**At Raspberry pi Side**

***Receiving code***

#include <RF24/RF24.h>

#include <RF24Network/RF24Network.h>

#include <iostream>

#include <ctime>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int temp=1;

/\*\*

\* g++ -L/usr/lib main.cc -I/usr/include -o main -lrrd

\*\*/

// CE Pin, CSN Pin, SPI Speed

RF24radio(RPI\_BPLUS\_GPIO\_J8\_15,RPI\_BPLUS\_GPIO\_J8\_24, BCM2835\_SPI\_SPEED\_8MHZ);

RF24Network network(radio);

// Constants that identifies this node

const uint16\_t pi\_node = 0;

// Time between checking for packets (in ms)

const unsigned long interval = 2000;

// Structure of our message

struct message\_t {

float temperature;

float humidity;

float soil\_moisture;

} messagee;

int main(int argc, char\*\* argv)

{

// Initialize all radio related modules

radio.begin();

delay(5);

network.begin(90, pi\_node);

// Print some radio details (for debug purposes)

// radio.printDetails();

// printf("Ready to receive...\n");

// Now do this forever (until cancelled by user)

while(temp)

{

//get the latest network info

network.update();

// printf(".\n");

// Enter this loop if there is data available to be read,

// and continue it as long as there is more data to read

while ( network.available() ) {

RF24NetworkHeader header;

message\_t message;

// Have a peek at the data to see the header type

network.peek(header);

// We can only handle the Temperature type for now

if (header.type == 't') {

// Read the message

network.read(header, &message, sizeof(messagee));

messagee = message;

// Print it out

printf("Temperature received from node %i: %f \n", header.from\_node, messagee.temperature);

printf("Humidity received from node %i: %f \n", header.from\_node, messagee.humidity);

printf("soil moisture received from node %i: %f \n", header.from\_node, messagee.soil\_moisture);

if(messagee.soil\_moisture<300)

printf("soil moisture : Water");

if(messagee.soil\_moisture>300 && messagee.soil\_moisture<700)

printf("soil moisture : humid");

if(messagee.soil\_moisture>700 )

printf("soil moisture : dry");

} else {

// This is not a type we recognize

network.read(header, &message, sizeof(message));

printf("Unknown message received from node %i\n", header.from\_node);

}

}

// Wait a bit before we start over again

delay(2000);

}

// last thing we do before we end things

return 0;

}