Learning Kit

USB Cable

Uno Board

Breadboard

Jumper Leads

Resistors

LDR Module

RGB LED

Buttons

8 x 8 LED Dot Matrix

Potentiometer

LED's

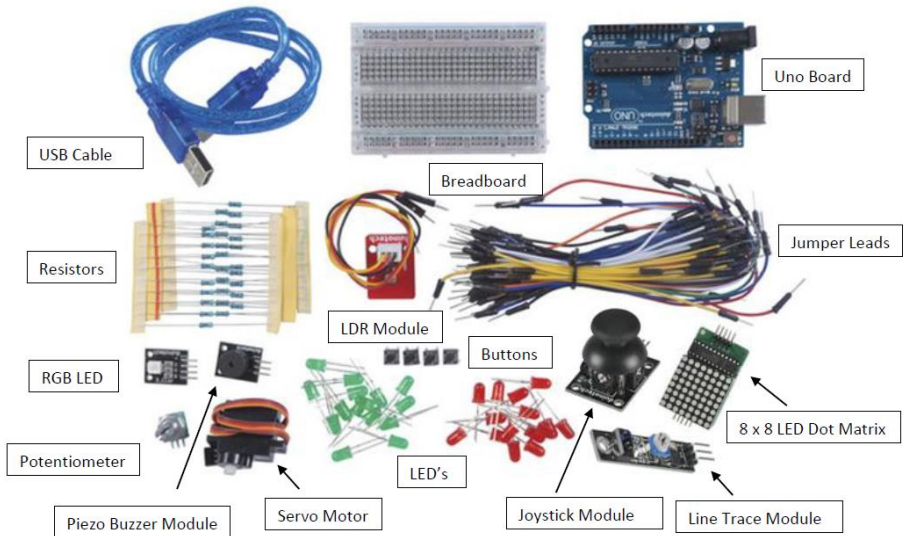Piezo Buzzer Module

Servo Motor

Joystick Module

Line Trace Module

What is Arduino®?

Arduino® is an open source hardware and software prototyping environment, but what does this mean? Open source means that the boards and programs have been developed by a community of people who are passionate about building their own ideas into projects, but also are willing to share their knowledge, designs, code and skills with other like-minded people. You not only have access to what's in this kit, but a wealth of shared resources, examples and project ideas to help you develop and learn. You can find out more at the Arduino® website: https://www.arduino.cc/

Arduino® Hardware:

The Arduino® boards are designed with education in mind- in fact they were developed at a University to make electronics easy for artists and other people unfamiliar with programming electronics. It's possible to build complex circuits without soldering, but simply by plugging jumper leads, modules and breadboard together with the main Uno board.

Arduino® Software:

The Arduino® IDE (integrated Developed Environment) is available as a free open-source download that works on Windows, Mac and Linux, and provides beginners with a simple intuitive programing language. It also allows advanced users to develop powerful programs using the vast range of downloadable libraries that the Arduino® community has created.

This Kit:

We have specially selected the components in this kit to allow an easy entry into the world of Arduino®, so that you can work with the examples included with the Arduino® IDE, and obtain a broad understanding of how the Arduino® software and hardware interact. This kit is also compatible with the full range of Duinotech shields and modules, which can be used to extend your Arduino® experience.

Getting Started

Before using your Duinotech Beginner's Kit, you'll need to download and install the Arduino® IDE from https://www.arduino.cc/en/Main/Software. Open the Arduino® IDE, and plug USB cable into the Uno board and a USB port on the computer. Let your computer install the drivers for the Uno board.

Project 1: Blink

Now we're going to open a sketch and program the Uno. A sketch is a set of instructions that the Arduino® IDE compiles into a program the Uno can understand. Click on File>Examples>01.Basics>Blink, and a new window will open with some sketch code. If you want to understand the code, see https://www.arduino.cc/en/Tutorial/Blink.

Now click Tools>Board>Arduino Genuino/Uno. This tells the IDE that we're using an Uno board. If you do use other boards in the future this might be something different.

Then click Tools>Port and then click the Serial Port of the Uno. It might be labelled 'Arduino/Genuino Uno'. If you're not sure which one it is, unplug the Uno board and try clicking into the menu again to see which Serial Port disappears. Plug the Uno board back in and check that the board reappears.

Finally, click on the [→] button to compile and upload your sketch. After a few seconds, the TX and RX lights on the Uno will flicker (this means that programming is occurring), then the L LED will flash on and off slowly- the Uno is running our sketch.

If you didn't get the LED flashing, check out the troubleshooting guide at https://www.arduino.cc/en/Guide/Troubleshooting

More Projects

LEDs:

See File>Examples>03.Analog>Fading (or Fade)
https://www.arduino.cc/en/Tutorial/Fade
https://www.arduino.cc/en/Reference/AnalogWrite

Button:

See File>Examples>02.Digital>Button
https://www.arduino.cc/en/Tutorial/Button
https://www.arduino.cc/en/Reference/DigitalWrite
https://www.arduino.cc/en/Reference/DigitalRead

Buzzer:

See File>Examples>02.Digital>toneMelody
https://www.arduino.cc/en/Tutorial/toneMelody
https://www.arduino.cc/en/Reference/Tone

Servo Motor:

See File>Examples>Servo>Sweep
https://www.arduino.cc/en/Tutorial/Sweep
https://www.arduino.cc/en/Reference/Servo

LDR and Potentiometer:

See File>Examples>03.Analog>AnalogInput
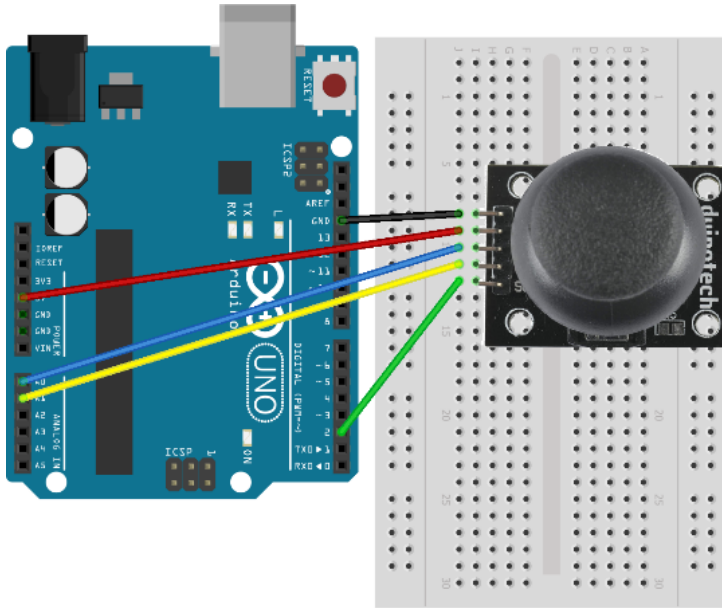https://www.arduino.cc/en/Tutorial/AnalogInput
https://www.arduino.cc/en/Reference/AnalogRead

Joystick:

The Joystick Module is actually the same as a two potentiometers and a button built into the one module. Connect the Joystick to the Uno via the breadboard as follows:

This is image was created with Fritzing

You can read the x-position of the joystick by doing:
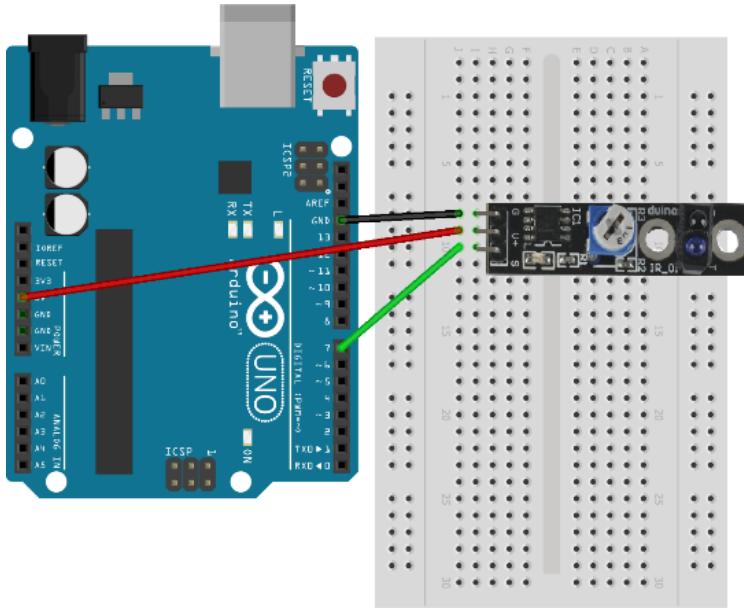
```
analogRead(A0);
```

and the y-position:

```
analogRead(A1);
```

See if you can combine the AnalogInput and Button sketches to work with the Joystick.

Line Trace:

The Line Trace Module can be connected as below. The input can be read with the digitalRead function (the same as the button examples).

This is image was created with Fritzing

This simple sketch can be used to display the sensor output on the Serial Monitor.
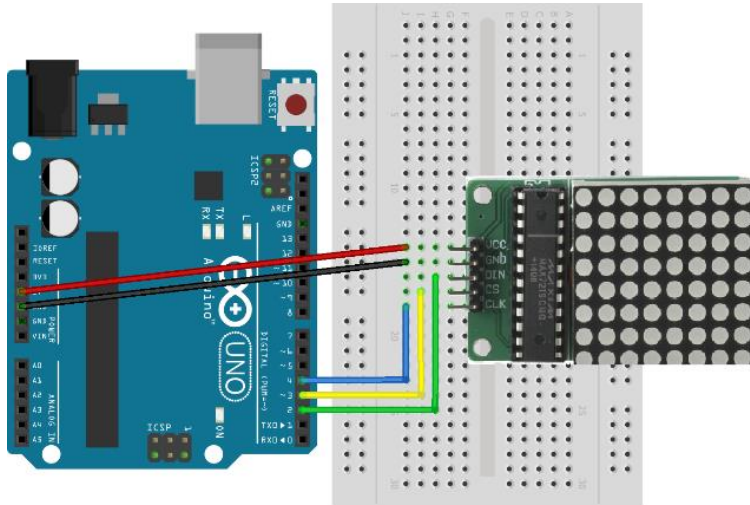
```
void setup() {
      Serial.begin(9600);
}
void loop() {
      Serial.println(digitalRead(7)?"HIGH":"LOW");
      delay(200);
}
```

LED Matrix:

See the references and sample code here:
http://playground.arduino.cc/Main/LEDMatrix
http://playground.arduino.cc/LEDMatrix/Max7219

This is image was created with Fritzing

More Resources

One of the great things about the Arduino® is the community, and there is a good chance that if you can't make something work, there's someone out there who has already worked out that problem and a Google search might even find them. Some helpful pages:

https://forum.arduino.cc/
https://www.arduino.cc/en/Reference/HomePage
https://www.arduino.cc/en/Tutorial/BuiltInExamples
http://playground.arduino.cc/