Contents

Integrated Design Project Electronics Introduction 2

Arduino Uno Wi-Fi REV2 and Adafruit Motor Shield REV2, 1 Max per Team 4

Arduino Description 4

Arduino when used in IDP 4

Adafruit Motor Drive Description 4

Understanding the power supplies 4

Running the Motors 5

Install Adafruit Motor Shield V2 library 5

Running the Example Code DC Motor 6

Hint/Tips 6

LED Module, Red, Green, White and Blue, 1 Max per team Red, Green, Blue and White 7

Description 7

Specifications 7

Pin Definition 7

Connection Diagram 7

Sample Code 8

Hint/Tips 8

Push Button, 2 Max per team 9

Description 9

Specification 9

Pin Definition 9

Connection Diagram 9

Sample Code 10

Hint/Tips 10

Crash Switch Module, 2 Max per team right and left 11

Description 11

Specification 11

Pin Definition 11

Sample Code 12

Hint/Tips 12

Line sensor Module, 4 Max per team 13

Description 13

Specification 13

Pin Definition 13

Sample Code 14

Hint/Tips 14
Ultrasonic Sensor Module, 1 Max per team 15
Introduction 15
Specification 15
Measuring Angle 15
Sample Code 16
Hint/Tips 16
Time of Flight Sensor Module, 1 Max per team 17
Description 17
Specification 17
Pin Definition 17
Software 17
Example Code 18
Hint/Tips 18
Standard Servo - 1 in kit a second can be supplied if needed 20
Description 20
Specification 20
Servo Size 21
Pin Definition 21
Sample Code 21
Putting it all together 22
The Prototyping Board 1 Max per team 22
Hint/Tips 24
Appendix 1 25
Prototyping Board layout 25

Integrated Design Project Electronics Introduction

It is highly recommended that as a team you work your way through the following examples describing the electronic modules supplied to you. The team can then utilize this information in the design and operation of the robot safely.

The electronics system is modular once you are familiar with the modules supplied it allows for the rapid production of your team robot.

The modules supplied are all tested before they are kitted to the team, they are robust reliable units which if treated correctly will last for years. If problems do occur the team will need to complete the return/exchange form in its entirety before a replacement is provided this can take up to 24 hours.

Note Nylon screws, nuts and washers must be used to mount sensor modules.

IDP Module Return/Exchange Request

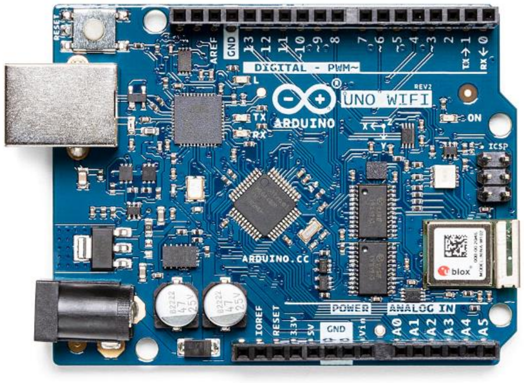
Arduino Uno Wi-Fi REV2 and Adafruit Motor Shield REV2, 1 Max per Team

Arduino Description

The Arduino UNO Wi-Fi Rev 2 features the secure **ATECC608** crypto chip accelerator, using the **ATmega4809 8-bit** microcontroller from Microchip. It also has an onboard IMU (Inertial Measurement Unit), **LSM6DS3TR** and features the **NINA-W102** Wi-Fi & Bluetooth® module from u-Blox.

For additional information please use the link below.
https://store.arduino.cc/products/arduino-uno-wifi-rev2

Arduino when used in IDP



D2, D13 i/o pins, **Max current 20mA Max Voltage 5V**

D0, D1 i/o pins, Avoid using pins shared with comms.

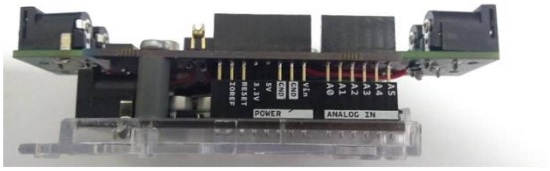
Arduino 5V output **Not Used**

GND reference common for all voltages

4 analog inputs, Max voltage 5V

2 analog pins, avoid using Motor Board comms.

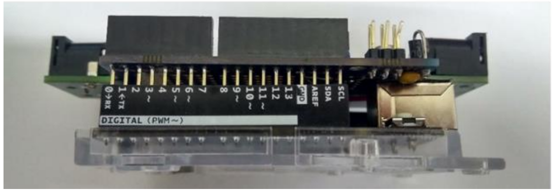
Adafruit Motor Drive Description



Board spacer, please do not remove/separate boards

3.3V, 5V pins cut intentionally, we do not use 3.3V and 5V comes from additional PSU on Adafruit boards

A4, A5 pins, remember avoid using these are used for comms.



D0, D1 pins, remember avoid using these are used for comms.

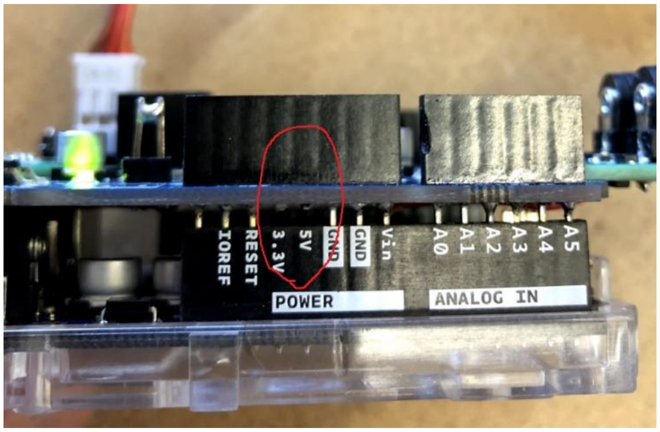
D2, D13 general purpose i/o

Servo outputs, connected to i/o pins 9, 10.

Understanding the power supplies

To allow fully autonomous operation, a 12V Li-Ion battery pack is supplied. The battery directly connects to the Arduino Vin socket. The stacking headers also pass the 12V directly to the Adafruit motor shield to provide a high current supply 12V to the motors. A DC-DC converter on the Motor shield also steps this supply down to 5V as a dedicated supply for the servos and your custom electronics.

The Arduino itself also contains 12V to 5V (and 3.3V) regulators. The pins for these have been removed from the stacking headers to prevent their use as the current output is limited from the Arduino and replaced by a dedicated 5v supply fitted to the Adafruit motor shield.

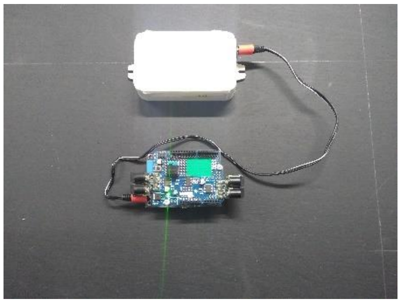


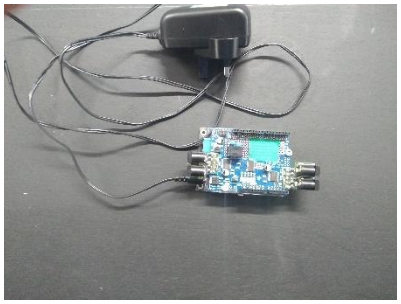
If the 12V on Vin is not present the Arduino has been modified to not power up this is to prevent damage primarily to your laptops also to the Arduino.

Note: The Orange Pip boards use a linear regulator to obtain 5V from the 12V input therefore has a greatly reduced current handling capability.

Running the Motors

Connect a 12v supply to the Arduino, either battery or mains adaptor,



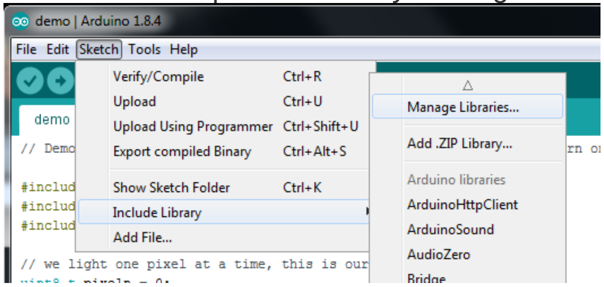


Install Adafruit Motor Shield V2 library

To use the shield on an Arduino, you'll need to install the Adafruit Motor Shield v2 library.

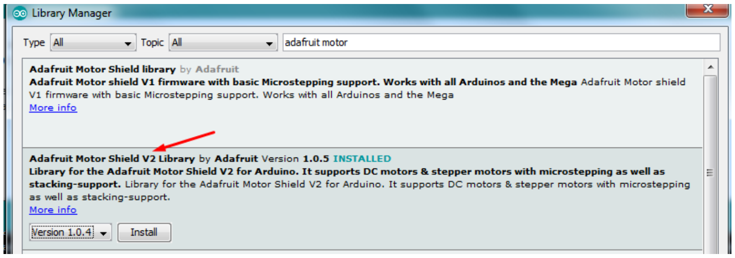
To begin controlling motors, you will need to [install the Adafruit_Motor_Shield_V2_Library library (code on our github repository)](). It is available from the Arduino library manager, so we recommend using that.

From the IDE open the library manager.



```
// Demo
#includ
#includ
#includ

// we light one pixel at a time, this is our
```

And type in **Adafruit motor** to locate the library. Click **Install**



Adafruit Motor Shield library by Adafruit
Adafruit Motor shield V1 firmware with basic Microstepping support. Works with all Arduinos and the Mega Adafruit Motor shield V1 firmware with basic Microstepping support. Works with all Arduinos and the Mega
More info

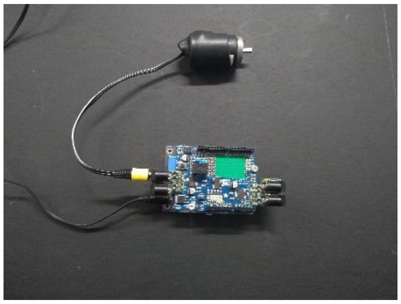
Adafruit Motor Shield V2 Library by **Adafruit** Version **1.0.5** INSTALLED
Library for the Adafruit Motor Shield V2 for Arduino. It supports DC motors & stepper motors with microstepping as well as stacking-support. Library for the Adafruit Motor Shield V2 for Arduino. It supports DC motors & stepper motors with microstepping as well as stacking-support.
More info

Running the Example Code DC Motor

The library comes with a few examples to get you started up fast.

First, restart the IDE to make sure the new library is loaded.

Connect a DC motor to motor port 1 Plug/Socket.



Once you have verified the motor is connected properly **and** you have the power LED lit up brightly, we can upload our code.

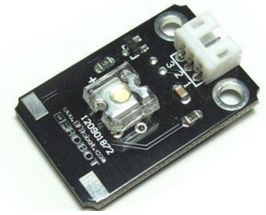
In the IDE, load **File->Examples->Adafruit_MotorShield->DCMotorTest**

You should see and hear the DC motor turn on and move back and forth, attaching a slip of paper or tape as a 'flag' can help you visualize the movement.

Hint/Tips

- If testing on the bench try to use your 12v mains adaptor, save your battery for table testing.
- Removing the power from the Arduino when plugging items in or out will greatly reduce the risk of breaking it.
- Remember if the team does break the Arduino it can take up to 24 hours to replace it please take care.

LED Module, Red, Green, White and Blue, 1 Max per team Red, Green, Blue and White



Description

Piranha LED light module is a special light module designed for easy plug and play Arduino light projects. When the LED light module is connected to Arduino, the digital pin (LOW on, HIGH off) is used to control it. The brightness of LED can be controller via PWM output.

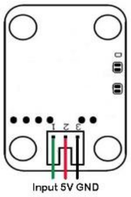
Note Nylon screws, nuts and washers must be used to mount this module.

Specifications

- Voltage: +3.3-5V
- Weight: 5g

Pin Definition

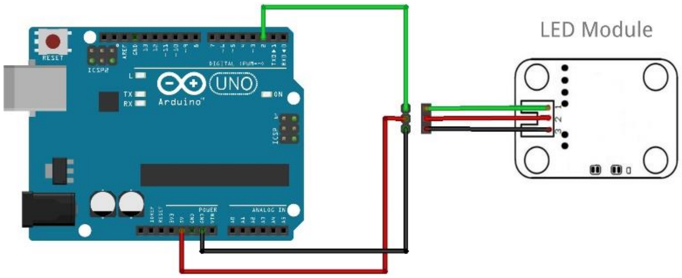
- 1, Input, Green
- 2, Power, Red
- 3, GND, Black



Connection Diagram

Check all voltages before connecting to the sensor module using your team bench multimeter.

Use the DuPont wire jumpers to connect the sensor module to the Arduino/Adafruit module



Sample Code

```
/*
IDP
David Paterson
LED Module Test Code V1
*/

int led = 2;

void setup()
{
  pinMode(led, OUTPUT);     //Set Pin3 as output
}
void loop()
{
        digitalWrite(led, HIGH);   //Turn off led
        delay(2000);
        digitalWrite(led, LOW);    //Turn on led
        delay(2000);
}
```

Hint/Tips

- Always check connections and supplies before connecting anything
- Your bench multimeter is the only meter you really need so get to know it.

Push Button, 2 Max per team

Description

This large button gives your Arduino project the first touch of the physical world. Simply plug into the IO expansion board for quick and easy plug and play.

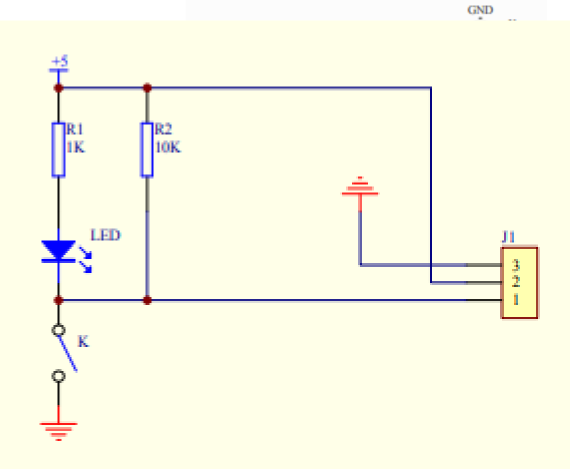
Note Nylon screws, nuts and washers must be used to mount this module.

Specification

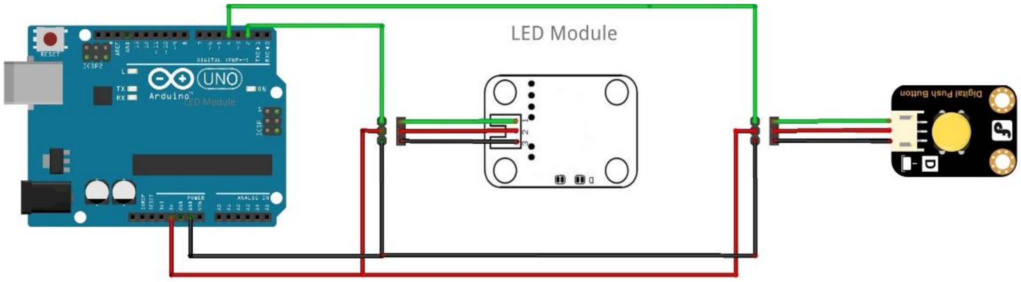
- Supply Voltage: 3.3V to 5V
- Indicator LED on board
- Easy 'plug and play'
- Large button keypad and high-quality first-class hat
- Interface: Digital
- Size:22x30mm

Pin Definition

- Pin 1, Digital output, Green
- Pin 2, VCC, Red
- Pin 3, GND, Black



Connection Diagram



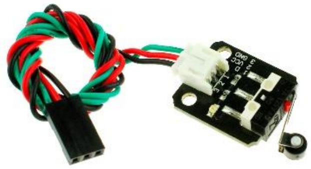
Sample Code

```
/*
  IDP
  David Paterson
  Push Button Module Example V1
  When you push the digital button the Led 2 will turn off otherwise the LED turns on.
*/
int ledPin = 2;                 // choose the pin for the LED
int inputPin = 3;               // Connect sensor to input pin 3


void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare pushbutton as input
}

void loop(){
  int val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    digitalWrite(ledPin, LOW);  // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

Hint/Tips

- Do you notice any “switch bounce” if so, how would you go about dealing with this?

Crash Switch Module, 2 Max per team right and left (Not always supplied)



Description

A miniature snap-action switch, also trademarked and frequently known as a micro switch, is an electric switch that is actuated by very little physical force. Micro switches are very widely used; among their applications are appliances, machinery, industrial controls, vehicles, and many other places for control of electrical circuits. They are usually rated to carry current in control circuits only, although some switches can be directly used to control small motors, solenoids, lamps, or other devices.

It integrates the pull-up resistor and the status indicator LED onboard for testing. The miniature snap-action micro switch with roller lever make it suitable for more different environment application.

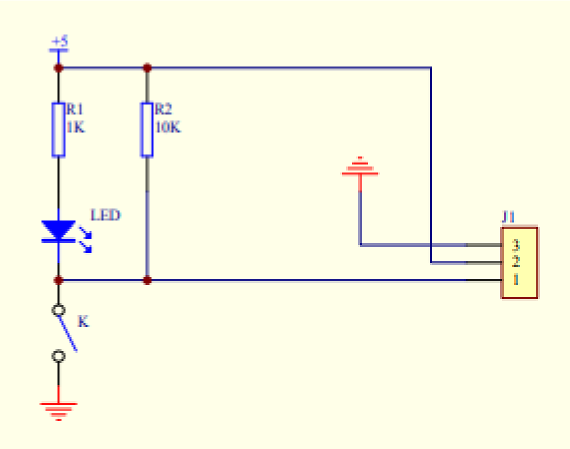
Note Nylon screws, nuts and washers must be used to mount this module.

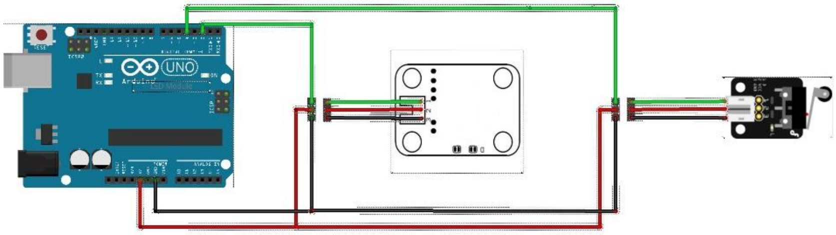
Specification

- Working Voltage: 5v
- Onboard status indicator LED
- Directly connected to the IO Expansion shield For Arduino
- M3 mounting hole x2
- Size: 30x20x8mm

Pin Definition

- Pin 1 Green Wire Digital output
- Pin 2 Red Wire 5V
- Pin 3 Black Wire GND





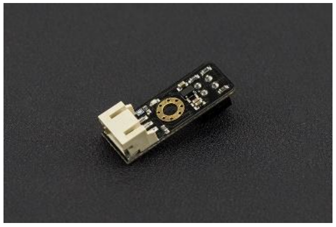
Sample Code

```
/*
IDP
David Paterson
Crash Switch Module Example Code V1
When you push the lever switch the Led 3 on the board will turn on
*/

int ledPin = 2;                  // choose the pin for the LED
int crashswitchPin = 3;                // Connect sensor to input pin 3


void setup() {
  Serial.begin(9600);           // Init the serial port

  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(crashswitchPin, INPUT);     // declare Micro switch as input
}

void loop(){
  int val = digitalRead(crashswitchPin);  // read input value
  if (val == HIGH) {                // check if the input is HIGH
    digitalWrite(ledPin, LOW);      // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH);     // turn LED ON
    Serial.println("Switch Pressed!");
  }
  delay(50);
}
```

Hint/Tips

-

Line sensor Module, 4 Max per team



Description

The line tracking (following) sensor will guide your robot by telling white from black quickly and accurately. It is commonly used in robot projects, and probably the best line following sensor on the market.

Line following is the most basic function of smart mobile robot. It is one of the easiest ways for a robot to navigate successfully and accurately. It will guide your robot by telling white from black quickly and accurately, via TTL signal.

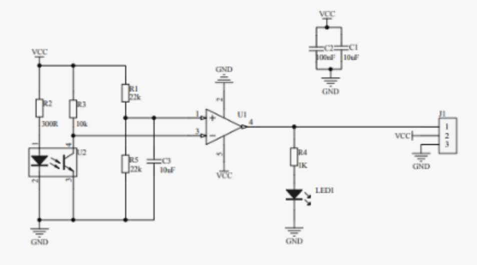
Note Nylon screws, nuts and washers must be used to mount this module.

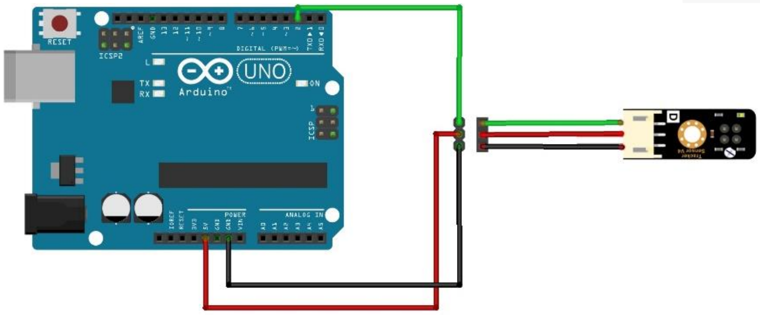
Specification

- Power supply: 3.3~5V
- Detecting Range: 1~2cm
- Operating current: <10mA
- Operating temperature range: 0°C ~ + 50°C
- Output interface: 3-wire interface (1 - signal, 2 - power, 3 - power supply negative)
- Output: TTL(Black for LOW output, White for HIGH output)
- Module Size: 10mm×28mm (1.1x 0.4 in)
- Module Weight: About 10g
- Version Update The best distance between objects such as the ground and the sensor is 1-2 cm. V5 removes the variable resistor and is therefore easier to use now.

Pin Definition

- 1, Output, Green
- 2, Power, Red
- 3, GND, Black





Sample Code

```
/*
IDP
David Paterson
Line Sensor Module Example Code V1
Move the line sensor across the black and white line, monitor on serial
*/

int leftlinesensorPin = 2;
int rightlinesensorPin = 3;               // Connect sensor to input pin 3


void setup() {
  Serial.begin(9600);           // Init the serial port

  pinMode(leftlinesensorPin, INPUT);      // declare LED as output
  pinMode(rightlinesensorPin, INPUT);     // declare Micro switch as input
}

void loop(){
  int valLeft = digitalRead(leftlinesensorPin);  // read left input value
    Serial.print(valLeft);
  int valRight = digitalRead(rightlinesensorPin);  // read right input value
   Serial.println(valRight);
  delay(100);
}
```

Hint/Tips

- Investigate what is the optimum height for the sensor? Use this information when designing the team chassis.

Ultrasonic Sensor Module, 1 Max per team

Introduction

This is an ultrasonic sensor with an open dual-probe, employing standard interface of Gravity PH2.0-3P vertical patch socket. The module is compatible with controllers with 3.3V or 5V logical level, such as Arduino and Raspberry Pi.

The sensor comes with built-in temperature compensation, which can greatly reduce measurement error caused by over high/low temperature. It adopts analog voltage output, and provides accurate distance measurement within 2-500cm with 1cm resolution and ±1% accuracy. The design of dual probe effectively decreases the dead zone. Users can check the measurement process conveniently by the onboard indicator.

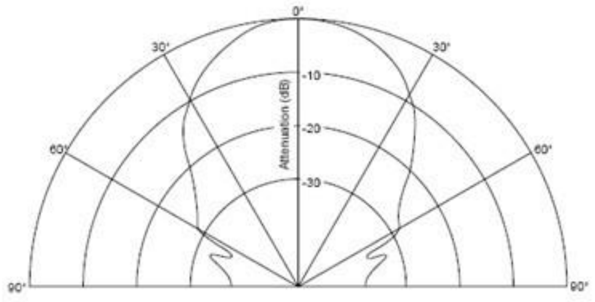
This small ranging module can be used in an extensive range of applications, for instance, robots with obstacle avoidance function, backing car annunciator, doorbell etc.

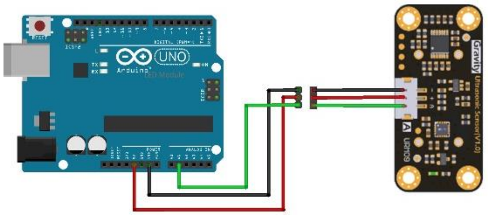
Note Nylon screws, nuts and washers must be used to mount this module.

Specification

- Power Supply: 3.3~5.5V DC
- Operating Current: 20mA
- Operating Temperature: -10℃～＋70℃
- Measurement Range: 2cm～500cm
- Resolution: 1cm
- Accuracy: 1%
- Acoustic Frequency: 38~42KHz
- Frequency: 30Hz Max
- Dimension: 47mm × 22 mm/1.85×0.87”
- Distance Formula: Distance=Vout(mV)×520/Vin(mV)

Measuring Angle





Sample Code

```
/*
IDP
David Paterson
URM09 Ultrasonic Sensor test
*/


#define  MAX_RANG      (520)//the max measurement value of the module is 520cm(a little bit longer than  effective max range)
#define  ADC_SOLUTION      (1023.0)//ADC accuracy of Arduino UNO is 10bit

int sensityPin = A0;    // select the input pin
void setup() {
  // Serial init
  Serial.begin(9600);
}
float dist_t, sensity_t;
void loop() {
  // read the value from the sensor:
 sensity_t = analogRead(sensityPin);
  // turn the ledPin on

 dist_t = sensity_t * MAX_RANG  / ADC_SOLUTION;//
 Serial.print(dist_t,0);
 Serial.println("cm");

 delay(500);

}
```

Hint/Tips

- It would be interesting to understand if different materials give different readings from a set distance..

Time of Flight Sensor Module, 1 Max per team

Description

The VL53L0X range finder is a high-precision distance finder that based on new Time-of-Flight (ToF) principle. VL53L0X provides accurate distance measurement whatever the target reflectance unlike traditional technology. It can measure absolute distances up to 2m.

DFRobot has introduced VL53L0X into Gravity Series. It provides Gravity-I2C interface, plug and play, supports 3.3V~5V power supply, and is compatible with more microcontrollers, and adapt to more application scenarios than others.

The VL53L0X integrates a leading-edge SPAD array (Single Photon Avalanche Diodes) and embeds ST's second generation FlightSenseTM patented technology. Its accuracy is ±3%, response time is less than 30ms, power consumption is only 20mW in normal operation mode, stand-by power consumption is 5uA.

The VL53L0X's 940nm VCSEL emitter (vertical cavity surface emitting laser) is totally invisible to the human eye, coupled with internal physical infrared filters, it enables longer ranging distance, higher immunity to ambient light and better robustness to cover-glass optical crosstalk.

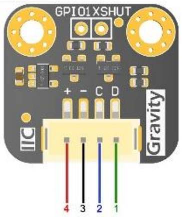
Note Nylon screws, nuts and washers must be used to mount this module.

Specification

- Power supply: 3.3-5V DC
- Operating Voltage: 2.8V
- Infrared emitter: 940nm
- Range: 30-2000mm
- FOV: 25°
- Ranging Accuracy: ±3%
- Sampling Time: <= 30ms
- Operating Temperature: -20-70 °C
- Interface Type: Gravity-I2C
- Product Size: 0.79*0.87 in

Pin Definition

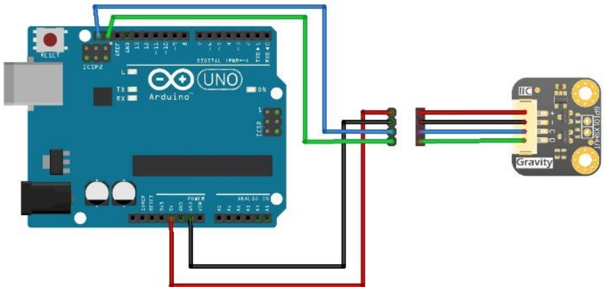
- Pin 1 Green Wire Data (SDL)
- Pin 2 Blue Wire CLK (SCL)
- Pin 3 Black Wire GND
- Pin 4 Red Wire 5V



Software

Like the Motor Sheild you will need to download the library to your IDE GitHub - DFRobot/DFRobot_VL53L0X

Once you have done this you will find the example code below.



Example Code

```
#include "Arduino.h"
#include "Wire.h"
#include "DFRobot_VL53L0X.h"

DFRobot_VL53L0X sensor;

void setup() {
  //initialize serial communication at 9600 bits per second:
  Serial.begin(115200);
  //join i2c bus (address optional for master)
  Wire.begin();
  //Set I2C sub-device address
  sensor.begin(0x50);
  //Set to Back-to-back mode and high precision mode
  sensor.setMode(sensor.eContinuous,sensor.eHigh);
  //Laser rangefinder begins to work
  sensor.start();
}

void loop()
{
  //Get the distance
  Serial.print("Distance: ");Serial.println(sensor.getDistance());
  //The delay is added to demonstrate the effect, and if you do not add the delay,
  //it will not affect the measurement accuracy
  delay(500);
}
```

Hint/Tips

- Like the Ultrasonic sensor it would be interesting to look at how different materials effect the distance reading from a fixed position.

Digital Magnetic Sensor 1 max per team

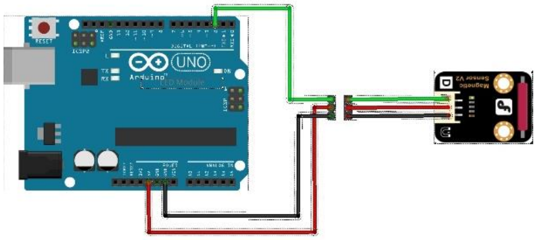
Description

This magnetic sensor knows whether there is a magnetic object nearby or not. And it correctly tells you through digital output.

Note Nylon screws, nuts and washers must be used to mount this module.

Specification

- Supply Voltage: 3.3V to 5V
- Indicator LED on board
- Interface: Digital
- Size: 22x30mm



Sample Code

```
int ledPin = 13;                // choose the pin for the LED
int inputPin = 2;               // choose the input pin
int val = 0;                    // variable for reading the pin status
void setup() {
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare pushbutton as input
}
void loop(){
  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    digitalWrite(ledPin, LOW);  // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

Hint/Tips

- Carry out investigation to identify the orientation for optimum magnetic field detection and distance

Standard Servo - 1 in kit a second can be supplied if needed

Description

DSS-M15S is a heavy-duty metal gear standard servo with 270°wide angle, high torque power, improved stability and durability.

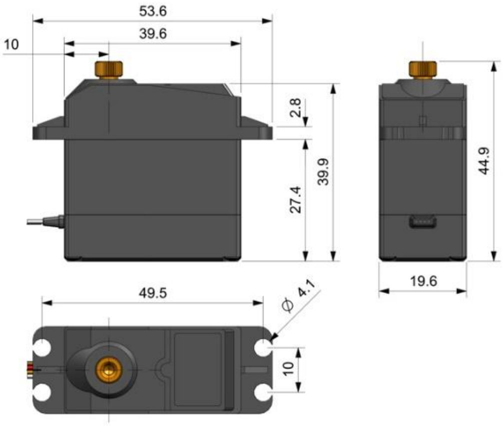
Able to take in 6V and deliver a strong torque power of over 15Kg, this tough 270°servo will never let you down when you need it. Tested with high load capacity for 12 hours long, this DSS-M15S servo demonstrates a maximum torque of 18Kg without much vibration or heat.

In addition, we enclose carefully selected steering parts, including an aluminum alloy wheel and a long bracket, to enhance servo performance.

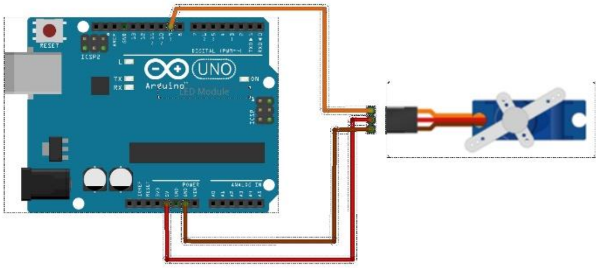
Specification

- Voltage: 4.8-6.8 Volts
- Torque: 15Kg*cm/17Kg*cm (5.0V/6.8V)
- Speed: 0.16 sec/60°/0.14 sec/60° (5.0V/6.8V)
- Free Run Current: 80mA/100mA (5.0V/6.8V)
- Stall Current: 1.8A/2A (5.0V/6.8V)
- Standby Current: 4mA/5mA (5.0V/6.8V)
- Rotation: 270°
- Wheel Gear: Metal
- Gear Ratio: 275:1
- Dimension: 40*20*40.5mm
- Signal: RC PWM
- Pulse Range: 500-2500 us
- Frequency: 50-330Hz（Arduion-compatible

Servo Size



Pin Definition



- 1, Signal, Orange
- 2, Power, Red
- 3, GND, Brown

Sample Code

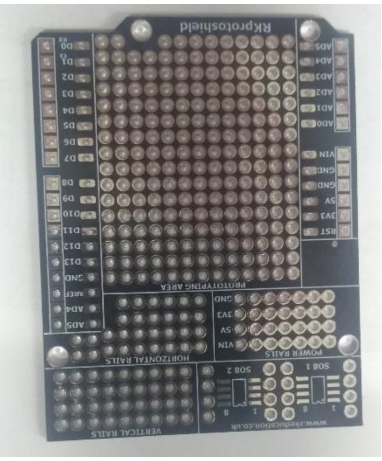
```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {

  myservo.attach(9);  // attaches the servo on pin 9 to the servo object

}

void loop() {

  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees

    // in steps of 1 degree

    myservo.write(pos);              // tell servo to go to position in variable 'pos'

    delay(15);                       // waits 15 ms for the servo to reach the position

  }

  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees

    myservo.write(pos);              // tell servo to go to position in variable 'pos'

    delay(15);                       // waits 15 ms for the servo to reach the position

  }

}
```

Putting it all together

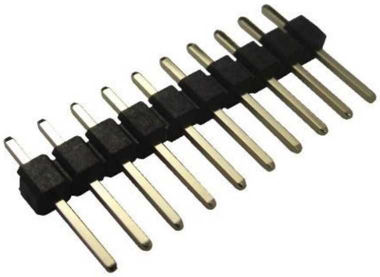
The team should now have a good understanding of the modules and how they can be implemented within the robot design.

There will obviously be a requirement to use more than one module so its time to get the soldering iron going putting together a interface for multiple modules.

The Prototyping Board 1 Max per team



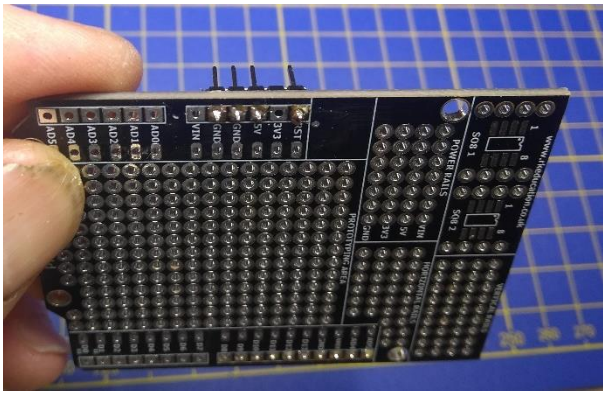
A selection of pin headers will also be supplied.



As a team you will need to agree upon a design for your proto board this will utilize pin headers to firstly make a connection between your Arduino/Adafruit Module then arrange pin header on the top of the proto board to plug in the modules.

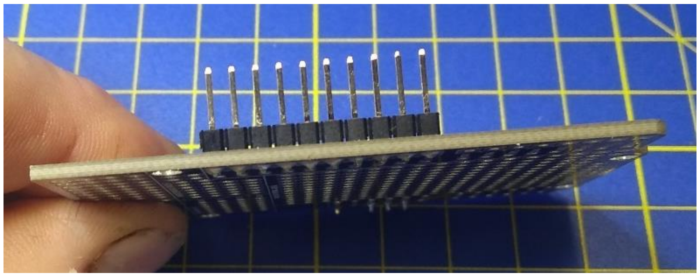
You will find a graphic of the proto board in appendix 1 which can be printed out and used to design they layout of the teams protoboard. Avoid using the greyed connections as these may cause the team problems.

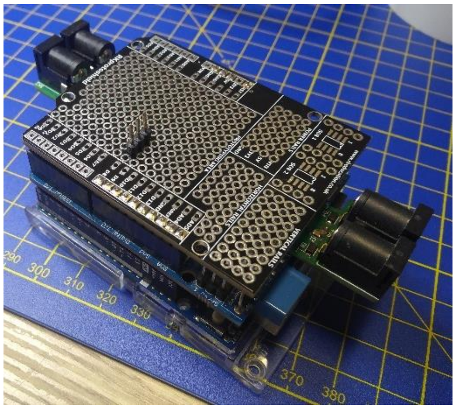
You will need to link a power supply and I/O to the pin header suitable for the module used.



Pin headers installed, not only 5V, GND and Reset used on this side of the proto board. Do not forget to have the board the correct way around.

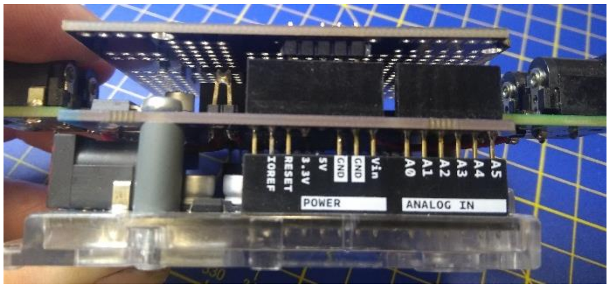
On the I/O side of the board connections made to D8-SCL

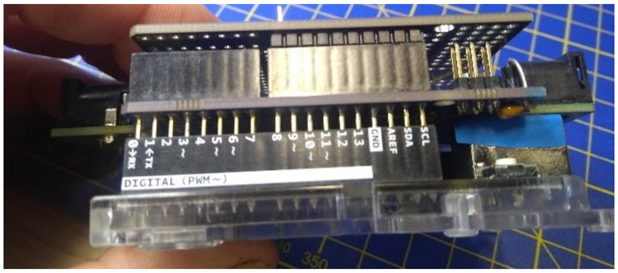




Pin header plugged into Arduino/Adafruit module, 1 pin header installed on the top ready to be connected to the required i/o and power supply.

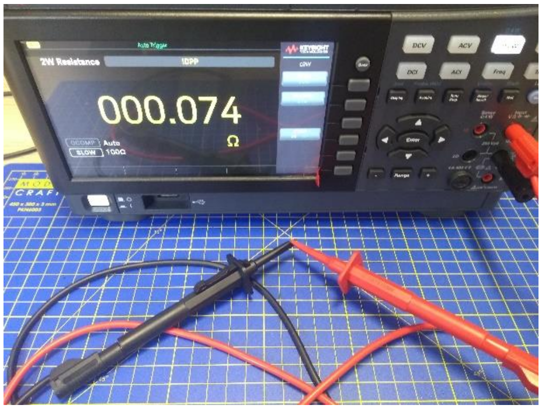
Side views.



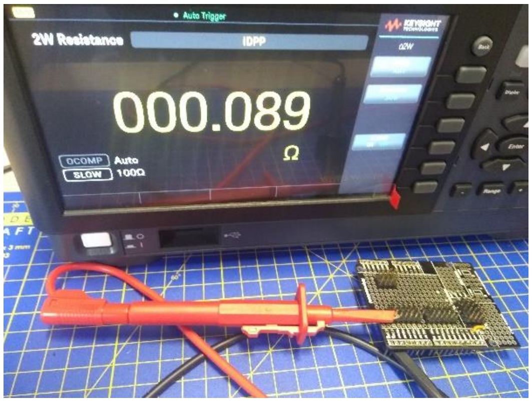


Before plugging anything into the Proto board or powering it up use the team bench multimeter to check for shorts and continuity.

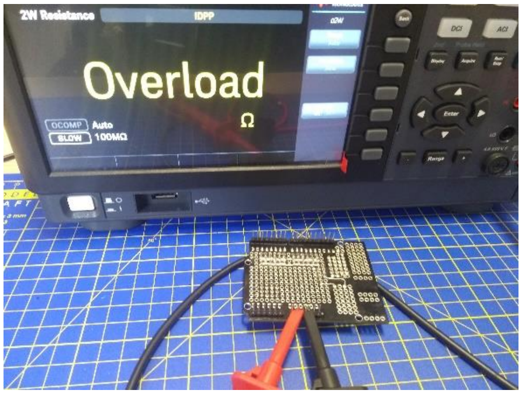
With the meter set to ohms short the probes together there should be little resistance indicated in this case .074 ohms or use the cont. setting



Checking continuity, I/O D2 to pin header all good .089 ohms or use the cont. setting



Checking for short circuit between 5V and GND all good open circuit

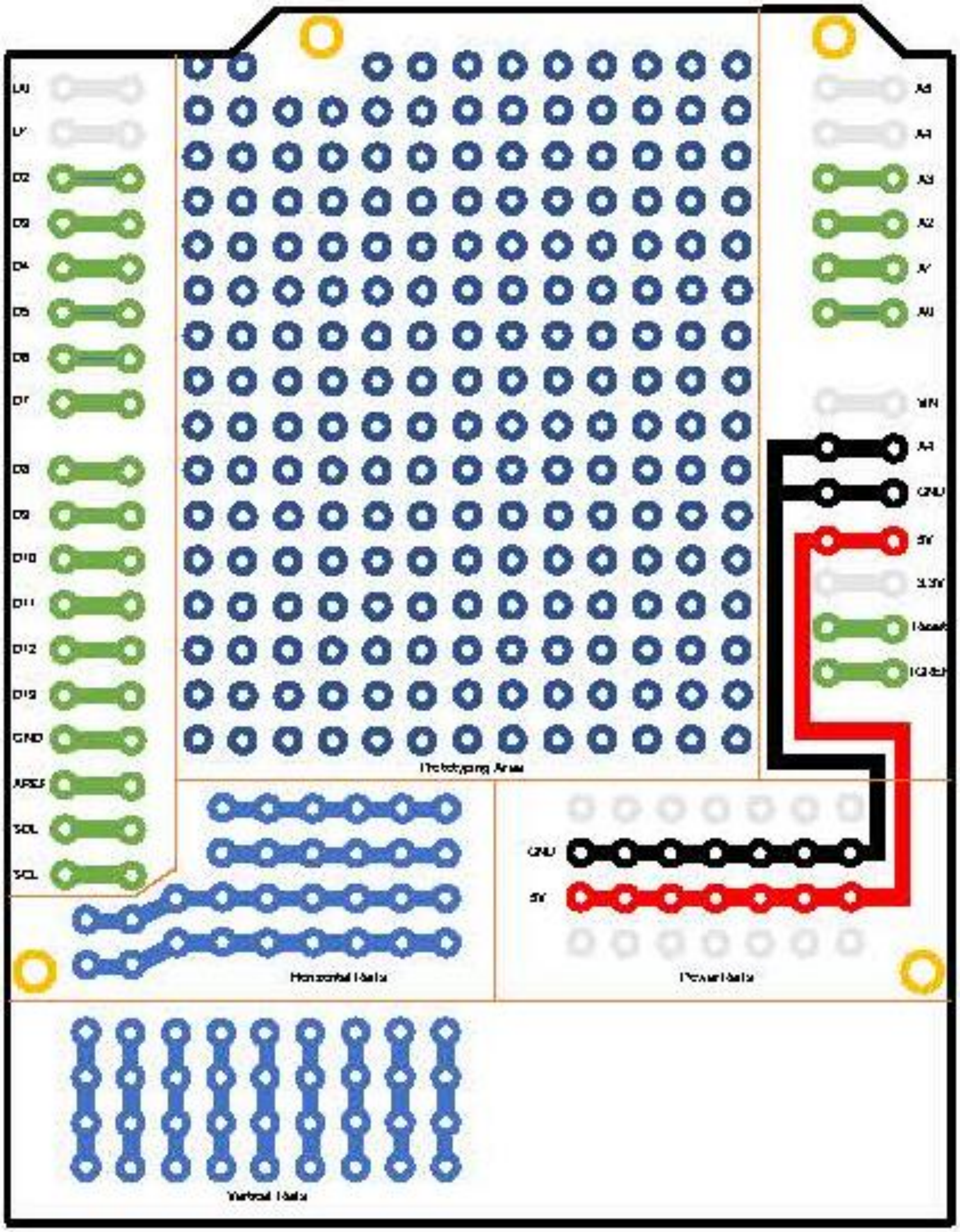


Hint/Tips

- The team will only get 1 prototyping board so get the design right before moving to the construction stage
- Allow some redundancy in your board to allow for modifications example, if the teams design requires 5 modules allow for 6
- Safety glasses and fume extraction must be used at all time when soldering.

Appendix 1

Prototyping Board layout


D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
GND
AREF
SDA
SCL
A5
A4
A3
A2
A1
A0
VIN
GND
GND
5V
3.3V
Reset
IOREF
Prototyping Area
GND
5V
Horizontal Rails
Power Rails
Vertical Rails