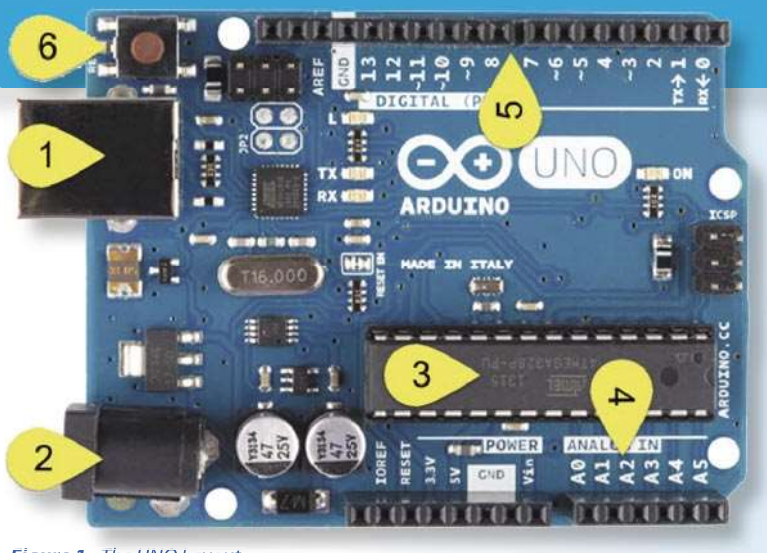

Figure 1 - The UNO Layout.

Interfacing with

Using Arduino within Engineering Science SQA Courses

National 5 Engineering Science requires learners to give examples of the use of microcontrollers in commercial and industrial applications, also the advantages/disadvantages of microcontroller based systems compared with hard wired electronic equivalent circuits.

Within Higher Engineering Science, Arduino (also PICAXE and STAMP) are all suitable platforms for developing electronic control systems (Unit Support Notes, SQA, Page 37). Typical programmable control systems will involve up to four inputs and four output signals. The controlling of the speed of a d.c. motor using pulse width modulation is also required.

Advanced Higher Engineering Science further develops programmable systems by introducing interfacing microcontrollers. Principles and applications of A-D and D-A conversion are also included.

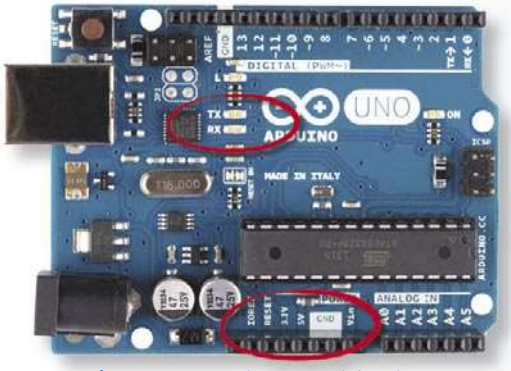

Figure 2 - Power Pins, transmit/receive indicator lights.

Introduction

As there are a number of Arduino boards available, selection can be difficult for the beginner. Perhaps the best 'starter' board is the Arduino UNO (Figure 1), it's simple and relatively straight forward to use and is a great 'springboard' for further interfacing with other boards in the Arduino family. UNO boards are currently priced at around £25. UNO has been available since September 2010.

The UNO interfacing board is connected to a computer via an 'A to B' USB cable The 'B' connection is shown at 'Ref 1' in Figure 1. Computers running Windows, MacIntosh, and Linux operating systems can be used with Arduino boards.

The Arduino board obtains power via the USB cable, but it also has a power input socket, as indicated by 'Ref 2' in Figure 1, this allows the board to operate independent of the computer, an AC adapter, 9 volts, 2.1 mm barrel tip, with the centre positive is recommended. Arduino can also be battery powered.

'Ref 5' as indicated in Figure 1 shows 14 digital input/output pins 0–13. These pins can be programmed for either digital input or output. Six of these 'digital' pins (3, 5, 6, 9, 10 and 11) also have another function, they can be programmed as Analogue Outputs.

Six dedicated analogue pins (Ref 4 in Figure 1) will take voltage readings from sensors and convert the voltage into values in the range 0–1023.

A reset button is indicated in reference 6 in Figure 1.

Introduction to programming using IDE (Integrated Development Environment)

Checking the communication port

With the Arduino board connected via an USB cable, it is important that the computer and the Arduino board are set up for correct communication. Using, in this case a Windows 7 platform, click 'Start' and right click over 'Computer', now left click 'Manage' as shown in Figure 3.

Figure 3 - After right click on 'Computer' select 'Manage'.

When 'manage' is selected, Figure 4 is displayed. Now select 'Device Manager'.

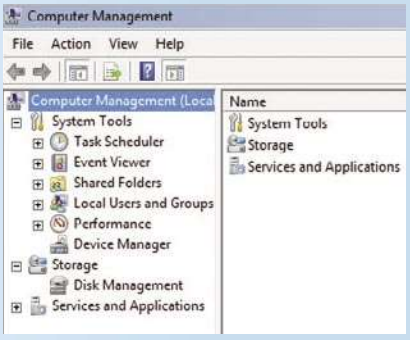

Figure 4 - Selection of Device Manager.

the Arduino UNO platform (part 1)

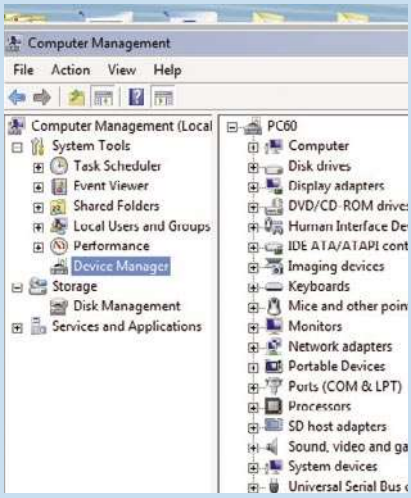

Figure 5 *- Device Manager selected.*

Figure 6 *- Arduino board Connection.*

Select 'Ports' as shown in Figure 5.

Now click Ports. You should see the port where Arduino Uno is connected (COM3) and that the computer is recognising 'Arduino'

Now, the Arduino board must be set up with COM3.

Figure 7

Double click the Arduino icon on the desktop as shown in Figure 7. If the Arduino software has not yet been downloaded, refer to the note at the end of this article.

From the IDE screen, select 'Tools' Click over 'Serial Port' COM3 should be selected, as shown in Figure 8. The computer and Arduino board are now able to communicate with one another.

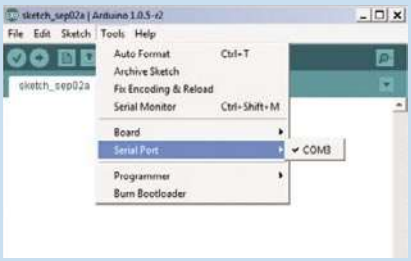

Figure 8 *- The IDE.*

The IIDE (Integrated Development Environment), is an editor which allows a program, called a 'Sketch' to be generated. The IDE is shown in Figure 8.

Arduino provides open source hardware and software. At the computer, software called the Integrated Development Environment (IDE) allows code to be written for Arduino. A 'Sketch' is the IDE code (a program), the sketch is then uploaded to the UNO board. The Sketch code requires to be compiled and finally converted into the form of code the Arduino microprocessor (Ref 3, Figure 1) can work with - this is an automatic process.

Figure 9 shows all the pull down menus available via the IDE (integrated Development Environment).

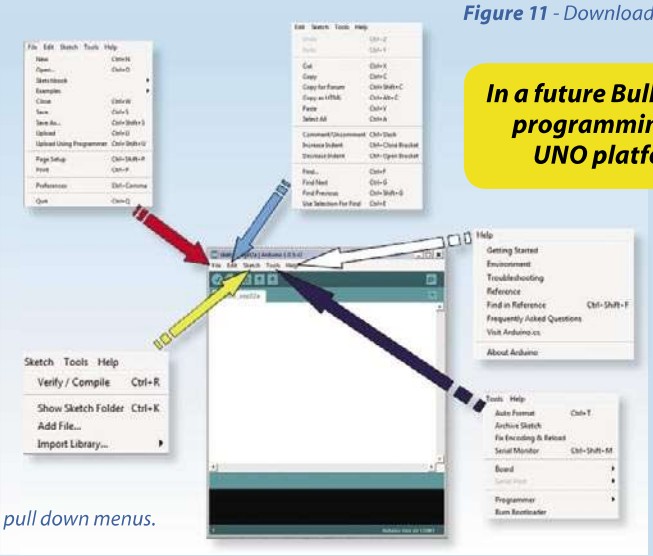

Figure 9 *- The IDE and associated pull down menus.*

Note:
Downloading Arduino IDE
Download from the following website http://arduino.cc/
The following website should be displayed (Figure 10).

Using a free 'Download' as indicated in Figure 10, scroll down until Figure 11 is displayed. Select Windows installer. The download process should be automatic resulting in the Arduino icon (Figure 7) displayed on desktop.

Figure 10 *- Arduino website.*

Figure 11 *- Download options.*

In a future Bulletin, Interfacing programming the Arduino UNO platform (part 2).