G1 Duck Hunt Game

Group Number: 1

Jesse Larson (jrlarson@ualberta.ca)

Qingqing Liu (qliu6@ualberta.ca)

Jing Lu (jlu9@ualberta.ca)

Summary

An interactive shooting game, designed for one or two players.

Abstract

Our Duck Hunt Game is a simple game is to “shoot” the “duck” to stop the moving of it. When the game start, different players need to pick up infrared pump action guns and “shoot” the “duck” for specific times to win the game. However “shooting” a stationary target would be too easy, so to add an extra degree of difficulty, the “duck” that needs to be hit moves. The target is connected to 2 strings; these two strings are connected to two stepper motors mounted on the ceiling. The target can then be moved around in horizontal and vertical directions by using the stepper motors to vary the length of the two strings. The target itself must be hit by an infrared shotgun pulse three times to stop the game. The shotgun must be reloaded /pumped between shots forcing the user to become active to successfully shut off the target. Once the “duck” is shooted specific times or run out of the game time, the two stepper motors move the target safely out of the way to its resting place on the roof. Since the target appears to fly and we use an infrared pump action shotgun, we named our project a Duck Hunt Game.

Table of Contents

Functional requirements of project

What does it do?

It is a shooting game, all aspects of the game are controlled by a DE2 board mounted near the ceiling. The position of the target is controlled by two strings attached to two pulleys mounted near the roof, the location of the target is always known and controlled safely by the DE2 board. The target must be able to reliably communicate with the DE2 Board while the target is in the playing area. This target to board communication, consists of telling the DE2 which team or player has successfully hit the target. The game is designed for only two players/teams due to original project goals.

Were those met? To what degree?

The game aspect - The Infrared hits were properly decoded in software via an interrupt caused by the infrared receiver component. Random motor control movement was verified, and generated in software. The rules were simple to implement, though they were not formally verified via rigorous testing. overall this functional requirement was reasonably met. We fell short by never implementing an audible signal to indicate gun hits to improve the gameplay experience.

The motor control aspect - Motor control was realized at its very basic level, due to hardware difficulties. The implementation of linear increasing velocity profiles was never successfully tested and implemented. Keeping the sensor bar in the safe area was also never formally verified either as a result of circuitry issues. Overall this aspect functionality fell short of functionality requirements.

The infrared communication aspect - This aspect of the game in its entirety was successfully realized. The infrared gun broadcast its signal in a small enough beam across a long enough distance that successfully hitting the target required some skill on the players part. The target successfully and reliably communicated with the DE2 everywhere in the entire play area, And the DE2 was able to successfully receive the proper ID of the gun that hit the sensor bar. Overall this aspect of the game was 100% successful.

Design and Description of Operation

Overall Description of Operation:

There is a DE2 board attached to the ceiling. This board controls all aspects of the game. There are several hardware components connected to the GPIO_O pins of the DE2. There is a 56 KHz infrared receiver, and two stepper motors with their supporting H bridge driver circuits. When a game starts, the DE2 board sends control signals to 2 stepper motors (also mounted on the ceiling) which move the sensor bar in an interesting and random manner about the play area below the DE2. The player uses an 38KHz infrared shotgun to shoot the sensor bar. the shotgun is capable of sending 2 different 38KHz infrared pulses. The sensor bar detects the 38 KHz pulse and converts the signal to 56 KHz and sends it to the DE2. There are two game modes, one where the first person to shoot the sensor bar 2 times wins, and another where you try to shoot the sensor bar as many times as you can in one minute, the DE2 outputs the high score the the LCD screen. Should the batteries in the sensor bar die, one of the buttons on the DE2 shuts off the game.

Infrared design

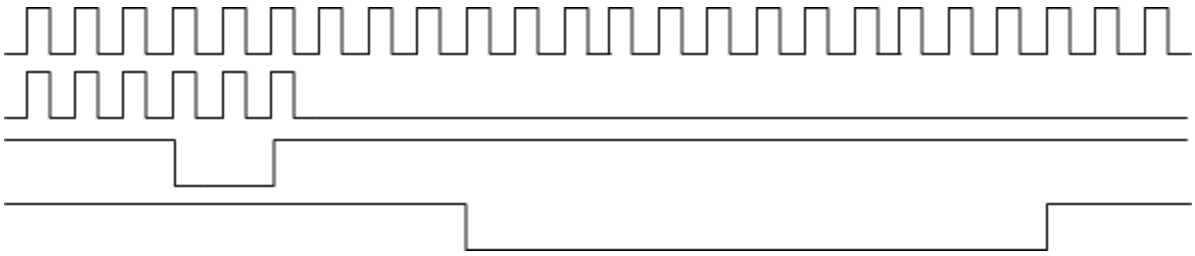
overview of infrared

There are two infrared links in this design. The one between the gun and sensor bar is 38 KHz, and the one between the sensor bar and DE2 board is 56 KHz. sadly a strong 38 KHz signal will punch through the bandpass filter of the 56 KHz receiver, therefore the line of sight between the gun and the 56 KHz receiver was removed to prevent a player from being able to cheat.

the design challenge

The primary design challenge for the infrared was to build a simple infrared protocol to accommodate the variance in valid output from a receiver, while generating the fastest, and most reliable communication possible. To give you some concept of the variance we give you the following timing diagram. The top pulse is given as a timing reference. The second line is the arrival of the minimum

length six pulse burst. The third line is the fastest and shortest valid output potentially caused by this six pulse burst while the fourth line is the slowest and longest valid output of the receiver in response to the six pulse burst.



This is the variance in output possible with just one infrared link.

While it would be easy to use a small FPGA as the sensor bar to decode the signal and then retransmit the bit sequence, thus allowing us to use existing infrared protocols, this option was not taken for one primary reason. That a FPGA is a relatively expensive piece of hardware to do a simple task which can effectively be done with smaller and cheaper parts, and should the motors fail to control the location of the sensor bar safely, wrecking a small FPGA would cause our project to become expensive rather quickly. So we elected cheap hardware option to NOR the output from the 38 KHz receiver (gun to sensor bar) with a 56 KHz oscillator and deal with all the timing variance by creating our own simple infrared protocol. Choosing this inexpensive hardware solution and deal with the timing variance by creating a custom protocol allowed us to reduced the cost of the sensor bar from about $130 (if the FPGA was used) to $33 dollars.

the protocol

The protocol was timing was built up starting with the minimum reliable signal that could be sent across the 56KHz link (sensor bar to DE2) this minimum length signal dictated the absolute minimum amount of time the 38 KHz receiver could hold its signal low. Since the protocol was built off of the shortest reliable signal. We achieved the fastest, most reliable transmission possible across our simple relay. Following the timing guidelines from both receiver data sheets the our custom protocol uses nine 38 KHz pulses to represents a 0 bit, while a valid 1 bit is 27 pulses long.

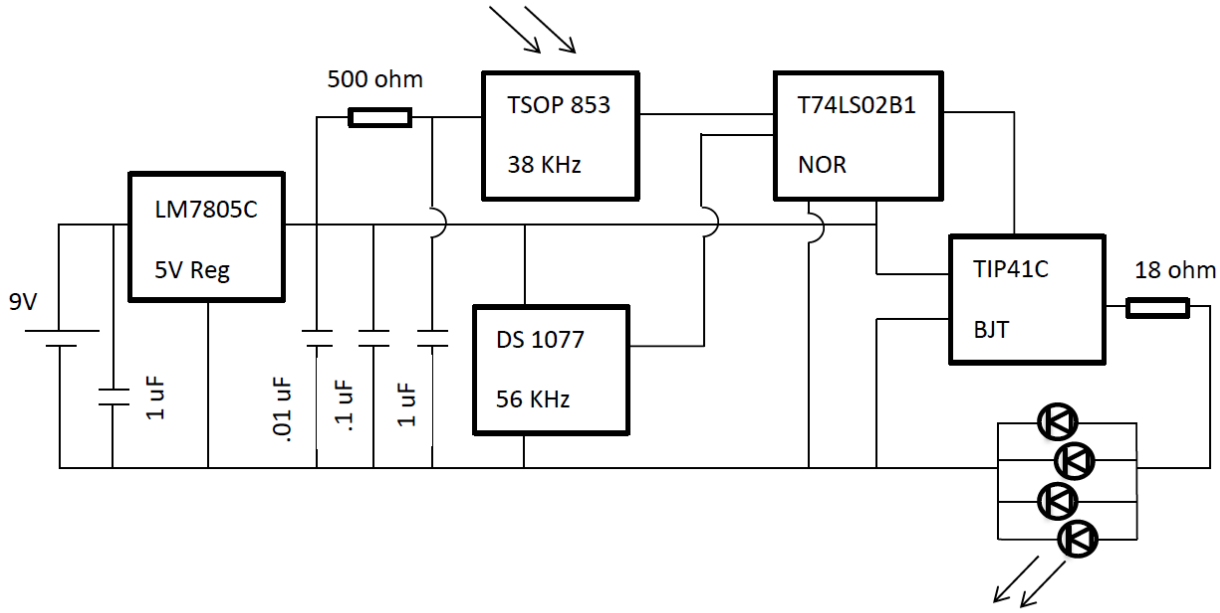
infrared data flow

Since the game was designed for only two players, only 1 bit needs to be transmitted to the DE2 board, and this bit is only transmitted upon a successful hit of the sensor bar. So the IO rate of the infrared making it to the board is negligible, However for the sake of stating the maximum I/O rate using this slow infrared medium, I will assume an equal distribution of 0 and 1 bits and constant streaming by one transmitter, the link would average a measly 1357 bits/second.

Now the infrared bit is first transmitted by the DE0 nano infrared gun transmitter, if this bit successfully hits the 38 KHz TSOP 853 receiver, the bit is relayed via 56 KHz infrared to the TSOP381 receiver connected to the DE2 board. On the DE2 board, there is a custom receiver component that decodes the output of the TSOP381 receiver and drives a single bit pulse to an altera_avalon_pio component. The PIO component edge captures the pulse from the receiver component in a register and interrupts the cpu. The gun bit can then be safely read from the PIO register in the interrupt routine, It is safe to do this only because infrared is such a slow communication medium. Almost 50000 cpu clock cycles will pass before another bit could potentially be driven to the PIO, thus there is zero chance that the PIO will have two overlapping captures of data from the receiver before its interrupt is serviced, especially considering no tasks can mask this interrupt.

the relay schematic

The following is the schematic for the sensor bar relay, I'll just give you the simple overview of the components. It has a 9v battery power supply, this supply is feed into a 5v regulator. The 5 volt regulator has adequate supporting capacitors to prevent voltage ripples, this is especially needed considering the varying draw of the DS 1077 oscillator. The resistor capacitor combo being fed into the 38 KHz TSOP 853 receiver is recommended by the vishay receiver to prevent electrical overstress of the component. the outputs of the oscillator and receiver are feed into a NOR gate, the NOR gate sadly can not safely provide adequate current to the infrared LEDs There for an BJT was used as an amplifier, admittedly it is completely over engineered, however this is what was readily available from our parts room. The reason for the 4 infrared LED was to achieve a wide broadcast area thus increasing the potential playing area for the sensor bar to move in.



the receiver hardware component

Firstly the output of the 56 KHz receiver at the DE2 board is feed into a custom sampling debounce component. This component effectively filters many spurious brief outputs by the receiver due to background noise. The custom debounce components were also used in the infrared transmitter to debounce the button and switch inputs rather than using an RC debounce circuit (for this situation gates are essentially free on the FPGA). After the receiver output passes through the debounce hardware the signal is checked to see if it is in the valid timing range for an infrared burst from the transmitters. If this is true it outputs a single pulse of the correct bit received to a PIO component as mentioned previously.

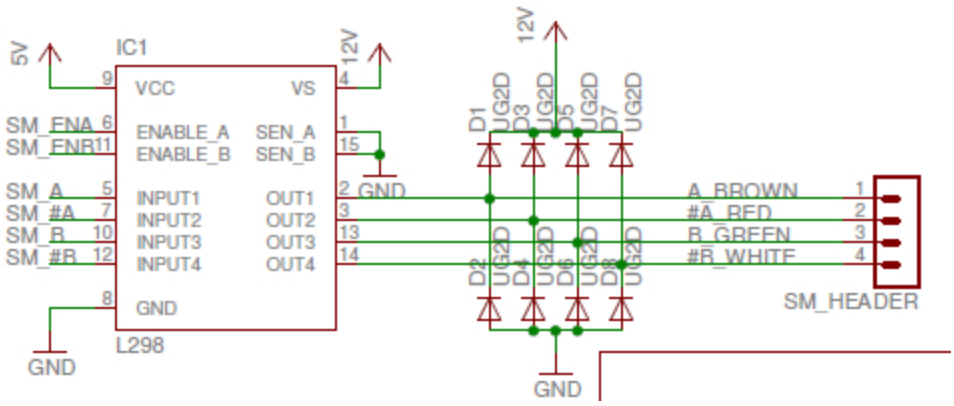
Motor controller design

overview of motor controller

Motor controller can receive commands from software and control motor operate in desired direction, acceleration and velocity (frequency). A series testing including accelerating, direction changing and disabling motor running except for linear increasing speed have been verified in and before acceptance test.

motor schematic

The schematic of motor displayed following, which includes all the signals we designed in this project. In real circuit, SM_ENA and SM_ENB are short circuit to be controlled by one output from DE2 board.



the design challenge

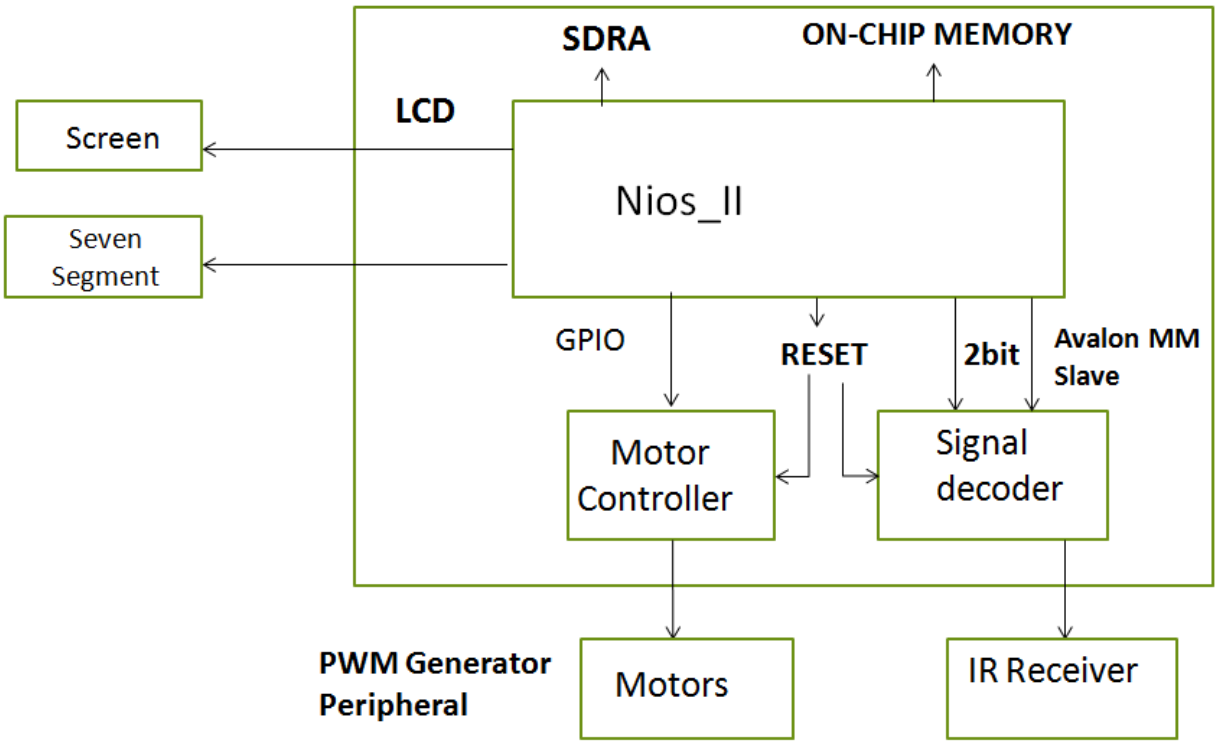
Motor controller design functionally as expected by a completed wiring circuit and successfully controlled both motor turning functionally. However, the circuit broke down at the night before demonstration. The potential reason of circuit breakdown could be:

1) Improper voltage plugged in (rarely, because replacing a new motor driver the circuit still out of work)

2) Internal short circuit (likely, because the circuit wiring more than 50 pins and 100 wires used), the short circuit spot could be anywhere in the circuit.

3) Common ground sharing (probably). Both motor drivers (pin 8, 1, 15) and power supply sharing the common ground pin. If any of them has wiring issue might result in numerous noise and connection problem.

Overall System Hardware diagram and Data Flow



Bill of materials

For the infrared transmitter (total cost $133.21)

- 2 push button switches (digikey P8006S-ND, $0.34)

http://www.digikey.ca/product-search/en?vendor=0&keywords=EVQPAC04M

- 1 rocker switch (digikey R1113QQ11-ND, $0.84)

http://www.digikey.ca/product-detail/en/RR1113QQ11/RR1113QQ11-ND/3778117

- 1 altera terasic DE0 board (digikey P0082-ND, $99.22)

http://www.digikey.ca/product-detail/en/P0082/P0082-ND/2625112

- 1 4AA battery holder with switch (digikey SBH341AS-ND, $3.03)

http://www.digikey.ca/product-detail/en/SBH341AS/SBH341AS-ND/275303

-4 AA batteries (digikey P646-ND, $0.44)

http://www.digikey.ca/product-detail/en/LR6XWA%2FB/P646-ND/2043737

-1 infrared diode (digikey 751-1204-ND, $0.68)

http://www.digikey.ca/product-detail/en/TSAL6200/751-1204-ND/1681339

- 1 short forty pin ribbon cable and header (estimate $10.00)

- 2 perfboard (estimate $15)

- screws, fasteners, wires, header, and mcdonalds straw (estimate $2.00)

For the target/sensor bar (total cost $33.06)

- 1 TSOP85 IR receiver on breakout board (sparkfun SEN-08554, $11.01)

*infrared detection range 45m, half angle detection at 55 degrees

https://www.sparkfun.com/products/8554

- 1 DS1077 programmable oscillator (sparkfun BOB-09116, $5.49)

https://www.sparkfun.com/products/9116

- 1 SN74LS02N Nor gate (digikey 296-1627-5-ND, $0.80)

http://www.digikey.ca/product-detail/en/SN74LS02N/296-1627-5-ND/277273

- 1 LM7805C 5v regulator (digikey LM7805CT-ND, $0.80)

http://www.digikey.ca/product-detail/en/LM7805CT/LM7805CT-ND/458698

- 1 TIP41C BGT amplifier (digikey TIP41C-ND, $0.80)

http://www.digikey.ca/product-detail/en/TIP41C/TIP41C-ND/1052519

- 1 9V battery (digikey P647-ND, $3.02)

http://www.digikey.ca/product-search/en?vendor=0&keywords=P647-ND

- 1 9v battery snap connector (digikey BS6I-HD-ND, $0.73)

http://www.digikey.ca/product-detail/en/BS6I-HD/BS6I-HD-ND/140455

- 4 LTE 4208 infrared led (digikey 160 1029-ND, $0.58)

http://www.digikey.ca/product-detail/en/LTE-4208/160-1029-ND/121707

- 2 resistors (estimate $0.04, based on bulk packaging)

- 4 capacitors (estimate $0.01, based on bulk packaging)
- 1 perfboard (estimate $7.50)
- screws, wires, and fasteners (estimate $0.50)

For the DE2 board and attached circuits (total cost $564.13)

- 1 alerta terasic DE2 development board (estimate $517.72)
- 1 TSOP38156 IR receiver (digikey TSOP38156-ND, $1.14)

http://www.digikey.ca/product-search/en?vendor=0&keywords=TSOP38156-ND

- 2 Howard P/N 1-19-4203 stepper motors (estimate $7.90)

http://www.mpja.com/Stepper-Motor-NEMA-17-12VDC-36-Deg/productinfo/4317%20MS/

- 4 L298 motor drivers (digikey 497-1395-5-ND 4*$5.57= $22.28)

http://www.digikey.ca/product-detail/en/L298N/497-1395-5-ND/585918

- 40 UG2D diodes(digikey UG2D-E3/73-ND 0.17733*64=$7.09)
- 1 perfboard (estimate $7.50)
- screws, wires, and fasteners (estimate $0.50)

Cost of entire project $730.40

Describe and evaluate reusable design units

For the DS1077 programmable oscillator

- maximintegrated.com provides a excel spredsheet for calculating which values registers need must be set to for generating a given clock signal [2].
- http://jondontdoit.blogspot.ca/2012/08/using-ds1077-programmable-oscillator.html provides arduino code for communicating with the DS1077 [4].
- Both of these resources are more useful than the maxim data sheet for the device, the data sheet is full of inconsistencies, both of these resources make communicating with the device easy.

For Stepper motor control

- eewiki.net provides simple schematics and input/output design for different applications of circuits.

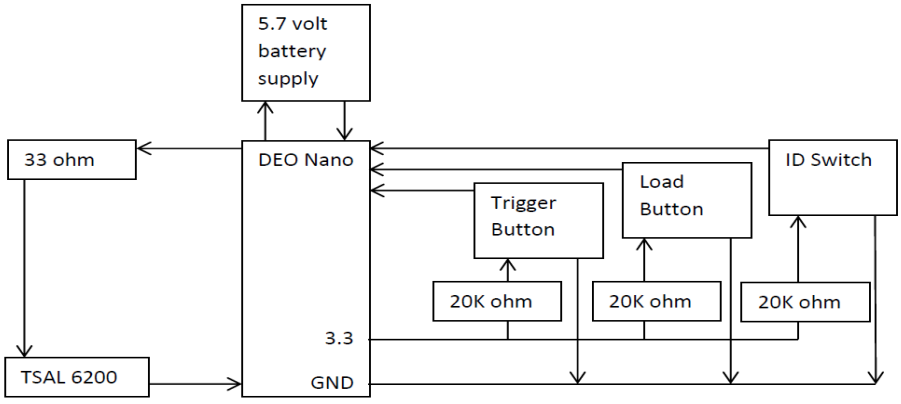
Datasheets

Infrared transmitter block diagram

measured current draw = 100 mA

approximate mAh of 5.7 battery supply = 2300 mAh

estimated gun operation time before battery replacement = 16.1 hours

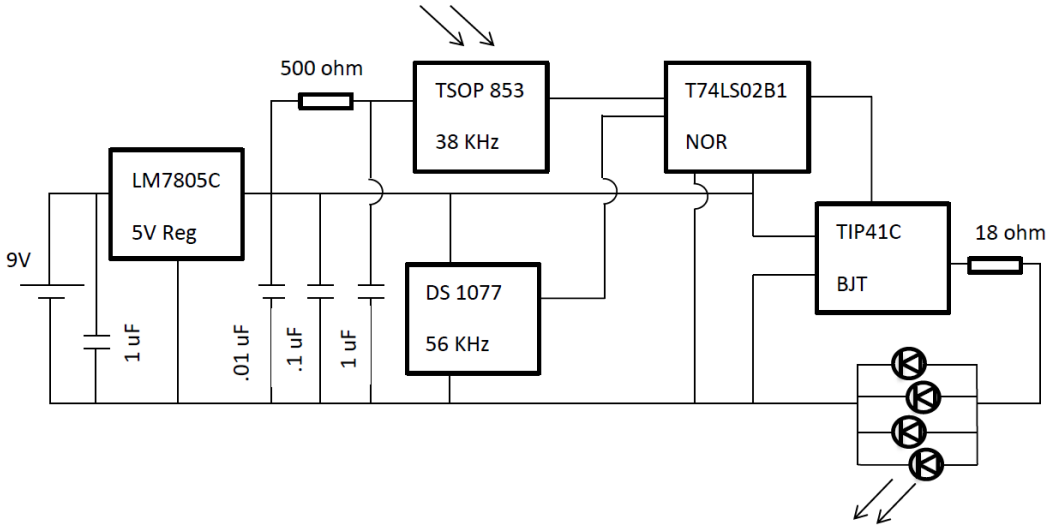


Circuit diagram of the infrared sensor bar

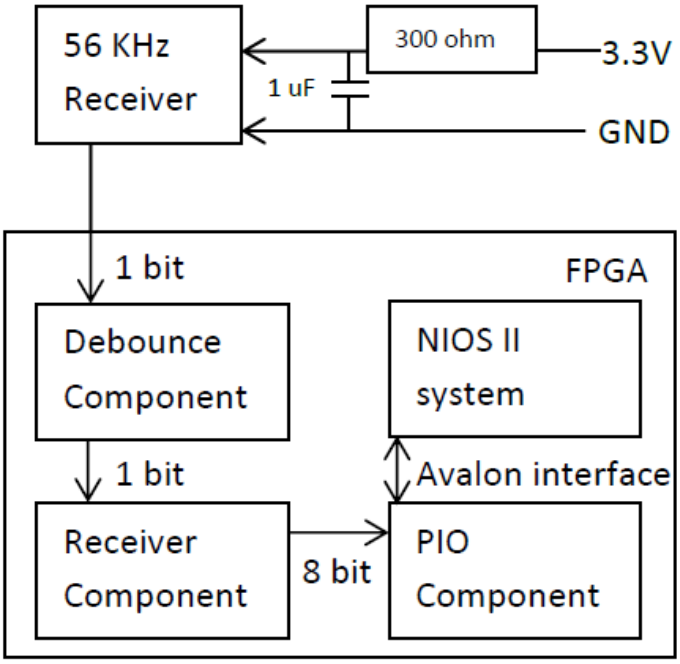
measured current draw with continuous infrared pulsing = 20 mA

approximate mAh of 9v battery = 150 mAh

estimated play time before battery replacement = 5.25 hours



Receiver FPGA component diagram



Performance, user-perspective block diagram, operating conditions.

Expand on the above IO signals to produce a datasheet of your complete embedded system.

Provide calculated and measured power {peak, idle, standby as applicable}. Ask Nancy for a DE2 power measurement wiring harness. Record V and I off bench power supplies.

Describe how each is calculated and measured .

Motor holding torque and detent torque

1)

Variables conversion:

Holding torque *600g*cm=0.0588399N*m (1g*cm=0.0588399N*m)*

Detent torque *80g*cm=0.00784532g*m*

Torque-speed relationship for holding torque is *y=0.0588399-x^2*

Dimension for both axis:

x-Hz or PPS(pulses per second) y-N*m

The maximum running speed (where y is 0) for our project is 250 rpm

250 rpm=4.1666rps=33.333333Hz

So the factor c could be found:

*C=0.0588399/x^2=0.0588399/(33.3333^2)=5.3*10^-5N*m*s^2*

To satisfy the maximum running speed, we have

*Y=0.0588399-5.3*10^-5*x^2*

Y is the torque and *Y=F*d=m*a*d*.

For the holding torque, the acceleration is the gravity, m is the 4 AAA batteries weight

*m=4*7.6*10^-3=0.0304kg*

*F=0.0304*9.8N=0.29792N*

So we have the distance relationship is

*0.29792*d=0.0588399N*m-5.3*10^-5x^2 (x is the Hz)*

So the range of d is *0.1975m-1.78*10^-6*x^2 [m]=19.75-1.78*10^-2*x^2[cm]*

If we run the motor velocity between 50 rpm to 250 rpm, then the range of d is 0 cm to 18.96cm

If we run the motor velocity between 100 rpm to 250 rpm, then the range of d is 0 cm to 16.58cm

2)

Detent torque *80g*cm=0.00784532g*m*

Torque-speed relationship for holding torque is *y=0.00784532-c*x^2*

The maximum starting frequency is 50 rpm, corresponding frequency is 6.667 Hz

Therefore, *c=0.00784532/6.667^2=1.77*10^-4N*m*s^2*

Y is the detent torque and *Y=F*d=m*a*d*.

And a_net=(g+a), if a_net largest, a should have the same direction with g, if the a smallest, then a should have the opposite direction with g. The minimum torque is 0 if a=-g. At this moment, frequency is 6.66Hz, or 50 rpm.

If the a_net has different direction with g, we have

*Y=0.0304kg*a_net*d=0.00784532-1.77*10^-4*x^2*

So *a_net*d=0.2581m-5.82*x^2*

This is the relationship between a_net, d and starting velocity we need to control.

Motor turning calculation

In this project, a desired range from 1Hz to 1000 Hz frequency would be applied random number generator. The corresponding velocity can be calculated as below:

From the datasheet, the motor turns 3.6 degrees/step. In VHDL design, both forward and backward are half step. If driving frequency is 1Hz, motor turns 1.8 degree for each pulse. Therefore, one revolution requires 360/1.8=200 pulse (200 seconds). The minimum velocity is 200/60=3.33 rpm.

For the maximum velocity, driving motor by 1000 Hz, motor turns 1.8 degree for each pulse therefore, one revolution requires 200 pulses taking 200/1000=0.2 second. The maximum velocity is 60/0.2=300rpm.

Sensor bar motion calculation

From motor turning calculation, the velocity range of motor is 3.33 rpm to 300 rpm

unit conversion: *3.33 rpm=0.0555555rps 300rpm=5rps*

In our project, the diameter of pulley is 2 cm. The velocity for minimum and maximum is:

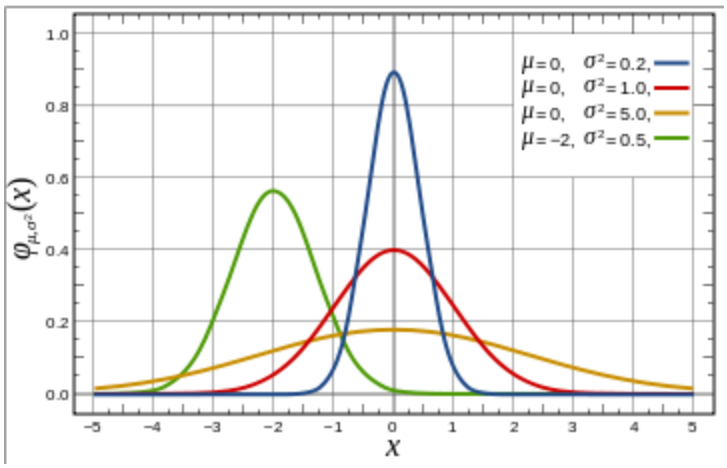
minimum velocity: *0.055555555*3.14*2cm=0.175cm/second*

maximum velocity: *5*3.14*2cm=15.7cm/second*

To avoid sensor bar moving in such high speed, a gaussian random number generator designed to minimize the probability of occurrence of upper bound and lower bound number.Therefore, large and small number are only limited instead avoided which can make the game more playable if the sensor bar can move with a various velocity.

Gaussian random number generator

In our project, we used Gaussian function to generate random number for frequency. There are about 80% random numbers are generated around the average number that is set for function. This property can reduce the jump between two random numbers to ensure the safety and functionality of stepper motors.



(Figure from wikipedia)

[2] Normalized Gaussian curves with expected value μ and variance σ^2. The corresponding parameters are $a = 1/(\sigma\sqrt{(2\pi)}), b = \mu, c = \sigma$

Background Reading

- TSOP 853 Data sheet for understanding how infrared detection and output works with the receiver

- This website goes does a brief summary of the nec infrared protocol, helped me understand different some strategies for implementing an infrared protocol. though i adopted an entirly different strategy than NEC.

http://techdocs.altium.com/display/ADRR/NEC+Infrared+Transmission+Protocol

-DUAL FULL-BRIDGE DRIVER L298, STMicroelectronics, http://www.st.com, 2000

From this article, we learned the block diagram and the properties of DUAL FULL-BRIDGE DRIVER L298, such as pin functions, thermal data, electrical characteristics, etc.

-ECE315 Fall 2013 Lab #5 Schematic, University of Alberta, 31/05/2013.

From the schematic, we learned the electrical element we need to control motors.

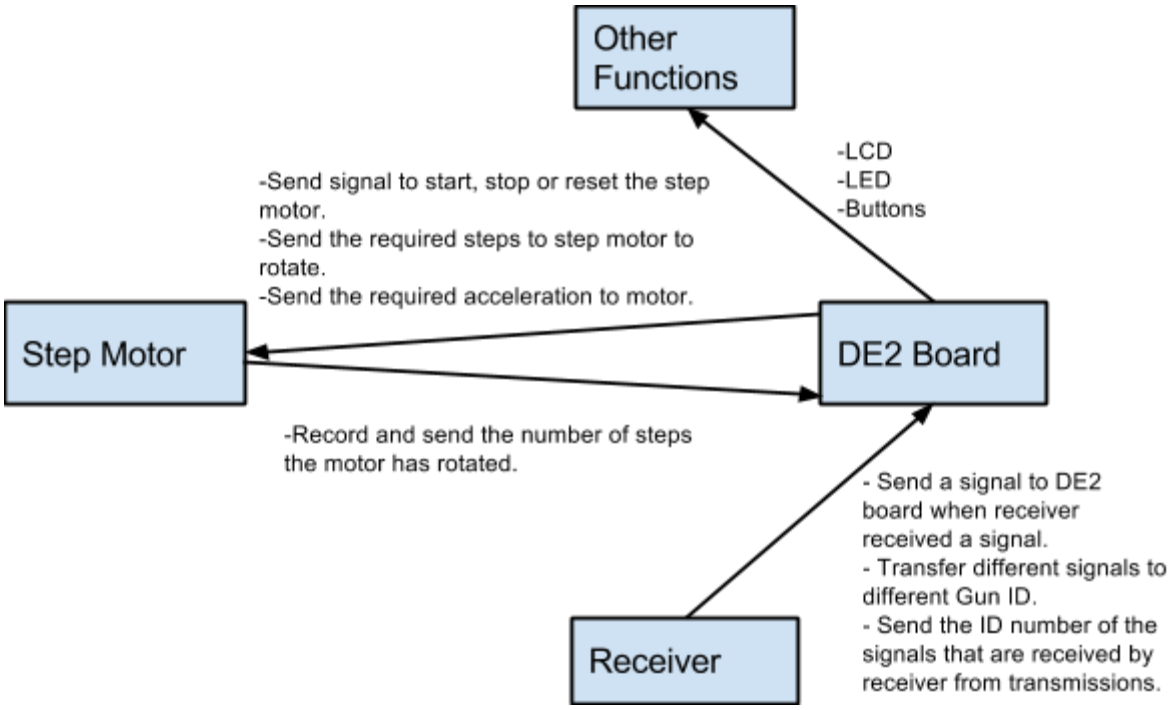
-18_GeneralRVs-3_Gaussian

http://webcache.googleusercontent.com/search?q=cache:DDM7M1aFYJEJ:https://engineering.purdue.edu/~ipollak/ece302/SPRING12/notes/18_GeneralRVs-3_Gaussian.pdf+&cd=2&hl=en&ct=clnk&gl=ca

-Gaussian function - Wikipedia

http://en.wikipedia.org/wiki/Gaussian_function

Software design



Library for step motor, speaker and receiver are all required in C code.

stdbool.h

C library for boolean function and mathematica calculation.

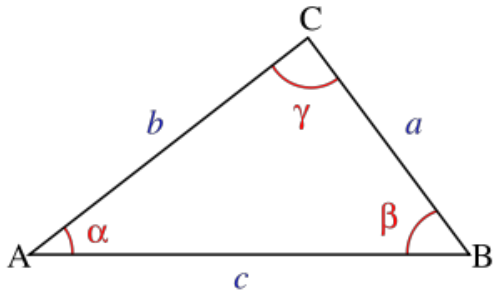
stdlib.h

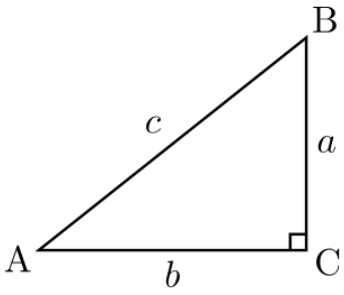
C library for generating random numbers.

math.h

C library for mathematical calculation.

In software design, the safe moving range is set in C program. Since the steps of motors, the width and height of safe area are recorded in C program, we can calculate the length of lines with cosine law. After knowing the length of lines, the right triangle law is used to calculate the ideal x axis and y axis of receiver. When the program notices that the point with x and y axis is moving to the bond of safe area, the program will change the direction automatically to pull the receiver away from dangerous place.





$$c^2 = a^2 + b^2 - 2ab\cos\gamma$$

(Figures from Wikipedia)

Test Plan

Software

1 - Test whether the C code can generate proper frequency to drive motor.

2 - Test whether the C limits its commands to the safe area of IR transmission.

Hardware

- Stepper Motor Tests -

1- confirm the motor has enough torque to move the sensor bar without slipping

2- Test whether the motor can be started, stopped and returned to its original position by the vhdl code that connects the DE2 board to the motor driver circuit. multiple commands to the vhdl in a short time period will discover the delay of this process of sending commands to these being realized in actual motor control.

- infrared tests -

1 - Test the maximum 38kHz link distance.

2 - Test that 38 kHz signals are filtered by the 56kHz receiver.

3 - Test the relay correctly modifies the 38kHz pulse so the data is transmitted correctly to the DE2 board

4 - Test that the relay communicates correctly beyond the corners of the play area

5 - Test the hardware to software communication is done quickly

Results of experiments and characterization

- infrared test results - see above test plan

1 - Test was successful, can broadcast an infrared pulse across the entire length of the lab room.

2 - This test failed, a strong 38KHz infrared pulse is not filtered by the 56 KHz receiver, its output is of variable length and unpredictable. Our solution is to remove the line of sight between the 38 KHz transmitter to the 56 KHz receiver. this effectively solved this filtering issue.

3 - Two oscilloscope leads were connected to the infrared link, and output was edge captured. and verified to be correct and within acceptable ranges.

4 - This test was also successful. the output from the 56 KHz receiver was observed while the infrared relay was placed outside of the designed playing area in all directions, the correct bit transfer was observed in all these cases.

5 - This test was also successful. The hardware to software communication is done via interrupts. The cpu is interrupted immediately after the infrared receiver component finishes receiving a bit. this bit is correctly printed to the LCD screen via software.

1.Gaussian random number generator simulation result

Average	*Standard Deviation*	*Average*	*Standard Deviation*	*Average*	*Standard Deviation*
20	*5*	*500*	*200*	*100*	*200*
Number	*Random Number*	*Number*	*Random Number*	*Number*	*Random Number*
1	*36*	*1*	*1157*	*1*	*1657*
2	*16*	*2*	*355*	*2*	*855*

3	*21*	*3*	*543*	*3*	*1043*
4	*14*	*4*	*285*	*4*	*785*
5	*24*	*5*	*661*	*5*	*1161*
6	*13*	*6*	*225*	*6*	*725*
7	*22*	*7*	*618*	*7*	*1118*
8	*8*	*8*	*224*	*8*	*524*
9	*24*	*9*	*678*	*9*	*1178*
10	*26*	*10*	*752*	*10*	*1252*
11	*17*	*11*	*397*	*11*	*897*
12	*24*	*12*	*686*	*12*	*1186*
13	*17*	*13*	*386*	*13*	*886*
14	*13*	*14*	*251*	*14*	*751*
15	*17*	*15*	*419*	*15*	*919*
16	*20*	*16*	*506*	*16*	*1006*
17	*27*	*17*	*812*	*17*	*1312*
18	*18*	*18*	*426*	*18*	*926*
19	*20*	*19*	*507*	*19*	*1007*
20	*19*	*20*	*482*	*20*	*982*

2.Motor random generated velocity and corresponding frequency after calculation:

Average	*Standard Deviation*
800	*200*

Number	*Random Frequency*
1	*17157*
2	*38166*
3	*29654*
4	*42734*
5	*26013*
6	*47618*
7	*27232*
8	*77159*
9	*25561*
10	*23763*
11	*35867*
12	*25353*
13	*36442*
14	*45371*
15	*34769*
16	*31016*
17	*22481*
18	*34434*
19	*30977*
20	*31968*

3.Power consumption measurement:

V=12.1v, I=0.47 mA, Duty cycle=37.5%

4.Performance, resolution, errors

Motor velocity:3.33rpm~300rpm

Driving frequency: 1Hz~1000 Hz

sensor bar moving velocity:0.175cm/sec~15.7cm/sec

5.Motor torque

Use a pen bag which weighs about 0.2lb can be dragged by a **single** motor from stayed to accelerate to maximum velocity we used in this project. The actually duck we used in this project is actually lighter than 0.2 lb. The torque is satisfied with our design.

6.Motor turning

By simply sending 500 pulses to drive motor turns 5 revolutions. The theoretic moving distance should be 31.4 cm. The actual displacement is 31cm and steps counted as 500. The error is slight and we can expect the motion of duck is controllable.

Safety

Since having a sensor bar moving around via strings above our heads is unwise, a frame was built to approximately chest height. This way the sensor bar moves around below this height to reduce the risk of being close lined, hit by the sensor bar, or hurt by the thin fishing line. However this introduces a new risk of the moving pulleys on the stepper motors are now reachable. Some guards need to be added to protect people from the pulleys. It also brings the circuitry down to a height that is accessible. The highest voltage and current in the circuit is 12v and 480 mA, to mitigate risk, the wires were safely connected together, and these connections wrapped properly with electrical tape. To further mitigate risk conduit should be used to run the lines from the motors to the driver circuit. Additionally a safety shield should be installed over the driver circuits to prevent users from touching or playing with the circuit. As well a future safety feature would be to use weaker fishing line, the current fishing line is very difficult to break, if the line were able break, this would reduce the likeliness of harm to a person should they accidentally be snagged by it.

Environmental impact

The infrared transmitter and link emit light at 940 nm at occasional intervals in 38kHz and 56kHz modulated pulses. Since the application of this device is for household use, the majority of the infrared will be kept in the house. Also since the gun does not use a previously developed IR protocol there is little chance of it sending a valid command to any other device. The IR subsystem of this project is RoHS compliant aside from the lead solder used to connect the breadboards, this was used only used as it is easier to work with, under actual production the infrared system would be entirely RoHS compliant. The data sheet for the stepper motors does not mention RoHS compliance so we can not make any statement about this submodule of the system.

Sustainability

Motor Power consumption

1.Input: lmax: 480mA L=43 mH

Voltage: 12V Revolution/step: 100 rev/sec

2.Formula:

Current through the coil is proportional to the time that the voltage has been applied, and inversely proportional to the inductance.

$$I= V*T/L \qquad T= I*L/V$$

For one step the current must go from 0 to Imax and back to 0, or alternatively from -Imax to +Imax.

$$I= 2*Imax \qquad T= L*Imax*2/V$$

T is the number of seconds for a single step.

To compute maximum revolutions per second - divide seconds per step by steps per revolution.

$$Rev/sec = V/(L*2*Imax)/(steps/rev)$$
$$=12/(0.00043*2*480*10^{-3})/100$$
$$=290.1rev/sec$$

$$Pmax= 2*Imax*V=2*0.48*12W=11.52W$$

Pmax occurs not when the motor is going max speed because the current is a triangle wave. Pmax occurs when the slope of the current is small compared to the on holding time of the step pulse.

The operating consumption

As the duty cycle in motor controller is 12/32=37.5%.Therefore, the average consumption is

$$P=V*I*duty\ cycle=12V*0.48A*37.5\%=2.16W$$

As an approximation, switching losses for each output are calculated as follows:

$$Psw= Psw\_rise + Psw\_fall$$

where Psw is the total switching loss (in watts) for one output, and Psw_rise is the power dissipated during the rising edge, and PSW_FALL is dissipated during the falling edge.

Expanding on this:

$$Psw\_rise = ½ \times Vm \times Iout \times Tr \times Fsw$$
$$=½*12v*0.48A*1/50,000,000sec*50,000,000$$
$$=2.88w$$

and

$$Psw\_fall = ½ \times Vm \times Iout \times Tf \times Fsw$$
$$=½*12v*0.48A*1/50,000,000sec*50,000,000$$
$$=2.88w$$

Then

$$Psw=2.88w+2.88w=5.76w$$

3.Output:

Speed: 3.33rpm~300rpm

Maximum Power: 11.52W

Average Power consumption: 2.16W

References:

[1] 26184.24F Datasheet, Allegro MicroSystems Inc., Worcester, Massachusetts, Copyright © 2002, 2003
http://www.sparkfun.com/datasheets/Robotics/A3967.pdf

[2] DS1077 EconOscillator/Divider Datasheet, Freq Calculator, Maxim Integrated ™, San Jose, CA,
http://www.sparkfun.com/datasheets/Components/SMD/DS1077.pdf
http://www.maximintegrated.com/design/tools/calculators/files/DS10775VFREQ.xls

[3] TSOP853 and TSOP 381 Datasheet, Vishay Semiconductors,
https://www.sparkfun.com/products/8554
http://www.vishay.com/docs/81743/temt6202.pdf

[4] I2C example for programming oscilator, sparkfun.com
http://www.sparkfun.com/datasheets/BreakoutBoards/DS1077v2.c

[5] Example Stepper Motor Control Programs
http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/stepper-motors/forth-c-examples

[6] PWM module design EEwiki
http://eewiki.net/pages/viewpage.action?pageId=4096117#SteppingMotorControl(withVHDL)-CodeDownloads

[7] 18_GeneralRVs-3_Gaussian
http://webcache.googleusercontent.com/search?q=cache:DDM7M1aFYJEJ:https://engineering.purdue.edu/~ipollak/ece302/SPRING12/notes/18_GeneralRVs-3_Gaussian.pdf+&cd=2&hl=en&ct=clnk&gl=ca

[8] Gaussian function - Wikipedia
http://en.wikipedia.org/wiki/Gaussian_function

[9] Gaussian distribution random numbers generator in C

http://habibidavid.blogspot.ca/2012/03/gaussian-distribution-random-numbers.html

[10] Motor consumption calculation tutorial

http://pegasus.cc.ucf.edu/~cham/eas5407/project/Intro_stepper.pdf

[11] Law of cosines

http://en.wikipedia.org/wiki/Law_of_cosines

[12] Right triangle

http://en.wikipedia.org/wiki/Right_triangle

[13] I2C arduino code

http://jondontdoit.blogspot.ca/2012/08/using-ds1077-programmable-oscillator.html

Appendices

Quick start manual

1. open the .qar file in quartus
2. compile the design and .sof program it to the DE2
3. connect the output of your 56 KHz receiver to GPIO_0(32), as well connect the additional EOS circuitry for the receiver between power and ground, follow the receiver hardware block diagram for additional information
4. connect the motor driver circuit to stepper motors, and to the proper GPIO pins, reference the top level file for the correct pin locations
5. follow the circuit diagram for constructing the infrared relay, suspend it from the strings from the stepper motor pulleys
6. move to eclipse and open up the software, build the project and run it as nios 2 hardware
7. open the Transmit_P.qar file, compile the design, generate the .jic file and program the DE0 nano. Be sure to follow the hardware diagram for the transmitter. GPIO_0_IN(0) maps to the load button, GPIO_0_IN(1) maps to the fire button, GPIO_0(2) maps to the ID switch, and GPIO_0(4) maps to the infrared led output.
8. press a key on the keypad to start a game mode (button 0 starts the 3 shot game after a 10 second delay, button 1 starts a three shot game immediately, button 2 starts a high score game after 10 seconds, button 4 starts a high score game immediately).
9. the switch on the far left resets the system should the software freeze or the motors go out of control.

Future work

1. the game was originally meant to be an alarm clock, adding the alarm feature would be great.
2. adding game functionality to record the fastest kill time of the duck and displaying a counter of the game time is a good addition.
3. adding a high score/and fastest kill time server to allow people to compete against their friends with the same alarm clock would be interesting.
4. adding a multi bit addressed transmission for the protocol would be great to prevent other devices from registering as a gun hit is a needed feature. and allowing more gun ID's
5. adding sound would be an obvious improvement.
6. implementing feedback control of the motors is needed.
7. changing the hardware from stepper motors to DC motors is also needed to move the sensor bar quicker.

Hardware documentation

See page 13 for the DE0 nano block diagram

See page 14 for the Relay circuit schematic

See page 14 for the DE2 receiver block diagram

See page 8 for the DE2 block diagram

Source code section

See attached software.zip file for C code

See the .qar files for VHDL code

See the attached testbench.zip for receiver testbench