# Collecting Data

# Introduction

The main reason for launching the balloon is to collect data at different altitudes and then analyse this data afterwards which means that we need a way to collect and save this data at regular intervals and then be able to access it when the balloon lands again. We also want to take pictures as they look rather cool and give a perspective on how high the balloon actually got. These objectives can be achieved in a number of ways.

## Choosing a Device

Perhaps the easiest option is an Arduino. They have little power consumption and many pins, which allow us to connect a lot of sensors. They are also small and lightweight, a key feature for this project. To begin with we tried the Arduino Pro Mini, also used with the tracker. It has 14 digital I/O pins and 6 analogue pins in a small form factor. Controlling all this is the ATmega328 processor running at 8Mhz. However, we soon found that it has a major drawback: the lack of processing power. Arduino cameras are often low resolution, even with the more powerful, 32-Bit ESP32 boards, which are clocked at 240Mhz. In comparison, the ATmega328 struggled to process images at all, meaning it was inadequate for our needs.

This led us to look at more powerful alternatives, namely the Raspberry Pi. There are a number of different models, but the Raspberry Pi Zero W is by far the best for this project as it is smaller and consumes less power. These versatile boards can be used for almost any project as they offer a full Linux environment paired with a wide range of I/O pins. They also have a CSI camera connector which allows the Raspberry Pi camera to be attached. This camera module is more advanced than our previous module and can capture resolutions of up to 1080p and the Raspberry Pi Zero has a clock of 1Ghz, substantially more that the Arduinos we used. The Linux environment also allows us to use different languages, namely Python, which we are more experienced with, to interact with the sensors.

## Powering the Device

Now we have chosen our device, we need to power it. In normal circumstances, the Raspberry Pi is normally powered by a USB Type B linked up to the mains. For obvious reasons, this isn’t possible at high altitudes. It is possible to power the Raspberry Pi from its 5v pin headers. Standard AA batteries don’t offer enough current at the 5v used by the Raspberry Pi, which means it fails to boot. This leaves us one other option, Li-Po batteries. These batteries can give a much higher current at lower voltages which makes them ideal for powering such devices. Additionally, these batteries connect easily to the Raspberry Pi’s pin headers, making them ideal. However, they operate at 3.7v when fully charged and the voltage drops as they discharge. This variation in voltage could be enough to break a Raspberry Pi. To fix this, we used a 5v regulator to ensure the voltage stayed at 5v. After testing, we found that it ran well and had a long enough battery life for the launch duration.

## The Sensors

There are three main measurements we want to take during the flight: temperature, pressure and humidity. Luckily, there are a range of sensors that meet our needs. To measure temperature and humidity, we chose the DHT22 sensor, which is often used in DIY weather stations because it is cheap and accurate. It also comes with lots of documentation for many different platforms such as the Raspberry Pi or Arduino. To measure pressure, we chose the BMP085 module. This also comes with lots of support and Adafruit also provides a library to use the module on the Raspberry Pi. We were able to setup both modules reasonably quickly on the Raspberry Pi and they both work well. We did some quick tests such as putting the modules in the fridge and freezer to ensure they work. The BMP085 library can estimate altitude, but we found this to be inaccurate when it tried to tell us we were at -85m below sea level when, in fact, we were at about 50m above sea level. This isn’t a problem as we have a much more accurate altimeter as part of the GPS module on the tracker.

## The Camera

The camera gave us quite a few problems. It had to be small, lightweight and cheap. We started off by trying to use an OV7670 camera module for Arduino. It can only take images at 480p and at a very slow shutter speed, which means that images were often blurred and of low quality. Primarily, we tried using it on the Arduino Pro Mini, but found the Arduino didn’t have the power to process the images. After this, we tried using it on the Raspberry Pi, but found the images were of bad quality and not good enough for our needs. By this point, we had decided that we were going to use the Raspberry Pi to collect sensor data. This gave us the idea to use the official Raspberry Pi Camera Module, which is capable of taking images up to 1080p, and therefore of much better quality. It can also record video and use higher shutter speeds. This will reduce the risk of getting blurred images. The Raspberry Pi Camera module doesn’t have autofocus and is instead focused to infinity. This is fine for our application because the ground will be a lot further for a majority of the flight.