Spring 2014

The Goddard Project

Cassandra DeNunzio
University of New Hampshire - Main Campus

Robert Galli
University of New Hampshire - Manchester

Cameron Perl
University of New Hampshire - Main Campus

Andrew Felicetti
University of New Hampshire - Main Campus

Samuel Cordeiro
University of New Hampshire - Main Campus

See next page for additional authors

Author
Cassandra DeNunzio, Robert Galli, Cameron Perl, Andrew Felicetti, Samuel Cordeiro, and Kevin Tierney

ECE 791/792 Final Report

Project Title: The Goddard Project

Team Members: Robert Galli, Cassandra DeNunzio, Samuel Cordeiro,
Kevin Tierney, Cameron Perl, Andrew Felicetti
Faculty Advisor: Dr. Thomas Miller

Courses Involved: CS 410, ECE 651, ECE 562, ECE 617, ECE 618, ECE 548, ECE 714, ECE 757

Project Completion Date: April, 2014

<u>Mission:</u>

The mission of The Goddard Project was to research, design, and experiment with electronics, mechanics, software design and artificial intelligence to produce a clever, fun, and loveable companion.

Table of Contents:

Synopsis:

The goal of this project was to design and build a robotic dog modeled after Goddard from the 2001 television series, *The Adventures of Jimmy Neutron Boy Genius*. Sheet metal was used for the body of the robot in order to keep the frame lightweight. Goddard travels on two wheels powered by electric motors and two caster wheels. The robot utilizes a Raspberry Pi as its master device and an Arduino Uno in order to control the robot. Essentially, Goddard acts and behaves like an ordinary dog from barking, to moving around, to being a great companion and friend. In addition, Goddard also has access to the extensive knowledge of the Wolfram Alpha database and can be controlled over a wireless internet connection using a keyboard or voice controls. This project was successfully presented and demonstrated at the 2014 University of New Hampshire Undergraduate Research Conference and received an Award of Excellence.

Project Comparison:

Figure 1. Comparison of real-life Goddard built by the Goddard Project team a cartoon Goddard from *The Adventures of Jimmy Neutron Boy Genius*

Implementation and Testing:

In order to meet our goals for this project, our team needed to develop a comprehensive plan for implementation. We met weekly as a team since the beginning of September 2013 to the end of April 2014. We felt that it was important with a large team of six members to meet frequently and discuss how our individual portions of the project would work together to form a greater whole. We took notes and minutes at every meeting and uploaded the documents to SkyDrive as well as pertinent research for the whole team to reference. This process concentrated the information related to the project for anyone on the team to access whenever it was needed. On top of weekly group meetings, sub-teams also met weekly and each member worked on his or her own time to accomplish a distinctive functionality of the robot.

Microcontroller Overview:

The Raspberry Pi was chosen because it is a mult-purpose linux computer. It was chosen because it can run a full fledged web server capable of communicating to the internet at large (Node.js). Node.js was helpful for performing speech recognition, querying the Wolfram database and performing text to speech. Additionally, having the Raspberry Pi run a Node.js server allowed us to build our user-interface with HTML5, CSS and Javascript which allowed our client-side robot controller to be platform independent. The Raspberry Pi was powered via a USB Hub (PiHub). Two connections were made from the USB Hub to the Pi; one connection to a high current port on the PiHub to the Raspberry Pi provided power whereas the second connection was used for serial data communication with the Arduino Uno. This was achieved by connecting a USB B cable from the PiHub to the Arduino which provided power and a means to exchange serial data.

The Arduino Uno was used because it abstracted much of the hardware control. Additionally, there were many open source libraries available for the Arduino as well as a large support community to help us if and when we ran into trouble. The Arduino allowed us to easily program servo motors for tail wagging, neck movement and head movement. The Arduino Uno was also integrated with Goddard's sensor network. Some other advantages of the Arduino Uno include many input/output pins, low power consumption, and compact nature. Figures 2 and 3 display our choice of microcontrollers with each of their capabilities labeled.

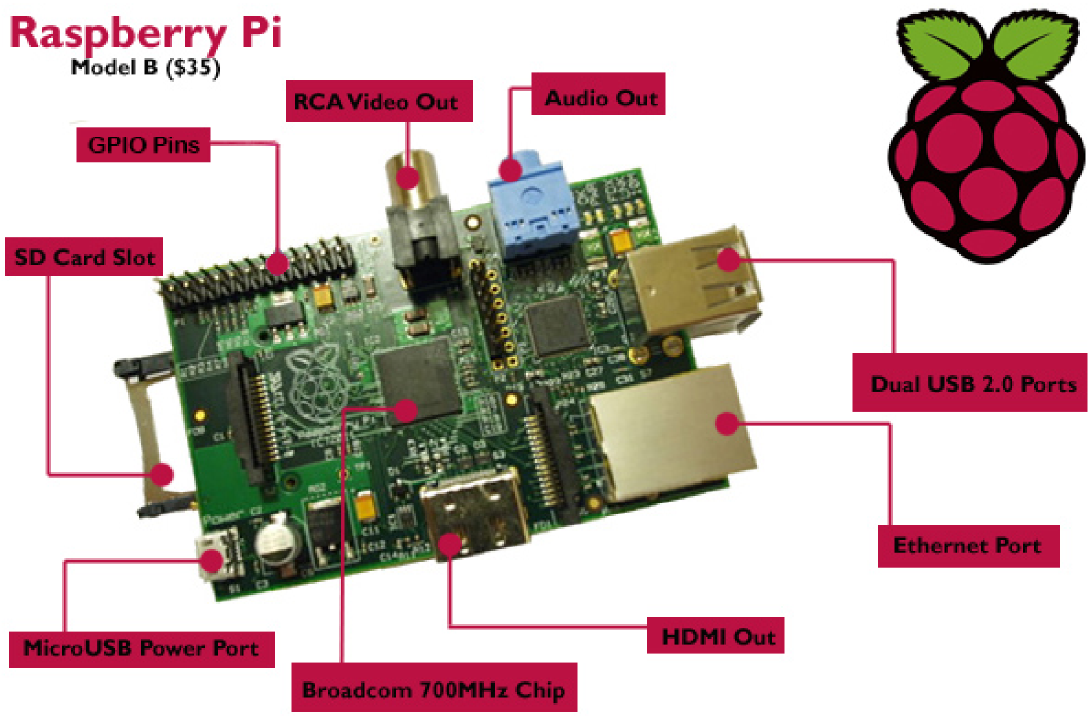


Figure 2. Raspberry Pi Model B

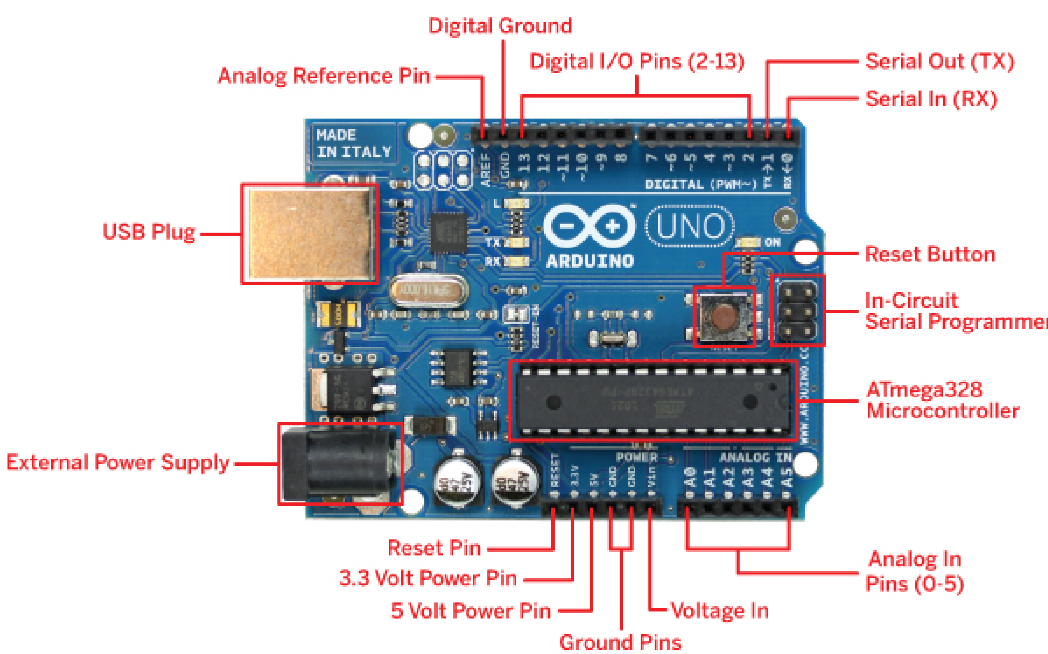


Figure 3. Arduino Uno

Electronics:

Goddard is comprised of many electrical components that contribute to his life-like qualities. Below is a list of the electronics that make up Goddard.

- Raspberry Pi: Acts as the interface between user and Goddard while intelligently commanding the robot
- Pi Hub: Delivers power to the Arduino and facilitates serial communication between the Arduino and Raspberry Pi
- Arduino Uno: Used to control the mechanical actions of Goddard and monitors the sensor network
- 29:1 Metal Gearmotors: Provides Goddard with the ability to move
- Dual MC33926 Motor Driver Carrier: Regulates current and governs the performance of the metal gearmotors
- Maxbotix Ultrasonic Rangefinder: Offers directional awareness
- Pololu IR Beacon Transceiver Pair: Allows Goddard to locate his dog bone using infrared technology
- Power HD Micro-Servo: Controls the movement of Goddard's tail
- Power HD Standard Servo: Manipulates Goddard's head and neck
- Pololu 38 kHz IR Proximity Sensor: Detects human interaction
- 12 V, 9500 mAH Ni-MH Rechargeable Battery: Powers each electronic component in Goddard

System Block Diagram:

The system block diagram is portrayed below in Figure 4.

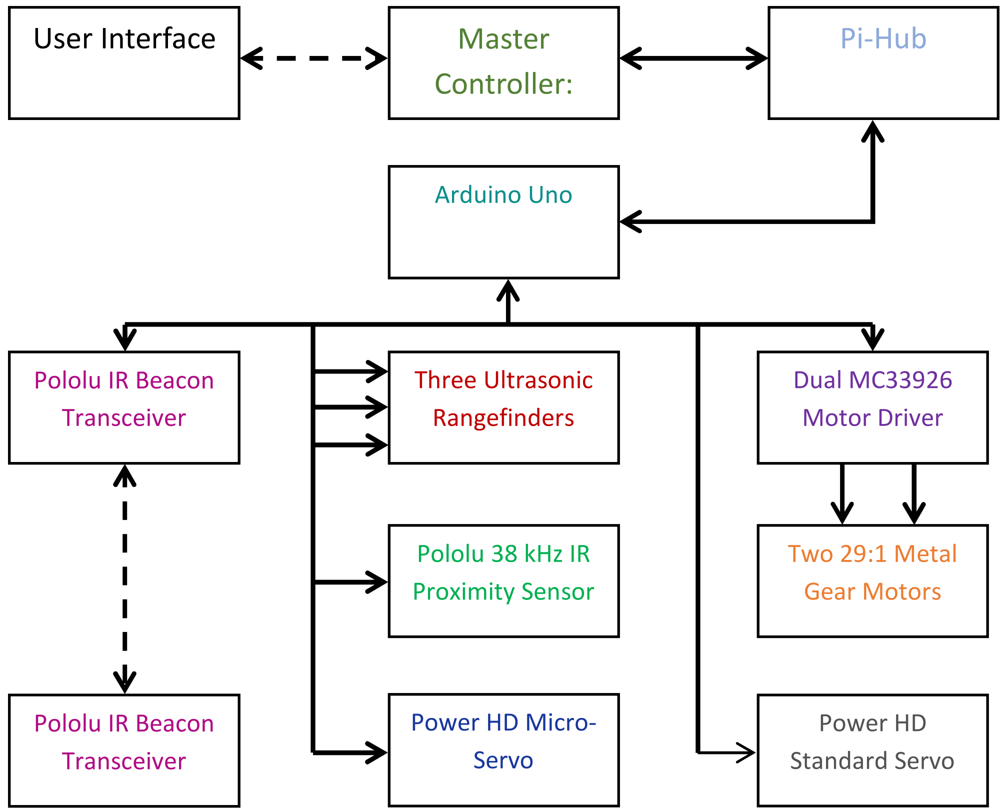


Figure 4. System Block Diagram

Project Roles:

Cassandra DeNunzio

- Executive Project Manager
- Programming Specialist
- Head of Arduino Code

Robert Galli

- Vice Project Manager
- Software Architect
- Pi Specializer
- Chief of IT

Kevin Tierney

- Motor Drive Architect
- Audio Technician
- Electromechanical Process Manager

Andrew Felicetti

- Motor Drive Architect
- Program Developer
- Lead Project Aficionado

Samuel Cordeiro

- Power Manager
- Mechanical Drafter
- Audio Expert

Cameron Perl

- Welding Master
- Mechanical Design Engineer
- Electrical Component Implementer

The mechanical processes of the project were handled mostly by Cameron Perl such as welding and incorporating electrical components into the design. Samuel Cordeiro was in charge of creating all necessary CAD drawings.

Samuel Cordeiro was also responsible for power regulation. It was his job to ensure that each device received its desired voltage.

Andrew Felicetti and Kevin Tierney were the sub group for motor control of the robot. This included movement, direction, and an autonomous drive feature. They created and managed the code that monitored Goddard's sensor network and developed corresponding algorithms to confirm Goddard's proper direction and movement.

Kevin Tierney and Samuel Cordeiro also designed and built the audio amplifier circuit which provided Goddard with his voice and sound.

Cassandra DeNunzio and Robert Galli handled the majority of the programming and microcontroller networking of the robot. Robert Galli was in charge of Goddard's ability to respond to voice commands and move and perform according to a keypad. Robert Galli served as the programming lead to ensure maximum interoperability between different pieces of code. Cassandra DeNunzio was in charge of the main code for the Arduino Uno and handling the mechanical actions of Goddard. Cassandra DeNunzio also put together the code for the various commands of Goddard and set the Arduino code up to accept USB serial information from the Raspberry Pi. Cassandra DeNunzio and Robert Galli both worked to achieve USB serial communication between the Raspberry Pi and Arduino Uno.

Body Construction:

The construction of the body of Goddard started in the fall. It began with simple sketches derived from looking at the cartoon character. This sketch was then turned into a CAD drawing. This CAD drawing proved important dimensions to follow during the construction process. After the designs were agreed by the group, choices of materials that were to be used needed to be made. Steel was chosen to be the best option because of its ease of use, availability, and cost. Aluminum would have been a better choice overall because of its light weight properties which would have made for less strain on the motors and servos, however it's much more time consuming to work with and more expensive in the quantities we required. The two main shapes of steel used were 22 gauge steel weldable sheet metal and ½ inch weldable steel angle iron.

After the materials were picked the construction could begin. Construction was begun by cutting the angle iron into the lengths following the CAD design and welding them into an open box shape to act as the frame. Once the basic frame was together the six walls of the body were cut from a large sheet of metal to the dimensions of the frame. The 4 sides were welded onto the frame. The bottom needed holes cut for the legs to be attached through before it could be welded onto the frame. The top needed to be able to open for access to the electronics that were going to be placed within so it was attached with two small hinges.

Since the basic body was completed the leg frames could then be attached. At first large springs were used for the legs. But after some simple testing they were found to be not rigid enough to support Goddard so they were substituted with rigid angle iron legs which were welded onto the frame through the holes cut in the bottom described earlier. At the bottom of the rear legs brackets were attached making attaching and removing the driving motors easy. At the bottom of the front legs caster wheels were attached. Then the fairly simple task of the tail was tackled by rolling a piece of sheet metal into an elongated cone shape with a rod coming out of the open end with a pivot point so when the tail is attached and connected to the tail servo it can wag back and forth. Another small hole needed to be cut in the back of the body to insert the tail. A small bracket was built to hold the servo that moves the tail.

The next major part of Goddard was the head and neck. The head needed to be as light as possible so it was made out of sheet metal. The shape of the head was constructed by bending and shaping an upper and lower jaw then attaching them with a main bolt. Some aesthetic design was worked in including the teeth, the dome, ears, and tongue. Once the head was completed it needed to be attached to the body with the neck which was made out of a long piece of angle iron, it was attached to the head with a hinge so the head could move up and down and a hinge between the neck and body so the entire neck could raise and lower. Some calculations needed to be done on the physics of the neck to make sure the servos had the torque to move the head. So the length of the lever arm of the neck needed to be made long enough to account for that.

Once the basic mechanics of Goddard were completed the appearance needed to be made to emulate the cartoon. To do this foot cups were created out of sheet metal and attached to the body with black corrugated drainage pipe. A similar material was used to cover the neck and hide the wires running from the head to the body. Now Goddard was ready for the paint scheme. A base coat of grey was applied to cover any imperfections and discolorations in the metal caused by welding and grinding. After the coat of grey, lines were drawn to make Goddard look like he was made out of a patchwork of small pieces of metal riveted together. Then small portions of Goddard were painted yellow like his foot cups, nose, and dome. Then Goddard was covered in a clear coat to protect from rust and other damage. Once the paint was dry Goddard was ready to have his electronics integrated.

Programming on the Raspberry Pi:

Firstly, a Node.js server was created to listen for commands from the user and perform the correct actions based on these commands. This included sending serial data to the Arduino Uno in order to instruct it to perform the correct action. Secondly a user interface was created. The user interface consisted of a variety of buttons that sent various commands from the client (a web page) to the server (hosted on the Raspberry Pi). This was done with HTML5, CSS3 and JavaScript. The keypad that was used to send commands to Goddard is shown in Figure 5.

Figure 5. Keypad

The third task was programming voice recognition. This was accomplished with the Annyang JavaScript library. The Annyang library was included in the client side code and allowed the web browser to recognize speech. "Goddard" was set as the attention word. This meant that whenever a user said the word "goddard" the code would start listening to the person speaking and stop listening when they finished speaking. The code would then parse what the user said. The first word after "goddard" was considered to be the action. Everything after the action was considered the argument. The action would trigger a specific block of code, for example, if a user said "goddard say hello world" Goddard would run the "say" function and pass it the argument "hello world". This leads into the fourth task; getting Goddard to speak. Goddard's voice was built using Espeak, a Linux program for text to speech that was installed on the Raspberry Pi. With this capability, Goddard could say any string of text issued by the user. Finally, the fifth task was giving Goddard his intelligence. This was accomplished with the help of the WolframAlpha API. The code was structured in such a way that when a user asked Goddard a question starting with who, what, when, where or why, Goddard would query wolfram and return an answer via text to speech.

<u>Programming on the Arduino:</u>

In designing the main code for the Arduino it was important to ensure that all of the code was “non-blocking.” We structured the code in this format to ensure that serial commands sent from the pi would be received and acted upon in a timely fashion. We did this by constantly listening for USB serial communication from the Raspberry Pi. When the Arduino Uno would receive serial data it would then read the data to find out which command was sent. We created a coding scheme that assigned each command a different letter or number so that the Arduino could easily process and distinguish the serial data that was sent.

Depending on the command, the Arduino Uno would run a subset of the main code. For bump the servo in Goddard’s head would sweep back and forth allowing his head to bob to the music. For dance Goddard would perform a jittery motion where he drives forward and backward rapidly then left to right rapidly. This gave the effect that Goddard was bopping his head and then shaking his hind end to create a dancing motion. For sleep Goddard would bend his head and neck down while he snored. This required the servos in his head and neck to both sweep downward. Sniff required a similar movement, but for a shorter time period. When Goddard was asked to roam, he would enter autonomous mode which is further described in the next section.

If Goddard was not receiving any serial data from the Raspberry Pi, the Arduino code was set up to check the sonar sensor positioned on his head. If this sonar sensor detected an object above it Goddard would assume that someone was trying to pet him and wag his tail in excitement. You could also make Goddard’s tail wag by telling him he is a good boy or pressing wag on the keypad. The tail wagging was triggered by sweeping a mini servo that was connected to Goddard’s tail.

It was crucial that after Goddard performed an action he would reset back to his normal position. This was done by adding code that described Goddard’s typical state which was immobile and the head and neck positioned at a medium level.

<u>Motor Control and Autonomous Movement:</u>

Goddard's movements are powered by dual high torque DC Metal Gearmotors with a gear ratio of 30:1. Both of these DC motors are attached to both of Goddard's rear wheels for maximum drive power. To allow Goddard a wider range of movements, both front wheels are freely spinning casters that enable Goddard to turn in place. A key element in the design of the movement of Goddard is the interface between both motors, power and the Arduino. To accomplish this task a Dual MC33926 Motor Driver was chosen for this task. This motor driver served as a great asset in simplifying both programming and powering the motors. Considering the motors need increased current during high demand situations and lower current during resting, the motor driver was able to internally regulate both current and voltage directly from the battery source and deliver the appropriate levels to the motors. The motor driver accepted a wide range of input voltages from as little as three volts to as high as twenty-eight volts. This flexibility was crucial to begin motor testing in earlier stages of development when a power source for Goddard was not yet selected. In addition to the ability to interface power directly from the battery to the motors, this driver also allowed for a wide range of control to the motors via digital high – low inputs. This worked perfectly considering the Arduino is a great digital output microcontroller. Utilizing the digital output pins on the Arduino, we were able to easily control each motors direction, speed, low-current sleep mode, slew rates and immediate shut off for the driver. The speed of each motor was controlled by a pulse width modulation output pin on the Arduino that was connected to the appropriate motors pulse width modulation input pin on the motor driver. This gave us the ability to very precisely control the speed of the motors from zero being the slowest and 255 being full speed.

Autonomous movement for Goddard was made possible by utilizing three ultrasonic sensors located on Goddard's east, west and one placed in Goddard's nose. A control equation was produced after various testing of Goddard that allowed him to roam in a walled environment with human interference. The code began by first checking his north sensor, if that was clear Goddard would proceed forward. If this sensor was blocked, Goddard proceeded to compare his east and west sensor to determine which direction had the clearest path and he would turn accordingly.

Power Regulation:

Proper regulation of power was a critical part of our overall design based upon the fact that each individual component required its own specific regulated voltage and current for sufficient performance. A single, 12 Volt, 9.5 Ah NiMH battery was chosen for the main power source, and was utilized for almost every electronic device within Goddard. This battery was selected because of its relatively low cost, long lasting lifespan, and quick charge rate. NiMH batteries are generally chosen for medium to large scale mobile robots, which also influenced our decision. A second 12 Volt, 1.3Ah lead acid rechargeable battery was also integrated within our distribution system, but only powered our two 15W speakers in order to minimize noise with an isolated ground. Once again, this battery was chosen to meet budget restrictions, and also fit within the remaining interior spaced available.

With respect to regulation, an LM323 linear voltage regulator was originally chosen to drive the Pi hub along with the Raspberry Pi and Arduino as they were wired via serial connection. The circuit was designed for 5 Volt 3 Amp maximum output. This proved to be insufficient as the Pi hub runs rich at a specific voltage of 5.2 Volts, 3 Amp maximum. In addition to the unexpected voltage drop across the hub, heat was also a pressing issue. It would be ideal to avoid using a heat sink or fan within our design. After realizing this, an LM2596 adjustable buck DC-DC converter was put in its place, and delivered the exact values for the Pi and Arduino, and dissipated little to no heat over long periods of run time. Because this converter is rated for an amazing efficiency of 92% and performed so well, a second was added to the board and successfully drove all servos at a regulated 6 Volts. Schematic for the LM2596 is provided in Figure 6.

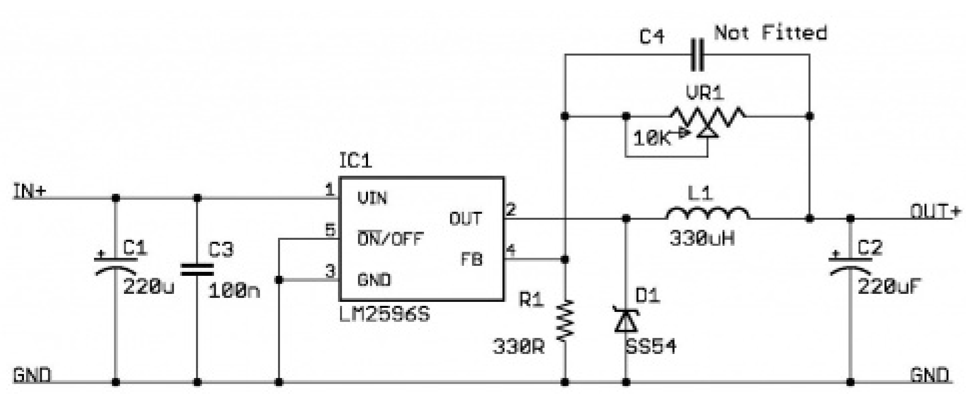


Figure 6. Schematic for the LM2596

With all microcontrollers and servos powered with safe regulation, the drive motors were the only major electromechanical components to tend to. A motor drive controller was placed in line with the Arduino with built in protection, and could handle up to 28 Volts DC. This allowed us to tap directly off of our 12 Volt rail with an originally, unregulated voltage.

Audio Amplifier Design:

In order to provide Goddard the ability to speak, two speakers were mounted on the inside of his body and an audio amplifier circuit was designed. The basic design of the circuit was based on the typical configuration for a 5W amplifier from the LM384 Audio Power Amplifier datasheet. However, after testing this configuration, it was clear that adjustments needed to be made because the sound was very distorted. The first change made to the circuit was adjusting the supply voltage. When the supply voltage was changed from 18 volts to12 volts it cleared up a lot of the distortion, but the audio signal was still somewhat noisy and it was far too quiet. In order to solve this problem, an input buffer stage was designed. The input buffer stage was designed using an LM741 operational amplifier, several resistors, ceramic capacitors and electrolytic capacitors. The op amp is configured to act as a buffer, and resistors R4 and R5 give the buffer its gain. The resistor network seen at the input was designed to ensure minimal offset voltage between the inverting and non-inverting inputs of the op amp. Finally, the capacitors were used to clear up the noise. The electrolytic capacitor is used for low frequency decoupling and the ceramic capacitors are used for high frequency decoupling. Thus the input buffer stage would eliminate noise and allow for a louder audio signal. Once the input buffer was built and tested, it was connected to the audio amplifier via a 10k potentiometer, which allows for volume control. The combination the input buffer stage and the 5W amplifier configuration resulted in speakers that produce a loud and clear sound. A schematic for the audio amplifier in the figure on the following page.

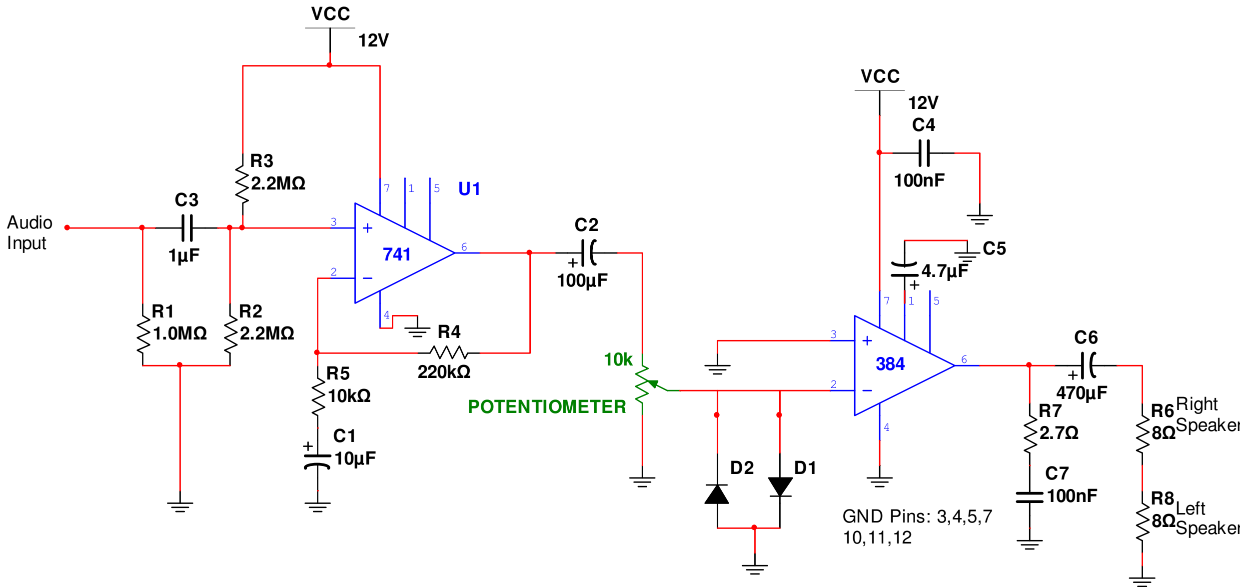


Figure 7. Audio Amplifier Circuit.

<u>Summary of Goddard's Commands</u>

Goddard has many advanced capabilities, all of which are listed below.

- Bark (plays an mp3 of Goddard's bark)
- Bump (plays instrumental hip hop while bobbing head)
- Come (Goddard finds his bone)
- Dance (plays music while moving around)
- DJ (plays music)
- Growl (plays an mp3 of Goddard's growl)
- Help (Displays a list of commands that Goddard understands)
- Who, What, Where, When, Why (queries Wolfram Alpha)
- Kill (stops audio)
- Kiss (plays an mp3 of a kiss)
- Move (manual movement)
- Roam (autonomous mode)
- Say and speak (allows Goddard to verbally communicate)
- Scold (plays mp3 of a whimper and Goddard puts head down)
- Sleep (Goddard snores and puts head down)
- Tweet (tweets to Twitter)

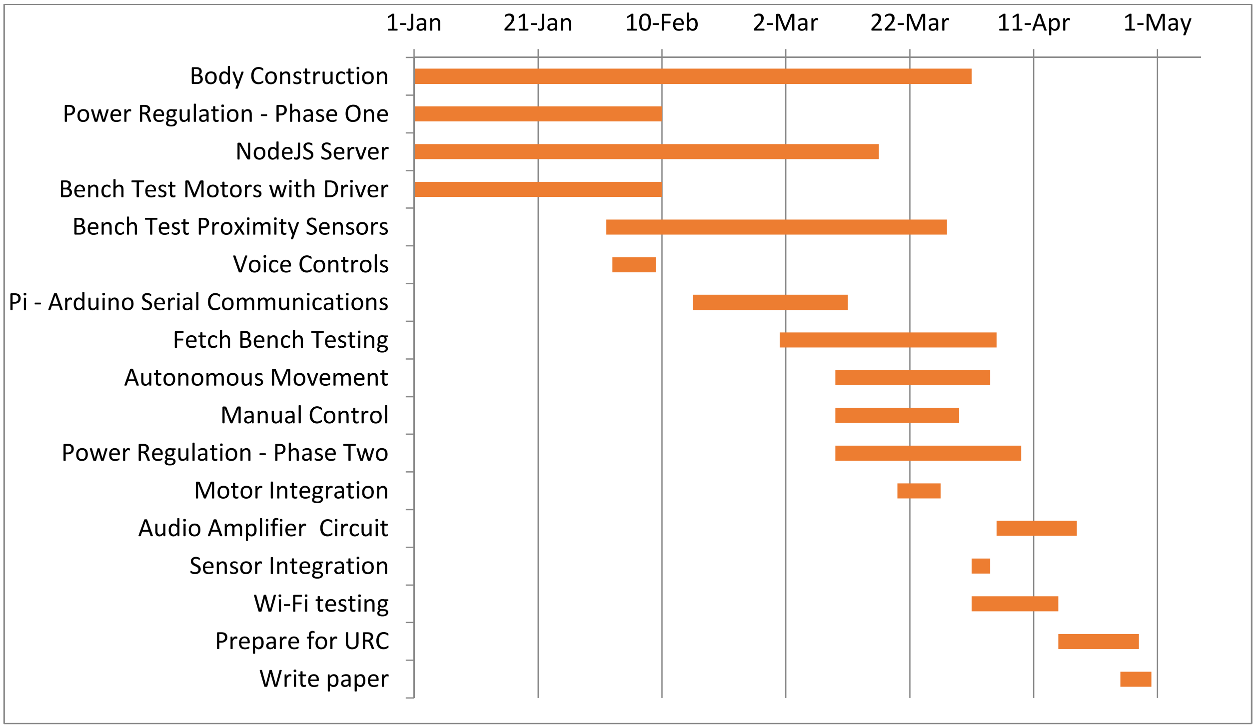
- Wag (tail wag)
- Wake (wakes up from sleep)

Project Related Research:

The Goddard Project was very unique in that it was incredibly interdisciplinary and required a wide range of research. Some of these research fields include but are not limited to:

- Linux
- Node.js
- Object Oriented Programming
- Open Source/Creative Commons
- Version Control/Coding in Teams
- Speech Recognition
- Speech Synthesis
- Artificial Intelligence
- Arduino
- Serial Communication
- Sensor Networks
- Control Systems
- Welding Techniques

Project Timeline:



<u>Final Budget for Design, Implementation, and Ultimate Product:</u>

Item	**Quantity**	**Price**	Funding Source
8 ft^2 Sheet Metal 22 gauge	1	$32.00	Funded by The Goddard Project
Toggle Switch	1	$5.00	Funded by The Goddard Project
3 ft x 3/16" Angle Iron	1	$30.00	Funded by The Goddard Project
Speakers	2	$14.00	Funded by The Goddard Project
Arduino Uno	2	$31.79	Donated by The Programmable Microelectronics Club at UNH
Maestro Mini	1	$39.95	School Funded
Raspberry Pi	1	$39.95	Funded by The Goddard Project
PiHub	1	$34.95	Funded by The Goddard Project
Battery 12 V 9.5 Ah Ni MH	2	$85.00	1: School Funded 1: Funded by The Goddard Project
Battery 12 V 1.3 Ah Lead Acid Battery	1	$26.00	Funded by The Goddard Project
NI MH 6V-12V Charger	1	$25.00	Funded by The Goddard Project
Battery 12 V Lead Acid Charger	1	$10.00	Funded by The Goddard Project
Motor Driver	1	$150.00	School Funded

Pololu Stamped Auminum L-Bracket Pair for Metal Gearmotors	1	$7.95	School Funded
Premium Jumper Wire 50pc Rainbow Assortment	1	$10.95	School Funded
Mid-Torque Servo Motor	1	$15.00	Funded by The Goddard Project
Pololu Universal Aluminum Mounting Hub	1	$7.95	School Funded
Metal Gearmotor 37Dx52L mm with 64 CPR Encoder	2	$39.95	School Funded
Dual MC33926 Motor Driver Carrier	1	$29.95	School Funded
Pololu Wheel 90x10mm Pair	1	$9.95	School Funded
Jumper Wires/Capacitors/Resistors	Many	Donated	Donated by UNH ECE
Sonar Range Finders	4	$20.00	School Funded
IR LED	1(20pc)	$7.87	Funded by The Goddard Project
IR Beacon Transceiver	2	$27.95	Funded by The Goddard Project
9 V Batteries (2	1	$12.00	Funded by The

pack)			Goddard Project
Voltage Regulator LM 323	2	$3.00	Funded by The Goddard Project
Buck Converter LM2596	2	$6.90	Funded by The Goddard Project
Logic Level Converter	2	$16.00	School Funded
Vishay TSSP58P38 IR Detector Module, 38 kHz (3 pack)	1	$2.75	School Funded
Pololu 38 kHz IR Proximity Sensor	1	$5.95	School Funded
Pololu Carrier with Sharp GP2Y0D805Z0F Digital Distance Sensor 5 cm	1	$6.95	School Funded
Power HD High-Torque Servos	2	$19.95	School Funded
Power HD Mini Servo	1	$7.95	School Funded
Acrylic Dome	1	$4.00	Funded by The Goddard Project
LEDs	9	Donated	Funded by UNH ECE

Funded by The Goddard Project: $430.47 (not including pre-existing parts)

Funded by the University of New Hampshire ECE Department: $577.15

Donated by the University of New Hampshire Programmable Microelectronics Club: $31.79

Total Cost: $1039.41

Conclusion:

The senior project experience was both challenging and rewarding. We learned the value of working in teams on a large multifaceted project and had the opportunity to research many different topics. Additionally, we were all able to help design an incredibly neat, intelligent, and lovable robot. We think that our final product is of value because Goddard brings a smile to the face of anyone who he meets, and that in itself, is priceless. We are all thankful to have had this experience and opportunity to work on an amazing project. The skills that we learned will certainly stay with us in all of our future endeavors.