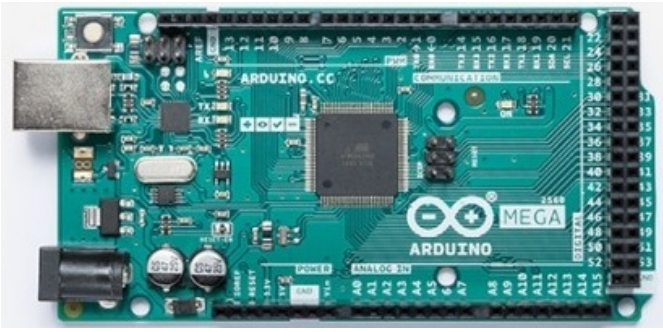
Use of Arduinos in model aircraft

What is an Arduino? It is a small, cheap computer board intended for reading sensors and controlling things, for example with motors. It was designed by a group in Italy (https://www.arduino.cc) to get people using computers in a more creative way. There is a set of these boards ranging in size from the Nano (18 x 45 mm and 7g) to the Mega (53 x102 mm and 37g). They differ mostly in the number of inputs and outputs. All are programmed using free software running on a personal computer. The whole project is Open Source, which means that the software is free and users share their programs and designs online. Don't worry if you have never written program code. There is a huge range of ready written code for just about any job you might want to do. You soon learn how to adapt the code if you really need to. It's all part of the fun.

Here is the smallest and largest (not to scale):



How does it work?

An arduino has several inputs of two types. These are pins onto which you can put voltages:

- Digital voltages, from a switch or other device, which have one of two values (0 and 1), for example 0V and 5V.
- Analogue voltages from sensors detecting such things such as light, sound, temperature, pressure, potentiometer voltages etc. These can have any value between say 0V to 5V. The Arduino digitises them, which means it measures them and gives the value a binary number, for example between 0 for 0V and 1023 for 5V In binary these are 0000000000 and 1111111111.

It also has several output pins from which signals may be sent:

- Digital outputs give 1 and 0 in the form of a voltage, for example 0 or 5V. These could be used for switching lights.
- Pulse width modulation outputs allow you to create varying signals. For example an on-off voltage could be used to drive a motor at different speeds, or a lamp at different brightnesses, by varying how long the signal is on rather than off (mark-space ratio). You could make sounds by sending varying signals to a loudspeaker. You can create the same servo signals that our receivers produce.

You either use standard code or write your own on your computer. You then send it to the Arduino through a lead. The chunks of code are called sketches.

How might we use one in a model?

Several (not multiple – yuk!) ideas spring immediately to mind.

An Arduino can create the pulse width modulation signals that vary from 1 to 2 milliseconds in length, to operate our servos. They can drive low power servos directly but might need an additional board to boost the current for larger ones. If you have a scale model with complicated undercarriage doors and mechanisms, you could build a sequencer that drives the door servos and retracts at a chosen speed and in the order you want. The Arduino would read a start signal from a receiver channel and then go into its retract sequences.

You could operate landing lights and steady or flashing navigation lights.

For rubber powered free flight models you could build an electric winder that would count the turns on a stepper motor. Yes, I have one of these in the design stage.

Free flight F1a gliders are released at speed from a 50m towline and follow a vertical S-shaped path of half a loop and half a bunt. They can gain up to another 50m in this way. The Arduino could control the elevator servo to do this without breaking the competition rules.

How to get started

You can buy a board, a power supply, a USB lead and a set of components for about £35, for example on eBay. You never know you might get hooked on these control systems and start building all kinds of clever things. It is probably best to start with the mid-sized Uno.

Uno board	£4.98
5V power supply	£5.19
USB lead	£2.50
Sensor kit (37 bits)	£11.99
Stepper motor	£3.69
Wires	£5.00