# T-100 Watchdog

Ismael Rivera, Journey Sumlar, Chris Carmichael, and Warayut Techarut

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — In an effort to combine multiple elements of both Electrical and Computer engineering we developed a home security robot, the Watchdog system is designed to observe an entry point into the users home and report to the user when an intruder enters their home. Upon this alert the user is able to take control of the robot through the mobile app, which is streaming a live feed from its onboard camera, giving the user the ability to maneuver through their house and observe whether there is an intruder so the proper authorities can be alerted.**

***Index Terms* — Motion Tracking, Wireless Communication, Motor Controller, Embedded System, Mecanum Wheels, Video Streaming.**

## I. Introduction

Group 4 of Spring/Summer 2014 wanted to apply the skills learned throughout our four years of study and apply them to our Senior Design Project. Since our team consists of half Computer Engineers and half Electrical Engineers, we wanted to use our senior design project as a chance to combine all of our experience and knowledge in these two related fields while pursuing new topics of interest for each team member. In order to accommodate all the skills acquired, we decided to use a robotic vehicle as the base of our project. The T-100 Watchdog is an autonomous robot whose main function is to detect any moving target, alert the user and allow the user to control the vehicle. The robot will be equipped with a webcam which will be used to capture frames and detect any movement. For better maneuverability, Mecanum Wheels will be utilized allowing the robot to change course and direction. A wireless subsystem will also be utilized which will give the ability to stream a live video feed as well as the ability to control the robot wirelessly via the mobile application. The Android mobile application will allow the user to view the live video feed, store snapshots, and control the robot remotely. The main board running all these tasks simultaneously will be a Beagle Bone Black with Node.js as the webserver of choice. This document will describe the research and decisions made to complete the project.

## II. Specifications

The design specifications of the T-100 Watchdog are as follows:

1) Dimension 12.5” x 9”
2) Operating Range 25 ft
3) Power Supply Voltage 15 v
4) Maximum Weight 20 lbs
5) Minimum Acceleration 1 ft/s2
6) Maximum Speed 3 ft/s
7) Battery Life ~3 hrs
8) Recharge Time < 8 hrs
9) Camera Resolution 640 x 480
10) Video Stream of 8-10 fps
11) Detect motion within 2-3s
12) Remote Controlled Ability

## III. System Components and Design

### *A. System Overview*

The T-100 Watchdog contains one main processing unit, the BeagleBone Black (BBB). The BBB will run the different subsystems required for the robot to work properly. In addition, it will communicate with the motor controller in order to update its position. A high-level design of the subsystems can be seen below in Figure 1.

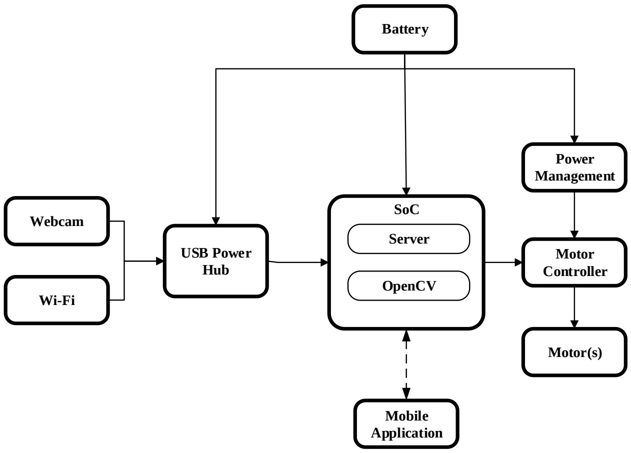


Figure 1 – T-100 Watchdog Subsystems

### *B. Beagle Bone Black*

The BeagleBone Black (BBB) Revision B is a low cost development board with a 1 GHz AM335x ARM® Cortext-A8 processor. It has 512MB DDR3 RAM, 2GB of on-board flash storage, microSD card slot, and a 3D graphics accelerator. For connectivity, it has a USB client for power and communication as well as a USB host. Since multiple devices will be connected to the BBB, a powered USB hub will be utilized to connect all the devices as well as provide the necessary power. In addition, it features a 10/100 Mbps Ethernet port, video-out HDMI, and two 46 pin headers. The BBB is capable of running various versions of Linux distributions that support the arm architecture. In order to prevent compatibility issues, Debian was utilized for this project since it's now the default OS being shipped with the BBB.

### *C. Wireless Communication*

In order to connect the BBB online and enable wireless communication a USB Wi-Fi Adapter was utilized. The Edimax EW-7811UN is a nano USB wireless adapter compatible with the BBB. It supports the 802.11b/g/n standards with data rates up to 150Mbps. For secured Wi-Fi connections, it supports 64/128-bit WEP, WPA, WPA2 Encryption and is WPS-compatible. The small size and low power consumption allows the BBB to be always connected to the Internet via wireless.

### *D. Webcam Camera*

Since the BBB is running Debian, the webcam to be utilized had to be compatible with the Linux UVC driver. In addition, the ability to support multiple picture resolutions was also made a requirement in order to test the performance. The Webcam camera of choice was the Microsoft VX-5000. It has a USB 2.0 connection, multiple resolutions are supported, and pictures in low light conditions are acceptable. In addition, the Microsoft VX-5000 features a bendable rubber stand which gives the ability to try different positions on the frame of the vehicle. The cameras dimensions are 45 mm x 45 mm x 102 mm. Since the camera is compatible with the Linux UVC driver, almost all of the commands in Linux used to capture images/videos from a camera work properly.

### *F. Video Streaming*

The video stream subsystem enables the viewer to see a live video stream of what the vehicle is seeing in real time. At any given time the user can connect using a web browser or the mobile application to view the video stream. The program being utilized to enable this feature is the MJPG-streamer, which captures JPGs from either a directory in the microSD card or straight from a camera connected to the BBB. In addition, the resolution can be set at different values allowing for greater frames per second video stream. Once the input has been set, the stream is sent as M-JPEG via HTTP where any web browser can view the stream. In addition, the user can use the Android application which also gives the ability to view the stream.

### *G. Computer Vision*

The computer vision subsystem is composed of two main parts, motion detection and tracking. In order to detect motion a simple background subtraction (pixel values) is performed every 500 milliseconds. Each subtraction's value is stored to compare it with a threshold. This allows us to prevent false positives. For the tracking, once motion has been detected the location of the pixels that triggered the motion detector are determined and then sent to the motor controller to move the vehicle in the direction based on the location of the pixels.

### *H. Mobile Application*

The mobile application was created for Android devices due to the group's familiarity and experience with the JAVA programming language. The IDE used was Eclipse along with the Android ADT/SDK plug-in. The application is made to be compatible with devices running Android 4.0 Ice Cream Sandwich and above, not including tablets. The mobile application will be composed of four activities: Registration, Login, Control, and Gallery. Every time the application is opened, the user will be shown the Login activity. From here, the user will have the choice to register for an account if one doesn't exist or login to an existing account. If the option to register for an account is selected, the user will be directed to the Registration activity. In the Registration activity, the user will fill some fields such as: username, password, first name, last name, cell phone number and provider, and finally email. The cell phone number and email is used to alert the user once any motion has been detected. A text/mms will be sent to the user via an email. The cellphone providers being supported are: AT&T, Sprint, T-Mobile, Verizon, MetroPCS, Boost Mobile, and Straight Talk. Depending on the user's selection a Short Message Service (SMS) and Multimedia Messaging Service (MMS) address will be created. Once all the fields have been filled and the user select the register button, the email, SMS, and MMS address are sent to the server via a PHP POST & GET method and is then stored in the server. Once the user has an account, the user can go ahead and login to the Control Activity. In the Control activity the user can view the live video stream, save snapshots to the device, and control the vehicle by using the digital on-screen joystick. Every snapshot that the user takes can be seen in the gallery activity.

### *I. Motors*

The motors that we have decided to use is the VEX 2-Wire Motor 393. This motor comes with an adjustable gear ratio to provide both a low speed and high speed mode. In the high speed mode it can reach 160 RPM. However, the torque drops to just under the 1.3Nm that we calculated as a requirement. This is an acceptable compromise as the result of our calculation was assuming a higher end range for the vehicles weight.

### *J. Motor Controller*

To save processing power on the main CPU we have decided to create our own motor controllers, which are capable of generating their own PWM signal for speed control, as a result the BeagleBone Black only has to send a simple analog signal rather than creating separate PWM signals for each of the motors. The controllers use an MSP430 to generate the PWM signal to control the speed of our motors. The MSP430 receives two signals from the processor, an analog signal which will be used to alter the duty cycle of the PWM being generated, and a digital signal to convey the desired rotation direction. The MSP430 will then alter the PWM that it is generating and send it along with two control signals to the H-Bridge. The two control signals will determine which of the transistors in the H-Bridge will allow current to flow through them, given certain combination of signals the current will be allowed to flow through the motor in one direction, and given a different combination the current will flow in the opposite direction. The PWM will be applied to a separate transistor which will act as an enable line for the circuit. With the Duty cycle of the PWM determining the duration that the current is allowed to flow during each cycle, this is how we can control the speed of the motor below in Figure 2.

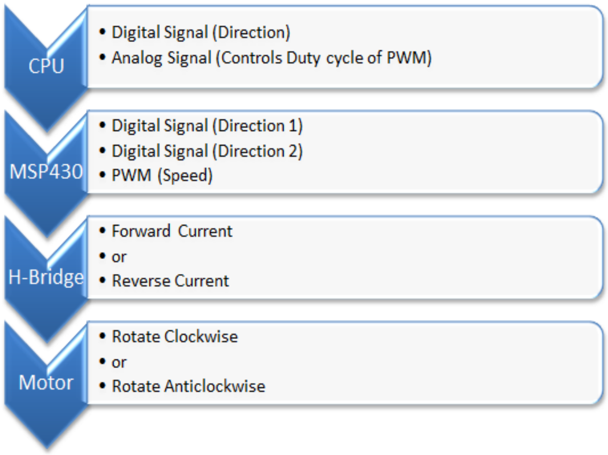

CPU
• Digital Signal (Direction)
• Analog Signal (Controls Duty cycle of PWM)
MSP430
• Digital Signal (Direction 1)
• Digital Signal (Direction 2)
• PWM (Speed)
H-Bridge
• Forward Current
• or
• Reverse Current
Motor
• Rotate Clockwise
• or
• Rotate Anticlockwise

Figure 2

### *K. Power*

The T-100 Watchdog will rely off of two different Power Supplies, a 14.8v 9600mAh Lithium Ion battery and a 7.4V Lithium Ion 6600mAh battery. The voltages from each of the batteries are regulated to power each of the different sub-components for the T-100 Watchdog. The system's requirements are as follows:

1) BeagleBone Black at ~5V 2A

2) USB Power Hub is at ~ 5V 1A

3) Motor Controllers has two inputs at ~10V 16A (4A per motor) and 3.3 V at 1.5A.

In order to fully regulate each these sub components we used the 14.8V lithium ion battery for the four motor controllers and the 7.4 V lithium ion batteries for the USB Power Hub and BeagleBone Black. This setup was used since the motor controller was pulling the most current out of all the different sub components and will expand the life of our battery by doing so.

USB Power Hub is used to provide extra USB connections for the BeagleBone Black. Even though the USB Power Hub can be powered from the BeagleBone, we chose to power the Power Hub separately from the BeagleBone Black because it is used to power many different components. The Webcam and Wi-Fi adapter are all connected to the USB Hub. The BeagleBone Black requires a steady input of 2A in order for the USB host to work properly. As a result the BeagleBone Black will be pulling current constantly.

In order to fully regulate these voltages two LM338 voltage regulators and one LM 350 regulators were utilized. The LM338 voltage regulators produce a maximum of 5A and the LM350 produces a maximum of 3A for loads. To extend the battery life of the batteries, a switch was utilized. The switch is a 12VDC 30A rocket switch. This switch was chosen because it will be able to withstand the current coming from the battery to power the system. The power system will be able to be fully functional for 3 hours.

### L. Wheels

Every single wheel has to be working together flawlessly in order to perform a precise movement. Keep in mind that the microcontroller executes instructions very fast but it still does it one instruction at a time. All four motors need to be orchestrated seamlessly. To move forward, all wheels must move forward. In order to go backward, all wheels must be going in the backward direction. Side to Side Translation is just using the different combination of wheels to cancel out the resultant force in the Y-direction.

As long as the correct pair is picked and the magnitude of the rotation is maintained, the movement should be translating at a constant rate. A diagram depicting the wheel movement is shown below in Figure 3.

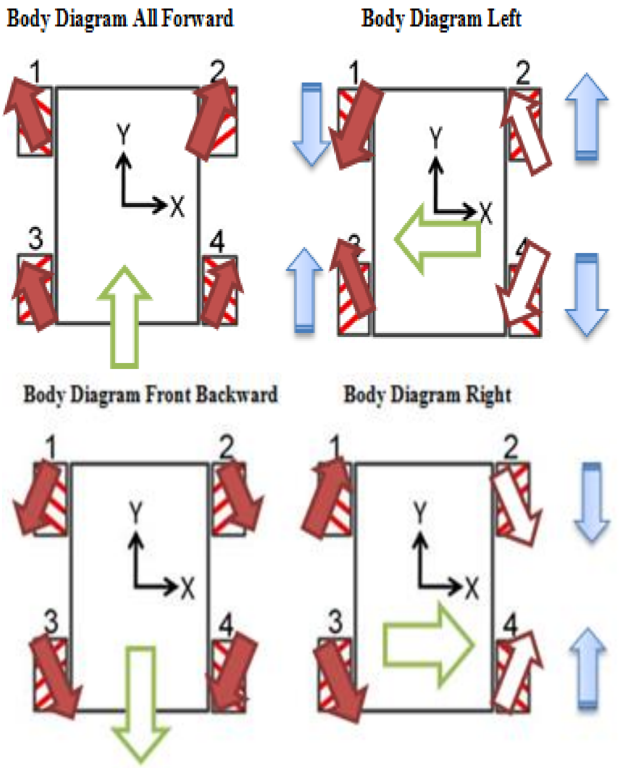


Figure 3

## IV. Software Detail

The main software components of the T-100 Watchdog are the BeagleBone Black (BBB) and the motor controller which runs on four MSP430's. The BBB runs all the high level software such as motion detection, tracking, video streaming, User Datagram Protocol (UDP) and web server. The BBB communicates via Pulse Width Modulation (PWM) ports to alert the motor controller in what direction to move. Each wheel has its own motor and each requires to be controlled separately. The MSP430 was the microcontroller of choice to control each motor/wheel. Code Composer studio was utilized to program each chip to enable the PWM and receive a signal which will trigger the motor. The BBB is running Debian 7.5 Wheezy along with Java 1.7 Soft-Float version. The motion detection and tracking was programmed in JAVA to make use of open-source API's. The UDP server, which handles the communication between Android device and BBB, was also programmed in JAVA to make the integration between the subsystems easier and prevent any compatibility issues during testing. The mobile application intended for Android devices is also programmed in JAVA along with the Android SDK. For the web server, Node.js was utilized because of its lightweight and efficiency.

### A. Computer Vision

The computer vision has two subsystems, motion detection and tracking. Each time a frame is taken from the camera, before it's analyzed, it is stored in the microSD card of the BBB. Every 500 milliseconds the program will check for any motion occurred. This is done by comparing the previous frame with the new frame and doing a background subtraction. The result of the subtraction is then stored and compared with the set threshold. If the result of the subtraction is higher than the threshold, meaning a big change has occurred, motion has been detected. As soon as motion is detected, a bash script is called within the JAVA program to alert the user via SMS, MMS, and Email. If the result is less than the threshold, then it's a false positive. The reason why the result is compared with the threshold value is to prevent running resource intensive algorithms which will make the BBB go into a higher processing power state, as a result utilizing more power. If the motion detected ends up being a positive one, the tracking subsystem gets activated. For the tracking algorithm, a similar method is utilized as the motion detection but this time the location of the pixels that have changed is found, providing with an 'x' and 'y' value which represents the location on the image. Depending on the resolution of the image being utilized, a specific 'x' pixel value will determine in what direction the vehicle moves. For example, if the image resolution being utilized is 640 x 480, the image is divided into three sections as shown below in Figure 4.

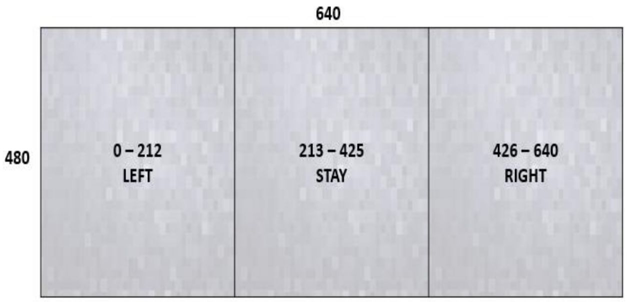


Figure 4

As can be seen from Figure 4, if the tracking determines that the target's pixel value is between 0 and 212 a signal will be sent to the motor controller for the vehicle to move left and the same will happen in the other two sections. Once the direction to where the vehicle is going to be moved has been determined a bash script is executed to communicate with the motor controller which will activate the required motors to move the vehicle in the determined direction. An overview of the computer vision subsystems are shown below in Figure 5.

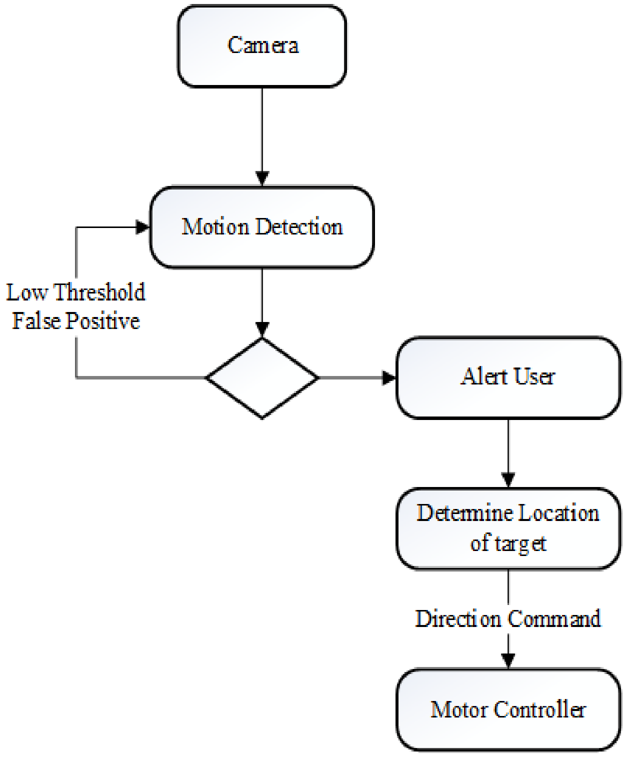


Figure 5

*B. Video Stream*

The MJPG-Streamer software contains an input plugin and an output plugin. The input plugin can be set to receive frames straight from a video camera or read frames from a specific folder location. The input plugin is set to read frames from directly from the webcam connected via USB to the BeagleBone Black. In addition, depending on the Internet connection available, the resolution of the image can be changed in order to improve performance if required. The output plugin's default option is to stream and provide a snapshot of the current frame via the Internet, with each option having its own URL address. A port has to be set, which is where the stream and snapshot will be available. In order to make the stream compatible with all major known browsers, the stream and snapshot will be embedded in a HTML <img> tag forcing the web browser to treat it as a regular image. An overview of the video streaming subsystem is shown below in Figure 6.

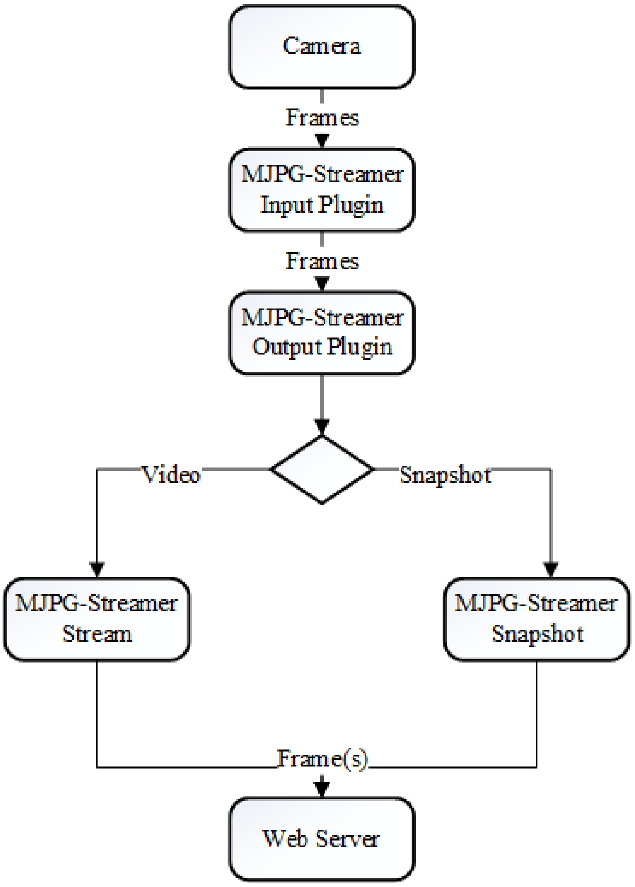


Figure 6

*C. Web Server*

For the web server, we had a choice of a local-webserver or a cloud-based solution such as the Google App Engine. An advantage of a cloud-based solution is that once the setup is completed, a static address is provided along with an API, making it easier to integrate. After careful consideration, the team decided to go with a local webserver running in the BBB instead. A scenario that supports our decision is that by having a cloud-based webserver, the data transmission would have to travel to an extra channel, thus increasing the amount of wait time and response as shown in Figure 7 below.

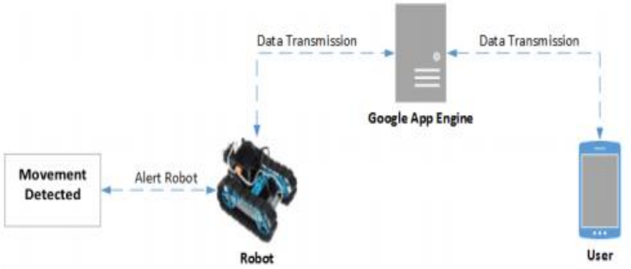


Figure 7

Since we would to be as close to real-time as possible, by having the web server run on the BBB we can remove one channel and at the same time have more control on how the data is handled. A disadvantage is that there is no API meaning the integration between the subsystems will take longer and now we have to handle IP/Address changes on our own. In order to solve the changes of the IP address, a free DNS provider will be used. By using a DNS provider, we can use an address similar to a website's address instead of memorizing the IP address. No-IP was the client of choice due to the simplicity of the setup, compatibility with the operating system, and low amount of resources required. Every time the ip changes on the BBB, the No-Ip client will update the address to point to the new IP address, this way a static address can be used in all the subsystems programs and scripts.

For the local based web server there were a few options that were considered such as Apache, OpenLiteSpeed, Nginx, and Node.js. The most memory intensive subsystem of all is the Computer Vision subsystem. As a result we needed a web server that would be easy to implement but at the same time require a very small amount of resources. In order to decide which one to use, a series of benchmarks were performed to test the amount of CPU usage and RAM usage while at idle and processing data. In the CPU benchmark, all the web servers utilized about the same amount. For the RAM benchmark, the results were very different as can be seen below in Figure 8.

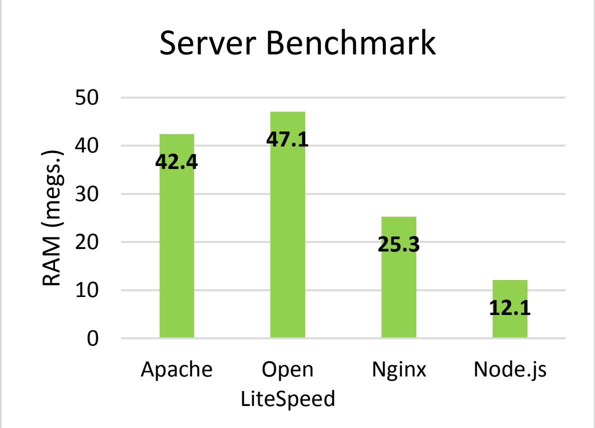


Figure 8

Node.js was the web server that utilized the least amount of RAM and is the one that is being utilized. Node.js is event driven and asynchronous. An advantage of Node.js is that it can be customized to only run specific tasks without the need of installing all the packages.

*D. Mobile Application*

The mobile application is how the user can watch the video stream on their mobile device at any time. The application will be composed of four Activities: Login, Register, Control, and Gallery as it can seen in Figure 9 below.

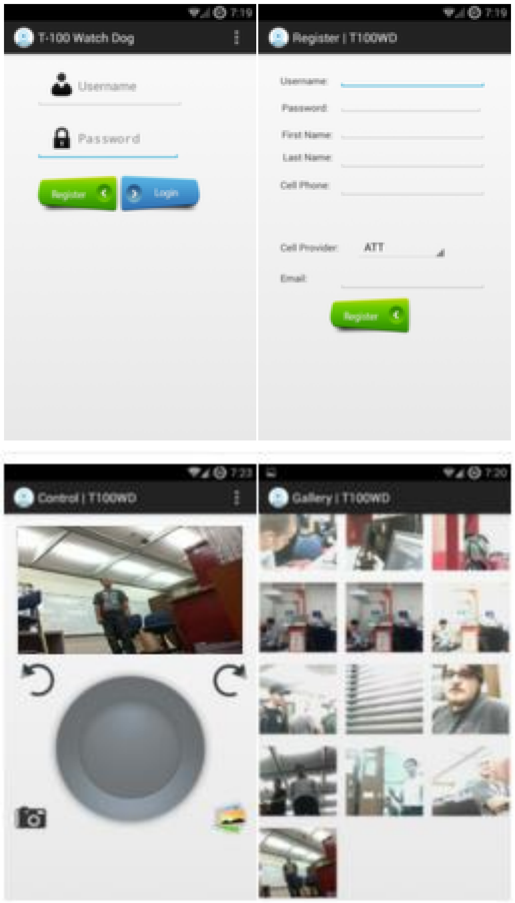


Figure 9

If the user doesn't have an account, one can be created by selecting the Register button in the Login activity. It will be assumed that there will only be one user stored in the database. Once the user has an account, the Login activity will be shown to the user to login and get access to the Control activity. The control activity contains a WebView which shows the live video stream and below the video streams a joystick where the user can control the vehicle. The commands will be sent via a UDP connection to the web server via the Internet. In addition, the user has the option to save a specific frame from the video stream. This is made possible thanks to the MJPG-Streamer output plugin which has a built-in function to only show a single snapshot. The user can also view all the snapshots stored in the device by selecting the gallery button within the application. From the Gallery activity the user has the ability to share any picture via MMS/Email by selecting it.

The remote control ability is achieved by utilizing a UDP Server-Client connection. The mobile application has the UDP client which received the direction of the virtual joystick and sends the command via the internet to the UDP Server. The UDP Server is running on the BeagleBone Black and is always listening for any incoming messages from the UDP client. Once a message is received, the command is parsed and sent to the motor controller. Depending on the command received, the required motors will be turned on to follow the command. A diagram depicting how the remote control works is shown below in Figure 10.

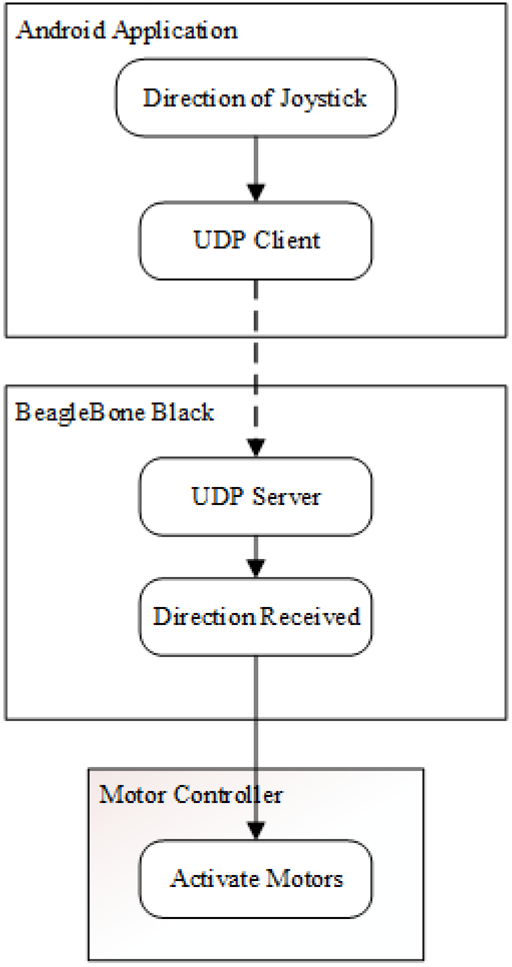


Figure 10

*E. Motor Controller*

The H-Bridge in the circuit was made using MOS-FET Transistors because they have lower losses than their BJT counterparts. The Bridge is then controlled by a series of resistors and BJT transistors which form a level shifting AND gate to incorporate the Direction signal and the PWM which are both generated by the MSP 430G2542 microcontroller. The schematic for the motor controller is shown in Figure 11.

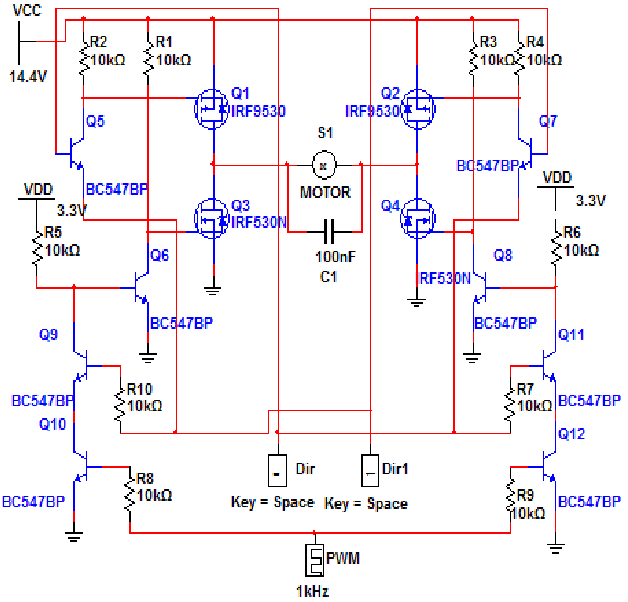


Figure 11

The specifications for the transistors used in the motor controllers are shown below in Table 1. This allows each controller to handle up to 100VDS and up to 12A which is more than enough for our project which will require the motors to run at up to 10V with less than 2A each.

| Transistors | VDS | Current |
|---|---|---|
| IRF 9530 P-Channel MOS-FET | 100V | 12A |
| IRF 530 N-Channel MOS-FET | 100V | 14A |
| BC547 NPN BJT | 47V | 100mA |

Table 1

## V. PCB

The PCB for the motor controllers was designed using Eagle Cad and was manufactured on two layer boards by OSH Park. Each controller measures 3.19x2.76 in and features thicker trace lines connecting the H-Bridge to the motor and the battery to support the current that will be pulled along that path, which can be seen in Figure 12.

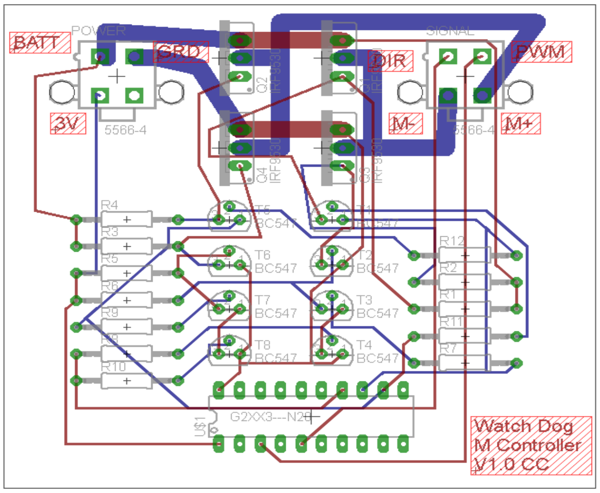


Figure 12

The Motor controller boards are connected to the BeagleBone Black development board via the DIR and PWM connectors, these inputs are relayed directly to the MSP430 on the motor controller and are used to adjust which MOS Transistors in the H-Bridge are active and adjust the duty cycle of the PWM being generated to control the speed of the motors. The other connections to the PCB are the 10V from the battery, which is the main driving voltage for the motors, 3 volts, to power the MSP430, the output connections for the Motors, and the ground line which gives the circuit a common ground with the other components of the project.

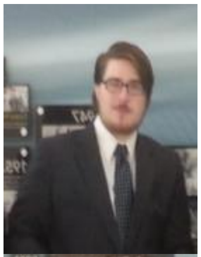
## VI. Conclusion

The general goal for this robot was system integration. With members consisting of both electrical and computer engineers, the result was a project that involved components from the ground level such as motor controllers to high level computer language to govern and synchronize the entire system. Modern technology has a much higher rate of development. A single invention can involve skills and knowledge from multiple disciplines. Our project offers exactly that for us. The robot involves different expertise to complete its internal system. The development process requires extensive teamwork and substantial collaboration. A carefully thought out schedule must be planned for every development phase, in order to meet the tight deadline. Upon completion of this project, we are able to take our idea and transform it into a tangible result.

## VII. Acknowledgement

Group 4 of the Spring 2014 – Summer 2014 Senior Design class would like to acknowledge and thank the Department of Electrical Engineering and Computer Science at the University of Central Florida, Boeing, and Dr. Richie for the help and advice.

## VIII. The Team



Chris Carmichael - is a senior electrical engineering student at UCF. His areas of interests include semiconductor micro and nano systems. He is planning on pursuing his MS in this field in the Fall of 2015.

Ismael Rivera is a graduating computer engineer from the University of Central Florida. He will start working with Lockheed Martin after graduation. His interests include technology, cars, and nature.

Journey Sumlar is a graduating senior from the University of Central Florida with a Bachelor's of Science degree in Electrical Engineering. He hopes for a career in power engineering.

Warayut Techarut is a senior majoring in Computer Engineering. He is currently with Siemens Wind Power and planning to return to attend graduate school.

## IX. References

[1] MJPG-Streamer
< http://sourceforge.net/projects/mjpg-streamer/>

[2] Eclipse IDE/Android SDK
< http://developer.android.com/sdk/index.html>

[3] Node.js
< http://nodejs.org/>

[4] No-IP
< http://www.noip.com/>

[5] BeagleBone Black (BBB)
< http://www. beagleboard.org>