AGRITECH MEETING - 6

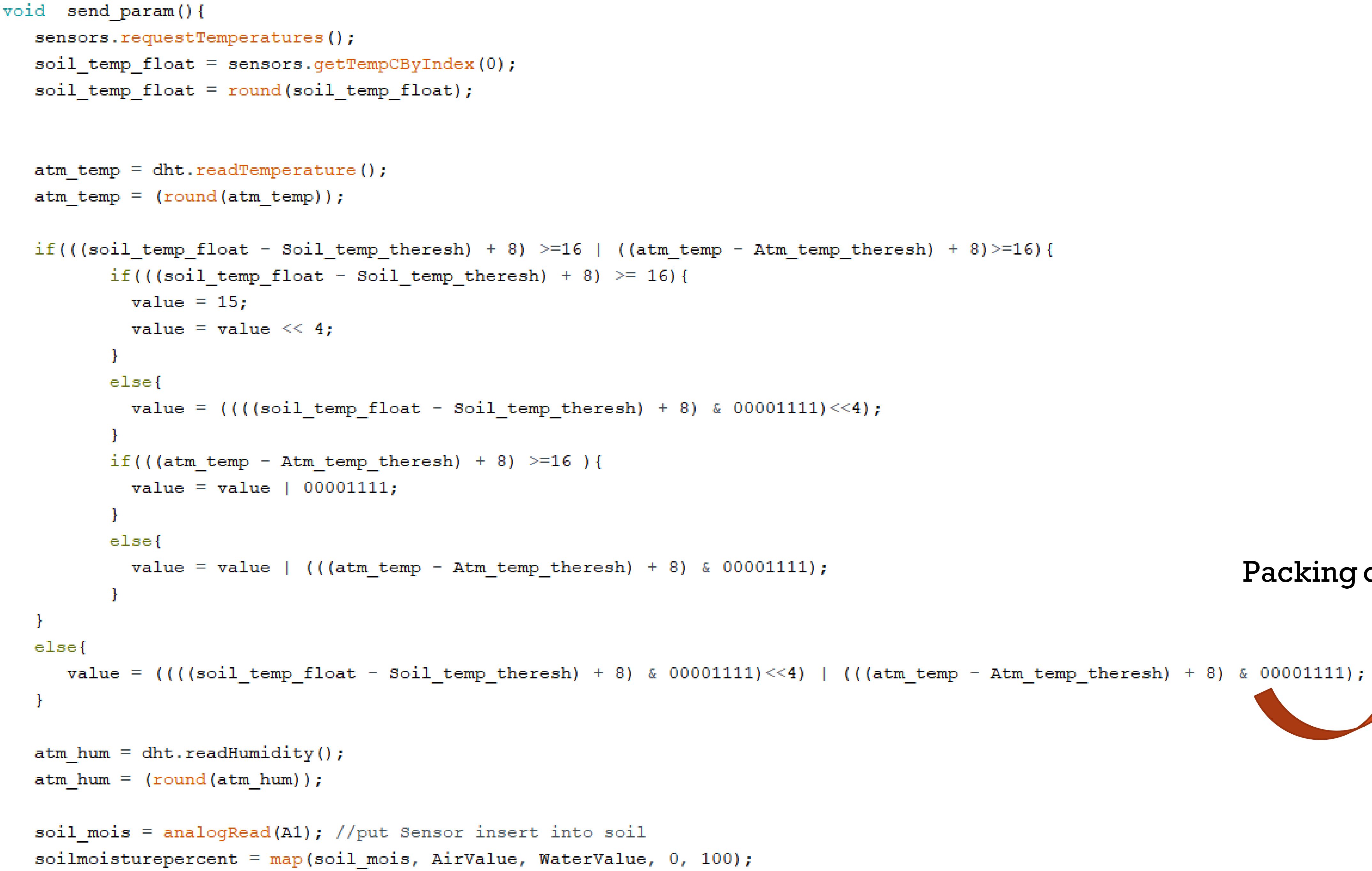
Prof Madhav Rao

DATA FORMATS

- Soil temperature : has precision of 2 - e.g. 30.06 this takes 15 bits. 5 bits for 30, 10bits for 0.06.
- Atmosphere temperature : has precision 1 – e.g. 29.7 this takes 15 bits. 5 bits for 29, 10 bits for 0.7
- Humidity : has precision 1 – e.g. 38.7 this takes 16 bits. 6 bits for 38, 10 bits for 0.7
- Soil moisture : It is analog data (range is from 0 to 1023) uses bits range from 1 to 9. But we are mapping it from 0 to 100 this takes bits from 1 to 7 bits.

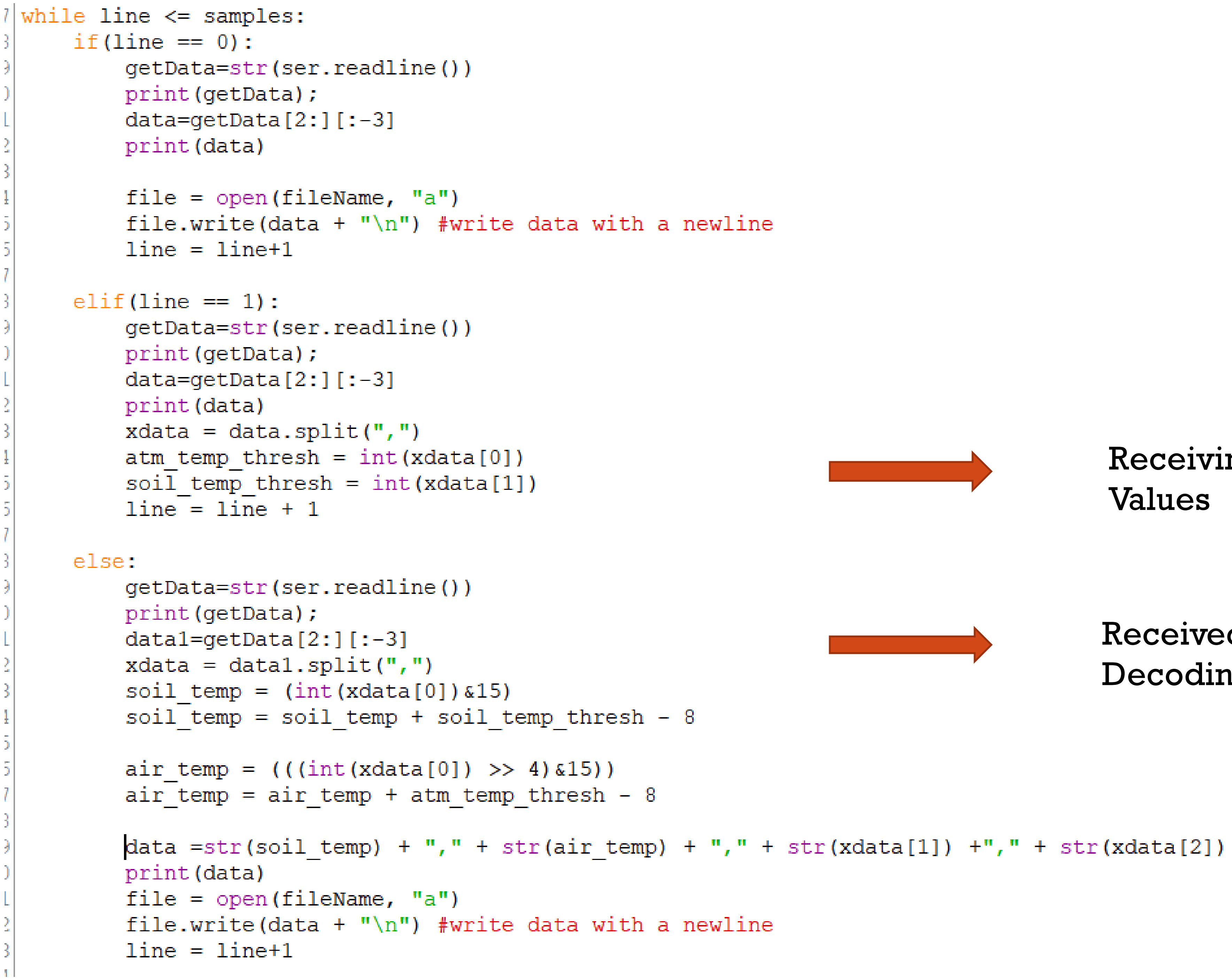
IDEA OF PACKING DATA

- If we don’t need precision of 1 or 2 then we can save lot of bits. If our message packet should have only 32bits, without precision we can pack 2 sets of data(soil temp and atmosphere temp) + 8bits(one data + 2bits can be used for something else). With precision we can pack only 1set of data (soil temp and atmosphere temp).
- Efficient way of sending data by reducing bits. If we know range of sensor value, we can manipulate it. For example, if we know temperature range varies in between (-7 to 7 from a certain value) then we can transmit just 4 bits in place of 6 bits at receiver side we add this certain value to that offset. By doing this we save 2 bits for each sensor value(without precision). By this method if the payload size is 32bit we can pack 4 sets of data(soil temp + atmosphere temp).



```
void  send_param(){
   sensors.requestTemperatures();
   soil_temp_float = sensors.getTempCByIndex(0);
   soil_temp_float = round(soil_temp_float);


   atm_temp = dht.readTemperature();
   atm_temp = (round(atm_temp));

   if(((soil_temp_float - Soil_temp_theresh) + 8) >=16 | ((atm_temp - Atm_temp_theresh) + 8)>=16){
          if(((soil_temp_float - Soil_temp_theresh) + 8) >= 16){
            value = 15;
            value = value << 4;
          }
          else{
            value = ((((soil_temp_float - Soil_temp_theresh) + 8) & 00001111)<<4);
          }
          if(((atm_temp - Atm_temp_theresh) + 8) >=16 ){
            value = value | 00001111;
          }
          else{
            value = value | (((atm_temp - Atm_temp_theresh) + 8) & 00001111);
          }
   }
   else{
      value = ((((soil_temp_float - Soil_temp_theresh) + 8) & 00001111)<<4) | (((atm_temp - Atm_temp_theresh) + 8) & 00001111);
   }

   atm_hum = dht.readHumidity();
   atm_hum = (round(atm_hum));

   soil_mois = analogRead(A1); //put Sensor insert into soil
   soilmoisturepercent = map(soil_mois, AirValue, WaterValue, 0, 100);
```

Packing of Data bits

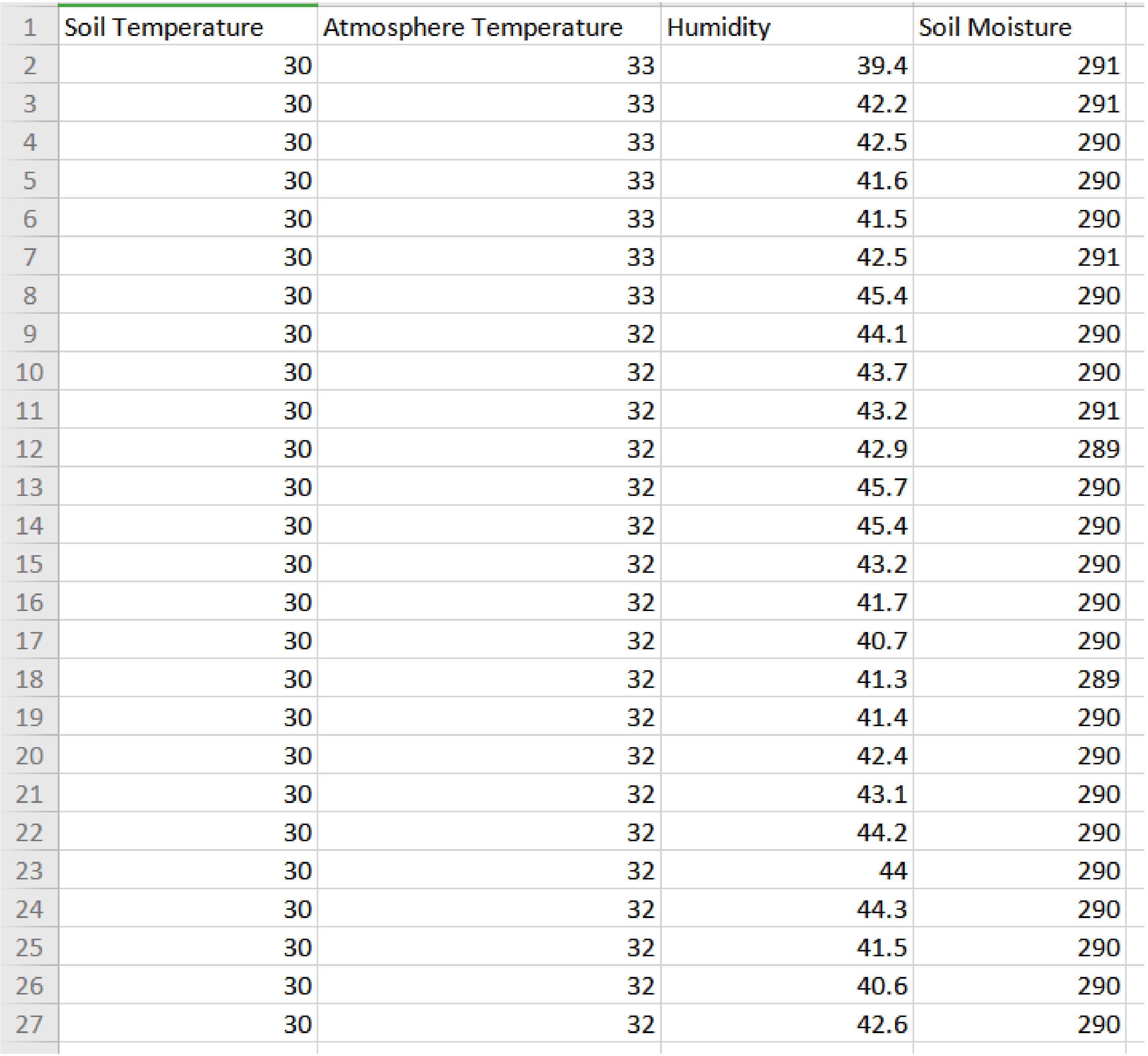


```python
while line <= samples:
    if(line == 0):
        getData=str(ser.readline())
        print(getData);
        data=getData[2:][:-3]
        print(data)

        file = open(fileName, "a")
        file.write(data + "\n") #write data with a newline
        line = line+1

    elif(line == 1):
        getData=str(ser.readline())
        print(getData);
        data=getData[2:][:-3]
        print(data)
        xdata = data.split(",")
        atm_temp_thresh = int(xdata[0])
        soil_temp_thresh = int(xdata[1])
        line = line + 1

    else:
        getData=str(ser.readline())
        print(getData);
        data1=getData[2:][:-3]
        xdata = data1.split(",")
        soil_temp = (int(xdata[0])&15)
        soil_temp = soil_temp + soil_temp_thresh - 8

        air_temp = (((int(xdata[0]) >> 4)&15))
        air_temp = air_temp + atm_temp_thresh - 8

        data =str(soil_temp) + "," + str(air_temp) + "," + str(xdata[1]) +"," + str(xdata[2])
        print(data)
        file = open(fileName, "a")
        file.write(data + "\n") #write data with a newline
        line = line+1
```

Receiving Threshold Values

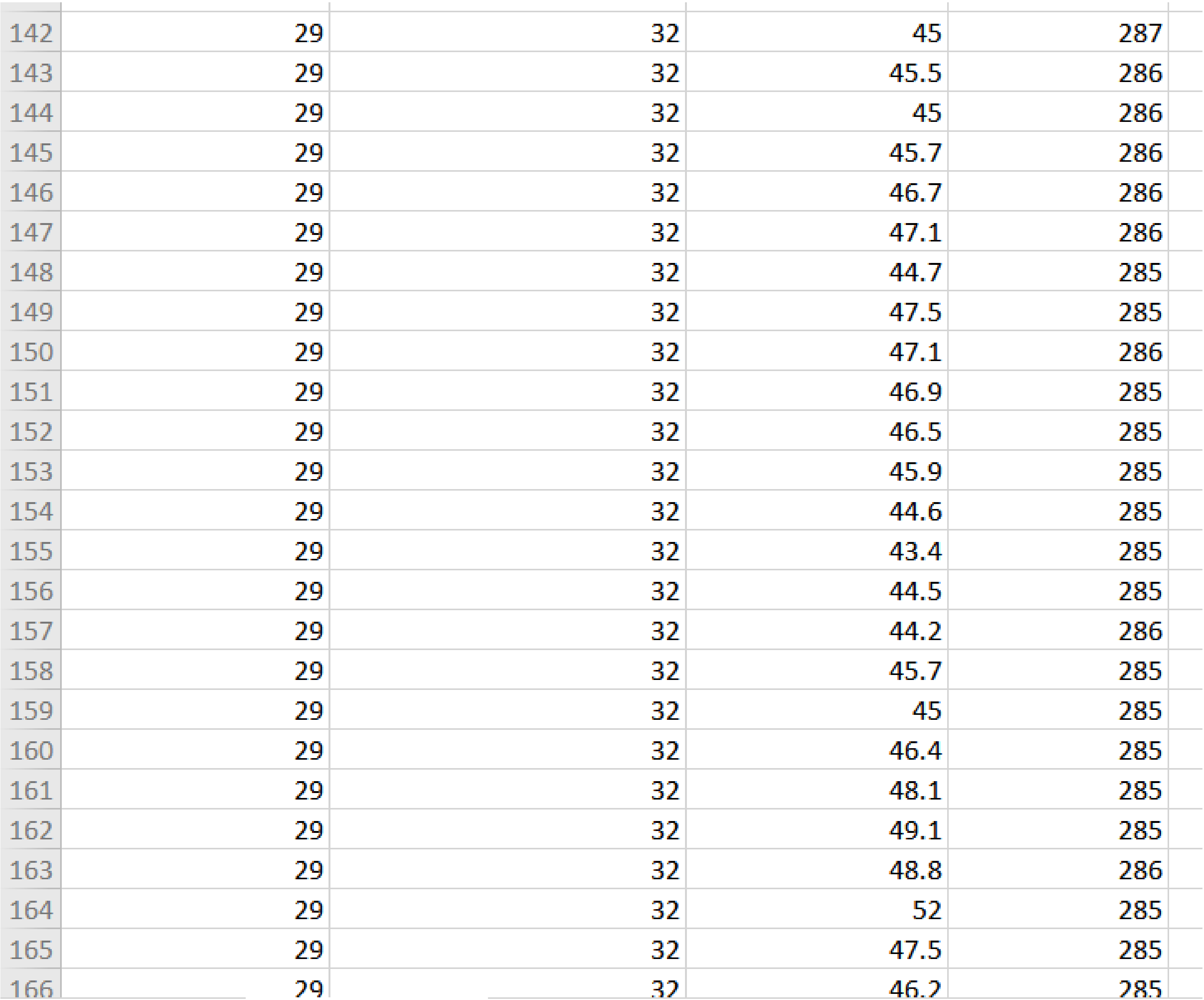
Received data and Decoding

- Receiver code for both cloud and Log data into Csv is almost same except the writing part so the above decoding part will be similar.
- Cannot have threshold value for Humidity and soil moisture like soil and atmosphere temperature as I was not able to predict in what range it can be.
- Need to decide for an encoding theorem to use for our bit packing data to get less bit errors. We can use general encoding schemes like Huffman encoding.
- Decoding part was working fine.



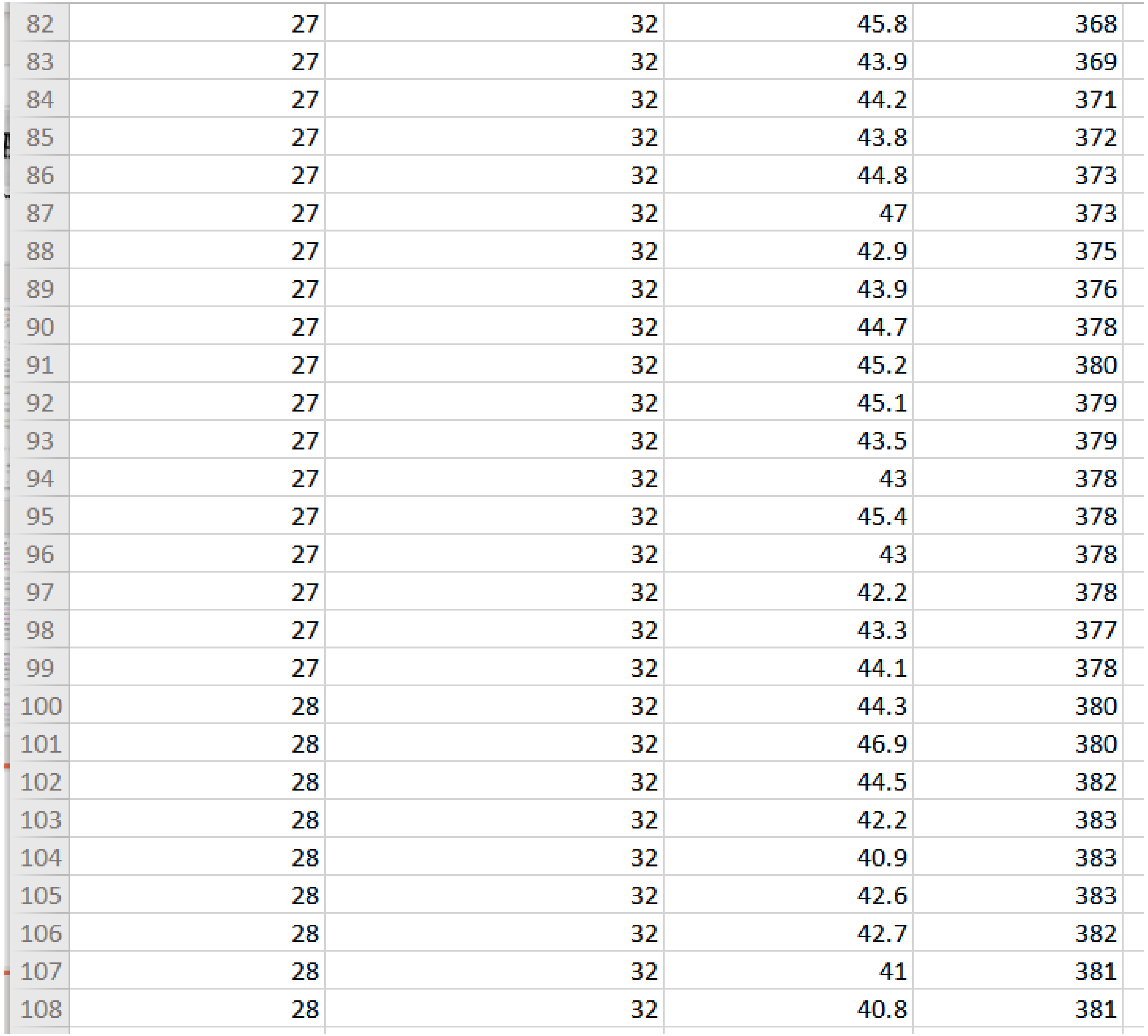
1	Soil Temperature	Atmosphere Temperature	Humidity	Soil Moisture
2	30	33	39.4	291
3	30	33	42.2	291
4	30	33	42.5	290
5	30	33	41.6	290
6	30	33	41.5	290
7	30	33	42.5	291
8	30	33	45.4	290
9	30	32	44.1	290
10	30	32	43.7	290
11	30	32	43.2	291
12	30	32	42.9	289
13	30	32	45.7	290
14	30	32	45.4	290
15	30	32	43.2	290
16	30	32	41.7	290
17	30	32	40.7	290
18	30	32	41.3	289
19	30	32	41.4	290
20	30	32	42.4	290
21	30	32	43.1	290
22	30	32	44.2	290
23	30	32	44	290
24	30	32	44.3	290
25	30	32	41.5	290
26	30	32	40.6	290
27	30	32	42.6	290

Data collected when the sensors placed in flower plot. Soil was moist(watered 20 mins before collecting data)



142	29	32	45	287
143	29	32	45.5	286
144	29	32	45	286
145	29	32	45.7	286
146	29	32	46.7	286
147	29	32	47.1	286
148	29	32	44.7	285
149	29	32	47.5	285
150	29	32	47.1	286
151	29	32	46.9	285
152	29	32	46.5	285
153	29	32	45.9	285
154	29	32	44.6	285
155	29	32	43.4	285
156	29	32	44.5	285
157	29	32	44.2	286
158	29	32	45.7	285
159	29	32	45	285
160	29	32	46.4	285
161	29	32	48.1	285
162	29	32	49.1	285
163	29	32	48.8	286
164	29	32	52	285
165	29	32	47.5	285
166	29	32	46.2	285

Data collected for wet soil Watered 5 mins before collecting data.



82	27	32	45.8	368
83	27	32	43.9	369
84	27	32	44.2	371
85	27	32	43.8	372
86	27	32	44.8	373
87	27	32	47	373
88	27	32	42.9	375
89	27	32	43.9	376
90	27	32	44.7	378
91	27	32	45.2	380
92	27	32	45.1	379
93	27	32	43.5	379
94	27	32	43	378
95	27	32	45.4	378
96	27	32	43	378
97	27	32	42.2	378
98	27	32	43.3	377
99	27	32	44.1	378
100	28	32	44.3	380
101	28	32	46.9	380
102	28	32	44.5	382
103	28	32	42.2	383
104	28	32	40.9	383
105	28	32	42.6	383
106	28	32	42.7	382
107	28	32	41	381
108	28	32	40.8	381

Data collected at evening – That part of the ground was watered morning