Moisture Sensor Tx & Rx (2 Lands) :

Tx:

#include <RH\_ASK.h>

#ifdef RH\_HAVE\_HARDWARE\_SPI

#include <SPI.h> // Not actually used but needed to compile

#endif

// Initialize the RadioHead ASK driver

RH\_ASK driver;

// Pin definitions for the moisture sensors

const int sensorPinL1 = A0; // Moisture sensor 1 connected to A0

const int sensorPinL2 = A1; // Moisture sensor 2 connected to A1

void setup()

{

// Initialize RF driver

if (!driver.init())

{

#ifdef RH\_HAVE\_SERIAL

Serial.println("RF module initialization failed!");

#endif

while (1); // Halt program if initialization fails

}

// Initialize sensors

pinMode(sensorPinL1, INPUT);

pinMode(sensorPinL2, INPUT);

#ifdef RH\_HAVE\_SERIAL

Serial.begin(9600); // For debugging

#endif

}

void loop()

{

// Read moisture sensor values

int moistureL1 = analogRead(sensorPinL1);

int moistureL2 = analogRead(sensorPinL2);

// Create a string message in the format "L1:XXXX L2:YYYY"

char message[20];

snprintf(message, sizeof(message), "L1:%d L2:%d", moistureL1, moistureL2);

// Send the message via RF

driver.send((uint8\_t \*)message, strlen(message));

driver.waitPacketSent();

#ifdef RH\_HAVE\_SERIAL

// Debugging: Print the sent message to the Serial Monitor

Serial.print("Sent: ");

Serial.println(message);

#endif

// Wait for 2 seconds before sending the next data

delay(2000);

}

Rx:

#include <RH\_ASK.h>

#include <LiquidCrystal.h>

#ifdef RH\_HAVE\_HARDWARE\_SPI

#include <SPI.h> // Not actually used but needed to compile

#endif

// Initialize the RadioHead ASK driver

RH\_ASK driver;

// Initialize the LCD with appropriate pins

LiquidCrystal lcd(7, 8, 9, 10, 12, 13); // RS, EN, D4, D5, D6, D7

void setup()

{

    // Initialize the RF driver

    if (!driver.init())

    {

#ifdef RH\_HAVE\_SERIAL

        Serial.println("RF module initialization failed!");

#endif

        while (1); // Halt program if initialization fails

    }

#ifdef RH\_HAVE\_SERIAL

    Serial.begin(9600); // For debugging

#endif

    // Initialize the LCD

    lcd.begin(16, 2); // Set up the LCD's number of columns and rows

    lcd.setCursor(0, 0);

    lcd.print("Initializing...");

    delay(2000);

    lcd.clear();

    lcd.print("Ready to Receive");

    delay(2000);

    lcd.clear();

}

void loop()

{

    // Buffer to hold received data

    uint8\_t buf[RH\_ASK\_MAX\_MESSAGE\_LEN];

    uint8\_t buflen = sizeof(buf);

    // Check if a message is available

    if (driver.recv(buf, &buflen))

    {

        // Null-terminate the received data to convert it to a string

        buf[buflen] = '\0';

        // Parse the received data (assumes format "L1:XXXX L2:YYYY")

        int moistureL1 = 0, moistureL2 = 0;

        sscanf((char \*)buf, "L1:%d L2:%d", &moistureL1, &moistureL2);

        // Display parsed data on the LCD

        lcd.clear();

        lcd.setCursor(0, 0);

        lcd.print("Land 1:");

        lcd.print(moistureL1);

        lcd.setCursor(0, 1);

        lcd.print("Land 2:");

        lcd.print(moistureL2);

#ifdef RH\_HAVE\_SERIAL

        // Debugging: Print the received message and parsed data to the Serial Monitor

        Serial.print("Received: ");

        Serial.println((char \*)buf);

        Serial.print("Moisture L1: ");

        Serial.println(moistureL1);

        Serial.print("Moisture L2: ");

        Serial.println(moistureL2);

#endif

        // Add a small delay to allow the LCD to update

        delay(2000);

    }

}