

Arduino for Art, Design & Technology – Project Book Notes

Paul Wilby, Head of Art, Design and Technology, [English Martyrs Catholic School](#)

These are my teaching notes on the “Arduino Projects Book”. This book comes (in hard copy) with [the starter pack](#) and provides a useful introduction to a wide range of applications for the Arduino in projects. However as I worked my way through the circuit construction and the programs I noted a number of pitfalls that could cause someone new to programming in C (like me) to give up.

General points – Punctuation

An important part of programming in C is the use of brackets.

Curly brackets { }

denote a loop function. If you click on a { the corresponding } will be highlighted by a box around it. I found it useful to go through a program clicking on each bracket and looking for the corresponding one to be highlighted. This helped me to make sure that I hadn't missed out any brackets.

Plain brackets () denotes where information needed by the program could be found for example whether a pin is an input or an output or what the state of the pin is (high or low). They are also used for comparisons such as less than, more than or equal to arguments. Again it was useful to click on each bracket and make sure that the corresponding open or close bracket had been included.

Square brackets [] are used to denote an array of numbers or information where the program looks up the position of the number in the array rather than the number itself. Project 07 – The Keyboard Instrument uses this function to denote musical frequencies and rather than having to keep entering the frequency each time the program looks at the position number of the array to obtain the required value. Similarly, as with other brackets each one can be clicked on to discover the corresponding bracket.

The semicolon ; is used after values or plain brackets the only exception being if the bracket is before a curly bracket. I have found it is very easy to miss out the semi colon and the program, frustratingly, will refuse to compile and will give you an error message telling you are missing one.

The use of **the decimal point .** caused an issue in the case of the ‘float’ instruction. This refers to a floating decimal point number rather than an integer (given by the ‘int’ instruction). When using the ‘float’ instruction any numbers, even if they are integers, must include the decimal point (for an example look at Project 3 – Love-O-Meter on page 49)

Speech marks “ ” are used along with the Serial.print instruction to print on screen the state of inputs, outputs or values.

The use of != means not equal to.



Notes on the projects

As I worked my way through each of the projects in the starter pack I came across some issues and things that I needed to do in order to get the circuit to work. I think it might be worth sharing these.

On a general note, while the book covers a fair amount of basic electronics and circuits I would advise a little caution. Some information is over generalised and some misleading. For example, on page 30 Ohm's law is introduced and used to calculate the value of the bias resistor for the LED. The calculation ignores the voltage drop across the LED itself and so with higher voltages may give a false value for the resistor that could end up damaging the LED. The use of the term "amperage" for current is annoying.

Project 03 The Love-O-Meter

This project introduces the TMP36 temperature sensor. This component produces a linear output temperature reading unlike many thermistors, 10mV change of output represents a temperature change of 1C. Note the need for the offset to read from 0C. The device can measure from -50C and will start from this temperature if the offset is not included in line 20.

Notice in line 15 of the code the 'float' instruction needs the decimal point included. This line results in voltage changes of around 5mV to be detected and acted upon this means that the device measures 0.5C steps.

The project uses analogue to digital conversion (ADC)

Note the need for a short delay when using ADC.

Project 04 Colour Mixing Lamp

This project introduces light dependant resistors and multicolour LED. It also includes the use of pulse width modulation (PWM). Output on the board marked with symbol ~ can produce a PWM output. This function is useful for motor speed control and light dimming.

Project 05 Mood Cue

This project introduces a variable resistor used as a potentiometer (ie all three pins connected) and a servo motor (a motor that uses gearing to move its axle through an angle of rotation up to a maximum of 180 degrees)

The project introduces 'libraries' or pre-set information that someone has compiled to convert output voltages to servo output angle.

The servo included in the pack has a three pin female plug attached to the lead. In order to connect this to the breadboard it comes supplied with a strip of header pins, I found this solution doesn't work so used some jump leads to make the connection. A better solution is to cut off the plug and solder header pins directly to the three wires from the servo motor.

I was very impressed by the accuracy and control of the servo. One turn of the potentiometer moved the servo almost 180 degrees and half a turn of the potentiometer produced a 90 degree turn.



For the fun of it I replaced the variable resistor with a light dependant resistor and a 10K resistor as a potential divider. The top of a bic biro with its small hole at the top acted as a good mask and allowed for slight changes in shadow to operate the servo.

Project 06 Light Theremin

This project introduces the 'tone' function and discusses the difference between the constant 50% duty cycle of the tone function and the variable duty cycle produced by the 'analogWrite' function. The calibration routine used to identify the most to least shadow and modify the program to accommodate different depths of shadow is an interesting and clever part of the program.

This program worked well and again the bic pen top worked well to give a good variation of shadow to obtain a full sound range.

Project 07 Keyboard Instrument

This project is a good example of an array. The frequencies of four musical notes is given in the array and the array uses the position of the notes in the array to identify the note rather than having to keep entering the note frequency in the 'tone' instruction. A resistor ladder is used to produce differing voltages that are converted to frequencies when the buttons are pressed.

In figure 1 the resistors are connected in series, pressing more than one button doesn't change the resistance the analogue port "sees" the voltage over the resistor nearest the junction to the analogue port.

The layout of the breadboard shown in figure 2 is the circuit shown in figure 3. This circuit does change the voltage "seen" by the analogue port when more than one button is pressed. However the values of the resistance chosen don't make significant changes. It is worth using the resistors in parallel formula to work out the voltages when a combination of buttons is pressed.

In the example array on page 83 line 2 generates a conflict error. It isn't needed so leave these two lines out.

Project 08 Digital Hourglass

In this project the timing function "millis" is introduced. This tracks how long the Arduino has been running in milliseconds. It makes the distinction between the "millis" function and the delay function. The delay function freezes the board's state until the delay is over, this means it can't take in information while the delay is going on.

The project describes how to extend the information storage to a 32 bit number by using the "long" function.

The code shows how to compare time and give an accurate time interval for timing events. The time interval for 10 minutes (600 000 milliseconds) is given in line 6. I got bored waiting the 10 minutes so changed it to 10 seconds (10 000 milliseconds)

The circuit uses a tilt switch.

Project 09 Motorized pinwheel



This project is good as it shows how to control motors using an external power source by using transistors, in this case a MOSFET. It also shows the need for a diode to remove damaging voltages caused by back emf from the motor.

Project 10 Zoetrope

This project takes some building. It builds on the pinwheel in project 09. It is a nice project to build and try out. The motor supplied is good quality and it can really spin the wheel. On a safety note it can easily fly off. The wheel can also launch cd's so be careful if using with students, good fun but may need safety specs just in case.

The potentiometer supplied keeps popping off the breadboard and then the speed increases. When held down it does have a good range of speed. The circuit does use a very useful IC called a H Bridge this makes it easier to reverse the motor direction and removes the need for including additional diodes to eliminate the back emf from the motor.

The code doesn't introduce any new functions and is fairly easy to follow. There are though a lot of lines of code and it easy to miss appropriate brackets, semicolons and capital letters (if used). It took me four attempts to fill in or correct code.

