

Final Report

FOR

**REMOTE CONTROL ROBOT WITH
CELL PHONE LINK**

December 12, 2005

Submitted By:

**Justin Schaper
Richard Lienou
Michael Area
Sudhendu Raj Sharma**

Advisor:

Dr. Carl Latino

Abstract

Team Robot Cell Inc. developed a working model of a one-way control link between a user control device (joystick) and a dumb robot (RC car) using two cell phones. The joystick provides the user with proportional control over the speed and steering of the RC car from a remote location. Included in this report are detailed descriptions of the subsystems and integration of this model. The major subsystems described are the control signal generator (transmitter), the filtering system, the speed control circuitry, and the steering control circuitry. The conception, implementation, and integration of the subsystems are explained. The resulting integrated performance of the subsystems and the system as a whole is documented as well.

Table of Contents

| | |
|-----------------------------------------------------------------------|----|
| Abstract | 2 |
| Table of Contents | 3 |
| Table of Figures | 4 |
| Introduction | 5 |
| Objective | 5 |
| Specifications | 6 |
| Design | 7 |
| Transmitter (Richard)..... | 9 |
| Approach to solution..... | 9 |
| Establishing a Control Signal through the Cell Phones (Michael) | 11 |
| Received Signal (Michael)..... | 13 |
| Filtering of the Received Signal | 13 |
| High-Pass Filter | 14 |
| Low-Pass Filter | 16 |
| Speed Control System (Raj)..... | 17 |
| Comparator Circuit | 18 |
| Microcontroller Circuit | 18 |
| Control Circuit | 22 |
| H-Bridge and Rear Motor | 22 |
| Steering Control System (Justin) | 23 |
| Reverse Engineering | 23 |
| Solution: | 24 |
| PIC Program..... | 25 |
| H-Bridge | 25 |
| Power (Justin) | 26 |
| Results and Conclusions | 27 |
| References..... | 28 |
| APPENDIX A..... | 29 |
| APPENDIX B | 40 |
| APPENDIX C | 43 |
| APPENDIX D | 47 |
| APPENDIX E | 60 |
| APPENDIX F..... | 62 |
| APPENDIX G | 67 |
| APPENDIX H..... | 82 |

Table of Figures

| | |
|----------------------------------------------------------------------------------------------|----|
| Figure 1 - Project Idea | 5 |
| Figure 2 - Block Diagram | 8 |
| Figure 3 – Control Signal to Phone Connection | 11 |
| Figure 4 - Speed Control Logic | 17 |
| Table 1 - Frequency Vs Duty Cycle. | 20 |
| Figure 5 - Duty Cycle Vs Frequency | 20 |
| Figure 6 - Duty Cycle Vs Count | 21 |
| Table 2 - Control Bit Value and Pulses | 22 |
| Figure 7 - Diagram of Steering system | 23 |
| Figure A.1 – Transmitter Block Diagram | 30 |
| Figure A.2 – Preliminary Design for a Sine Wave Generator (Not Used in Final Design) | 30 |
| Figure A.3 – Second Preliminary Design for Sine Wave Generator | 30 |
| Figure A.4 – Pseudo Sine Wave Generator (Used in Final Design)..... | 31 |
| Figure A.5 – Output of One Pseudo Sine Wave Generator (High Frequency Channel) | 31 |
| Figure A.6 – Voltage Divider and Operational Amplifier Buffer | 31 |
| Figure A.7 – Summing Amplifier..... | 32 |
| Figure A.8 – Control Signal Generator (Transmitter) Schematic..... | 32 |
| Figure A.9 – Comparator Circuit..... | 33 |
| Figure A.10 - Low-to-High Transition Interrupts..... | 33 |
| Figure A.11 - Flow Chart for Acquiring Number of Counts from Timer upon Interrupts. | 33 |
| Figure A.12 - Servo Motors Control Circuit..... | 34 |
| Figure A.13 – UAF42A Universal Active Filter IC Circuit Diagram | 34 |
| Figure A.14 – UAF42A Typical Application Circuit | 34 |
| Figure A.15 – Actual High-Pass (Band-Pass) Filter Frequency Response..... | 35 |
| Figure A.16 – Maxim MAX291 8 th Order Low-Pass Filter Typical Circuit..... | 35 |
| Figure A.17 – Low-Pass Filter Actual Frequency Response Plot | 35 |
| Figure A.18 – Theoretical Low-Pass Filter Response From MAX291 Datasheet[2]..... | 36 |
| Figure A.19 – Generated Final Control Signal | 36 |
| Figure A.20 – Signal Generated (Top) and Signal Received | 37 |
| Figure A.21 – Latency of about 300ms between the Signal Generated and Signal Received..... | 37 |
| Figure A.22 - Receiver Board..... | 38 |
| Figure A.23 - ADC Setup | 39 |
| Figure B.1 – Joystick and Signal Generator Box..... | 41 |
| Figure B.2 – Female Two Pin Headset Cord to Circuit Connector | 41 |
| Figure B.4 - A Typical Servo Motor..... | 42 |
| Figure C.1 – Band-Pass Filter with center frequency of 2.2 kHz..... | 44 |
| Figure C.2 – Program Indicates Nearly -3dB Attenuation at Low Cut-off Frequency | 44 |
| Figure C.3 – Program Indicates Nearly -3dB Attenuation at High Cut-off Frequency | 45 |
| Figure C.4 – Resistance Values for Use with Figure A.15 to Build Filter | 45 |
| Figure C.5 – Cascade Layout to Obtain a 4 th Order Filter Using Two UAF42As | 46 |
| Figure C.6 – Theoretical High-Pass Filter Frequency Response..... | 46 |
| Figure E.1: Budget..... | 61 |

Introduction

Objective

The primary objective of this project was to build a cell phone link between a transmitter and a dumb robot and provide the capability to operate the robot from a remote location.

The purpose of using the cell phone was to make the operation possible from any remote location in the world where cell phone use is available. The product would include two interface systems. One interface would operate between the transmitter and a sending cell phone, and a second interface would operate between the receiving cell phone and the dumb robot. The interface on the sending side would allow production and encoding of signals suitable for transmission via a cell phone. The interface on the receiving end would process the signals received by the cell phone and control the dumb robot. The simple diagram below illustrates the concept of the project.

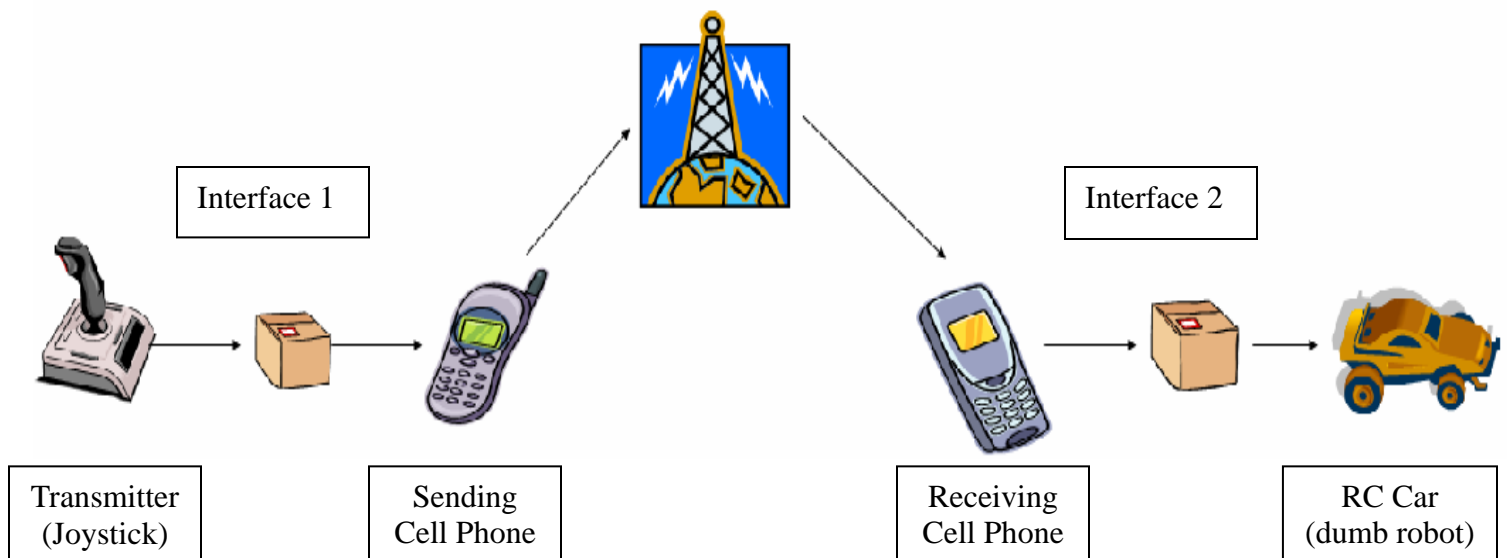


Figure 1 - Project Idea.

Specifications:

The specifications and the constraints that the team had to meet in order to successfully complete the project are listed below.

- The operation of the robot would be indistinguishable from an off-the-shelf model with the exception of a small delay caused by the cell-phone connection. As long as this delay was negligible, the robot would perform just as well.
- The off-the-shelf model would have variable and proportional speed and steering control to allow full control. Therefore, the final design would have this function also.
- The cell phone could not be modified. Only the standard available features (speaker, microphone, hands-free set, etc.) could be used.
- The cell phones used for the connection could be specified models to ensure working communication through a hands-free jack. However a custom-built cell phone was not permissible.
- The hand-held transmitter used could be a custom-built transmitter. Using the actual transmitter that came with the dumb robot is not a requirement.
- The system would be an open-loop system. The robot would not communicate back to the transmitter or make decisions.
- The connection between cell phones could be manually activated. The transmitter and receiver did not have to make or receive a call.
- The system should be a low power device. It must be portable, operating off reasonable number of batteries.
- The final product should be presentable, with components laid out on PCB. The circuitry should be covered, packaged in an aesthetically presentable manner.

Design

The dumb robot chosen for the project was a RC car since it was easier to control. Several options were considered for the design of the system. The first option was to send DTMF signals through the cell phone and decode it at the receiving end and control the car. However, since the robot needed to be fully controllable and a DTMF circuit could only send 16 tones at maximum, this idea was not chosen. The other option was to use actual voice commands to control the robot. However the team could not come up with a satisfactory decoding scheme and this idea was not implemented either. The team decided to use another option, which was to send two single-tone frequencies in a continuous manner and decode them at the receiving end. This option worked best with the requirements of the project since the two frequencies sent could be easily adjusted on the transmitter side. This would allow a larger continuous range of frequencies to be used which would allow a full analog range of control for the car. On the receiving end, it would be relatively easy to decode the signal and change it into control signals.

The following block diagram summarizes the overall final design that used to solve the problem. The subsystems shown in the diagram are explained in more detail below.

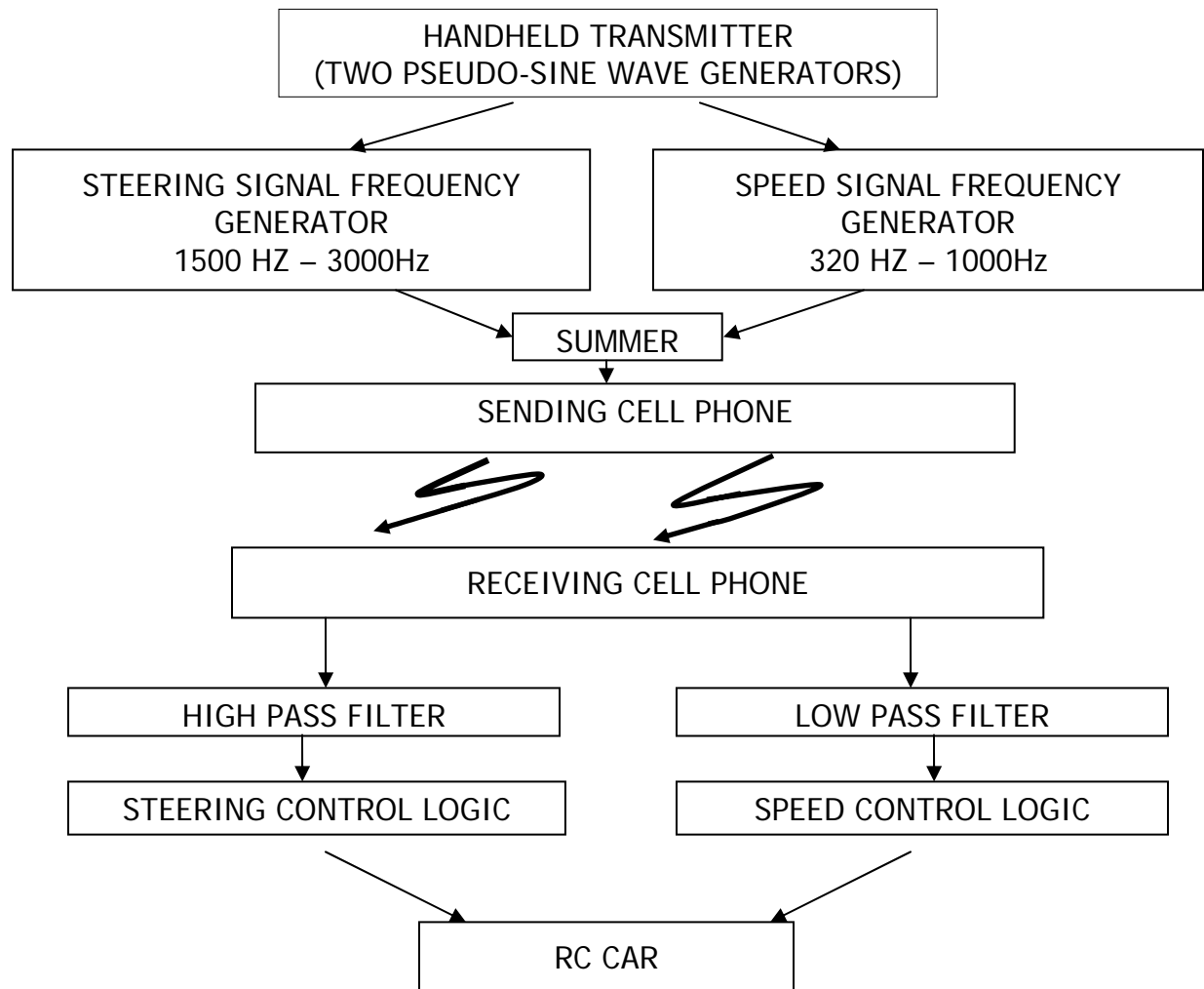


Figure 2 - Block Diagram

Transmitter (Richard)

The hand-held control signal generator (transmitter) contains two signal generators which produce signals that are used to control the speed and the steering of the car. These signal generators are called pseudo-sine wave generators. The first generator generates frequencies ranging from 320 Hertz to 1000 Hertz; this signal is used to control the speed. The second generator generates frequencies ranging from 1500 Hertz to 3000 Hertz; this signal is used to control the steering of the robot. The two sets of frequencies are added together by an operational amplifier summer and sent to the sending cell phone. The block diagram and circuit schematic of the entire transmitting system are shown in Appendix A (Figures A.1 and A.9, respectively).

Approach to solution

Many parameters were taken into consideration while designing the transmitter. A transmitter that was low powered, easy to build, used a reasonably low number of components, inexpensive, reliable, and, most importantly, generated variable frequencies up to about 3500 Hertz (to utilize the human voice and cell phone frequency range) was desired.

The initial approach was to use the original transmitter that came with the robot (car) and to modify it as necessary. After several considerations, the team realized that it was more difficult to modify a product to fit all the requirements. After discussions with the advisor, the team realized designing a new transmitter was the most viable option. The team then decided to design and build a new transmitter which would have two sine wave generators: one signal to control the speed and one to control the steering.

One of the first preliminary designs of the transmitter is shown in Appendix A (Figure A.2). This design was built on the breadboard level and tested. However, it did not work well as the frequency was hard to change using only one parameter. Another preliminary design is shown in Figure A.3 of Appendix A. The frequency of this circuit needed to be digitally controlled and would require an additional digital circuit. This would only increase the required size and complexity of the transmitter and, therefore, was not implemented. A third design used a 555 timer to create an oscillating square wave. This design was used in a few tests involving the cell

phones. However, the square waves had to be filtered down to sine waves before being summed, and proved to be less controllable than the final design.

The final design decided upon by the team is the circuit in Figure A.4 of Appendix A. This circuit satisfied all the requirements and produced frequencies which were easily controllable using potentiometers. The frequencies produced by the original circuit ranged from 6 Hertz to 6 Kilohertz. In order to generate signals with frequencies in the human voice range the team modified the circuit by replacing the 1 Mega-ohm (R5) resistor. A 5 Kilo-ohm potentiometer was used to generate the higher signal range. A 50 Kilo-ohm potentiometer was placed in parallel with a 47 Kilo-ohm resistor and in series with a 4.7 Kilo-ohm resistor to generate the lower signal range. The circuit had three outputs: the pseudo-sine, square and triangle waves. The team had the option to choose the signal that worked the best through the cell phones. The pseudo-sine wave was chosen because it would produce the least number of harmonics and was the best choice for summing two signals together. As the name implies, the pseudo-sine wave is not a pure sine wave but an approximation of a sine wave produced from a diode rectified square wave. The duty cycle of the pseudo sine wave was not fifty-percent, but this proved to be inconsequential. Appendix A (Figure A.5) shows the wave form of one of the sine wave generators. As one can see, the signals also contained a certain level of DC.

The amplitudes of the original sine waves were too high (approximately 3 volts) for the cell phone microphone input and needed to be reduced. This was done using the circuit in Figure A.6 of Appendix A which is simply a voltage divider and an operational amplifier as a buffer to isolate the divided output from rest of the circuit.

Both sine waves were then sent through the cell phones to control the speed and the steering simultaneously. To accomplish this task, a summing amplifier was implemented to add the two sine waves together. The circuit for this summing amplifier is given in Figure A.7 of Appendix A. The signal output of this summing amplifier was sent to the sending phone. A waveform of the final control signal from summing amplifier is shown in Appendix A (Figure A.19). This is described more in the following section of this report.

In addition to battery connectors and power switch, a reset button was installed. This switch (synchronization button) was placed in series with a 10 ohm resistance and connected between the final control signal (the output of the summing amplifier) and ground. The purpose of this button was to correct a stability issue that occurred when sending signals through the cell phones. This stability issued occurred occasionally but unpredictably, causing the receiving cell phone to stop receiving anything other than noise. When the button is pressed, the 10 ohm resistor connection between signal and ground clears (brings to ground) the control signal momentarily and allows the receiving phone to re-acquire the signal. The 10 ohm resistor was necessary to keeps the signal from shorting to ground entirely, as this would sends a signal to the cell phone to end the call. The circuit of the final design is shown in Figure A.8 of Appendix A. This circuit is powered with two 9 volts batteries, one for the positive rail and one for the negative rail.

Establishing a Control Signal through the Cell Phones (Michael)

The control signal is sent to the sending cell phone through a hands-free headset cord. Every standard headset cord is comprised of four wires which are for the microphone signal, the speaker signal, and two common or ground signals. However, not every headset cord has the same cell phone connector. A headset cord connection to the cell phone was used on the sending and receiving end of this system. Using this direct connection to the cell phone ensured that the cleanest and most intact signal was sent to or received from the phone. The microphone ground wire is connected directly to the local ground of the signal generator circuit while the microphone signal connection wire is connected directly to the output of the summing amplifier to relay the control signal. The simple diagram in Figure 3 below demonstrates this connection.

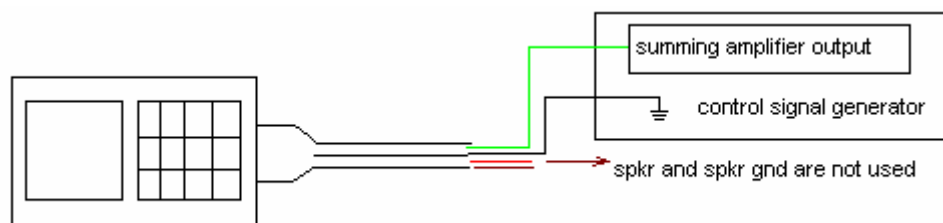


Figure 3 – Control Signal to Phone Connection

The headset cord used in this system is only compatible with a Sony Ericsson T610 model cell phone. However, the system could be modified to connect to any cell phone with a hands-free jack and a microphone input. This modification can be done by splicing the required headset cord and soldering a two-pin male connector to the microphone and ground wires. The two-pin male connector must be one which is compatible with the two-pin female connector inside the control signal generator (transmitter) box. Similarly, the headset cord used on the receiving end of this system is compatible only with a Nokia 6019i model phone, but the receiving end connector can be modified in the same way described for the sending connector. To better understand the sending and receiving cell phone connections, a photograph of the two pin male and female connectors is provided in Appendix B (Figure B.2 and Figure B.3).

The control signal provided by the summing amplifier, has an amplitude of approximately 580 millivolts, Peak-to-Peak. This amplitude is ideal for microphone input. In early attempts to send a sinusoidal signal through a cell phone, only an extremely distorted signal could be salvaged at the receiving end. This was due to the amplitude of the sine wave being extremely too high for the microphone input of the cell phone. Before understanding that the control signal must be reduced, a speaker taped to a microphone was used. Although this allowed the signal to be sent and received, this method introduced a large amount of unwanted noise into the system.

Received Signal (Michael)

Once a connection is established between the cell phones, the received signal has a typical amplitude of 1 volt Peak-to-Peak. The amplitude of the received signal is contingent upon the specific phone model used as well as the speaker volume level setting of the phone. The Nokia 6019i was set at volume level five. The receiving interface was fine tuned for this particular phone model and volume level. This fine tuning included placing a 10 Kilo-ohm resistor between the speaker signal and the ground connections at the receiving end. If a different model phone and headset connector cord is used by the method previously indicated, this resistance could be changed to better match the impedance of the specific phone's headset jack and increase the clarity and stability of the received signal. The system may operate with a different model phone without any modification of this resistance, but the stability of the received signal could be compromised.

The received signal always includes noise, but this noise can be filtered out to provide a clean, reliable signal. The noise is expected due to the voice-quality sound transmission inherent to cell phones.

Filtering of the Received Signal

The received signal is doubled and buffered using a TL074IN DIP (Dual In-line Package) low-noise quadruple operational amplifier IC (Integrated Circuit). The 2 volt signal is connected directly to the input of the high-pass filter. The low-pass filter expects a positive signal oscillating somewhere between 0 and 4 volts. Before the signal is sent to the low-pass filter, the 2 volt signal is amplified again to about 4 volts Peak-to-Peak using another operational amplifier included in the TL074IN package. It is also shifted up to approximately range from 0 to 4 volts by the addition of a 2 volt DC signal. For a more detailed picture of the connections and resistance values used for the amplifiers please see the receiving circuit schematic in Appendix A (Figure A.22).

Both, the low-pass and high-pass, filters are Butterworth filters. Butterworth filters were used because they provide monotonic pass-band and stop-band frequency responses. This characteristic was desirable for this system to prevent potential erratic responses to any unexpected frequency ranges which might damage the circuitry of our system or cause the servo motors on the RC car to respond in an unpredictable manner. Butterworth filters are also the most familiar and, therefore, were the easiest to work with.

High-Pass Filter

The signal sent to the input of the high-pass filter is a 2 volt Peak-to-Peak signal which comes from an output of the TL074IN IC. The high-pass filter expects a 0 volt DC input signal, and therefore, no DC is added to the signal until after it has been filtered.

Two UAF42A DIP universal active filter IC chips are used to create a 4th order Butterworth band-pass filter to isolate the high frequency channel. The 4th order Butterworth polynomial which describes the frequency response is given by the following equation:

$$H_4(s) = \frac{1}{s^4 + 2.613 s^3 + 3.414 s^2 + 2.613 s + 1}$$

A band-pass filter is actually used instead of a high-pass filter to remove very high frequency harmonic distortions and noise in the signal which may be aliased as lower frequency artifacts (signal pieces) and cause interference. The filter will still be referred to as the high-pass filter for distinguishing and referencing purposes.

Each of the UAF42A universal filter chips is capable of providing a 2nd order filter. Cascading two chips creates a 4th order filter. The UAF42A IC is comprised of three low-noise operational amplifiers and several precise (+/- 0.5%) discrete resistors and capacitors. Positive and negative 9 volt supplies are used to power each IC. The circuit diagram for the UAF42A can be found in Appendix A (Figure A.13). Please refer to the references section of this report to find the hyperlink to the complete datasheet for this filter IC [4].

The design of the high-pass filter was done using a DOS-based computer program specifically for the UAF42 IC [5]. The computer program allowed the specification of the filter type, center frequency, and bandwidth of the filter to be used. The program provided resistance values to be used between certain pins of the IC as specified in the typical filter application circuit shown in Figure A.14 of Appendix A. Figure C.4 of Appendix C provides a screenshot of the filter program indicating the correct resistance values to be used to create the desired filter.

A center frequency of 2.2 Kilohertz was specified which gives a lower cut-off frequency of 1.477 Kilohertz and an upper cut-off frequency of 3.277 Kilohertz (where the center frequency equals the square root of the product of the lower and upper cut-off frequencies.) The lower cut-off frequency of 1.477 Kilohertz is near-ideal because the frequencies created by the control signal generator were very near this frequency. Frequencies below 1 Kilohertz needed to have enough attenuation as to not interfere with the low frequency channel. This filter accomplishes these requirements. The upper cut-off frequency was not as ideal. However, the upper-cutoff frequency was not as critical, since frequencies above 3 Kilohertz were not created by the control signal generator (transmitter.) An actual frequency response plot for this filter can be found in Appendix A (Figure A.15). The measured frequency response was found with the filter as a part of the complete integrated system.

A similar active high pass filter could have been designed using other low-noise operational amplifiers and discrete components without the need for IC chips. However, the ICs provided precise capacitor values and dependable Operational Amplifiers. This combination resulted in a much more reliable response than what the other alternative would have provided.

Low-Pass Filter

The signal received by the input of the low-pass filter is a 4 Volt Peak-to-Peak signal centered at about 2 Volts (2 Volt DC offset). The signal comes from the TL074IN IC. The low-pass filter IC will accept an input signal between -1 volts and 4 volts.

A Maxim MAX291 IC is used to create an 8th order Butterworth low-pass filter. The 8th order Butterworth polynomial which describes the frequency response is given by the following equation:

$$H_8(s) =$$

$$\frac{1}{s^8 + 5.126 s^7 + 13.14 s^6 + 21.85 s^5 + 25.69 s^4 + 21.85 s^3 + 13.14 s^2 + 5.126 s + 1}$$

The low-pass filter was created using the example circuit diagram from the MAX291 datasheet as shown in Appendix A (Figure A.16). The circuit is powered using positive 5 volts and ground (0 Volts). The circuit contains a built in oscillator which can be set using a capacitance between pin one of the IC and ground of the circuit. The internal oscillator value is defined by the following equation:

$$f_{OSC} \text{ (kHz)} \approx \frac{10^5}{3C_{OSC} \text{ (pF)}}$$

C_{OSC} is a capacitance between ground and pin one of the IC

The cut-off frequency is equal to 1/100th of the internal oscillator frequency. Using $C_{OSC} = 330$ picofarads, the internal oscillator frequency and the corresponding cut-off frequency is 100 kilohertz and 1 kilohertz, respectively. The theoretical frequency response for this filter from the Maxim MAX291 datasheet [6] is shown in Appendix A (Figure A.18).

The actual frequency response of this filter is very nice (as expected for an 8th order filter.) The high attenuation of frequencies before 1500 Hertz is the most important feature. The actual frequency response plot for this filter when fully integrated is shown in Appendix A (Figure A.17).

Speed Control System (Raj)

The speed control subsystem was responsible for converting the low frequency band signals (320Hz-1000Hz) into signals that would control the rear motor (speed) of the car.

The initial idea for this setup was to use a frequency to voltage converter chip and convert the incoming signals into voltages that would control the motor. However, due to insufficient knowledge about the operation of the chip and unsuccessful attempts of converting frequencies into desired voltage ranges, a different idea was incorporated. The idea was to use a microprocessor (PIC) to measure the frequency of the incoming signal and produce pulse-width-modulated (PWM) signals based on the frequency of the incoming signal.

The block diagram for the speed control logic is presented below in Figure 4. The individual blocks in the speed control logic are discussed in the following sections.

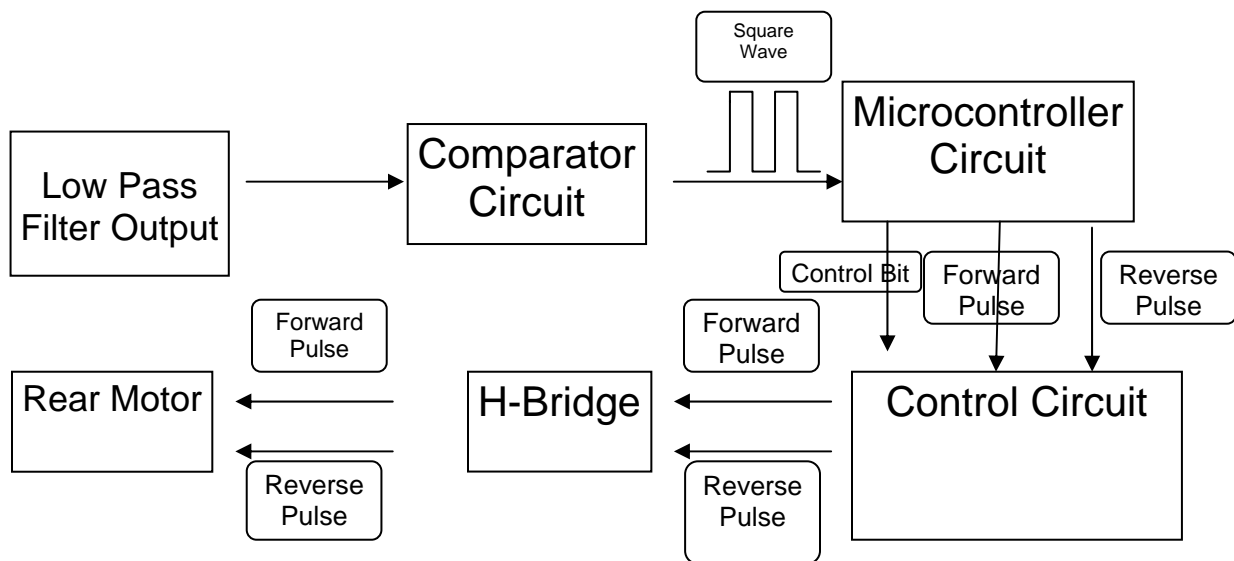


Figure 4 - Speed Control Logic

Comparator Circuit

The function of the comparator circuit is to convert the output of the low pass filter into square waves of amplitude 0 volts to 5 volts. This was necessary because the amplitude of the sine wave output of the filter had magnitudes varying between 0 volts and 4 volts with a DC offset of 2.2 volts. These amplitude values would not always be high or low enough for the microcontroller to register low-to-high transitions in the input. For more information on use of low-to-high transitions, please refer to the *Microcontroller Circuit* section following this section.

The comparator circuit was built using a LM358 low power operational amplifier. Refer to Appendix A.9 for the diagram. The threshold value for the comparator was set to about 2.2 volts using a voltage divider. To get the 2.2 volts, a 10 Kilo-ohm resistor is used along with a 10 Kilo-ohm potentiometer in series with a 5.1 Kilo-ohm resistor. The output of this comparator circuit was a square wave ranging from 0 volts to 5 volts with the same frequency as the incoming wave.

Microcontroller Circuit

The function of the microcontroller circuit was to read in the frequency of the incoming signal. This was done using a PIC16F876A microcontroller which is a 28-pin DIP chip.

The PIC has a built in feature that can read low-to-high and high-to-low transition on specific pins as an “Interrupt” function and can be programmed to perform a different subroutine in case of an “Interrupt.” Refer to Figure A.10 for a diagram of Interrupts.

The PIC also has another built in feature called “Timer.” The function of the “Timer” is to count up in units of instruction cycles. An instruction cycle for a PIC is determined by the oscillator which provides the microcontroller with a clock. Four clock cycles for the oscillator is equivalent to one instruction cycle. For example, if a 20MHz oscillator is used for clocking the PIC,

$$1 \text{ clock cycle} = \frac{1}{20,000,000} = 0.05 \text{ microseconds}$$

$$1 \text{ instruction cycle} = 4 \text{ clock cycle} = 0.2 \text{ microsecond.}$$

Therefore the “Timer” of a PIC clocked with a 20 Megahertz oscillator can count up in units of 0.2 microseconds.

The “Timer” feature of the PIC, along with the “Interrupt” feature can be combined to measure the time duration between two interrupts. Please refer to Appendix A.11 for a flow chart of how to measure counts with interrupts.

Whenever the PIC receives an interrupt, it goes to an Interrupt Subroutine. The value of count from the timer is stored and the timer is reset to 0 and the PIC comes out of the subroutine and timer starts counting. When another interrupt occurs, the PIC goes back to the subroutine, gets the number of counts between the last interrupt, resets the timer again, and this process continues. The time duration between any two interrupts can thus be measured and this time duration is the Time Period of the incoming signal. The following formulas can be used to calculate the time period and frequency of the incoming signal.

$$\text{Time Period} = (\text{No. of Counts from Timer}) * 0.2 \text{ microsecond.}$$

$$\text{Frequency} = 1 / (\text{Time Period})$$

Once the frequency/time period of the incoming signal is found, it is mapped into a PWM signal of duty cycle varying between 0% and 95 %. The duty cycle is determined by the time period of the incoming signal which in term is determined by the position of the joystick in the y-direction. As the joystick changes is position along y-axis, so does the frequency produced by the transmitter and so does the Timer Values of the PIC. The following table shows the relationship between the joystick position, frequency produced, duty cycle produced and the direction of motion produced.

| Joystick Position (with reference to y-axis) | Frequency Range | Counts | PWM Duty Cycle Range | Motion |
|-------------------------------------------------|-----------------|-------------|----------------------|---------|
| 0 | 440Hz | 11363 | 0% | Neutral |
| 0 → +ve | 440Hz-1000Hz | 11363→5000 | 0%-95% | Forward |
| 0 → -ve | 440Hz-320Hz | 11363→15625 | 0%-95% | Reverse |

Table 1 - Frequency Vs Duty Cycle

As seen from the table, when the joystick is in the 0 position with respect to y-axis, the frequency produced is 440 Hertz and there is no motion in either the forward or the reverse direction (i.e. this is the “neutral” position.) When the joystick is moved in the +ve direction, the frequency increases from 440 Hertz to 1000 Hertz and so does the duty cycle from 0% to 95% in the forward direction. Similarly when the joystick moves from the 0 position to –ve direction, the frequency decreases from 440 Hertz to 320 Hertz and the duty cycle increases from 0% to 95% with the motion being in the reverse direction.

The relationship in the above table can also be represented in graphical form as shown below

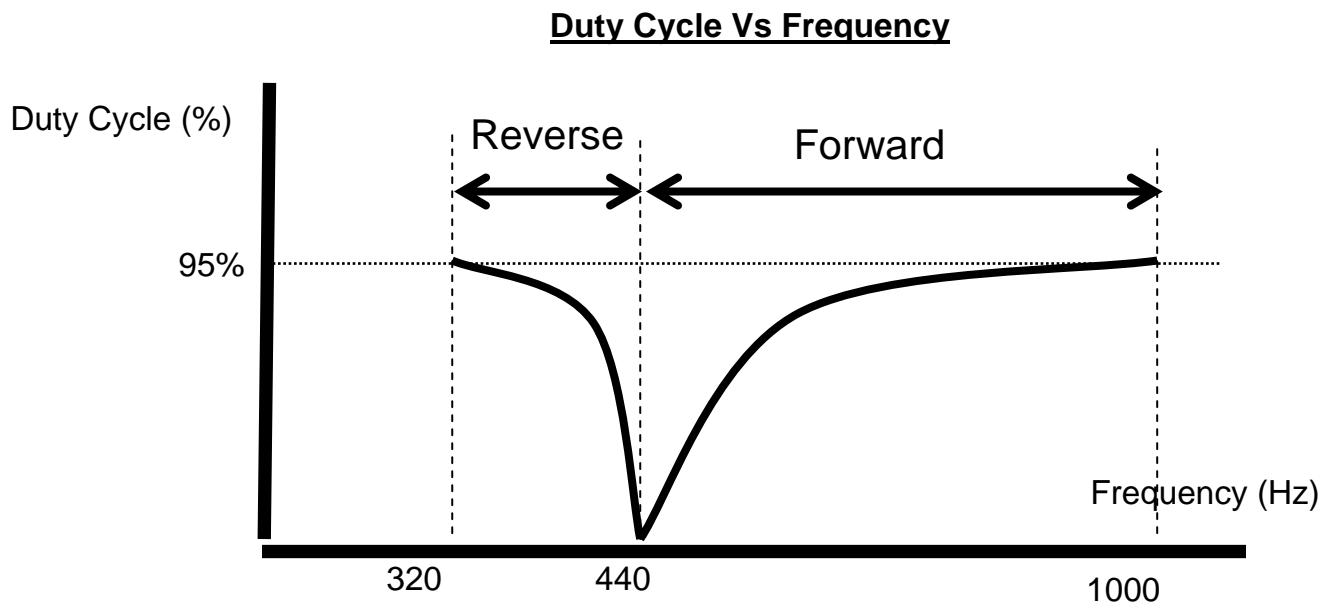


Figure 5 - Duty Cycle Vs Frequency

The above graph shows non-linear relationship between the duty cycle and frequency. This is because the transmitter itself has a non-linear relationship between the frequency and the joystick position. The joystick's movement in +ve direction produces about 560 Hertz change in frequency where as in the reverse direction the change in frequency is only about 120 Hertz.

However in terms of period of the signal and the number of counts provided by the counter, the graph looks as shown following figure.

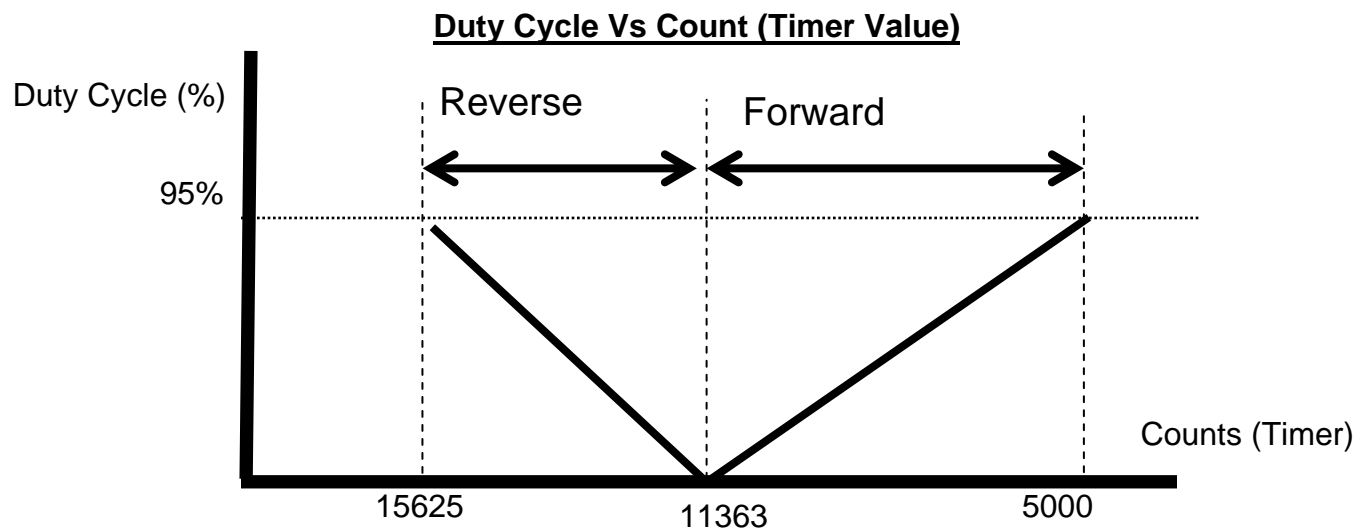


Figure 6 - Duty Cycle Vs Count

As seen from the above figure, the relationship of duty cycle versus count (from timer) is linear due to the fact that there is a big difference in count values at lower frequency where as there is relatively smaller difference in count values at higher frequencies. A frequency of 440Hertz maps to count of 11364, frequency of 320 Hertz maps to 15625 and frequency of 1000 Hertz maps to 5000. As we can see, the difference between the count values is not very much.

There are also some safety bands added to the boundaries of the actual expected frequency band of 320 Hertz to 1000 Hertz. Many times there are glitches in the cell phone or in the transmitter itself that produces frequencies outside the expected band. The safety band on the lower frequency side is about 70 Hertz and on the upper frequency is about 100 Hertz. So any signal between 320Hz and 250 Hz would be considered 320 Hertz and treated as full reverse (95% duty cycle). Similarly, any signal between 1000 Hertz and 1100 Hertz is considered 1000 Hertz and treated as full forward.(95% duty cycle). There is also another feature that reduces the effect of

glitches that cause the frequency to drastically change. In case of any erroneously large changes in frequency, the PIC rejects the change in frequency and uses the previous frequency.

For the actual program, code and the flow chart for the program, please refer to Appendix C.

Control Circuit

The function of the control circuit is to ensure that the motor is not damaged due to any glitches in the circuit that causes both forward and reverse pulse to be sent out at the same time. This is done by using the “Control Bit” sent by the microcontroller along with the Forward and Reverse pulse and using discrete logic 7400 series chips. Please refer to Appendix A (Figure A.12) for the circuit diagram.

The circuit contains an inverter (SN74LS04) and two 2-input AND gates (SN74LS08) and are connected as shown in the diagram (Figure A.12). The following table illustrates how the control bit controls which pulse to allow to the motor.

| Control Bit | Pulse Allowed |
|-------------|---------------|
| 0 | Reverse |
| 1 | Forward |

Table 2 - Control Bit Value and Pulses

As seen from the table when the control bit is high (or 1), the forward pulse is allowed and when the control bit is low (or 0), the reverse pulse is allowed.

H-Bridge and Rear Motor

The function of the H-Bridge is to provide sufficient current amplification so that the Forward and Reverse Pulses can turn the rear motor forward and reverse.

The H-bridge and motor used were both present in the RC car already. Reverse engineering was done to find out the connections going into the H-bridges and the Forward and Reverse pulses were directed to these connections.

Steering Control System (Justin)

Steering System

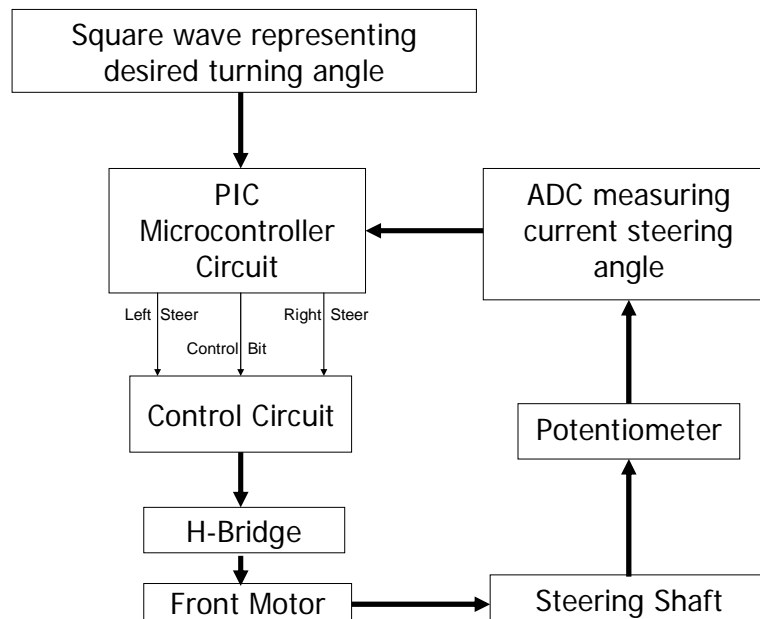


Figure 7 - Diagram of Steering system

Reverse Engineering

Work on the Steering System began by again looking at how the original car was controlled. The car's steering system consisted of a basic servo motor setup similar to the one shown in Figure B.4. A typical directional servo consists of a motor, a potentiometer, and the circuitry necessary to control the angle of the motor shaft given a Pulse Width Modulated (PWM) signal representing the desired angle. The motor shaft is then connected to the steering shaft through a number of gears. The potentiometer is mounted on the steering shaft to measure the current angle of the steering shaft. The circuitry then compares the desired angle to the current angle and sends control signals to the motor to make the necessary correction. Unfortunately, in the attempt to discover how the original system could be used within the new system, the original control circuitry for the servo was damaged.

Solution

It was determined that the motor and potentiometer alone could be used to control the steering angle. The idea was to use a PWM signal created by the same PIC used to measure the higher frequencies which correspond to the desired steering angle. In order for this to work, the desired angle was compared to the current angle and powered the motor in the appropriate correcting manner. The PIC already has the desired angle after measuring the incoming frequency. The potentiometer would obviously be used to get the current angle, however an Analog-to-Digital Converter (ADC) circuit would have to be constructed for this purpose. Three wires previously ran from the car's servo to the potentiometer. These wires were cut and connected directly from the potentiometer to the receiving circuitry instead. The green wire runs to one end, the yellow wire runs to the other end, and the white wire runs to the middle terminal, known as the wiper terminal. The green wire is connected to the regulated 5 volt supply, and the yellow wire is connected to ground. The white wire is then the center of a voltage divider and used as the output signal. The voltage at the white wire is then directly related to the current position of the steering shaft. As the shaft, moves in one direction or the other, the potentiometer rotates that direction which causes the voltage of the white wire to increase or decrease appropriately. The purpose of the ADC is then to convert this voltage to a binary number that the PIC can use. The PIC 876A actually has Analog-to-Digital pins with built-in ADCs (Analog-to-Digital Converters.) However, an external parallel connecting 8-bit ADC was chosen instead. The ADC is then connected directly to an 8 pin port on the PIC. The decision to go with the external ADC was made for two reasons. The first was that the value of the ADC could be easily read by placing either a voltmeter or an oscilloscope at the 8 output pins. LEDs were placed at the outputs in a similar manner to that shown in Appendix A (Figure A.23). This allowed a visual for the binary values that were being sent to the PIC. The second was that the team had little experience using internal ADC methods in PICs, and a successful method was never determined. The ADC chosen was an ADC0804. The values of the ADC were recorded for the middle position and the two extremes, left and right of the steering shaft. The ADC value was approximately 128 in the center. It got as low as 74, and as high as 181.

PIC Program

The program measures the period of the incoming signal using the interrupt procedure described above. The number of counts is then scaled to range from 128 to 256. Then the 8 bit number corresponding to the current angle of the shaft is read in from the ADC. This number is scaled to range from 0 to 128 using the reference values measured earlier. This allows the correcting value (desired – current) to range from 0 to 256. This method eliminates the need for negative numbers by making 128 correspond to no error between the desired value and the current value. If the correcting value is greater than 128, a correction is needed in the right direction; if the value is less than 128, a correction is needed in the left direction.

The plan was to start with proportional control and then move to more advanced control for the motor the steering if necessary. However, in the first few tests, there was a large amount of drag on the motor by the turning front wheels, and the motor reaction time was very low. The decision was made to just make the wheels turn quickly at a fixed rate until the error was very small, and then to start doing proportional control to bring the error close enough to zero that the steering angle would be considered correct by the user. This was set as a dead band in the code. If the wheels are within about 10 degrees of the desired angle, the software considers this good enough and quits reacting until the desired changes again. This system of control seemed to work very well, with the exception of a few coding bugs. For example, the ADC value actually started going beyond the previously measured maximums and minimums causing the program to freeze up. This was easily corrected however by adding a few lines of code that set the variable to the extreme whenever the reading went past the extreme. This control system was then set as our final controller. However, the PIC also needed a way to power the front motor.

H-Bridge

It has already been stated that the rear motor was powered with an H-bridge circuit already present on the car. However, there was not a convenient device available to power the front motor. The PWM control discussed above along with another H-bridge to move the motor appropriately was used. The H-bridge chosen was a SN754410NE-1. It is a Quadruple Half-H Driver. It was chosen because it comes in a 16 pin DIP and it was readily available. It supplied the necessary power to move the motor and responded to 5 volt inputs from logic chips.

Therefore, the same logic safety control circuit could be used that was used for the rear motor. The advanced Proportional Control System including the H-bridge moved the motor appropriately and the drag on the wheels minimized overshoot issues. The system performed well enough to give a descent amount of proportional steering. That is, you could turn the wheels to a desired angle, within a degree of acceptable error, as specified by the amount of movement in the joystick.

Power (Justin)

The team originally hoped to power the logic and the motors with the same battery. The battery that came with the car was a 7.2V 1500mAH rechargeable battery. Unfortunately, the rear motor put a very large load on this battery, and the voltage often fell well below 5 volts when driving hard. This would make it very difficult to regulate and keep a consistent 5 volt supply necessary for control of the logic chips and micro-controllers. Therefore, when integrating the systems, the truck carried 4 batteries— the original 7.2 to power only the rear motor (for speed), one 9V battery for a +9V rail, one 9V for a -9V rail, and one 9V that supplied power to the front motor and to a 5V regulator that powered the ICs. However, the last 9V would run low very quickly; a quick fix for this was to add a fifth battery to separate the ICs from the steering. The next step was to cut the number of batteries down. It was decided that the 7.2V battery should remain devoted to the rear motor. To power everything that required a positive voltage, a rechargeable 9.6V 100maH battery was used. This supplied power to the 5V regulating circuit and to the H-bridge for the front motor (for steering), and it also acted as the approximate +9V rail. Another battery was necessary for the negative rail however and a standard 9V battery was still used for this.

On the transmitter side, the power requirements were of course much less, and only two batteries were necessary. One 9V for the positive rail and one 9V for the negative rail were used.

The 7.2V (1500mAH) battery for the speed of the car should last approximately 69 minutes since it draws 1.3 A at full speed. The battery used is old, however, and does not last as long as it should. Similarly, the 9.6V (1000mAH) battery for the steering should last approximately 260 minutes (130 mA constant consumption, 230 mA consumption when turning.)

Results and Conclusions

The overall performance of the system was satisfactory and it met all the specifications. The user could send signals through the cell phone and control both speed and steering of the car. The transmitter produced the correct frequency ranges and the receiver received them correctly (please refer to Figure A.20.). The speed and steering control system decoded the signals and controlled the car very well. The latency of the overall system was around 300ms (please refer to Figure A.21). The majority of the latency was introduced by the cell phone-to-cell phone communication. The latency was calculated with the two cell phones in close proximity. It can be expected to increase as the distance between the cell phones increase because the signal transmission time between the cell phones would increase (if having to route through more towers, satellite, etc). However, the processing time in both the transmitter and receiver circuitry was minimal and thus the overall system performance was fast enough. The car was moving and steering smoothly and the delay was negligible. With more time and money, a camera could be integrated on the car to get a visual for where the car is going. This project was very challenging, but the team prevailed.

In conclusion, the project was a success. There were many obstacles that really tested the team's resolve. Disagreements and misunderstandings created tension between team members and even with the team's faculty advisor. However, keeping a professional approach with a determination to successfully complete the project held the team together. The team members grew more and more accustomed to working together. Working hard and relying on each other, the team succeeded on completing the project on time, and succeeded in demonstrating its performance.

References

- [1] Analog Edge, “A Quick Sine Wave Generator,” December 11, 2005
<http://www.national.com/nationaledge/jun04/article.html>

- [2] Dallas Semiconductor, “Digitally Controlled Sine-Wave Generator,” December 11, 2005
http://www.maxim-ic.com/appnotes.cfm/appnote_number/2081

- [3] Instructions, Parts List, and Schematic. Operational Amplifier Function Generator, CK103 – Opamp Function Generator, October 2005.
<http://electronickits.com/kit/complete/meas/ck102.pdf>

- [4] Datasheet, UAF42AP, Burr-Brown (Texas Instruments) Universal Active Filter, November 2005.
<http://focus.ti.com/lit/ds/symlink/uaf42.pdf>.

- [5] Burr-Brown Corporation, Texas Instruments, Inc. (1991, 2001). *Filter42* [Computer program]. UAF42 Design Program. Tucson, AZ: Applications Engineering. (Application Program No. AB-035)
<http://focus.ti.com/docs/toolsw/folders/print/filter42.html>

- [6] Datasheet, MAX 291, Maxim 8th-Order, Low-Pass, Switched-Capacitor Filters, October 2005.
<http://pdfserv.maxim-ic.com/en/ds/MAX291-MAX296.pdf>

- [7] Photograph, “A servo disassembled,” Seattle Robotics Society, November 2005.
<http://seattlerobotics.org/guide/servos.html>

- [8] Datasheet, ADC0804, National Semiconductor 8 bit A/D Converter, November 2005.
<http://www.national.com/ds/DC/ADC0801.pdf>

APPENDIX A

System Schematics, Diagrams, and Plots

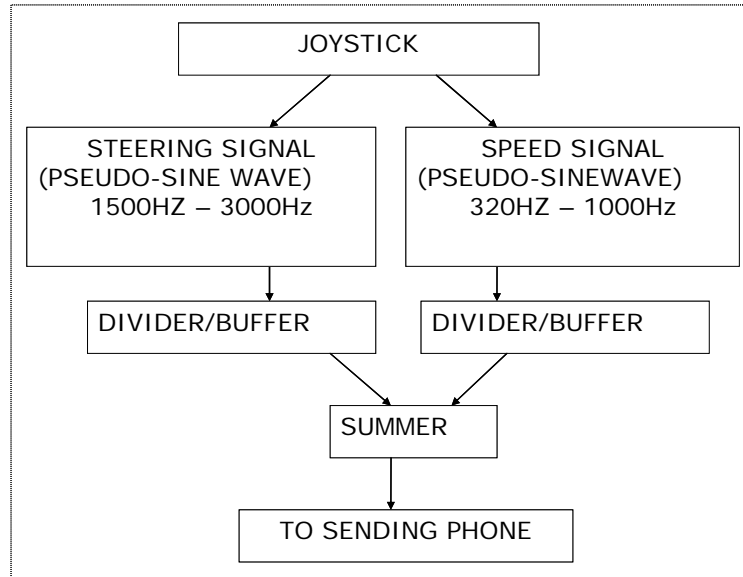


Figure A.1 – Transmitter Block Diagram

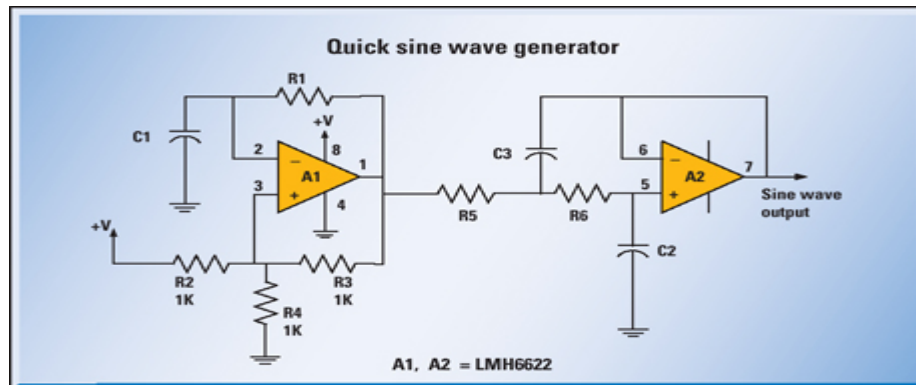


Figure A.2 – Preliminary Design for a Sine Wave Generator (Not Used in Final Design) [1]

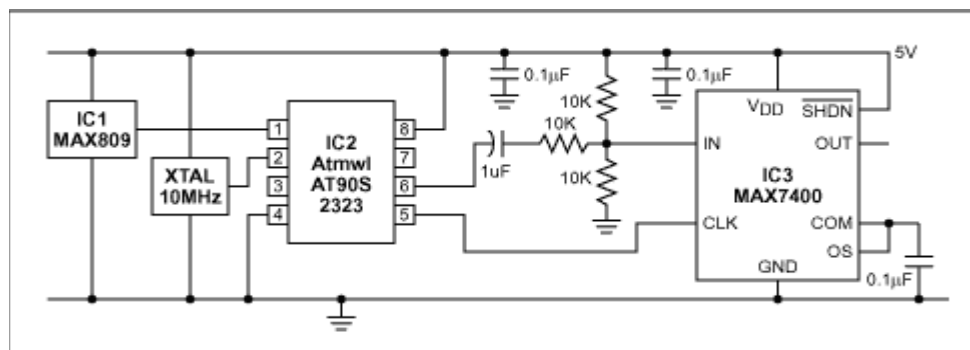


Figure A.3 – Second Preliminary Design for Sine Wave Generator [2]

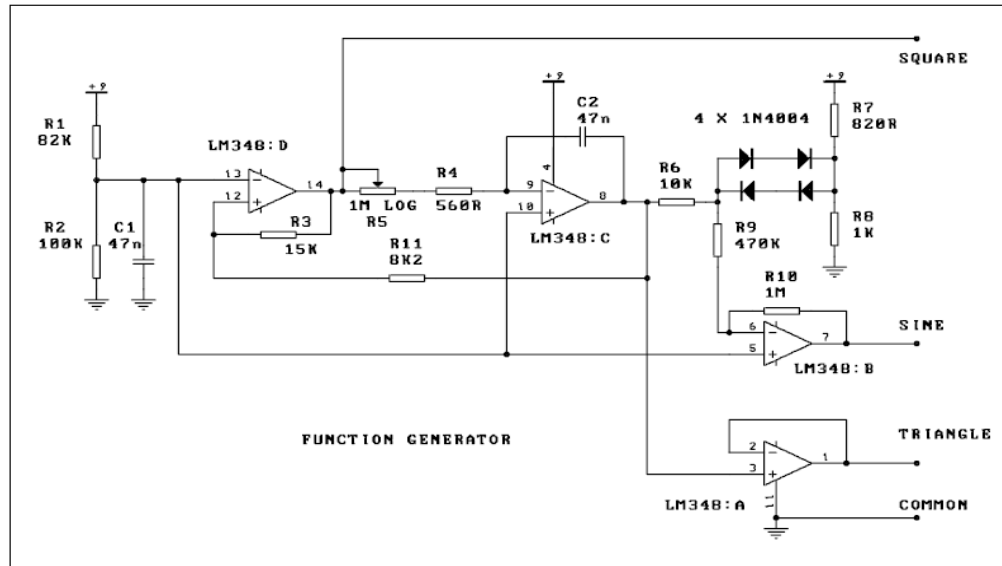


Figure A.4 – Pseudo Sine Wave Generator (Used in Final Design) [3]

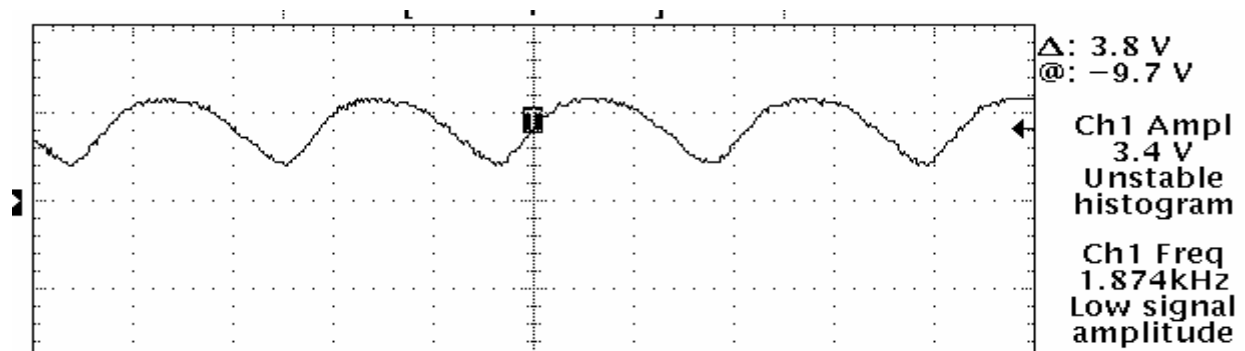


Figure A.5 – Output of One Pseudo Sine Wave Generator (High Frequency Channel)

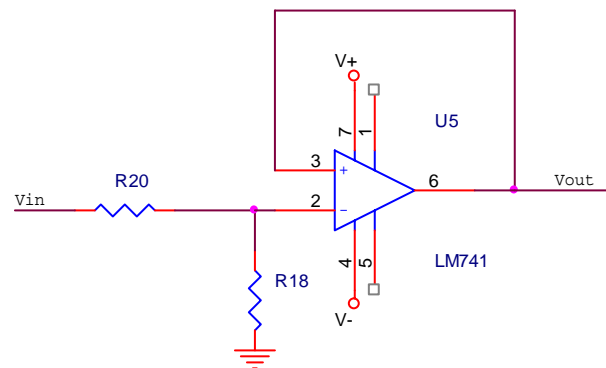


Figure A.6 – Voltage Divider and Operational Amplifier Buffer

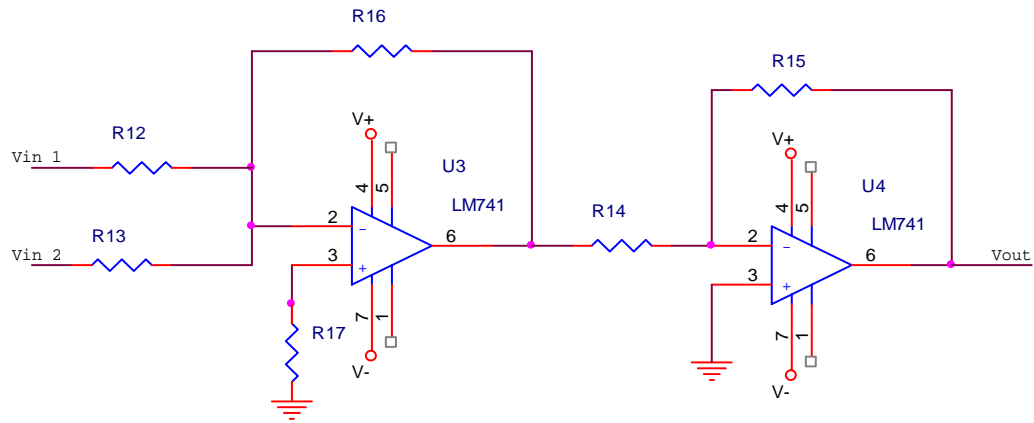


Figure A.7 – Summing Amplifier

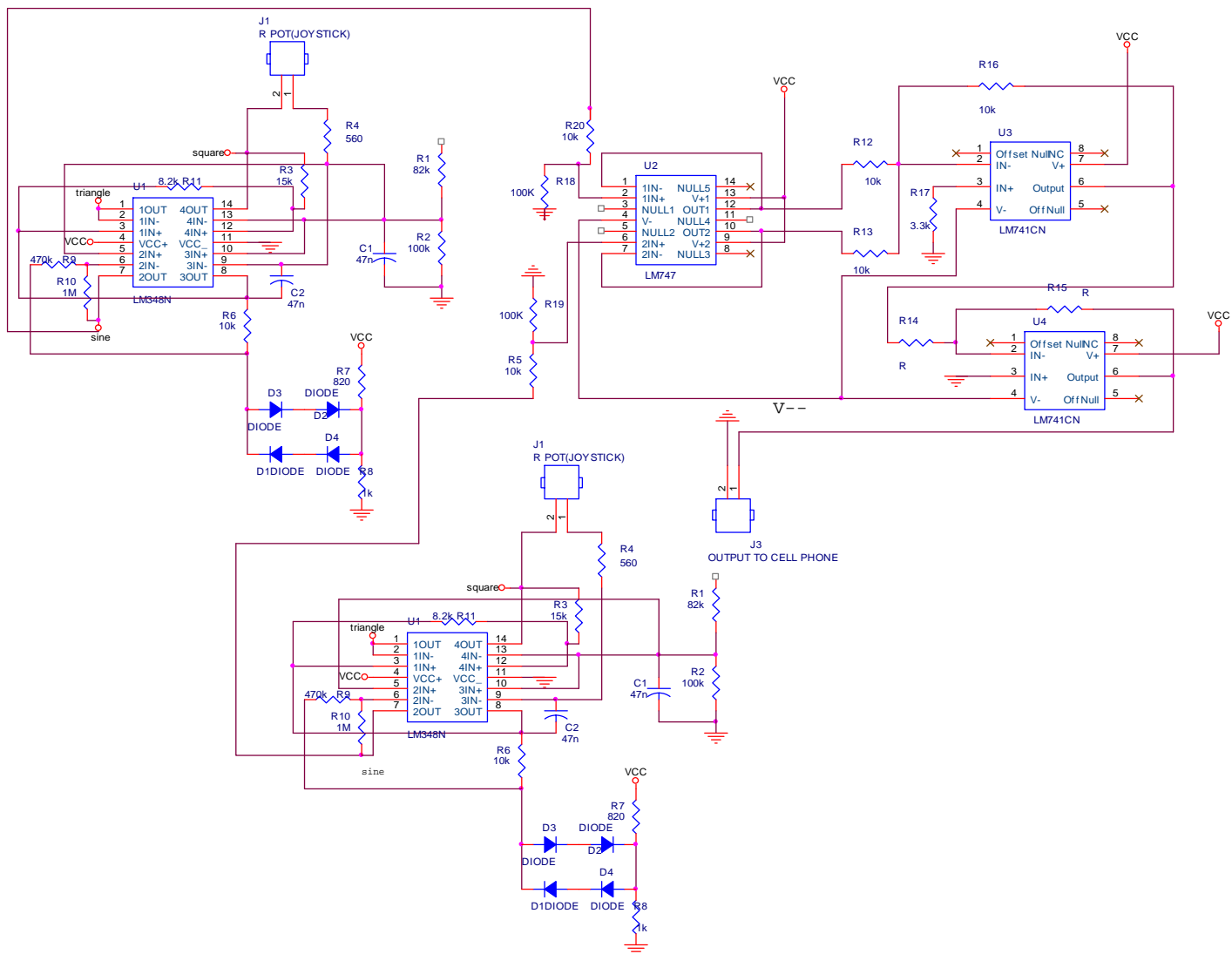


Figure A.8 – Control Signal Generator (Transmitter) Schematic

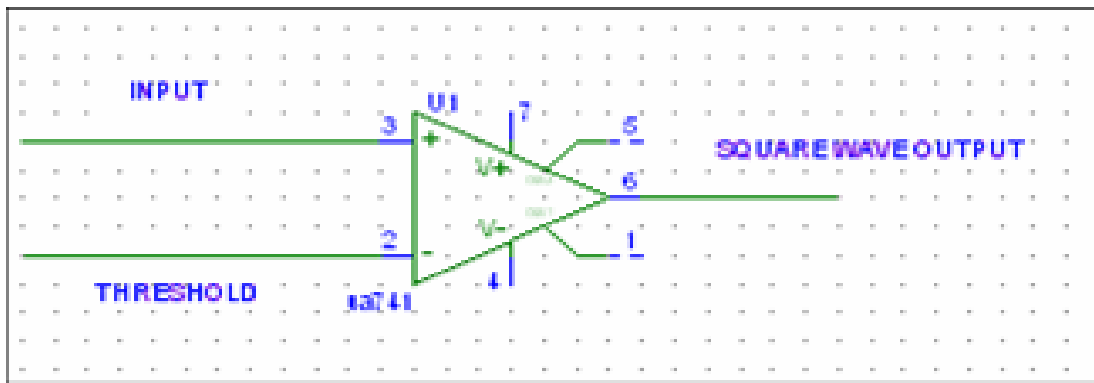


Figure A.9 – Comparator Circuit

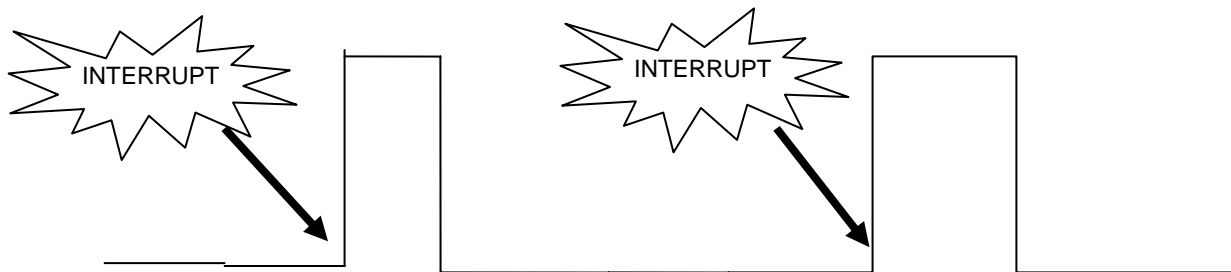


Figure A.10 - Low-to-High Transition Interrupts

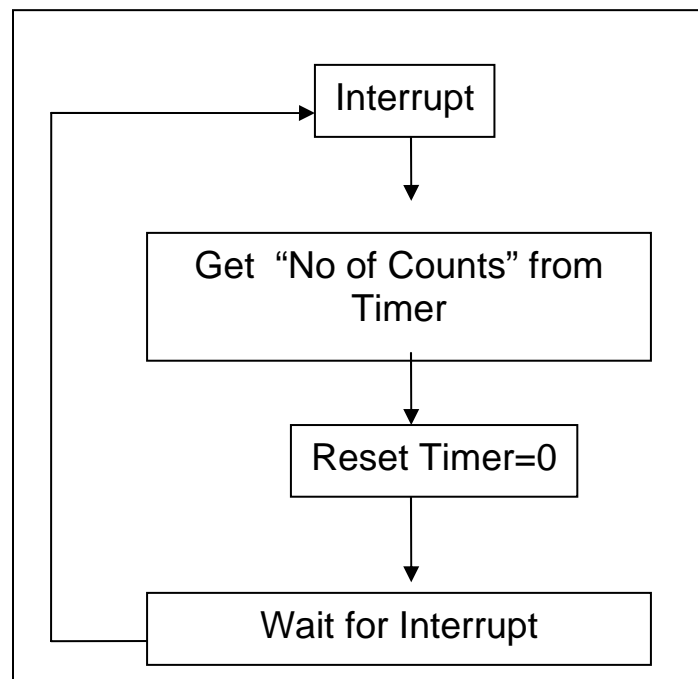


Figure A.11 - Flow Chart for Acquiring Number of Counts from Timer upon Interrupts.

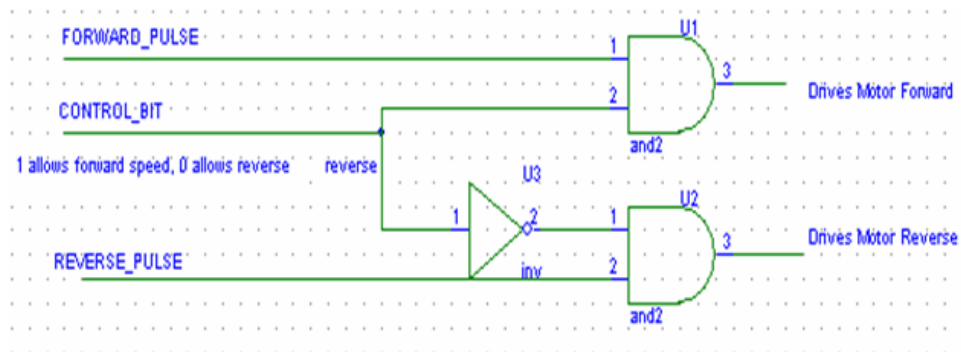


Figure A.12 - Servo Motors Control Circuit

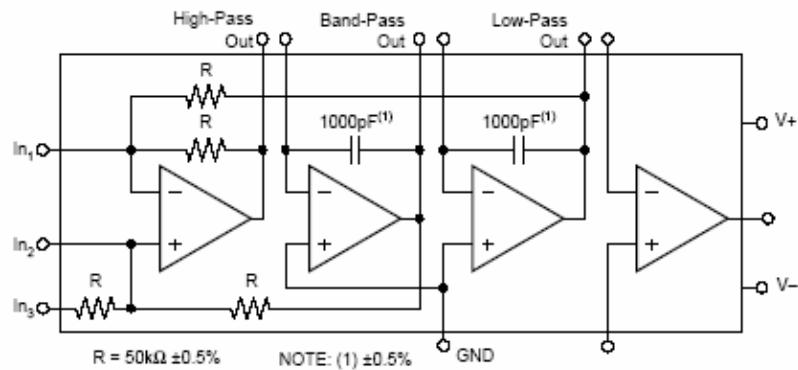


Figure A.13 – UAF42A Universal Active Filter IC Circuit Diagram [4]

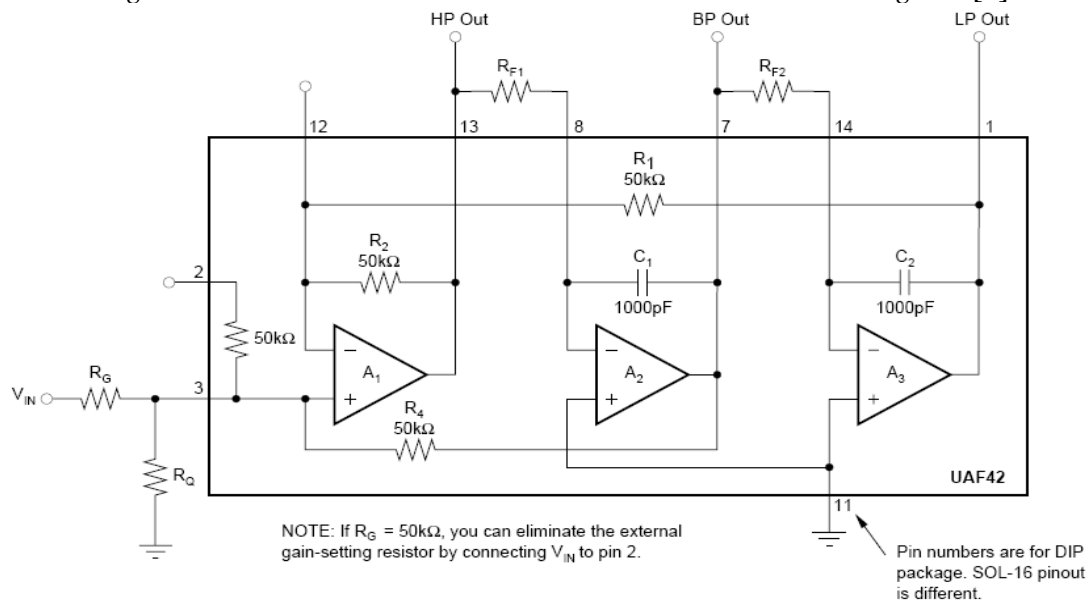


Figure A.14 – UAF42A Typical Application Circuit [4]

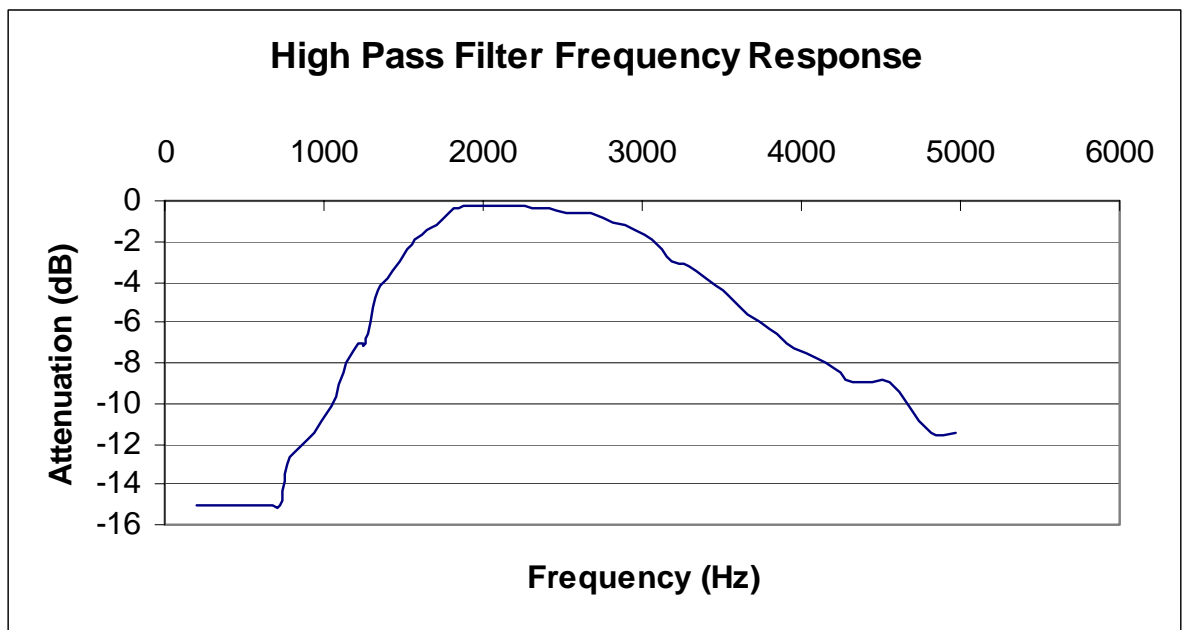


Figure A.15 – Actual High-Pass (Band-Pass) Filter Frequency Response

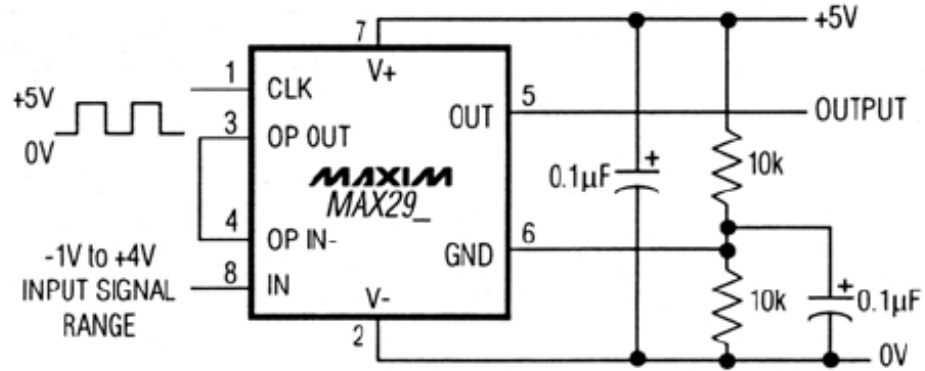


Figure A.16 – Maxim MAX291 8th Order Low-Pass Filter Typical Circuit [6]

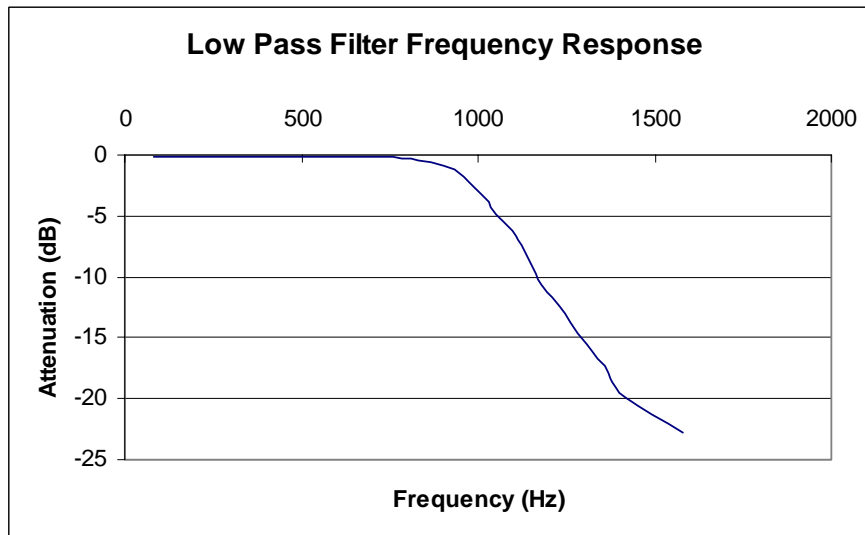


Figure A.17 – Low-Pass Filter Actual Frequency Response Plot

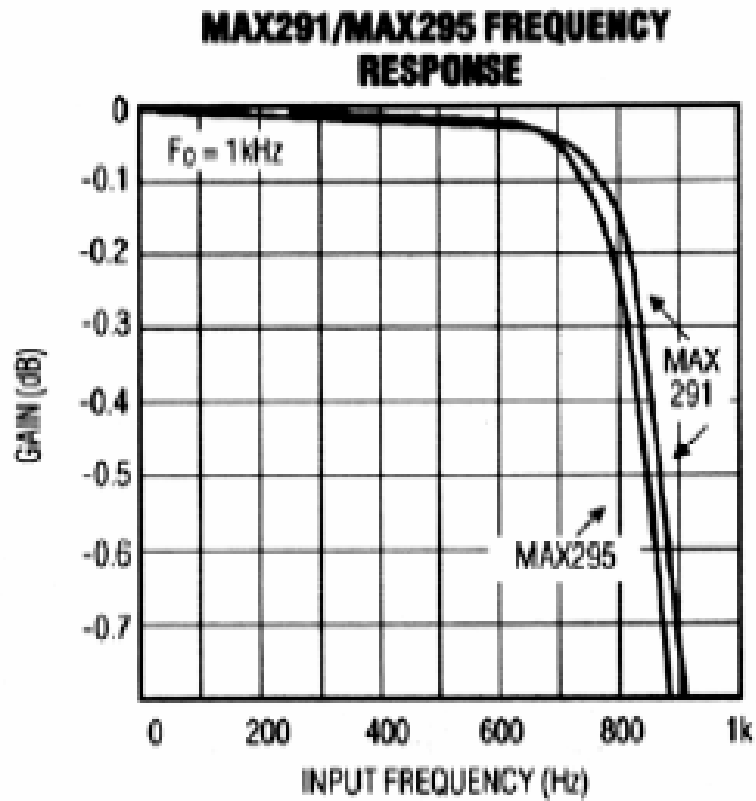


Figure A.18 – Theoretical Low-Pass Filter Response From MAX291 Datasheet [6]

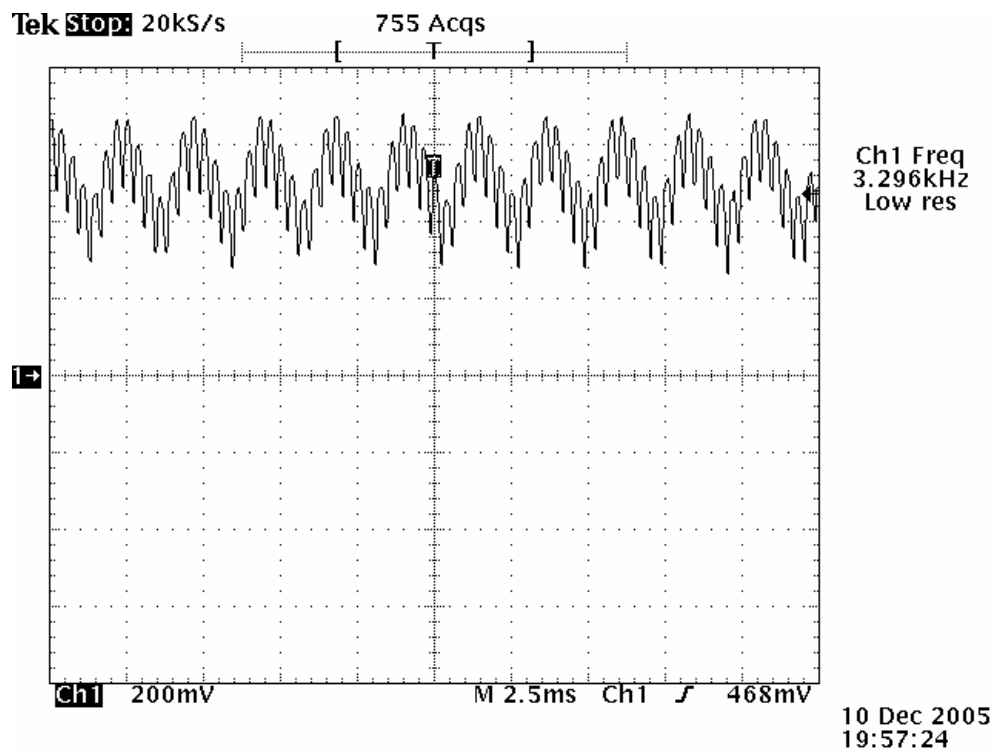


Figure A.19 – Generated Final Control Signal

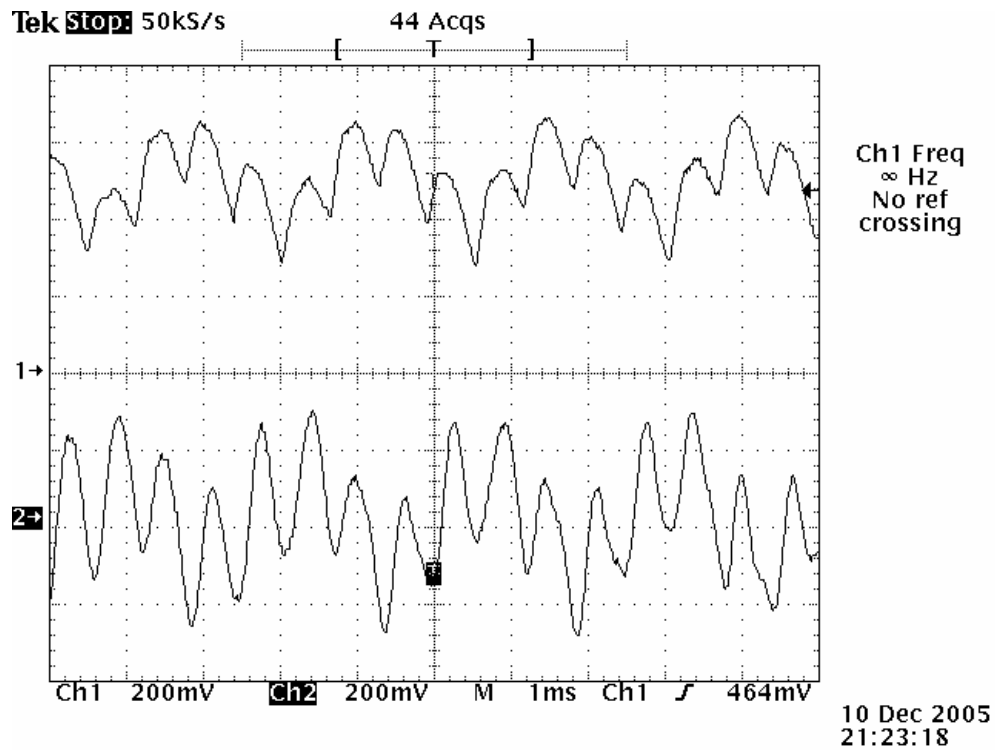


Figure A.20 – Signal Generated (Top) and Signal Received

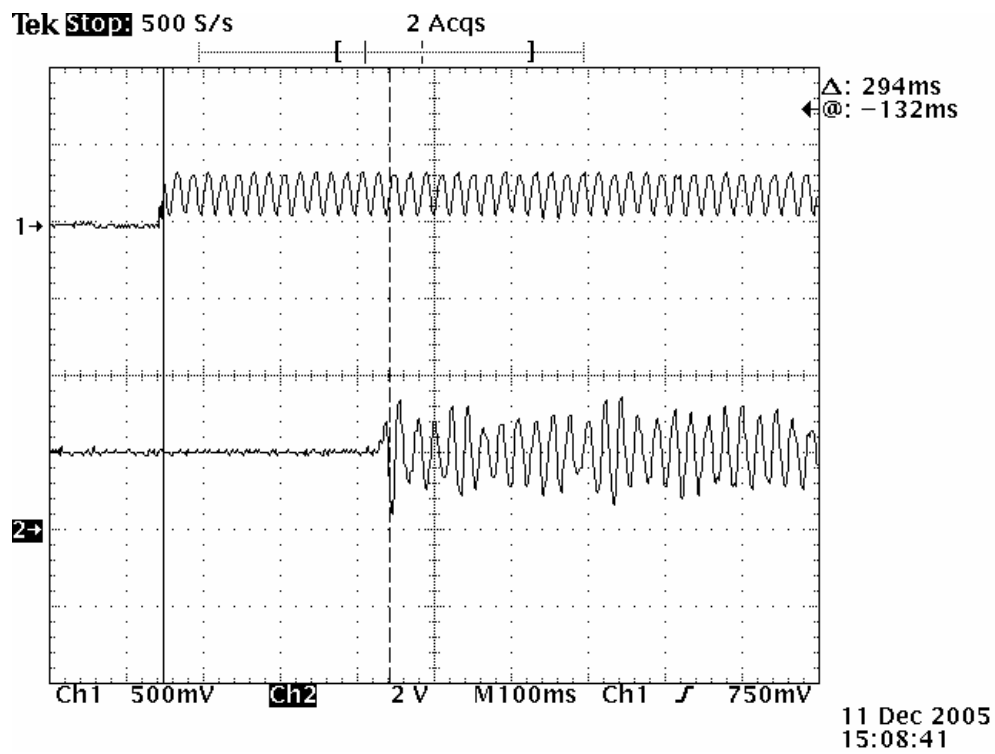
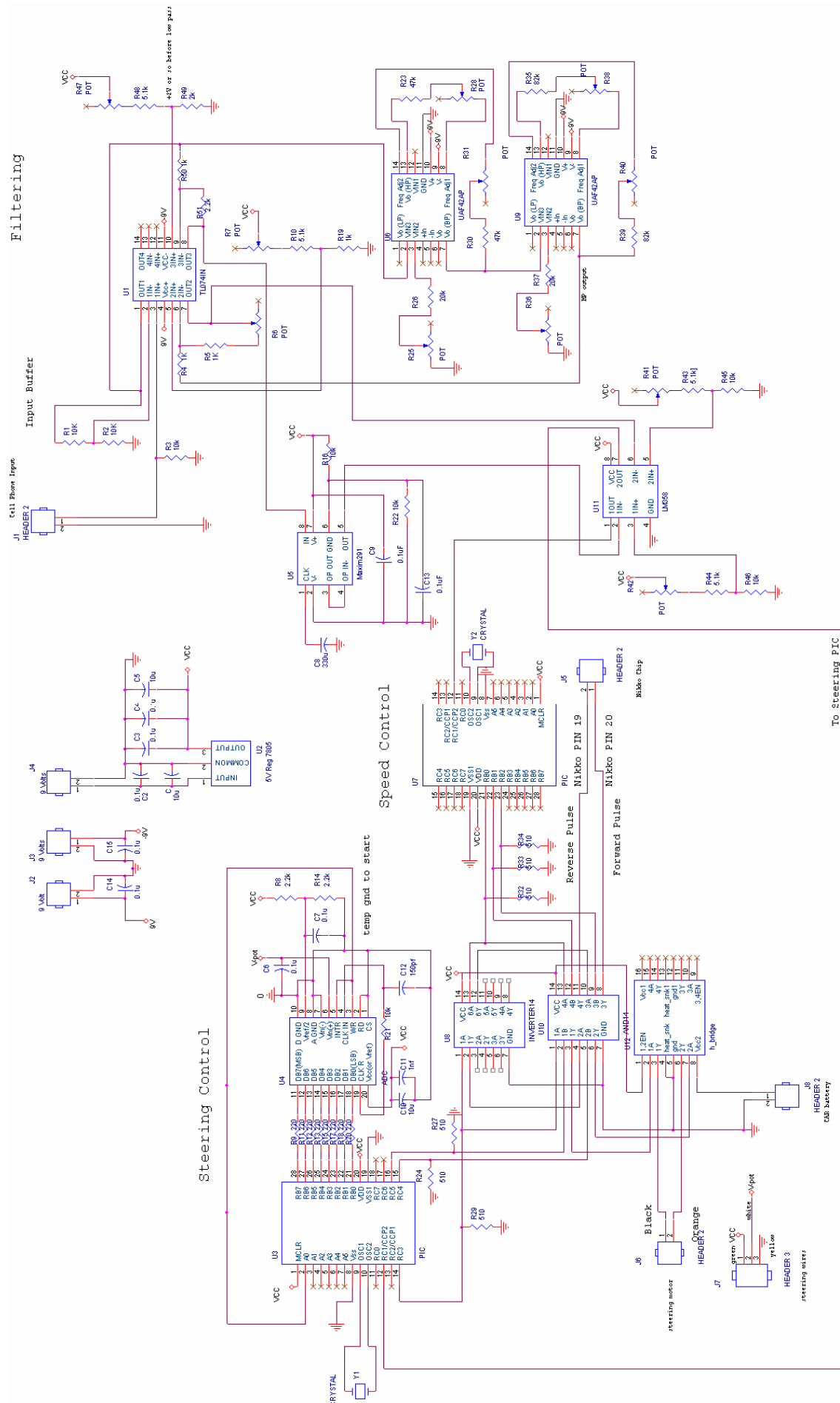
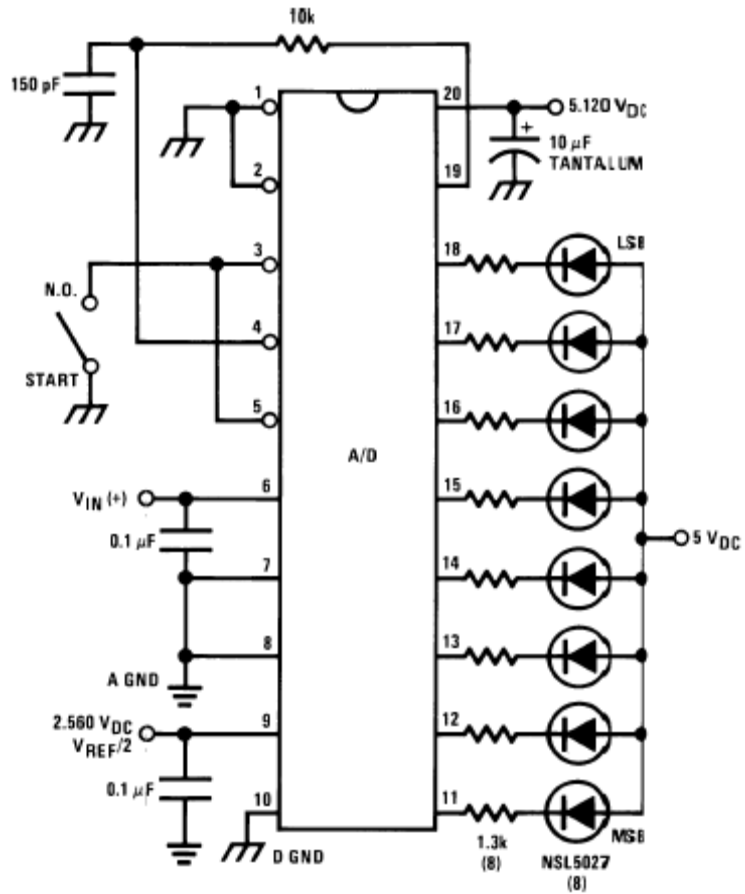


Figure A.21 – Latency of about 300ms between the Signal Generated and Signal Received





DS005671-18

Figure A.23 - ADC Setup [8]

APPENDIX B

Photographs of System and Connections



Figure B.1 – Joystick and Signal Generator Box

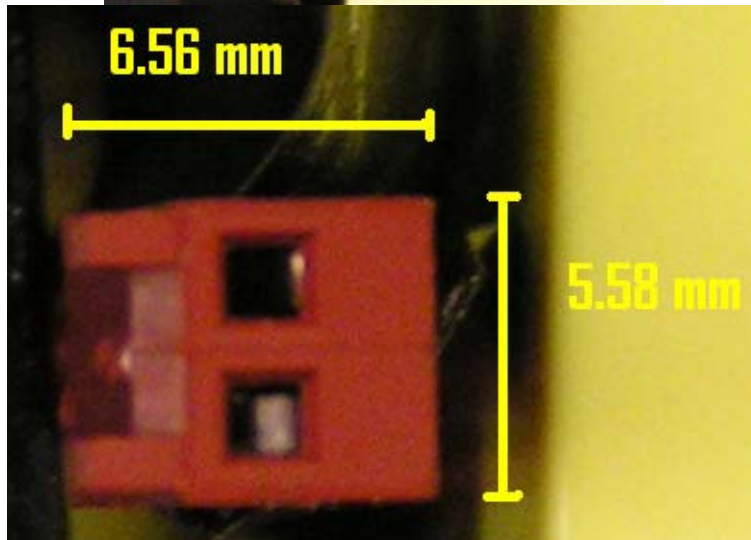
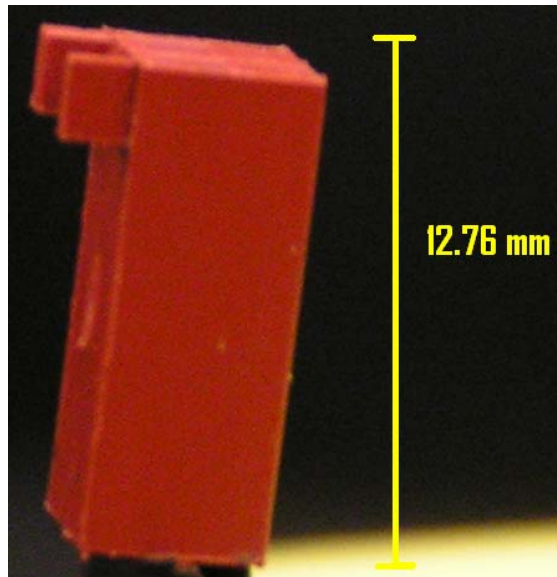


Figure B.2 – Female Two Pin Headset Cord to Circuit Connector

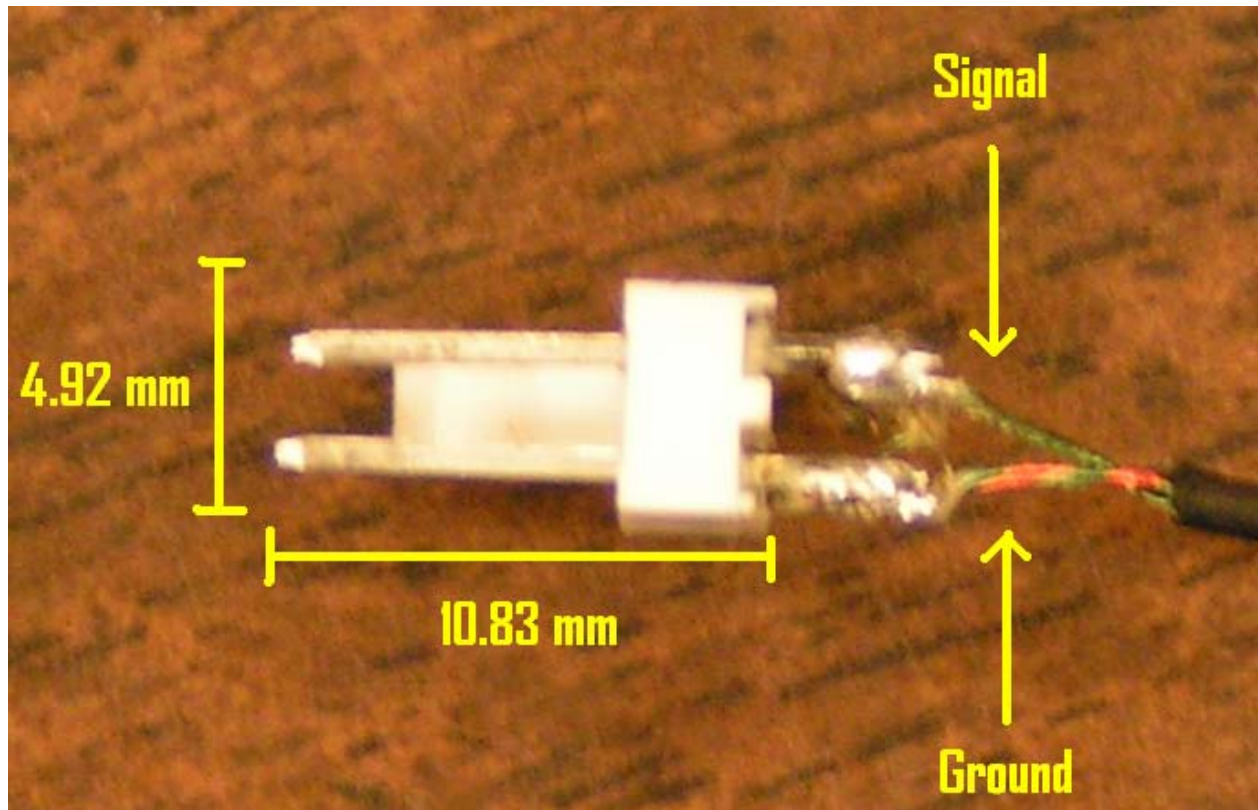


Figure B.3 – Male Two Pin Headset Cord to Phone Connector

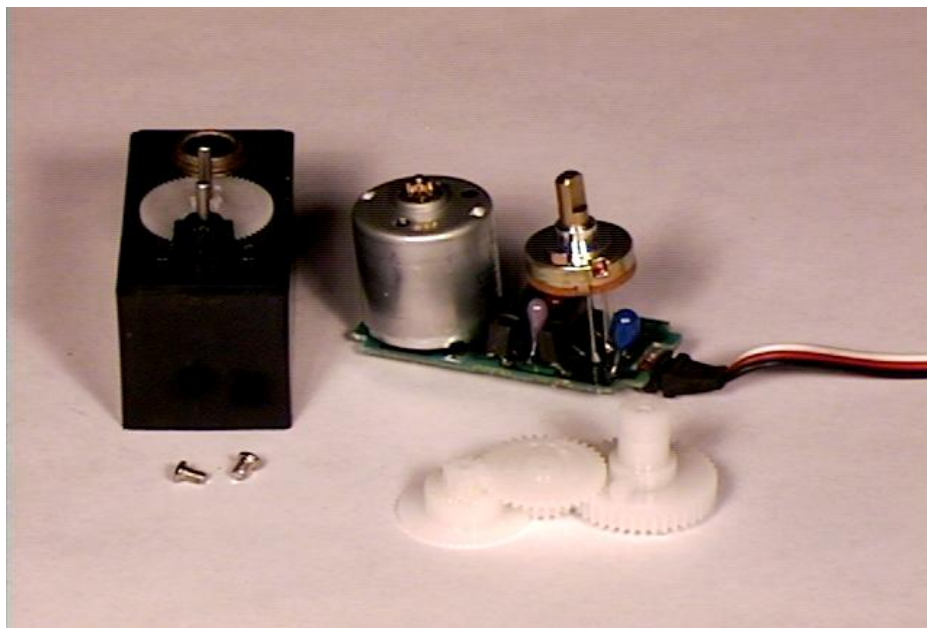


Figure B.4 - A Typical Servo Motor [7]

APPENDIX C

High-Pass Filter Design Program Screenshots

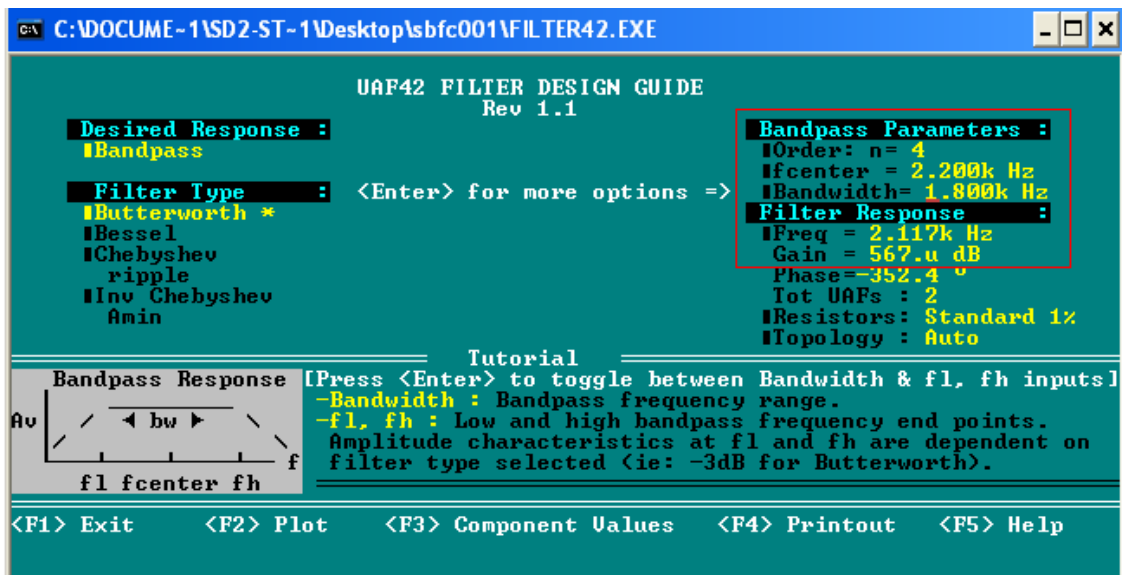


Figure C.1 – Band-Pass Filter with center frequency of 2.2 kHz [5]

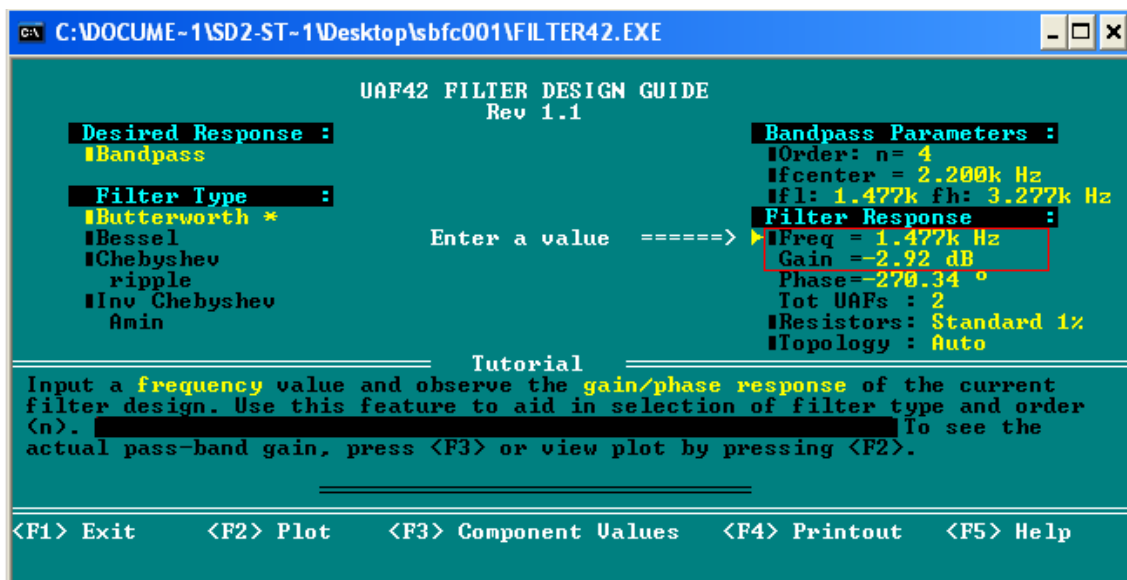


Figure C.2 – Program Indicates Nearly -3dB Attenuation at Low Cut-off Frequency [5]

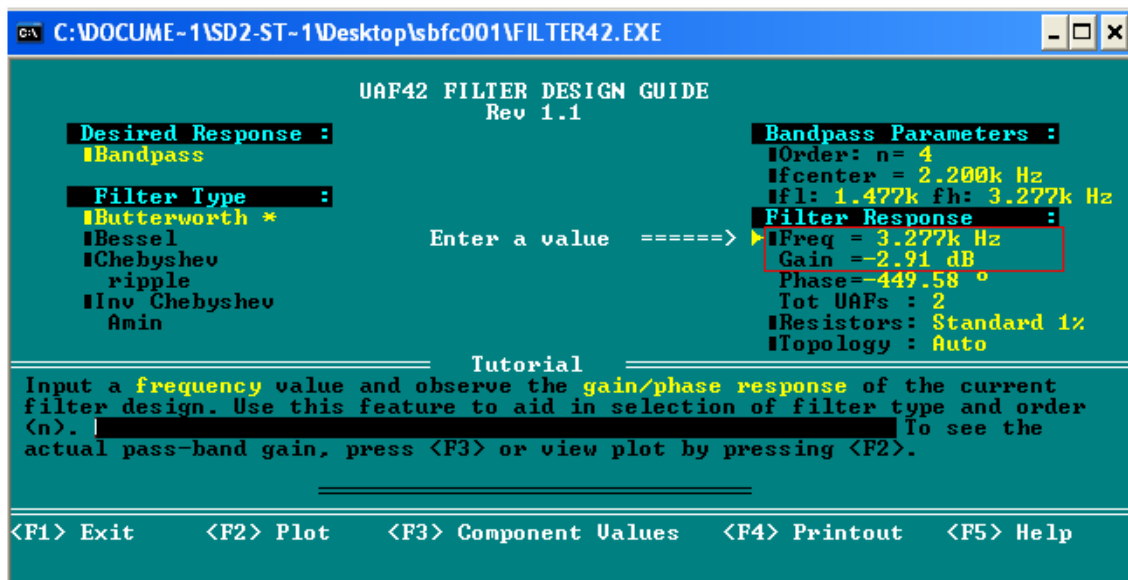


Figure C.3 – Program Indicates Nearly -3dB Attenuation at High Cut-off Frequency [5]

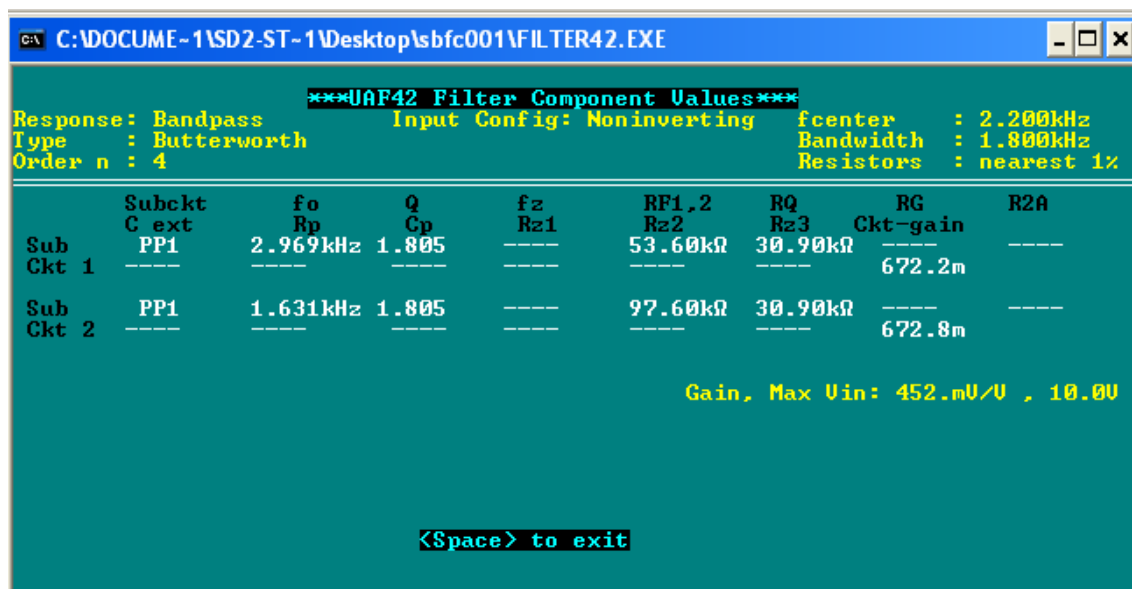


Figure C.4 – Resistance Values for Use with Figure A.15 to Build Filter [5]

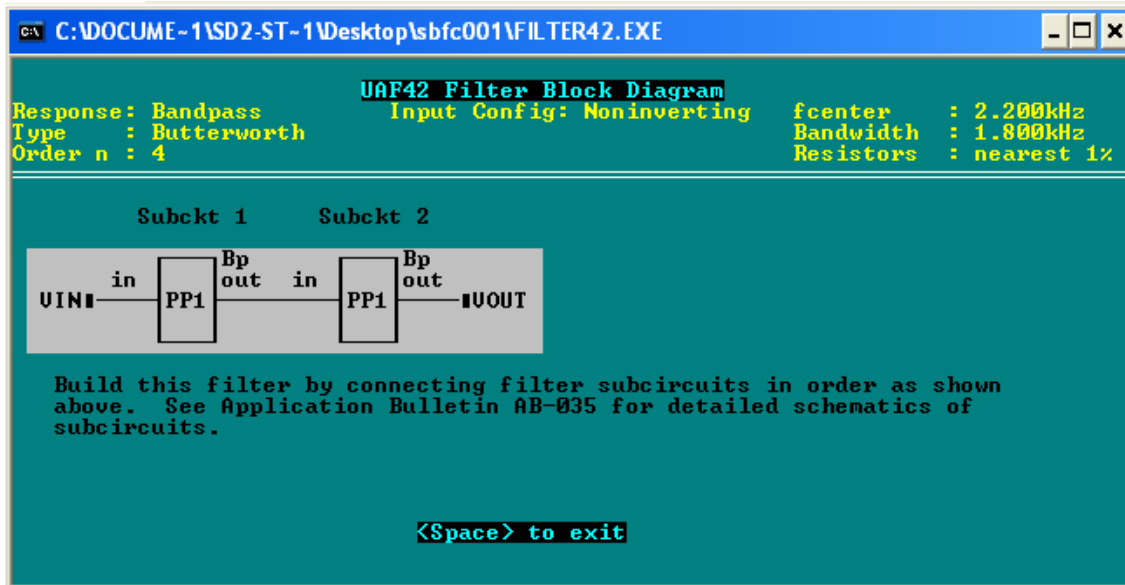


Figure C.5 – Cascade Layout to Obtain a 4th Order Filter Using Two UAF42As [5]

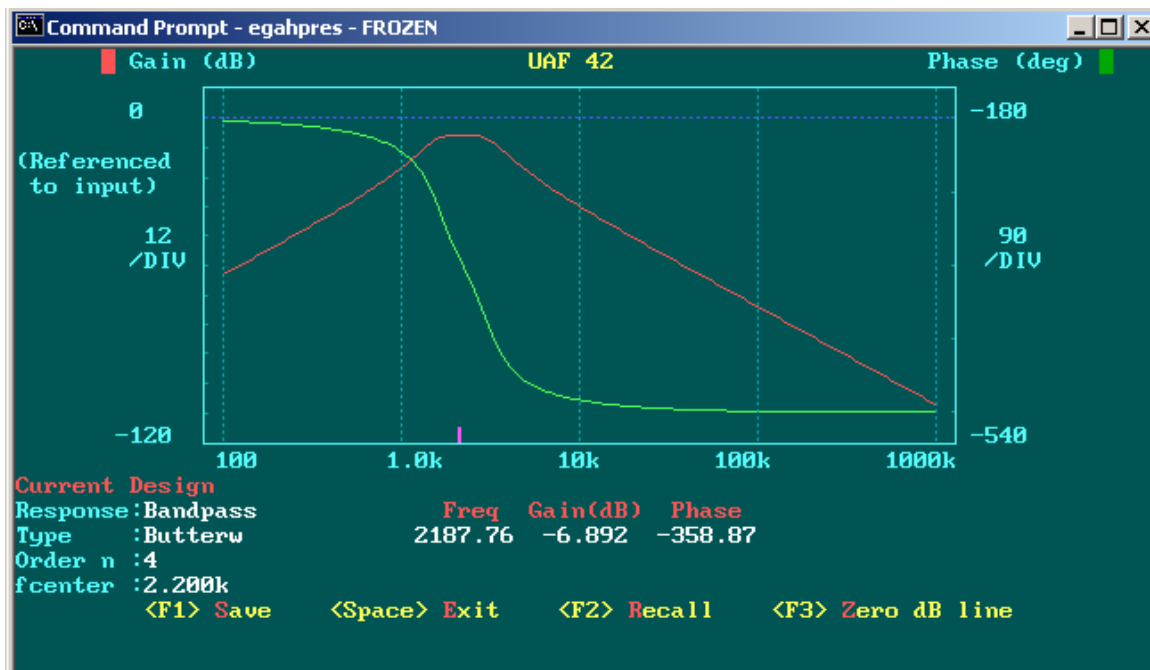
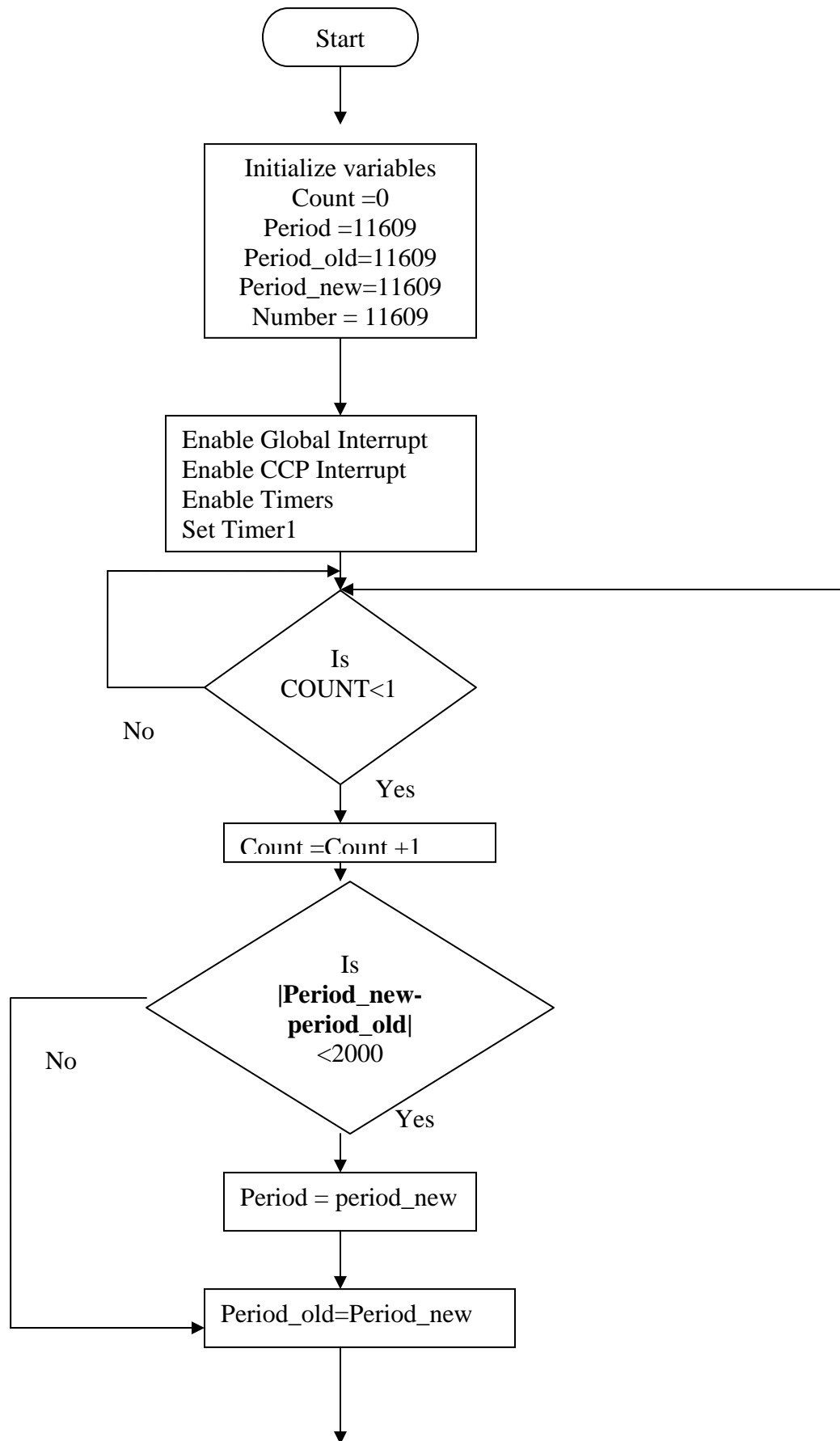


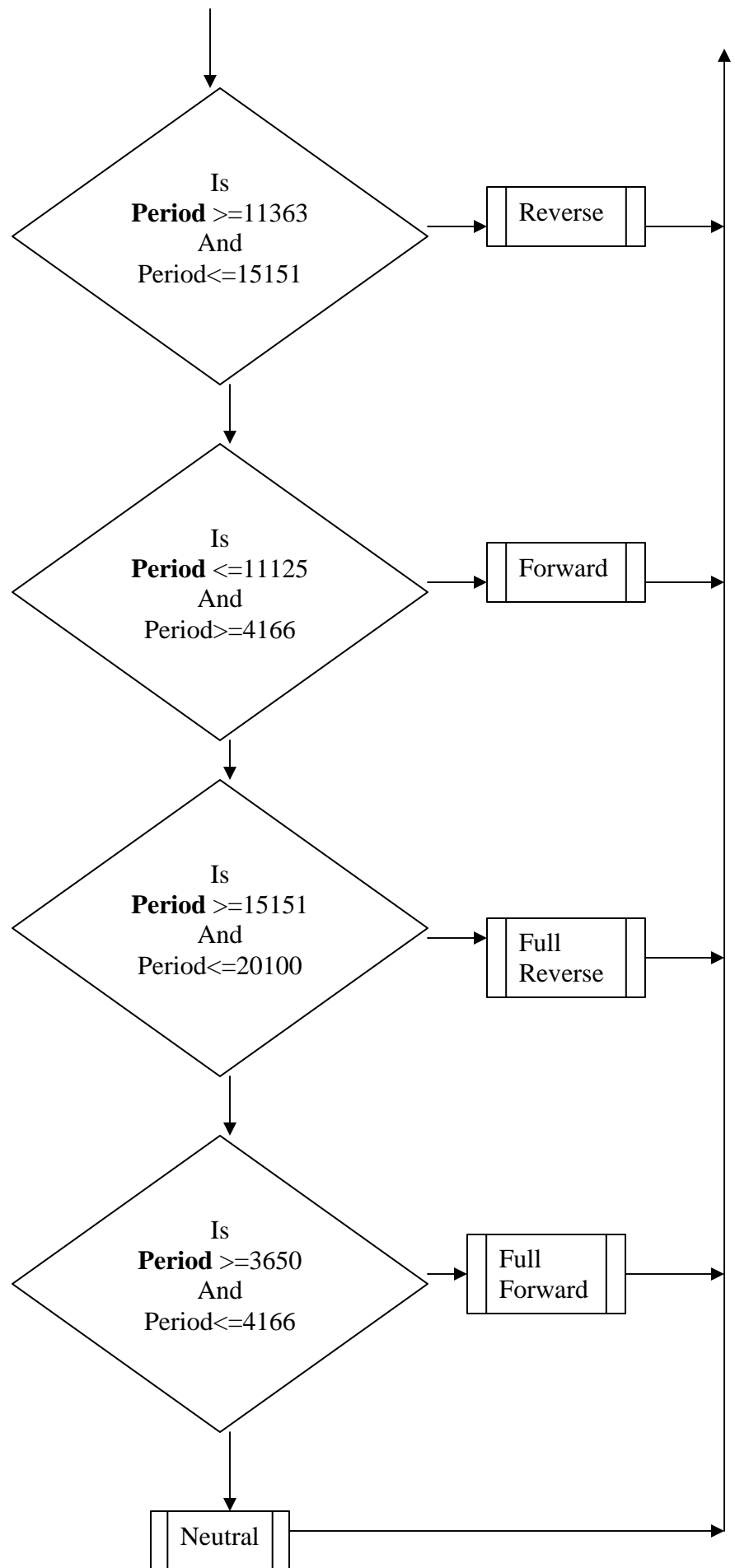
Figure C.6 – Theoretical Filter Frequency Response [5]

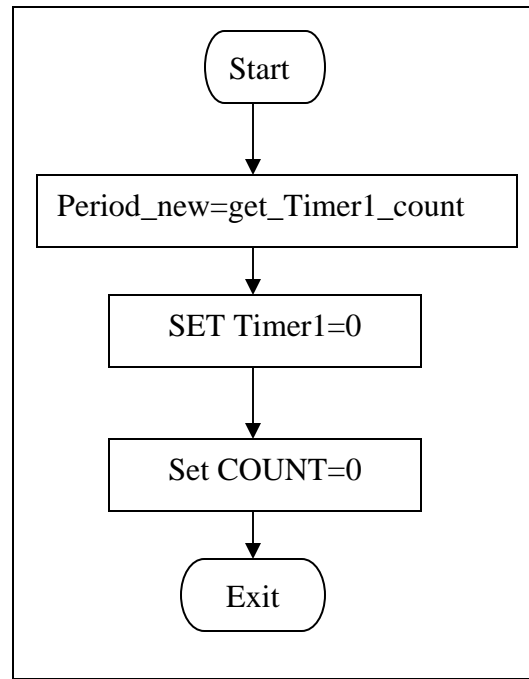
APPENDIX D

Software Flow Chart and Program Code for Speed and Steering Control

Speed Control Flow Chart

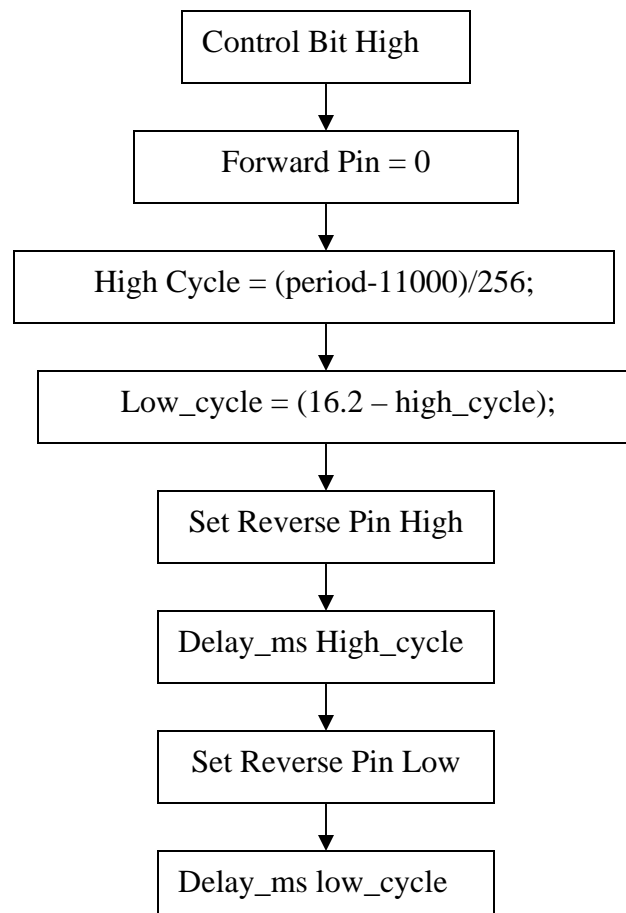




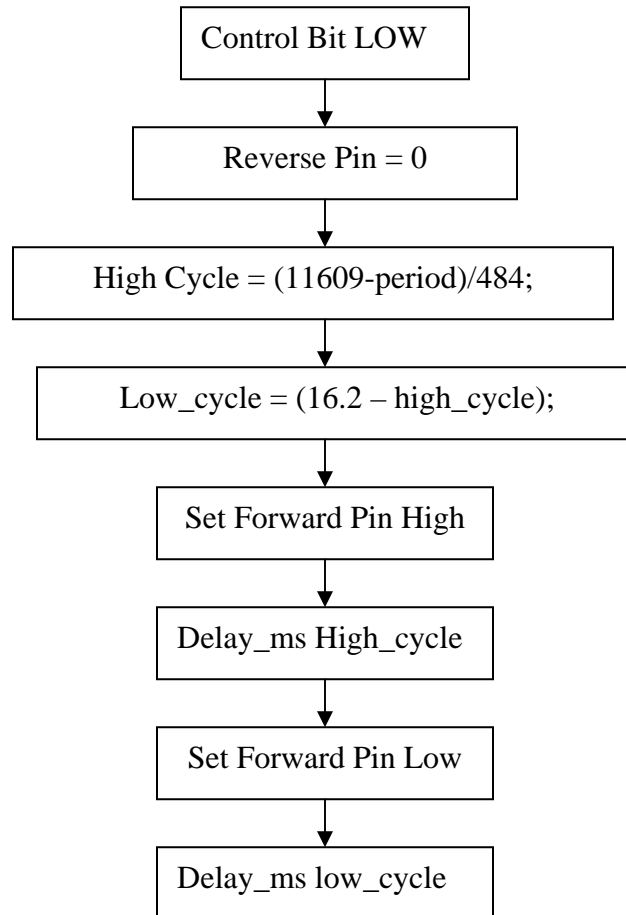


**INTERRUPT
SUBROUTINE**

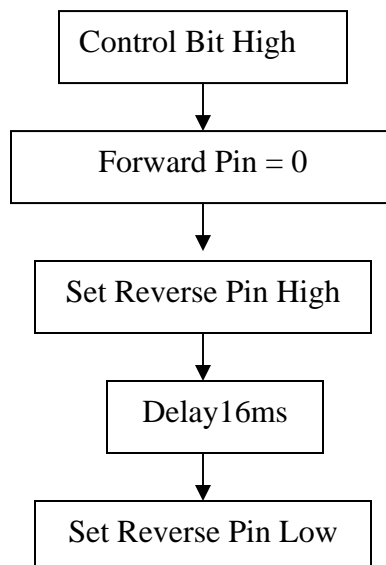
REVERSE Function



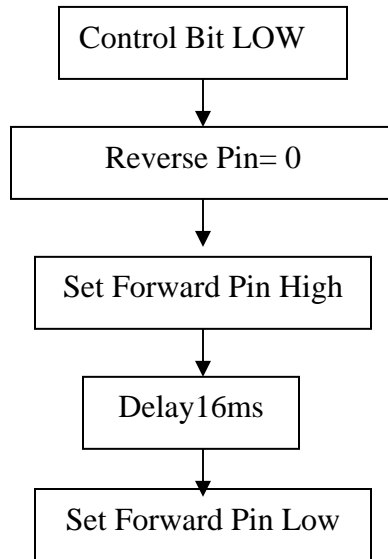
FORWARD Function



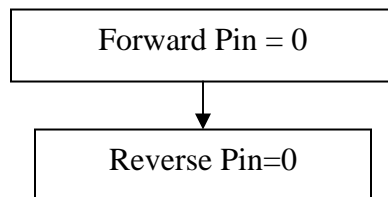
FULL REVERSE Function



FULL REVERSE Function



NEUTRAL Function



Speed Control Code

```
#include <16F876A.h>
//#include <16F876A.h>
#use delay(clock=20000000, RESTART_WDT)
#fuses HS, NOWDT, NOPROTECT, PUT, NOBROWNOUT, NOLVP
#include <STDLIB.H>
#use rs232(baud=9600,xmit=PIN_c6,rcv=PIN_c7)
#use standard_io(b)
#use standard_io(c)
port_b_pullups(TRUE);

// We are using "Timer Value" between the interrupts as our "Period"
// The actual relationship between them is, Period = Timer Value * 0.2 microsecond.

//global variables:
long period = 11609;           // period
long period_new = 11609;      // current period
long period_old = 11609;      // previous period
long high_cycle;              // high_cycle duration for a pulse
long low_cycle;               // low_cycle duration for a pulse
int count = 0;                // Set Master Count to 0

// Below is the INTERRUPT SUBROUTINE
//*****
#int_CCP2
CCP2_isr(){
    period_new = (CCP_2); //get period in terms of timer_1 counts
    set_timer1(1);        // Reset Timer1 to 0
    count = 0;            // Reset the Master Count to 0
}
//*****

//Below is the MAIN program which produces PWM pulses with Period of 16ms

void main()
{
    delay_ms(1000); //delaying 1 sec in the beginning to prevent any glitches
    set_tris_c(0xFF); // PORTC set as input
    set_tris_b(0x00); // PORTB set as output

    // Setting up timers and interrupts

    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1); // 200ns resolution
    setup_ccp2(CCP_CAPTURE_RE);             // interrupt on rising edge of c1
    enable_interrupts(INT_CCP2);             // interrupt on CCP2 rising edge
```

```

enable_interrupts(global);           // enable_interrupts(int_rb);

//=====
//Pins b1 is for REVERSE, b2 is for FORWARD and b0 is for CONTROL BIT
//=====

while (true){
  if(count<1){ //master if, runs so long as an interrupt has occurred

    count++; //sets the count to one. If it does not get an interrupt
             //it will do nothing

    // The following "if" reduces the effect of any drastic change in period
    // produced by glitch in the cell phone link or transmitter
    // If the difference between previous period and new period is greater than
    // 2000, it's a glitch, go with the previous period. If ok, store the new period

    if (((period_old-period_new)<2000)||((period_new-period_old)<2000))
    {
      period = period_new;
    }

    period_old = period_new;

//REVERSE.....REVERSE.....REVERSE.....REVERSE
// For frequency between 320 Hz and 440 Hz

    if ((number >= 11363) && (number < 15625)){ //frequency between
                                              //320 and 440Hz

      output_high(pin_b0);           //Control bit high, REVERSE set
      output_low(pin_b2);           // forward pin disabled
      high_cycle=((number-11300)/290);
      low_cycle=((16.2-high_cycle));
      output_high(pin_b1);
      delay_ms(high_cycle);
      output_low(pin_b1);
      delay_ms(low_cycle);
    }//ends if

// FORWARD.....FORWARD.....FORWARD.....FORWARD
// for frequency between 450 Hz and 1000Hz

    else if ((number <= 11125) && (number > 5000)){ //frequency between
                                              // 450 and 1000 Hz

```

```

    output_low(pin_b0);    //Control bit low, FORWARD set
    output_low(pin_b1);    // reverse disabled
    high_cycle=((11000-number)/400);
    low_cycle=(16.2-high_cycle);
    output_high(pin_b2);
    delay_ms(high_cycle);
    output_low(pin_b2);
    delay_ms(low_cycle);
} //ends else if

// FULL REVERSE.....FULL REVERSE.....FULL REVERSE.....
// Takes care of the fluctuations in the lower frequency region (250Hz<f<330 Hz)

    else if ((number >= 15625) && (number <20000)){ //frequency between
                                                //250 and 320Hz

        output_high(pin_b0);    //Control bit high, reverse set
        output_low(pin_b2);    // forward disabled
        output_high(pin_b1);
        delay_ms(16);
        output_low(pin_b1);
    } //ends else if

// FULL FORWARD.....FULL FORWARD.....FULL FORWARD.....
//Takes care of the fluctuations in the upperfrequency region ( 1000Hz<f<1100Hz)

    else if ((number <=5000) && (number > 4545)){ //frequency between
                                                //1000 and 1100Hz, full speed

        output_low(pin_b0);    //Control bit low, FORWARD set
        output_low(pin_b1);    // REVERSE disabled
        output_high(pin_b2);
        delay_ms(16);
        output_low(pin_b2);
    }

//NEUTRAL.....NEUTRAL.....NEUTRAL.....NEUTRAL
//NO OUTPUT, CAR STOPPED
    else{
        output_low(pin_b2); //FORWARD disabled.
        output_low(pin_b1); //REVERSE disabled
    } //ends else
}
}
}

```

Steering Control Logic Code

Steering Code

```
//#include <16F877A.h>
#include <16F876A.h>
#use delay(clock=20000000, RESTART_WDT)
#fuses HS, NOWDT, NOPROTECT, PUT, NOBROWNOUT, NOLVP
#include <STDLIB.H>
#use standard_io(b)
#use standard_io(c)
#define div 1/64
```

```
//port_b_pullups(TRUE);
```

```
//global variables:
long period=0;
int high_cycle;
int low_cycle;
int pot_value;
long desired;
int current;
int difference=128;
int count=0;
int i=0;
long value[5];
long sum=0;
int j;
```

```
#int_CCP2
CCP2_isr() {

    period=(CCP_2);
    set_timer1(1);

    if (i==5){
        i=0;
    }

    value[i]=period;
    i=i+1;
    count=0;
}
```



```

void main()
{ set_tris_a(0x00);          //set porta as output
  output_low(pin_a0);
  set_tris_c(0x02);          //c7-c2,c0 as output, c1 as input
  set_tris_b(0xFF);          // PORTB as input from ADC
  delay_ms(1000);
  set_tris_a(0xFF);          //set portA as input, no longer required, floating

// Setup timers and interrupts
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1); // 200ns resolution
setup_ccp2(CCP_CAPTURE_RE);             // interrupt on rising edge of c1
enable_interrupts(INT_CCP2);             // interrupt on CCP2 rising edge
enable_interrupts(global);               // enable_interrupts(int_rb);

// Main loop
while (true){

if(count<1){ //master if, runs so long as an interrupt has occurred

  count++; //sets the count to one. If it does not get an interrupt
           //before the loop completes
  sum=0;
  for(j=0;j<5;j++){
    sum=sum+value[j];
  }

  pot_value=input_b(); //value of pot voltage from ADC
  desired=(sum*div);
  if(pot_value<64){ //The minimum pot value was tested to be
    current=0;      //64, and the max was measured to be a
  }                 //little smaller than 192. If the values
  else if (pot_value>192){ //gets out of this range, current is set
    current=128;      //to the correct limit.
  }
  else{
    current=(pot_value-64);
  }
  difference=desired-current;

//=====
==
// c4 is left steering and c5 is right steering, c3 is control bit below

```

```

//=====
==
if ((desired>=128)&&(desired<=256)&&(current>=0)&&(current<=128)){
    //First if, does normal reactions if frequencies between 1.5kHz and
2.85kHz
    if(difference>=152){        //right steering on
        output_high(pin_c3);    //control bit high
        output_low(pin_c4);     //left steering off
        high_cycle=(12);
        low_cycle=4;
            output_high(pin_c5);
            delay_ms(high_cycle);
            output_low(pin_c5);
            delay_ms(low_cycle);
        }
    else if(difference>=136){
        output_high(pin_c3);    //control bit high
        output_low(pin_c4);     //left steering off
        high_cycle=((difference/8)-16);
        low_cycle=16-high_cycle;
            output_high(pin_c5);
            delay_ms(high_cycle);
            output_low(pin_c5);
            delay_ms(low_cycle);
        }

        else if (difference<=104){ //left steering on '104
        output_low(pin_c3);    //control bit = 0
            output_low(pin_c5);    //right steering off
        high_cycle=(12);
        low_cycle=4;
            output_high(pin_c4);
            delay_ms(high_cycle);
            output_low(pin_c4);
            delay_ms(low_cycle);
        }
    else if(difference<=120){
        output_low(pin_c3);    //control bit = 0
            output_low(pin_c5);    //right steering off
        high_cycle=(16-(difference/8));
        low_cycle=16-high_cycle;
            output_high(pin_c4);
            delay_ms(high_cycle);
            output_low(pin_c4);
            delay_ms(low_cycle);
    }
}

```

```

    }

    else if ((difference>120)&&(difference<136)){
        output_low(pin_c4);
        output_low(pin_c5);
    }
}

else if(((desired>256)&&(desired<275)){ //allows the frequency to go as low as
//1400Hz and still work, suggesting max right.
    if ((current<116)&&(current>10)){
        output_high(pin_c3); //control bit high
        output_low(pin_c4); //left steering off
        high_cycle=(8);
        low_cycle=8;
        output_high(pin_c5);
        delay_ms(high_cycle);
        output_low(pin_c5);
        delay_ms(low_cycle);
    }
}

else if(((desired<128)&&(desired>115)){ //allows the frequency to go as high as
//3.34kHz and still work, suggesting max left.
    if((current>2)&&(current<128)){
        output_low(pin_c3); //control bit = 0
        output_low(pin_c5); //right steering off
        high_cycle=(8);
        low_cycle=8;
        output_high(pin_c4);
        delay_ms(high_cycle);
        output_low(pin_c4);
        delay_ms(low_cycle);
    }
}
else{
    output_low(pin_c4);
    output_low(pin_c5);
}
}
}
}
}

```

APPENDIX E

Budget

This cost estimate excludes the cost of the vehicle (robot). An example of such a robot is an electric RC car which can range from 20 to 200 dollars.

Total Cost for System: \$233.64

| Component | Cost per Unit | Units in Transmitter | Units in Receiver | Total units used | Cost per total number of units |
|-----------------------------|---------------|----------------------|-------------------|------------------|--------------------------------|
| Standard Resistors | \$0.05 | 33 | 40 | 73 | \$3.65 |
| Capacitors | \$0.25 | 4 | 16 | 20 | \$5.00 |
| One-turn Potentiometers | \$1.00 | | 10 | 10 | \$10.00 |
| Sliding Potentiometers | \$0.97 | 2 | | 2 | \$1.94 |
| 5 Volt T220 Regulator | \$0.50 | | 1 | 1 | \$0.50 |
| 20 MHz Crystal (oscillator) | \$1.00 | | 2 | 2 | \$2.00 |
| UA747 | \$4.00 | 1 | | 1 | \$4.00 |
| LM348 | \$0.50 | 2 | | 2 | \$1.00 |
| LM741 | \$0.50 | 2 | | 2 | \$1.00 |
| LM358 | \$0.50 | | 1 | 1 | \$0.50 |
| TL074IN | \$0.50 | | 1 | 1 | \$0.50 |
| SN74LS04 | \$0.35 | | 1 | 1 | \$0.35 |
| SN74Ls08 | \$0.35 | | 1 | 1 | \$0.35 |
| UAF 42AP | \$10.62 | | 2 | 2 | \$21.24 |
| Max 291 | \$10.40 | | 1 | 1 | \$10.40 |
| ADC 0804 | \$2.00 | | 1 | 1 | \$2.00 |
| SN754410NE-1 | \$2.00 | | 1 | 1 | \$2.00 |
| PIC 16F876A | \$8.00 | | 2 | 2 | \$16.00 |
| End Mill Bits | \$11.00 | 0.5 | 0.5 | 1 | \$11.00 |
| Drill Bits | \$5.00 | 1.5 | 1.5 | 3 | \$15.00 |
| Router (Contour) Bit | \$9.00 | 0.5 | 0.5 | 1 | \$9.00 |
| 60 degree T1 | \$16.00 | 0.5 | 0.5 | 1 | \$16.00 |
| Milling Pad | \$3.00 | 0.5 | 0.5 | 1 | \$3.00 |
| Copper | \$6.00 | 0.5 | 0.5 | 1 | \$6.00 |
| Wire (purchased in rolls) | \$4.00 | 0.15 | 0.15 | 0.3 | \$1.20 |
| Solder (purchased in lbs) | \$15.00 | 0.02 | 0.02 | 0.04 | \$0.60 |
| 8 Pin Sockets | \$0.37 | 2 | 2 | 4 | \$1.48 |
| 14 Pin Sockets | \$0.37 | 3 | 5 | 8 | \$2.96 |
| 16 Pin Sockets | \$0.92 | | 1 | 1 | \$0.92 |
| 20 Pin Sockets | \$0.92 | | 1 | 1 | \$0.92 |
| 28 Pin Sockets | \$2.00 | | 2 | 2 | \$4.00 |
| Standard 9V batteries | \$1.94 | 2 | 1 | 3 | \$5.82 |
| 9.6V battery and charger | \$25.00 | | 1 | 1 | \$25.00 |
| Power Switches | \$3.00 | 1 | 1 | 2 | \$6.00 |
| Reset button | \$1.00 | 1 | | 1 | \$1.00 |
| 9V battery connectors | 0.25 | 2 | 1 | 3 | \$0.75 |
| 9.6V battery connectos | 2.99 | | 1 | 1 | \$2.99 |
| Joystick | \$8.99 | 1 | | 1 | \$8.99 |
| Case | \$5.41 | 1 | (made our own) | 1 | \$5.41 |
| Hands-free Cord | \$5.00 | 1 | 1 | 2 | \$10.00 |
| Cell-phone holder | \$5.00 | | 1 | 1 | \$5.00 |
| Heat Shrink | \$2.39 | 0.5 | 0.5 | 1 | \$2.39 |
| Stand-offs for case | \$2.79 | 0.5 | 0.5 | 1 | \$2.79 |
| Velcro Straps | \$2.99 | 0.5 | 0.5 | 1 | \$2.99 |

Figure E.1: Budget

APPENDIX F

Individual Contributions

Individual Contributions

Each team member wrote the sections of the report pertaining to his contributions (or where his name is noted in parentheses within the report.)

Mike

- Cell Phone Communication
- Sine Wave Generators
- Filter Design and Implementation
- Soldering
- Transmitter Box/Joystick

Justin

- Speed and Steering Software
- Integration and Debugging
- CAD Layout Design
- 555 and Frequency-to-Voltage Circuits for Early Testing
- Soldering

Raj

- Speed Control Logic
- Comparator Circuit
- CAD Layout Design
- Integration and Debugging
- Speed and Steering Software

Richard

- Fourth Order Low Pass and High Pass Butterworth Filters Used for Early Testing.
- Milled All Boards
- Majority of Soldering
- Case Construction for the Receiving Circuit
- Help on Transmitting Signal Between the Phones

APPENDIX G

Mentor Meeting Minutes

Date: 08/29/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 1

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

This meeting was an introductory meeting where the faculty mentor, Dr. Latino. The team had some questions about the design requirements and what was expected to be done. Dr Latino answered them and further provided the team some useful insight on his 50/90 rule emphasizing the project needed to be done as early as possible to have enough room at the end to work with any kind of last stage problems. A deadline was set for the project completion and the days were divided into Yellow Zone, Green Zone and Red Zone. The yellow zone emphasized that the team had to decide on a design by the end of the zone. The green zone emphasized that the team need to have a working system by the end. The red zone emphasized that the team need to have done all necessary debugging and completed the project.

The goal for the next week was to research and come up with alternative solutions to the project.

Dr. Carl Latino

Date: 09/07/2005 (Wednesday)

Time: 4:00pm-5:00pm

Meeting No:2

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the important points discussed in the meeting.

1. The speed and steering of the RC car needed to be fully controllable.
2. A variable frequency generator, which could be easily adjusted, would help for transmitting signal. This would allow analog and a wide range of frequencies.
3. Two sets of frequencies would be used, one for speed and one for steering. This allows 2 degrees of freedom.
4. Filters would be used on the receiving end to filter out low and high frequencies.
5. The joystick for the sending side could be built.
6. The connection between the cell phones will be manually made on both sides.

The task needed to be completed by next week.

1. Build 555 timer circuit whose frequency can be adjusted with a single pot.
2. Find the bandwidth of the cell phone.
3. Look for a car that has fully proportional speed and steering control.
4. Do some research on filters.
5. Come up with a basic solution approach.

Dr. Carl Latino

Date: 09/12/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No:3

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

.

The following were the main points discussed.

1. A 555 timer circuit did not produce a 50% duty cycle square wave and thus would result in large number of harmonics at the other end. A flip flop could be used with the 555 timer circuit to make a 50% duty cycle wave and this would reduce the number of harmonics.
2. A pure sinusoid wave would be a better idea in terms of reducing the number of harmonics.
3. A summer circuit was needed to add the two sinusoids.
4. Frequency range on the cell phone was about 300Hz to 3100 Hz.
5. Frequency-to-voltage converters could be used on the receiving end to convert frequency to voltage and then use the voltage.
6. A test-setup between the cell phone was required to see how waves were received on the receiver side.

Task needed to be completed by next meeting.

1. Frequency-to voltage converter circuit.
2. A pure sine wave generator.
3. A summer circuit.
4. A test-setup between the cell phones to send and measure signals.

Dr. Carl Latino

Date: 09/19/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No:4

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. Frequency-to-voltage circuit was not working very well.
2. The summer circuit worked well.
3. The filter circuits were in process of being built.
4. A satisfactory pure sinusoid generator could not be built. Tried using 50% duty cycle square wave and sending it through filters. Results not very good.
5. Team had problems with sending sine waves through cell phone and receiving end. The signal received was not stable and sometimes was just noise.
6. The input signal probably needed to be buffered as the cell phone would work as load to the function generator.
7. Team had one more week before the “Yellow Zone” would end.

The tasks needed to be completed by next meeting.

1. Have the frequency-to-voltage circuit working.
2. Have the filters working.
3. Have the sine-wave generators working.
4. Be able to send sine waves through the cell phone.
5. Have idea of how speed and steering worked. Specially how the servo worked,

Dr. Carl Latino

Date: 09/26/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 5

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. Still having problems receiving sinusoids in the receiving end.
2. Frequency-to-voltage converter circuit did not work well at all. A new idea for measuring frequency was designed. It was to use a PIC and count the zero-crossings.
3. Filters working very well.
4. Still having problems generating pure Sine wave.
5. Looked up some Pseudo-Sine wave generators on internet. Had ordered the parts to make it.
6. Research done on Speed and Steering Control for the car. Speed control used simple PWM signals. Steering used PWM too but had a feedback from the potentiometer in the servo.

Following things needed to be accomplished before next meeting.

1. PIC program that would calculate the frequency.
2. Signal transmission between cell phones.
3. Research on Phase-Locked Loops (could be used for servo control).
4. Pure Sine wave generator.

Dr. Carl Latino

Date: 10/03/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 6

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. The PIC program done. It could calculate the frequency. Next step would be to convert to PWM's.
2. Pseudo-Sine wave generator built. Worked ok.
3. Still had trouble sending waves to the other end.
4. Working of Servo System researched. However, burned the on-board chip. So would need to design the servo system ourselves.
5. Designed new Low Pass filters. IT used a MAXIM chip and was 8th order low pass filter. Worked very good and had a better attenuation.
6. Also designed a new High Pass. It works better than the previous.

Tasks needed to be done.

1. Complete and demonstrate the Speed Control System.
2. Complete and demonstrate the Steering Control System.
3. Try to mill boards and get rid of all the wires.
4. Have a wired system working and demonstrate.
5. Get the signal transmission portion working.

Dr. Carl Latino

Date: 10/10/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 7

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. The Speed Control system was working.
2. The Steering Control system was ok. The code needs to be improved.
3. The filters work well.
4. The wired system works ok except for the steering,
5. Planning to mill the boards.
6. Echo Cancellation could be a problem with the transmission of signal. Research needed to be done.

Tasks needed to be completed:

1. Fully working wired system set-up.
2. Mill the boards.
3. Fix the steering system.
4. Research Echo-Cancellation.

Dr. Carl Latino

Date: 10/17/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 8

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. Found a solution to the signal transmission problem. Amplitudes of the signals were too big. Had to bring in down to about 500 mV. Worked good. Could control the speed with the cell phone connection.
2. Steering System working good too.
3. Wired system working good.
4. Need to make adjustments to the wired system to have the wireless system working.
5. Boards milled. Needed to put components.
6. Working on transmitter. Making adjustment to the potentiometers to have the desired signals outputted.

Tasks needed to be completed:

- 1 Have everything assembled.
2. Have a working system. Do necessary debugging.
3. Have a working transmitter.

Dr. Carl Latino

Date: 10/24/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No:9

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. The receiver board was ready. Still doing some debugging.
2. Transmitter working ok. Needed some adjustments to the pot.
3. Steering system twitching a little i.e. unstable. Working on making the code better and reducing effects of noise.

Tasks needed to be completed.

1. Make the code better.
2. Finish debugging.
3. Make cases for transmitter and receiver.
4. Start writing the final report.

Dr. Carl Latino

Date: 10/31/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No:10

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. The system working very well.
2. Some changes in the program code to make it better.
3. Working on the cases for transmitter and receiver.

Task needed to be completed.

1. Finish the cases.
2. Make the program code better.

Dr. Carl Latino

Date: 11/07/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No:11

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. The team had a demonstration. Worked well.
2. Team decided to reduce the number of power supplies. Instead of using four 9 volt batteries, use the 7.2 volts battery that came with the car to power the steering H-Bridge. Reduce the number of 9 volt battery to 3.

Tasks needed to be completer.

1. Finish putting switches and finishing touches.
2. Reduce the number of batteries.

Dr. Carl Latino

Date: 11/14/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 12

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. Reduced the number of batteries (on the receiver side) to three 9 volts and one 7.2 volts.
2. The overall system worked well.

Tasks needed to be completed:

1. Finishing touches.

Dr. Carl Latino

Date: 11/21/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 13

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

1. Reduced the number of batteries to one 9.6 volts, one 9 volt and one 7.2 volt(came with the car)
2. Burned some high pass filter chips. Did not have extras. So ordered them.

Tasks needed to be completed:

1. Finishing touches.

Dr. Carl Latino

Date: 11/28/2005 (Monday)

Time: 4:00pm-5:00pm

Meeting No: 14

Team Members: Raj Sharma Justin Schaper Mike Area Richard Lienou

Faculty Mentor: Dr. Carl Latino.

The following were the main points discussed.

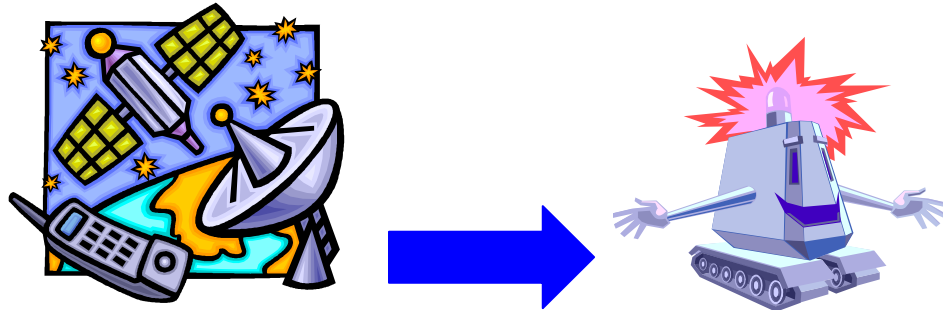
1. Prepare for the Project Inspection. Preparing poster and slides.
2. Prepare final report.
3. Prepare for the oral examination.

Dr. Carl Latino

APPENDIX H

Status Reports

Status Report 1



ROBOTCELL INC.

Presents

REMOTE CONTROL ROBOT WITH CELL PHONE LINK

Contributors:

Sudhendu Raj Sharma
Richard Lienou
Michael Area
Justin Schaper

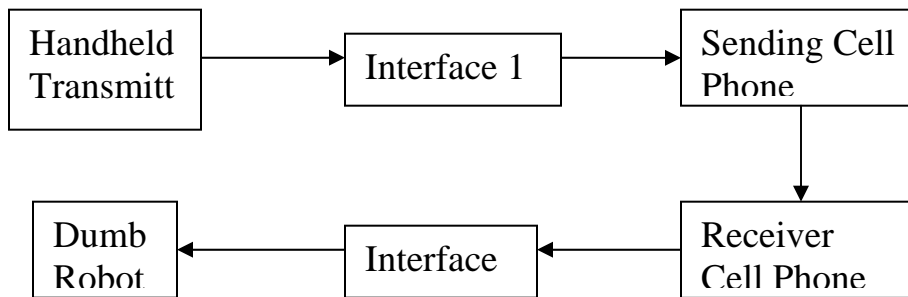
PROJECT STATEMENT

- Design and build a system which will be used to remotely control a “dumb” robot through cell phones.
- The purpose of the cell phone link is to allow the robot to be operated (open loop) from a remote location.
- When the project is completed, the robot should operate the same way like a similar off-the-shelf unit without the cell phone link
- The two cell phones must not be modified.

SPECIFICATIONS

- The system must have an interface between the transmitter and the sending phone and another interface between the receiver phone and the robot.
- The cell phones should be any common cell phone. However, a specific model can be chosen because of handsfree set connection type.
- Very negligible delay compare to the operation of the off-the-shelf unit one
- The transmitter must have two independent channels to control the speed and the steering of the robot
- The robot should not communicate back to the transmitter and make decisions (open loop)
- The system should be a low power device
- The final product should look good (presentable)

BLOCK DIAGRAM



SOLUTION

- We decided to build our own transmitter which will consist of two frequencies generators (oscillators) with frequencies in the range of 200Hz–3000Hz.
- The two frequencies channels will have ranges between 130Hz -1400Hz and 1500Hz – 3000Hz respectively.
- We will use potentiometers to control the frequencies.
- **Interface 1** will just be wires/cables connecting the transmitter to the handsfree set jack of the sending phone.
- Interface 2 will be a control unit box with two subsystems:
 - Subsystem A will filters to separate the two frequencies channels received from cell phone 1
 - Subsystem B will convert frequencies to for example voltages and codes to control the motor speed and the robot direction.

LIST OF TASKS ACCOMPLISHED

- Researched possible solutions for the project
- Wrote and sign contract
- Completed the task list
- Wrote the project description
- Purchased a RC car and examine its transmitter circuit
- Designed and build one oscillator with frequency ranging from 131Hz to 1443Hz
- Tested the transmission of one frequency channel through the cell phones

LIST OF TASKS TO BE ACCOMPLISHED BY NEXT STATUS REPORT

- Design and build the second oscillator with frequency ranging from 1500 Hz – 3000Hz
- Design and build a frequency to voltage converter
- Design and build the converter circuit to control the direction of the robot
- Design and build filters to separate the two sets of frequencies (two channels)
- Have two cell phone available for transmission test
- Prepare and submit the design review
- Prepare status report 2

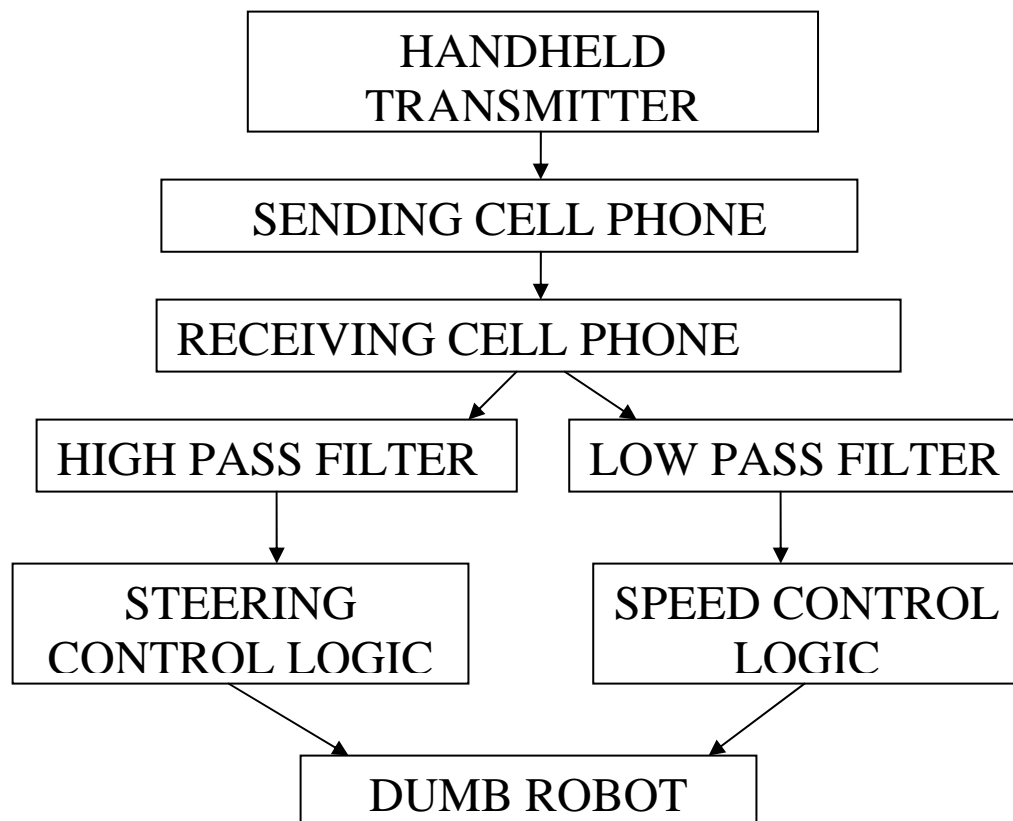
| Week | August | | September | | | | October | | | | November | | | | December | | |
|-----------------------------------------------------------------------------------------------------------|----------|----------|-----------|----------|----------|----------|---------|----------|----------|-----------|-----------|----------|-----------|-----------|-----------|----------|-----------|
| Task | wk1 (22) | wk2 (29) | wk3 (5) | wk4 (12) | wk5 (19) | wk6 (26) | wk7 (3) | wk8 (10) | wk9 (17) | wk10 (24) | wk11 (31) | wk12 (7) | wk13 (14) | wk14 (21) | wk15 (28) | wk16 (5) | wk17 (12) |
| Write up and sign Contract | | | | | | | | | | | | | | | | | |
| Complete Task List with Individual Assignments | | JS/MA | | | | | | | | | | | | | | | |
| Write Project Description | | | | | | | | | | | | | | | | | |
| Purchase RC car with Analog controls | | | MA | | | | | | | | | | | | | | |
| Design Oscillator Circuit with adjustable frequency output | | | RS JS | | | | | | | | | | | | | | |
| Prepare 1st Status Report | | | RL | RL | | | | | | | | | | | | | |
| Deliver Oral part of Report | | | | RL | | | | | | | | | | | | | |
| Mock Evaluation | | | | | | | | | | | | | | | | | |
| Determine how to control motors. If separate servo control is needed, more tasks will be added | | | | RS | | | | | | | | | | | | | |
| Build and test oscillator circuit and microphone recognizability | | | | RS/JS | RS/JS | | | | | | | | | | | | |
| Assuming successful test, optimize circuit and design matching receiving circuit and controls for RC car. | | | | RL/MA | RL/MA | | | | | | | | | | | | |
| Design Review | | | | | RS | | | | | | | | | | | | |
| Prepare 2nd Status Report | | | | | | MA | MA | | | | | | | | | | |
| Deliver Oral part of Report | | | | | | | | MA | | | | | | | | | |
| Formal Evaluation #1 | | | | | | | | | | | | | | | | | |
| Build and test entire communication link and motor control | | | | | | | | | | | | | | | | | |
| If not already done, build second slightly modified system for the second channel, (steering/speed) | | | | | | | | | | | | | | | | | |
| Lots more testing and tweaking | | | | | | | | | | | | RL | RL | RS | | | |
| Prepare 3rd Status Report | | | | | | | | | | JS | JS | | | | | | |
| Deliver Oral part of Report | | | | | | | | | | | | JS | | | | | |
| Formal Evaluation #2 | | | | | | | | | | | | | | | | | |
| Project Inspection | | | | | | | | | | | | | | | | | |
| Oral Examination | | | | | | | | | | | | | | | | | |
| Final Report | | | | | | | | | | | | | | | | | |
| Senior Exit Survey | | | | | | | | | | | | | | | | | |
| Mike Area | | | | MA | | | | | | | | | | | | | |
| Richard Lienou | | | | RL | | | | | | | | | | | | | |
| Justin Schaper | | | | JS | | | | | | | | | | | | | |
| Raj Sharma | | | | RS | | | | | | | | | | | | | |
| All | | | | | | | | | | | | | | | | | |
| Prepare By | | | | | | | | | | | | | | | | | |
| Verified by all | | | | | | | | | | | | | | | | | |

Note: Those tasks marked as "All" will be broken into sub-tasks assigned to individuals

REMOTE CONTROL ROBOT WITH CELL PHONE LINK

**Justin Schaper
Richard Lienou
Michael Area
Sudhendu Raj Sharma**

Block Diagram of Final System



Completed Tasks

- Available bandwidth for signal transmission

300Hz to 2900Hz

(300 Hz – 1500 Hz and 1700 Hz – 2900 Hz)

- Signal between phones
- Op amp pseudo sine wave generator

Range 6-7000 Hz

- Summing amplifier circuit

Adds the low and high channels for simultaneous transmission

- Low pass filter

1500 Hz cut-off frequency

- Speed control

PIC receives square wave, outputs PWM of varying duty cycle

Logic circuit determines forward/reverse

Current/Future Tasks

- Joystick with potentiometers to control the sending frequency

300 – 1500 Hz for Y-Axis

1700 – 2900 Hz for X-Axis

- Square wave at receiver for PIC

Comparator and/or filter

- Control system for steering

PIC to convert frequency to desired turning angle

- Integration of sub-systems

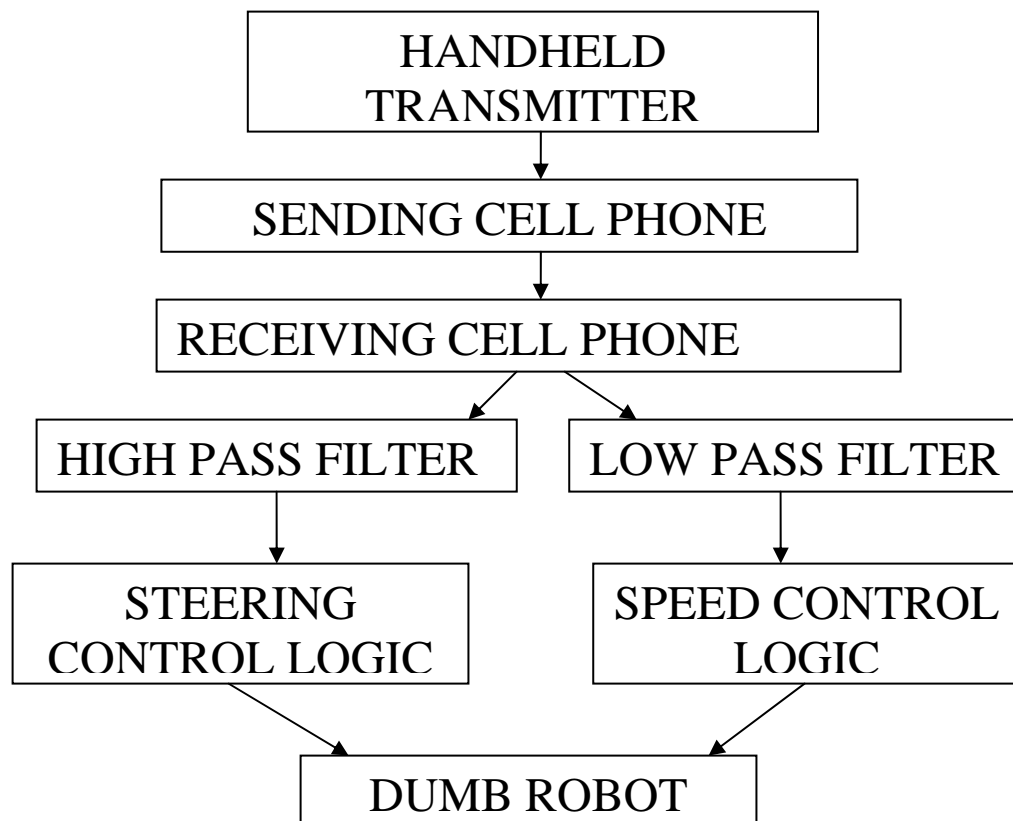
- PCB milling

[illegible]

REMOTE CONTROL ROBOT WITH CELL PHONE LINK

**Justin Schaper
Richard Lienou
Michael Area
Sudhendu Raj Sharma**

Block Diagram of Final System



Tasks Completed Since the Last Status Report:

- Joystick with potentiometers to control the sending frequency

300 – 1000 Hz for Y-Axis

1500 – 3100 Hz for X-Axis

- Filtering and comparing system that takes the “messy” signal from the cell phone and creates a square wave with the same frequency as that of the controlled signal.

- Control system for steering:

A PIC program is used that converts frequency measurements to a desired turning angle. Using a pot mounted on the steering servo, a reading of the current angle is read into the PIC with an ADC. It then calculates how fast and in what direction to turn the steering motor based on the relationship of the current angle to the desired angle.

- Integration of sub-systems: The transmitting system creates and combines the signals. The receiving system buffers and filters the signal into two channels. The signals are then converted to square waves and sent to the PIC for measuring. The PIC then controls the speed and steering angle of the car with H-bridge circuits connected to the front and rear motors.
- PCB milling: The transmitting has been completed. The receiving board has been milled and is being soldered and tested.

Tasks yet to be completed:

- Finish soldering and testing the receiving board and final design.

- Debug the cell phone connection:

Sometimes the signal does not transmit right away or is lost after connection is made. Loading the input momentarily with a speaker or small resistor causes the signal to begin transmitting. However, leaving the load on the signal causes a loss of the connection.

- Smooth out controlling and fix motor “twitching”.

The square waves sent to the PICs are not perfect and the PIC measured value may not always be correct. Although the error only occurs for a very short period of time, it causes the PICs to momentarily output the incorrect signal to the motors. Also, the joystick does not always return to the exact center position.

- Reconstruct the car with modifications to carry the PCB, batteries, and cell phone while still looking presentable.
- Final Report

Specs at a Glance

1. Final system is indistinguishable from a comparable off-the-shelf unit
2. Specific model cell phones
3. Hand-held transmitter will be different from original transmitter
4. The robot will not be able to “talk-back” or make decisions
5. The connections between the cell phones will be manually activated
6. The system will be low power (using batteries)
7. The final product will be presentable and cosmetically pleasing.

Revised Schedule

| Week | November | | | | December | | | |
|-----------------------------------------------------------------------------------------|----------------|-----------|-----------|-----------|-----------|--|--|--|
| Task | wk12 (7) | wk13 (14) | wk14 (21) | wk15 (28) | wk17 (12) | | | |
| Integrate system and tweak sub-systems | | | | | | | | |
| Deliver Oral part of Report | JS | | | | | | | |
| Formal Evaluation #2 | | | | | | | | |
| Project Inspection | | | | | | | | |
| Oral Examination | | | | | | | | |
| Final Report | | | | | | | | |
| Senior Exit Survey | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | Mike Area | | | MA | | | | |
| | Richard Lienou | | | RL | | | | |
| | Justin Schaper | | | JS | | | | |
| | Raj Sharma | | | RS | | | | |
| | All | | | | | | | |
| Note: Those tasks marked as "All" will be broken into sub-tasks assigned to individuals | | | | | | | | |