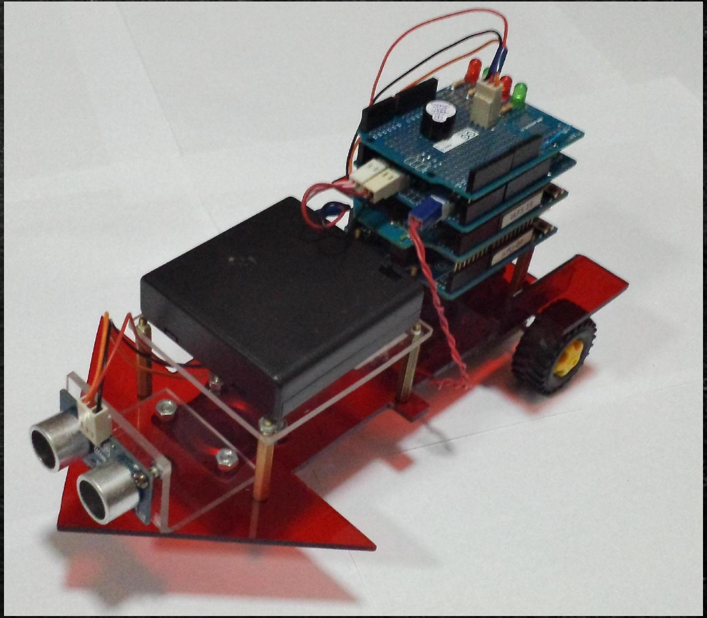
Arduino
(Iron-Car Mk III)



Mulliana Yusuff
Chia Jun Jie
Daniel Soh Kang Wen
Ilyasa Syakir

Content

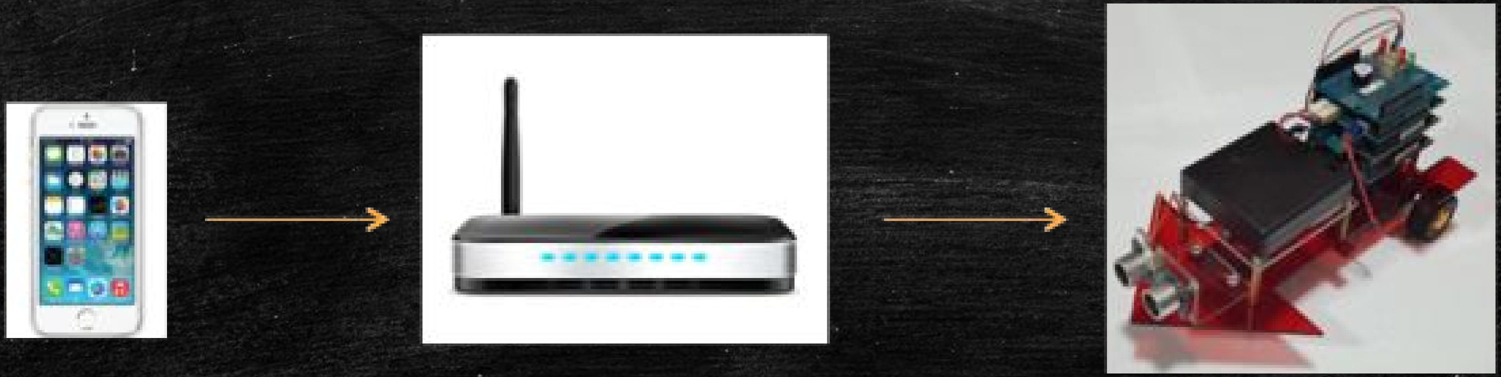
1) Introduction
2) Mechanical Aspects
3) Software
4) Enhancement
5) Project Demonstration

Introduction

Project introduction
Objectives
Overall system design

Project Introduction

Design a Wi-Fi Controlled In-vehicle Surveillance System



Objectives

Main Objective:

1) Construct a 4-Wheel electric cart with basic maneuver controls

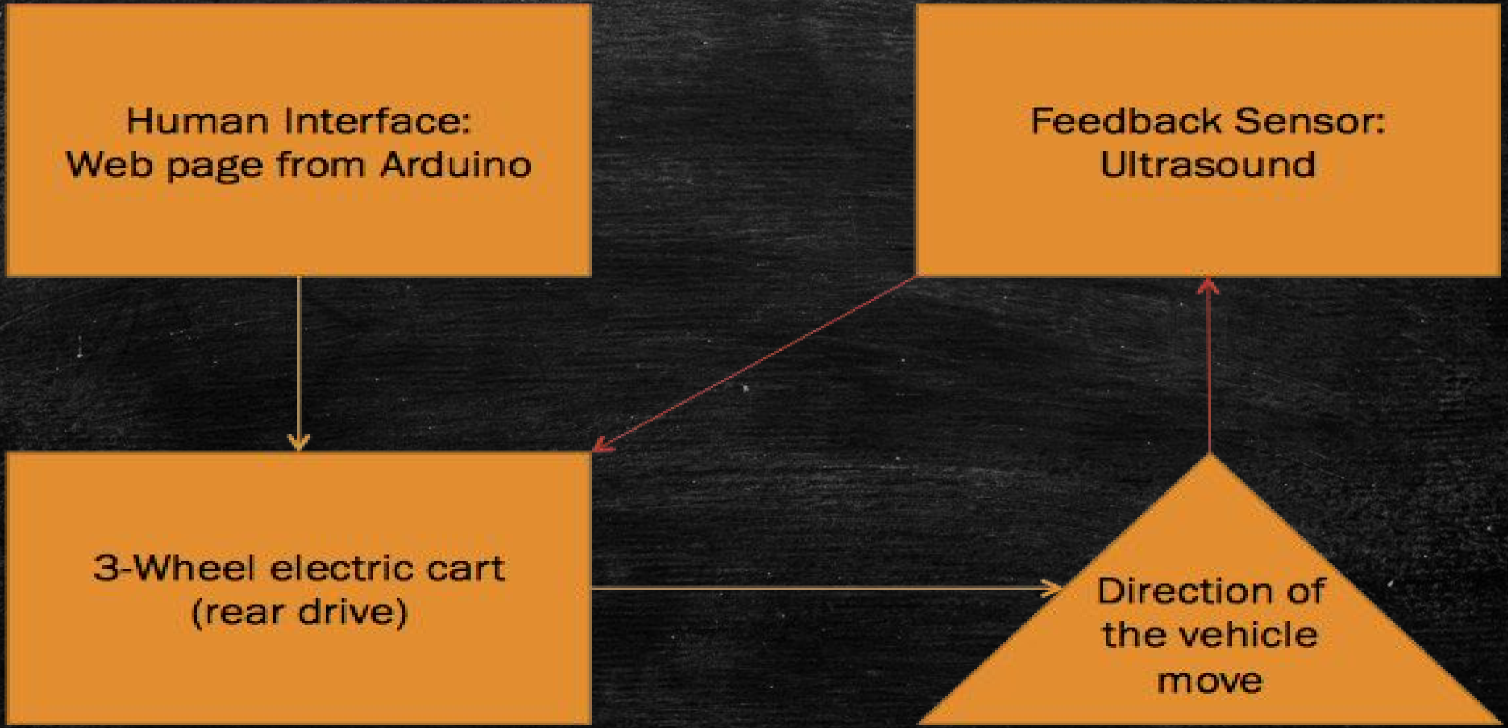
2) Control electric cart through Wi-Fi connectivity

3) Using LED as an indication of the cart's movement

Enhancement:

1) Using an ultrasound sensor to detect the distance of obstacles and avoid them

Overall System Design



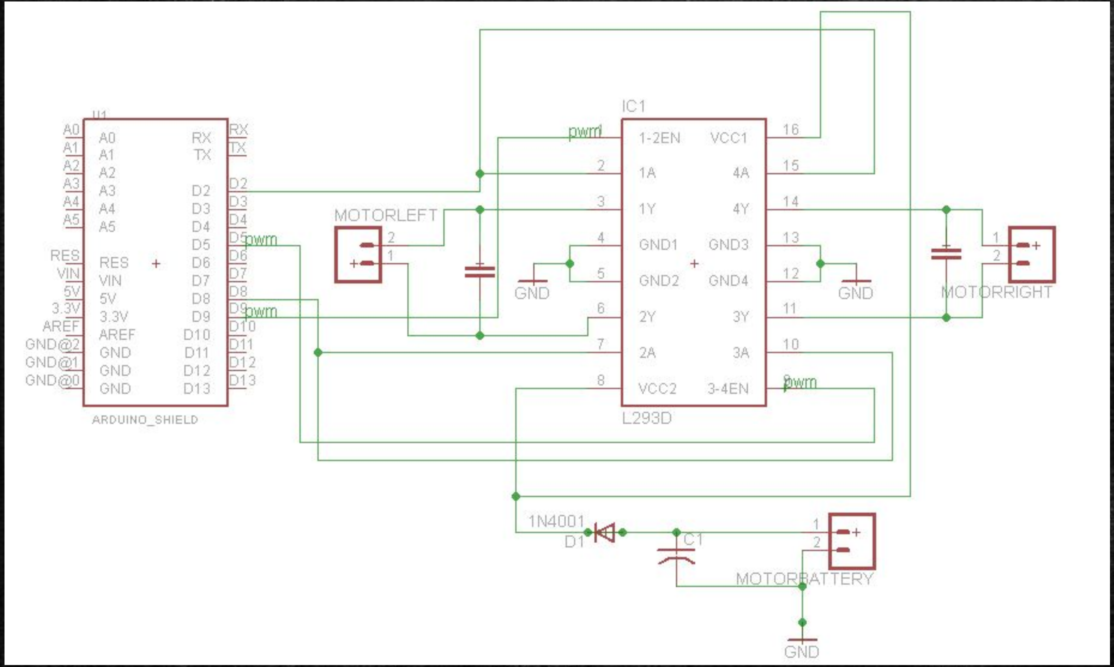
Human Interface:
Web page from Arduino

Feedback Sensor:
Ultrasound

3-Wheel electric cart
(rear drive)

Direction of
the vehicle
move

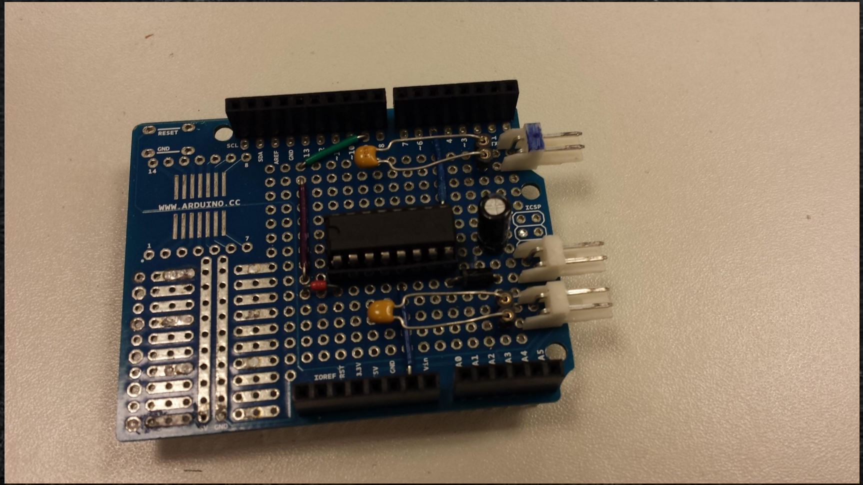
System Design (Schematic)

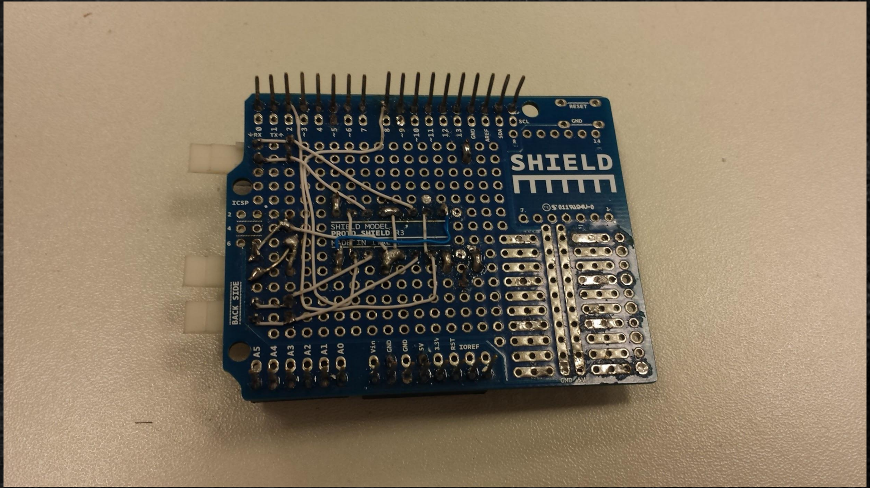


WiFi-Shield

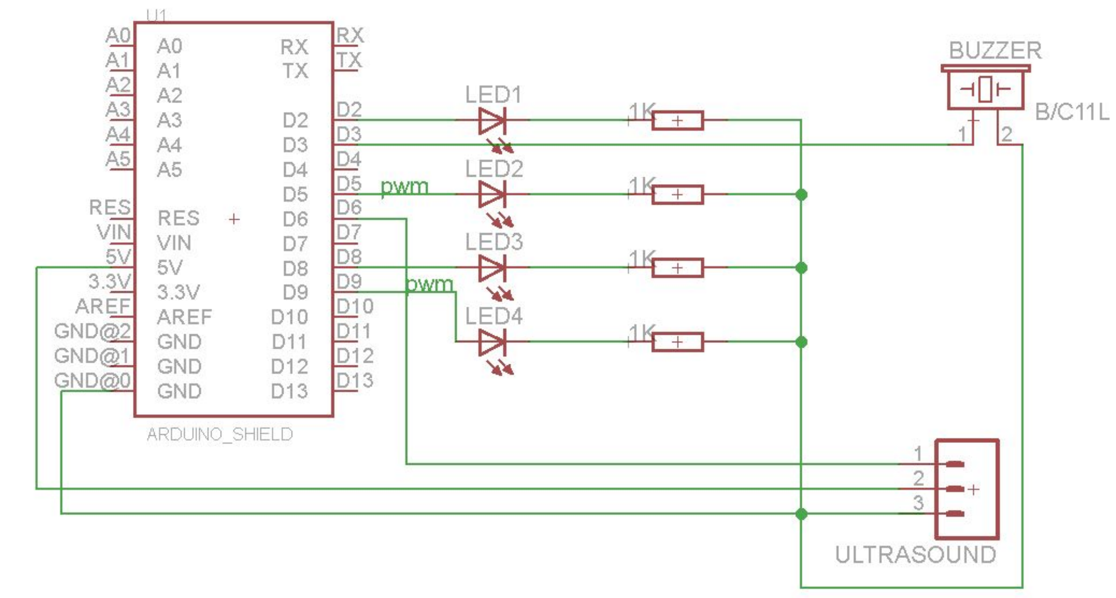
Arudino Uno
14 Digital input/output pins (6 pins Pwm 3,5,6,9,10, 11)
6 analog input
4,7,10,11,12,13 (Cannot be used. Used by wifishield)

Pin 2 -DIRA1/B2
Pin 8- DIRB1/A2
Pin 9- Enable for Right Motor (Pwm)
Pin 5- Enable for Left Motor (Pwm)


WWW.ARDUINO.CC
RESET
GND
ICSP
IOREF
RST
3.3V
5V
GND
Vin
A0
A1
A2
A3
A4
A5

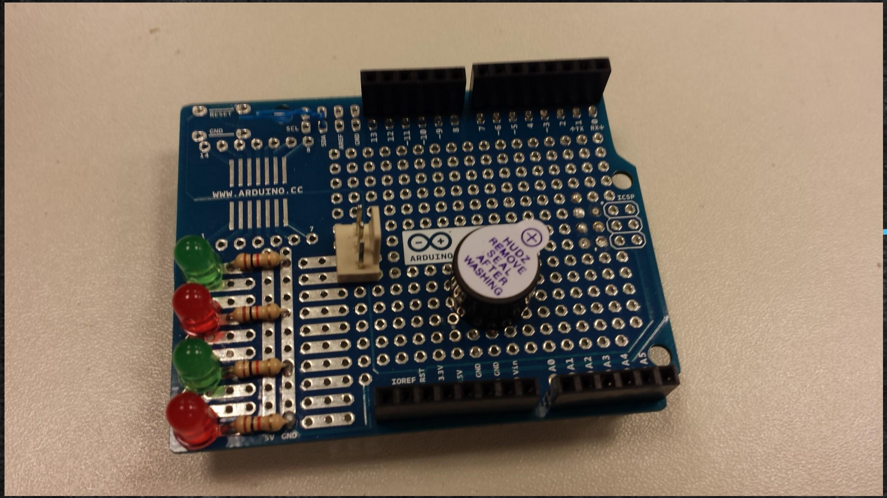

SHIELD
SHIELD MODEL
PROTO SHIELD R3
ICSP
BACK SIDE
RESET
GND
SCL
A5
A4
A3
A2
A1
A0
Vin
GND
GND
5V
3.3V
RST
IOREF

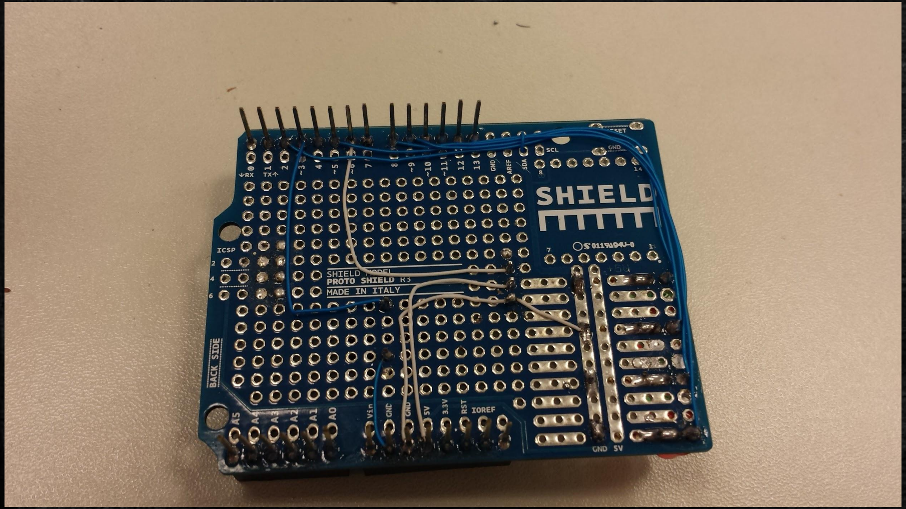
System Design (Schematic)

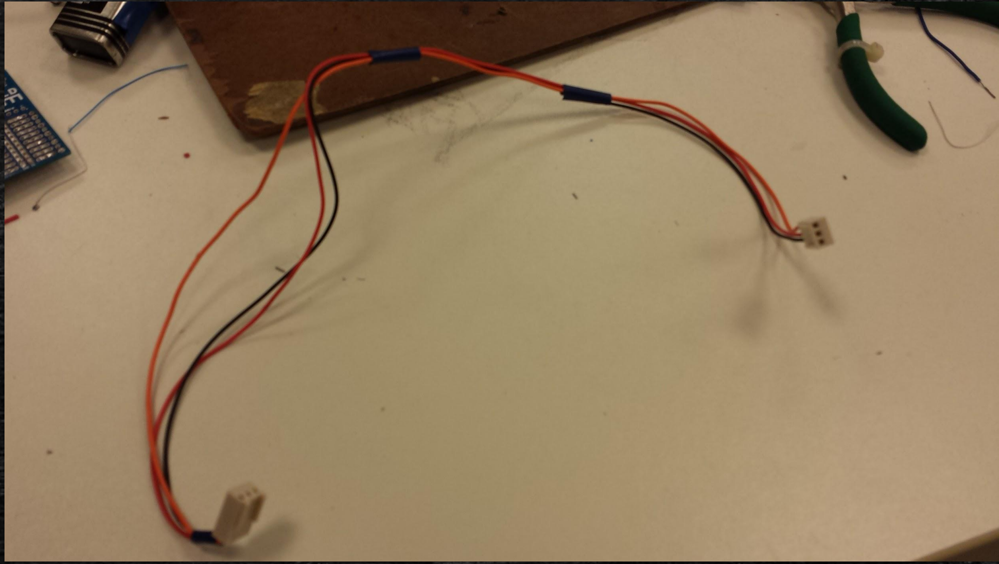


Buzzer - Pin 3
Ultrasound 5v - 5v of WifiShield
Ground -Ground of WifiShield
Signal- Pin 6 of WifiShield
Leds 1- Pin 2 of Wifishield
Leds 2- Pin 5 of Wifishield (Pwm)
Leds 3- Pin 8 of Wifishield
Leds 4- Pin 9 of Wifishield (Pwm)

LED and Ultrasound Sensor


RESET
GND
SCL
SDA
AREF
WWW.ARDUINO.CC
ICSP
ARDUINO
HUDZ REMOVE SEAL AFTER WASHING
IOREF
RST
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
5V GND


RESET
SCL
GND
SHIELD
ICSP
SHIELD MODEL
PROTO SHIELD R3
MADE IN ITALY
BACK SIDE
A5
A4
A3
A2
A1
A0
Vin
GND
GND
5V
3.3V
RST
IOREF
GND 5V



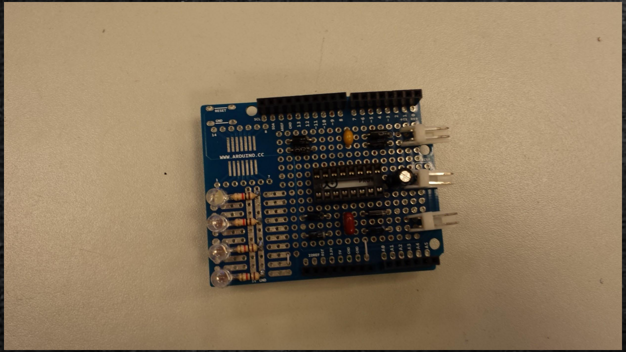
LEDS

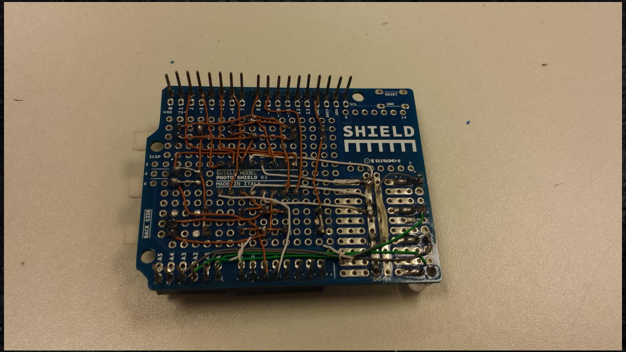
Enable (Pwm) = Green (Pin 2 and 9)

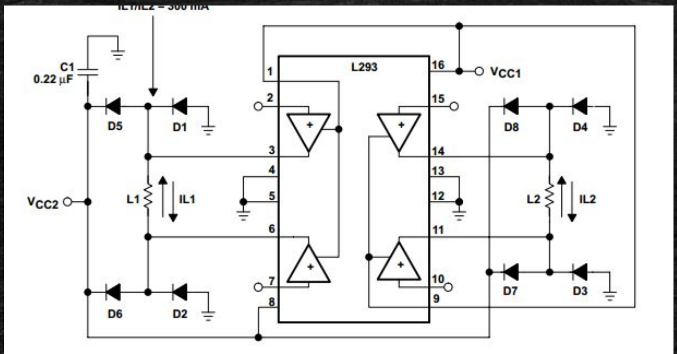
Drive = Red (Pin 5 and 8)

Right			**Left**
(Pin 2 Red)	Pin 5(Green)	Pin 8(Red)	Pin 9(Green)
Forward = Red	Green	Not Blinking	Green
Back = Not Blinking	Green	Red	Green
Left = Red	Green(High Pwm)	Not Blinking	Green(Small Pwm)
Right =Red	Green(Small Pwm)	Not blinking	Green(High pwm)

Problems Faced and learning points







Always Check datasheet before soldering.

Diodes are already implemented inside the IC.

Protect IC and the Source

Prevent backward current to the source.

Problems Faced and learning points

- It is not easy to strip the wire without wire stripper. By using cutter, the wire easily broken.
- The board is very small, have to organise the components well in order for neater wiring.
- Always do continuity check after soldering each point to make sure it is connected.
- Not to expose any bare wires. Solder have to becareful so that it won't burn other wires.
- Have to desolder 2 points at a time to take out a component.

Hardware

Car design
Difficulties

Car Design

We have designed the electric cart to make it look like a normal Tamiya car. We have taken the picture of the Tamiya car on the right as an inspiration. The arrowhead and the folded wheel skirts are there for the aesthetics points. Also, we have chosen a transparent red acrylic with 4mm of thickness so that it does not break easily and could withstand the weight of the Arduino, battery pack and also the sensor.

Difficulties

We encountered a few problems while making the acrylic car:

1) The width of the car we made for version 2 was too wide. The wheels could not be fully attached to the axle of the gear motor.

2) The folding of the wheel skirts was tedious because we had to reheat for about 3 to 4 times to get a right angle because the space of the equipment we used was too small.

3) The gear motor ratio has different speeds for both motors. We disassembled the gear motor and tried to assemble to the low speed gear ratio, however, both the motors were turning in different speeds.

4) Some holes for the mounts have to be drilled slowly in order not to crack the acrylic piece.

Software

Controlling the orientation of the electric cart

Programming
Difficulties faced

Programming - Truth Table

HIGH = 1

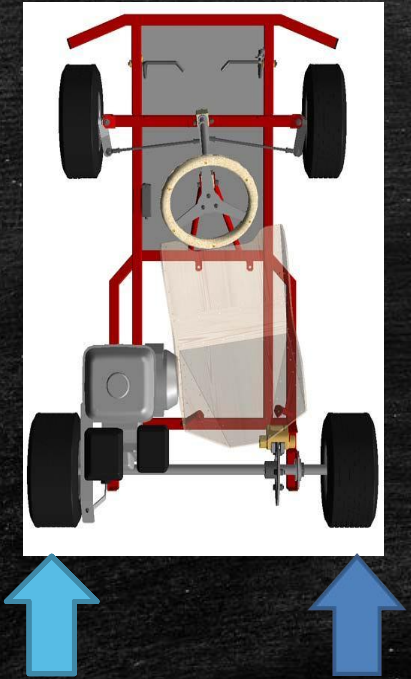
LOW = 0

		RIGHT MOTOR			LEFT MOTOR		
	Function	ENABLE 1	DIRA1	DIRB1	ENABLE 2	DIRA2	DIRB2
	I/C pin	1	2	7	9	10	15
	Arduino Pin	9	2	8	5	8	2
Direction	Front	1	1	0	1	0	1
	Reverse	1	0	1	1	1	0
	Left	Small PWM	1	0	Higher PWM	0	1
	Right	Higher PWM	1	0	Small PWM	0	1

Programming – Moving Forward

HIGH = 1

LOW = 0



Left Wheel

Anti-clock wise

Right Wheel

Clock wise

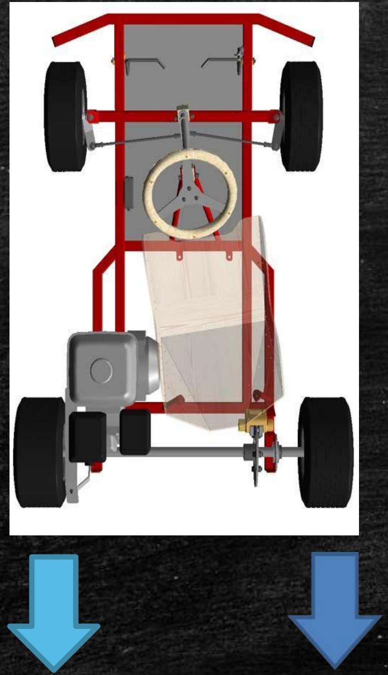
	LEFT MOTOR		
Function	ENABLE 2	DIRA2	DIRAB
I/C pin	9	10	15
Signal	1	0	1

	RIGHT MOTOR		
Function	ENABLE 1	DIRA1	DIRB1
I/C pin	1	2	7
Signal	1	1	0

Programming – Moving Backwards

HIGH = 1

LOW = 0



Left Wheel

Clock wise

Right Wheel

Anti-clock wise

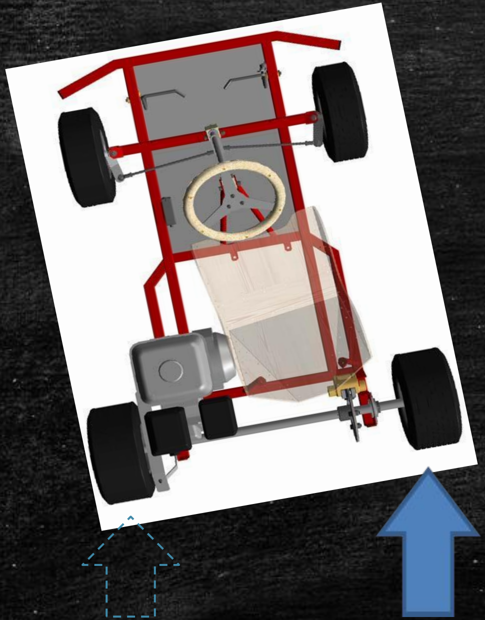
LEFT MOTOR			
Function	ENABLE 2	DIRA2	DIRAB
I/C pin	9	10	15
Signal	1	1	0

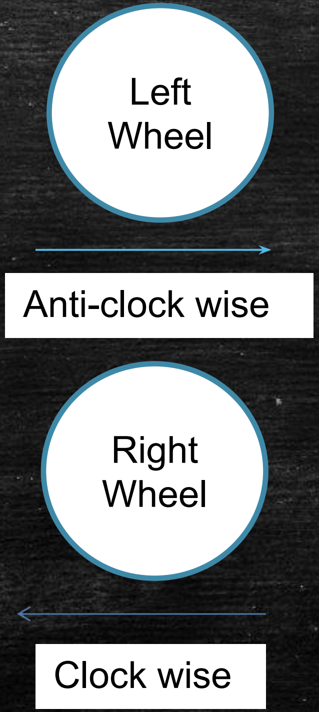
RIGHT MOTOR			
Function	ENABLE 1	DIRA1	DIRB1
I/C pin	1	2	7
Signal	1	0	1

Programming – Moving Left

HIGH = 1

LOW = 0





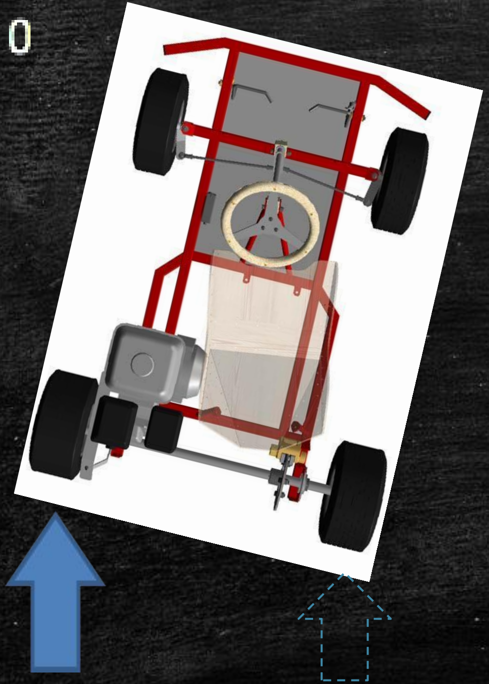
LEFT MOTOR			
Function	ENABLE 2	DIRA2	DIRAB
I/C pin	9	10	15
Signal	Low PWM	0	1

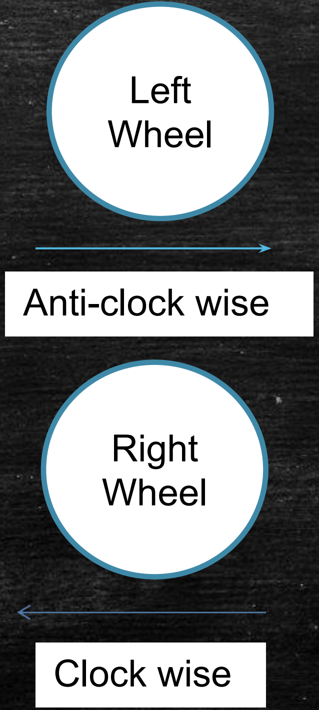
RIGHT MOTOR			
Function	ENABLE 1	DIRA1	DIRB1
I/C pin	1	2	7
Signal	High PWM	1	0

Programming – Moving Right

HIGH = 1

LOW = 0





LEFT MOTOR			
Function	ENABLE 2	DIRA2	DIRAB
I/C pin	9	10	15
Signal	High PWM	0	1

RIGHT MOTOR			
Function	ENABLE 1	DIRA1	DIRB1
I/C pin	1	2	7
Signal	Low PWM	1	0

Difficulties

- Wi-Fi shield not updated to the latest version
- Not compatible to Arduino 1.0.5
- Use Arduino 1.0.3 to load the Wi-Fi program
- Understanding the physics behind the movement
- Program was haywire during the first few testing phases
- Program was modified to remove the errors

Enhancement

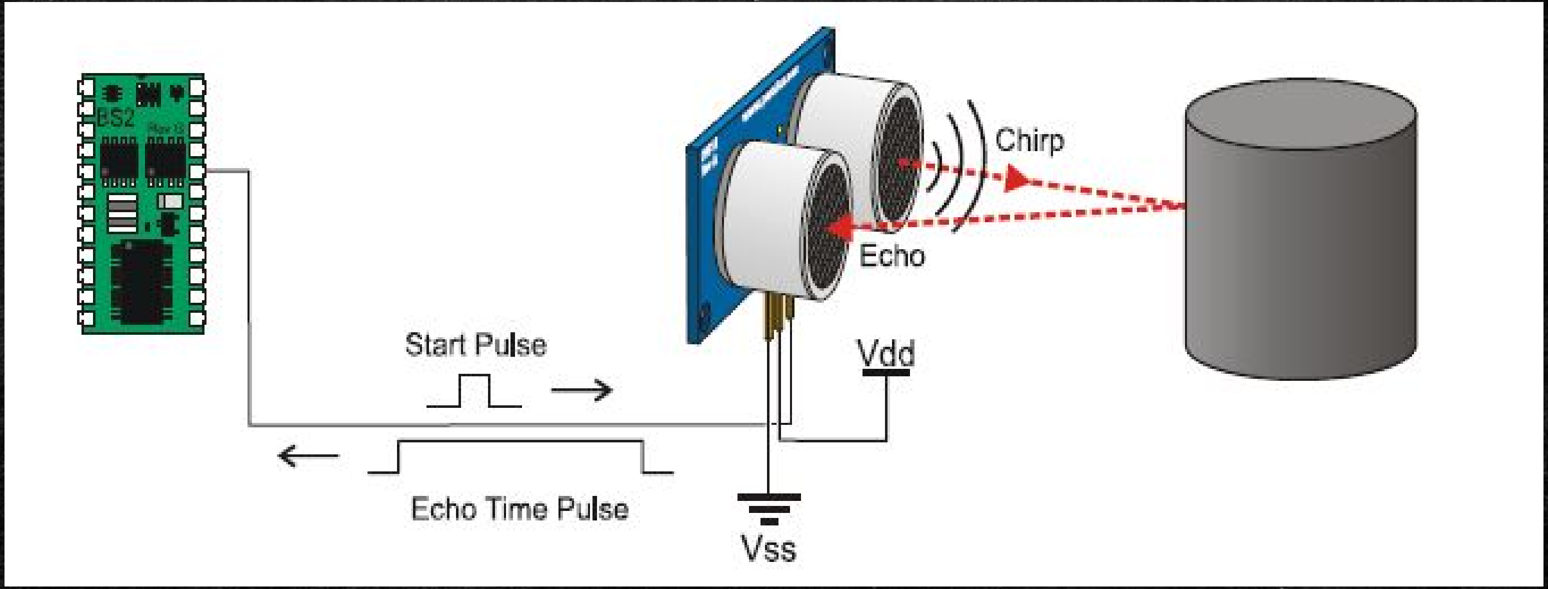
Arduino Ultrasound Sensor and Time Interrupt

Details
Programming
Difficulties faced

Details of the Arduino Ping))) ™

- Parallax PING)))™ ultrasonic distance sensor
- 2 cm (0.8 inches) to 3 meters (3.3 yards)

How does it work?

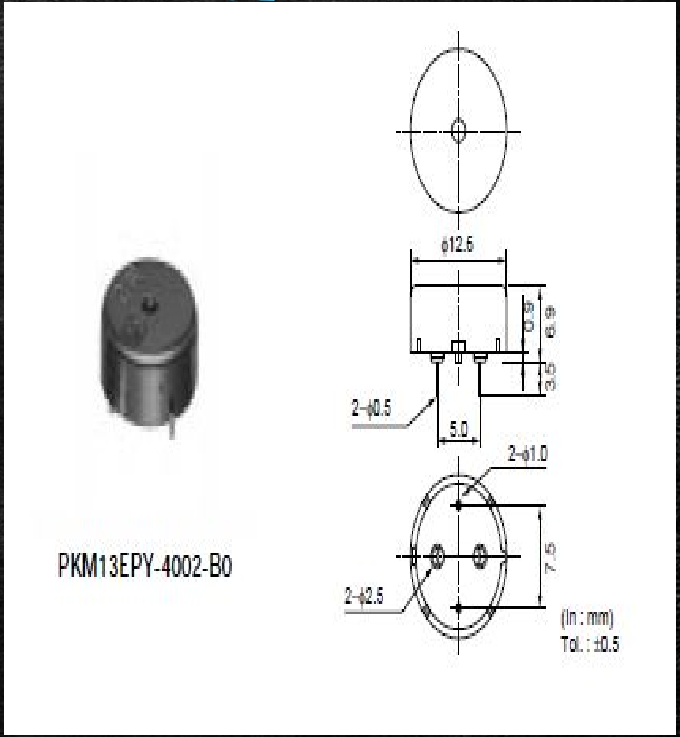


Ultrasound Limitation

- Ultrasound will not work outside its detecting range
- Object too far(>3m) : Cut off the measurement at 118 inches
- Objects too near (<2cm): The measurement will longer than the actual distant. Time delay is to fast for the Ultrasound to detect. Ultrasound ended up measuring a couple time during period.

Buzzer Details

- Features
- Low power consumption
- No contacts therefore, no noise and highly reliable



Buzzer Program

```
void Buzzerdetector(){
long inches;


inches = pingOutput(); //calling function


if(inches != 0 && inches < 9){
tone(buzzer,(9000/inches),100);
 }
}
```

Timer interrupt

- Useful in executing two or more functions at once
- 3 timer found in Arduino Uno:
  - Timer 0 and Timer 2 are 8 bit timers
  - Timer 1 is a 16 bit timer
- Default functions:
  - Timer 0:
    - Executes the delay(), delay(), millis(),micros() and other delay function
  - Timer 1:
    - Executes the servo functions
  - Timer 2:
    - Tone() function, for buzzer

Setting the desired timer frequency

1. Decide on the time and calculate its frequency
2. Use a pre-scalar, if needed, to adjust the Arduino system frequency
3. Calculate the compare match register to decide which timer to use
   - To use timer 0,2 value >256(2ˆ8)
   - To use timer 1 value >65536(2ˆ16)

Pre-Scalar

- Arduino System clock is 16MHz
- Pre-scalar are used adjust the Arduino system clock to useful frequency
- Pre-scalar setting: 1, 8,64, 256, and 1024
- **(timer speed (Hz)) = (Arduino clock speed (16MHz)) / Pre-scalar**

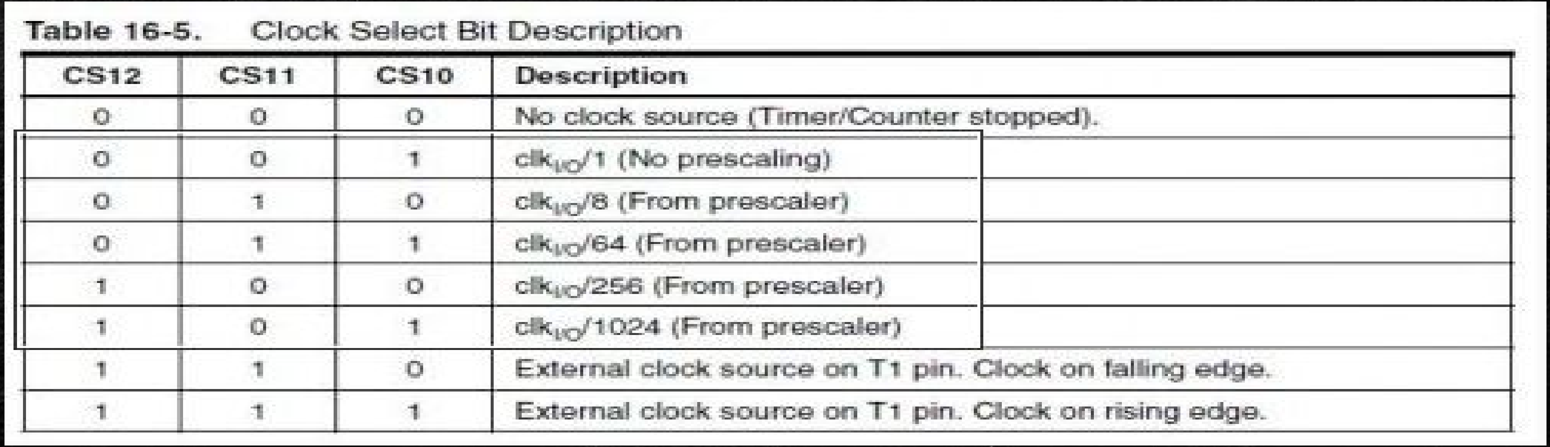
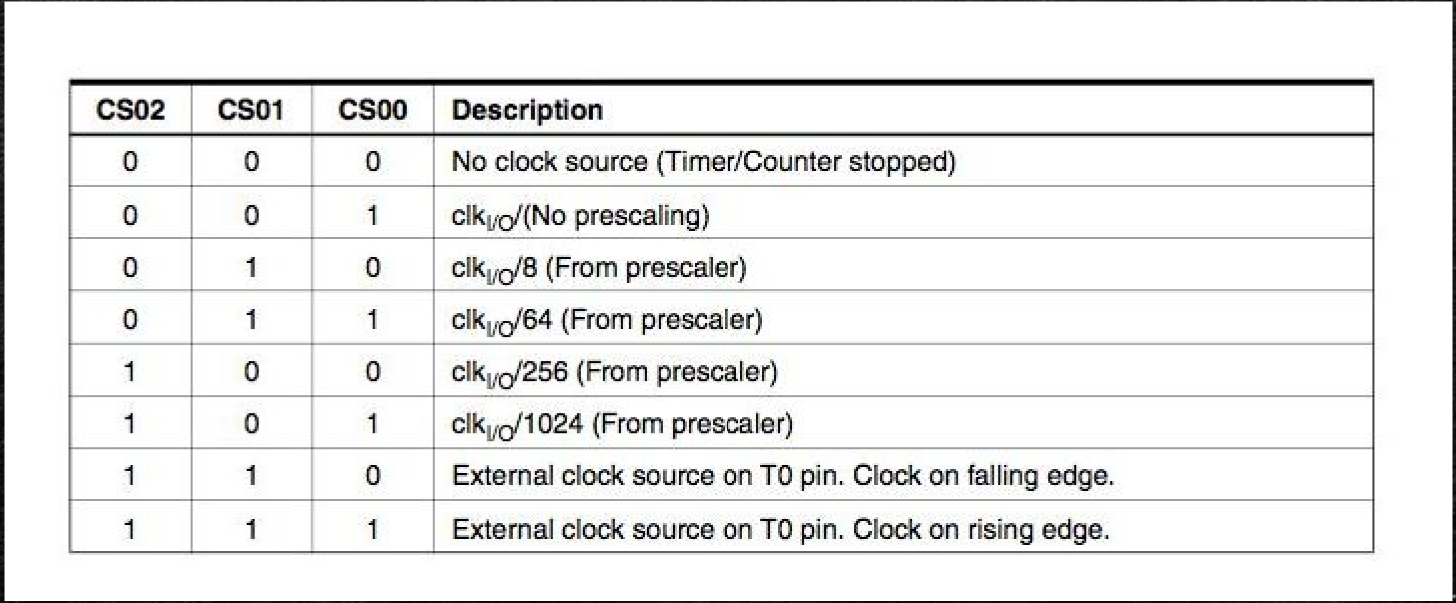


Table 16-5. Clock Select Bit Description

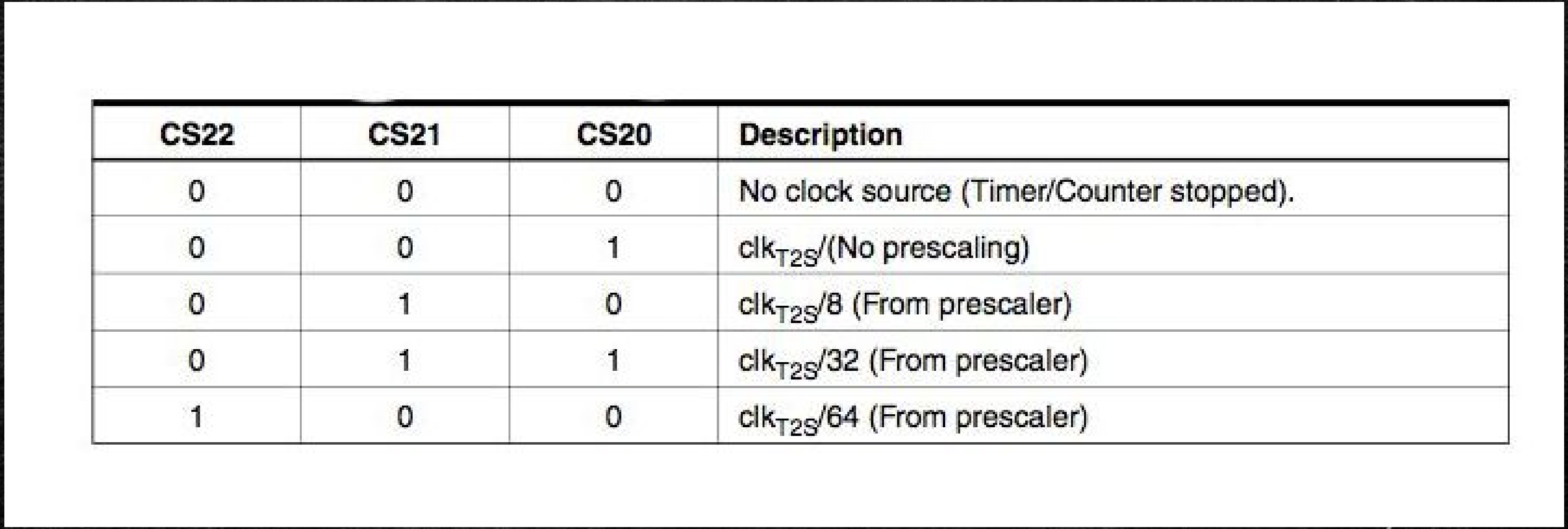
CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk$_{I/O}$/1 (No prescaling)
0	1	0	clk$_{I/O}$/8 (From prescaler)
0	1	1	clk$_{I/O}$/64 (From prescaler)
1	0	0	clk$_{I/O}$/256 (From prescaler)
1	0	1	clk$_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Timer 0



CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Timer 2



CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk$_{T2S}$/(No prescaling)
0	1	0	clk$_{T2S}$/8 (From prescaler)
0	1	1	clk$_{T2S}$/32 (From prescaler)
1	0	0	clk$_{T2S}$/64 (From prescaler)

Compare match register

- Compare match register = [16 MHz/ (pre-scalar * desired interrupt frequency)] – 1
  - the -1 is in there because the compare match register is zero indexed

If interrupt every 0.1s (f=10Hz):

compare match register = [16,000,000 / (pre-scalar * 10)] -1

with a pre-scalar of 1024 you get:

compare match register = [16,000,000 / (1024 *10)] -1

= 1561.5 => 1561

since 256 < 1561 < 65,536, you must use timer1 for this interrupt.

Program

```
// initialize timer1
  noInterrupts();           // disable all interrupts
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1  = 0;

  OCR1A = 1561;             // compare match register 16MHz/1024/10Hz
  TCCR1B |= (1 << WGM12);   // CTC mode
  TCCR1B |= (1 << CS12) | (1 << CS10);    // 1024 prescaler
  TIMSK1 |= (1 << OCIE1A);  // enable timer compare interrupt
  interrupts();             // enable all interrupts
```

Difficulties

- Incorporating the Ultrasound with the Wi-Fi program
  - Ultrasound function only occur once during the program
  - Delay causes the program to hag while web page replies back to the Arduino
  - Timer delay is use to run the ping function simultaneously with the main Arduino program

References

http://potat0-kun.deviantart.com/art/Mini-4-wheel-drive-Tamiya-Car-258355809

Kart picture – http://www.diyden.net/Top-view_1.jpg

- Ultrasound
  - http://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf
- Buzzer:
  - http://arduino.cc/documents/datasheets/PIEZO-PKM22EPPH4001-BO.pdf
- Timer Interrupt:
  - http://www.instructables.com/id/Arduino-Timer-Interrupts/?ALLSTEPS
  - http://letsmakerobots.com/node/28278