Magnetic Field Controlled Robot

Elec 291 Project 2: Group B1

Utilizing dual microcontrollers and a time varying magnetic field we have created a small, motor equipped robot that can be wirelessly operated.

Yuankang Wang 18415142

Jacob Law 48718150

Matthew Gill 13249157

Adam Hynek 13206157

Eric Fast 45035152

Graham Whyte 45688158

Dr. Jesus Calvino-Fraga, Laboratory Section 1

Table of Contents

[Table of Figures 1](#_Toc447738110)

[Introduction 2](#_Toc447738111)

[Investigation 4](#_Toc447738112)

[Idea Generation 4](#_Toc447738113)

[Investigation Design 4](#_Toc447738114)

[Data Collection 4](#_Toc447738115)

[Data Synthesis 4](#_Toc447738116)

[Data Analysis 4](#_Toc447738117)

[Design 4](#_Toc447738118)

[Use of Process 4](#_Toc447738119)

[Need and Constraint Identification 5](#_Toc447738120)

[Problem Specification 5](#_Toc447738121)

[Solution Generation 6](#_Toc447738122)

[Solution Evaluation 7](#_Toc447738123)

[Detailed Design 7](#_Toc447738124)

[Nunchuk Controller 7](#_Toc447738125)

[Software: Transmitter 7](#_Toc447738126)

[Transmission Protocol 8](#_Toc447738127)

[Hardware: Transmitter 9](#_Toc447738128)

[Hardware: Magnetic Field Receiver 10](#_Toc447738129)

[Solution Assessment 11](#_Toc447738130)

[Overall Assessment 11](#_Toc447738131)

[Conclusion 11](#_Toc447738132)

[Design 11](#_Toc447738133)

[Functionality 11](#_Toc447738134)

[Challenges and Solutions 11](#_Toc447738135)

[Hours 11](#_Toc447738136)

[Life-Long Learning 12](#_Toc447738137)

[Life-Long Learning 12](#_Toc447738138)

[Appendices 12](#_Toc447738139)

# Table of Figures

Figure 1: Transmitter Software Block Diagram 3

Figure 2: Receiver Software Diagram 4

Figure 3: Nunchuk Conversion Table 8

Figure 4: Transmission Protocol 9

Figure 5: Man-Hour Breakdown 11

# Introduction

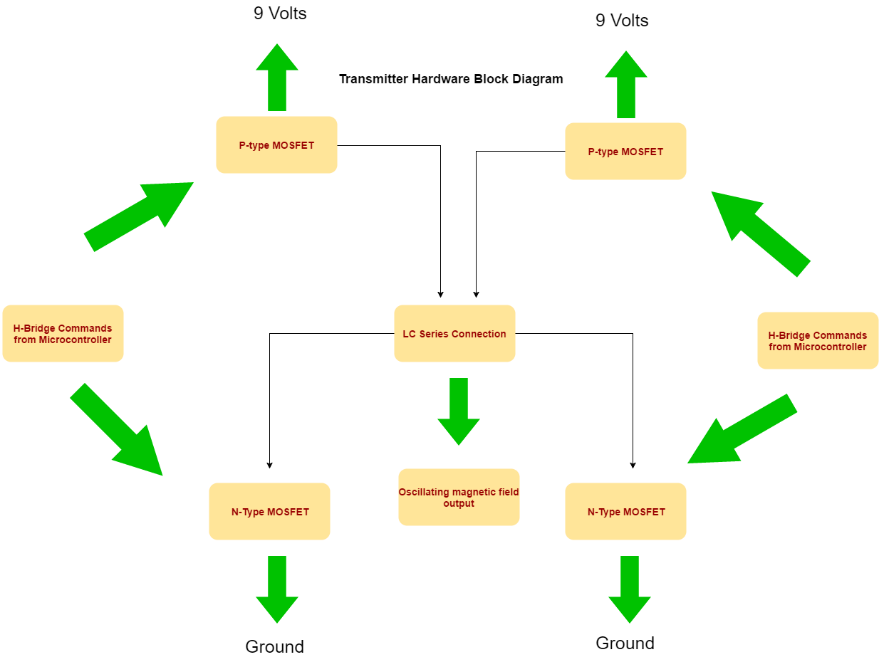
The ability to communicate wirelessly is a key component of a many technical systems. We have created a system that allows for the control a small robot by transmitting commands using a time varying magnetic field.

A 9V 15 KHz AC voltage is applied across a series connected inductor (L) and capacitor (C) causing the inductor to generate an oscillating magnetic field. The AC voltage is achieved by placing the LC series connection in an H-Bridge MOSFET configuration and using an F38X microcontroller to apply 9 Volts to alternate sides of the LC connection at a frequency of 15 KHz. This effectively drives the voltage across the inductor to 200 Volts and creates a magnetic field with sufficient strength to facilitate communication.

Information is transmitted digitally by alternately turning this field on and off. The user controls the information sent through the field by using a Wii Nunchuk controller. The output of the Nunchuk is fed into the F38X microcontroller which has been programmed to interpret commands and operate the magnetic field. The transmitter hardware and software block diagrams are shown below in figures 1 and 2.

C:\Users\Matthew\Downloads\Trans_BD.png

Figure 1: Transmitter Software Block Diagram



The magnetic field induces voltages across a receiver, dual parallel LC “tank circuits” present on the robot. The signal is then filtered with hardware and converted to a square wave so that it can be read as digital data. A second F38X microcontroller has been programmed to interpret the transmitted signals. The microcontroller in turn operates two DC motors connected to wheels through two H-Bridges MOSFET configurations to execute movement commands.

An additional mode, in which the robot attempts to remain at a constant distance and orientation on the magnetic field, has been created and the robot can alternate between this autonomous mode and the user control mode.

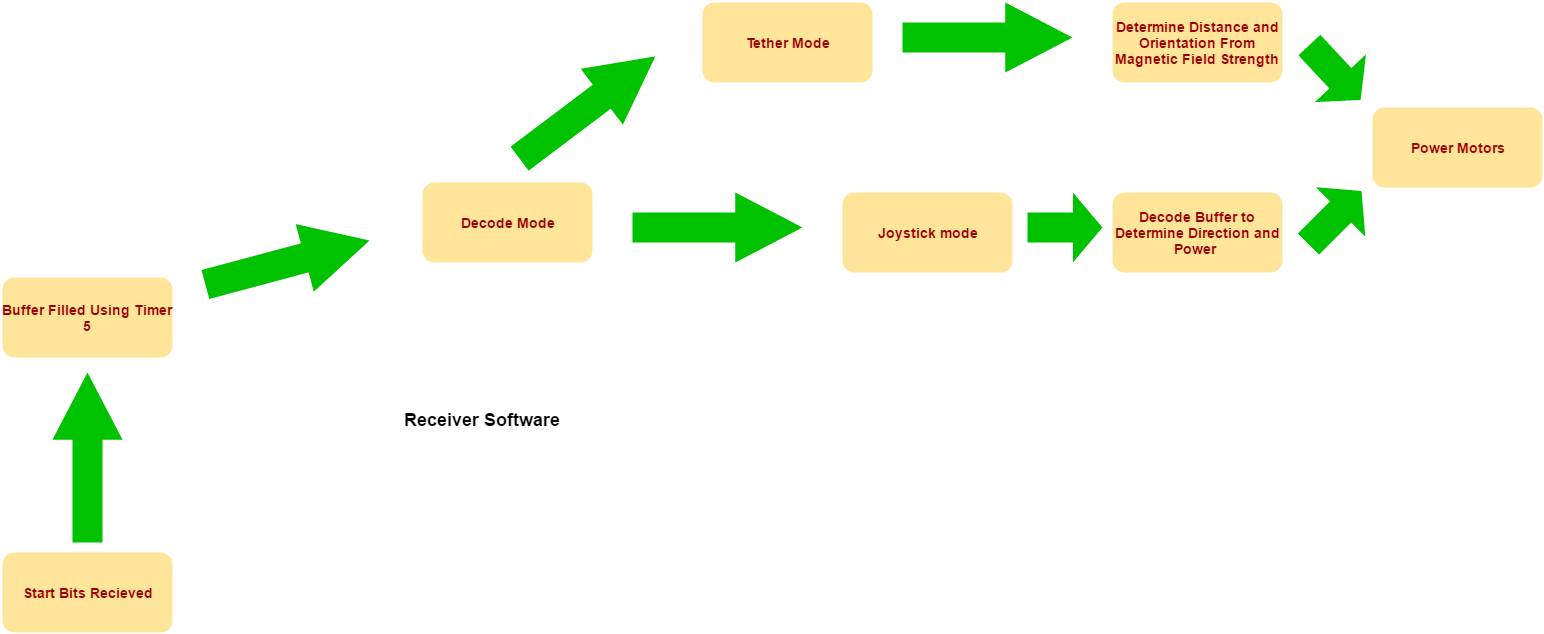


Figure 2: Receiver Software Diagram

# Investigation

## Idea Generation

## Investigation Design

## Data Collection

## Data Synthesis

## Data Analysis

# Design

## Use of Process

We split our group into two separate teams, the transmitter team and the receiver team. Each team was tasked with designing both the hardware and software necessary to accomplish the specified design requirements.

Since communication via the magnetic field was the only means of transmitting data between the two systems, the method of encoding data was a significant factor that determined how each system was designed and function. Because of this, the method of encoding data, and the rate at which it would be transmitted, was decided upon first and was the result of collaboration between the two teams. Following that each team proceeded to design their system knowing either what they had to be transmitting or what they would be receiving.

## Need and Constraint Identification

We identified the needs and constraints for this design by determining the required operation and features as specified by the project handout. The major points for each category are summarized below.

Needs:

1. Able to consistently communicate by pulsing a magnetic field
2. Able process signals and operate motors accordingly
3. Able to move forward, back, left and right
4. Able to operate independently of the user in a dedicated track mode

Constraints:

1. Must use F38X Microcontrollers programmed in C
2. Must be Battery Operated
3. Must use discrete MOSFET Drivers to control motors
4. Must transmit data at a constant frequency
5. Must be able to operate at at least 60cm

## Problem Specification

Additional design requirements were determined from the needs and constraints by establishing the interface between the different required components and their operation.

The use of a magnetic field as a communication medium created additional design requirements:

1. The ability to raise the voltage across a LC series circuit to ~200 V peak to peak to generate a magnetic field of sufficient strength
2. The ability to turn the field on and off for the specified transmission intervals
3. The ability to amplify the signal received because of the inverse cube relation between field strength and distance from source
4. The ability to convert the oscillating voltage generated by the magnetic field into a form usable by the microcontroller as digital signal
5. An transmission protocol that both the receiver and transmitter will follow

The need for a tracking mode created additional design requirements:

1. The ability to determine and monitor orientation in the magnetic field
2. The ability to determine and monitor distance from transmitter

The use of the Nunchuk controller introduced design requirements:

1. The ability to receive values from the Nunchuk
2. The ability to convert the values received into motor powers and directions

## Solution Generation

In order to process the signal we developed a custom communication protocol. The magnetic field would send pulses at the baud rate, which would then be converted to a digital signal and used for controlling the motors. The signal was filtered and amplified appropriately before being fed into a comparator and sent to the microcontroller. After determining the mode of operation, the system would either decode the rest of the signal or send a string of 1’s and relinquish control of the robot to the magnetic field.

Our communications protocol for the joystick went through several iterations. Initially, it consisted of 11 start bits, two bits to determine which buttons were pressed, two bits to determine the direction of the joystick, a padded zero, seven bits for the left motor power, another padded zero, and seven bits for the right motor for a total of 32 bits. Unfortunately, the magnetic field was not consistent enough to transmit a signal of this size. The protocol was changed to five start bits, a padded zero, two bits to determine which buttons were pressed, two bits to determine the direction of the joystick, two padded zeros, and three bits for the power of both motors for a total of 15 bits. This change improved the consistency of signals properly processed, but removed the ability to control the motors independently.

The communications protocol for the tether was fairly simple. Upon reading a 1 for the mode bit, the receiver would ignore the transmitted signal. Instead, voltages at two of the pins were compared with experimentally determined coefficients to detect the current orientation and distance between the receiver and the transmitter.

## Solution Evaluation

## Detailed Design

### Nunchuk Controller

The Nunchuk is a handheld device that consists of two pushbuttons and a joystick. It is used to control the robot in control mode. Communication with the microcontroller is achieved by using the I2C transmission protocol. The joystick outputs two values based on its displacement from its neutral, (0, 0), position. The two numbers correspond to x-y coordinates with values between -127 and 127 with forward, for example, being (0,127) and left being (-127,0). Each button is active high and received as either a 0 or a 1. It is powered by 3.3V which it receives from the microcontroller. Its relation to the other transmitter hardware is shown in the transmitter wiring diagram in Appendix \_\_. Data retrieval is accomplished in the dedicated Nunchuk and I2C functions in the Transmitter Software code in Appendix \_\_.

### Software: Transmitter

This module processes incoming data from the user via Wii Nunchuk and sends the corresponding mode and motor commands to the robot via the magnetic field.

The Nunchuk retrieval functions place the values between -127 and 127 indicating joystick displacement into the variables joy\_y and joy\_x. To account for slight variations, the interval [-5,5] is considered a zero. Since the robot can only move in the four primary directions, the logic sets the left and right motor directions (variables l\_dir and r\_dir) based on the sign of joy\_x and joy\_y. The power supplied to each motor is acquired from the magnitude of joy\_x or joy\_y and is divided by 14 per the transmission protocol described below. This is accomplished on lines 437-469 in Appendix\_\_. The table below summarizes the conversion logic.



Figure 3: Nunchuk Conversion Table

In order to send a single bit via the magnetic field the transmitter oscillates the LC H-Bridge, described in the transmitter hardware section, at a frequency of 14.95 KHz for 10 milliseconds in order to generate a field. This is accomplished using both the send\_bit (285) line and send\_int (line 306) functions, which in turn utilizes timer 2 in auto reload mode to oscillate the output pins at the resonance frequency. Whenever the program receives interrupts (line 319) from timer 2 it inverts the H-bridge output pins, effectively oscillating the H-Bridge.

### Transmission Protocol

The transmission protocol in Nunchuk mode consists of an 18-bit packet sent via the magnetic field. Each bit is sent for 10 milliseconds. The first 5 bits are the “start” bits, 5 consecutive ones. The start bits are five consecutive ones because it is an impossible combination to generate through operation in Nunchuk control mode and therefore helps prevent the receiver by mistakenly processing incorrect information. A constant zero separates the start bits from the next 4 bits, which consist of the states of buttons one and two, as well as the left and right motor directions, respectively. Both buttons are necessarily zero in Nunchuk mode (see below). Another constant zero is followed by the left motor power in 3-bit binary, which is in turn followed by another zero and the right motor power in 3-bit binary.

If button one is high, the transmitter automatically turns the magnetic field on, forcing the robot into tether mode. Similarly, if button 2 is high the transmitter turns off the magnetic field and forces the robot to receive input from the IR receiver. The transmission protocol is summarized in figure 5 below.



Figure 4: Transmission Protocol

### Hardware: Transmitter

The transmitter hardware schematic is shown in Appendix \_. It consists of three separate components, the LC H-Bridge, an F38x microcontroller, and the Nunchuk input.

The LC H-Bridge consists of an array of N and P Type MOSFETS, two BJTs, and an LC series connection. They transistors are assembled, as shown in the schematic, in an “H” configuration with the LC in the middle. With this configuration if opposite voltages, 3.3 as opposed to 0, are applied to the BJTs, nine volts drop across the LC series connection. By changing which BJT receives a high voltage, the side of the LC that is at a higher voltage changes. By oscillating the BJTs an AC square wave voltage is applied across the LC series. The voltages are applied to the BJTs through the output of the F38x microcontroller, the pins of which are connected to the bases of the BJTs through a current limiting resistor. By oscillating at the resonance frequency of 14.95 KHz, the voltage across the LC can raised as high as 220 V peak. It is this voltage across the inductor that produces the magnetic field for transmission

A small PCB, the PART NUMBERRRRR, plugs directly into the Nintendo Wii Nunchuck controller in order to get information from the joystick and buttons, to the microcontroller. It has 5 rails to connect to the Nunchuck, and $ pins that connect to the breadboard: Data Out, System Clock, 3.3V in, Ground. Data Out is attached to pin 0.0 (SDA1) and the Clock to pin 0.1 (SCL1).

### Hardware: Magnetic Field Receiver

The hardware for the receiver is composed of two separate circuits: two motor controllers, and a signal receiving/processing circuit. Each is run with its own battery pack, and can run independently of the other.

The signals are received by two parallel LC circuits, so as to be able to measure orientation relative to the controller. Both LC circuits then go through an operational amplifier to increase the voltages to a useable level. After these first stages of amplification, each then go through a precision diode to reduce the signals to only positive voltages so the microcontrollers can process them without being harmed. After the precision diodes, each go through a passive RC low pass filter to reject noise and the high frequency that the magnetic field is oscillating at. Before the passive filter, one of the lines goes to a comparator to output a square wave. Since it has not yet been filtered, the output of this first comparator is fed through an RC low pass filter to reject any unwanted magnified noise. It is then fed back into another comparator to output a clean square wave of data at the proper hi and low voltages, 5 volts and 0 volts. This circuit is powered by a 9 volt battery that is fed into a 5 volt regulator.

The motor driving circuit is made of two H-bridges, and 4 optocouplers. The LED’s in the optocoupler are driven by the microcontrollers. The motors and phototransistors are powered by 4 AA batteries in series to achieve 6 volts. For a more detailed treatment of H-bridges, see the transmitter hardware section. The H-bridges used to control the motors are similar to the one used in the transmitter but in place of BJTs, optocouplers are used to prevent “noise” from the motors distorting signals.

For component values, diagrams, IC part numbers, and pin connections please refer to appendix HARDWARE DIAGRAMS.

### Software: Receiver

This module operates on the robots onboard F38x microcontroller. It receives data via the magnetic field and operates the motors via the H-Bridges. It has three distinct operating modes, Nunchuk mode, where it receives commands from the user via the magnetic field, tether mode, where it tracks the location of the transmitter, and IR mode, where it receives commands from the user via the IR receiver.

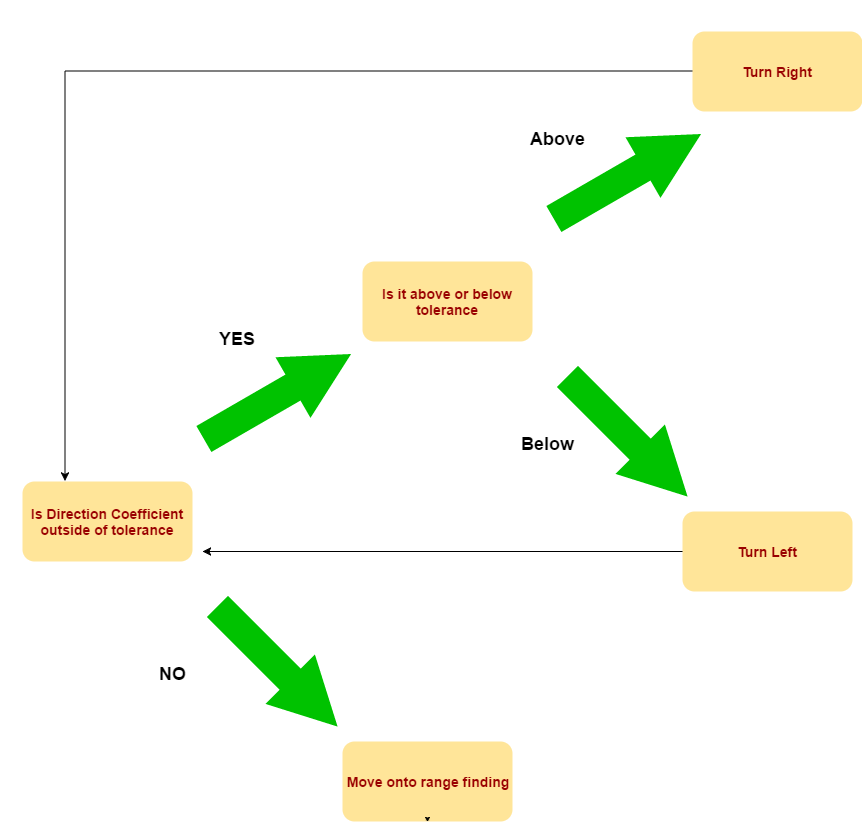
In tether mode the robot maintains a constant distance from, and orientation towards, the transmitter. This is accomplished through the use of both inductive receivers present in the hardware. By comparing the magnitude of the field (and therefore the voltage across) each inductor the robot is able to determine its orientation in the field. This is accomplished by first creating a direction coefficient (variable dir\_coef) which is simply the quotient of the left inductor voltage over the right inductor voltage. Following that the program executes the logic shown below in the figure.

Figure 5: Tether Logic

In order to obtain a stable signal, the interrupt must trigger ten times before continuing. If more than five 1’s are read, the bit is a logic 1. If not, it is a logic 0.

If the start sequence, described in transmission protocol, has been received, data collection will begin. If a zero logic is received the start counter will be reset. Once the start sequence has been received, each received bit will be placed into the buffer. Data will continue to be read until the buffer is full. The interrupt is then temporarily disabled while the signal is processed.

After the direction is read, the corresponding motors are powered according to the last three bits. The technique used to power the motors is pulse-width modulation. After the direction and power have been read, Timer 2 is used to set the corresponding pins for the appropriate amount of time. For instance, if the power is set to 7, the pin will be set high until the power changes. If the power is set to 1, the pin will be set high for 1/7 of the period of the square wave, then set to 0.

## 

## Solution Assessment

The 32-bit signal solution was determined to be unfeasible by examining the received signal. Since several of the bits would change in an untraceable way, we decided that the cause was the signal itself and not the software. As a result, we switched our design to the 15-bit signal.

### Overall Assessment

# Conclusion

## Design

## Functionality

## Challenges and Solutions

## Hours

The following chart outlines the hours worked on different aspects of this project. Figure 6 below shows cumulative man-hours (i.e. 3 people working for 12 hours is written as 36 hours).

|  |  |
| --- | --- |
| Time Breakdown of Magnetic Field Robot (Man-Hours) | |
| Meetings | 24 |
| Hardware | 250 |
| Firmware | 80 |
| Testing | 182 |
| Added Features & Functionality | 15 |
| Total | 551 |

*Figure 6: Man-Hour Breakdown*

# Life-Long Learning

As a group, we possessed all the necessary skills to complete the magnetic field controlled robot in a professional and timely manner. Having some team members that prefer hardware, and others that prefer firmware made assigning tasks and forming small teams exceptionally effective.

Overall, three classes helped us the most: ELEC201, ELEC202, CPSC259, and of course ELEC 291. ELEC201/202 gave us the understanding of dc and ac circuits that was required to design our own circuit for the magnetic field controlled robot. Overcoming hardware issues, such as unwanted noise, was made easy after learning about filters. CPSC 259 taught us how to program in pairs, and program in a modular way. And finally, having experience with microcontrollers from a semester of challenging labs made building all the firmware from scratch manageable. We had also learnt how to search large datasheets for small pieces of information when needed

# Life-Long Learning

# Appendices