sparkfun
START SOMETHING

SHOP   **LEARN**   BLOG   SERVICES

Introduction to the Raspberry Pi GPIO and Physical Computing

CONTRIBUTORS: *ASASSY*

♡ **FAVORITE** | 6

GPIO Pins Overview

Okay, so we've hooked up the necessary hardware to the Raspberry Pi, and you're itching to start using it as a general-purpose PC for web surfing and Youtube watching and looking at cat memes. But where the Pi *really* shines is with its 40-pin GPIO (General Purpose Input & Output). The GPIO enables you to switch devices on and off (output) or receive data from sensors and switches (input). What this really means is that you can connect the Raspberry Pi to everything from a smart mirror to a weather station, to an asset tracking robot that displays its coordinates on a web server. Let's take a deeper dive into what's possible with the GPIO header! We can reference the GPIO Pinout Diagram provided by the Raspberry Pi Foundation below as a visual guide for understanding the pins:
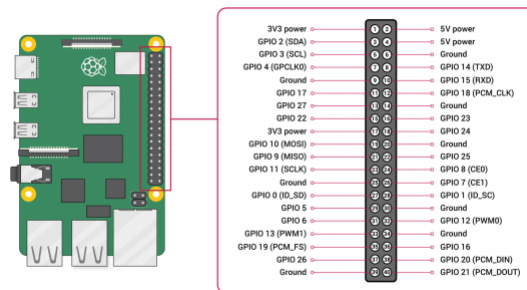
Power Pins

Let's start with the power pins, which include both 3.3V and 5V. These pins transmit power as output to power attached peripherals.

Ground

And if we're going to be rigging up electrical circuits here with power, then we'll need a ground. You'll find plenty of grounds within the pins as well.

Standard GPIO Pins

Take out the power and ground pins, and you'll be left with the pins that are dedicated to sending output and receiving input. All of these can be used for straightforward input/output tasks!

Chatty Cathy Pins

Some of the standard GPIO pins are used for communication purposes. Here's a quick overview of these communication protocols!

- **SPI pins** – The Serial Peripheral Interface (SPI) is a communication protocol used to transfer data between micro-computers like the Raspberry Pi and peripheral devices. The MISO pin receives data, and the MOSI

pin sends data from the Raspberry Pi. Furthermore, the serial clock pin sends pulses at a regular frequency between the Raspberry Pi and the SPI device at the same speed in which the devices to transfer data to each other.

- **UART pins** – UART stands for universal asynchronous receiver-transmitter, which is a physical circuit designed to send and recieve data.
- **PWM pins** – PWM means "pulse width modulation," which is a communication protocol best used with stuff that moves and lights up: motors, LEDs, and so on.
- **I2C pins** – I^{2}C is short for inter-integrated circuit (two "inters" or I"squared"C). It works similarly to SPI, but it doesn't force you to use nearly so many pins.

Identifying Pins

There are two ways to identify GPIO pins...the first is by its physical position on the board, or its BOARD name. It also has a Broadcom SOC channel (BCM) name. By opening the terminal and running `pinout` as a command, it will return a map of the Raspberry Pi GPIO pins and their names based on the Broadcom chip on board.
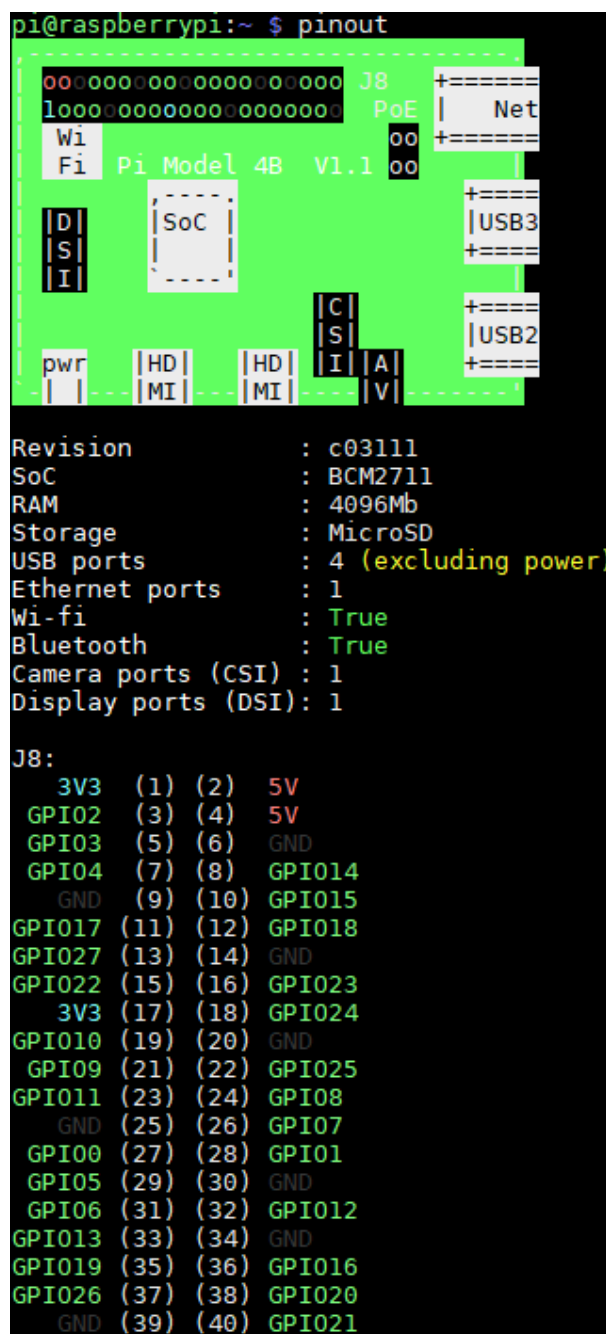
Map for the Raspberry Pi 4 based on the pinout command.

If you really want to get deep into the Raspberry Pi GPIO pinout, there is an incredibly thorough Raspberry Pi GPIO pinout guide that is interactive and steps through each type of pin on-board. This general overview outlined above should be enough to get us started on our project though!