

An Introduction to Arduino a low cost digital prototyping platform

Clive Barry, Mogue Carpenter, Aileen Drohan, David Kirwan, and Martin
Walshe

South East Makerspace,
Old Printworks, Thomas Hill,
Waterford City, X91 TW63
info@southeastmakerspace.org
<https://www.southeastmakerspace.org>



Abstract. This workshop offers an introduction to the Arduino prototyping platform, useful for artists, hobbyists and the wider maker community. Each participant receives a pack containing an Arduino Uno and components to perform each of 5 simple experiments. No prior coding experience is needed as all code will be provided. The only requirement is the Arduino IDE, and a packed lunch!

Keywords: arduino, digital electronics, prototyping, introduction

Table of Contents

An Introduction to Arduino a low cost digital prototyping platform	1
<i>SEMS</i>	
Introduction	3
What is an Arduino?	3
Where might an Arduino be used?	4
Workshop Aims	6
Basic Circuit Theory	7
Electronic Components	10
Basic Arduino Coding Concepts	11
Experiment 1 - Blink	14
Experiment 2 - Button	16
Experiment 3 - Generating Music	18
Experiment 4 - MP3 Player	23
Experiment 5 - Phenakistoscope Creation	26
Thanks for taking part!	29
Glossary	30
Bibliography	32

Introduction

What is an Arduino?

"Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike." [Arduino, 2015a]

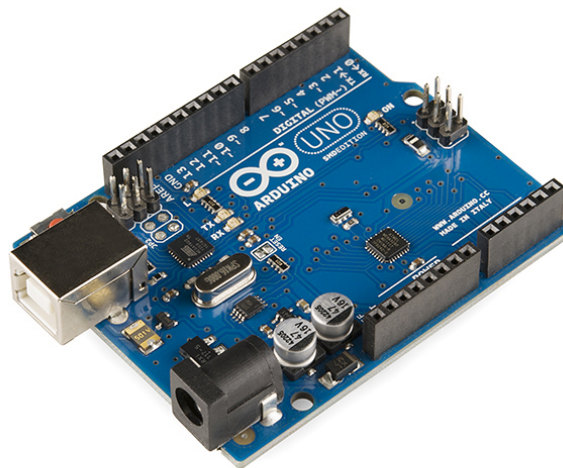


Fig. 1: Arduino Uno R3 [Wikipedia, 2013]

Open Source

Open Source can be described as resources and software that can be used, redistributed, rewritten and in some circumstances sold free of charge. The schematics and component list for hardware can also be open source as is the case with the Arduino platform [Culkin, 2011].

Electronics

Technology which makes use of the controlled motion of electrons through different media [Culkin, 2011].

Prototype

An original form that can serve as a basis or standard for other things [Culkin, 2011].

Platform

Hardware architecture with software framework on which other software can run [Culkin, 2011].

Where might an Arduino be used?

A few contrived examples of where one might use an Arduino in order to automate some process:

Automatic Dog's Water Bowl

An dog owner wants to ensure her pet is never left without water. She attaches a system for measuring the water level in the dog's bowl. Her Arduino is programmed to measure this value every 5 minutes. If this level falls below a certain value, a valve is opened and a water pump is activated to fill up the water bowl. It then sends an SMS to the owner to let her know that the dog is in safe hands.

- Read inputs from a water level sensor
- Control a valve which lets water flow
- Control speed of a water pump
- Send SMS to owner about giving the dog water

Bird Table Camera

A rising social media ornithologist wishes to share pictures from all the visitors to the bird table in his garden. He mounts an infra-red movement sensor on the bird table attached to an Arduino which is configured to record an image and send it to Twitter. His neighbours marvel at how many crows he's feeding.

- Read inputs from a movement sensor
- Control a camera shutter
- Transmit image back to PC
- Send tweet with picture of the bird table visitor

Fingerprint Door Lock

A student is sick of forgetting his keys and being locked out of his house. He uses a fingerprint scanner and an Arduino to make a biometric fingerprint door lock. He needs only scan his thumb print now and the door will unlock.

- Read inputs from a fingerprint sensor
- Compares the finger print against an authorised fingerprint
- Records the time and date a finger was pressed on the scanner
- Makes audio error tone if the fingerprint was invalid
- If valid fingerprint it unlocks the door

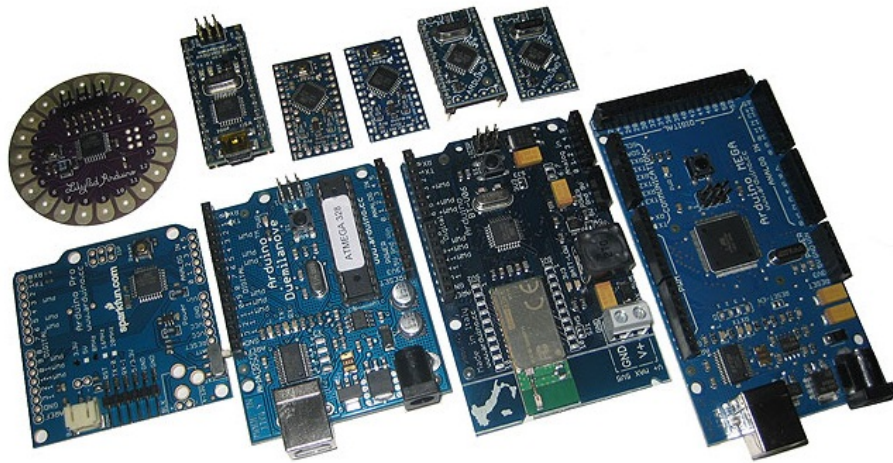


Fig. 2: Arduino Family

There are multiple different flavours of Arduino, each with different capability levels and or specialised features. It is through this variety which makes it possible to use the Arduino in a wide range of situations from home automation to wearable technology.

Workshop Aims

In this workshop the aim is to give you a crash course in digital electronics, and provide you with the basic skill-set to start using the Arduino or similar micro-controllers in your future projects.

Workshop Requirements

Each person will require the following:

- PC, either Linux, Mac or Windows can be used
- Arduino IDE pre-installed (Internet at the makerspace is flaky!)
- A sambo to keep you going

Learning Outcomes

Each person will leave with:

- Arduino starter kit
- Crash course in digital electronics
- Confidence to use Arduino in future projects

Arduino Starter Kit Contents

The Arduino starter kit contains the following components, which we will be making use of during the workshop.

- 1 × Arduino Compatible R3 Uno
- 1 × Breadboard
- 16 × jumper wires various colours
- 20 × 5mm LED's assorted colours
- 10 × 10k Ω resistors
- 10 × 330 Ω resistors
- 1 × RGB LED
- 1 × photo resistor
- 2 × push buttons
- 1 × DFRobot MP3 Module
- 1 × 1GB SD Card
- 1 × 5 Ω speaker

Basic Circuit Theory

In an electrical circuit there is a fundamental relationship between voltage, current and resistance and it is explained by Ohms Law [ElectronicsTutorials, 2015].

Voltage

Voltage, (SI Unit: V - Volts) is the potential energy of an electrical supply stored in the form of an electrical charge. Voltage can be thought of as the force that pushes electrons through a conductor and the greater the voltage the greater is its ability to push the electrons through a given circuit [ElectronicsTutorials, 2015].

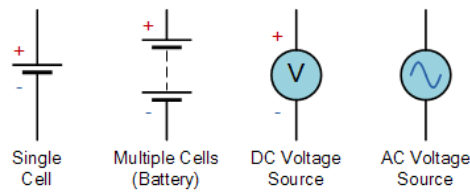


Fig. 3: Voltage Symbols [ElectronicsTutorials, 2015]

Current

Current, (SI Unit: A - Ampere) is the movement or flow of electrical charge and is measured in Amperes. It is the continuous and uniform flow (called a drift) of electrons (the negative particles of an atom) around a circuit that are being pushed by the voltage source [ElectronicsTutorials, 2015].

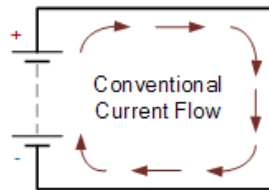


Fig. 4: Current Symbols [ElectronicsTutorials, 2015]

Resistance

Resistance, (SI Unit: Ω - Ohms) of a circuit is its ability to resist or prevent the flow of current (electron flow) through itself making it necessary to apply a greater voltage to the electrical circuit to cause the current to flow again. Note that Resistance cannot be negative in value only positive [ElectronicsTutorials, 2015].

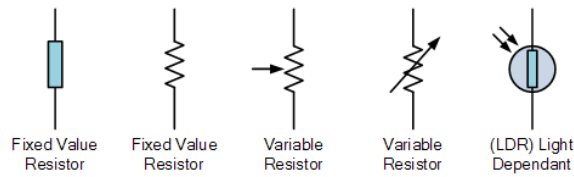


Fig. 5: Resistance Symbols [ElectronicsTutorials, 2015]

Ohm's Law

The following equation Ohm's Law explains the relationship between Voltage, Current and Resistance for an electrical circuit. For more information see: [ElectronicsTutorials, 2014].

$$V = I \times R \quad (1)$$

$$I = \frac{V}{R} \quad (2)$$

$$R = \frac{V}{I} \quad (3)$$

Fig. 6: Ohm's Law

Analogue vs Digital Signals

The world we live in is an analogue one. There are an infinite number of possible combinations of colours smells and sounds. The digital world however is not infinite. It is discrete and finite where we are limited by several factors, such as

memory and computational capabilities. In order to represent a real world thing digitally, it is by necessity that some information is lost.

Analogue Signal Examples

The following are some examples of analogue signals:

- Temperature
- Sound
- EM Radiation

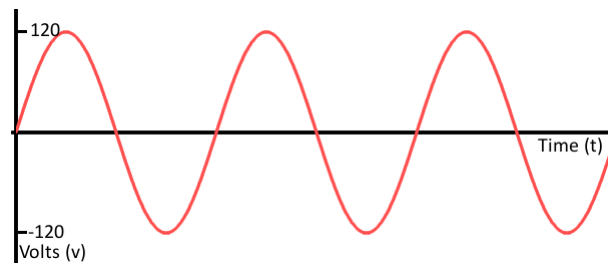


Fig. 7: Analogue Signal [Lindblom, 2015]

Digital Signal Examples

The following are some examples of digital signals:

- Morse Code
- WIFI
- Binary

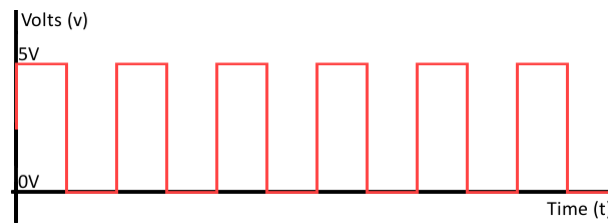


Fig. 8: Digital Signal [Lindblom, 2015]

Electronic Components

The following section contains more information on some of the various components included in the *Introduction to Arduino* workshop.

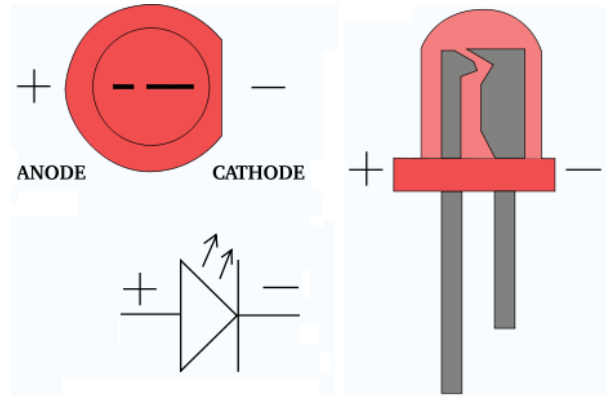


Fig. 9: LED Component [Sapkota, 2010]

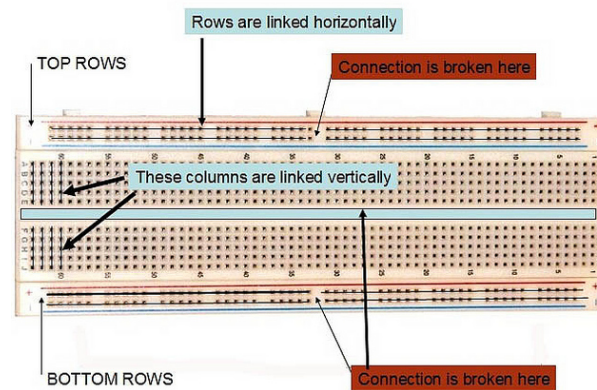


Fig. 10: Breadboard [Sapkota, 2010]

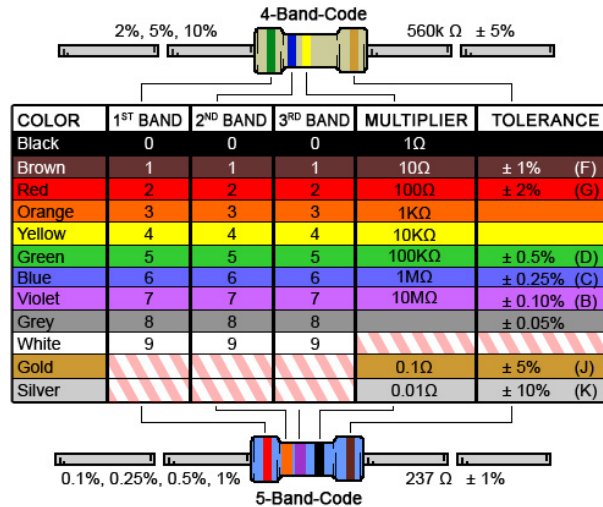


Fig. 11: Resistor Chart [Digikey, 2016]

Basic Arduino Coding Concepts

Variables

A variable is a way of naming and storing a value for later use by the program, such as data from a sensor or an intermediate value used in a calculation [Arduino, 2015b]. Table: 1 shows some of the basic variable

Type	Example
char	'a'
string	'Hello there'
byte	0 to 255
int	-32,768 to 32,767
unsigned int	0 to 65,535
long	-2,147,483,648 to 2,147,483,647
unsigned long	0 to 4,294,967,295
float	-3.4028235E38 to 3.4028235E38

Table 1: Variable Types in Wiring

types available to us for use in the Arduino. There are other more advanced data types such as Arrays and Pointers, which we may use, but

won't explain them in this introduction. More information can be found regarding these data types online. See the Arduino online reference library at: <https://www.arduino.cc/en/Reference/HomePage>.

setup() function

The `setup()` function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each power-up or reset of the Arduino board [Arduino, 2015c].

```
void setup(){ // setup initializes serial
  Serial.begin(9600);
}
void loop(){ // write to serial, then wait 2 seconds
  Serial.write(" Hello World");
  delay(2000);
}
```

loop() function

After creating a `setup()` function, which initializes and sets the initial values, the `loop()` function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board [Arduino, 2015d].

```
// Create int variable to represent
// a physical pin on the Arduino
int buttonPin = 3;

// setup initializes serial and the button pin
void setup()
{
    Serial.begin(9600);
    pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop()
{
    if (digitalRead(buttonPin) == HIGH)
        Serial.write('H');
    else
        Serial.write('L');

    delay(1000);
}
```