Include all libraries

#include <SPI.h> , #include <OneWire.h>, #include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 0

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

RH\_ASK rf\_driver;

Initialize the function

float read\_mq7(void); , float read\_mq2(void); , float read\_ph(void);, float get\_temperature(void);

Initialize ASK Object

rf\_driver.init();

initialize the variables

float PH, Temperature, MQ7value,MQ2value;

String transmit ,transmit\_msg;

Read the PH value by calling the read\_ph () function

PH = read\_ph();

**A**

**A**

convert the data into string concinate the msg string and converted data to transmit

transmit = "PH value " + transmit\_msg;

Transmit the data character by character as per the mgs length

rf\_driver.send((uint8\_t \*)msg\_ph, strlen(msg\_ph));

Read the Temperature value by calling the get\_temperature() function

Temperature = get\_temperature()

convert the data into string concinate the msg string and converted data to transmit

transmit = "temperature value " + transmit\_msg;

Transmit the data character by character as per the mgs length

rf\_driver.send((uint8\_t \*)msg\_temperature, strlen(msg\_temperature));

Read the MQ7value by calling the read\_mq7()function

MQ7value = read\_mq7();

**B**

**B**

convert the data into string concinate the msg string and converted data to transmit

transmit = "MQ7 value " + transmit\_msg;

Transmit the data character by character as per the mgs length

rf\_driver.send((uint8\_t \*)msg\_mq7, strlen(msg\_mq7));

Read the MQ2value by calling the read\_mq2()function

MQ7value = read\_mq7();

converte the data into string concinate the msg string and converted data to transmit

transmit = "MQ2 value " + transmit\_msg;

Transmit the data character by character as per the mgs length

rf\_driver.send((uint8\_t \*)msg\_mq2, strlen(msg\_mq2));

Include dependant SPI Library

#include <RH\_ASK.h>

#include <SPI.h> , #include <LiquidCrystal.h>

Initialize ASK Object

rf\_driver.init();

Initialize the string to store received data

String str\_out;

LCD’s number of columns and rows

lcd.begin(20, 4); lcd.begin(20, 4);

Set buffer to size of expected message

uint8\_t buf[11];

uint8\_t buflen = sizeof(buf);

**A**

**A**

if (rf\_driver.recv(buf, &buflen))

**true**

**Flase**

Display the data not received

lcd.print("data not received" );

Display the received data on LCD

lcd.print(str\_out );

Links

1. <https://www.farnell.com/datasheets/1682209.pdf>

2. <https://www.elprocus.com/what-is-arduino-uno-r3-pin-diagram-specification-and-applications/>

3. <https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_>

4. <https://www.application-datasheet.com/pdf/dfrobot/509083/sen0161.html>

5. [https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/DS18B20.pdf](http://www.quick-teck.co.uk/ElectronicElement/eeList.php?typeId=97#title)

6. <https://lastminuteengineers.com/ds18b20-arduino-tutorial/>

7. <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/>

8. <https://components101.com/sensors/mq2-gas-sensor>

9. <https://microcontrollerslab.com/433mhz-rf-receiver-module-pinout-applications-arduino-examples/>

10. <https://components101.com/asset/sites/default/files/component_datasheet/433%20MHz%20RF%20Transmitter%20Module_0.pdf>

11. <https://www.elprocus.com/lcd-16x2-pin-configuration-and-its-working/#:~:text=The%20term%20LCD%20stands%20for,emitting%20diodes%20and%20seven%20segments>.

12. <https://www.elprocus.com/interface-lcd-liquid-crystal-display-using-arduino/>

13. <https://www.semiconductorforu.com/application-lcd-liquid-crystal-display/>

Algorithm

1. **Algorithm for Arduino code at measuring terminal**

**Step1:** start

**Step2:** include a radiohead library for Rf module interfaceStep1:

**Step3:** Include dependant SPI Library for communicating with transmitter

**Step4:** include a oneWire library for temperature sensor

**Step5**: Data wire is plugged into digital pin 0 on the Arduino

**Step6:** Setup a oneWire instance to communicate with any OneWire device

**Step7:** Pass oneWire reference to DallasTemperature

**Step8:** Create Amplitude Shift Keying Objec

**Step9:** initialize function to read carbon monoxide value

**Step10**: initialize function to read smoke gas value

**Step11**: initialize function to read PH value of solution

**Step12:** initialize function to read temperature value of solution

**Step13**: Initialize ASK Object

**Step14:** initialize the variables to store the input value of the sensors

**Step15:** call the function to measure the get the PH data

**Step16:**converte the data into string for transmitting through the rf module

**Step17**:concinate the msg string and converted data to transmit

**Step18:**Compose output character

**Step19:**transmit the data using the inbuild function of RH library

**Step20:**wait for the data to be send

**Step22:** call the function to measure the get the temperature data

**Step23**:converte the data into string for transmitting through the rf module

**Step24:**concinate the msg string and converted data to transmit

**Step25**:Compose output character

**Step26:**transmit the data using the inbuild function of RH library

**Step27**:wait for the data to be send

**Step29:** call the function to measure the get the MQ7 data

**Step30:**converte the data into string for transmitting through the rf module

**Step31**:concinate the msg string and converted data to transmit

**Step32:**Compose output character

**Step33:**transmit the data using the inbuild function of RH library

**Step34:**wait for the data to be send

**Step36**: call the function to measure the get the MQ2 data

**Step37**:converte the data into string for transmitting through the rf module

**Step38**:concinate the msg string and converted data to transmit

**Step39:**Compose output character

**Step40**:transmit the data using the inbuild function of RH library

**Step41**:wait for the data to be send

**Step42:** defining the function to read the ph value

**Step43**: defining the function to read the temperature value

**Step44:** defining the function to read the carbon monoxide value

**Step45:** defining the function to read the smoke gas value

**Step46**:END

**Algorithm for Arduino code at display terminal**

**Step1:** Include dependant SPI Library

**Step2:** Create Amplitude Shift Keying Object

**Step3:** include the liquid crystal diaply library

**Step4:** Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

**Step5**: string to store the recived data

**Step6:** Initialize ASK Object

**Step7:** set up the LCD’s number of columns and rows:

**Step8:** Clears the LCD screen

**Step9:** Set buffer to size of expected message

**Step10**: Check if received packet is correct size

**Step11:** set the cursor to column 0, line 1

**Step12**:if is true then print str\_out value

**Step13**:else print "data not received"

**Step14:** END