GW Tech Collective Arduino Workshop

Brandon Waller

October 1, 2017

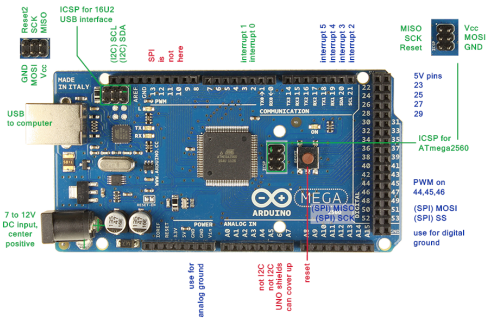
1 What is an Arduino

Arduino is a n open source computer that runs an ATmega328 microprocessor. It allows for fast prototyping for electronic, and allows for easy communication with sensors.

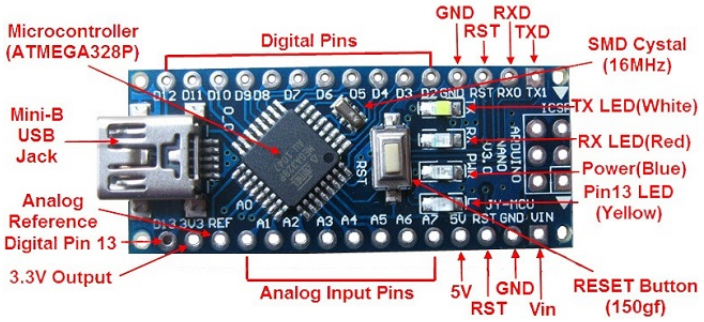
1.1 Types of Arduino Boards

Arduino UNO:

Arduino Mega:



Arduino Nano: Each of these boards is essentially the same with the main varia-



tion being the number of I/O pins and size. The UNO is considered to be the standard form factor.

2 Arduino Programming

2.1 Arduino IDE

To program an Arduino download the Arduino IDE here.

2.1.1 Configuring the IDE

- Tools > Board, select the correct board from the drop down list
- Tools > Programmer, select AVRISP mkII
- Tools > Port, Select the port that the Arduino is connected to. If you are unsure, unplug the Arduino and look for the port that is missing

2.1.2 Components

- Serial Monitor (ctrl + shift + m), method for communicating with the Arduino
- Verify, checks for errors within your code
- Upload, verifies code and uploads to the Arduino on the selected port
- Libraries, code from other sources that can be used in your own code. For example, the LiquidChrystal library can be used to interact with LCD displays.

2.2 Arduino Commands

- `#define name val`: Creates a global variable that cannot be changed
- `pinMode`: configures a pin as an input or output

  ```
  pinMode(2, INPUT) //configure pin 2 as an input
  pinMode(3, OUTPUT) //configure pin 3 as an output
  ```

- `Serial.begin(9600)`: enables the serial port

  `Serial.print()` print to serial port

  `Serial.read()` read from serial port

- `delay()`: pauses program execution

  ```
  delay(1000) //pause for 1s
  delay(250) //pause for 250ms
  ```

- `digitalWrite(pin, voltage)`: sends 5 or 0 volts to a pin

  ```
  digitalWrite(3, HIGH) //turn on pin 3
  ```

- `digitalRead(pin)`: Read the value of a pin (HIGH or LOW)
- `void setup()`

  Initialization declare pin modes and global variables, Serial begin

- `void loop()`

  Persistent code, the Arduino will repeat the code in `void loop()` until a new program is uploaded

  Try to keep as much code in other functions as possible to keep the main loop readable

2.2.1 Blink

Program to turn on and blink an led Before execution, take an LED and place the longer leg in pin 12 and the shorter one in GND

```
#define LED_PIN 12

void
setup()
{
  pinMode(LED_PIN, OUTPUT);
}
void
loop()
{
  //turn on LED
  digitalWrite(LED_PIN, HIGH);
  //wait 0.5s/500ms
```

```
  delay(500);
  //turn off LED
  digitalWrite(LED_PIN, LOW);
  //wait 0.5s/500ms
  delay(500);
  //start from top of void loop
}
```

3 Sensors and Components

Name	Function	Connection
Ultrasonic (HC-SR04)	Distance	Vcc - 5V, Trig - Digital Output, Echo - Digital Input, GND - GND
DHT 11/22	Temperature and Humidity	Link
Gyroscope and Accelerometer (MPU-6050)	Angle and Acceleration	link
7 Segment Display	Displaying Numbers	Link
Breadboard	Platform for building circuits	+/- rail connected in series, all numbered rows connected in series
Servo and Motors	Movement	Link

3.1 Porgramming with Sensors

Reading and Ultrasonic Sensor

Ultrasonic sensors work by sending out a pulse then reading the pulse's reflection. Distance is calculated by using the time difference between the pulse out and the pulse in:

```
#define ECHO 2
#define TRIG 3
double pulse_time = 0;
double in = 0;
double cm = 0;
/*
 *  Read the distance reading by
 *  an ultrasonic sensor and
 *  print the reading in in, cm
 */
double
read_ultrasonic(int ech, int trig)
{
 digitalWrite(trig, LOW);
 delayMicroseconds(5);
 digitalWrite(trig, HIGH);
```

```
 delayMicroseconds(10);
 digitalWrite(trig, LOW);

  return pulseIn(echo, HIGH) / 2;
  /*
   * Return time / 2 because the pulse
   *traveles to the object and backe
   */
}
void
setup()
{
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  Serial.begin(9600);
}
void
loop()
{
  pulse_time = read_ultrasonic(ECHO, TRIG);

  cm = pulse_time / 29.1;
  in = pulse_time / 74;

  Serial.print("in");
  Serial.print(in);
  Serial.print(" cm: ");
  Serial.println(cm);

  delay(5 * 1000);
}
```

3.1.1 Logic

Using the data from sensors, using the following commands to make decisions:

```
//Turn on LED is reading from ultrasonic is less than 5 inches
#define ECHO 2
#define TRIG 3
#define LED_PIN 4
double pulse_time = 0;
double in = 0;
double
read_ultrasonic(int ech, int trig)
{
 digitalWrite(trig, LOW);
 delayMicroseconds(5);
 digitalWrite(trig, HIGH);
 delayMicroseconds(10);
 digitalWrite(trig, LOW);

  return pulseIn(echo, HIGH) / 2;
```

```
  /*
   * Return time / 2 because the pulse
   *traveles to the object and backe
   */
}
void
setup()
{
  pinMODE(LED_PIN, OUTPUT);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  Serial.begin(9600);
}
void
loop()
{
  pulse_time = read_ultrasonic(ECHO, TRIG);
  in = pulse_time / 74;

  if(in < 5)
    digitalWrite(LED_PIN, HIGH);
  else
    digitalWrite(LED_PIN, LOW);
}
```

Other control statement:

Name	Function
`if(condition){/*code*/}`	Executed following code if the statement is true
`else if(condition){/*code*/}`	Executed if the if is false and the condition is true
`else`	Evaluate only if all other if/else if in the block are false
`for(i = 0; i < 10; i++){/*code */}`	Run `code` 10 times
`i = 0; while(i < 10){/*code */; i++;}`	Run `code` 10 times
Logic Operations:	
$a < b$	True if a is less than b, false if b is equal to or greater than a
$a > b$	True if a is greater than b, false if b is equal to or less than a
$a == b$	True if a is the same as b, false if not
$a >= b$	True if a is greater than or equal to b, false if not
$a <= b$	True if a is less than or equal to b, false if not
$a \&\& b$	True if a and b are both true, false if not
$a \|\| b$	True if a or b are true, false if not

Mathematical Operations