computer Science and Informatics (IJCSI), Vol-1, Issue-3.

[11] M. Klingmann, “Accelerometer-Based Gesture Recognition with the iPhone”, Goldsmiths University, MSc in Cognitive Computing, London, September 2009.

[12] Global, S., Jubeh, R., Raesch, S. L., and Z¨undorf A., “Using the Android Platform to control Robots”, Kassel University Germany.[Online].

Available: www.innoc.at/fileadmin/user\_upload/\_temp\_/RiE/.../65.pdf

[13] Available: http://en.wikipedia.org/wiki/Android\_(operating\_system).

[14] Do, H. M., Mouser, C. J., and Sheng, W., “Building a Telepresence Robot Based on an open source”, ASCC Lab, Oklahoma State University

USA, Dec. 2011.

[15] Nasereddin, H. H. O., and Abdelkarim, A. A., “Smartphone Control Robots Through Bluetooth”, International Journal of Research and Reviewsin Applied Sciences (IJRRAS), Vol-4, Issue-4, Sept, 2010.

[16] International Journal of Software and Web Sciences (IJSWS) [www.iasir.net](http://www.iasir.net)on ANDROID CONTROLLED SPY-ROBOT Prof. Y. M. Naik, Chiranjivi. M. Deshpande, Ravija.R. Shah, Rashmi.R. Kulkarni Department of Computer Science and Engineering, Hirasugar Institute of Technology Nidasoshi.

[17] Kanjo E., Bacon J., Roberts D., and Landshoff P., "MobSens: Making Smart Phones Smarter," IEEE Pervasive Computing, vol. 8, no. 4, pp. 50-57, Oct 2009.

[18]Shaer O., Leland N., Calvillo-Gamez E. H., and Jacob R. J., "The TAC paradigm: Specifying tangible user interfaces," Personal Ubiquitous Computers”., vol. 8, no. 5, pp.359-369, Sep. 2010.

Website Referred:

1. www.circuitco.com/support/beagleboneblack
2. www.beagleboard.org/Products/BeagleBone%20Black‎
3. www.beagleboard.org/Getting%20Started‎
4. [www.developer.android.com/training/basics/firstapp](http://www.developer.android.com/training/basics/firstapp)

**13. Data Sheet**