**Pi-rates: Build guide**

**The coding:**

First, we programmed the temperature sensor onto a spare Raspberry Pi so that it would check whether the drink in the cup was cool enough to drink and checked that it worked.

<https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/> we used this website to source the code for the temperature sensor

Then we coded the screen on another spare Pi to make it show images. It was quite difficult to code, but we were able to use emojis to communicate the different temperatures and flow rates.

this was the emoji for when the liquid was too hot.

this was the emoji used to show that the liquid was cold

this was the emoji used to show that the user had not drunk enough water

this was used to tell the user they had drunk a decent amount of water

this was the emoji used to show that the user had drunk a healthy amount of water.

<https://www.waveshare.com/wiki/2inch_LCD_Module#Download_Examples> we used the code from this website to code the screen, substituting their images for the ones above.

<https://realpython.com/python-command-line-arguments/#displaying-arguments> we used this website to help us give the python program the name of our image files

Next, we coded the water flow sensor on to the Pi which held the temperature sensor. This would measure how much water was drunk out of the cup and show whether the user had drunk enough. When we tried to test it would just show zeros, but eventually we found out the code we used was dividing the answer by 160 so the answer would be zero. We changed it to make sure it didn’t divide every time.

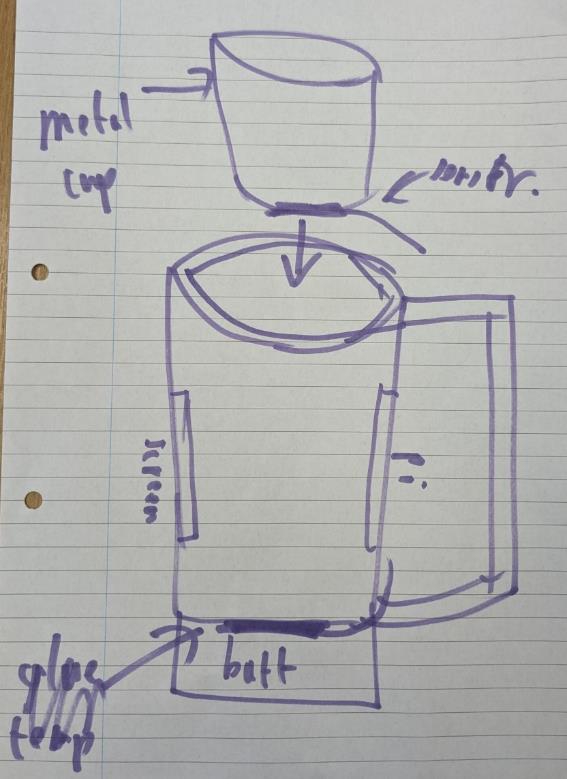
<https://www.dfrobot.com/forum/topic/317058> we used this code to monitor how much someone drunk from the cup per second.

<http://razzpisampler.oreilly.com/ch07.html> this was used to help us connect the button to the raspberry pi and for it to listen for when it was pressed.

<https://www.influxdata.com/blog/what-is-time-library-in-python-helpful-guide/> we used this to help us track time, because the original code used time.sleep causing the button press not to work, instead it keept checking to see if the time has been over a second, allowing the button press to work

**The building:**

Firstly, we designed and 3D printed the outer cup model. When designing it we made sure that it had spaces for buttons and the Pi to attach to the outside of the cup, that it was deep enough to hold all the hardware and there was a handle to hold the cup by whilst drinking. We had to slope the edges of the holders for the mini pi and screen to avoid the edges breaking or hurting someone.

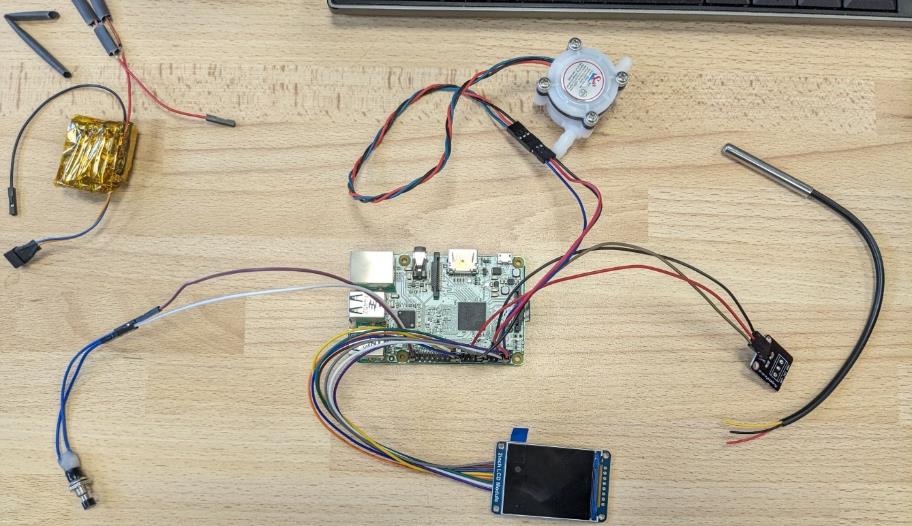
This was our initial design for the cup, from which our cup was based on:

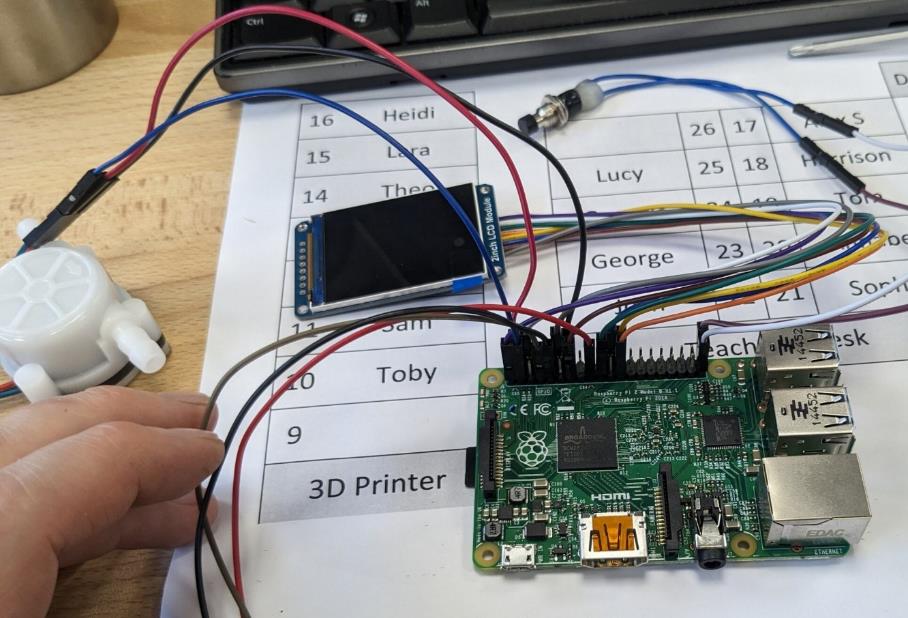
This is the rendering for the design of our cup, we then 3D printed this: <https://github.com/SANGERA2/smart-pi-cup/raw/refs/heads/main/cup%20render.mp4>

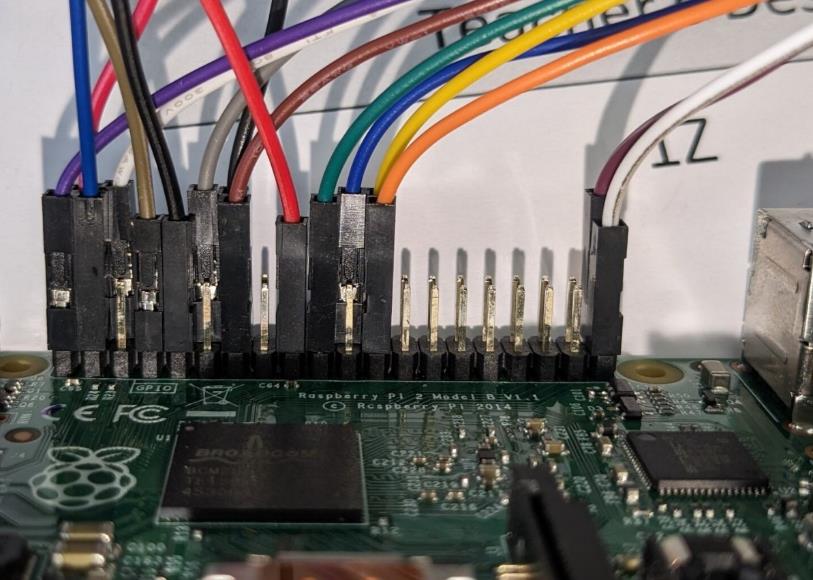
Then, once the coding had been done, we also soldered all the wires on to the mini Pi. Then we fitted the Pi and the wiring into the inside of the 3D printed cup. We were also able to use a metal cup for the interior so that no liquid was able to get onto the circuits and the wires would be protected. This cup can be removed to see the inner hardware and to clean the cup.

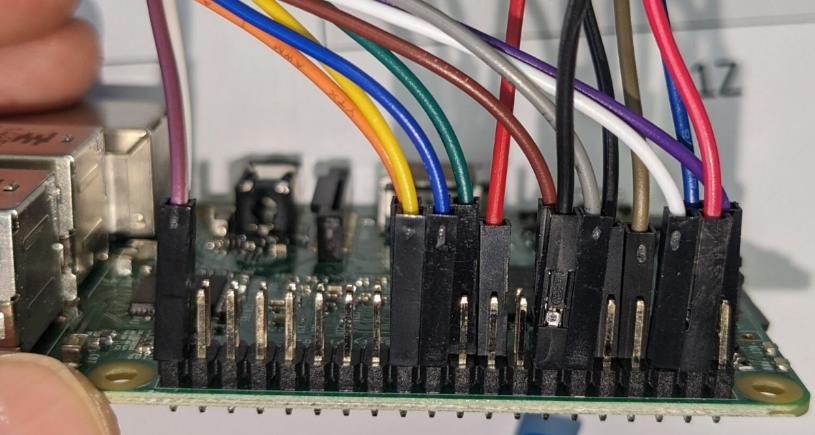
These images show the wiring we used; we first tested them on a normal sized pi to make sure everything worked

**Normal pi**

 birds eye view

closer up, angled birds eye view

view from one side of the pi

view from the other side of the pi

Use these lists to place the cables in the correct pins. You will need either female to female wires of male to female wires for testing using the normal sized pi, but will need to expose wire when soldering them on to the mini pi.

Temperature sensor

Black – ground – pin 9

Red – 3V3 – pin 17

Brown -GPIO4 – pin 7

Flow sensor

Black – ground – pin 14

Red – 5v – pin 4

Blue – GPIO3 – pin 5

Red button – on/off

This is connected to the battery pack instead of directly to the pi, the battery pack is then plugged into the pi

Black button- controls the screen

White – ground – pin 39

Purple – GPIO21 – pin 40

Screen

Grey – GPIO18 – pin 12

Brown – GPIO27 – pin 13

Blue – GPIO25 – pin 22

Yellow – GPIO8 – pin 24

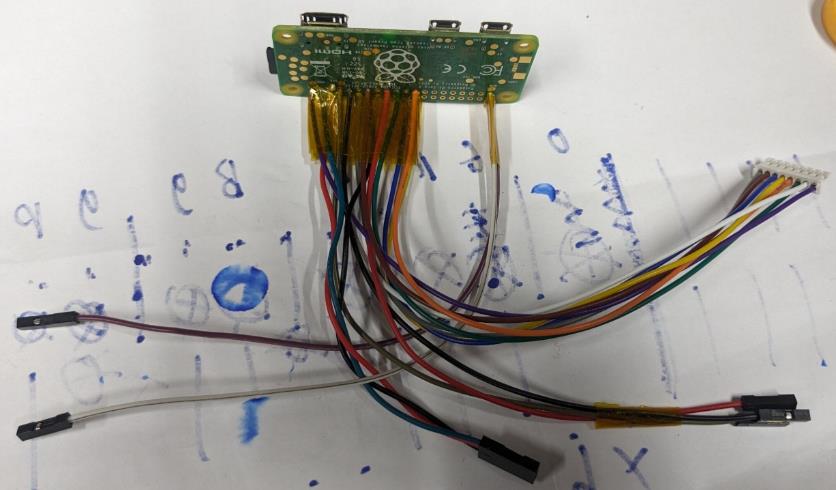
Orange – Gpio11 – pin 23

Green – GPIO10 – pin 19

White – ground – pin 6

Purple – 3V3 – pin 1

**Raspberry Pi Zero**

the wiring attached to the Raspberry Pi Zero

This is the empty 3D printed cup

This is both cups.

this is the Pi Zero attached to the cup

Screen embedded in the cup

All the wiring in the cup

This is the temperature sensor attached to the cup with Kapton tape to stop the temperature being affected by the outside air