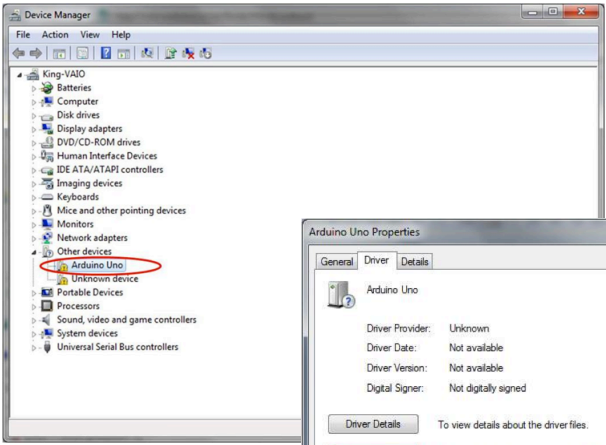
PROTO-PIC


# Starter Guide for Arduino

includes example projects

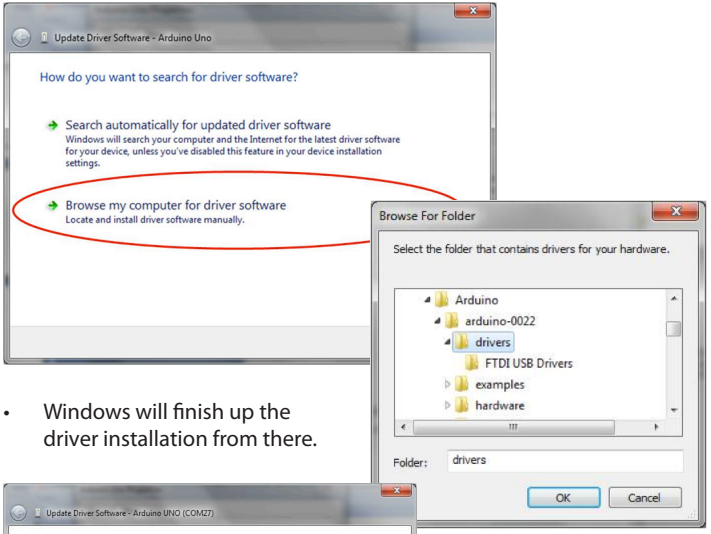
# INSTALLATION GUIDE

- First you will need to download the latest version of the Arduino software. This can be found at: arduino.cc/en/main/software.
- Install the software and driver appropriate to your Operating System- the following assumes Windows 7 is used.
- Plug in your board and wait for Windows to begin its driver installation process. If Windows fails to find the device driver then click on the Start Menu and open up the Control Panel.
- While in the Control Panel navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO" (or the name of the processor you are connecting)

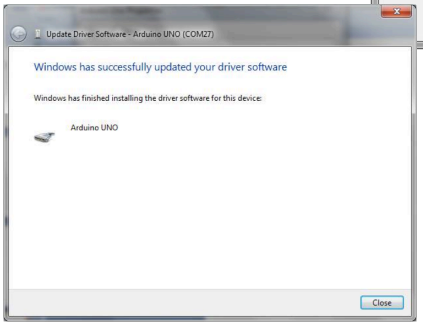



- Right click on the "Arduino UNO" port and choose the "Update Driver Software" option.

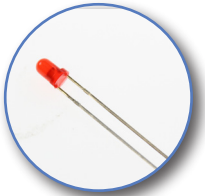
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno's driver file named "Arduino UNO.inf"

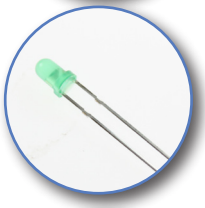


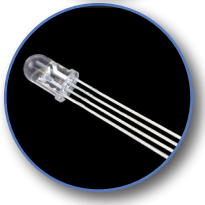

- Windows will finish up the driver installation from there.



## LEDs







An LED or Light Emitting Diode comes in a whole range of colours and sizes. We have a few of the most popular in this kit. They are polarity sensitive devices (you need to connect them up the right way round to get them to work!).They have a Short Leg and a Long Leg. The Long Leg indicates the Anode (or positive), which means the Short Leg is the Cathode (or negative). There is also a flat section of the LED head, which also indicates the Cathode – handy to know if you have already trimmed both legs to the same size!

**Red LED.** A 3mm Red LED. For the datasheet please visit:
http://proto-pic.co.uk/3mm-red-led-diffused/

**Green LED.** A 3mm Green LED. For the datasheet please visit:
http://proto-pic.co.uk/3mm-green-led-diffused/

**RGB LED.** This is a special kind of LED which is like three separate LEDs rolled into one. One Red, one green and one blue. By lighting each of these differently coloured sections by varying amounts, you can create a vast range of other colours. The one in your kit is a Common Cathode version – the longest leg out of the four is the Cathode. For more info please goto:
http://proto-pic.co.uk/5mm-rgb-led-common-cathode/

## Breadboard

Breadboards are fantastic things for quickly prototyping circuits. Each column of holes is connected together internally so components can be easily connected together. We have included a generously sized breadboard for all the projects you will soon want to do!
http://proto-pic.co.uk/half-size-breadboard/

## Resistors

A resistor is a small component which has a big place in an electronic circuit. It can be used to limit the flow of current through the circuit or to tie things up to a positive voltage or down to Ground. There's nothing worse than a "floating" input.

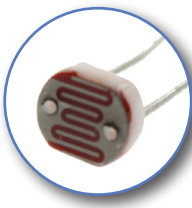
**330 ohm resistor** – the colour bands on this resistor are Orange, Orange, Black. For the datasheet please visit:
http://proto-pic.co.uk/330-ohm-1-4-watt-pth/

**1k ohm resistor (1000 ohms)** – the colour bands on this resistor are Brown, Black, Red. For the datasheet please visit:
http://proto-pic.co.uk/1k-ohm-1-4-watt-pth/

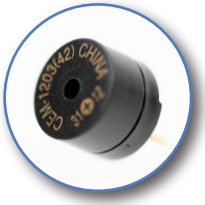
**10k ohm resistor (10000 ohms)** – the colour bands on this resistor are Brown, Black, Orange. For the datasheet please visit:
http://proto-pic.co.uk/10k-ohm-1-4-watt-pth/

## LDR

An LDR or Light Dependant Resistor is really a resistor whose resistance changes depending on how much light falls on to it. For more info please goto:
http://proto-pic.co.uk/mini-photocell/



## Piezo Buzzer

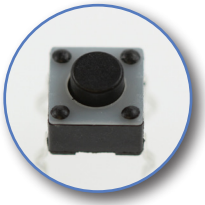



This cool little buzzer will fit nicely into your project and will emit little squeals of delight when you drive it. The detailed info can be found here:
http://proto-pic.co.uk/buzzer-pc-mount-12mm-2-048khz/

## Thermistor

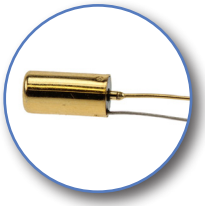
This is a resistor who's resistance changes by temperature. A cool simple project could be to have a buzzer squeal more and more dependant on the body temp of the user that touches the thermistor. For the datasheet on the one incuded goto:
http://proto-pic.co.uk/thermistor-10k/

## Push button switch



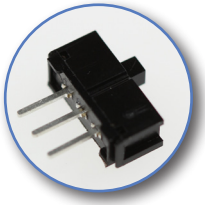
a simple momentary (when you press it it's closed and when you let go it's open) that will fit nicely into the included Breadboard. For more info goto:
http://proto-pic.co.uk/mini-push-button-switch/

## Tilt Switch



This little switch will be closed whenever it is upright, however if you tilt it over the contacts open and it wont conduct.
http://proto-pic.co.uk/tilt-sensor/

## Slide Switch



This switch has three contacts, so you can wire it up to trigger two different things. Also known as a SPDT switch (Single Pole Double Throw), more info can be found here:
http://proto-pic.co.uk/spdt-mini-power-switch/

## Capacitors

A capacitor is a component which stores energy. It can be charged up over time and then discharged. Commonly used in electronic circuits for anything from timing to blocking DC.

10nF Capacitor. This Ceramic Capacitor is marked with '103' It does not matter which way round you connect it up.
http://proto-pic.co.uk/ceramic-capacitor-10nf-50v/

100nF Capacitor. Also known as 0.1uF, This Ceramic Capacitor is marked with '104' It does not matter which way round you connect it up.
http://proto-pic.co.uk/capacitor-ceramic-0-1uf/


100uF Electrolytic Capacitor. This capacitor has the negative lead marked with a row of minus symbols. It definitely matters which way round you connect it!
http://proto-pic.co.uk/electrolytic-capacitor-100uf-50v/

## Diode

A diode is a component that blocks the flow of current in one direction only. It must be connected up the correct way round and the white band on the diode body indicates the cathode.
http://proto-pic.co.uk/diode-rectifier-1a-50v-in4001/

## USB A – B Cable

You will use this a lot! Used for uploading your programs onto the Arduino, It has a rectangular 'A' connector for connection to a PC and a square 'B' connection on the other end for plugging into the Arduino.

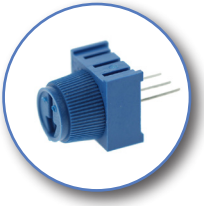
## Jumper Wires

We provide a big selection of different lengths and colours of male to male jumper wires. They plug into the breadboard and the connection pins on the Arduino and are used to connect all your cool projects together.
http://proto-pic.co.uk/breadboard-jumper-wire-pack/

## PP3 Battery Clip

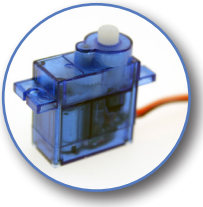
You can use this to connect a PP3 battery (the 9V rectangular type) to the DC input Jack on the Arduino.
http://proto-pic.co.uk/9v-battery-clip-with-2-1mm-plug/

## Potentiometer



This is a resistor where the resistance can be varied by turning the knob on top. Widely used in electronics. The old volume dial on TVs was effectively a variable resistor. The one included is 10 kOhm which means is resistance can be changed from roughly 0-10 kohms.
http://proto-pic.co.uk/trimpot-10k-with-knob/

## Servo Motor



A typical servo motor consists of a small geared electric motor which has an in-built potentiometer which allows for position control. Great for use in RC projects and anything where you want something to move to a specific position
http://proto-pic.co.uk/analogue-9-gram-servo/

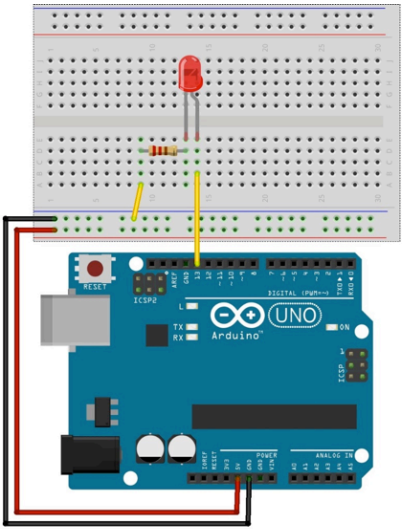
# Project 1 LED Blink

In this project we will connect up an LED with a RESISTOR – we will explain the polarity of the LED, and why we use a RESISTOR.
First find :

**330 Ohm RESISTOR (Orange – Orange - Brown Stripes)**
**LED (Any colour, but not the one with 4 legs)**

The RESISTOR is an unpolarised component, this means that it can be mounted in either direction.
The LED is polarised, if you look at the LED body base it has a flat section, this shows the CATHODE or minus side, if you look at the legs, there is a longer leg, this shows the ANODE or plus side. In some circuits you may come across, the minus side MAY be marked as GND, GROUND or with a symbol that looks like :

*Place the components on the breadboard and connect up the Arduino as follows :*



The LED has the short leg towards the RESISTOR. The RESISTOR is used to limit the current going through the LED, with out it the LED would burn out as it can only take about 20mA of current.

**Copy the following code into the Arduino Program :**

```
int led = 13;  //This is the PIN the LED is controlled from
int onTime = 1000;
int offTime = 1000;
// the setup routine runs once when you press reset, or power up the Arduino
void setup()
{
   pinMode(led, OUTPUT);    // initialize the digital pin as an output.
}
// this loop routine runs over and over again forever:
void loop()
{
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(onTime);                  // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the voltage LOW
  delay(offTime);                 // wait for a second
}
```

Upload this to the Arduino, and the LED Should blink on for 1 second & off for 1 second.

To change the time, just modify the values of onTime and offTime.

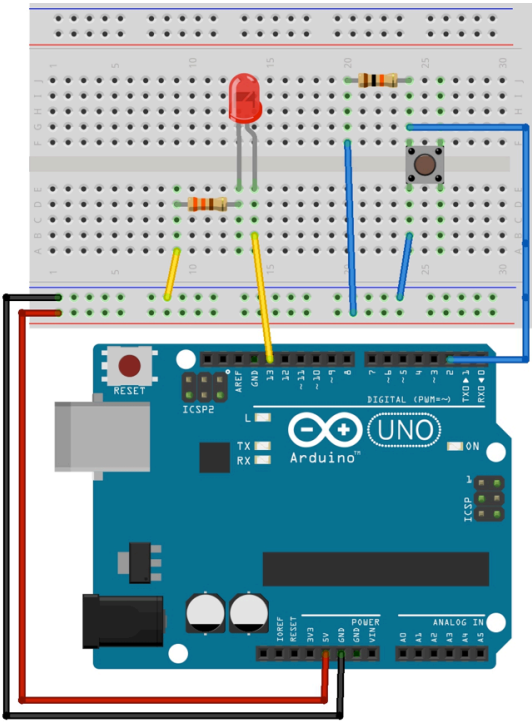
***Congratulations, you have just completed your first project.***

# Project 2 LED – With Button control

In this project we will connect up an LED with a RESISTOR – and a BUTTON SWITCH with a pull-up resistor.
First find :

**BUTTON SWITCH**

*Place the components on the breadboard and connect up the Arduino as follows :*



As you can see the LED part of the circuit is the same as project 1, with the addition of a BUTTON SWITCH and a 10K Ohm RESISTOR (Brown – Black – Orange).
The RESISTOR connects pin 2 to 5V (called a pull-up RESISTOR, as it pulls the pin UP to a known voltage), giving a limited current, so pin 2 detects a HIGH signal. When you press the button, you short circuit pin 2 to ground, and the pin detects a LOW signal. The Arduino can be programmed to do something if it detects a HIGH or LOW signal on its pins. In our case, we will light an LED.
**Copy the following code into the Arduino Program :**

```
int buttonState = 0;       // variable for reading the pushbutton status
void setup()
{
  // initialize the LED pin as an output:
  pinMode(13, OUTPUT);     //You can set it just using its number
  // initialize the pushbutton pin as an input:
  pinMode(2, INPUT);
}
void loop()
{
  // read the state of the pushbutton value:
  buttonState = digitalRead(2);

  // check if the pushbutton is pressed. if it is not, the buttonState is HIGH:
if (buttonState == HIGH)
{
    // turn LED off:
    digitalWrite(13, LOW);
  }
  else
{
    // turn LED off:
    digitalWrite(13, HIGH);
  }
}
```

When you press the button the LED should light, and when you let go it goes out. Try to see what you have to change to get it to reverse this, ie – When you are not pressing the button the LED is ON, and when you press it it goes OFF.

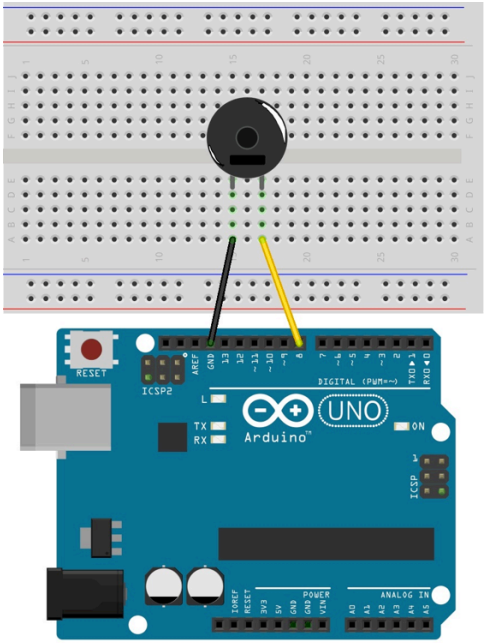
# Project 3 LETS MAKE NOISE

In this project we will connect up the BUZZER & make some noise.
Find :

**The BUZZER**

The BUZZER is a polarised component, if you have a look on the back you will see the pins marked with + and -. The + will go to the connection going to pin 8, and the – to the connection going to GND on the Arduino.

*Place the components on the breadboard and connect up the Arduino as follows :*



**Copy the following code into the Arduino Program :**

```
void setup()
{
  tone(8,262,250); //(outputPIN,note,length)
  delay(325);
  tone(8,196,125);
  delay(162.5);
  tone(8,196,125);
  delay(162.5);
  tone(8,220,250);
  delay(325);
  tone(8,196,250);
  delay(325);
  tone(8,0,250);
  delay(325);
  tone(8,247,250);
  delay(325);
  tone(8,262,250);
  delay(325);
  noTone(8); //Turn off the sound
  }
void loop()
{
 //This just sits doing NOTHING
}
```

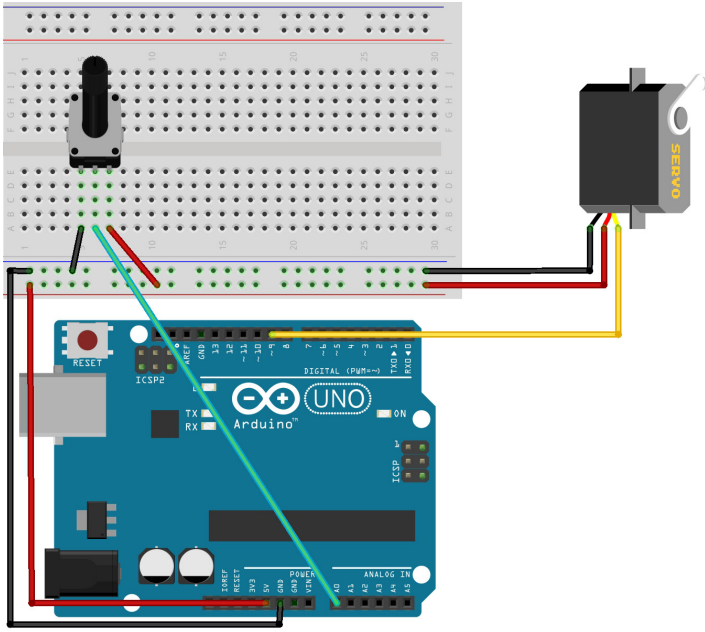
Upload this to the Arduino, the BUZZER Should play a short tune. If you look at the code, this should give you an idea on how to implement a start up routine, using sound or LEDs etc.

# Project 4 SERVO – Making it move

In this project we will connect a ROTARY POTENTIOMETER and a SERVO MOTOR – we will show how to read an analog voltage (a voltage between 0 Volts and 5 Volts) rather than an On/Off digital signal.

The POTENTIOMETER is not polarised, but it is good practice to have 0 Volts on the first pin (usually on the left), and 5 Volts on the right hand pin. The centre pin is called the WIPER or TAP pin as this is the output from a connector that wipes across the resistor in the inside.
The SERVO MOTOR connector is polarised, and has 3 connections: Brown – 0 Volts, Red – 5 Volts and Orange - signal.

*Place the components on the board as follows :*



The POTENTIOMETER you have in the kit may be different from this one, but the internal and external connections are the same.

**Copy the following code into the Arduino Program :**

```
#include <Servo.h>  //This loads a Library file telling the Arduino how to use the SERVO
Servo myservo;       // create servo object to control a servo
int potpin = 0;          // analog pin used to connect the potentiometer
int val;                 // variable to read the value from the analog pin
 void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
 void loop()
{
  val = analogRead(potpin);          // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 179);    // scale it to use it with the servo (value between 0 and 179)
  myservo.write(val);                // sets the servo position according to the scaled value
  delay(15);                      // waits for the servo to get there
}
```

Upload this to the Arduino and the SERVO should move when the POTENTIOMETER is rotated back and forth.

This can be used as the basis of a servo tester. Try removing the POTENTIOMETER and adding a couple of buttons to make it open and close. See if you can figure out where the code needs changed to do this.

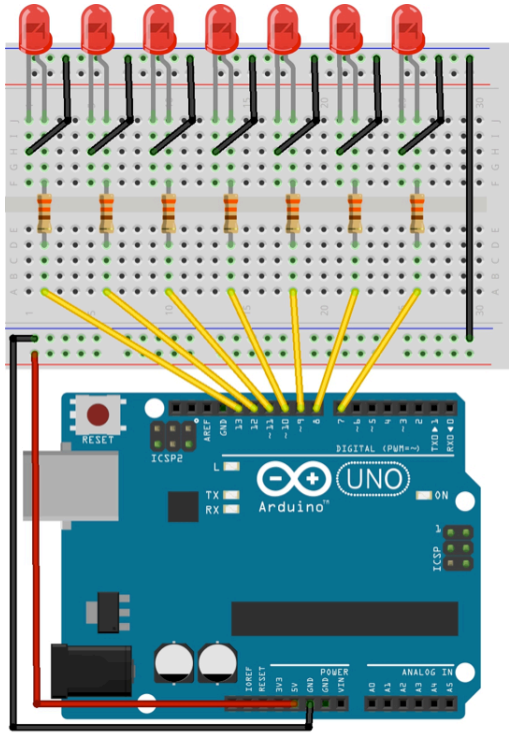
# Project 5 LARSON SCANNER

In this project we will connect up 7 LEDs with RESISTORS, and program the Arduino to light them in sequence similar to Knight Rider's KITT or a Cylon's Visor.

First find :

**7 x 330 Ohm RESISTORs (Orange – Orange – Brown)**
**7 x LEDs (RED)**

*Place the components on the breadboard and connect up the Arduino as follows :*



**Copy the following code into the Arduino Program :**

```
int spd=100;                    //Change this to change the scanning speed.
void setup()
{
for (int outpin = 7; outpin < 14; outpin++)  //This sets pins 7 to 13 as
outputs (Less typing this way)
{
pinMode(outpin,OUTPUT);
}
}
void loop()
{
 for (int strobe = 7; strobe < 13; strobe++)  //Turn on (then off) pins 7 to 12
{
 digitalWrite(strobe,HIGH);
 delay(spd);
 digitalWrite(strobe,LOW);
}
 for (int strobe = 13; strobe > 7; strobe--)  //Turn on (then off) pins 13 to 8
{
 digitalWrite(strobe,HIGH);
 delay(spd);
 digitalWrite(strobe,LOW);
}
}
```

From what you know, can you change the program to change the delay using the POTENTIOMETER?!


For more examples and further information on this kit please see our online documentation at

**learn.proto-pic.co.uk**

**www.proto-pic.co.uk**

Technical Support +44 (0) 1592 572092