Data Acquisition and Handling (DAH) Manual

Matt Needham (CO) matthew.needham@cern.ch

Ben Wynne b.m.wynne@ed.ac.uk

Senior Honours

Semester 1 2021/22

Practical Advice for Building Circuits

In an electronics laboratory there is some basic equipment, which you will inevitably have to use in the future as a physicist. These are:
- The Digital Multimeter, probably the most useful piece of kit that you will use – it is invaluable to have one at home.
- The Oscilloscope: this is the next most common electronics workhorse. The oscilloscope allows you to measure how a voltage in a circuit changes with time. This is particularly useful for looking at periodic signals.
- A power supply unit is a device for providing (a range of) voltages to power a circuit.
- A breadboard allows you to easily connect components together using short pieces of wire. A picture and a more detailed description is given in Appendix B.

The following are suggestions for building the circuits in the practical exercises. Lay out the physical circuit so that it models the layout of the circuit diagram.

- Lay out circuits neatly.
- Use coloured wires systematically.
- Connect the circuit ground, oscilloscope ground, signal generator ground and power supply ground together.
- Test complicated circuits as you build them --- it is much easier to spot faults as they occur.

A common experience of students in this laboratory is that they build an electronic circuit which (initially!) fails to work as expected. This is normal. One of the purposes of the laboratory is to teach you how to respond to this situation, that is, what to do next. You will get lots of practice …

- Don't panic!
- Be logical and systematic --- proceed in small, secure steps rather than one giant intuitive leap.

More specifically, check the following:

1. Is everything switched on?
2. Is the wiring correct? Check again with pencil and paper.
3. Is the ground wiring correct?
4. Poor connections to the breadboard can be a problem --- ensure that any wires inserted into the breadboard are held securely.
5. The wires used by the breadboard are single strand and can fracture if flexed unduly --- check the continuity of the wires with the digital multimeter.
6. Unplug the power supply from your circuit and test it with the digital multimeter. Do you have the voltage you expect?
7. Unplug the signal generator from your circuit and test it with the oscilloscope. Do you have the input waveform that you expect?
8. Are the correct d.c. voltages reaching the breadboard? The circuit components? Test with the digital multimeter.

9. Is the oscilloscope probe attenuation set correctly? Is the oscilloscope set to AC or DC coupling? What are your trigger settings?
10. Are the correct components being used?
11. Are one (or more) components faulty? Try swapping components.

None of these help? Ask a demonstrator.

In this course you will make extensive use of a Raspberry Pi, a credit card sized computer which allows you to connect many types of peripherals, including Analog to Digital and Digital to Analog Converters, Input/Output Expander, Temperature Sensor, etc. The following sections are descriptions of the Raspberry Pi properties and how to connect it with peripherals.

Raspberry Pi Hardware Specifications

Update 2017: We have replaced all Raspberry Pi's with the new model Raspberry Pi 3.

The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ (which was used previously in this course) and the Raspberry Pi 2 Model B.

Whilst maintaining the board format the Raspberry Pi 3 Model B brings you a more powerful processor, a Quad-Core 64bit CPU, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity.

The Raspberry Pi is built off the back of the Broadcom BCM2835 chip. The BCM2835 is a multimedia application processor geared towards mobile and embedded devices.
On top of this, several other components have been included to support USB, RCA, Composite Video and SD card storage. The following figure highlights some of the core-components of the Raspberry Pi board with a description of each provided:
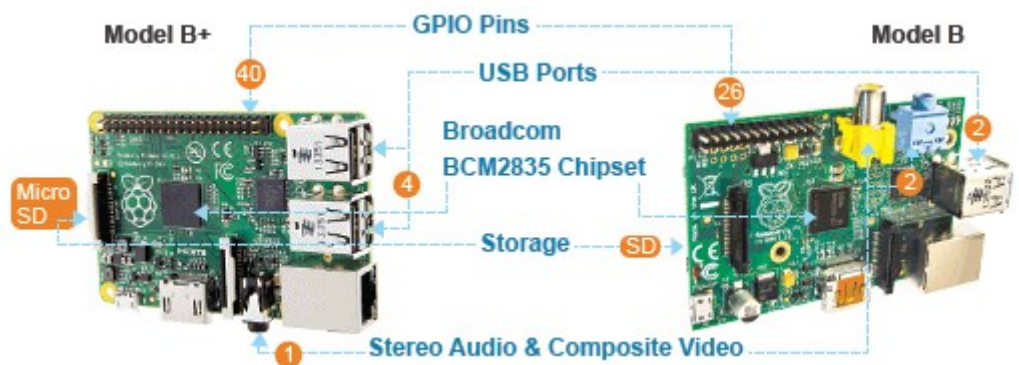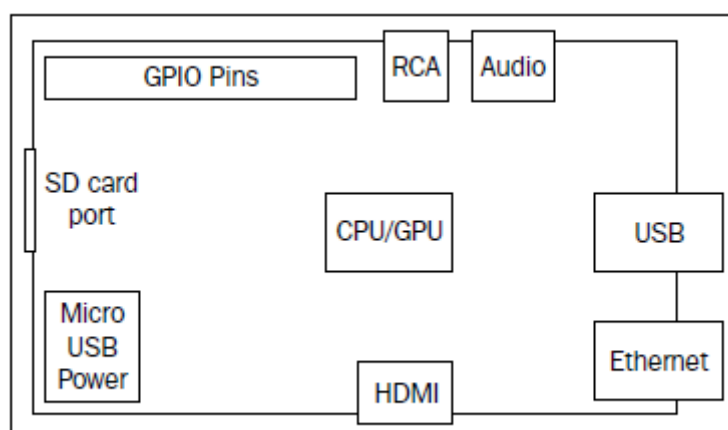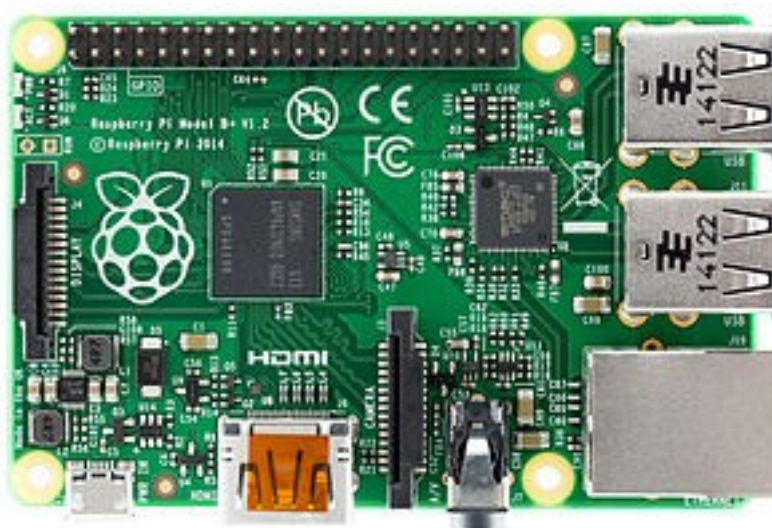
Dimensions

The Raspberry Pi is a small device coming in at 85.60mm x 53.98mm x 17mm and weighing only 45g. This makes it perfect for home automation, where a small device can be placed in a case and mounted inside an electrical box, or replace an existing thermostat device on a wall.

3.5mm analog audio jack (model B)

The 3.5mm analog audio jack allows you to connect headphones and speakers to the Raspberry Pi. This is especially useful for audio and media player based projects.

Composite RCA port (model B)

You are probably familiar with the composite cables used to hook up your DVD player to the TV. They usually come in the red, white, and yellow plug variety. The Raspberry Pi has a port for attaching the yellow video cable from your TV to it, allowing you to use your TV as a monitor.

USB 2.0 ports (4 (2) on model B+(B)) plus one micro USB

USB is one of the most common methods for connecting peripherals and storage devices to a computer. The Raspberry Pi model B+ (B) comes equipped with four (two) USB ports, allowing you to hook up a keyboard and mouse when you get started and a micro USB port for powering your device.

HDMI port

The High Definition Multi-media Interface (HDMI) port allows the Raspberry Pi to be hooked up to high-definition televisions and monitors that support the technology. This provides an additional option to the composite RCA port for video and additionally supports audio.

Should you wish to stream video and audio from the web to your TV, this is the port you would want to use.

SD card port

The main storage mechanism of the Raspberry Pi is via the SD card port. The SD card will be where we install our operating system and will act as our basic hard disk. Of course, this storage can be expanded upon using the USB ports.

512 MB SDRAM shared with GPU

The Raspberry Pi model A comes equipped with 256 MB and 512 MB on the newer models (B and B+). This isn't a huge amount, and much less than you would expect on a PC, where RAM is available in gigabytes. However, for the type of applications we will be building, 256 MB or 512 MB of RAM will be more than enough.

CPU

The Raspberry Pi comes equipped with a 700 MHz, ARM1176JZF-S core – part of the ARM 11 32-bit multi-processor core family. The CPU is the main component of the Raspberry Pi, responsible for carrying out the instructions of a computer program via mathematical and logical operations. The Raspberry Pi is in good company using the ARM 11 series and has joined the ranks of the iPhone, Amazon Kindle, and Samsung Galaxy.

GPU

The graphics-processing unit (GPU) is a specialized chip designed to speed up the manipulation of image calculations. In the case of our Raspberry Pi, it comes equipped with a Broadcom VideoCore IV capable of hardware accelerated playback and support for OpenGL. This is especially useful if you want to run games or video via your Raspberry Pi, or work on 3D graphics in an open source application such as Blender.

Ethernet port

The Ethernet port is the Raspberry Pi's main gateway to communicating with other devices and the Internet. You will be able to use the Ethernet port to plug your Raspberry Pi into a home router such as the one you currently use to access the Internet, or a network switch if you have one set up.

GPIO pins

The General Purpose Input/Output (GPIO) pins on the Raspberry Pi are the main way of connecting with other electronic boards such as the Arduino. As the name suggests, the GPIO pins can accept input and output commands and thus can be programmed on the Raspberry Pi. The Arduino shields will be attached to the GPIO via a bridge shield allowing us to transfer data from sensors soldered to the device back to the Raspberry Pi. This is especially useful in home automation projects, where we may wish to store sensor data or manipulate motors based upon a program running on the Raspberry Pi's operating system. On model A

and B we have 26 pins and on model B+ we have 40 pins while the first 26 are identical with models A and B.

Status LEDs

The model B board also has five status LEDs that have the following meanings:
- The *OK* LED indicates SD card access; it blinks whenever the Pi tries to access the SD card. You can control this LED by software, so it's not completely accurate.
- As soon as you connect a power supply to the Pi, the *PWR* LED turns on.
- The *FDX* LED shows whether your LAN is running full duplex.
- At every LAN activity, the *LNK* LED blinks.
- The *10M* LED indicates whether the Pi's Ethernet link is running at 10Mbit/s or 100Mbit/s. When this LED is on, the Pi runs at 100Mbit/s

The model B+ board has 2 status LEDs:
- The *OK* LED (green) indicates SD card access; it blinks whenever the Pi tries to access the SD card. You can control this LED by software, so it's not completely accurate.
- As soon as you connect a power supply to the Pi, the *PWR* LED (red) turns on.

Basic peripherals

A power supply, a keyboard, a mouse, and a display.

What the Pi Does Not Have

The Pi does not have a real-time clock (RTC) with a backup battery, and it does not have a Basic Input Output System (BIOS).

BIOS is a program stored in read-only memory (ROM) that runs on a PC at startup. Among other things, it's responsible for configuring new devices and for determining the boot order. The Pi has no BIOS, so it always boots from an SD card. Even if you have a perfectly valid installation of an operating system on a USB stick or an external hard drive, you cannot boot it. Of course, you can still use external storage devices, but you cannot use them to boot the Pi.

The SD card – the Raspberry Pi's storage device

An **SD (secure digital)** card is a form of portable high performance storage medium available for electronic devices ranging from cameras to PCs.
The Raspberry Pi comes equipped with an SD card slot allowing us to insert an SD card and use it as our devices' main storage mechanism, much like a hard disk on a PC.
While you can use other storage mechanisms such as a USB drive or USB external hard drive, the SD card is small and thus lends itself better to embedded devices such as those found in home automation projects. The Raspberry Pi supports larger SD cards such as those with 64 GB of storage space. Model B+ has a micro-SD card slot.

Pre-installed versus a blank SD card: Since the Raspberry Pi has been released, a number of websites are offering preloaded SD cards that come installed with one of the operating systems that are available for the Raspberry Pi.

These are a good option for amateur enthusiasts looking to get started with the Raspberry Pi, who do not want to go through the setup process and are happy with a pre-loaded single operating system.

The Rasbian Operating System

Like every computer, the Raspberry Pi needs an operating system, and the preferred one for the Raspberry Pi is Linux. That's partly because it's free, but mainly it's because it runs on the Pi's ARM processor while most other operating systems work only on the Intel architecture. Still, not every Linux distribution will run on the Pi, because some do not support the Pi's particular type of ARM processor.

You need to install the operating system on an SD card. You can actually create several SD cards, each with a different operating system, so at the end you'll have a pretty versatile system that you can turn into completely different machines by simply replacing the card.

Linux is still the most popular choice for an operating system on the Pi, and it helps you to get the most out of the Pi. Even if a Linux distribution runs on the Pi, it will often look and behave different from its "regular desktop PC" equivalent, because it might use a windows manager that does not need a lot of resources. Also, you won't find all of the applications you're used to such as many popular web browsers or office products.

There are some limitations around installing the operating system. You can't insert a DVD into the Pi and install it, and because the Pi has no BIOS you cannot boot from an external USB drive either. You also cannot copy an ISO image of a DVD to an SD card. Instead, we need a snapshot of a system that has already been installed and that we can boot from. So, you have to create or find an image of a Linux distribution that you can copy to an SD card, and it has to be compatible with the Pi.

Raspbian is based upon the Debian Linux operating system and has been optimized for use with Raspberry Pi. *Mike Thompson* and *Peter Green* of raspbian.org developed it and while not an official product of the Raspberry Pi foundation, is the operating system the foundation recommends for beginners on their website. Raspbian has now been updated to the new stable version of Debian, which is called Buster.

For those of you not familiar with Linux, it is a group of open source operating systems that uses the Linux Kernel and provides an alternative to applications such as Windows.

Mac OS X users may be familiar that they are using a Unix-like operating system that gives them many of the command-line functionalities that Linux users are familiar with. They will also find some similarities with the Raspbian operating system, which we are installing on the Raspberry Pi.

There are several reasons for deciding to go with the Raspbian operating system that are listed as follows:

- Raspbian has a desktop environment similar to Windows and Mac called LXDE, so it provides an easy transition for those not familiar with Linux command line.
- It comes pre-installed with software that will be useful for writing code for the Raspberry Pi and Arduino such as Python. It also includes other software that you may be interested in exploring that has an educational bent. One example is Scratch, a tool for introducing programming to children.
- The operating system has been tailored to run on the Raspberry Pi. The code compilation is optimized for on-chip floating-point calculations (hard-float) rather than a slower software-based method.
- There is wide spread community support for the operating system, meaning that as you move forward with projects beyond this book, there will be plenty of resources as well as help available to you.

Running the Raspberry Pi

Starting the Desktop

After you've successfully logged into your Raspberry Pi, you'll still see not much more than a shell prompt. Start the desktop to see some more colors using the following command.

studentnn@dahpimm ~ $ startx

where nn and mm are your student number in the lab, and the number of the Pi

The desktop environment you've just started is named LXDE. Most programs are available from the buttons in the toolbar at the top.

To leave LXDE, use the menu icon (raspberry) at the top left of the screen. Click on item "Shutdown" and you will be given three options: (i) "Shutdown", (ii) "Reboot" and (iii) "Exit to command line". If you select "Exit to command line" you will arrive at the shell prompt and you need to run

studentnn@dahpimm ~ $ logout

Python on the Pi

Python is an *interpreted language,* which means that you can write a program or script and execute it directly rather than compiling it into machine code. Interpreted languages are a bit quicker to program with, and you get a few side benefits. For example, in Python you don't have to explicitly tell the computer whether a variable is a number, a list, or a string; the interpreter figures out the data types when you execute the script. The Python interpreter can be run in two ways: as an interactive shell to execute individual commands, or as a command line program to execute standalone scripts. To start python on the Raspberry Pi, run

studentnn@dahpimm ~ $ python3

If you have saved code in a .py file, run it like this

studentnn@dahpimm ~ $ python3 my_code.py

Appendix A: GPIO Pins

You need to learn how to tinker with electronics. That's why the Raspberry Pi has an expansion header that makes it easy to connect it to your own electronics projects. You'll learn how to build your own small electronics devices and control them with the Pi. To build all projects, you need only a few cheap parts:
- A half-size breadboard
- A few LEDs (red, yellow, and green)
- A few resistors in the range of 220Ω to 1kΩ
- female/male jumper wires

Meet the Pi's GPIO Pins

To connect your own electronics projects to the Pi, you can use the expansion header in the top-left corner of the Pi. It consists of 26 pins arranged in two rows containing 13 pins each. The top row contains the even-numbered pins, and the other row contains the odd-numbered pins. That is, the first pin in the lower row is pin 1, and you can find the label "P1" on the Pi below the pin.

In Figure, you can see the meaning and the numbering of the pins. With pin 6, the Pi can share a common ground with your electronics projects. Using pins 1 and 2, you can power external devices connected to the Pi with 3.3 volts or 5 volts. The Pi limits the output of pin 1 to 50mA, while pin 2 allows for a current draw that depends on the USB input current. Pins 4, 9, 14, 17, 20, and 25 are reserved for future enhancements, so you cannot use them in your own projects. The remaining pins are general-purpose input/output (GPIO) pins that you can use as digital input or output pins. Note that the GPIO pin names do not correspond to the pin numbers of the expansion header, see also Appendix D for the Raspberry Pi B+ J8 header.

|	5V	-	Ground	GPIO14	GPIO15	GPIO18	-	GPIO23	GPIO24	-	GPIO25	GPIO8	GPIO7
	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Pin	2	4	6	8	10	12	14	16	18	20	22	24	26
Pin	1	3	5	7	9	11	13	15	17	19	21	23	25
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	3v3	GPIO0	GPIO1	GPIO4	-	GPIO17	GPIO21	GPIO22	-	GPIO10	GPIO9	GPIO11	-

Raspberry Pi model B header.

Appendix B: Breadboard Basics

Breadboards are one of the most fundamental pieces when learning how to build circuits. In this tutorial, you will learn a little bit about what breadboards are, why they are called breadboards, and how to use one. Once you are done you should have a basic understanding of how breadboards work and be able to build a basic circuit on a breadboard.
https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard#power
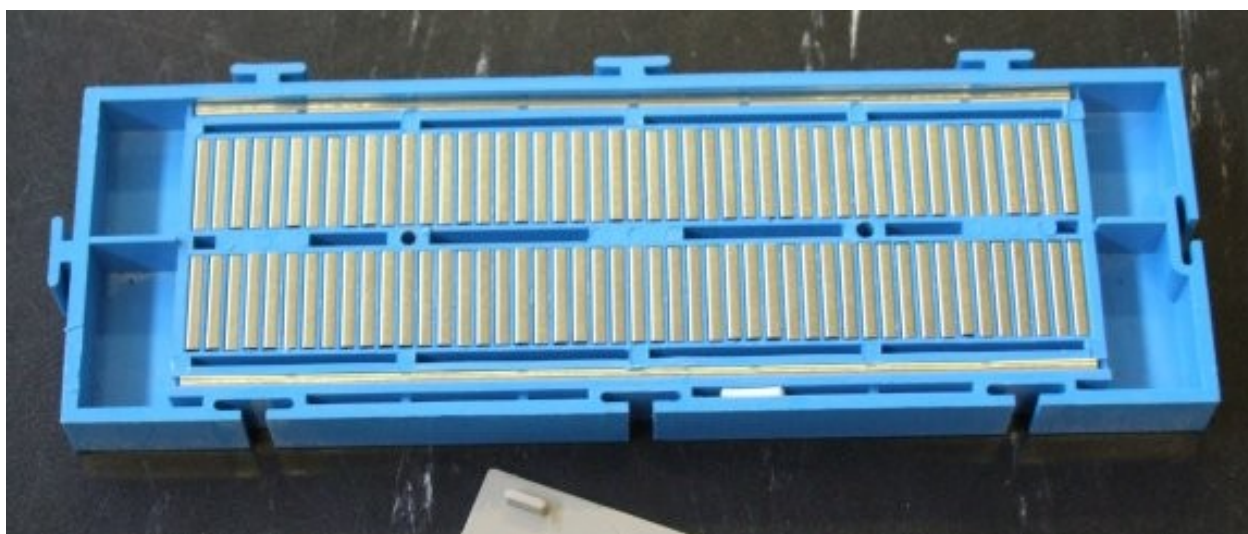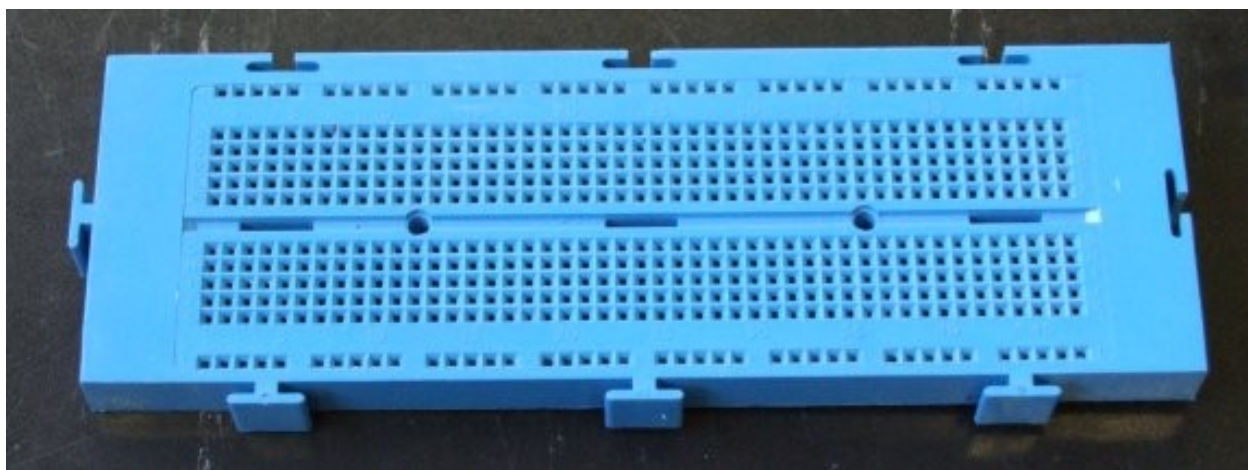
Here we have a breadboard where the backplate has been removed. You can see lots of horizontal rows of metal strips on the bottom of the breadboard.
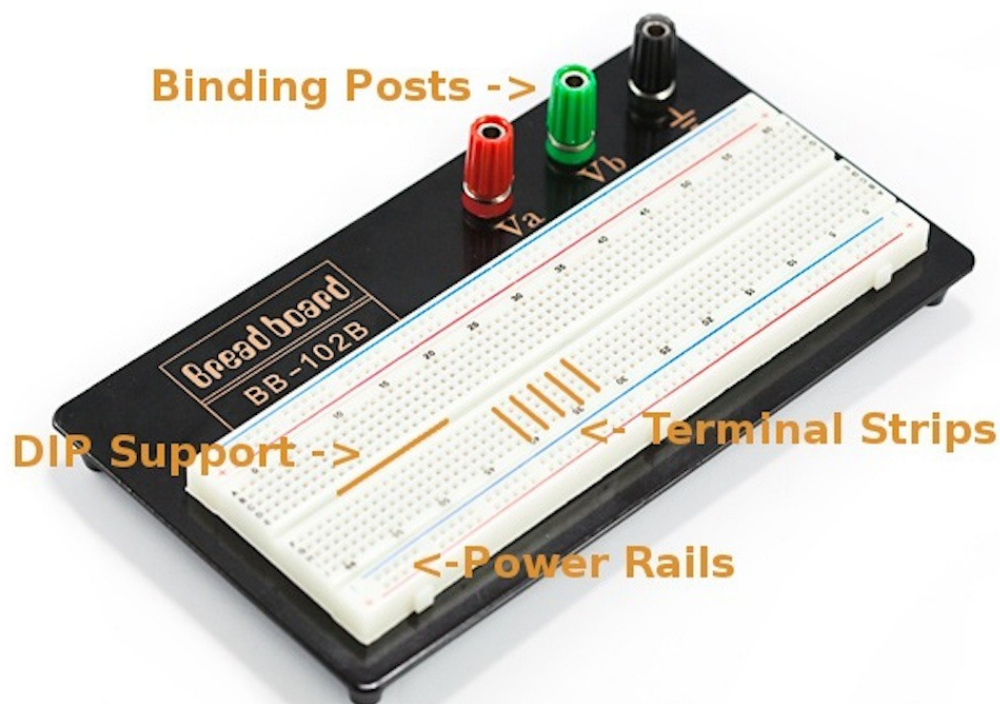
Terminal Strips

The tops of the metal rows have little clips that hide under the plastic holes. These clips allow you to stick a wire or the leg of a component into the exposed holes on a breadboard, which then hold it in place.

Once inserted that component will be electrically connected to anything else placed in that row. This is because the metal rows are conductive and allow current to flow from any point in that strip.

Notice that there are only five clips on this strip. This is typical on almost all breadboards. Thus, you can only have up to five components connected in one particular section of the breadboard. The row has ten holes, so why can you only connect five components? You'll

also notice that each horizontal row is separated by a ravine, or crevasse, in the middle of the breadboard. This ravine isolates both sides of a given row from one another, and they are not electrically connected. We'll discuss the purpose of this in just a bit, but, for now, just know that each side of a given row is disconnected from the other, leaving you with five spots for components on either side.
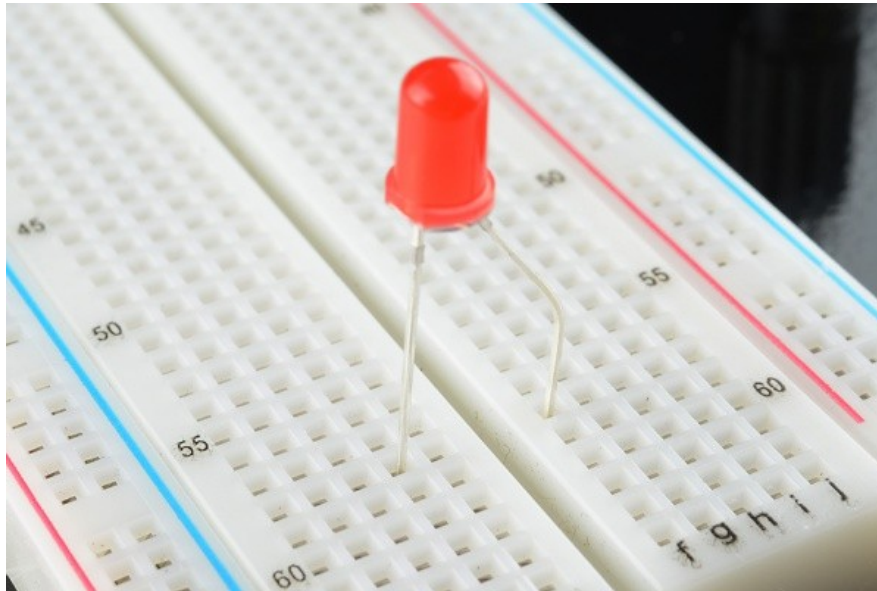

An LED inserted into a breadboard. Notice how each leg of the LED is placed on either side of the ravine. This prevents the connections to the LED from being shorted.

Power Rails

Now that we've seen how the connections in a breadboard are made, let's look at a larger, more typical breadboard. Aside from horizontal rows, breadboards usually have what are called power rails that run vertically along the sides.

These power rails are metal strips that are identical to the ones that run horizontally, except they are, typically all connected. When building a circuit, you tend to need power in lots of different places. The power rails give you lots of easy access to power wherever you need it in your circuit. Usually they will be labeled with a '+' and a '-' and have a red and blue or black stripe, to indicate the positive and negative side.
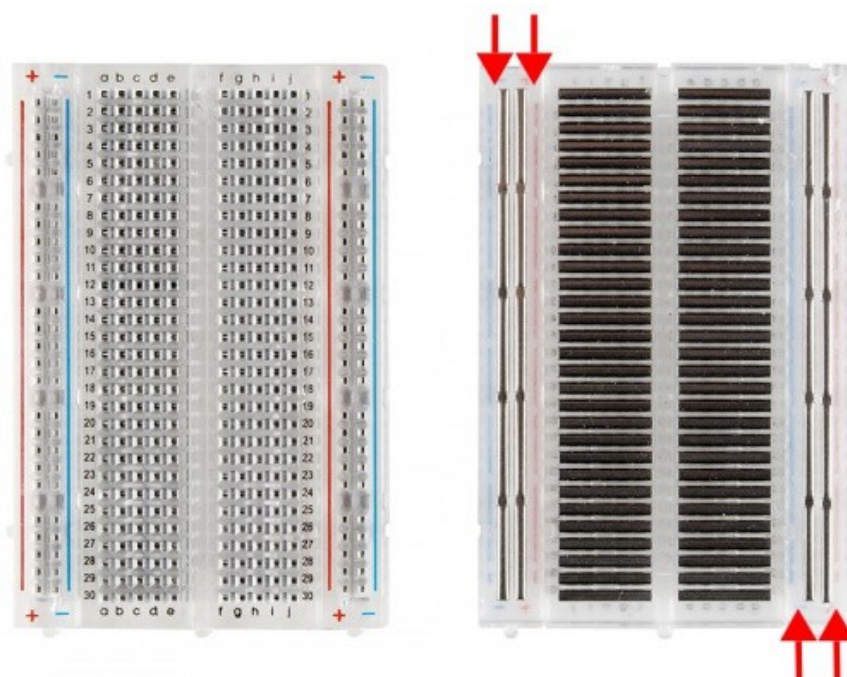
A medium-size breadboard with the adhesive back removed to expose the power rails.

It is important to be aware that the power rails on either side are not connected, so if you want the same power source on both sides, you will need to connect the two sides with some jumper wires. Keep in mind that the markings are there just as a reference. There is no rule that says you have to plug power into the '+' rail and ground into the '-'rail, though it's good practice to keep everything in order.
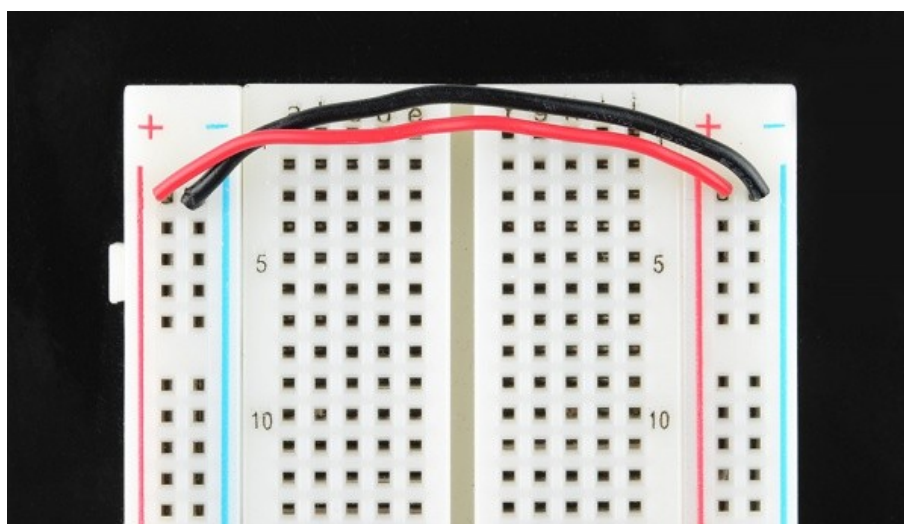

Two jumper wires used to connect the power rails on both sides. Always attach the '+' to '+' and the '-' to '-'.

DIP Support

Earlier we mentioned the ravine that isolates the two sides of a breadboard. This ravine serves a very important purpose. Many integrated circuits, often referred to as ICs or, simply, chips,

are manufactured specifically to fit onto breadboards. In order to minimize the amount of space they take up on the breadboard, they come in what is known as a Dual in-line Package, or DIP.

These DIP chips have legs that come out of both sides and fit perfectly over that ravine. Since each leg on the IC is unique, we don't want both sides to be connected to each other. That is where the separation in the middle of the board comes in handy. Thus, we can connect components to each side of the IC without interfering with the functionality of the leg on the opposite side.
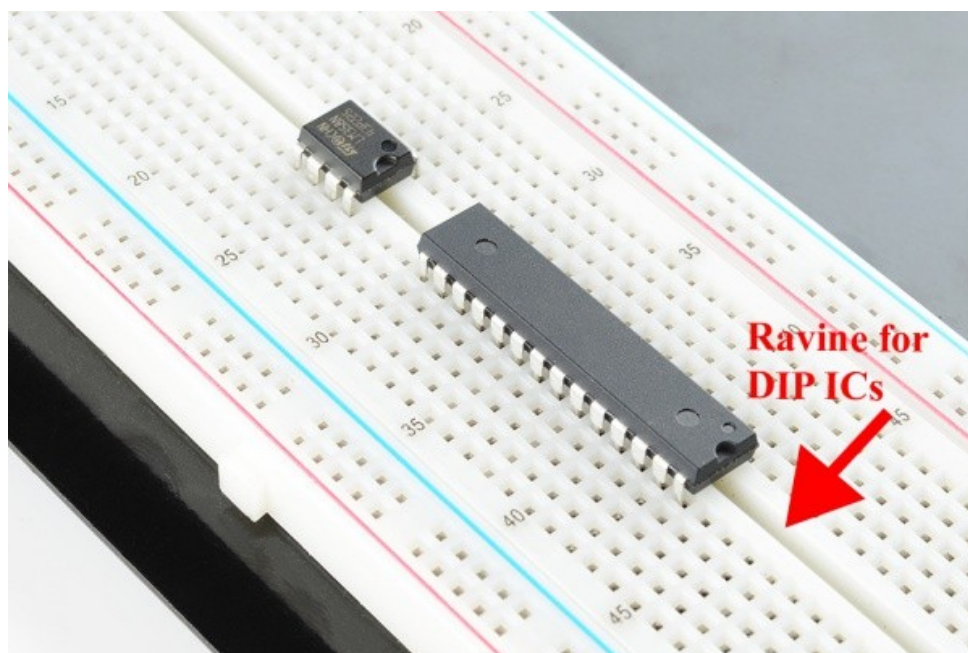
*Two DIP ICs, the LM358 (top), a very common op-amp, and the ever-popular **ATMega328 microcontroller** (bottom).*

Rows and Columns

You may have noticed that many breadboards have numbers and letters marked on various rows and columns. These don't serve any purpose other than to help guide you when building your circuit. Circuits can get complicated quickly, and all it takes is one misplaced leg of a component to make the entire circuit malfunction or not work at all. If you know the row number of the connection you are trying to make, it makes it much simpler to plug a wire into that number rather than eyeballing it.

## Providing Power to a Breadboard

Binding Posts

As mentioned in the previous section, some breadboards have binding posts that allow you to connect external power sources.

The first step to using the binding posts is to connect them to the breadboard using some jumper wires. Although it would seem that the posts are connected to the breadboard, they are not. If they were, you would be limited to where you could and couldn't provide power. As we've seen, breadboards are meant to be totally customizable, so it would make sense that the binding posts are no different.

With that, we have to connect wires to the posts in order to connect them to the breadboard. To do that, unscrew the post until the hole going through it is exposed. Slide the stripped end of your jumper wire through the hole, and screw the post back down until the wire is firmly connected.

Typically, you only need to connect a power and ground wire from the posts to the breadboard. If you need an alternate power source, you can use the third post.
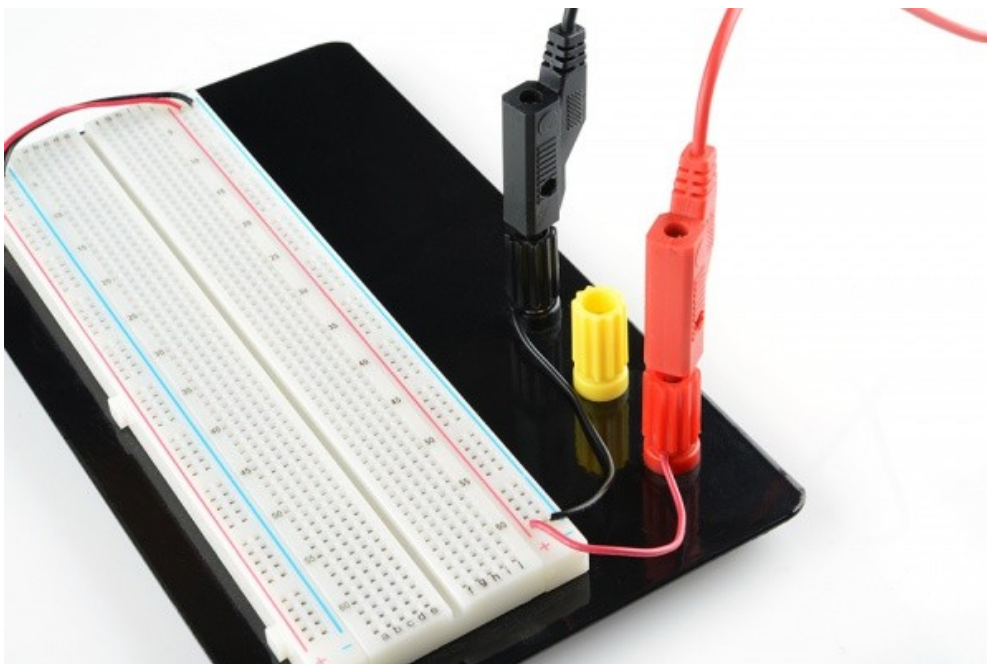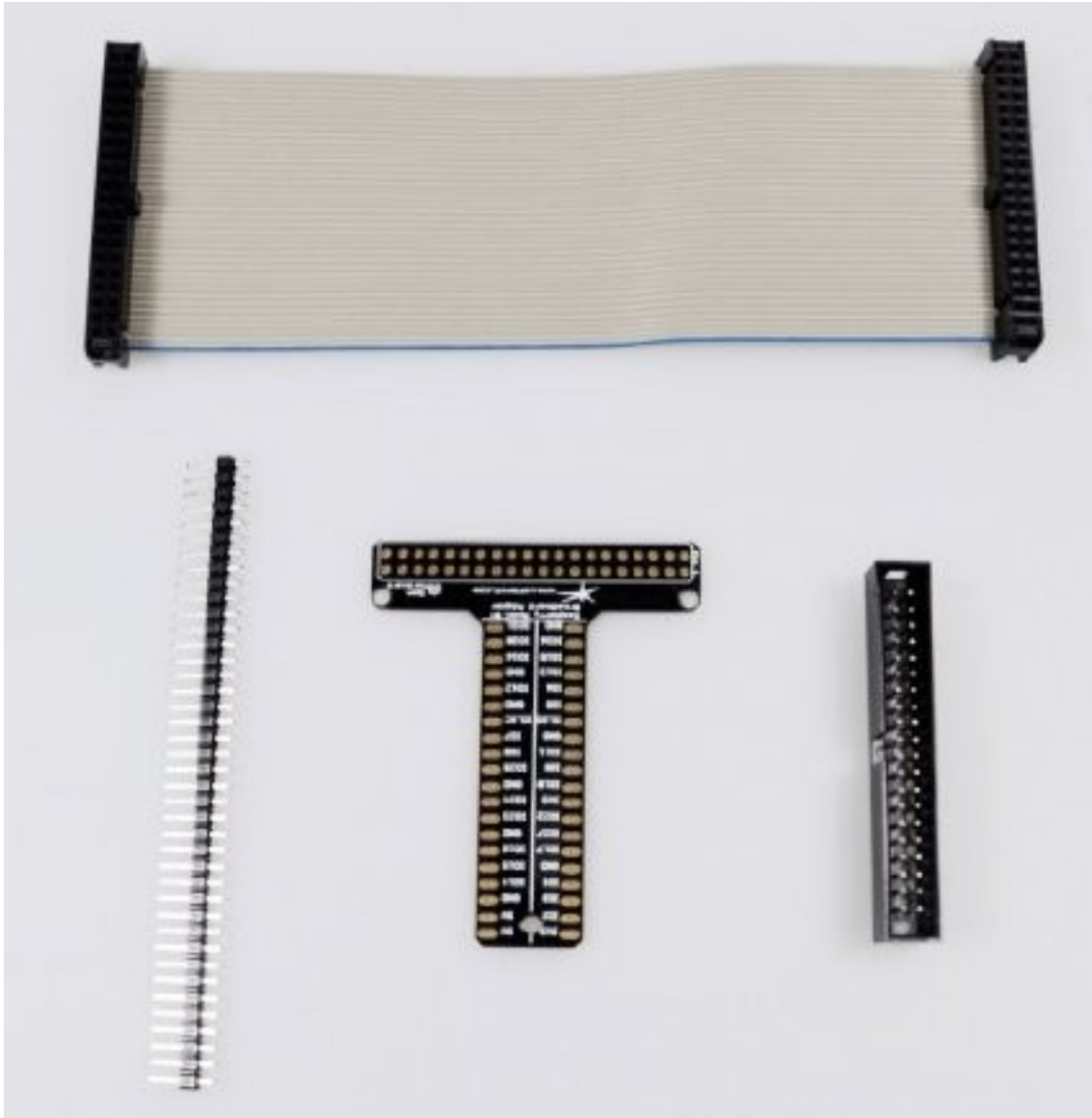

Breadboard Power Supplies

Yet another method for powering your breadboard is to use one of the many breadboard power supplies available. SparkFun carries a number of kits and boards that you can use to plug power directly into your breadboard. Some allow you to plug a wall wart directly into the breadboard. Others allow you to pull power directly from your computer via the USB connections. And, almost all of them have the capability to adjust the voltage, giving you a full range of the common voltages needed when building circuits.
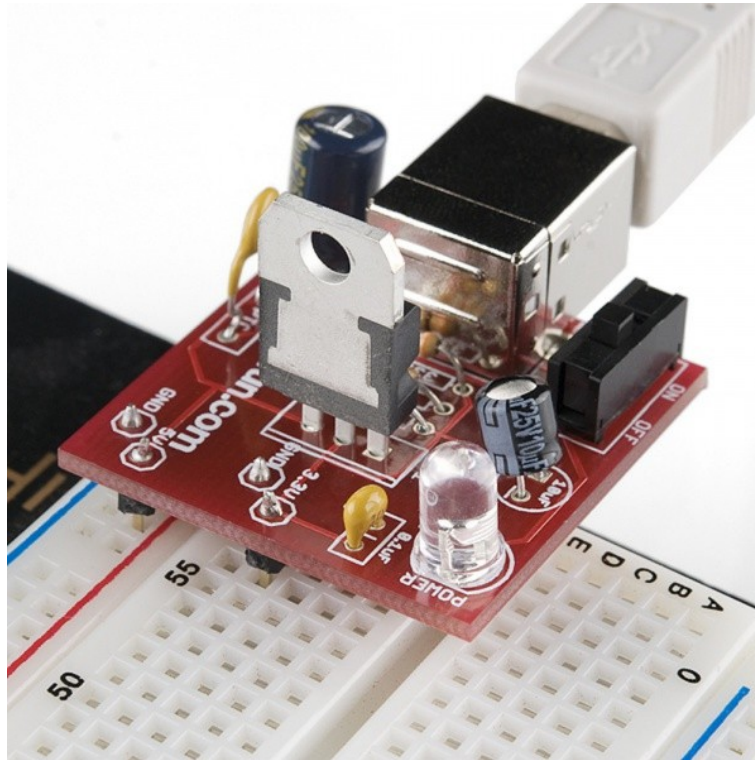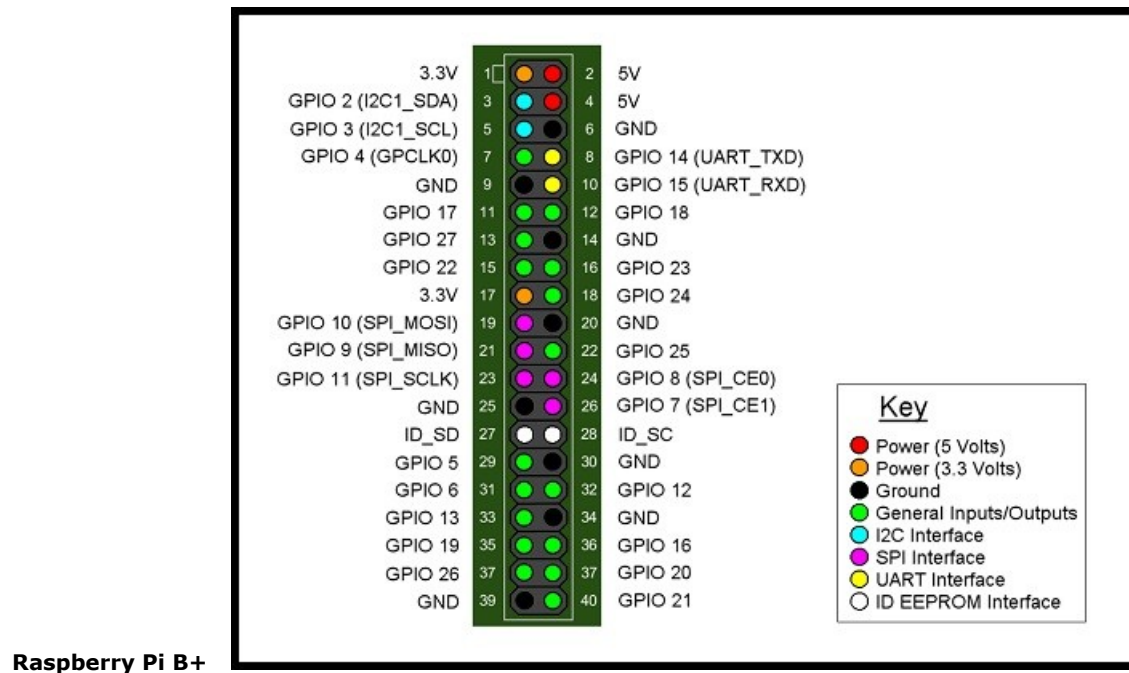
A SparkFun USB Breadboard Power Supply that pulls power from your computer's USB and has the option to choose between 3.3V and 5V.
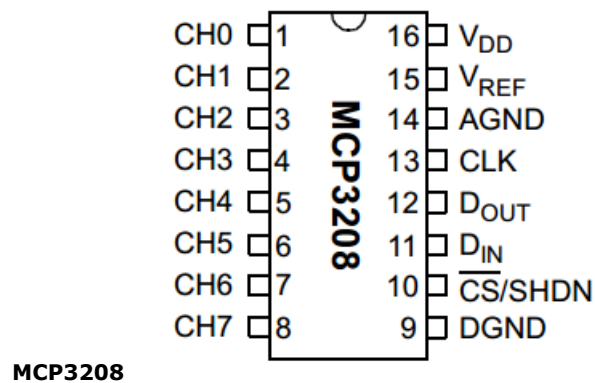
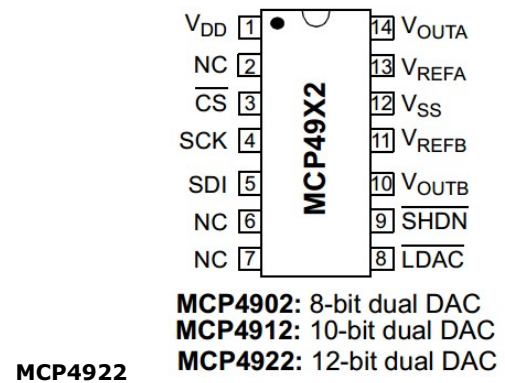# Appendix C: Raspberry Pi B+ to Breadboard Adapter
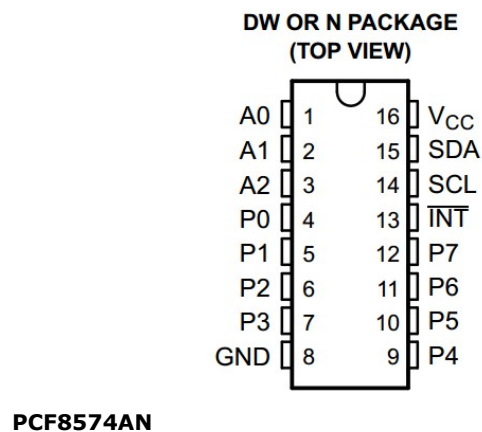
Appendix D: Raspberry Pi B+ J8 header

3.3V	1	● ●	2	5V
GPIO 2 (I2C1_SDA)	3	● ●	4	5V
GPIO 3 (I2C1_SCL)	5	● ●	6	GND
GPIO 4 (GPCLK0)	7	● ●	8	GPIO 14 (UART_TXD)
GND	9	● ●	10	GPIO 15 (UART_RXD)
GPIO 17	11	● ●	12	GPIO 18
GPIO 27	13	● ●	14	GND
GPIO 22	15	● ●	16	GPIO 23
3.3V	17	● ●	18	GPIO 24
GPIO 10 (SPI_MOSI)	19	● ●	20	GND
GPIO 9 (SPI_MISO)	21	● ●	22	GPIO 25
GPIO 11 (SPI_SCLK)	23	● ●	24	GPIO 8 (SPI_CE0)
GND	25	● ●	26	GPIO 7 (SPI_CE1)
ID_SD	27	● ●	28	ID_SC
GPIO 5	29	● ●	30	GND
GPIO 6	31	● ●	32	GPIO 12
GPIO 13	33	● ●	34	GND
GPIO 19	35	● ●	36	GPIO 16
GPIO 26	37	● ●	37	GPIO 20
GND	39	● ●	40	GPIO 21

Key
- ● Power (5 Volts)
- ● Power (3.3 Volts)
- ● Ground
- ● General Inputs/Outputs
- ● I2C Interface
- ● SPI Interface
- ● UART Interface
- ○ ID EEPROM Interface

Raspberry Pi B+

MCP3208

CH0	1		16	V_{DD}
CH1	2		15	V_{REF}
CH2	3	MCP3208	14	AGND
CH3	4		13	CLK
CH4	5		12	D_{OUT}
CH5	6		11	D_{IN}
CH6	7		10	$\overline{CS}$/SHDN
CH7	8		9	DGND

MCP4922

V_{DD}	1		14	V_{OUTA}
NC	2		13	V_{REFA}
$\overline{CS}$	3	MCP49X2	12	V_{SS}
SCK	4		11	V_{REFB}
SDI	5		10	V_{OUTB}
NC	6		9	$\overline{SHDN}$
NC	7		8	$\overline{LDAC}$

MCP4902: 8-bit dual DAC
MCP4912: 10-bit dual DAC
MCP4922: 12-bit dual DAC

**DW OR N PACKAGE
(TOP VIEW)**

A0	1		16	V_{CC}
A1	2		15	SDA
A2	3		14	SCL
P0	4		13	$\overline{INT}$
P1	5		12	P7
P2	6		11	P6
P3	7		10	P5
GND	8		9	P4

PCF8574AN

```
Pinout & Wiring for Pi to MCP3208
```

MCP 3208 Pin	Pi Pin Name
==============	=============
16 VDD	3.3 V
15 VREF	3.3 V
14 AGND	GND
13 CLK	GPIO11 SPI0_SCLK
12 DOUT	GPIO09 SPI0_MISO
11 DIN	GPIO10 SPI0_MOSI
10 CS	GPIO08 CE0
9 DGND	GND

```
Pinout & Wiring for Pi to 4922
```

MCP4922 Pin	Pi Pin Name
==============	=============
1 VDD	3.3 V
13 VREFA	3.3 V
11 VREFB	3.3 V
12 VSS	GND
4 SCK	GPIO11 SPI0_SCLK
5 SDI	GPIO10 SPI0_MOSI
3 CS	GPIO07 CE1
9 nSHDN	3.3 V
8 nLDAC	GND

```
Pinout & Wiring for Pi to PCF8574
```

PCF8574 Pin	Pi Pin Name
==============	=============
16 VCC	3.3V
15 SDA	GPIO2 (I2C1_SDA)
14 SCL	GPIO3 (I2C2_SCL)
8 GND	GND
3 A2	GND
2 A1	GND
1 A0	GND

Appendix D: Logic

3.3V

R
1K

LED

GPIO

GPIO

LED

R
1K

GND
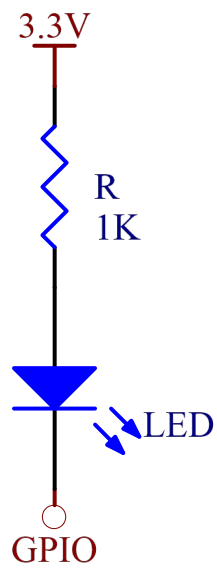
"Negative Logic"
or
"Active Low"
LED
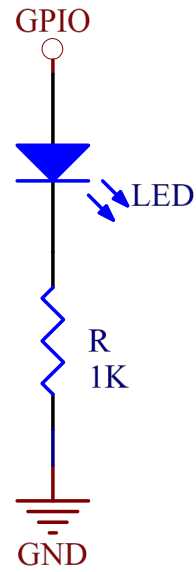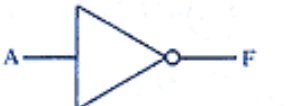
"Positive Logic"
or
"Active High"
LED

3.3V

R
1K

○ GPIO

SW-PB

GND
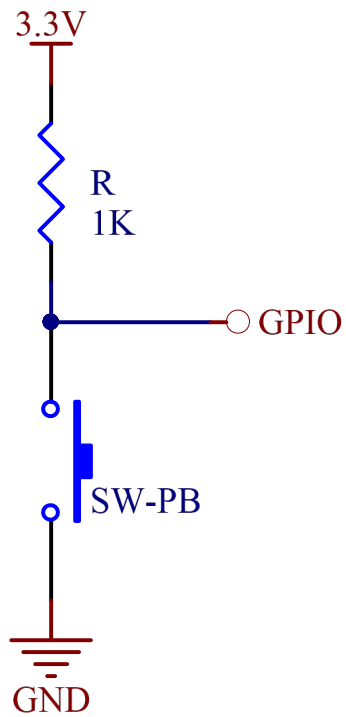
"Negative Logic"
or
"Active Low"
Switch

3.3V

SW-PB

○ GPIO

R
1K

GND
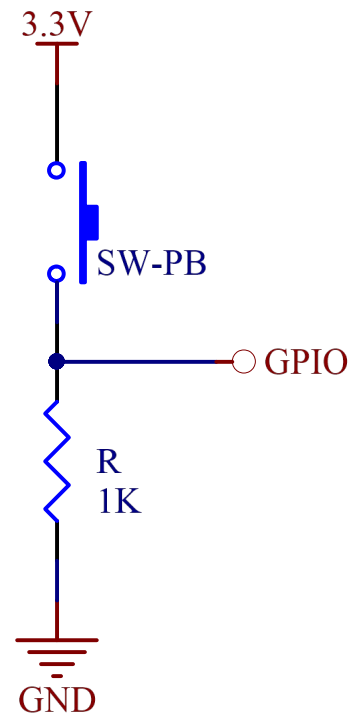
"Positive Logic"
or
"Active High"
Switch

Name	Graphic Symbol	Algebraic Function	Truth Table
AND	A ———⊐ ⊐— F B ———⊐	$F = A \cdot B$ or $F = AB$	A B | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR	A ———⊐ ⊐— F B ———⊐	$F = A + B$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
NOT	A ——▷o— F	$F = \overline{A}$ or $F = A'$	A | F 0 | 1 1 | 0
NAND	A ———⊐ ⊐o— F B ———⊐	$F = (\overline{AB})$	A B | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0
NOR	A ———⊐ ⊐o— F B ———⊐	$F = (\overline{A + B})$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0