E1: Diodes and an introduction to microcontrollers

This practical is in two halves. The first, besides the electronics content, is designed to further (re-)introduce you to the electronics labs so you can further (re-)familiarise yourself with the basic pieces of kit on the work-stations. The subject matter is diodes – we will compare properties of two types of diodes operating under DC and AC conditions, highlighting how real devices deviate from ideal behaviour. In the second half, you will be connecting the Arduino, which is a microcontroller, to diodes.

You will physically be handling at least the following items.

Components: pn diode, Schottky diode, Capacitor, resistor
Benchtop electronic/electrical units: DC power supply, Function Generator, Oscilloscope, Digital Multimeter (DMM), Arduino Microctroller, Raspberry Pi computer.
Connectors and leads: Proto-board, 4 mm leads (banana connectors), BNC cables and connectors, T-pieces, Crocodile clips, tinned solid copper core wires to use on the proto-board

Principles / Fundamental Ideas:

Current Voltage Characteristics - these are fundamental to device physics and device technology. It is also an archetypical example of a quantitative measurement of response of a physical system to stimuli which lab-based physics experiments are often about.
AC and DC
Rectification
Digital and Analogue Electronics

Pre-lab Activities

Read this script. Look at the data-sheets, watch the video on diodes (link from Moodle). Draw diagrams, write out any theory etc you think you might need during the practical so you can concentrate on the hands-on activities. As much as you can, think through what you will do in the lab.

Guide to activities:

Part I - Diodes

(a) Measure the DC characteristics of two types of diodes (pn and Schottky) (1 - 2 hrs).
(b) Measure some AC characteristics of two types of diodes (pn and Schottky) (0.5 - 1.5 hrs.)

Part II - Introduction to the Arduino

Fading LED and looking at the PWM output (1 - 2 hrs)

--- Check Point ---

Analogue output: Using PWM with a low-pass filter (10 min. minimum)
Automated IV characteristics - repeat part I and measure an LED as well in a few minutes (or shorter)! (30 min. minimum)

--- Extensions ---

See the end of this script for details on extensions.

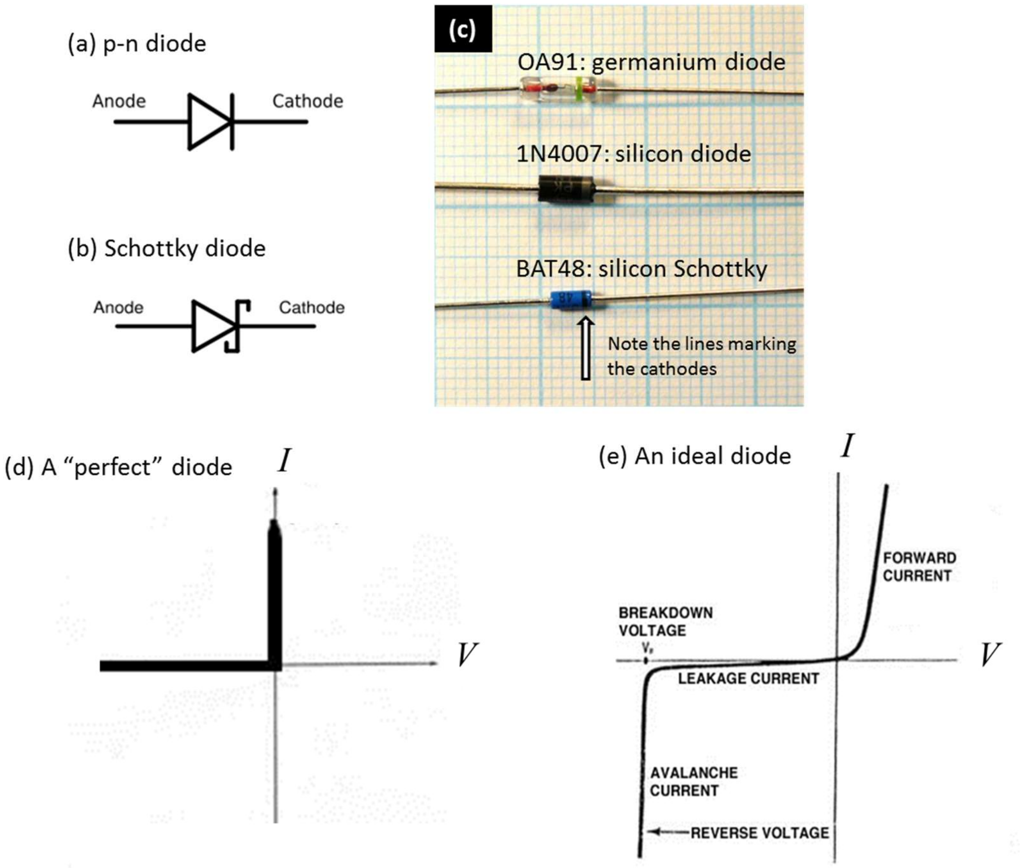


Figure 1: Diodes and IV characteristics. (a) A symbol for a p-n diode. (b) A symbol for a Schottky diode. (c) A photograph showing diodes examined in this experiment. (d) IV characteristics of a "perfect" diode. (e) An "ideal" diode. [Images (d) and (e) from American Microsemiconductor: www.americanmicrosemi.com]

Part I: Diodes

Background

Diodes are extensively used for a wide range of applications and familiarity with their basic properties is vital for understanding electronics and semiconductor devices. A perfect diode lets the conventional current flow only one way, in the direction of the arrow head symbol [Figure 1(a) and (b)] from "anode" to "cathode". This means that it has a low resistance when the current is flowing in the direction of the arrow, but a very high resistance when the current tries to flow the other way. Real diodes [Figure 1 (c)] are usually marked with a line around the cathode end of the diode.

Semiconductor diodes are widely used in electronic circuits and can be classified into two main groups: p–n junctions in which the diode is formed at the interface between a p–type semiconductor (holes are the majority carrier) and an n–type semiconductor (electrons are the majority carrier), and Schottky diodes, formed at the interface between a metal and a semiconductor.

Both types are important as practical devices; they also allow exploration of the underlying solid state physics. The full physical explanation of the processes occurring at such junctions is complex, but we do not need to know precisely what is going on in order to make widespread and sensible use of these devices. For example, in many practical applications, we can treat diodes as perfect "one-way valves" for electric current; in other words, they exhibit zero resistance in one direction, but infinite resistance in the other [Figure 1(d)].

However, this is a very simplified view of diode behaviour. If we look closely at the relationship between *V* and *I* for diodes, we see that this is not strictly true. In the forward direction, we see a small but non-zero resistance with current increasing exponentially with bias voltage. The ideal diode equation (also known as the Shockley diode equation) gives a reasonably accurate description of diode behaviour:

$$I = I_s\left(\exp\left(\frac{qV}{\eta kT}\right)-1\right) \tag{1}$$

where *I* is the current through the diode (+ve for forward bias; –ve for reverse bias), I_S is the reverse saturation current, *q* is the electronic charge (1.6×10^{-19} C), *V* is the applied bias voltage (+ve for forward bias; –ve for reverse bias), *k* is the Boltzmann constant (1.38×10^{-23} J/K), η is the ideality factor and *T* is the junction temperature in Kelvin. The dimensionless ideality factor, η, depends upon the relative magnitudes of the diffusion current and the recombination current present in the diode. It can also depend upon how a particular diode was fabricated. It usually has a value between 1 and 2. From the above equation, we can see that even for small forward voltages, the exponential term will dominate and the current will increase exponentially.

In reverse bias, the exponential term rapidly diminishes and we are left with $I = -I_S$. Thus, for reverse bias, the ideal diode equation predicts that the leakage current is small, roughly constant and equal in magnitude to I_S. Note the equation doesn't take into account the breakdown region of a diode [Figure 1(e)].

The Experiment

In this section, you will be provided with:

- a pn diode
- a shottky diode
- a 1K resistor

We will be using two different diodes: a silicon pn diode (1N4007) and a silicon Schottky diode (BAT 48). Before you begin, consult the appropriate data sheets (or if you are reading this in advance, any other appropriate resources) and try to find the maximum dc reverse bias voltage and maximum dc forward current that the two diodes will tolerate. Identify typical values for the reverse current, for each diode type up to some reasonable maximum bias voltage you might apply. You will find that the information provided may not explicitly be in the form you are looking for since the sheets are designed for engineers, and different information is provided depending on their use. You only need order of magnitude guideline figures here, just so that you know what voltages you can apply or current you can drive through them without causing them damage. **[Try not to spend more than 5-10 minutes on this – or better, have a go before you arrive (data sheets available on Moodle) – please ask if you are stuck!]**

Section (a): DC

(i) Forward Bias

Set up the circuit shown in Figure 2 and measure the variation of forward current through the diode as the voltage across it is varied. You should find that the current varies over several orders of magnitude for a relatively small range of bias voltages. You only need to record a handful of data points, but ensure that the current varies over a few decades (e.g. from 10 mA to 10 μA). Repeat the experiment with the two diodes (silicon-pn (1N4007) and the silicon Schottky (BAT48)).

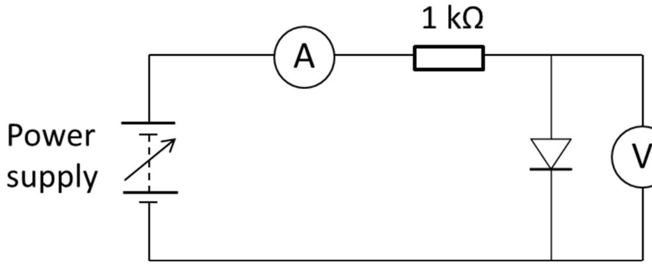


Figure 2: A circuit diagram for measuring the I-V characteristics of a diode under forward bias. The resistor is used to limit the current through the diode.

Look at the diode equation (equation 1) with which you would like to compare your data and think about what would be the most sensible way to plot your data. The equation predicts that if you plot the data in a particular way, you should obtain a straight line graph. Plot your data to produce forward I–V characteristics for the two diodes.

Analysis: Does your data agree with the ideal diode equation? From your plot, find values for I_S and T. The ideality factor should be about 1 for both these devices.

(ii) Reverse Bias

Now reverse the diode connections and repeat the measurements up to a reverse bias voltage of about 20 V. Since the diode has a high resistance in reverse bias, it is important to check that the voltmeter is not affecting the current measurement by drawing a current similar to that drawn by the diode. To check this, you can disconnect the voltmeter and then observe the ammeter to see if the reading changes. Repeat the experiment with the other diode. **[Discuss this with a demonstrator if you are unsure.]**

Plot your results. Does it agree with the ideal diode equation? Compare your results with your values for I_S obtained from your forward bias characteristic.

Section (b): Half-wave rectifier

So far, we have been looking at the DC properties of the diodes. We now examine their responses to time dependent voltages. Consider the circuit in figure 3 and think about what you would expect when an AC voltage is applied by the function generator. Connect up the circuit, with an oscilloscope to measure the time dependent voltage (i.e. the oscilloscope is the voltmeter in the diagram).

Use the 1N4007 diode first and apply a sinusoidal input voltage of about 8 V peak-to-peak. Observe the output waveform as the frequency is increased from about 1 kHz to about 1 MHz. Sketch the output waveforms in your logbook, including axis scales, and make a note of any important features.

Now replace the 1N4007 with the BAT48 and repeat the experiment.

Analysis: Comment on your results. How do the two diodes compare, AC and DC?

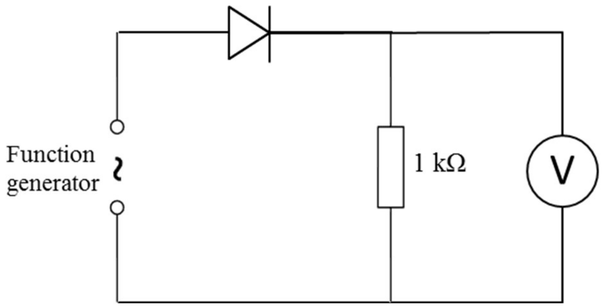


Figure 3: A half-wave rectifier. Use an oscilloscope to measure the time varying voltage.

Part II: Using a Microcontroller

Controlling an LED with an Arduino.

In E0, you will have successfully been able to make an LED that's built-in to the Arduino turn on and off, controlled by Python code running on the Pi. Here, you will make a discrete LED to the same thing, and then adjust the brightness.

Try loading up the same program that you used to make the on-board LED blink. Now try using a digital output pin (one of the output pins on the Arduino) to make a separate LED blink. The digital outputs on the Arduino provide either 0 V or 5 V ("False" or "True, ""off" or "on"). Use a 1 kΩ resistor in series with your LED to limit the current through the LED (just so that you don't blow them up – some LEDs are sensitive). A circuit diagram of a suggested circuit is shown below. The alteration you need to make to your Python code might be as simple as changing the address (a number) of the pin.

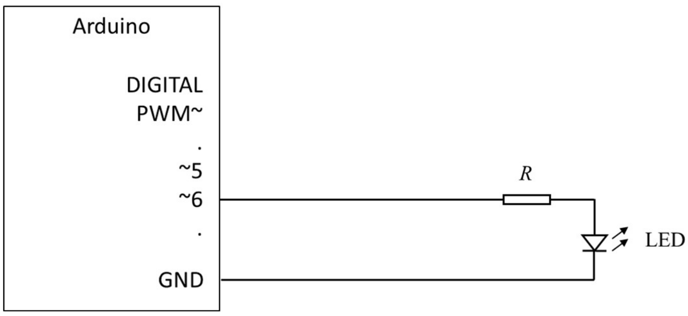


Figure: A possible circuit for lighting an external LED. Output pin 6 is used in this case.

We will now try making the LED fade or brighten gradually – that is, we will try to make the amount of light emitted by the LED increase or decrease with time. Unlike the DC voltage supplies you have on your bench whose output can be tuned continuously with a knob, the Arduino does not have real analogue outputs, but has Pulse Width Modulation (PWM). You will find that some of the output pins are marked with " ~ " – these are the ones capable of giving PWM output. [(Beware that these are not all equivalent…)] In PWM, the output is a repetition of square pulses, and can still only be set to one of two values at any one time. However, the ratio between the time spent set at the high value and low value can be adjusted, in order to adjust the time-averaged voltage and hence the name (PWM). This is what we will attempt here.

Take a look at the appropriate Jupyter Notebook provided on the Moodle page and try to adapt it to make an LED fade or brighten repeatedly.

Observe the voltage at the PWM output pin with an oscilloscope (make sure that the shielding on the scope is connected to the ground (and not the other way round))! Describe what you notice in your logbook. Is it doing what you thought it was doing?

--- Check Point ---

Show a demonstrator your oscilloscope screen and have a discussion!

--- Check Point ---

Keep your circuit connected how it is – the next bit only requires you to add one component so do have a go and see what happens!

Analogue output: Using PWM with a low-pass filter (20 min. minimum)

Try adding a 680 μF capacitor to your LED circuit as shown in the figure below. The idea is that the capacitor and the output resistance of the Arduino pin form a low pass filter. Take a look at the PWM output again with your scope. What do you notice on the scope? What do you notice about the way the LED fades or turns on and off? Can you explain what's going on?

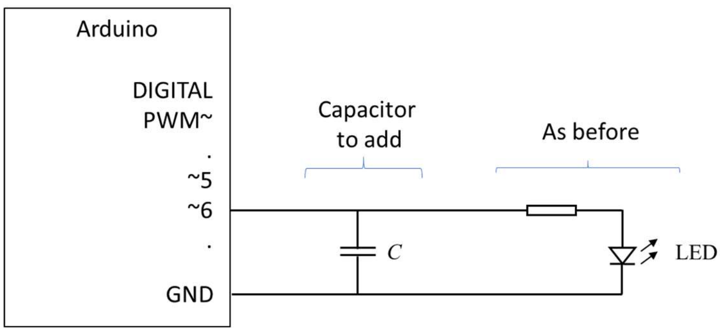


Figure: A possible circuit for lighting an external LED, but with an added capacitor.

If you have time, try smaller values of capacitances and see if it does what you expect!

Automated IV characteristics - repeat part I and measure an LED as well in a few minutes (or shorter)! (30 min. minimum)

Now consider this circuit below where A1 and A2 connect to the analogue inputs of the Arduino. The analogue inputs of the Arduino measure relative to the ground, so the voltage that would be measured at A1 will be the voltage across the capacitor (filtered PWM DC output), A2 will be the voltage across the 100 Ω shunt resistor and therefore measures current through the diode ($I = V/R$, where R, the resistance is a conversion factor) and the difference in voltages V(A1)-V(A2) is the voltage across the diode.

If we sweep the PWM output voltage, that would constitute sweeping the DC voltage across the diode, and measuring voltages at A1 and A2, we would be making a measurement of voltage applied across the diode and the current measured through it. That is, we have made a simple source-measure system!

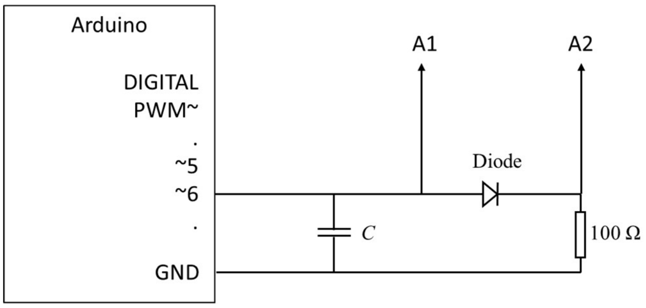


Figure: A possible circuit for measuring the I-V characteristics of a diode.

Have a go! Try the diodes you measured this morning. Try LEDs and compare them all, easily using Python! Once it's set up, it's trivial to change your diode and run again and again! You could write code to analyse your data too (or just think through how it would be done)!

--- Extensions ---

Have a go at the activities that interest you! Some of these are open ended.

Consolidation
If there are extensions on E0 you wanted to try but didn't get the time, you could try those here too.

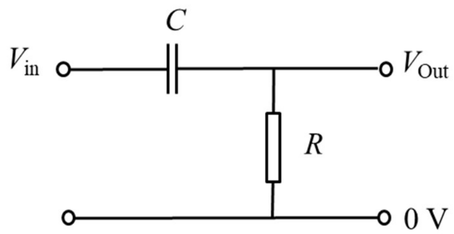
***AC output from PWM*?** (20 min.)
If you put a DC blocker at the PWM output, would the signal transmitted be AC? [Simple circuit diagrams from year one lecture notes are shown in the figure below.] How does your signal change with the value of the PWM output (on-off ratio)?

***How would you change polarity*?** (10-20 min. for bridge, otherwise open-ended)
Could you add a rectifying bridge to the AC voltage and get negative DC voltage? Give it a go! .] How does your signal change with the value of the PWM output (on-off ratio)? You measured the current through the diode in Part I under both forward and reverse bias. Can

you make the Arduino do that too without your having to swap the wires around and without using a relay by adapting what you have just done? You could add further filters to change it further too?

***Further exploration of the PWM output*.**
What is the rise time? Do you see ringing (where is it from and can you get rid of it)? Do all the PWM outputs operate at the same frequency? Can you measure the output impedance at the PWM output pin (this might be easier after E3)?

***How fast can you go*?**
How fast can you sweep the DC output in the source-measure program? What difference does it make to the data? What are the constraints due to the Arduino, using Firmata or your circuit parameters?



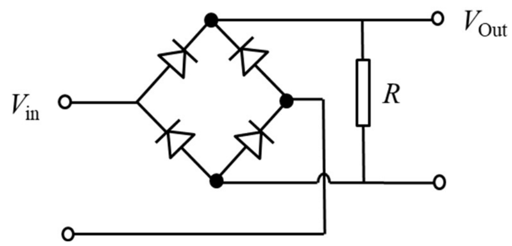


Figure: *Left*: A simple DC blocker circuit. *Right*: A full wave rectifying circuit.

End of E1.

KT/JM