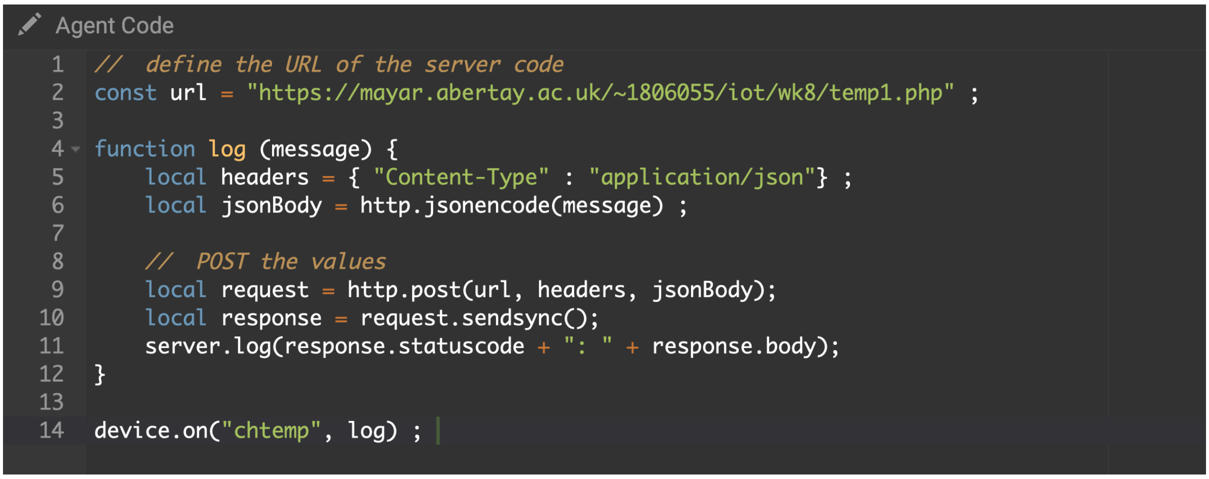
Week 7 Commentary

You should document the final program involving 4 sensors collecting data. A diagram may be useful as will the final program. Make sure you define correctly the values output in their correct units. Include the final printout of you can. Outline how you tested your program – does it output the correct values?

The program takes in input from the 4 sensors that are attached to the ElectricImp device, and displays them onto a website page, wherein there is code that decodes decoding them via JSON and PHP.

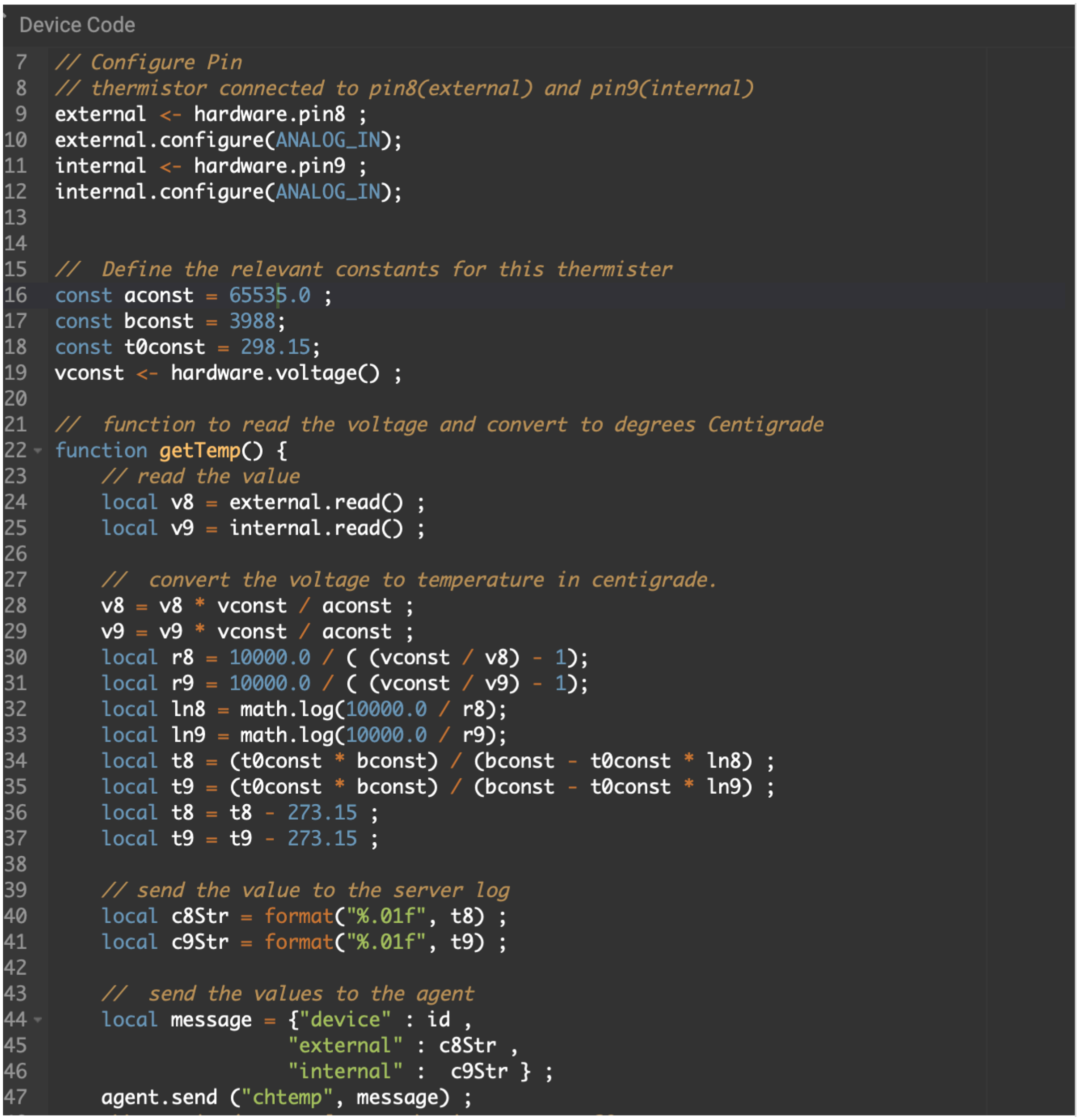


Agent Code

```
1   //  define the URL of the server code
2   const url = "https://mayar.abertay.ac.uk/~1806055/iot/wk8/temp1.php" ;
3
4   function log (message) {
5       local headers = { "Content-Type" : "application/json"} ;
6       local jsonBody = http.jsonencode(message) ;
7
8       //  POST the values
9       local request = http.post(url, headers, jsonBody);
10      local response = request.sendsync();
11      server.log(response.statuscode + ": " + response.body);
12  }
13
14  device.on("chtemp", log) ;
```

The above code shows the Agent Code for the program that is stored on the Imp that allows the data to be shown on a web page. The address to view the code is on line 2 of the above screenshot and is included in the code as it is needed so that the program has somewhere to output the sensor readings to. As you can see in lines 5 and 6, the program encapsulates a function called log that encodes the values into a JSON array, which is then sent via the lines 9 onwards.

For the Imp to get the readings from the sensor and turn that into useable data however, required more code, this time in the Device Code section of the ElectricImp development platform.



```
7   // Configure Pin
8   // thermistor connected to pin8(external) and pin9(internal)
9   external <- hardware.pin8 ;
10  external.configure(ANALOG_IN);
11  internal <- hardware.pin9 ;
12  internal.configure(ANALOG_IN);
13
14
15  //  Define the relevant constants for this thermister
16  const aconst = 65535.0 ;
17  const bconst = 3988;
18  const t0const = 298.15;
19  vconst <- hardware.voltage() ;
20
21  //  function to read the voltage and convert to degrees Centigrade
22  function getTemp() {
23      // read the value
24      local v8 = external.read() ;
25      local v9 = internal.read() ;
26
27      //  convert the voltage to temperature in centigrade.
28      v8 = v8 * vconst / aconst ;
29      v9 = v9 * vconst / aconst ;
30      local r8 = 10000.0 / ( (vconst / v8) - 1);
31      local r9 = 10000.0 / ( (vconst / v9) - 1);
32      local ln8 = math.log(10000.0 / r8);
33      local ln9 = math.log(10000.0 / r9);
34      local t8 = (t0const * bconst) / (bconst - t0const * ln8) ;
35      local t9 = (t0const * bconst) / (bconst - t0const * ln9) ;
36      local t8 = t8 - 273.15 ;
37      local t9 = t9 - 273.15 ;
38
39      // send the value to the server log
40      local c8Str = format("%.01f", t8) ;
41      local c9Str = format("%.01f", t9) ;
42
43      //  send the values to the agent
44      local message = {"device" : id ,
45                      "external" : c8Str ,
46                      "internal" :  c9Str } ;
47      agent.send ("chtemp", message) ;
```

Therefore, more code was needed, and this above screenshot shows the bulk of the Device Code that was used to both receive the raw data from the sensor, and to turn that into a usable temperature reading, specially turning the readings into degrees Centigrade/Celsius, which is a unit much more familiar to us. Lines 9-12 take input from the physical pins on the ElectricImp device, which are connected to the external sensor device. From there, lines 16-19 define constants that will be used in the conversion process further on in the code, and lines 22-25 read the internal and external voltages of the Imp device, utilising multiple sensor readings simultaneously.

The purpose of lines 28-37 is the formula that is used to convert the raw data reading into °C, and lines 40 and 41 send the 2 variables that are calculated at the end of that to the server log. The final 4 lines on the screenshot sends the values into the Agent Code (which then encodes into a JSON array, which I explained at the start of this document.

I managed to test this program first by using a control. I left the temperature sensor on the desk that I was using and then check the output that this gave. It gave a reading of 20.4 degrees Celsius. As there was not a thermostat in the room I was working in, there was no other way to check however conventional room temperatures vary between 20-22°C, and the reading the sensor gave was in line with this, therefore I accepted that the calculation has worked.

To show that the sensor was able to deviate from this, I placed my hand directly onto the thermistor, and after some time, temperature rose by a couple degrees, showing that the sensor is working as it should.