Hovercraft (Electrical Engineering)

Dr. John Helferty

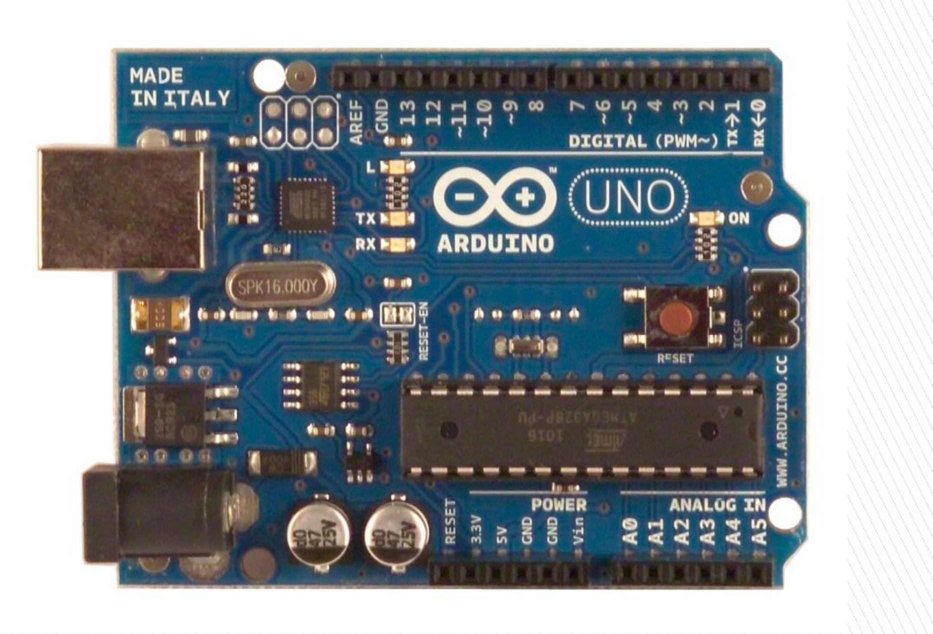
Electrical Components

- Arduino
- Wi-Fi Shield
- Relay
- H-Bridge
- Standard Servos
- DC Fans
- Impeller
- Battery
- PCB
- Headers

Arduino Microcontroller

- This is the main processor for your hovercraft

- All decisions for control will be programmed using this

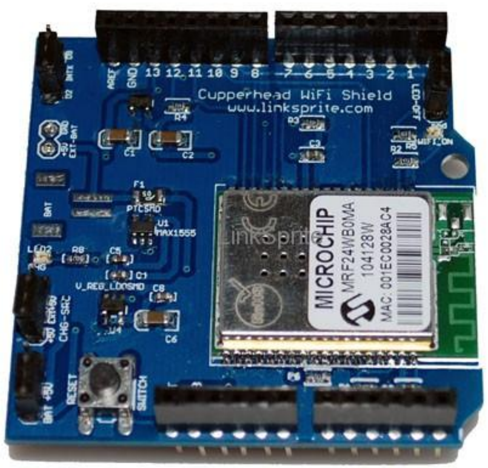


Wi-Fi Shield

- The cool thing about the Arduino is that people develop boards that plug into it

- These are called “Shields”

- The one we will be using allows the Arduino to create a Wi-Fi network



Relay

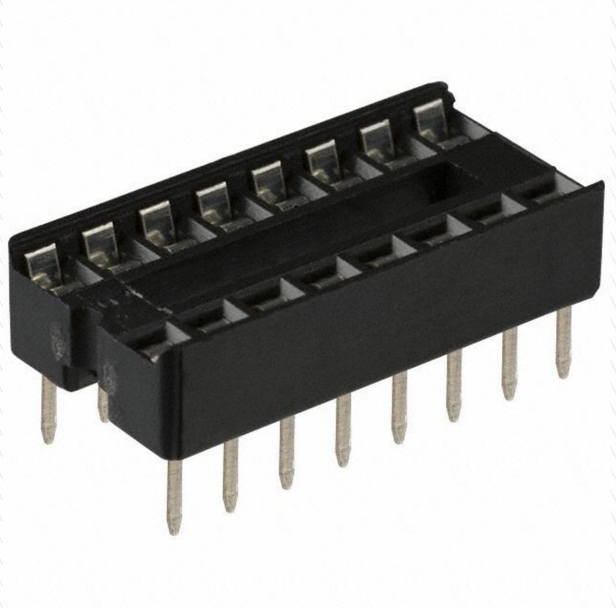
- A relay is a electronically controlled switch

- It can connect and disconnect a circuit when given a signal

- Makes an awesome *click* when turned on or off

H-Bridge

- This integrated circuit (IC) is responsible for driving the fans of the hovercraft
- The Arduino can only output 20 milliamps
- The fans require .53 amps
- This amplifies the current so the fans can be controlled by the Arduino
- DO NOT SOLDER HEADER INTO THE PCB. USE IC HOLDER ---------------→



Standard Servos

- These are not the same servos that we have used during class but work in much the same way
- A PWM signal is sent to the servo and the position will change accordingly
- This allows the rudders to be moved left and right to turn

DC Fans

- This will be the main force for movement forward for the hovercraft
- The fans turn on, the hovercraft goes forward

Impeller

- This provides the lift necessary for the hovercraft to hover

- Without it, it's just a craft

- Turning this on and off will be controlled by the relay

Battery

- This is a LiFePO412V 3.3 Amp hour battery

- It will provide power for the Arduino, Fans, Impeller and Servos

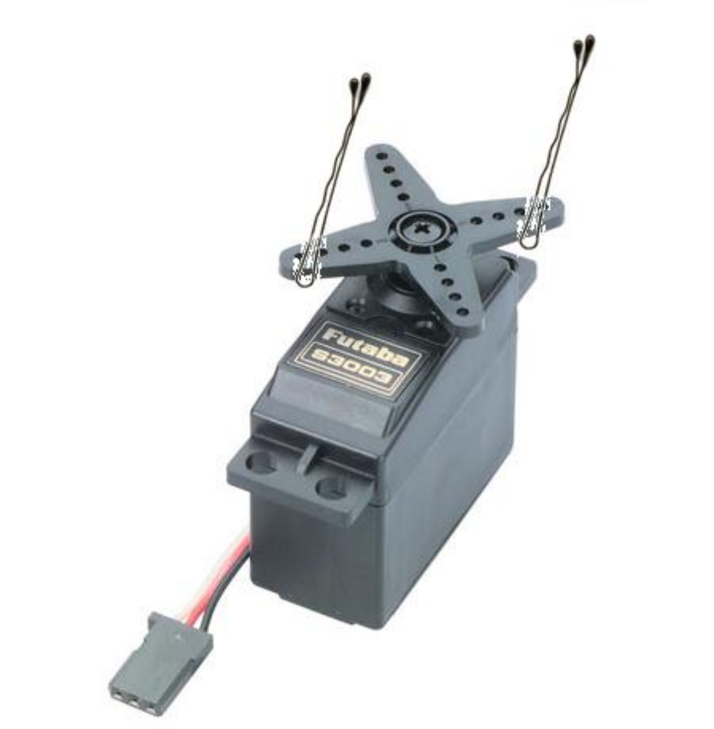
- 3.3 Amp hours means the battery can supply 3.3 Amps for 1 hour, 1.65 Amps for 2 hours and so on

Headers

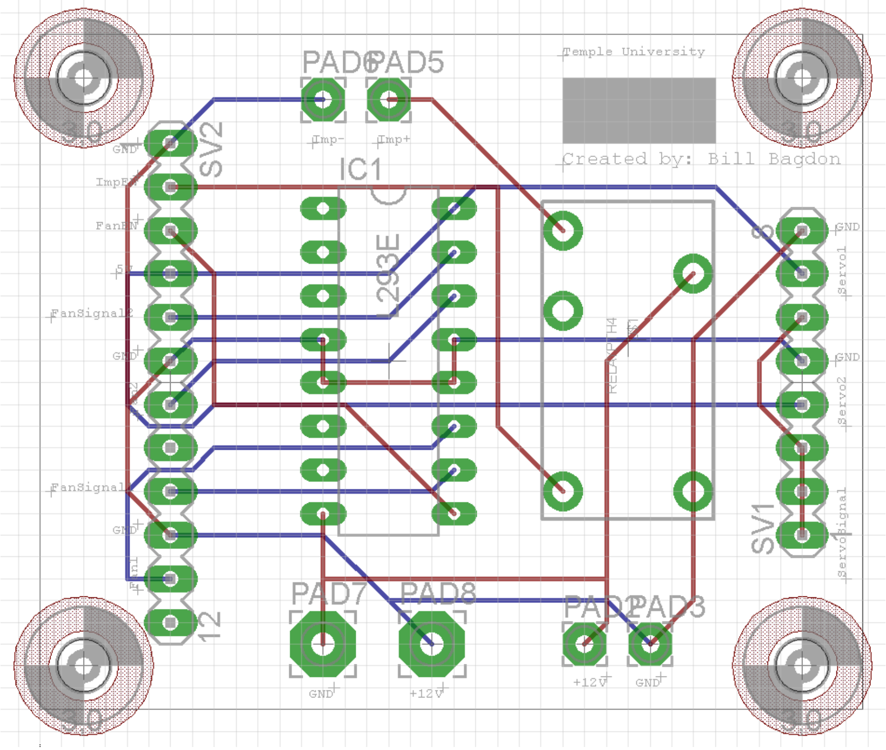
- These are used to make quick connections to solder points on the board
- This allows the use of jumper cables and easier changing of pins on the PCB and Arduino

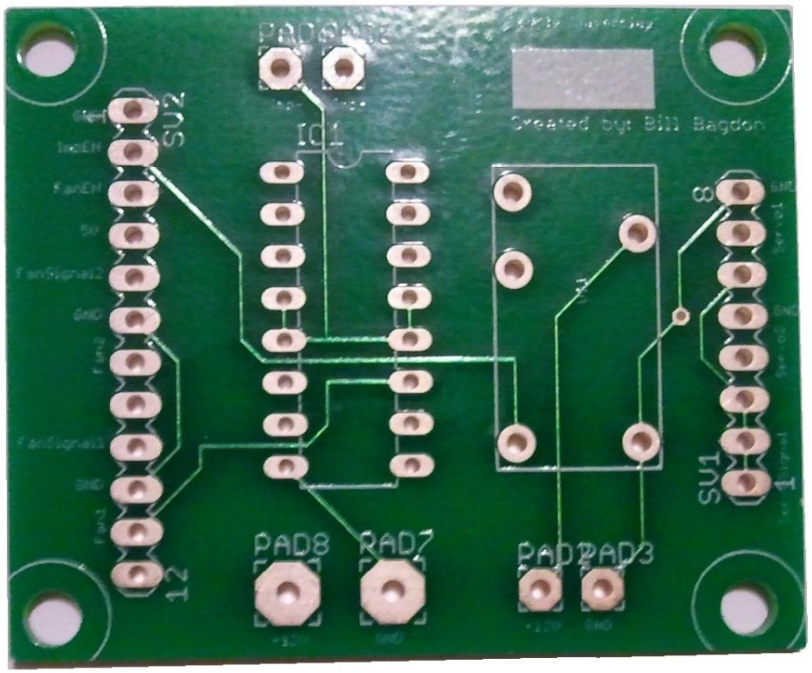
Rudders

- The bobby pins can be used to mount rudders onto the servos.
- You can slide the corrugated cardboard side onto the pins to support the rudders

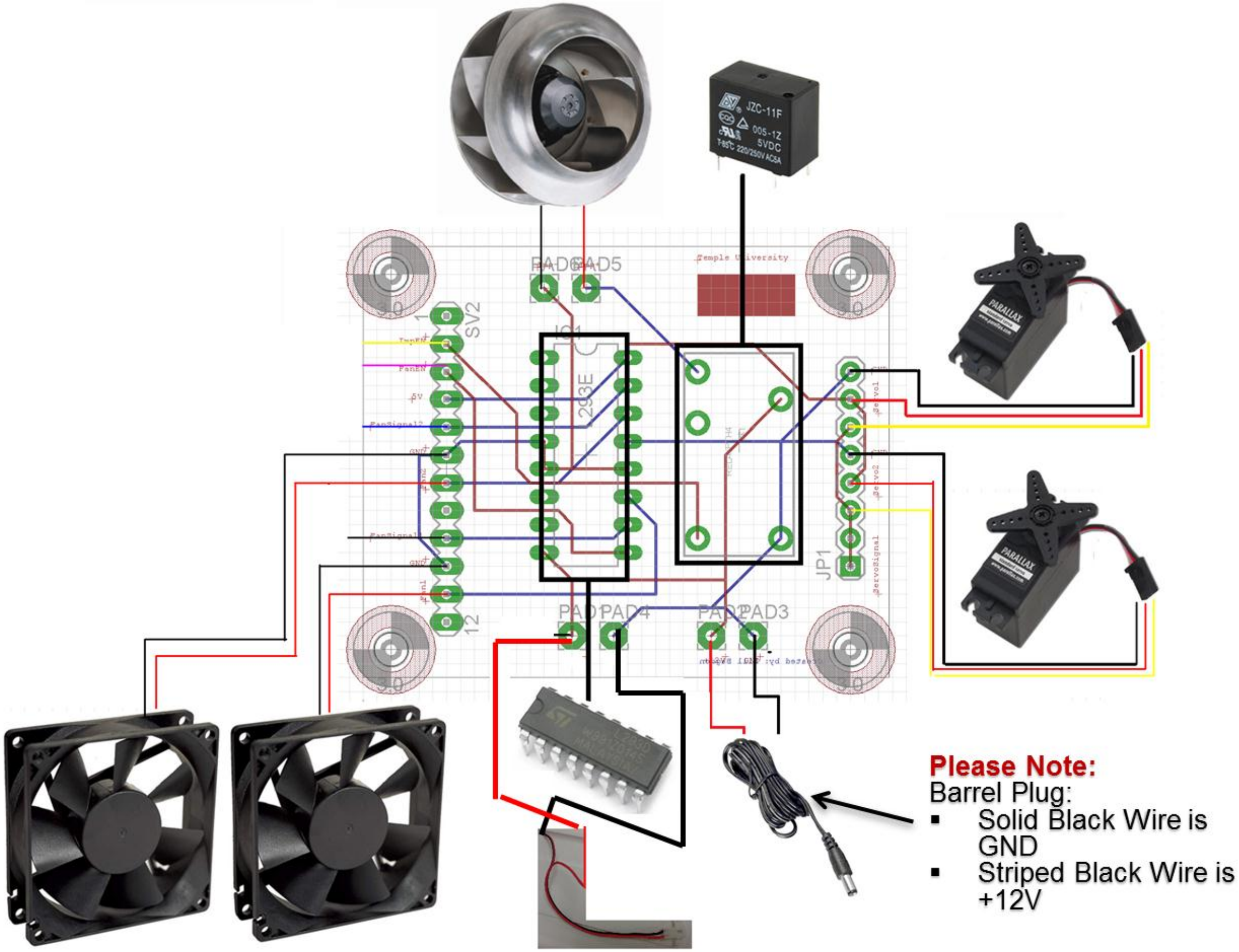


PCB


PAD6
PAD5
Temple University
Created by: Bill Bagdon
Imp-
Imp+
IC1
L293E
SV2
GND
ImpEN
FanEN
5V
FanSignal2
GND
Fan2
FanSignal1
GND
Fan1
12
RELAY PTH4
8
GND
Servo1
GND
Servo2
ServoSignal
SV1
1
PAD7
PAD8
PAD2
PAD3
GND
+12V
+12V
GND


PAD6
PAD5
Temple University
Created by: Bill Bagdon
IC1
SV2
GND
ImpEN
FanEN
5V
FanSignal2
GND
Fan2
FanSignal1
GND
Fan1
12
8
GND
Servo1
GND
Servo2
ServoSignal
SV1
1
PAD8
PAD7
PAD2
PAD3
+12V
GND
+12V
GND

Wiring the PCB

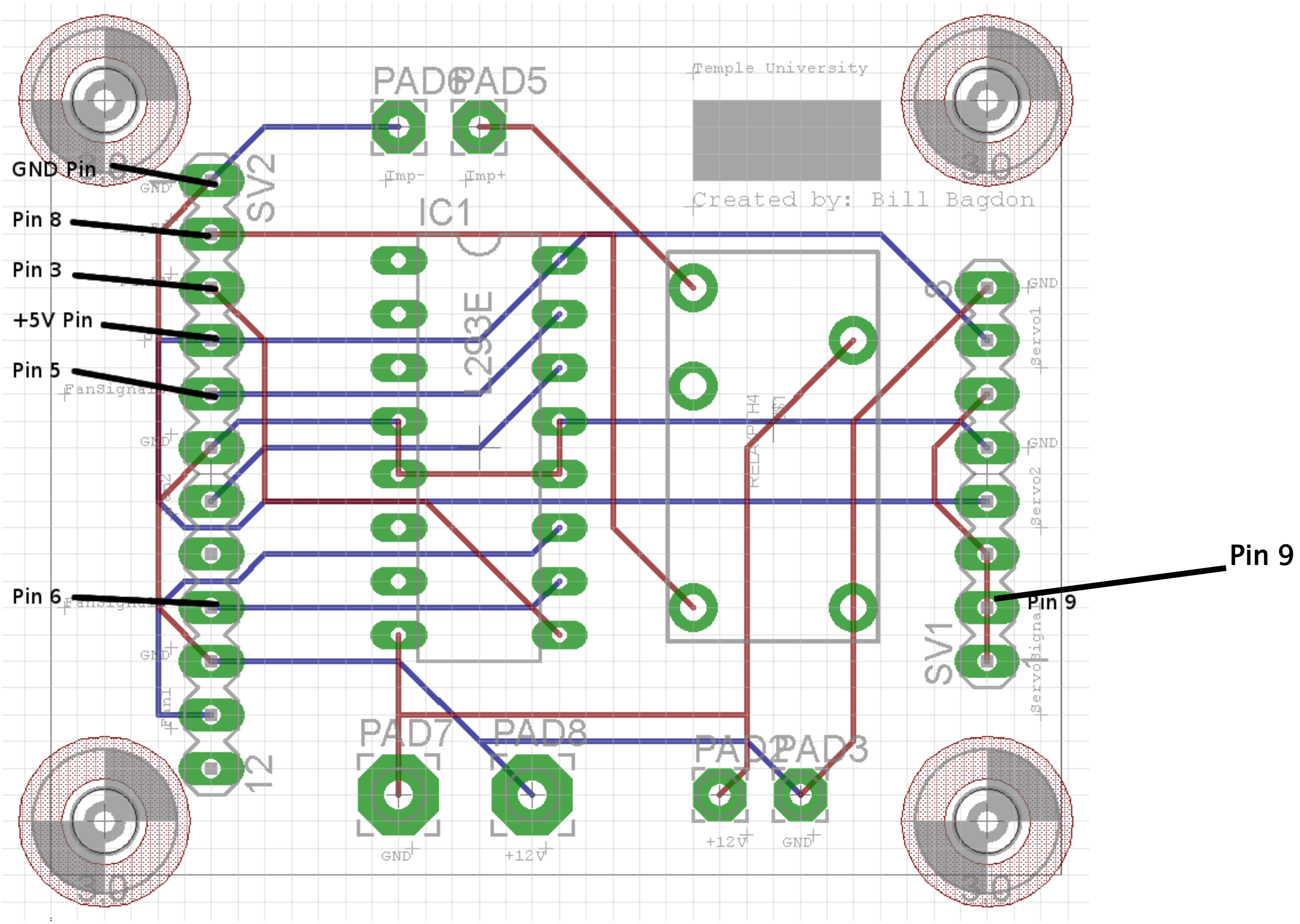


Please Note:

Barrel Plug:

- Solid Black Wire is GND
- Striped Black Wire is +12V

Arduino Pin Connections



Barrel Plug

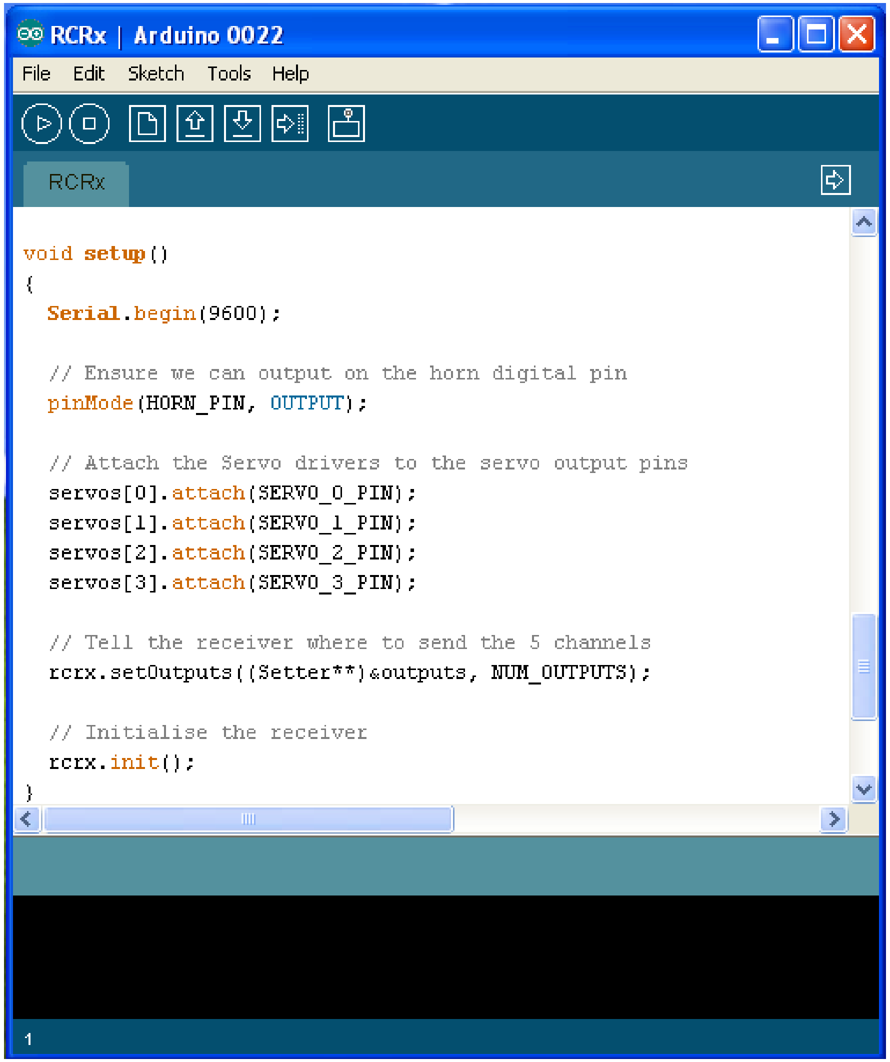
- Solid Black Wire is GND
- Striped Black Wire is +12V

Programming

- Libraries
  - AccelStepper
  - dataflash
  - RCKit
  - WiShield

Open and Upload:

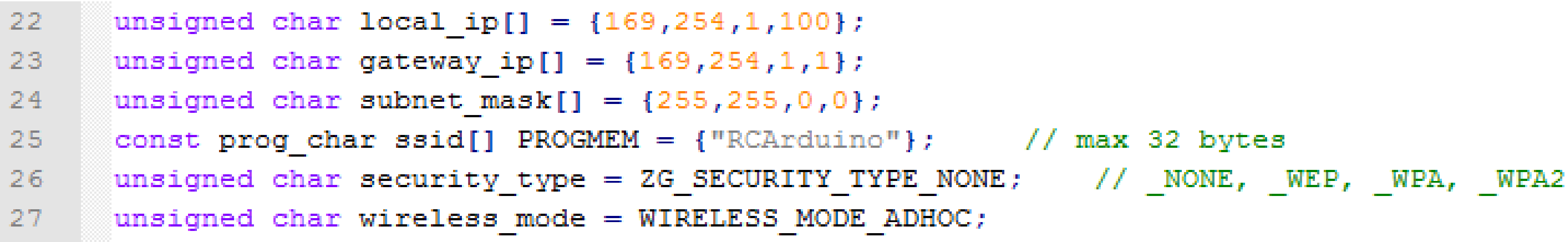
- remoteV5.ino



```
void setup()
{
  Serial.begin(9600);

  // Ensure we can output on the horn digital pin
  pinMode(HORN_PIN, OUTPUT);

  // Attach the Servo drivers to the servo output pins
  servos[0].attach(SERVO_0_PIN);
  servos[1].attach(SERVO_1_PIN);
  servos[2].attach(SERVO_2_PIN);
  servos[3].attach(SERVO_3_PIN);

  // Tell the receiver where to send the 5 channels
  rcrx.setOutputs((Setter**)&outputs, NUM_OUTPUTS);

  // Initialise the receiver
  rcrx.init();
}
```

Programming

- Changing the Wi-Fi Settings
- Go to Libraries\RCKit\RCRx.cpp
- Adjust the following code accordingly:

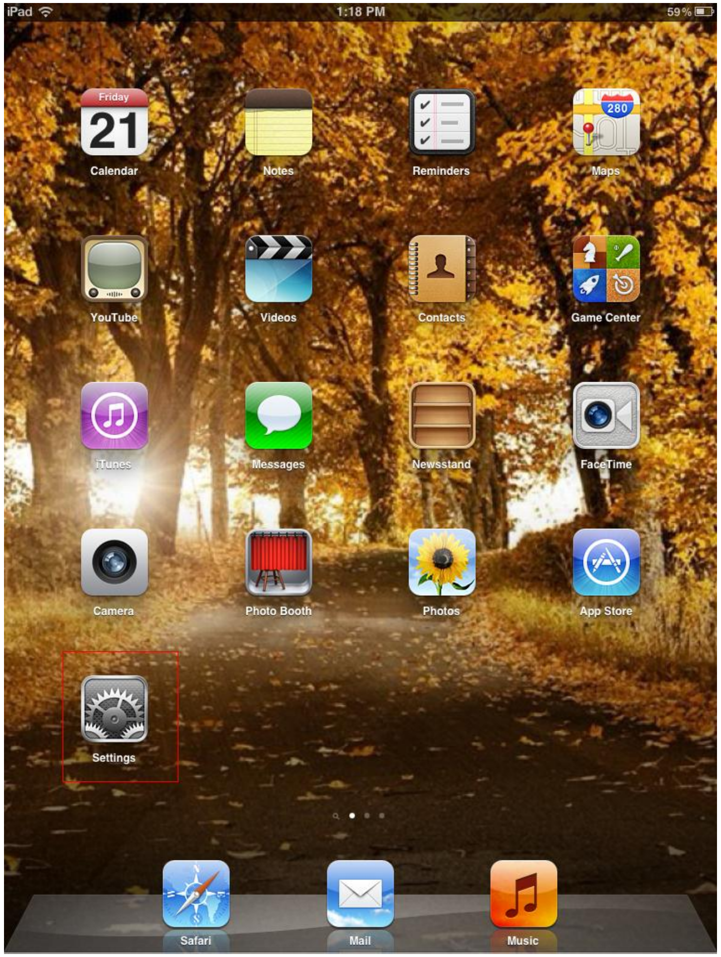


```
unsigned char local_ip[] = {169,254,1,100};
unsigned char gateway_ip[] = {169,254,1,1};
unsigned char subnet_mask[] = {255,255,0,0};
const prog_char ssid[] PROGMEM = {"RCArduino"};     // max 32 bytes
unsigned char security_type = ZG_SECURITY_TYPE_NONE;    // _NONE, _WEP, _WPA, _WPA2
unsigned char wireless_mode = WIRELESS_MODE_ADHOC;
```

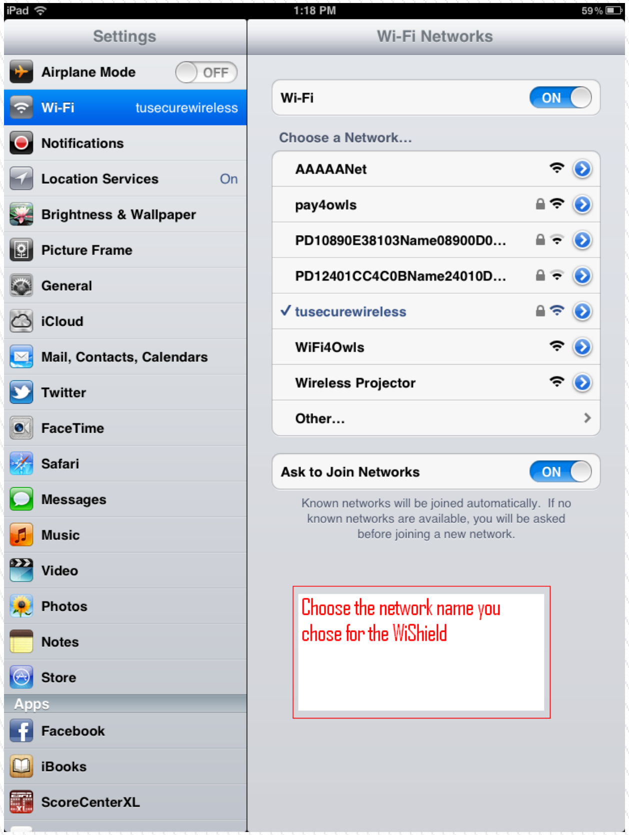
Using the iPad

- Finding the Wi-Fi signal
- Opening and running the app
- Controlling the hovercraft

Choose Settings



Your network name will appear within 30 seconds of powering the Arduino



Airplane Mode OFF

Wi-Fi tusecurewireless

Location Services On

Wi-Fi ON

✓ tusecurewireless

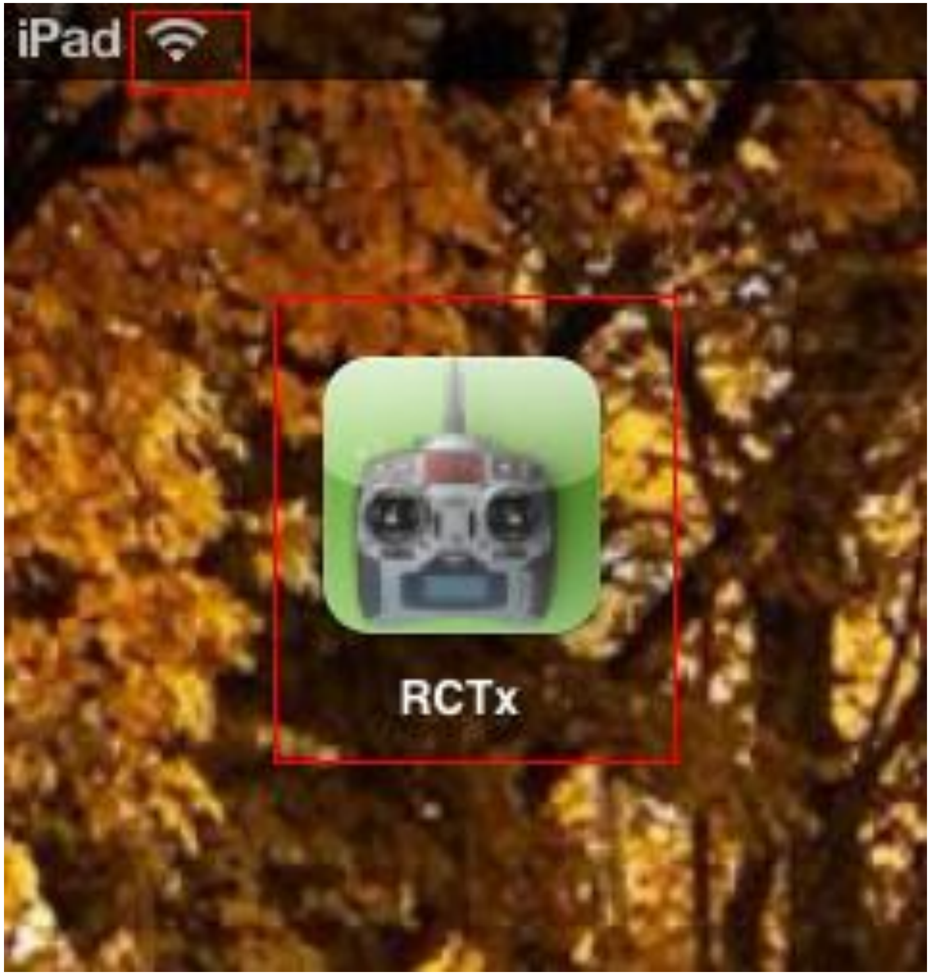
Ask to Join Networks ON

Known networks will be joined automatically. If no known networks are available, you will be asked before joining a new network.

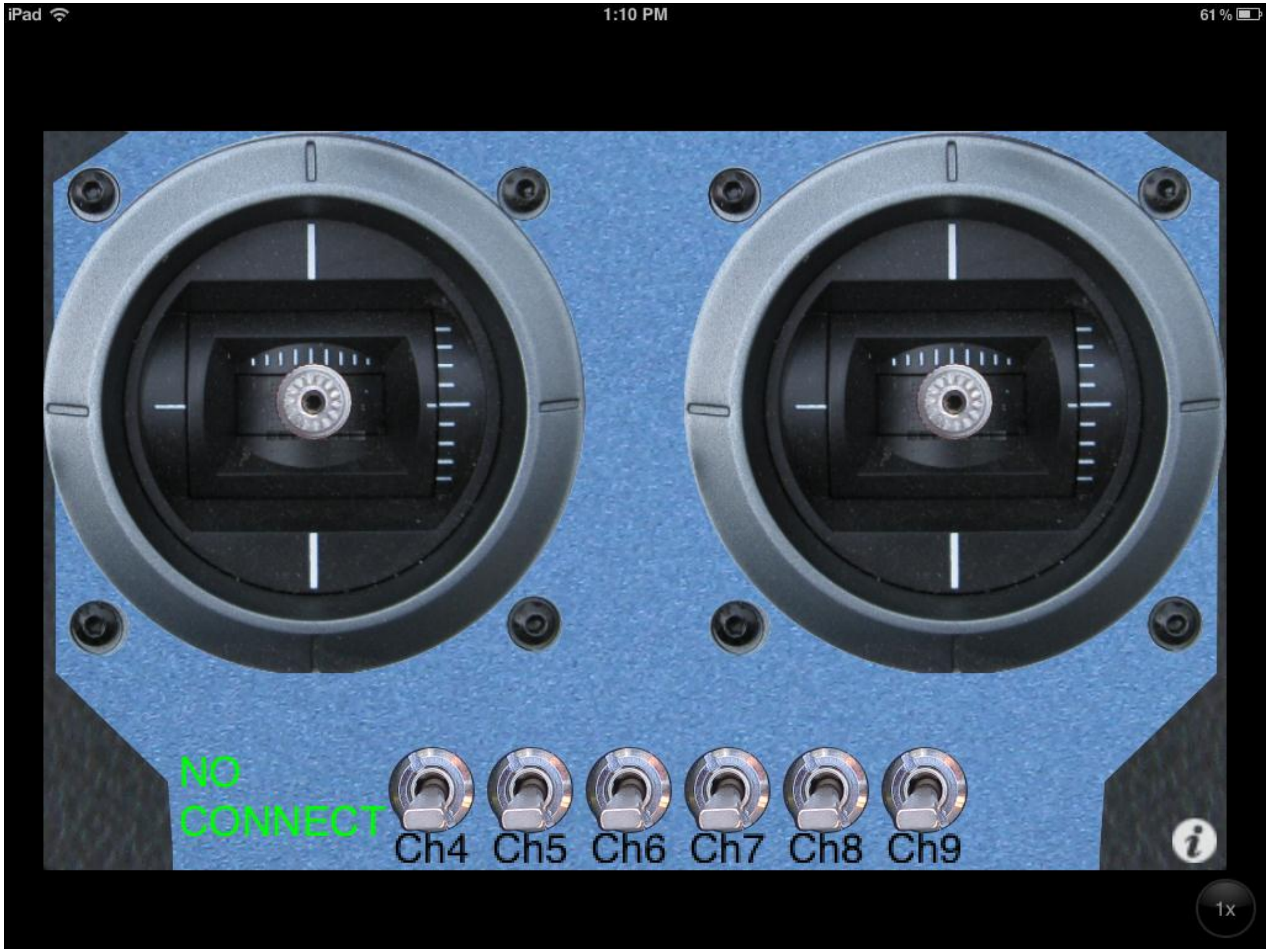
Choose the network name you chose for the WiShield

- install the RCTx app on your iPhone.
- Build and upload your RCRx program you your Arduino+WiShield hardware
- Power up the Arduino
- Turn on your iphone, go to Settings, Wi-Fi. Enable Wi-Fi
- After about 30 seconds, you should see the RCArduino network appear as an available network on the iPhone. Tap on it. Now tap on the RCArduino network details arrow to the right of the RCArduino line. Select 'Static'. Enter an IP Address of 169.254.1.1. Enter a Subnet Mask of 255.255.0.0
- After about 10 seconds, the iPhone should be successfully connected to the RCArduino network. You now have an ad-hoc connection to the Arduino. The Arduino will have the address 169.254.1.100 and the iPhone wil have address 169.254.1.1
- Start the RCTx app on the iPhone.
- After about 5 seconds, you should see the NO CONNECT in the bottom left corner change to show RSSI and the correct battery voltage (if the Arduino is so equipped).
- Move the josticks and buttons. This will send RCOIP commands to the Arduino. RCRx in the arduino will convert them to analog output signals to drive your hardware. Have fun.

Wait for the Wi-Fi signal icon to appear then open RCTx



Control the hovercraft


iPad
1:10 PM
61 %
NO CONNECT
Ch4
Ch5
Ch6
Ch7
Ch8
Ch9
i
1x