Important Safety Instructions

Safety considerations are some of the most important issues when designing and building a project. It is important that nothing on the project does any harm to anyone interacting with the specified project. In this case an interactive Wheel of Fortune game cannot harm the disabled adults that will be playing it at the ATCO sheltered workshop.

All of the wires containing electrical signals throughout the game will be insulated and covered in some sort of plastic coating so that they will not fray or split at inconvenient times. The wires must also not be exposed so therefore they will be entirely enclosed in the base of the game. This will help to prevent any tripping hazards, which could be quite detrimental to the players of the game. This is especially important due to the fact that the players have many disabilities and most of them have limited motor skills in which any small tripping hazard will be amplified. By enclosing all of the wires inside a plastic box there will be no outside access. Therefore, no one can get injured by any stray wires.

The batteries used to power the game are also not hazardous to anyone’s health. Alkaline batteries are not considered hazardous waste under the US Environmental Protection Agency so therefore they are the optimum battery of choice. Lead acid batteries should not, and will not, be used throughout this project due to the fact that they could be very hazardous to everyone interacting with the game. Lead-acid batteries use a chemical reaction to do work on a charge and produce a voltage between their output terminals. This sounds like a good thing but in reality it produces many hazards. This is the main reason that lead-acid batteries will not be used in the game.

When dealing with motors there are many mechanical safety issues that become prevalent as a device is worked upon. A voltage regulator must be used as an adaptor so that the motor unit will not get too much power which will in turn cause it to burn out and possibly start a fire. The wires from the motor must also be covered for the personal safety. This will also help to prevent wires from touching, which will in turn short-circuit the motor. A PCB board will be used to decrease the change of anything shorting out. Proper occupational safety and health administration (OSHA) standards need to be followed in the connecting of the motor to the game itself. The motor must be set apart from any wires and people so that the spinning will not cause any harm. A thirty revolutions per minute (rpm) motor was used so that it will not cause the spinner to spin so fast that it would cause bodily harm to the players playing the game. However thirty rpm’s will be fast enough to make the wheel spin, which will in turn captivate the adults playing the game. The actual spinner on the game will not pose any risks to the adults playing the game. The wheel will only be spinning at thirty rpm’s for a very small amount of time (approximately two seconds) and then it will slowly come to a complete stop. This feature will minimize any injuries that could come to light such as body parts getting hit and/or stuck by the spinning of the wheel. Power will be given to the wheel for just enough time to ensure appropriate spinning speed but not enough to cause bodily harm.
Parts and Accessories

The following parts and accessories are included with the Interactive Wheel of Fortune

Wheel
Dowels (attached to the wheel)
Stickers
Plastic box and cover
Aluminum shaft
2 battery enclosures
6 battery clips
Wooden support system
Motor
Aluminum plate
PCB board
Wooden 2x4 boards
Plastic enclosure
LED’s
2 switches
Speaker
Motion sensor
Miscellaneous wires
Screws
Nails
Washers

Figure 1: Complete Game System
Features

The Interactive Wheel of Fortune has many special and unusual features. These features need to accommodate the players who suffer from diseases such as cerebral palsy and mental retardation. This game is not detrimental to the workers’ stimulation and sense of engagement.

The first special feature is the ability to have two modes of spinning the wheel. The first method includes the use of a motion sensor. The players can wave a part of their body over the sensor which will in turn give power to the motor creating the spinning of the wheel. The motion sensor has a beam angle of 38 degrees horizontal and 22 degrees vertical. The second method of spinning the wheel includes the use of a function module with a wireless remote. The remote has one button in which the adults can press. This will also give power to the motor which will in turn spin the wheel. The function module works as follows: once the button is pressed information will leave the encoder of the remote, travel through the airways and will be received by the decoder of the function module. The remote has a 100 foot transmission range. This is optimal so that if the disabled adults are unable to move around easily and freely they can still play the game from their seat. This will lead to a much more enjoyable game as more of the adults can interact and be stimulated. The function module is small enough to fit inside the base of the game. It measures 14.67 cm across, 4.5 cm high, and 8.1 cm thick. By having it inside the box any risks stemming from it will be decreased to zero.

Another special feature of the game is that it features two different ways in which the questions and answers can be read. Staff members can read the questions and answers from the provided clear protective folder. These questions are designed to be at the elementary level. By doing so this will make it of ease for the players to answer. However, careful consideration was taken to make sure that the players would not be insulted by having too easy questions. The questions can also be “read” by use of a microcassette. Questions and answers have been recorded onto a cassette. This is done in case the staff members cannot be present. The question was asked on the tape and then was followed with a ten second pause and then the answer was spoken.

Visual enhancement was highly stressed by the client. The players need to be stimulated while it is not their turn. Thus many features were added to keep their attention throughout the playing of the game. The first feature includes audio comments that are “spoken” upon activation of the wheel. These comments are made possible due to the use of the SP03 text-to-speech synthesizer. “Great Job!” is an example of such a comment that will be said throughout the game. This will help excite the players while the game is being played. Bright colors were used throughout the construction of the game. This was done to visually enhance the players while they were playing the actual game. LED’s were also placed along the base of the game. They will be blinking periodically throughout the playing of the game.
Table of Contents

1. Introduction 5-17
   1.1 General Overview of Device 5-16
   1.2 Step-by-Step Instructions 17
2. Maintenance 18-28
3. Technical Description 29-49
4. Trouble Shooting 50-70
1. Introduction

1.1 General Overview of Device

Wheel

The wheel was donated by Mike Zenker from Kernersville, North Carolina. He works for Euro-Tech. The mold well was actually made by Peter Meier Incorporated. It is actually part of a lazy Susan device. The wheel is made out of polypropylene plastic. This plastic is very durable and will withstand any sudden movements (i.e. drops, scratches, etc.). The dimensions of the wheel are as follows:

- Diameter = 60.96 cm
- Thickness = 5.423 cm

The wheel is divided into 14 even pie pieces. The manufacturing stamps were also removed by the use of sand paper. Figure 2 shows the original wheel before any painting was completed.

![Figure 2: Unpainted, Original Wheel](image)
Base of the Game

A durable and sturdy plastic box is used for the base of the game. The box came from the United States Plastic Corporation. The base being used is the StakPak CrossPak™ plastic enclosure. This was chosen due to having reinforced walls and bottom. Thus it is extremely sturdy. This was also optimal to use because it was pre-colored. The blue color was chosen because it matched the colors used in the wheel and was the brightest color offered. Figure 3 shows a picture of the boxes that we could have chosen; we chose the blue one.

Figure 3: Plastic Box

The dimensions of the base are as follows:
21.5” x 13” x 6.75”

or
54.61cm x 33.02cm x 17.145cm

By having the base this large all of the components are able to fit inside the base. The function module and PCB board are just some of the components that will be inside the box. This is imperative so that if one of the players spills a drink or anything else the electrical components will be safe. The wheel will not directly touch the base of the box. By having this happen there will be no added friction during the spinning of the wheel.
**Clicker**

Visual and auditory enhancement was a primary factor during the construction of the game. Thus a clicker device was devised. A wooden block is attached to the corner of the lid of the box. It is attached at such an angle so that when the wheel spins no contact between the two will happen. Elastic tubing was then used and attached to the wooden block. The dowels that are attached to the top of the wheel will come into contact with the elastic tubing when the wheel is spun. This will provide a clicking noise. It will also determine which pie piece each player lands upon. Figure 4 shows the clicker set-up.

![Clicker Set-up](image)

**Figure 4: Clicker Set-up**

**Support System**

There is of course a support system built into the game. A hollow wooden block was built so that the motor could be freely suspended. This was important because wires needed to be connected to the bottom of the motor. Without having wires connected no power would be fed to the motor itself. Figure 5 shows the support system without anything added onto it.
Figure 5: Wooden Support System

The dimensions of this wooden block are as follows (all in inches):

- **Outside**: 5.061” x 5.01” x 3.38"
- **Inside**: 2.115” x 2.105” x 3.38”
- The width of each piece of wood is approximately 2”

Once the support system was built the next step was to attach it to the bottom of the box. Screws and washers were used to prevent the plastic from stretching over time. Without using washers the holes that the screws go into could get bigger as time went on. Two washers and screws were used. Figure 6 shows this. Figure 7 depicts the inside of the box with the wooden block firmly attached.
Figure 6: Bottom of the Box with Screws & Washers

Figure 7: Inside View
Wooden planks were then attached to each side of the support system as well as to the sides of the box. Figure 8 shows the completed support system.

![Figure 8: Complete Support System](image)

These pieces of wood were securely screwed to the bottom of the box in the same manner as previously stated. The pieces of wood are on the left 7.6 inches (19.304 cm) and on the right 9.4 inches (23.876 cm), when looking at the picture above. By adding this extra support the wooden block was now firmly held in place.

**Motor In-Casement**

The motor needed to be held in place in a secure area. The wooden support system provided the necessary room underneath the motor. However, it did not hold up the motor entirely. Thus an aluminum sheet was cut and put into place using screws. Lock-screws were then used to ensure that the motor would not slip out during the playing of the game. In figure 9 you can see a picture of this.
The motion sensor is placed on the front of the box. In order for it to be properly activated something must be waved in front of it just one time. Figure 10 shows what the motion sensor looks like. Only the tip of the sensor pokes out of the box.

Figure 11 and Figure 12 both show different angles of the sensor angle.
The motion sensor has a beam angle of 38 degrees horizontal and 22 degrees vertical. This is important to remember because the positioning of the sensor on the box relates directly to how must the sensor will sense.

Wireless Remote

Due to the fact that the adults playing the game have some disabilities that limit their motor function there will be two different types of spin activators. The first type of activators will be a push button on a remote control that will wirelessly activate the wheel to spin. The function module, which is the transmitter will be placed into the base of the game. This will ensure that the factor of safety is at the highest and optimal level. Figure 13 shows what the function module looks like. Figure 14 shows what the remote looks like. The remote that is being used is the one-button one shown to the far left in the picture.
The function module must sit in the acceptable position that can be seen in Figure 15. By having it sit like this the remote will have a 100 foot transmission range.
SP03 Text-to-Speech Synthesizer

The SP03 test-to-speech synthesizer was used to provide audio comments. It was linked into the circuit of the game. As the wheel spins it will generate a random audio comment to be played. Figure 16 shows what the SP03 looks like. Figure 17 shows how the audio commands were programmed.

![SP03 text-to-speech Synthesizer](image)

**Figure 14: SP03 text-to-speech Synthesizer**

![SP03 Audio Loaded Commands](image)

**Figure 15: SP03 Audio Loaded Commands**

**Voice Recorder**

The voice recorder is part of the game so that the disabled adults can still play the game when the staff members are not present. Set questions and answers are recorded onto a
tape. This will help the adults at the workshop to feel more independent. It is not part of
the game circuit. It runs off of 2 double A batteries. It will also not be placed inside the
base of the game. It will be an addition to the game which will go along with the
questions and answers booklet. Figure 16 shows a picture of the voice recorder that is
provided with the game.

![Microcassette Voice Recorder](image)

**Figure 16: Microcassette Voice Recorder**

**Questions and Answers Book**

Twenty questions and answers are provided with the game. They are a separate piece of
the game. That means that they are not part of the general circuit of the game. They also
do not fit inside the box. They are at an elementary level so that the adults can easily
answer them. The following picture (Fig. 17 shows the booklet).
Figure 17: Questions & Answers Book
1.2 Step-By-Step Instructions

Power Game

Turn Game On
“Welcome to the Interactive Wheel of Fortune” audio comment played

Choose a method of spinning

Choose a Method of Spinning

Wireless Remote    Motion Sensor

Wheel Spins
Audio Comment Played During the Spinning of the Wheel
Wheel Stops
Question is Asked
Answer
If Correct, Points are Awarded to the Player
Spin Again
2. Maintenance

Battery Life

In our project we used multiple 9 Volt batteries. The following figure shows an example of the batteries used.

![Image of 9 Volt Batteries]

**Figure 17: 9 Volt Batteries**

The motor used in our project requires two 9 volt batteries. When the wheel is spinning it will use 40 mA. When the motor starts to spin the wheel it will then use 200 mA. Each battery can provide 500 mA.
The SP03, motion sensor, and microcontroller each require a 9 volt battery with a positive 5 volt regulator. According to the following graph provided by Duracell, the battery will be able to provide 50 mA continuously for 10 hours.

Graph 1
Battery Replacement

As with all non-rechargeable batteries they will have to be replaced eventually. In order to accommodate this we have made it relatively easy for the staff members to change them themselves.

Step 1: Turn off all power to the game

Step 2: Unscrew back of battery holder

![Figure 18: Unscrew](image-url)
Step 3: Unclip Batteries from battery clips

Step 4: Exchange batteries with new ones

Step 5: Reattach cover to the base of the game
Environmental impacts have to do with the packaging and electronics of the game. All of the products used in the development of the game must be environmentally safe. The battery used to supply power must be certified and suitable for the use of the game. An alkaline battery will be used because it is much safer than a lead acid battery. Some components may age and need to be replaced. Thus it is important that all components be able to be thrown away and not harm the environment. It is very important to keep the environment in mind when designing any new product.

![Proper Disposal](image_url)

**Figure 20: Proper Disposal**

Proper disposal of the batteries after use must also be taken into consideration.
Water and other liquids that could potentially spill should be avoided at all costs

![No Water](image1.png)

**Figure 21: No Water**

When dealing with anything that has an electronic component it is important to be extra careful. Therefore, one must pay attention to the following sign.

![Hazardous](image2.png)

**Figure 22: Hazardous**
Mechanical

Wheel must stay press-fitted

Inspection of the game before playing is an important thing to do
  As playing is done over time screws may become loose
  Components may not fit as well as they once did
  Galling of the plastic may occur
  Components attached to the base may loosen with time

Always check the aluminum shaft before playing the game
  - it may loosen as time progresses
  - lock screw can be tightened using included allen wrench

Figure 23: De-attaching lock screw
Wooden block may begin to wobble during spinning

Solution = Wooden 2x4’s may have to be replaced
    This can be tested just by spinning the wheel
    Shaky or even

Motor could become loose
    Potential to fall out of the wooden and aluminum support system

Solution = tighten black screws found in the middle of the aluminum block

Figure 24: Motor Attachment
Electrical

Wires could short circuit
- metal parts touching together cause a short in the circuit
- could lead to fire damage

![Short Circuit](image)

**Figure 25: Short Circuit**

Fire
- too much voltage/current running though a particular wire

![Electrical Fire](image)

**Figure 26: Electrical Fire**
Adherence of PCB board to electrical safe box could deteriorate over time

Solution = new tape
Recommendation = use scotch foam tape

Figure 27: Recommended Mounting Tape
LEDs
- Will burn out over time
- Will need replacing

Replacement:
1.) Bulb is what needs to be replaced from actual Christmas lights
2.) Bulb can easily be pushed out
3.) Attach new bulb
4.) Test by turning power on for game

Figure 28: Recommended Replacement Bulbs
3. Technical Description

Technical Description

The most important aspect of the Interactive Wheel of Fortune game is the main circuit design. It consists of a motion sensor, 4-relay function module, PIC 16F874 (microcontroller), 4 positive five volts regulators, 1 positive 12 volts regulator, IRF540 N-type Power MOSFET, 6 Battery connections and a string of 24 LEDs. The entire PCB board design is shown in Figure 29. This layout will be further broken down to show where each of the components are attached.

Figure 29: The schematic of the overall design for the microcontroller and the motor.
Figure 30: The PCB layout for the main circuit
Motion Sensor

A depiction of the motion sensor is shown in Figure 31. The motion sensor was placed on the right side of the front side of the game. It has three connections; Voltage input, ground and Voltage Output. A wire was soldered onto each of the pins and Figure 34 shows how determine which pin is which by looking at the pins in a L shape configuration. The top pin when looked at L configuration is the ground pin, the corner pin is the output pin and the right pin is the input pin. If you are still not able to determine the pin configuration, then look at the other end of the wire where it is attached to the PCB board. The ground wire is attached to point 11, the voltage output wire is attached to point 12, and the voltage input wire is attached to point 13. This can be seen in the PCB layout shown in Figure 34. The voltage input point is connected to the voltage output of a 5 volts regulator (Unit). In order to troubleshoot the motion sensor, the primary route is to check if 7 to 9 volts of power supply is being input into the voltage regulator (VR3) and check to see if 3 to 5 volts are being input into pin 13.

Figure 31: A picture of the motion sensors.
Figure 32: Shows how the 3 pins of the motion sensors are connected into the internal diagram of the sensor itself
Figure 33: Shows what the range of detection for the motion sensor is. Also the bottom figure shows what each pin is configured if you look at the motion sensor in a certain way.

The next step in determining if the motion sensor works or not is to check for voltage at the voltage output pin. In order, to do this place black pin of the multi-meter and place it on point 11 and the red pin onto the point 12. This should show a voltage of 5 volts. If the voltage is not seen, and the power is being input to the motion sensor, then there can be only two things that might be causing the problem. Either the wires attached to the motion sensor are soldered together or that the motion sensor has been burnt out. If soldering is required, then switch off the game solder the wires. If soldering is not required then shut off the game for 2 minutes and let the motion sensor restart and if it still does not work, then a new motion sensor will be required. This motion sensor was bought from digikey, part number 1810.
Figure 34: The part of the PCB where the motion sensor and its power supply are connected.
Function Module

The function module is made up of two components, the keyfob remote and the receiver unit. The keyfob remote consists of one button. Every time the button is pressed, the receiver unit on the function module receives the signal and output this using on the pins. This is described in details further below. The main form of sending and receiving is based on setting the address. Figure 35 shows all of the main component of the function module.

1. Antenna
2. DC Power Connector (5 to 16VDC)
3. “Quick-Connect” Alternate DC Input
4. Switched Device Connectors
5. Address Selector DIP Switches

Figure 35: Main components of the Function module.
The address of the function module can be easily set by either turning the pins on and off (the pins located on the Address selector DIP Switches). In order to determine which pins to turn on and which ones to turn off, basically look at the Figure 36.

<table>
<thead>
<tr>
<th>Address Bit</th>
<th>Transmitters</th>
<th>Receivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A1</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A2</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A3</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A4</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A5</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A6</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A7</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A8</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
<tr>
<td>A9</td>
<td>Float, Gnd</td>
<td>Float, Gnd, Vcc</td>
</tr>
</tbody>
</table>

**Figure 36: This shows how to set the address of the receiver and transmitter units for the wireless control of the game.**

The transmitter unit used in this game was the one button keyfob remote and the address was set to all grounds. However, using pins 9 and 10 on the receiving unit (FCTN-RLY4) are always floating, it is very important that the corresponding pins on the transmitting keyfob remote are also floating. In order to do this, we took a razor blade and cut the traces on pins 8 and 9. Figure 37 shows which pins were cut and figure 38 shows the Keyfob remote after the cuts.
Figure 37: Keyfob remote pins that needed to be cut

Figure 38: The keyfob remote pins after the cut.
Figure 39 also shows where the lithium battery for the keyfob remote is placed. This will not need to be changed for 5 to 7 years. In order to supply the voltage supply for the receiving unit, a direct 9 volt battery was used. The function module has the capability of running, if the power supplied to it is between 5 to 17 volts. This is very important, because it required one less electrical in the game. Figure 39 shows where the ground and the power supply are supplied (Figure X: ‘quick connet” alternate DC input).

![Diagram of RELAYS and GND with 5-16VDC](image)

**Figure 39:** Shows where the power supply and the ground are connected to the function module. Please look at Figure X for the over view of the function module.

In order to get the voltage output from the receiving unit, every the button was pressed; a wire was directly soldered onto the surface of the PCB board in the function module. Figure 40 shows the schematic of the PCB board and the zoomed in schematic of the pin attached to the wire and Figure 41 shows the PCB with the wire attached to it. In order to make this connection strong, a wire wrap was used to decrease the pull on the wire.
Figure 40: The schematic of the function module with a zoomed in schematic where a wire was attached to get the output voltage.
Figure 41: This shows the wire connected to the PCB board of the function module (FM)

This wire runs from the function module to the main circuit PCB Board through a hole that was drilled on top of the function module box. This wire is connected to the FM pin on the PCB board. Further details on how these pins are connected to the microcontroller are described in the microcontroller section of the report.
In order to get audio aspect into the game, a SP03 module was used. SP03 is already, ready to use manufactured unit that converts text into speech. 16 phrases were installed onto the chip using the RS232 connection. Figure 42 shows how the SP03 connection was made to the RS232 wire.

**Figure 42: Connection between the Sp03 and the RS232 wire.**
Figure 43: Shows the Pin configuration of the SP03

The SP03 is connected to 8 wires that run from the pins (gnd, sel 0, sel1, sel2, sel3, sel4, Status, +5 volts) to the 8 pin connection on the PCB board. In order to provide power and ground a 9 volt battery was used with a 5 volt regulator (VR2). The G1 pin was connected to the ground output of the voltage regulator and the S51 pin was connected to supply the 5 volt output of the voltage regulator to the SP03. The front face of the SP03 faces the voltage regulator. The SP03 was not directly connected to the PCB board due to the smaller pins of the PCB board compared to that the squared larger pins of the SP03.
Figure 44: Shows the pins where the 8 wires from the Sp03 are connected to. The eight pin connection is the one on the electrical unit called SP03.
The other main circuit used in this Interactive Wheel of Fortune game was to the Audio amplifier. The audio amplifier provides a 1 watt increase in gain from the original output of the SP03. Figure 45 shows the schematic of the circuit. The main component of the circuit is the LM 386 low voltage amplifier. The general pin layout of the component is shown in Figure 46. The PCB board was provided by the electrical engineering department so a design was never made.

**Figure 45:** The amplifier circuit used for the audio amplifier

**Figure 46:** The pin layout of the LM386 audio amplifier.
**Microcontroller**

The most important component for the electrical aspect of the game is the microcontroller. Figure 47 shows the pin diagram of the microcontroller. In order for the microcontroller to work, 4 major connections need to be made. Primarily, a 2-5 volts need to be supplied to pin 1 (MCLR/VPP). Pins 12 and 31 need to be grounded and pins 11 and 32 need to powered by 5 volts. Finally, a 4 MHz oscillator needs to be connected between 13 and 14, with 22 Pico farad capacitors grounded. Figure 48 shows how this part of the microcontroller is connected to its corresponding connection.

**Figure 47: The overview of the pins for the microcontroller.**
Figure 48: The figure shows the PCB layout of this part of the circuit.

A resistor was placed in the RM pins to make sure that no more than 5 volts is being supplied into the microcontroller. A 5 volt voltage regulator (VR1) was used with a 9 volt battery to power the microcontroller.

The other connection made to the microcontroller include the output of the motion sensor, function module and the output to the gate pin of the MOSFET to switch the motor on and off at a certain frequency.
MOSFET and motor

The MOSFET used in order to switch the motor on and off at very high frequency to control the speed at which the motor stops was IRF540. IRF540 is an N-type power MOSFET. It has three connections, gate, drain, and source. The gate pin is directly traced to the capture 1 pin on the microcontroller. The source pin is traced to the ground pin powering the MOSFET and the motor. The drain pin is traced to the negative connection of the motor and the positive of the motor is traced to the voltage output pin of voltage regulator (VR12). This is a 12 volt voltage regulator, which has a two 9 volts battery in series connected to it. A diode is placed parallel to the motor connection in order to prevent back powering, which usually occurs every time motor is powered.
Figure 50: The configuration of each pin of the IR540 MOSFET

### ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DS}$</td>
<td>Drain-source Voltage ($V_{GS} = 0$)</td>
<td>100</td>
<td>V</td>
</tr>
<tr>
<td>$V_{DGR}$</td>
<td>Drain-gate Voltage ($R_{GS} = 20 , k\Omega$)</td>
<td>100</td>
<td>V</td>
</tr>
<tr>
<td>$V_{GS}$</td>
<td>Gate-source Voltage</td>
<td>± 20</td>
<td>V</td>
</tr>
<tr>
<td>$I_D$</td>
<td>Drain Current (continuous) at $T_C = 25^\circ C$</td>
<td>22</td>
<td>A</td>
</tr>
<tr>
<td>$I_{DP}$</td>
<td>Drain Current (continuous) at $T_C = 100^\circ C$</td>
<td>15</td>
<td>A</td>
</tr>
<tr>
<td>$I_{DM}^*$</td>
<td>Drain Current (pulsed)</td>
<td>88</td>
<td>A</td>
</tr>
<tr>
<td>$P_{tot}$</td>
<td>Total Dissipation at $T_C = 25^\circ C$</td>
<td>85</td>
<td>W</td>
</tr>
<tr>
<td>Derating Factor</td>
<td></td>
<td>0.57</td>
<td>W/°C</td>
</tr>
<tr>
<td>$dV/dt$</td>
<td>Peak Diode Recovery voltage slope</td>
<td>9</td>
<td>V/ns</td>
</tr>
<tr>
<td>$E_{AS}$</td>
<td>Single Pulse Avalanche Energy</td>
<td>220</td>
<td>mJ</td>
</tr>
<tr>
<td>$T_{stg}$</td>
<td>Storage Temperature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_J$</td>
<td>Max. Operating Junction Temperature</td>
<td>-55 to 175</td>
<td>°C</td>
</tr>
</tbody>
</table>

Figure 51: The absolute maximum rating of the MOSFET.
Figure 52: Shows how the MOSFET is connected to the motor and to the 12 volt voltage regulator powering the motor.

Switches

Two switches were used to turn 6 batteries on and off attached to the sp03, microcontroller, motion sensors, function module, motor.
4. Troubleshooting

Trouble shooting the motion sensor

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>POSSIBLE CAUSE</th>
<th>HOW To FIX IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>After turning the Switch on, there is no voltage going through</td>
<td>Dead batteries</td>
</tr>
</tbody>
</table>

| 2. | Check Voltage regulator input and output, and not working | Voltage regulator is damaged | Replace the Voltage regulator |
3. Check the motion sensor output, whether or not the output is occurring | Motion Sensor damaged | Turn switch off, turn it back on, still not working, replace it by new motion sensor.

Figure 53: The output of the motion sensor can be checked between pins 11 and 12.
### Troubleshooting the Function Module

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><strong>Microcontroller not inputting the output of the motion sensor</strong></td>
<td><strong>Microcontroller pin damaged</strong></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><strong>Function module not working</strong></td>
<td><strong>Dead batteries</strong></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><strong>Function module not working</strong></td>
<td><strong>Wire connection loose</strong></td>
</tr>
<tr>
<td></td>
<td>Function Module not working</td>
<td>Wire is not soldered correctly</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------</td>
<td>--------------------------------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Function Module not working</th>
<th>Antenna arrangement</th>
<th>Rearrange the antenna to the position advised in figure X.</th>
</tr>
</thead>
</table>

![Diagram](Poor) ![Diagram](Optimum) ![Diagram](Acceptable)
9. Function Module not working | Addressing might have been accidentally changed | Refer to the technical section of the report to make sure that all of first 7 pin of the function module are grounded.

10. Function Module not working | FOB remote battery died | Replace the battery

Trouble shooting the Motor

11. Motor not spinning, but LEDS are working | Power supply to the motor | Check the power supply to the VR12 voltage regulator. How much is being inputted and how much is being outputted.

12. Motor not spinning, but the LEDS are working | Wire connection at pins 9 and 10 | Solder the wires again
| Motor not spinning, but the LEDs are working | Problems with the MOSFET | Check to see if the 5 volts are being inputted into the MOSFET, every time the game is initiated. If not then there is problem with either the MOSFET or the microcontroller pin. Described next |
| 14. | Motor not spinning, but the LEDs are working | MOSFET damaged | Unsoldered the MOSFET and replace it with new MOSFET (IRF540). |
| 15. | Motor not spinning, but the LEDs are working | Microcontroller pin damaged | Reload the program provided onto a new microcontroller and replace the old microcontroller |
| 16. | Motor not spinning, but the LEDs are working | MOSFET receiving input from the function module, but not outputting anything (5 volts) | Unsoldered the MOSFET and replace it with new MOSFET (IRF540) |
| 17. | Motor not spinning, and LEDs are not working | Microcontroller pins damaged | Reload the program provided onto a new microcontroller and replace the old microcontroller |
| 18. | Motor not spinning, but the LEDs are working | Microcontroller not receiving power | Change batteries to the microcontroller. If still not working also change batteries for the LEDs and the motor |
| 19. | Motor not spinning, but the LEDs are working | Microcontroller pin damaged | Reload the program provided onto a new microcontroller and replace the old microcontroller |

Troubleshooting the SP03

| 20. | Audio comments not heard | No power supplied | Change batteries |
| 21. | Audio comments not heard | The wire connection might be loose | Unsolder and soldered again all the wires associated with the SP03 |

| 22. | Audio comments not heard | Microcontroller pin damaged | Reload the program provided onto a new microcontroller and replace the old microcontroller |

<p>| 23. | Audio comments not heard | Microcontroller pin damaged | Reload the program provided onto a new microcontroller and replace the old microcontroller |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>24.</td>
<td>Audio comments not heard</td>
<td>The volume is set to low</td>
</tr>
<tr>
<td>25.</td>
<td>Audio comments not heard</td>
<td>Speaker damaged</td>
</tr>
<tr>
<td>26.</td>
<td>Audio comments not heard</td>
<td>Audio amplifier circuit damaged</td>
</tr>
<tr>
<td>27.</td>
<td>LEDs</td>
<td>Battery supply low</td>
</tr>
</tbody>
</table>
### Mechanical troubleshooting

#### 28.

<table>
<thead>
<tr>
<th>LEDs</th>
<th>Microcontroller pin damaged</th>
<th>Reload the program provided onto a new microcontroller and replace the old microcontroller</th>
</tr>
</thead>
</table>

#### 29.

<table>
<thead>
<tr>
<th>Wheel is wobbling</th>
<th>Press fit is loose</th>
<th>Press fit it again</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base is broken</td>
<td>The game fell down</td>
<td>Ask for remanufacturing</td>
</tr>
</tbody>
</table>
USE THIS PROGRAM, IF A NEW MICROCHIP IS REQUIRED

#include P16F874.inc
__config(_CP_OFF & _PWRTE_ON & _XT_OSC & _WDT_OFF &
_BODEN_OFF & _LVP_OFF)
list P=PIC16F874, F=INHX8M, C=160, N=77, ST=OFF, MM=OFF, R=DEC, X=OFF

W EQU H'0000'
F EQU H'0001'
Bank0RAM equ H'20' ;Start of Bank 0 RAM area
MaxCount equ 50 ;Number of loops in half a second

;***** VARIABLE DEFINITIONS
cblock Bank0RAM
w_temp ; variable used for context saving
S_temp ; variable used for context saving
CIAMSB ; Timer1 CIA1 MSB register
CIALSB ; Timer1 CIA1 LSB register
DELAY_COUNT ; Variable for Delay
DELAY_COUNT2 ; Variable for Delay
Count1
Count2
Count3
Restore
FRO
Randomvar
endc

MOVLF     macro literal,dest
movlw literal
movwf dest
endm

MOVFF     macro source,dest
movf  source,W
movwf dest
endm

;******************************************************************************
;    VECTORS
;******************************************************************************
ORG    H'000' ; processor reset vector
goto   START ; go to beginning of program
ORG H'0004' ; Origin for the interrupt vector of any enabled peripheral(check)
GOTO   INT ; PER_INT = PERIPHERAL
INTERRUPT

61
START;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
---------Start loop
CALL INITIAL
MAINLOOP
CALL MAIN
GOTO MAINLOOP

INITIAL;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
--------Initialize
CLRF INTCON
BSF STATUS,5 ;Go to bank1.
MOVLF B'00000001',PIE2 ;Enable Capture1 and Timer1
CLRF TRISD
MOVLF b'00000100', TRISE
MOVLF b'11010011',TRISC
CLRF TRISA
MOVLF B'111100001',TRISB ;Set I/O for PORTB
MOVLF 249,PR2
MOVLF B'11000001',OPTION_REG ;Set interrupt on rising edge.
MOVLF B'00000110', ADCON1
BCF STATUS,5 ;Go back to bank 0.
CLRF TMR0
MOVLF B'000000101',CCP2CON
movlw b'00000000' ;move returned bits into W
movwf CCPR1L ;set duty cycle for CCP2
MOVLF b'00000100', CCP1CON ;Set the capture register for every fourth rising edge. MOVLF b'00000110', T1CON ;Timer1 initialization
MOVLF b'01001101', T2CON
CLRF TMR1H ;Initialize Timer1H nibble to 0(required due to software overhead).
CLRF PORTD ;Set TRISD as output.
clf PORTC
CLRF CCPR2L
CLRF CCPR2H;
omlw 0x00 ;Turn the LED oFF by first putting
;movwf PORTD
MOVLF B'11000000',INTCON ;Turn on global interrupt, RBO/INT external interrupt and flag.
return
MAIN ..................................................................................................................................................
-----Main program
  bcf STATUS,0
  MOVLF B'111000',PORTA
  call DELAY
  call DELAY
  call DELAY
  call DELAY
  COMF PORTA
  call DELAY
  call DELAY
  call DELAY
  Return

DELAY
  movlw 0xFF
  movwf DELAY_COUNT
  movlw 0xFF
  movwf DELAY_COUNT2
LOOP decfsz DELAY_COUNT,1
goto LOOP
decfsz DELAY_COUNT2,1
goto LOOP
return

INT ; THIS part of the interrupt routine is to make sure that an interrupt is occurring

  MOVLF B'1111111', PORTA
  call DELAY
  call DELAY
  call DELAY
  call DELAY
  call DELAY
  call DELAY
  call DELAY
  call DELAY
  MOVFF CCPR2L,CIALSB
  MOVFF CCPR2H,CIAMSBB
; Set aside W and STATUS
  movwf w_temp ;Copy W to RAM
  swapf STATUS,W ;Move STATUS to W without affecting Z bit
  movwf S_temp ;Copy to RAM (with nibbles swapped)

; Execute polling routine
Poll
  btfsc INTCON,INTF ;Test PB0/INT flag
goto RPG ;Echo RPG outputs to lower two LEDs
btfs c PIR2,CCP2IF ;test for Capture interept
goto Random ;if so goto Intensity

; Restore STATUS and W and return from interrupt
    bcf STATUS,0
    swapf S_temp,W ;Restore STATUS bits (unsapping nibbles)
    movwf STATUS ; without affecting Z bit
    swapf w_temp,F ;Swap W_TEMP
    swapf w_temp,W ; and swap again into W without affecting Z bit
    retfie ;Return from mainline code; reenable interrupts

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; This subroutine uses the Timer 1 value at interrupt to generate
; sixteen different random numbers depending on the time it falls within
; sixteen different intervals.

Random
    bcf PIR2,CCP2IF ;clears the capture interrupt

    call DELAY ; moves the timer1 value to W
    movf CIAMSB,0 ; Moves the timer1 value from
    sublw b'00010000' ; Literal (16)
    btfs c STATUS,0 ; If number is larger than 16 skip the
    goto Random1 ; If number is smaller than 16
    output 1

    movf CIAMSB,0 ; moves the timer 1 value to W
    sublw b'00100000'
    btfs c STATUS,0
    goto Random2

    movf CIAMSB,0
    sublw b'00110000'
    btfs c STATUS,0
    goto Random3

    movf CIAMSB,0
    sublw b'01000000'
    btfs c STATUS,0
    goto Random4

    movf CIAMSB,0
    sublw b'01010000'
    btfs c STATUS,0
    goto Random5
btfsc STATUS,0
goto Random6
movf CIAMSB,0
sublw b'01110000'
btfsc STATUS,0
goto Random7
movf CIAMSB,0
sublw b'10000000'
btfsc STATUS,0
goto Random8
movf CIAMSB,0
sublw b'10010000'
btfsc STATUS,0
goto Random9
movf CIAMSB,0
sublw b'10100000'
btfsc STATUS,0
goto Random10
movf CIAMSB,0
sublw b'10110000'
btfsc STATUS,0
call Random13
movf CIAMSB,0
sublw b'11000000'
btfsc STATUS,0
call Random14
movf CIAMSB,0
sublw b'11100000'
btfsc STATUS,0
movf CIAMSB,0
sublw b'11110000'
btfsc STATUS,0
goto Random15
movf CIAMSB,0
sublw b'11111111'
btfsc STATUS,0
goto Random16
Retfie

Random1
MOVLF b'00000001', Randomvar
MOVLF b'00011111', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random2
MOVLF b'00000010', Randomvar
MOVLF b'00100001', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random3
MOVLF b'00000011', Randomvar
MOVLF b'00100011', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random4
MOVLF b'00000100', Randomvar
MOVLF b'00100101', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random5
MOVLF b'00000101', Randomvar
MOVLF b'00100111', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random6
MOVLF b'00000110', Randomvar
MOVLF b'00101001', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random7
MOVLF b'00000111', Randomvar
MOVLF b'00101010', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

Random8
MOVLF b'00001000', Randomvar
MOVLF b'00101011', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random9
MOVLF b'00001001', Randomvar
MOVLF b'00101111', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random10
MOVLF b'00001010', Randomvar
MOVLF b'00101111', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random11
MOVLF b'00001011', Randomvar
MOVLF b'00110001', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random12
MOVLF b'00001011', Randomvar
MOVLF b'00110011', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random13
MOVLF b'00001100', Randomvar
MOVLF b'00110101', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random14
MOVLF b'000001001', Randomvar
MOVLF b'00110101', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random15
MOVLF b'000001011', Randomvar
MOVLF b'00111001', Count2
MOVFF Count2, Restore
call SPIN
goto Poll
Random16
MOVLF b'000001000', Randomvar
MOVLF b'00111101', Count2
MOVFF Count2, Restore
call SPIN
goto Poll

;;;;;;;;;;RPG;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
RPG
bcf INTCON,INTF ; Clear the RBO/INTFlag
goto Poll

;;;;;;;;;;SPIN;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SPIN
bsf T2CON,2 ; turns timer 2 on
movlw b'11111111' ; move returned bits into W
movwf CCPR1L ; set duty cycle for CCP2
MOVLF b'00001100',CCP1CON
MOVLF b'101010', PORTA
call Delay
call SP03
call Delay
movlw b'11000111' ; move returned bits into W
movwf CCPR1L ; set duty cycle for CCP2
MOVLF b'00001100',CCP1CON
call DELAY
call DELAY
call DELAY
movlw b'10101011' ; move returned bits into W
movwf CCPR1L ; set duty cycle for CCP2
MOVLF b'00001100',CCP1CON
call DELAY
call DELAY
call DELAY
movlw b'01100011'
movwf CCPR1L ; set duty cycle for CCP2
MOVLF b'00001100',CCP1CON
call DELAY
call DELAY
call DELAY
movlw b'00110010'
movwf CCPR1L ; set duty cycle for CCP2
MOVLF b'00001100',CCP1CON
call DELAY
call DELAY
call DELAY
movlw b'00000000'
movwf CCPR1L ; set duty cycle for CCP2
MOVLF b'00001100',CCP1CON
MOVLF b'010101', PORTA
call Delay
return

;;;;;;;;;SP03;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SP03
clrf PORTD
INCF Randomvar,0
movwf PORTD
CALL DELAY
CLRF PORTD
Return

;;;;;;;;Delay;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Delay0to1;;;;;;;;;;
;;;;;;;;;;This subroutine runs the motor at 100% duty Cycle between 2 to
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4 seconds.
Delay
MOVLF b'11111111', Count3 ;stores 255 into the count3
decfsz Count1,1 ;decrements count1 by 1
goto Delay1
Return
Delay1
decfsz Count3,1
goto Delay2
goto Delay
Delay2
decfsz Count2,1
goto Delay2
MOVFF Restore, Count2
goto Delay1
;;;;;;;;;;;;;;;;;;SP03;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
end
PLEASE DO NOT DO ANY OF THE ELECTRICAL TROUBLESHOOTING. ONLY TECHNICIANS OR ELECTRICIAN ARE PREMITTED TO DO THE TROUBLESHOOTING. DO NOT TRY TO SOLDER OR UNSOLDER ANY OF THE COMPONENTS.