Εικόνα που περιέχει λογότυπο

Περιγραφή που δημιουργήθηκε αυτόματα

Περιεχόμενα

[Περιγραφή προϊόντος 2](#_heading=h.gjdgxs)

[Τεχνική περιγραφή προϊόντος 3](#_heading=h.30j0zll)

[Τεχνικές Προδιαγραφές 4](#_heading=h.1fob9te)

[MQ-8(H) gas sensor: 4](#_heading=h.3znysh7)

[FSR402 force sensor: 4](#_heading=h.2et92p0)

[DHT11 temperature and humidity sensor: 4](#_heading=h.tyjcwt)

[Waveshare flame sensor: 4](#_heading=h.3dy6vkm)

[Buzzer 5V - Passive: 4](#_heading=h.1t3h5sf)

[LCD1602 I2C: 4](#_heading=h.4d34og8)

[Συνδεσμολογίες προϊόντος 4](#_heading=h.2s8eyo1)

[Buzzer 5](#_heading=h.17dp8vu)

[LCD - RGB 3.3V/5V 5](#_heading=h.3rdcrjn)

[MQ-8(H) gas sensor 5](#_heading=h.26in1rg)

[Waveshare flame sensor 6](#_heading=h.lnxbz9)

[FSR-402 Force Sensor 7](#_heading=h.35nkun2)

[DHT11 Temperature and Humidity Sensor 7](#_heading=h.1ksv4uv)

[Σχεδίαση δικτύου 7](#_heading=h.44sinio)

[Ψευδοκώδικας transmitter 8](#_heading=h.2jxsxqh)

[Ψευδοκώδικας receiver 9](#_heading=h.z337ya)

[Κώδικας προϊόντος 9](#_heading=h.3j2qqm3)

[Κώδικας ανιχνευτή 9](#_heading=h.1y810tw)

[Κώδικας ειδοποιητή 15](#_heading=h.4i7ojhp)

# Περιγραφή προϊόντος

Το σύστημα αισθητήρων KitchenGuard είναι η απόλυτη λύση ασφάλειας της κουζίνας. Με προηγμένους αισθητήρες για ανίχνευση αερίου, θερμοκρασίας, υγρασίας, βάρους και φλόγας, το KitchenGuard παρέχει ολοκληρωμένη κάλυψη για να διατηρεί την κουζίνα σας ασφαλή και να την αναβαθμίσει, εξοπλίζοντας της με τις πιο σύγχρονες τεχνολογικές δυνατότητες.

Σχεδιασμένο για επαγγελματικές κουζίνες, όπως αυτές που βρίσκονται σε ξενοδοχεία και εστιατόρια, το σύστημα KitchenGuard είναι εύκολο στην εγκατάσταση και χρήση. Απλώς συνδέστε τους αισθητήρες στην παρεχόμενη πλακέτα Arduino και είστε έτοιμοι! Το σύστημα παρέχει παρακολούθηση σε πραγματικό χρόνο των διαρροών αερίου, των αλλαγών θερμοκρασίας, των επιπέδων υγρασίας, των διακυμάνσεων του βάρους και της δραστηριότητας της φλόγας, προειδοποιώντας σας για πιθανούς κινδύνους πριν γίνουν σοβαροί.

Με το KitchenGuard, θα έχετε το κεφάλι σας ήσυχο γνωρίζοντας ότι η κουζίνα σας προστατεύεται όλο το εικοσιτετράωρο. Είτε βράζετε νερό, είτε τηγανίζετε φαγητό ή ψήνετε ένα κέικ, οι αισθητήρες θα εντοπίσουν τυχόν πιθανά προβλήματα και θα σας ειδοποιήσουν αμέσως, ώστε να μπορείτε να λάβετε μέτρα για να αποτρέψετε μια καταστροφή.

Εκτός από τους συμπεριλαμβανόμενους αισθητήρες, το KitchenGuard μπορεί επίσης να ενσωματωθεί με άλλες έξυπνες συσκευές, όπως έξυπνους λαμπτήρες ή ηχεία, για να παρέχει ακόμη μεγαλύτερη λειτουργικότητα και έλεγχο. Επιπρόσθετα, με τη δυνατότητα προσαρμογής του συστήματος για να καλύψει τις συγκεκριμένες ανάγκες σας, το KitchenGuard είναι η τέλεια λύση για κάθε επαγγελματική κουζίνα που λαμβάνει σοβαρά υπόψη την ασφάλεια.

Επενδύστε στο KitchenGuard σήμερα και διατηρήστε την κουζίνα σας ασφαλή και αναβαθμισμένη.

# Τεχνική περιγραφή προϊόντος

Το KitchenGuard είναι μια έξυπνη συσκευή ασφαλείας κουζίνας που χρησιμοποιεί έναν συνδυασμό αισθητήρων για τον εντοπισμό πιθανών κινδύνων ασφαλείας στην κουζίνα. Η συσκευή είναι εξοπλισμένη με αισθητήρες θερμοκρασίας, υγρασίας, αερίου, φλόγας και δύναμης, οι οποίοι συνεργάζονται για να παρακολουθούν το περιβάλλον της κουζίνας σε πραγματικό χρόνο.

Ο αισθητήρας θερμοκρασίας LM35 του KitchenGuard μετρά τη θερμοκρασία της κουζίνας, επιτρέποντάς της να ανιχνεύει υπερθέρμανση και πιθανούς κινδύνους πυρκαγιάς. Ο αισθητήρας θερμοκρασίας και υγρασίας DHT11 μπορεί να χρησιμοποιηθεί για την ανίχνευση διαρροών νερού, ενώ ο αισθητήρας αερίου MQ-8(H) είναι ικανός να ανιχνεύει επικίνδυνες διαρροές αερίου.

Ο αισθητήρας δύναμης FSR402 χρησιμοποιείται για την ανίχνευση της παρουσίας κατσαρόλας σε μια σόμπα, επιτρέποντας στη συσκευή να ειδοποιεί τους χρήστες εάν μια κατσαρόλα δεν έχει τοποθετηθεί σωστά ή λείπει. Ο αισθητήρας φλόγας Waveshare μπορεί να ανιχνεύσει φλόγες, επιτρέποντας στον KitchenGuard να ανιχνεύει πυρκαγιές και να ειδοποιεί τους χρήστες να αναλάβουν δράση.

Η οθόνη LCD1602 I2C του KitchenGuard εμφανίζει πληροφορίες σε πραγματικό χρόνο σχετικά με το περιβάλλον της κουζίνας, συμπεριλαμβανομένης της θερμοκρασίας, της υγρασίας και τυχόν ανιχνευμένων κινδύνων. Η συσκευή είναι επίσης εξοπλισμένη με βομβητή που μπορεί να χρησιμοποιηθεί για να παρέχει ηχητικές ειδοποιήσεις στους χρήστες όταν εντοπίζεται κίνδυνος.

Συνολικά, το KitchenGuard παρέχει μια ολοκληρωμένη λύση για την ασφάλεια της κουζίνας, επιτρέποντας στους χρήστες να εντοπίζουν και να ανταποκρίνονται σε πιθανούς κινδύνους προτού γίνουν σοβαρό πρόβλημα.

# Τεχνικές Προδιαγραφές

## MQ-8(H) gas sensor:

* Gas detection range: 100-10000ppm
* Sensitivity: 1.0ppm
* Operating voltage: 5V DC
* Power consumption: 150mA
* Interface: Analog output

## FSR402 force sensor:

* Force sensing range: 0-100g
* Sensitivity: 0.2V/g
* Operating voltage: 0V to 5V
* Operating temperature range: -30°C to +70°C
* Interface: Analog output

## DHT11 temperature and humidity sensor:

* Temperature measurement range: 0°C to +50°C
* Humidity measurement range: 20%RH to 90%RH
* Accuracy: ±2°C, ±5%RH
* Operating voltage: 3V to 5V
* Interface: 1-Wire interface

## Waveshare flame sensor:

* Flame detection range: 760-1100nm
* Sensitivity: adjustable via potentiometer
* Operating voltage: 3.3V to 5V
* Operating current: <20mA
* Interface: Digital output

## Buzzer 5V - Passive:

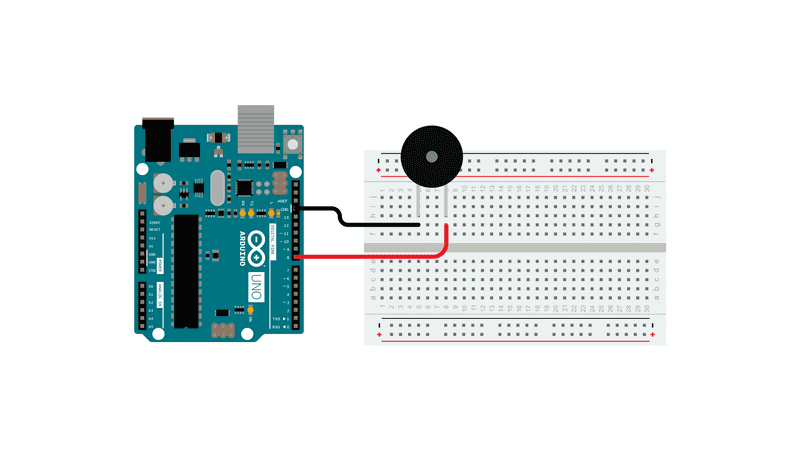
* Operating voltage: 5V DC
* Sound level: 85dB at 10cm
* Operating current: <30mA
* Interface: Digital output

## LCD1602 I2C:

* Display: 16 characters x 2 lines
* Operating voltage: 5V DC
* Interface: I2C

# Συνδεσμολογίες προϊόντος

## Buzzer



## LCD - RGB 3.3V/5V

Εικόνα που περιέχει ηλεκτρονικό εξάρτημα, στοιχείο κυκλώματος, ηλεκτρονικός μηχανικός, παθητικό στοιχείο κυκλώματος

Περιγραφή που δημιουργήθηκε αυτόματα

## MQ-8(H) gas sensor

Εικόνα που περιέχει ηλεκτρονικές συσκευές, ηλεκτρονικός μηχανικός, ηλεκτρονικό εξάρτημα, στοιχείο κυκλώματος

Περιγραφή που δημιουργήθηκε αυτόματα

Σημείωση : Αντί Α0 🡪 Α5

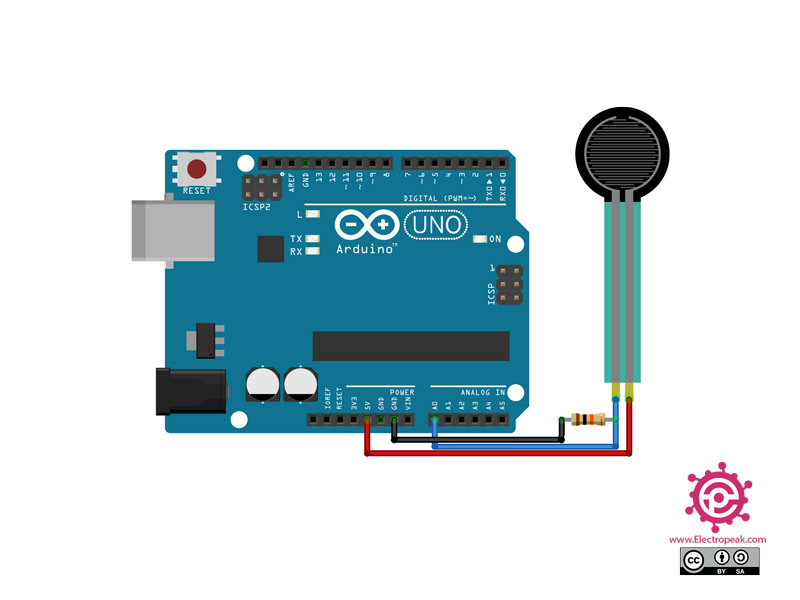
## Waveshare flame sensor

Εικόνα που περιέχει κύκλωμα, ηλεκτρονικός μηχανικός, ηλεκτρονικές συσκευές, ηλεκτρονικό εξάρτημα

Περιγραφή που δημιουργήθηκε αυτόματα

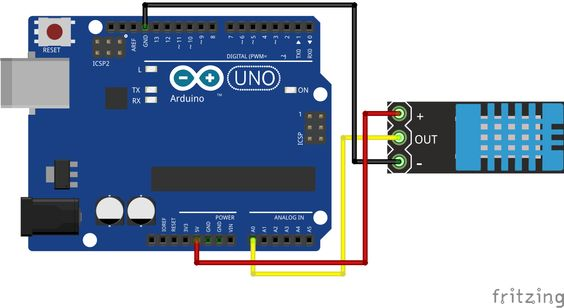
Σημείωση: αντί Α0 🡪 Α2

## FSR-402 Force Sensor



## DHT11 Temperature and Humidity Sensor

Σημείωση: αντί Α0 🡪 Α1



# Σχεδίαση δικτύου

Με το KitchenGuard κάθε κόμβος βρίσκεται σε κάθε κουζίνα, όπου με τους αισθητήρες κάνει τις κατάλληλες μετρήσεις για δύναμη, αέρια, φωτιά, θερμοκρασία, υγρασία. Σκοπός είναι να υπάρχει έλεγχος της καλής λειτουργίας από όλες τις κουζίνες ενός εστιατορίου, οπότε χρειάζονται αυτές οι μετρήσεις στην κάθε μια.

Με αυτές τις μετρήσεις γίνονται διάφοροι υπολογισμοί και μέσω των υπολογισμών ξεχωρίζουμε 6 περιπτώσεις σύμφωνα με τις οποίες, ο κόμβος αποθηκεύει διαφορετική μέτρηση κάθε φορά στην μνήμη του. Η ταχύτητα δειγματοληψίας (Baud rate) είναι 9600 samples/sec.

Έπειτα κάθε κόμβος (πομποί) επικοινωνεί με τον έναν συγκεκριμένο κόμβο (δέκτης)  που χρησιμοποιούμε για τον έλεγχο. Στέλνει τις απαραίτητες μετρήσεις ανάλογα με την περίπτωση και ο κόμβος ελέγχου τις αξιοποιεί.

Η επικοινωνία επιτυγχάνεται με το σύστημα Aloha και RF ασύρματες συσκευές. Δηλαδή όταν η μέτρηση είναι έτοιμη, ο κόμβος στέλνει αίτημα στον δέκτη και αν η επικοινωνία γίνει σωστά και ο δέκτης λάβει την τιμή, στέλνει αναγνώριση. Ο κάθε κόμβος αξιοποιεί ένα συγκεκριμένο τμήμα του χρόνου.

## Ψευδοκώδικας transmitter

If sensors begin

   Print (MQ2 warming up!)

End if

Loop

  Print (Begin. Sensor value)

  Read temperature

  Print (Temperature is temp\_var °C)

  Read force

  Print (Force is force\_var)

  Read gas

  Print (Gas value is gas\_var)

  Read flame

  If there is flame

    Print (Close fire)

    Else if there is distant fire

    Print (Distant fire)

    Else

    Print (No fire)

  End if

Read humidity

Print (Humidity is humi\_var%)

If temperature high

  If force high

    Status is 1

    Sensor\_value is temperature

  Else

    Status is 2

    Sensor\_value is temperature

Else if gas high

  Status is 3

  Sensor\_value is gas

Else if there is fire

  Status is 4

  Sensor\_value is 000

Else if humidity high

  Status is 5

  Sensor\_value is humidity

Else

  Status is 6

  Sensor\_value is 0

End if

Print (Status is status\_var. Sensor value is sensor\_value)

If sensor\_value >100

 Save "Status sensor\_value" on a string variable

Else

 Save "Status 0sensor\_value" on a string variable

End if

Send saved string

## Ψευδοκώδικας receiver

Screen show (KitchenGuard Actication)

Get string data

Status is data[0]

Sensor\_value  is data[2:4]

Cases for status

Case 0

Screen show (All good)

Case 1

Screen show (Water boiling Pot on place)

Case 2

Screen show (Water boiling Pot missing)

Buzzer

Case 3

Screen show (Gas leak)

Buzzer

Case 4

Screen show (Fire! Check kitchen)

Buzzer

Case 5

Screen show (Water leak)

Buzzer

Case 6

Screen show (Device problem. Contact us)

End case

# Κώδικας προϊόντος

## Κώδικας ανιχνευτή

//#include "pitch.h"

//#include "LiquidCrystal.h"

//#include "Waveshare\_LCD1602\_RGB.h"

#include "DHT.h"

#include <RF22.h>

#include <RF22Router.h>

#define MY\_ADDRESS 12

#define DESTINATION\_ADDRESS 5

RF22Router rf22(MY\_ADDRESS);

//

//// Defintions for the humidity sensor

#define DHTPIN 7

#define DHTTYPE DHT11

DHT dhtsensors(DHTPIN, DHTTYPE);

/\* variables for the LCD screen \*/

//int Contrast=75;

//LiquidCrystal lcd(8, 9, 5, 4, 3, 7);

//Waveshare\_LCD1602\_RGB lcd(16,2);  //16 characters and 2 lines of show

//int r,g,b,t=0;

float temp\_value\_2;

boolean successful\_packet = false;

// to distinguish the notes, set a minimum time between them.

const int force\_sensor = A1 ; //force sensor

const int gas\_sensor = A5 ; // gas sensor

const int lcd\_output = 6 ; // LCD display

const int flame\_sensor = A2; // flame sensor

int force\_value ; // values of force sensor

float gasValue ;  // values of gas sensor

//for flame

const int sensorMin = 0;     //  sensor minimum

const int sensorMax = 1024;  // sensor maximum

int sensor\_value = 0 ; //the value of the sensor we are going to send

void setup() {

  Serial.begin(9600);

  if (!rf22.init())

    Serial.println("RF22 init failed");

  // Defaults after init are 434.0MHz, 0.05MHz AFC pull-in, modulation FSK\_Rb2\_4Fd36

  if (!rf22.setFrequency(436.0))

    Serial.println("setFrequency Fail");

  rf22.setTxPower(RF22\_TXPOW\_20DBM);

  //1,2,5,8,11,14,17,20 DBM

  //rf22.setModemConfig(RF22::OOK\_Rb40Bw335  );

  rf22.setModemConfig(RF22::GFSK\_Rb125Fd125);

  //modulation

  // Manually define the routes for this network

  rf22.addRouteTo(DESTINATION\_ADDRESS, DESTINATION\_ADDRESS);

  // analogWrite(lcd\_output,Contrast);

  // begin the humidity sensor

  dhtsensors.begin();

  Serial.println("MQ2 warming up!");

  //delay(20000); // allow the MQ2 (gas sensor) to warm up

}

void loop() {

  /\* we have 7 statuses

     0 --> Everything ok

     1 --> Water boiling with pot

     2 --> Water boiling without pot

     3 --> Gas leak

     4 --> Fire

     5 --> Water leak

     6 --> Issue with device

     These will be used to send the correct case to our receiver \*/

  int status = 6 ;

  bool issue = false ;

 Serial.println("begin") ;

 Serial.println(sensor\_value) ;

  // int temp\_adc\_val;

  // int temp\_val;

  int  temp\_value\_2 = dhtsensors.readTemperature(); /\* Read Temperature \*/

  Serial.print("Temperature: ");

  Serial.print(temp\_value\_2);

  Serial.print("°C ~ ");

  //  temp\_val = (temp\_adc\_val \* 4.88); /\* Convert adc value to equivalent voltage \*/

  //  temp\_val = (temp\_val/10); /\* LM35 gives output of 10mv/°C \*/

  //

  //  /\* Priting the temperature values  \*/

  // Serial.print("Temperature = ");

  // Serial.print(temp\_val);

  // Serial.print(" Degree Celsius\n");

  //   force sensor

  force\_value = analogRead(force\_sensor);  // read the input on analog pin 0

  Serial.print("Force = ");

  Serial.println(force\_value); // print out the value you read

  gasValue = analogRead(gas\_sensor); // read analog input pin 0

  Serial.print("Gas Value: ");

  Serial.println(gasValue);

  int flame\_value  = analogRead(flame\_sensor); // read the analog input for flame sensor

  int range = map(flame\_value,  sensorMin, sensorMax, 0, 3);

  // range value:

  switch (range) {

  case 0:    // A fire closer than 1.5 feet away.

    Serial.println("\*\* Close  Fire \*\*");

    break;

  case 1:    // A fire between 1-3 feet away.

    Serial.println("\*\*  Distant Fire \*\*");

    break;

  case 2:    // No fire detected.

    Serial.println("No  Fire");

    break;

  }

//  Serial.println("flame:") ;

//  Serial.println(flame\_sensor) ;

   // read humidity and print it

  int humi  = dhtsensors.readHumidity();

  Serial.print("Humidity: ");

  Serial.print(humi);

  Serial.print("%");

  //

  if( temp\_value\_2 > 26 ) // Water Boiling, Pot placement  (TODO with more sensors : Stove )

  {

    sensor\_value = temp\_value\_2 ;

    //issue = true ;

    if( force\_value > 600) // Pot on place

    {

      status = 1 ;

    }

    else // Pot not on place

    {

      status = 2 ;

    }

  }

  else if( gasValue > 500) { // Gas leak  TODO with specific gas sensors ( e.g CO ) + temp : kitchen conditions

    //issue = true ;

    sensor\_value = gasValue ;

    status = 3 ;

  }

  else if ( range == 0 ) {  // Fire

    //issue = true ;

    sensor\_value = 000 ;

    status = 4 ;

  }

  else if( humi > 60 ){

    sensor\_value = humi ;

    status = 5  ;

  }

  else {

      status = 0 ;

      sensor\_value = 0 ;

  }

  Serial.println("status is") ;

  Serial.println(status) ;

  Serial.println("sensors value is") ;

  Serial.println(sensor\_value) ;

  // send the values to receiver

  char status\_read[RF22\_ROUTER\_MAX\_MESSAGE\_LEN];

  uint8\_t status\_send[RF22\_ROUTER\_MAX\_MESSAGE\_LEN];

  memset(status\_read, '\0', RF22\_ROUTER\_MAX\_MESSAGE\_LEN);

  memset(status\_send, '\0', RF22\_ROUTER\_MAX\_MESSAGE\_LEN);

  if (sensor\_value>100)

  {

  sprintf(status\_read, "%d %d", status, sensor\_value);

  }

  else

   {

  sprintf(status\_read, "%d 0%d", status, sensor\_value);

  }

  status\_read[RF22\_ROUTER\_MAX\_MESSAGE\_LEN - 1] = '\0';

  memcpy(status\_send, status\_read, RF22\_ROUTER\_MAX\_MESSAGE\_LEN);

  successful\_packet = false;

  Serial.println(status\_read);

  while (!successful\_packet)

  {

    if (rf22.sendtoWait(status\_send, sizeof(status\_send), DESTINATION\_ADDRESS) != RF22\_ROUTER\_ERROR\_NONE)

    {

      Serial.println("sendtoWait failed");

    }

    else

    {

      successful\_packet = true;

     // Serial.println("sendtoWait Succesful");

    }

  }

  delay(1000);

}

## Κώδικας ειδοποιητή

#include <RF22.h>

#include <RF22Router.h>

#include "pitch.h"

#include "Waveshare\_LCD1602\_RGB.h"

#define MY\_ADDRESS 5

RF22Router rf22(MY\_ADDRESS);

const int buzzer = 4 ;

char strNumber[2];

char sensor\_value[4] ;

//e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.

int noteDuration = 1000 / 4 ;

String second\_line ;

// to distinguish the notes, set a minimum time between them.

// the note's duration + 30% seems to work well:

int pauseBetweenNotes = noteDuration \* 1.30;

Waveshare\_LCD1602\_RGB lcd(16,2);  //16 characters and 2 lines of show

int r,g,b,t=0;

void setup() {

  Serial.begin(9600);

   // initialize LCD

  lcd.init();

  lcd.setCursor(0,0);

  lcd.send\_string("KitchenGuard");

  lcd.setCursor(0,1);

  lcd.send\_string("Activation");

  if (!rf22.init())

    Serial.println("RF22 init failed");

  // Defaults after init are 434.0MHz, 0.05MHz AFC pull-in, modulation FSK\_Rb2\_4Fd36

  if (!rf22.setFrequency(436.0))

    Serial.println("setFrequency Fail");

  rf22.setTxPower(RF22\_TXPOW\_20DBM);

  //1,2,5,8,11,14,17,20 DBM

  rf22.setModemConfig(RF22::GFSK\_Rb125Fd125);

  //modulation

  delay(2000);

}

void loop() {

  uint8\_t buf[RF22\_ROUTER\_MAX\_MESSAGE\_LEN];

  char incoming[RF22\_ROUTER\_MAX\_MESSAGE\_LEN];

  memset(buf, '\0', RF22\_ROUTER\_MAX\_MESSAGE\_LEN);

  memset(incoming, '\0', RF22\_ROUTER\_MAX\_MESSAGE\_LEN);

  uint8\_t len = sizeof(buf);

  uint8\_t from;

  int received\_value = 0;

  if (rf22.recvfromAck(buf, &len, &from))

  {

    buf[RF22\_ROUTER\_MAX\_MESSAGE\_LEN - 1] = '\0';

    memcpy(incoming, buf, RF22\_ROUTER\_MAX\_MESSAGE\_LEN);

    //Serial.println(incoming);

    //Serial.print("got request from : ");

    //Serial.println(from, DEC);

    received\_value = atoi((char\*)incoming);

    strNumber[0] = incoming[0];  // Assign the character to the first element of the array

    strNumber[1] = '\0';      // Null-terminate the array

    int status = atoi(strNumber) ;

    //Serial.println(status);

    sensor\_value[0] = incoming[2];

    sensor\_value[1] = incoming[3];

    sensor\_value[2] = incoming[4];

    sensor\_value[3] = '\0' ;

    second\_line = "Value:" + String(sensor\_value) + "pos:" + from;

    Serial.println("2 is") ;

    Serial.println(incoming[2]) ;

    Serial.println("3 is")  ;

    Serial.println(second\_line) ;

    switch(status){

      case 0:

        lcd.clear() ;

        lcd.setCursor(0, 0);

        //lcd.print("All good :)");

        lcd.send\_string("All good :)");

        break ;

      case 1:

        lcd.setCursor(0, 0);

        lcd.send\_string("Water Boiling");

        //lcd.print("Water Boiling");

        lcd.setCursor(0, 1);

        //lcd.print("Pot on place");

        lcd.send\_string("Pot on place");

        break ;

      case 2:

        lcd.setCursor(0, 0);

        //lcd.print("Water Boiling");

        lcd.send\_string("Water Boiling");

        lcd.setCursor(0, 1);

        //lcd.print("Pot missing");

        lcd.send\_string(second\_line.c\_str());

        tone(buzzer,NOTE\_B7, noteDuration); // play a tone for prefined time

        delay(pauseBetweenNotes);

        // stop the tone playing:

        noTone(buzzer);

        break ;

      case 3:

        lcd.setCursor(0, 0);

        //lcd.print("Gas leak!!!");

        lcd.send\_string("Gas leak!!!");

        lcd.setCursor(0, 1);

        //lcd.print("Check kitchen!");

        lcd.send\_string(second\_line.c\_str());

        tone(buzzer,NOTE\_DS8, noteDuration);

        delay(pauseBetweenNotes);

        noTone(buzzer);

        break ;

      case 4:

        lcd.setCursor(0, 0);

        //lcd.print("Fire!!!");

        lcd.send\_string("Fire!!!");

        lcd.setCursor(0, 1);

        //lcd.print("Check kitchen!");

        lcd.send\_string("Check kitchen!");

        tone(buzzer,NOTE\_DS6, noteDuration);

        delay(pauseBetweenNotes);

        noTone(buzzer);

        break ;

      case 5:

        lcd.setCursor(0, 0);

        //lcd.print("Water leak!");

        lcd.send\_string("Water leak!");

        lcd.setCursor(0, 1);

        //lcd.print("Check kitchen!");

        lcd.send\_string(second\_line.c\_str());

        tone(buzzer,NOTE\_DS6, noteDuration);

        delay(pauseBetweenNotes);

        noTone(buzzer);

        break ;

      case 6:

        lcd.setCursor(0, 0);

        //lcd.print("Water leak!");

        lcd.send\_string("Device problem");

        lcd.setCursor(0, 1);

        //lcd.print("Check kitchen!");

        lcd.send\_string("Contact Us!");

        break ;

    }

    delay(1000);

  }

  lcd.clear() ;

}