

# Rather Useful Seminars

Fun in the  
Embedded World  
with Arduino

Rob Miles

## What is an Arduino?

- Arduino isn't a thing, it's an open standard for the hardware and software that make up a microcontroller based computing platform
- You can buy "official" Arduino boards based on the standard, but you can also get "clones"
- You can even make your own devices based on the design

## What is a Microcontroller?

- The Arduino platform is based around microcontroller devices
- A *microcontroller* is a single chip which contains all the components that make up a working computer
  - Central Processor Unit (CPU)
  - Program Storage
  - Data Storage
  - Input/Output
  - Clock

## Microcontroller vs Microprocessor

- A microprocessor is a single chip that contains all the processing elements of a computer
  - Address and data registers, timing, microcode etc
- Microprocessors need chips around them to provide program and data storage
  - They can be used as the basis of a computer, but they are not a computer
- A microcontroller contains the microprocessor element and all the other bits too
  - It just needs power and a clock to work

## Why isn't everything powered by microcontrollers?

- Microcontrollers are very small computers
- They are smaller than a modern computer by a factor of around 500
  - Very limited space for programs and data
  - Slower clock speed
- But then again, they only cost a few pounds

## Arduino Devices

- Arduino themselves make and sell processor boards
- But they also make the designs openly available for others to implement
- This means that there are lots of Arduino clones



## Arduino Devices

- Arduino themselves make and sell processor boards
- But they also make their designs openly available for others to implement
- This means that there are lots of Arduino clones
  - You can get Funduino boards for less than each £10 on ebay



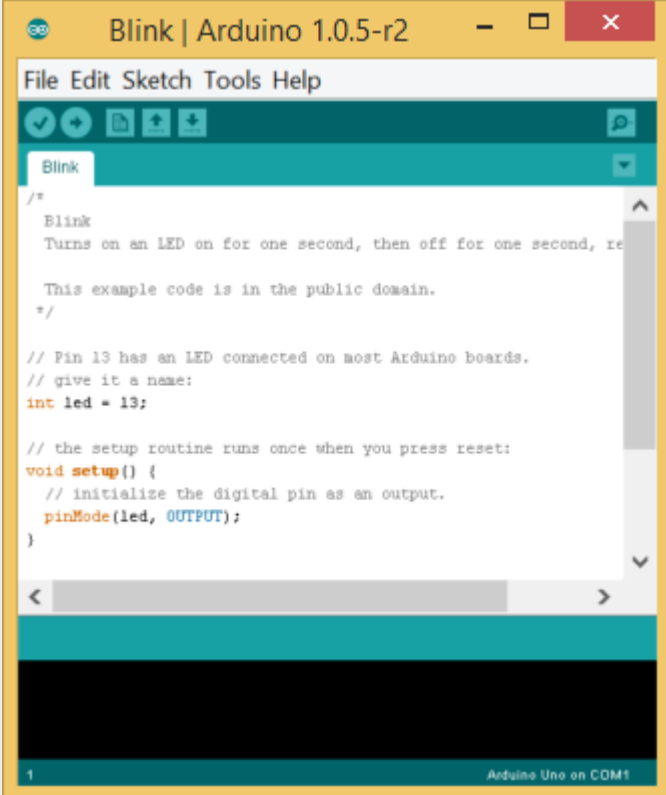
# The Missing Operating System

- Most computers have an operating system
  - This looks after things like the filestore and starts and stops programs for the user
- The Arduino does not have an operating system
- When the power comes on the program just runs



## Compiling code for the Arduino

- The Arduino platform provides an environment that includes an editor and a compiler
- This runs on PC or Mac and lets you write the C code which is compiled and then deployed into a connected Arduino device



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0.5-r2". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The main editor area displays the following C code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeating.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

At the bottom of the IDE, a status bar indicates "1" and "Arduino Uno on COM1".

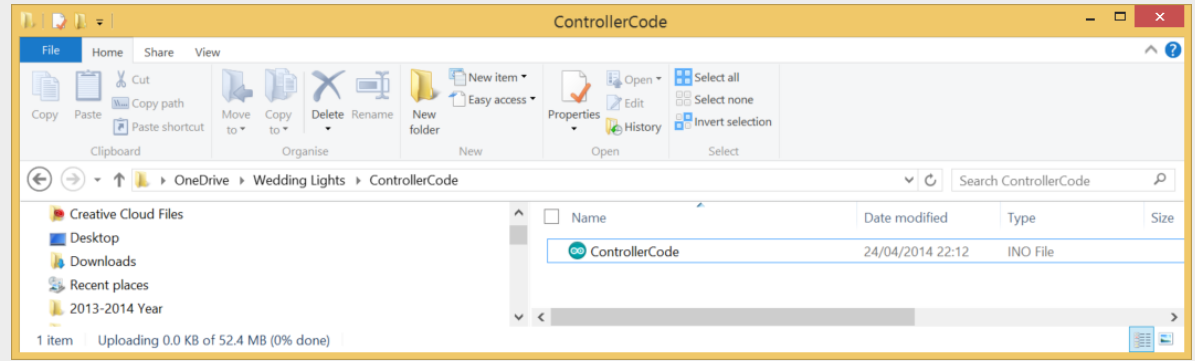
## Programming Language

- You write the Arduino programs in C++
- You can use assembler if you like (but this is only required if you have a serious need for speed)
- Most of learning the Arduino is learning how the library routines work, not the programming language....

## The Bad News

- The program just runs in the device when you press the Upload button
- If the program isn't correct there is no debugging support
- You survive by adding print statements that you can view on the host computer

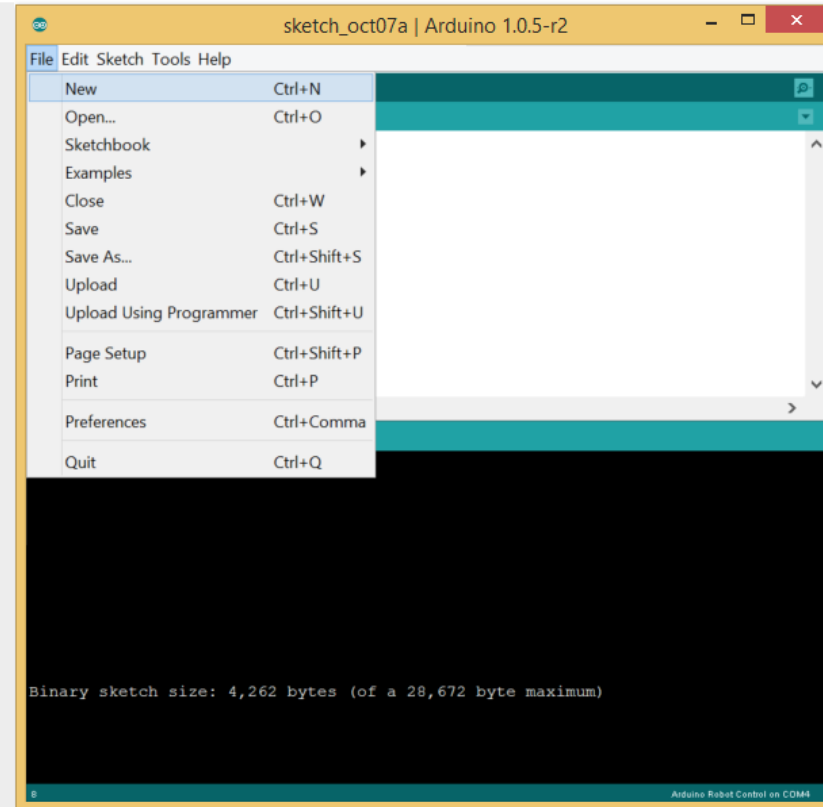
## Sketches



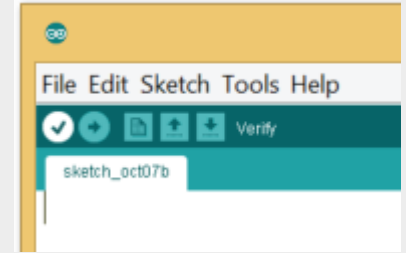
- An Arduino program is called a *sketch* and is stored in a folder with the same name as the sketch
- You can create a new Arduino program from the editor

## New Sketch

- To make a new Sketch just select New from the File menu
- This makes a brand new, empty, sketch
- You give the sketch a name when you save it



## Verifying a Sketch



- To verify a sketch you click the tick underneath the file
  - Or select Verify/Compile from the Sketch Menu
  - Or press CTR+R
- The environment will try to compile the program

## Compilation Errors

```
core.a(main.cpp.o): In function `main':  
C:\Program Files (x86)\Arduino\hardware\arduino\cores\robot/main.cpp:11:  
undefined reference to `setup'  
C:\Program Files (x86)\Arduino\hardware\arduino\cores\robot/main.cpp:14:  
undefined reference to `loop'
```

1

Arduino Robot Control on COM4

- If you try to verify an empty application you will get compilation errors
- The compiler will complain there is no setup function and no loop function

## setup and loop functions

- When an Arduino program starts the setup method is called to set up any hardware
- Then the loop method is repeatedly called to update the state of the embedded device
- This is performed by code in the Main method (which calls these methods)



## The Arduino main function

```
int main(void)
{
    setup();
    for (;;) {
        loop();
    }
    return 0;
}
```

- This is a simplified version of the Arduino main method that runs when the system starts

## The Arduino main function

```
int main(void)
{
    setup();
    for (;;) {
        loop();
    }
    return 0;
}
```

- Call the setup method to set up the hardware

## The Arduino main function

```
int main(void)
{
    setup();
    for (;;) {
        loop();
    }
    return 0;
}
```

- A for loop that runs for ever
  - They could have written “while(true)”

## The Arduino main function

```
int main(void)
{
    setup();
    for (;;) {
        loop();
    }
    return 0;
}
```

- Call the loop method to update the hardware

## The Arduino main function

```
int main(void)
{
    setup();
    for (;;) {
        loop();
    }
    return 0;
}
```

- This code is unreachable as the loop will never end

## Making an Arduino program

```
void setup()  
{  
}  
  
void loop()  
{  
}
```

- When we write an Arduino program we do it by filling in the setup and the loop methods

# Making an Arduino program

```
void setup()  
{  
}  
  
void loop()  
{  
}
```

- This is very similar to the way that games work

# The led flasher setup method

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

- This setup method configures pin 13 on the Arduino device as an output



## Arduino Connections

- The Arduino standard provides for a set of connections
  - Digital on/off signals
  - Analogue signals
  - Connections to other devices (sensors, displays etc)
- These are arranged as two sets of sockets on the device
- You can connect to the microcontroller just by pushing wires into the sockets



# Arduino pin numbering

- Each of the pins on the Arduino is numbered and the numbers map onto those used in the programs that you write
- You can find the corresponding pins on the different Arduino devices
- They are connected to the same ports on the microcontroller

## Arduino and libraries

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

- The `pinMode` method is part of the input/output library that ships with the Arduino SDK

## Arduino Libraries

- There are loads of libraries shipped with the Arduino

### Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

#### Structure

- `setup()`
- `loop()`

#### Control Structures

- `if`
- `if...else`
- `for`
- `switch case`
- `while`
- `do... while`
- `break`
- `continue`
- `return`

#### Variables

##### Constants

- `HIGH | LOW`
- `INPUT | OUTPUT | INPUT_PULLUP`
- `LED_BUILTIN`
- `true | false`
- `integer constants`
- `floating point constants`

##### Data Types

- `void`
- `boolean`
- `char`
- `unsigned char`

#### Functions

##### Digital I/O

- `pinMode()`
- `digitalWrite()`
- `digitalRead()`

##### Analog I/O

- `analogReference()`
- `analogRead()`
- `analogWrite()` - *PWM*

##### Due only

- `analogReadResolution()`
- `analogWriteResolution()`

- You can find a language and a library reference at:

<http://arduino.cc/en/Reference/HomePage>

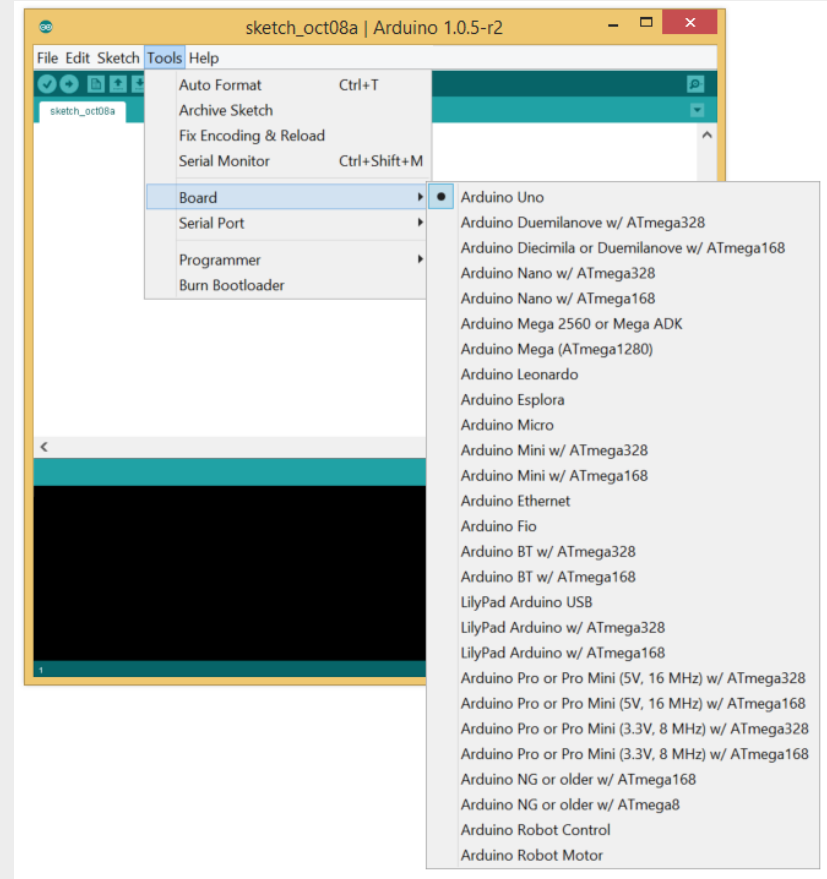
# The led flasher loop method

```
void loop() {  
  digitalWrite(led, HIGH);    // turn the LED on  
  delay(1000);                // wait for a second  
  digitalWrite(led, LOW);    // turn the LED off  
  delay(1000);                // wait for a second  
}
```

- The loop method flashes the led
  - digitalWrite and delay are library methods
- The loop method is called repeatedly when the device is running

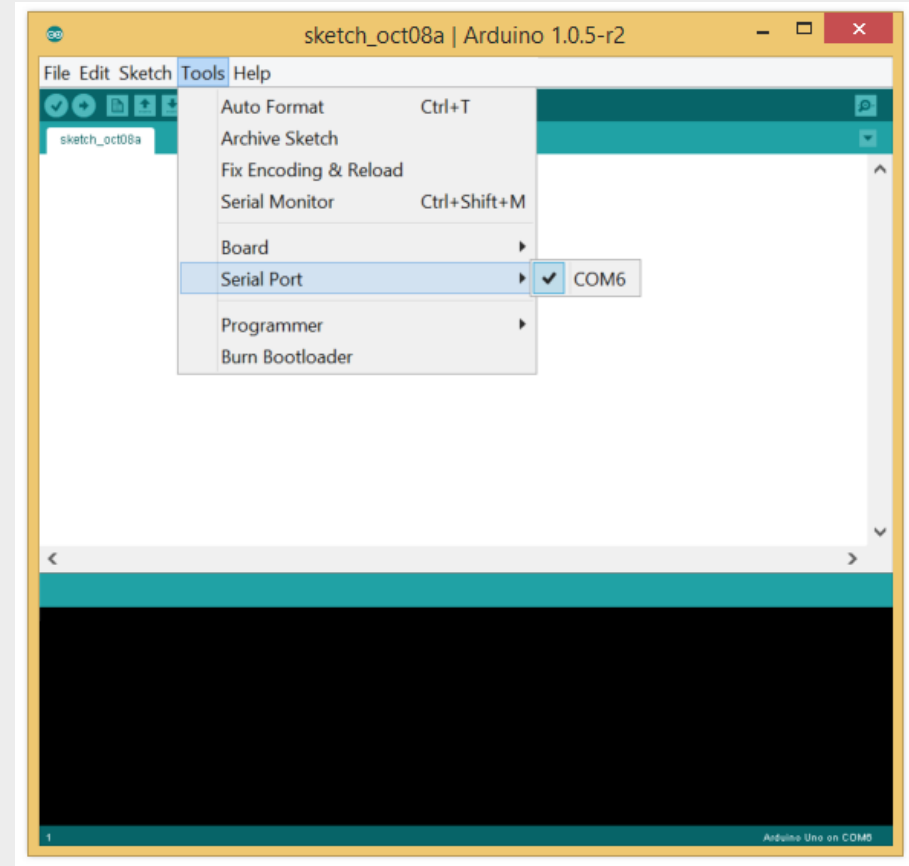
## Selecting a device

- There are lots of Arduino compatible devices
- You need to select the one that you are using when you deploy the program



## Selecting a port

- The Arduino connects to a serial port on your PC
- You need to select the correct one
- PC, Mac and Linux devices work slightly differently in this respect



# Rather Useful Seminars

## DEMO 01

Flashing a LED with an Arduino



## Debugging with the Serial Port

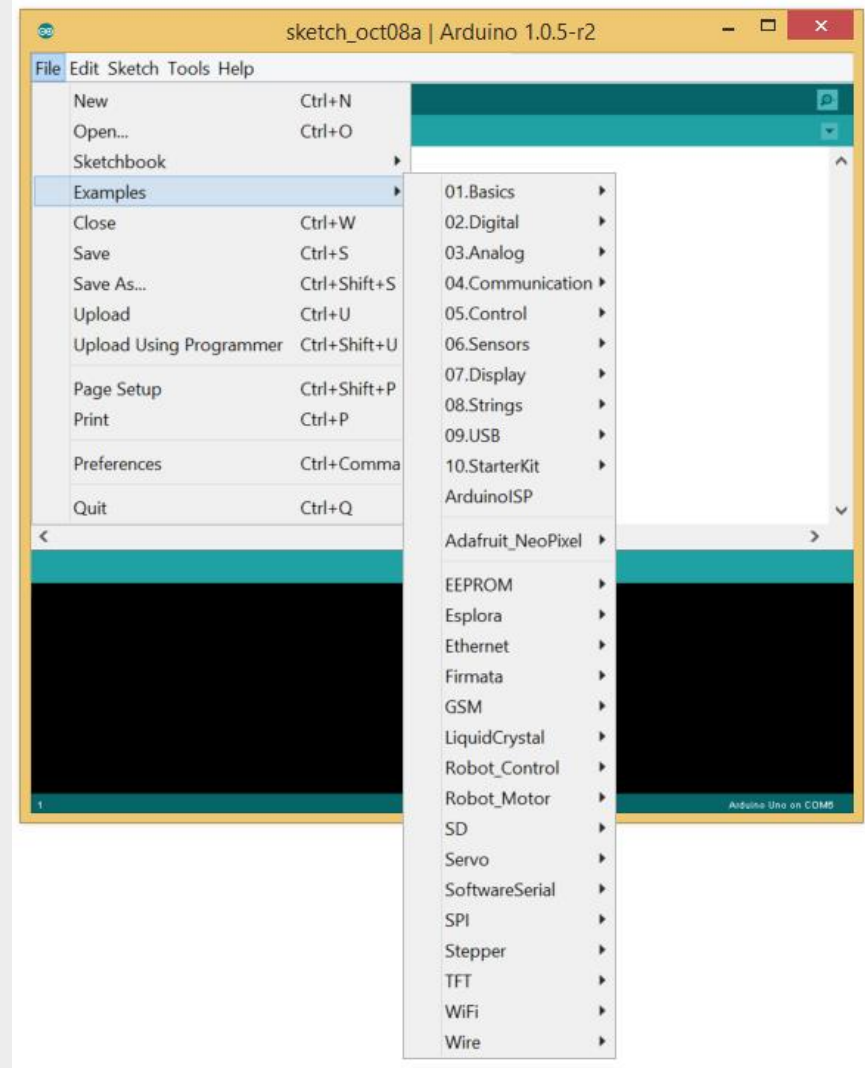
```
Serial.begin(9600); // open in setup
```

```
Serial.println("Flash"); // use in loop
```

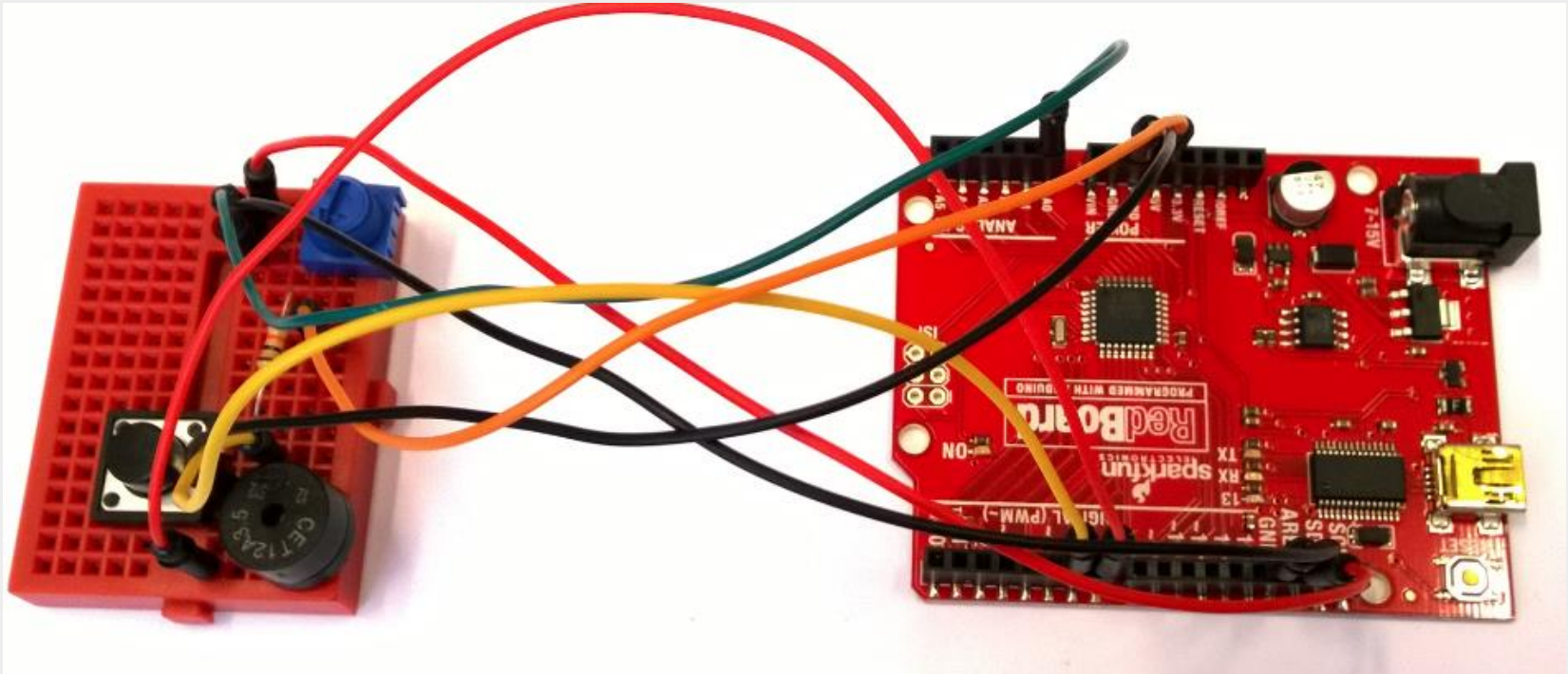
- You can use the serial port on the Arduino to allow your program to send messages back to you

## Sample code

- There are lots of sample programs
- For some of them you will need to plug in extra hardware



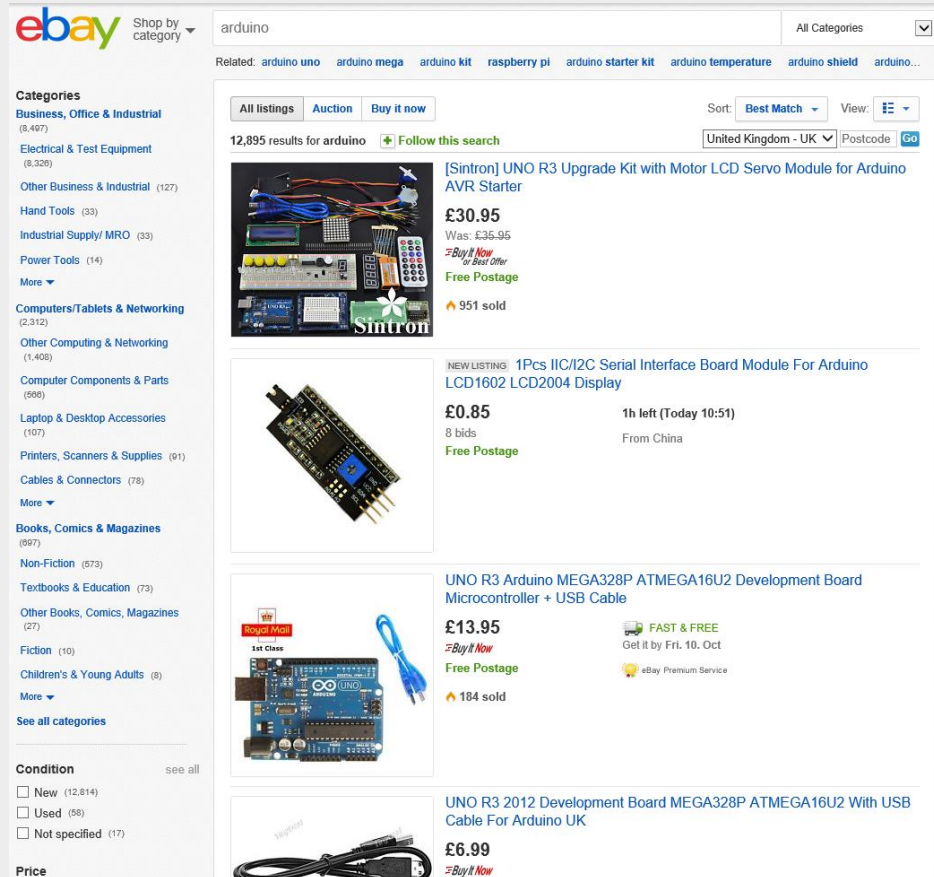
# Arduino Hardware Hacking



- You can make simple circuits using a “breadboard” and some jumper cables

## Arduino Stuff

- You can make really interesting things using Arduinos
- There are tons of cheap interfaces available on the internet



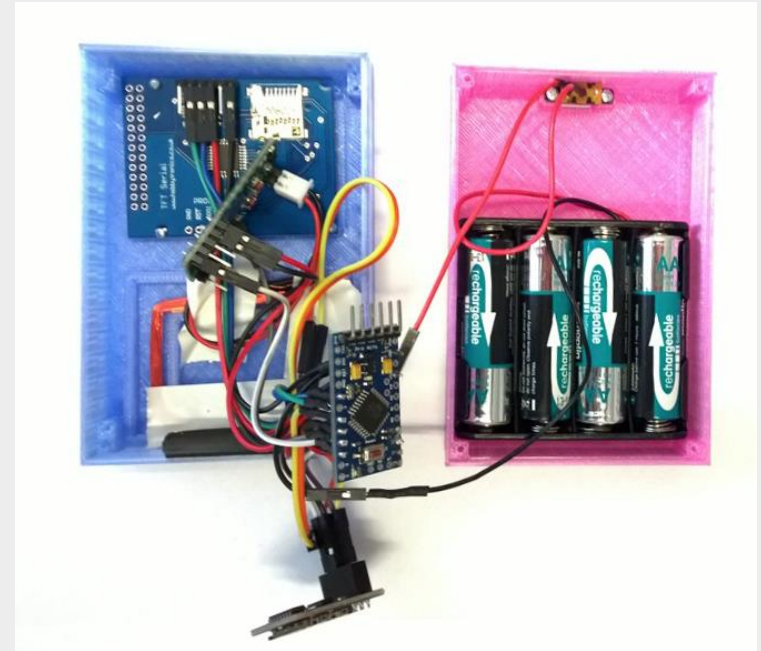
The screenshot shows an eBay search results page for 'arduino'. The search bar at the top contains 'arduino' and the 'Shop by category' dropdown is open. Below the search bar, there are navigation options like 'All listings', 'Auction', and 'Buy it now'. The search results are sorted by 'Best Match' and show 12,895 results. The first three results are:

- [Sintron] UNO R3 Upgrade Kit with Motor LCD Servo Module for Arduino AVR Starter**  
£30.95 (Was: £35.95)  
Free Postage  
951 sold
- NEW LISTING: 1Pcs IIC/I2C Serial Interface Board Module For Arduino LCD1602 LCD2004 Display**  
£0.85  
8 bids  
Free Postage  
1h left (Today 10:51)  
From China
- UNO R3 Arduino MEGA328P ATMEGA16U2 Development Board Microcontroller + USB Cable**  
£13.95 (1st Class)  
Free Postage  
184 sold

Additional filters on the left include 'Categories' (Business, Office & Industrial, Electrical & Test Equipment, etc.), 'Condition' (New, Used, Not specified), and 'Price'.

## Tags of Fun

- The Tag Reader that we used at the Welcome Party last week was powered by an Arduino
  - Arduino Pro-Mini processor
  - RFID tag reader
  - SD card reader
  - Colour LCD panel

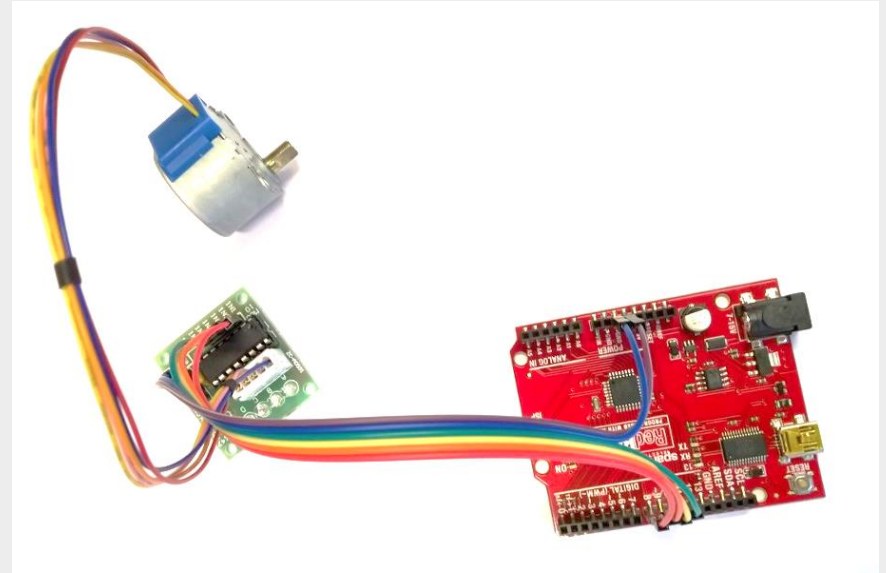


## Tags of Fun

- It was surprisingly cheap to make:
  - Arduino Pro-Mini processor    £5.50
  - RFID tag reader    £5.00
  - SD card reader    £5.00
  - Colour LCD panel    £24.00 (oops)
- I've since found a place in China that does LCD panels for around £3.00 .....



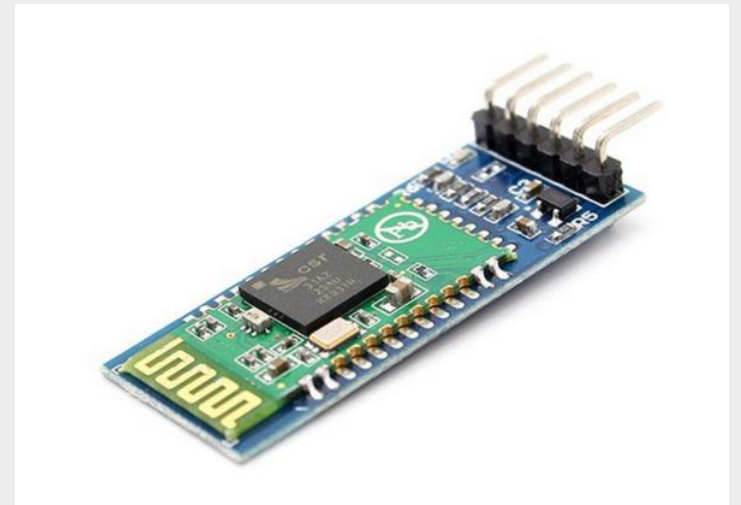
## Moving Things



- This stepper motor and driver board costs £1.50 from China
- You can control the movement of the motor output very precisely

## Connected Things

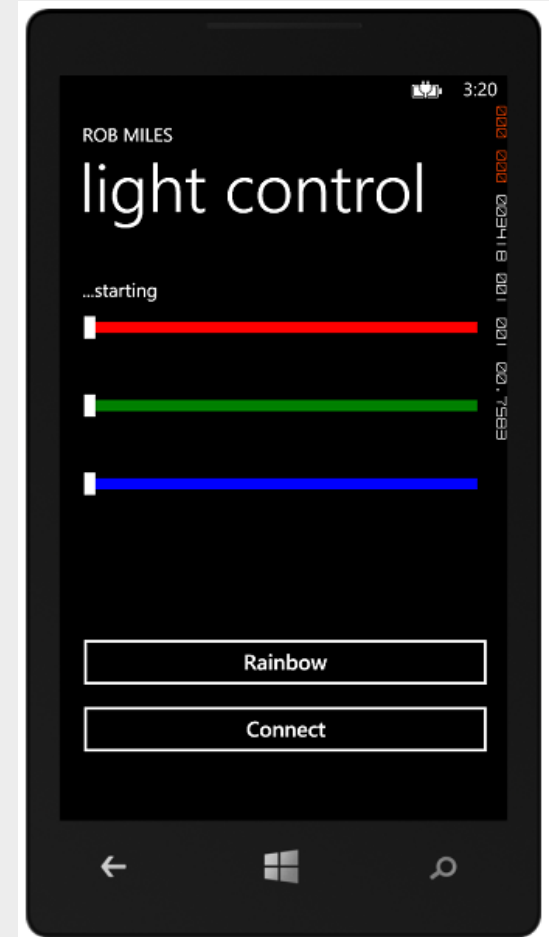
- For around £5.00 (there's that price again) you can get a Bluetooth adapter
- This lets you connect your Arduino devices to Smartphones and PCs very easily
- You can also use WiFi and 2.6GHz devices





## Connected Applications

- I'm using the Bluetooth adapter on the device to talk to my Windows Phone
- That means that I can now create mobile, connected systems that I can carry round in my pocket
- Which is rather nice



# Rather Useful Seminars

## DEMO 02

Wedding Lights

## Resources

- Arduino site:  
[arduino.cc](http://arduino.cc)
- C4DI Arduino pages:  
[www.robmiles.com/arduino](http://www.robmiles.com/arduino)
- These slides:  
[www.ratherusefulseminars.com](http://www.ratherusefulseminars.com)

Also available....

[www.robmiles.com](http://www.robmiles.com)