**TRANSMITTER SIDE CODING:**

#include "lcd.h"

#include "gps.h"

#include "gsm.h"

#include <ultrasonic.h>

#include<SoftwareSerial.h>

SoftwareSerial Serial(6, 7);

#include"dht11.h"

DHT dht;

bool leakage\_flag = false;

int sensorInterrupt = 0; // interrupt 0

int sensorPin = 2; //Digital Pin 2

unsigned int SetPoint = 400; //400 milileter

/\*The hall-effect flow sensor outputs pulses per second per litre/minute of flow.\*/

float calibrationFactor = 90; //You can change according to your datasheet

volatile byte pulseCount = 0;

float flowRate = 0.0;

unsigned int flowMilliLitres = 0;

37

unsigned long totalMilliLitres = 0;

unsigned long oldTime = 0;

ULTRASONIC U1;

#define D\_temp A0

#define ir\_pin 3

#define gas\_pin 4

bool refill\_flag = false;

bool gas\_flag = false;

void ultrasonic(void);

void setup()

{

Serial.begin(9600);

Serial.begin(9600);

lcd.begin(16, 2);

U1.begin(A1, A2);

pinMode(sensorPin, INPUT);

digitalWrite(sensorPin, HIGH);

pinMode(ir\_pin, INPUT);

dht.dht\_read(D\_temp);

lcd.clear();

38

lcd.setCursor(1, 0);

lcd.print("DRAINAGE\_BLOCK");

lcd.setCursor(3, 1);

lcd.print("MONITORING");

delay(2000);

lcd.clear();

attachInterrupt(sensorInterrupt, pulseCounter, FALLING); //you can use Rising or

Falling

//gps();

}

void loop()

{

if ((millis() - oldTime) > 1000) // Only process counters once per second

{

temperature();

lcd\_update();

ultrasonic();

lcd.setCursor(9, 0);

lcd.print("D:");

lcd.setCursor(11, 0);

int pin = digitalRead(ir\_pin);

39

lcd.print(pin);

if (pin == 0)

{

Serial.print('D');

gsm("1234567890", "DRAINAGE OPEN");

}

lcd.setCursor(13, 0);

lcd.print("G:");

lcd.setCursor(15, 0);

lcd.print(digitalRead(gas\_pin));

lcd.setCursor(0, 1);

if (digitalRead(gas\_pin) == 0)

{

Serial.print('B');

gsm("1234567890", "METHANE DETECTED");

}

lcd.print("F:");

detachInterrupt(sensorInterrupt);

//We also apply the calibrationFactor to scale the output based on the number of

pulses per second per units of measure (litres/minute in this case) coming from the

sensor.

40

flowRate = ((1000.0 / (millis() - oldTime)) \* pulseCount) / calibrationFactor;

oldTime = millis();

flowMilliLitres = (flowRate / 60) \* 1000;

totalMilliLitres += flowMilliLitres;

unsigned int frac;

lcd.setCursor(2, 1);

lcd.print(flowMilliLitres, DEC);

if (flowMilliLitres > 5)

{

Serial.print('E');

lcd.setCursor(4, 1);

lcd.print("flow blocked");

gsm("1234567890", "FLOW BLOCKED ");

}

}

pulseCount = 0;

attachInterrupt(sensorInterrupt, pulseCounter, FALLING);

}

//Insterrupt Service Routine

41

void pulseCounter()

{

// Increment the pulse counter

pulseCount++;

}

void temperature(void)

{

dht.dht\_read(D\_temp);

if (dht.temperature > 33)

{

Serial.print('A');

gsm("1234567890", "HIGH TEMPERATURE");

}

}

void lcd\_update(void)

{

lcd.setCursor(0, 0);

lcd.print("T:");

lcd.setCursor(2, 0);

lcd.print(dht.temperature);

42

delay(1000);

}

void ultrasonic(void)

{

lcd.setCursor(5, 0);

lcd.print("U:");

int cm = U1.ultra();

if (cm < 10)

{

Serial.print('C');

lcd.print("F");

gsm("1234567890", "DRAINAGE FULL");

}

else if (cm > 10 && cm < 30)

{

lcd.print("M");

}

else if (cm > 30)

{

lcd.print("E"); } }

43

**RECIEVER SIDE CODING:**

#include<SoftwareSerial.h>

SoftwareSerial Serial(2, 3); //RX TX

void iot\_send(String text);

char inchar;

# define D\_buzzer 7

void setup() {

Serial.begin(9600);

Serial.begin(9600);

digitalWrite(D\_buzzer, LOW);

pinMode(D\_buzzer, OUTPUT);

}

void loop() {

while (Serial.available() > 0)

{

inchar = (char)Serial.read();

Serial.println(inchar);

switch (inchar)

{

case'A':

44

digitalWrite(D\_buzzer, !digitalRead(D\_buzzer));

digitalWrite(D\_buzzer, HIGH);

delay(2000);

digitalWrite(D\_buzzer, LOW);

iot\_send("\*HIGH TEMPERATURE#");

break;

case'B':

digitalWrite(D\_buzzer, !digitalRead(D\_buzzer));

digitalWrite(D\_buzzer, HIGH);

delay(2000);

digitalWrite(D\_buzzer, LOW);

iot\_send("\*METHANE DETECTED#");

break;

case'C':

digitalWrite(D\_buzzer, !digitalRead(D\_buzzer));

digitalWrite(D\_buzzer, HIGH);

delay(2000);

digitalWrite(D\_buzzer, LOW);

iot\_send("\*DRAINAGE FULL#");

break;

45

case'D':

digitalWrite(D\_buzzer, !digitalRead(D\_buzzer));

digitalWrite(D\_buzzer, HIGH);

delay(2000);

digitalWrite(D\_buzzer, LOW);

iot\_send("\*DRAINAGE OPEN#");

break;

case'E':

digitalWrite(D\_buzzer, HIGH);

delay(2000);

digitalWrite(D\_buzzer, LOW);

iot\_send("\*DRAINAGE BLOCKED#");

break;}}

delay(1000); }

void iot\_send(String text){

for (int i = 0; i < text.length(); i++){

Serial.write(text[i]);

}

delay(3000);

}