University of Nottingham
UK | CHINA | MALAYSIA

Open Day: The NEOBoard

The Nottingham Electronics Outreach Board

Department of Electrical & Electronic Engineering

Table of Contents

1. Introduction

Welcome to the Department of Electrical and Electronic Engineering's Open Day. During this activity, you will make use of one of our project labs to complete a short design task to give you a sample of what we do here on our degree course at Nottingham.

You will be constructing a microcontroller-driven 'electronic dice', aptly called the Nottingham Electronics Outreach Board, or NEOBoard for short. Once built, you will be able to randomly 'roll' different-sized dice (d4, d6, d8, d10, d12, d20 and d100), displaying the result on a seven-segment display. You can also make use of the display, input buttons and speaker for a range of custom projects in the future.

2. The NEOBoard Activity Overview

The aim of the activity is for you to construct an electronic dice which makes use of different components to display a random number and play a dice roll round. Figure 1 shows a completed NEOBoard, which is what you will be able to take home with you at the end of the activity.

Figure 1: The Complete NEOBoard

In front of you, you should have a parts kit, as shown in Figure 2. This contains several different components as well as a partially populated NEOBoard.

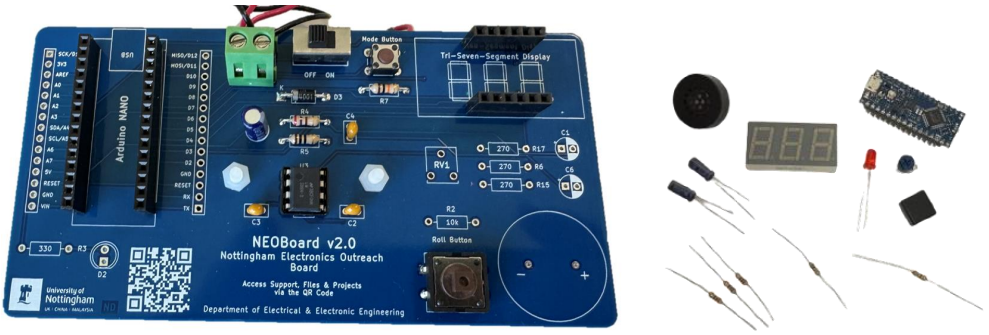


Figure 2: The NEOBoard Parts Kit

2.1 The NEOBoard Parts Checklist

Check that you have the following components, if you are missing any, please let us know.

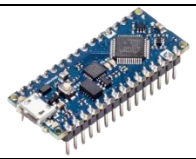
Item Description	Quantity	Notes	Photo
Partially Populated NEOBoard PCB	1	With the Battery Pack & Snap Lead Attached. The LM386 Audio Amplifier Chip Should Already be in the Socket - Details on How to Correctly Identify the Polarity is Included in '3.4 The Speaker & Audio Amplifier Circuit'	
Arduino NANO Every Microcontroller	1		
USB A-Micro Cable	1	Used to Power & Program the Arduino NANO Every Microcontroller	
LED	1	Colours May Vary Between Red, Green & Yellow. Details on How to Correctly Identify the Polarity is Included in '3.6 The LED'	
Tri-Seven Segment Display	1		
270Ω Resistor	3	Details on How to Identify the Values of the Resistor are Included in '3.5 The Resistor'. The Image is Illustrative Only	
330Ω Resistor	1		
10kΩ Resistor	1		
10uF Capacitor	2	Details on How to Correctly Identify the Polarity is Included in '3.2 The Capacitor'	
Speaker	1		
10kΩ Trimpot	1		
Push Button (Large)	3		
Push Button (Large) Cap	3	Colours May Vary Between Red, Green & Black – These Clip onto the Top of Large Push Button	

Table 1: NEOBoard Parts Checklist

2.2 Visual Checklist

Your task is to finish populating the NEOBoard by soldering the remaining components into the correct locations. If you have not soldered before, do not worry! Please ask one of the staff or student ambassadors for a demonstration as well as reading through *'4. Quick Guide to Soldering'.*

Figure 3 shows the partially populated board, as well as markers for each of the component locations. Tick each one off once you have soldered it in place to ensure you haven't missed anything, including pushing the seven-segment display into the headers!

Once fully constructed, you will upload the example code following the process described in *'5. Programming the NEOBoard'.*

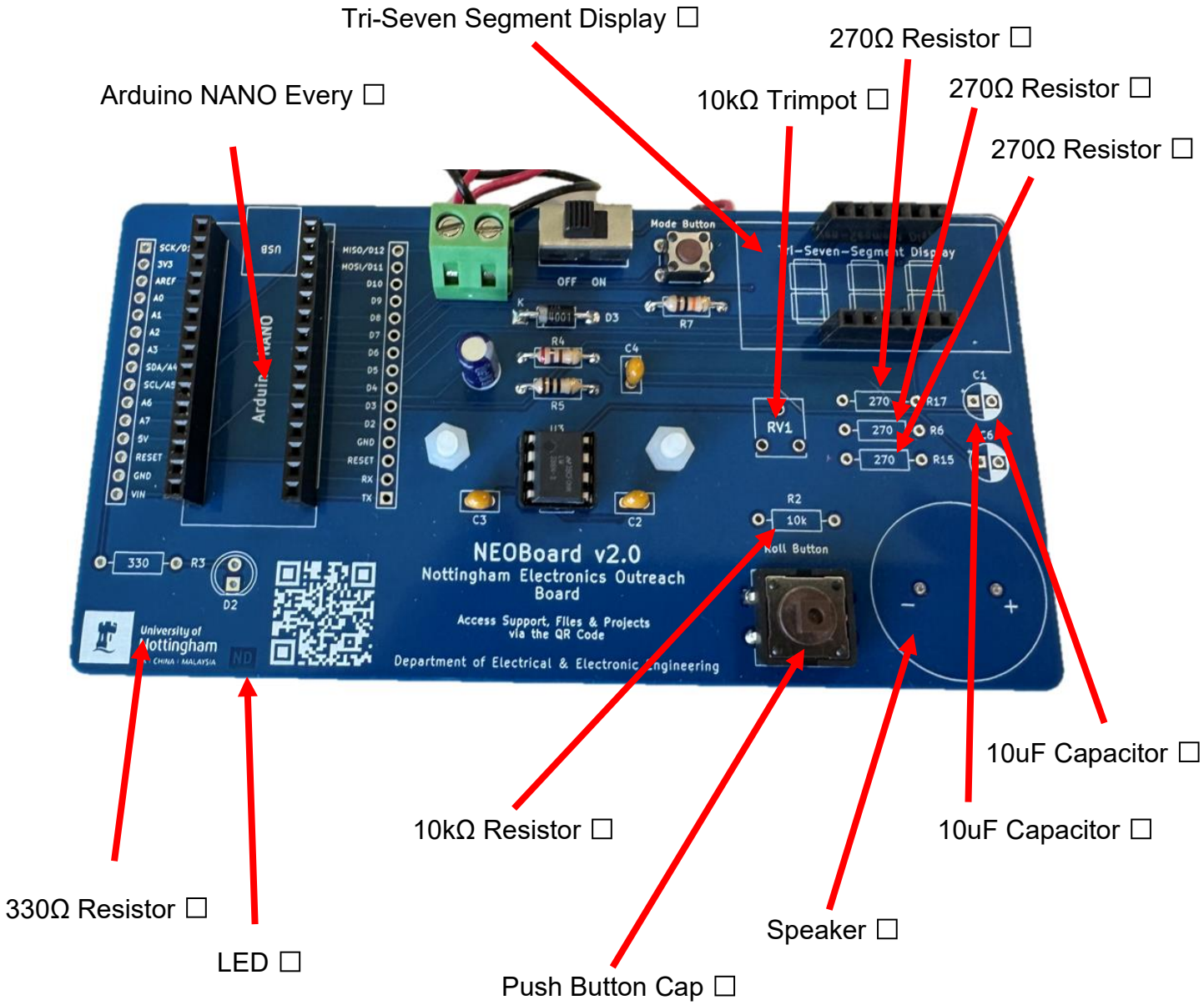


Figure 3: Partially Populated NEOBoard & Component Checklist

3. Breaking Down the NEOBoard

This section will break down each of the key elements of the NEOBoard. You do not need to read and understand this to do the activity, but the following information will give you an insight into what each component does, and how it comes together into a working system. In the resistor, capacitor and LED subsections, there is also useful information regarding how to identify a resistor's value and identify the polarity of a capacitor and LED respectively.

3.1 The Seven-Segment Display

Within the NEOBoard, a tri-seven-segment display is used to display the 'rolled' number, as shown in Figure 4 (left) below. A seven-segment display gets its name from the seven individually illuminated elements, labelled a-g with a decimal point, as shown in Figure 4 (right). The 'tri' term refers to the three 'digit' displays on the module.

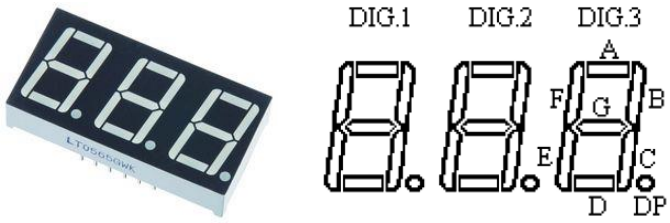


Figure 4: Tri-Seven Segment Display (left) & Seven-Segment Labelling (right)

The specific tri-seven segment display module used is common cathode, meaning that the cathodes (i.e. negative pin) of all segments in each digit are connected together. The anode (i.e. positive pin) for all of the seven segments per display are then controlled via the microcontroller pins by applying a positive voltage. Seven segment displays also come in a common anode variety with a similar operating principle but share a common positive pin, with segments illuminated with a zero voltage on the negative pins.

If this were a one-digit display, then the common cathode would be connected directly to ground. However as there are three digits, then in order to control each digit separately, the common cathode pin for each digit is connected to a separate GPIO pin on the microcontroller. To illuminate a particular segment on a particular digit, then the anode pin is set 'HIGH' and the relevant digit's common cathode pin is set to LOW, where the default state is 'HIGH' so these aren't illuminated.

3.2 The Capacitor

Within the NEOBoard, there are three 10uF capacitors, two of which you must solder. Capacitors are used to store electrical charge and, in this case, are used as part of the audio amplifier circuit described in '3.4 The Speaker & Audio Amplifier Circuit'.

Figure 5: Capacitor

The value of a capacitor can be read from the side of the casing. A capacitor does have polarity. The negative side can be identified by the light shading on the casing and the shorter component leg. This shading must match the PCB footprint shading for correct orientation.

3.3 The Push Button

Within the NEOBoard, there are two sizes of push button, large and small, shown in Figure 6 left and right, one to trigger a ‘roll’ and the other to switch between the different dice sizes respectively. The large button has a button cap of varying colours so that pushing it is easier.

Figure 6: Push Button; Large (left) & Small (right)

A push button is a simple mechanical switch that completes or breaks a circuit when it is pressed. In our case, a circuit is completed when the button is pressed, and so a high signal is read by the microcontroller. This high signal is only active whilst the button is pressed, and returns to low when the button is released.

3.4 The Speaker & Audio Amplifier Circuit

Within the NEOBoard, there is a speaker which is used to audibly play sounds. To amplify the audio signals, an audio amplifier circuit, driven by the LM386 operational amplifier chip, is also used, as shown in Figure 7 below. Take note of the + and – indicators on the speaker casing to correctly orientate it in the NEOBoard.

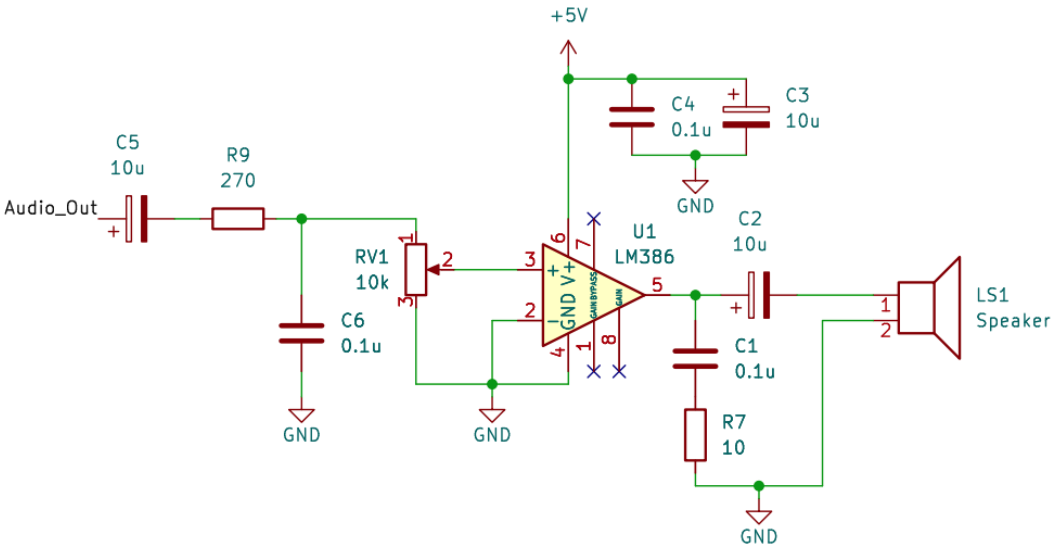


Figure 7: LM386 Audio Amplifier Circuit

Whilst the full breakdown of how this circuit works is not included here, to summarise, the key stages are as follows:

1. The audio signal is inputted to the non-inverting (+) pin of the LM386 through a variable resistor. Without this resistor, there will be a lot of noise in our audio output.
2. The gain, i.e. how much we multiply the input by to get our output, is set as the chip default of 20, so pins 1 and 8 are left unconnected.
3. There are two decoupling capacitors used on pin 6 to filter out any low-frequency noise from the power supply which otherwise would have been amplified and distorted our audio output.
4. Once amplified, the audio is outputted from pin 5 and fed into the speaker, with additional capacitors used for filtering and smoothing.

Polarity of the LM386 Audio Amplifier Chip

As part of your parts kit, you have been given an LM386 audio amplifier chip. Whilst this can only go in the black 8-way socket, it is vital that you push the chip into the socket the correct way around.

You will notice that at the top of the socket, there is a notch which indicates where the top of the chip should be placed. On the chip itself, there is a dot – this indicates the top of the chip. Therefore, the side of the chip with the dot on should match the notched side of the socket, as shown in Figure 8 below. If you are unsure, please ask.

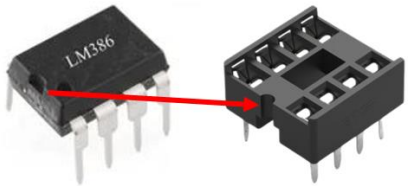


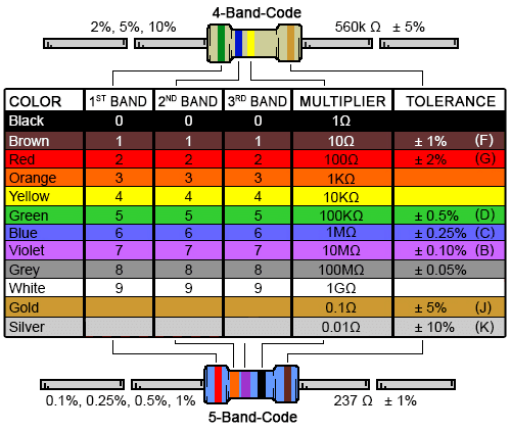
Figure 8: Correct Chip & Socket Orientation & Placement

3.5 The Resistor

Within the NEOBoard, there are a number of resistors of different resistance values, which have different purposes. These include; pull-down resistors to keep a signal low when the button isn't pressed, current limiting resistors to prevent the LED/seven-segment display from being damaged and as part of the audio amplifier circuit to filter the audio signals.

Resistors, as shown in Figure 9 (left), are the most common and well-known passive electrical components. A resistor resists and limits the flow of electricity i.e. current, in a circuit, and is measured in Ohms (Ω).

There are two ways in which we can determine the resistance of a specific resistor; the first is with a multimeter, which we won't use, and the second is with the resistor colour band chart, as shown in Figure 9 (right).



COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8	100MΩ	± 0.05%
White	9	9	9	1GΩ	
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)

Figure 9: Example of Through-Hole Resistors (left) & Resistor Colour Band Chart (right)

Within your parts kit, you have five resistors – three of which are 270Ω and one of which is 330Ω, used as current limiting resistors for the seven segment and LED. The final one is10kΩ, which is used as a pull-down resistor for the button.

- 270Ω; the bands are: Red-Violet-Brown-Gold.
- 330Ω; the bands are: Orange-Orange-Brown-Gold or Orange-Orange-Black-Black-Gold.

- 10kΩ; the bands are: Brown-Black-Orange-Gold or Brown-Black-Black-Red-Gold.

A note about resistor placement on the NEOBoard is that they have no polarity. This means that they can be put into the PCB in any orientation and will function the same i.e. provide the same amount of resistance.

3.6 The Light Emitting Diode (LED)

Within the NEOBoard, there is one LED, which comes in varying colours, as shown in Figure 10 (left), which is used to indicate power.

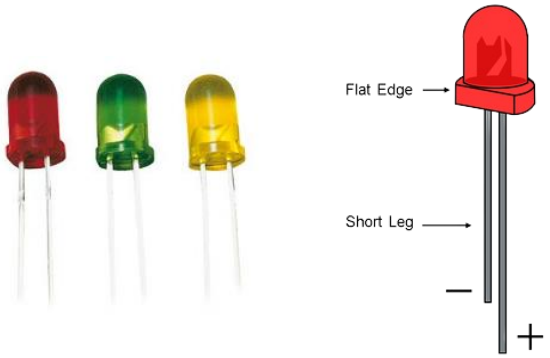


Figure 10: Light-Emitting Diodes (left) & Polarity Indicators (right)

Whilst a diode is a semiconductor device that only allows current to flow in one direction, an LED differs from a typical diode by emitting light when a current is passed through it. This means that it has polarity and must be carefully orientated in the NEOBoard to ensure correct operation. Typically, two observations can be used to identify the positive and negative sides of an LED, shown in Figure 10 (right). The first is that the two legs of an LED vary in length, with the longer being the anode i.e. positive, whilst the negative, i.e. cathode, is shown by the shorter leg. The second method is that on the base of the LED, there is a flat edge. This indicates the negative side and can be used when the LED has been soldered and the component legs already trimmed.

4. Quick Guide to Soldering

You do not need to have done any soldering before this activity. In the lab today there is lots of support available from both staff and current students – feel free to ask for help if you would like additional help or a demonstration of soldering.

This section is to act as a quick guide to soldering for the first time, but this is a skill that will quickly improve with practice.

It is important to remember that soldering performs two main tasks; these being:

- Providing an electrical connection in a circuit.
- Fixing a component to the circuit board, also known as a printed circuit board (PCB).

Therefore, when we solder, we need to ensure that not only is a good ‘clean’ solder joint made, but also that the component will remain fixed in place. For example, consider the problems if components were to ‘fall off’ from a board used in an industrial robot or your laptop/phone.

When soldering, we use a number of basic tools, including:

		
Soldering Iron	Soldering Iron Station & Extractor Fan	Solder
		
Goggles	Wire Cutters	Desoldering 'Solder Sucker' Pump

Table 2: Common Tools Used in Soldering

Before we start, it is always best to check your soldering iron to ensure that:

- The electrical cable is of a high standard, with no damage.
- The 'PAT' safety test is up to date.
- The solder tip is of a high standard i.e. not bent and is shiny.

If your soldering iron isn't up to scratch, let us know and we will replace it with a new one.

4.1 Preparation

The first task is to get the soldering iron warmed up, which can take a minute or two – the temperature should settle on either 375°C or 400°C depending on the solder station preset. Once the soldering iron is heated, wipe it on the gold wire wool to remove any oxidation – this will show as a dark residue. If it does not come away easily you may need to use tip cleaner which we can show you how to use if needed.

You now need to apply a small amount of solder to 'tin the tip'. The aim here is to give a shiny layer of solder as shown in Figure 11 (right).

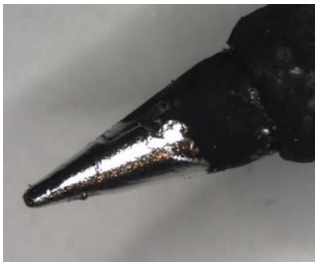



Figure 11: Wire Wool (left) & the Tip when 'Tinned' with Solder (right)

Now it is time to place the component in the board. It is recommended you read to the end of this section before soldering any components into the NEOBoard. When doing this, ensure that the component is placed flush to the board and that, if required, it is aligned/orientated correctly. Please ask if you have any queries.

You can also bend the legs of the component out slightly to help keep it in place, as shown in Figure 12.

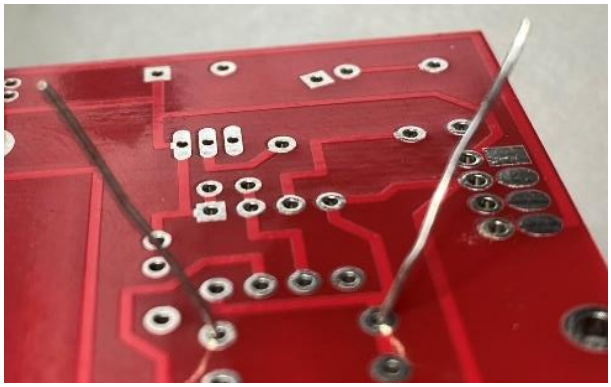


Figure 12: Component Legs Slightly Bent to Prevent the Component from Falling Out

It is important to remember that when soldering, we are aiming for 'the ideal solder joint' which resembles a cone or volcano – as shown in Figure 13.

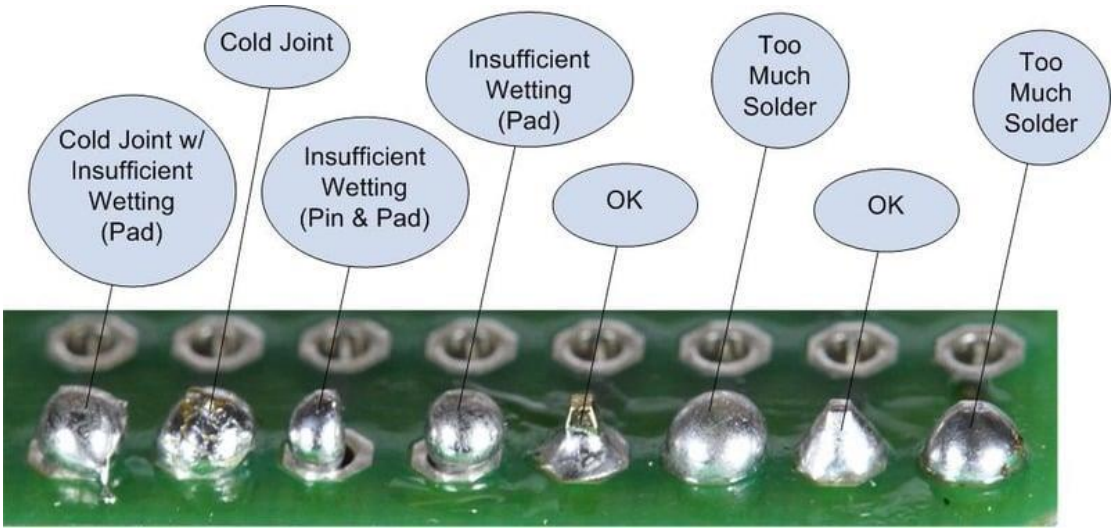


Figure 13: Solder Joint Guide

4.2 The Soldering Process

Now we know what we are aiming for, it is time to make the solder joint — this is a five-stage process consisting of:

1. **Heating the Joint:** Be sure to heat both the solder pad and the component leg or pin; a small amount of solder on the tip will help the heat flow.

2. **Apply the Solder**: Touch the end of the solder to the joint so that it contacts both the solder pad and the component leg or pin. It should melt and flow smoothly onto both the pin and the pad. If the solder does not flow, heat the joint for another second or two and try again.

3. **Let It Flow**: Keep heating the solder and allow it to flow into the joint. It should fill the hole and flow smoothly onto both the solder pad and the pin or component leg.

4. **Let It Cool**: Once enough solder has been added to the joint and it has flowed well onto both the component leg and the solder pad, remove the iron from the joint and allow it to cool undisturbed.

5. **Trim the Leg**: Use your wire cutters to trim the component leg close to the board. Please wear goggles when doing this as component legs may 'fly off'. This step applies only to components with wire legs. It is not necessary to trim the pins on integrated circuit chips or surface mount components – but don't worry, none of these are in your kits today.

This is pictorially shown below:

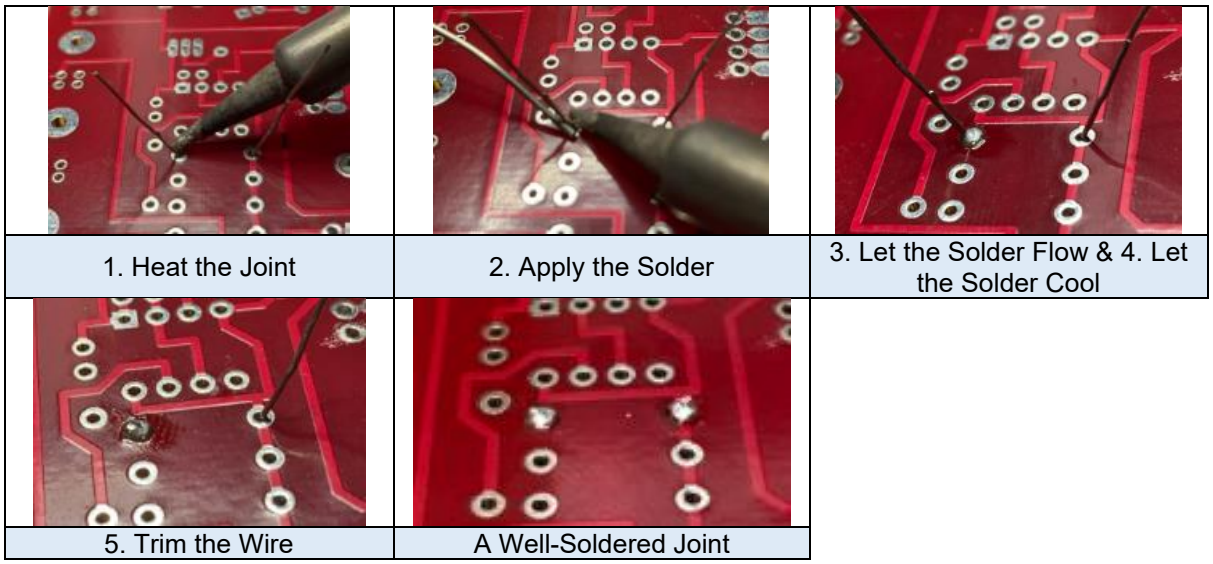


Figure 14: The Stages of Soldering

Have a go and please make use of the support on offer. Whilst soldering is a key skill in electrical and electronic engineering, it is developed through continuous practice – so whilst your first joint may not be ideal, you will see a noticeable improvement with your final few solders by the end of this activity. However, **if in doubt regarding the orientation and placement of components, it is always best to check prior to any soldering** – as it is much easier to fix any issues before the components are soldered down.

5. Programming the NEOBoard

5.1 The Example Code

There is one 'ready to go' example code available for you to upload to the NEOBoard, called 'NEOBoard_Dice_Example'. The example code is broken down along with the function of each component. Additionally, within the example code, once opened, you can read through the notes on how the program, and the functions, work.

NEOBoard_Dice_Example

This example code allows the NEOBoard to function as a dice, randomly 'rolling' a number which is displayed on the display. The larger 'Roll' button is used to randomise and display a

number between a particular range. The smaller ‘Mode’ button selects the dice to roll i.e. what range the randomised value will be within. The selected mode is displayed and are:

- d4; rolls a 4-sided dice and displays a random value between 1 and 4.
- d6; rolls a 6-sided dice and displays a random value between 1 and 6.
- d8; rolls an 8-sided dice and displays a random value between 1 and 8.
- d10; rolls a 10-sided dice and displays a random value between 1 and 10.
- d12; rolls a 12-sided dice and displays a random value between 1 and 12.
- d20; rolls a 20-sided dice and displays a random value between 1 and 20.
- d100; rolls a 100-sided dice and displays a random value between 1 and 100.

Sounds play out of the speaker when selecting different dice modes and when ‘rolling’ the dice. To mute the board, and silently use the NEOBoard, set the ‘MUTE’ variable at the top of the code to ‘true’. By default, this is set to ‘false’ so audio will play.

When ‘rolling’ the dice, a rolling LED effect is shown on the tri-seven segment display. As a result, this delays the ‘rolled’ number from being displayed by approximately second. To disable this effect and show the number quicker, set the ‘ROLL_EFFECT’ variable at the top of the code to ‘false’. By default, this is set to ‘true’ so the effect will be shown.

5.2 Uploading the NEOBoard Example Code

The following section will demonstrate the process of uploading code to the NEOBoard. The screenshots included will show the ‘NEOBoard_Dice_Example’ code being uploaded, however, the process is identical with any other code that you create. Once followed, if you are using the same device then you do not need to repeat the installations.

On the desktop, you should see a ‘NEOBoard_Dice_Example’ folder which contains the example code inside. Double-click the sketch ‘ino’ file to open it in the Arduino IDE, you should see a window like Figure 15 – the Arduino IDE version may differ.

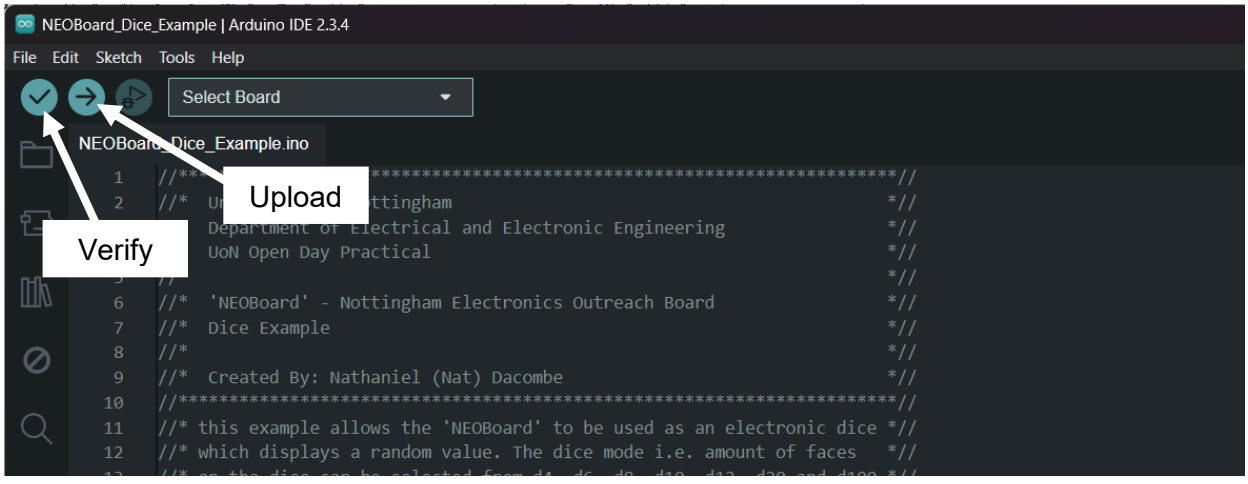


Figure 15: Arduino IDE & Example Code

Within the Arduino IDE, in the top left, there is a ‘tick’ and an ‘arrow’, which are the ‘Verify’ and ‘Upload’ buttons respectively. You can verify the code without any microcontroller being connected, and this will highlight any compilation errors or verify that the code can be uploaded via a message in the output window at the bottom. You can then click the upload button to program a microcontroller once it is connected correctly, as detailed below.

As shown in Figure 15, you can see that the code has already been written for you and is ready to upload ‘as is’. The comments in grey after the ‘//’ on each line fully explain how the code works so please take the time to read through these.

Before you can upload to the NEOBoard, the Arduino NANO Every microcontroller that we are using is not installed by default. Go to the boards manager by clicking ‘Tools → Board → Boards Manager…’, as shown in Figure 17 below.

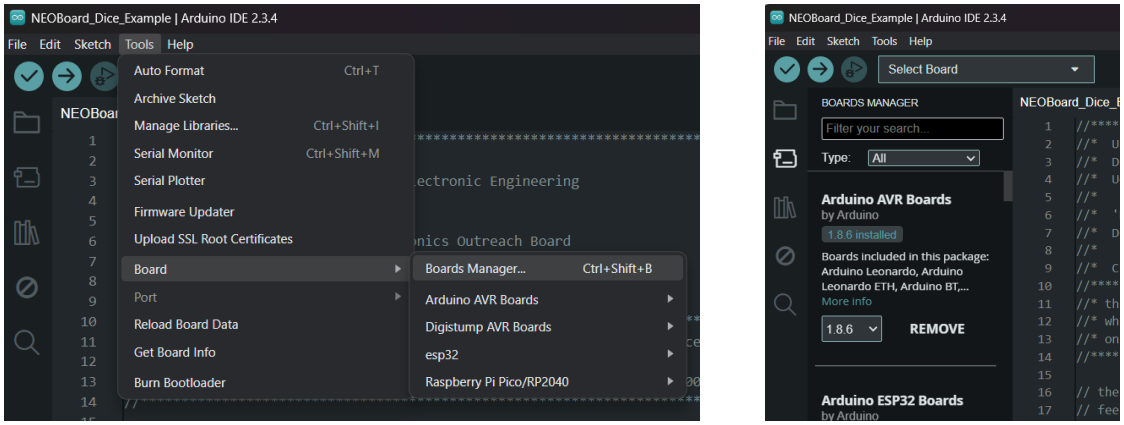


Figure 16: Boards Manager Select Menu (left) & Menu Tab (right)

Within this menu tab, type ‘megaavr’, and **install the latest version of ‘Arduino megaAVR Boards’ by Arduino.** This may take several minutes and will show ‘Installed’ when complete – installation messages for each board will appear in the output window.

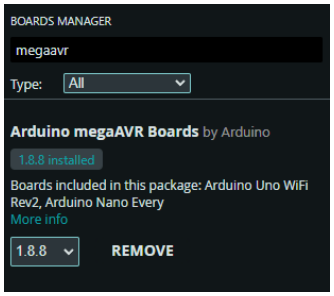


Figure 17: Boards Manager Window Post-Installation

To select the correct microcontroller board, select ‘Tools → Board → Arduino megaAVR Boards → Arduino NANO Every’.

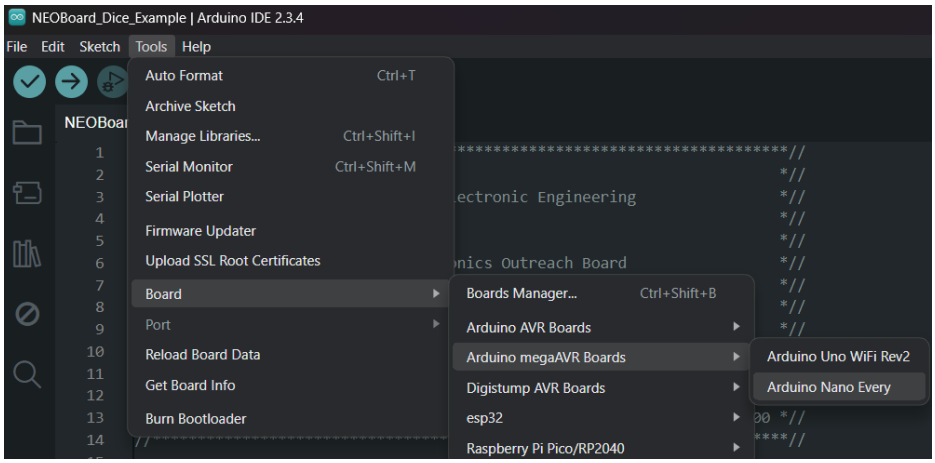


Figure 18: Selecting the Arduino NANO Every Board

Now that everything is installed, use the USB cable to connect the Arduino NANO Every microcontroller to the PC (or your laptop). Ensure that the correct board is selected and that you have selected the correct COM port under 'Tools → Port'. The correct port will have the description shown in Figure 19, although the port number may be different.

Figure 19: Arduino NANO Every Port Description

Once selected, click upload – this may take several minutes. Ignore the red 'avrdude' warning that is displayed in the output window. Once you see a message that says 'Done Uploading.' In the bottom right, you have successfully uploaded the code, and your NEOBoard should be working – **keep the USB cable plugged in to power it!** If the NEOBoard isn't working or the code hasn't been uploaded, see the troubleshooting steps on the next page.

5.3 Troubleshooting Steps

Troubleshooting is a key engineering skill, and whilst not a complete guide, please check the following, and feel free to ask for help if it still doesn't work.

If the Code Didn't Upload…

- To check that you have the correct COM port on Windows, open your 'Device Manager' by right-clicking the Windows icon in the bottom left and selecting 'Device Manager'. Under 'Ports (COM & LPT)' you are looking for the COM port number that is listed as 'USB Serial Device'.
- Disconnect and reconnect the USB cable at both ends. You may also want to try a different USB port on the PC/laptop – don't forget to recheck the COM port number.
- If you are still stuck, please ask one of the ambassadors or staff in the lab for help.

If the Code Did Upload…

- Check all of the components which you soldered, and resolder if needed;
    - Are all of the solder joints a high quality?
    - Are the LED and seven-segment display placed in the correct way round?
    - Are the correct resistor values used in the correct place?
- If all of the solder joints are good, then, with a screwdriver, rotate the blue 'trimpot', shown in Figure 20, located between the large button and the display. Gradually rotate this all the way in one direction, and then in the other direction, whilst pressing the large button. All being well, you should hear the quality and volume of the sound being played vary – stop rotating when you have found the position that delivers the best quality output.
- If you are still stuck, please ask one of the ambassadors or staff in the lab for help.

Figure 20: Blue Trimpot on the NEOBoard

6. What's Next?

You have now completed the NEOBoard activity, but what's next? Firstly, **this document and the NEOBoard are yours to keep and play around with**. You can power the NEOBoard, either with the USB cable, or if you'd prefer a more portable board, you can place three standard AA batteries in the battery pack at the back and ensure that the power switch is switched on.

We would love see what you do with the NEOBoard, so consider the following:

- Modify the example code to play different sounds or add music snippets
  - https://github.com/robsoncouto/arduino-songs/tree/master contains a large quantity of 'composed' sound and song sequences which you can copy into the code. Click on the folder, then the .ino file to view the code.
- Use the Arduino NANO Every microcontroller, buttons and seven-segment display in your own custom projects – the NEOBoard has been designed to be a versatile development board which can be used in most DIY electronics projects.

7. Useful Links

Arduino IDE
https://www.arduino.cc/en/software

Under 'Software', download the Arduino IDE relevant to your operating system. Note that this installer will not include the board or library setup so you will need to repeat these steps.

GitHub Repository
https://github.com/University-of-Nottingham-EEE-Projects/NEOBoard

This contains all of the example code used today, as well as resources related to the NEOBoard PCB.

For more information and ideas on what you could do next, please look at the following links. Please do let us know what you create, either with the NEOBoard or your own projects, by emailing nathaniel.dacombe@nottingham.ac.uk

Arduino NANO Every Project Ideas
https://www.instructables.com/Arduino-42/

This contains a list of over 200 different Arduino NANO-related projects – have a go at these, use them for inspiration or come up with your own entirely new ideas!

8. Thank You for Visiting!

Finally, all that is left to say is thank you for visiting the Electrical and Electronic Engineering Department today! We hope you've enjoyed this activity and we look forward to welcoming you as a student – remember to take your NEOBoard and documentation home with you!