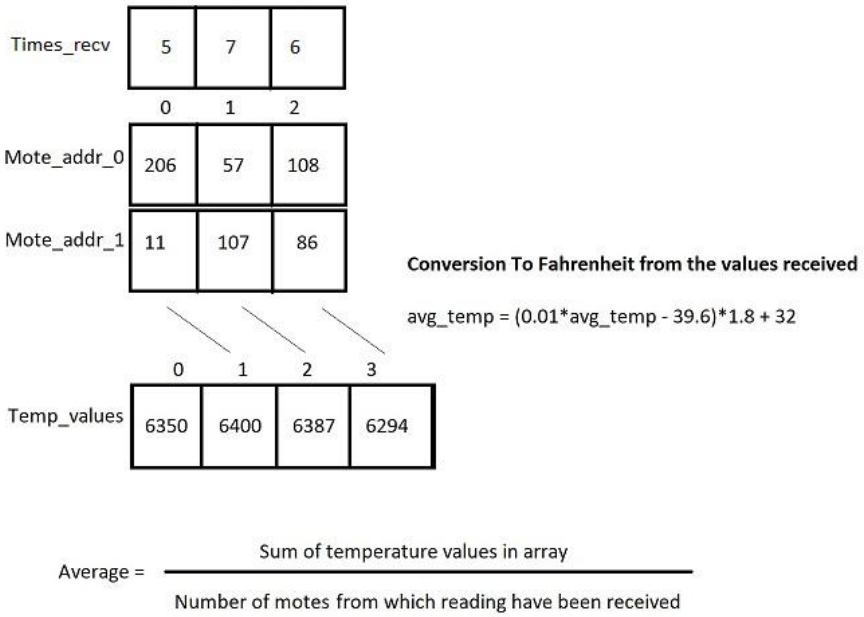
Finding average temperature of a room

Introduction

In this project, we find the average temperature of a room. Each mote collects temperature data from various other motes and averages it out by considering even the value the mote generated itself. Uses case can be when the motes are deployed in a fairly large room and we can go to any mote to find the average temperature of the room.

Algorithm



So, here's the explanation of what actually goes on (each mote does the same thing):

Temp_values array –

Mote gets the temperature from its own sensor and stores it in the first location in tem_values array. The temperature data from other motes is stored in subsequent locations in the array. We use this array to find the average temperature by looping through the array and taking sum of temperature values received from motes and dividing it by number of motes from which readings have been received.

The temperature conversion from the data received from motes to celcius was found in https://github.com/contiki-os/contiki/blob/master/dev/sht11/sht11.h

The conversion is T = -39.60 + 0.01*t where ‘t’ is the value we got from temperature sensor of motes and ‘T’ is the value in celcius. We then converted that value to Fahrenheit.

In our code, we use variable ‘i’ to track the number of motes from which data is received.

Motes_addr_0/1 array –

The function of these arrays is to keep track of addresses of motes from which data has been received. This is used to index the temp_values array. So, temperature values from one mote gets stored only in one designated location in the array. This is used to prevent mixing up of temperature values from the motes.

Times_recv –

This is used to keep track of how many times the values are received from the motes. This useful to know if the mote is no longer sending the data. So, we can remove the data of the mote which is no longer sending temperature values from the calculation tables (they are the temp_values and motes_addr_0/1 arrays)

Display using LEDs –

The 3 LEDs were used to display the temperature values as follows.

Red LED glowed as many times as the units place digit in the temperature value.

Green LED glowed as many times as the tens place digit in the temperature value.

Blue LED glowed as many times as the hundredth place digit in the temperature value.

Functionalities in our code

1. Our code can identify the motes address dynamically, as in we don’t need to specify the address of the motes from which we’re receiving data. Code takes care of that.

2. Its scalable as in we can have a system with say three motes deployed at first. If we add 10 more motes with same code, all the motes get average of values from all other motes and averaging is done perfectly. (In our code I’ve designed it for maximum of four motes. But, if I change the temp_values and motes_addr_0/1 array sizes to say 20, it’ll be capable of supporting up to 20 motes.)

3. Our code can detect if any motes have failed. We then discard the temperature values from those motes and continue to find the average. (In our code, I wait for 40 times to receive values from each mote. If I don’t get values after 40 seconds, I discard the values. We can change this value to our convenient time too.)

Results

The temperature value found in the room at my house was 74 F.

The temperature value found in the TLA was 76 F.

The outside temperature today in morning was 80 F.