



Universidad del País Vasco  
Euskal Herriko Unibertsitatea

BILBOKO  
INGENIARITZA  
ESKOLA  
ESCUELA  
DE INGENIERÍA  
DE BILBAO

INDUSTRIA INGENIARITZA TEKNIKOKO ATALA

SECCIÓN INGENIERÍA TÉCNICA INDUSTRIAL

**INDUSTRIA ELEKTRONIKAREN ETA AUTOMATIKAREN  
INGENIERITZAKO GRADUA  
GRADU AMAIERAKO LANA  
2016 / 2017**

**ENTZUMEN URRITASUNA DUTEN PERTSONEI ZUZENDUTAKO SOINU-  
-SEINALEEN SISTEMA ANTZEMALE/ ADIERAZLE BATEN DISEINU ETA  
INPLEMENTAZIOA**

**2. DISEINUA**

**IKASLEAREN DATUAK**

IZENA: IGOR

ABIZENAK: LLONA AGIRRE

SIN.:

DATA: 2017/04/25

**ZUZENDARIAREN DATUAK**

IZENA: NEKANE

ABIZENAK: AZKONA ESTEFANIA

SAILA: TEKNOLOGIA ELEKTRONIKOA

SIN.:

DATA: 2017/04/25

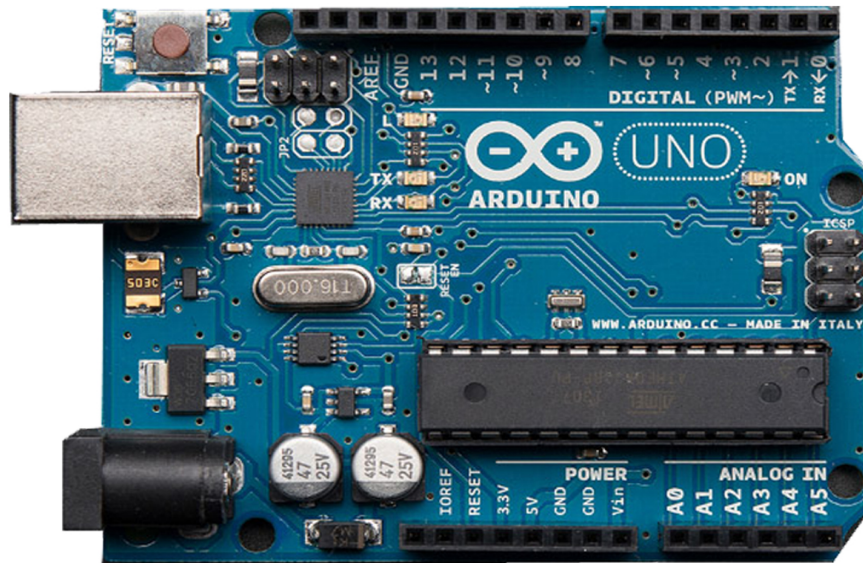
## **AURKIBIDEA**

- 1. Hardwarea**
- 2. Muntaia fisikoa/ prototipoak**
- 3. Softwarea**
- 4. Bukaerako diseinua**
- 5. Hobekuntza proposamenak**

## 1. HARDWAREA

### Arduino Uno

Bukaerako diseinuan Arduino UNO garapen plaka bi erabili dira.



1.1. Irudia. *Arduino UNO plaka*

### Ezaugarri teknikoak

- *Mikrokontrolagailua: ATmega328P*
- *Funtzionamendu voltaia: 5V*
- *Elikadura voltaia (gomendagarria): 7-12V*
- *Elikadura voltaia (limitea): 6-20V*
- *I/O pin digitalak: 14 (horietako 6 PWM irteera ahalbidetu)*
- *PWM I/O digitalak: 6*
- *Sarrera analogikoak: 6*

- *DC korrontea I/O pinetatik:* 20mA
- *DC korrontea 3.3 V-eko pinetik:* 50mA
- *Flash memoria:* 32kB (ATmega328P), horietatik 0.5kB bootloader-ak erabiliak
- *SRAM:* 2kB (ATmega328P)
- *EEPROM:* 1kB (ATmega328P)
- *Osziladore printzipala:* 16MHz
- *LED inetgratua:* 13. atakari esleitua
- *Dimentsio fisikoak:* Luzera( 68.6mm), zabalera( 53.4mm), zama (25g)

### **Mikrofonoa (Breakout plaka)**

UNO plakarekin bateragarria den Adafruit etxeko MAX4466 electret mikrofono-amplifikagailua aukeratu da.



1.2. Irudia. MAX4466 mikrofonoa

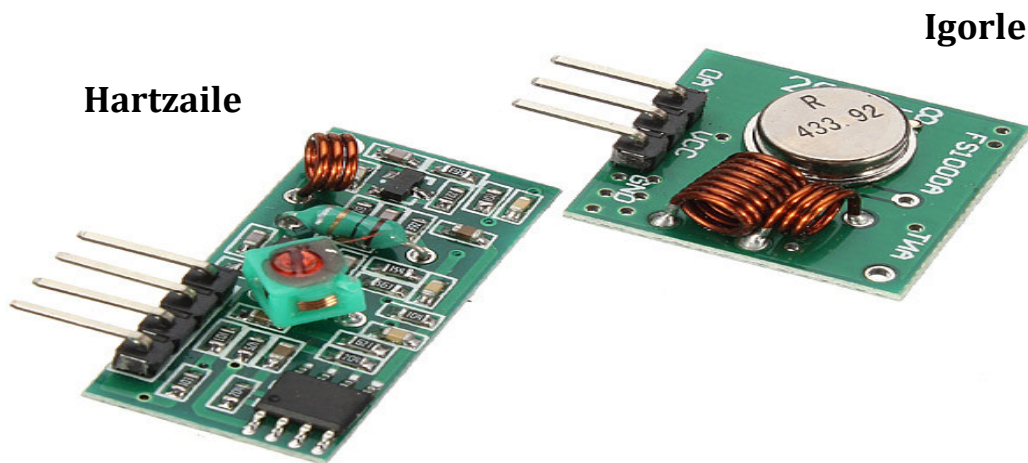
## Ezaugarri teknikoak

- *Irabazpena*: gain from 25x to 125x. Irabazpena ( $A_v$ )  
200Vpp-etik hasita, 1Vpp baino gehiagokora.
- *VCC konexioa*: Elikadura voltaia 2.4 - 5 V tartean, korrante zuzenera
- *GND konexioa*: lurrerako konexioa.
- *OUT pina*: irteera analogikoa\*

\*Oharra: Irteerako seinaleak,  $VCC/2$  balioko DC seinale akoplamedua du.

Arduino UNO plakak korrante zuzena soilik onartzen duenez. Jatorrizko seinalearen erreferentzia  $VCC/2$  balio bat mugitu edo desplazatu egiten da, balio negatiboen diskriminazioa ekiditeko.

## RF moduluak



1.3. Irudia. *RF moduluak*

## Ezaugarri teknikoak

### ZIRKUITU IGORLEA

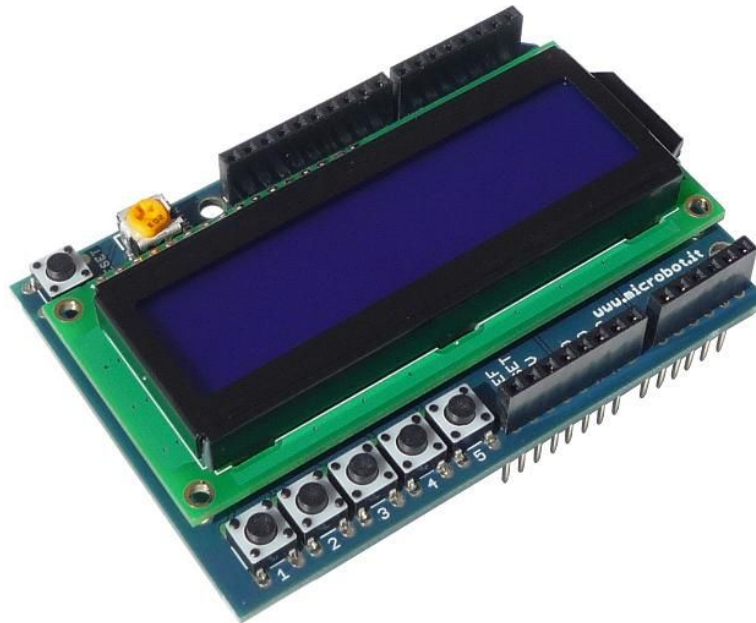
- *Tansmizio irismena*: 20- 200m (elikaduraren voltaiaren arabera)
- *Lan voltaia*: 3.5-12V DC
- *Data transferentzia abiadura*: 4kB/s
- *Transmizio potentzia*: 10mW
- *Igorpen maiztasuna*: 433MHz
- *Pinak*: VCC, GND eta DataIn (datuen sarrera)
- *Dimentsioak*: 19mm x 19mm

### ZIRKUITU HARTZAILEA

- *Harrapaketa banda*: 433MHz
- *Lan voltaia*: 5V DC
- *Korronte ez operatiboa*: 4mA
- *Sentikortasuna*: -105DB
- *Pinak*: VCC, GND eta 2x DataOUT (datuen irteera)
- *Dimentsioak*: 30 x 14 x 7 mm

## LCD shield modulua

Aurretik *Memoria* dokumentuan aipatu bezala, Arduino plakari gainezarri dakioken *LCD shield* modulua erabiliko da.



1.4. Irudia. LCD shield

## Ezaugarriak

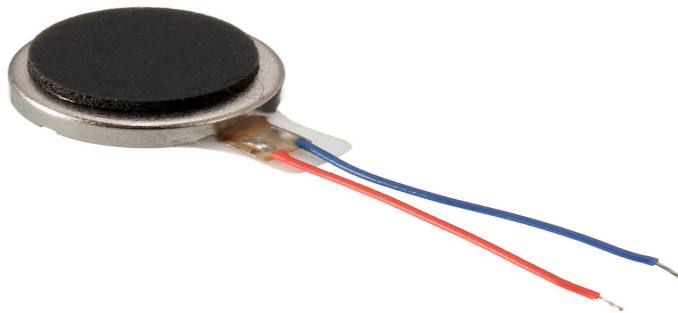
16 zutabe eta 2 lerroko konfigurazioa duen LCDa.

UNO plakara zuzenean txertatzeko diseinatuta dagoenez, konexio guztiak kointzidentek dira (Elikadura, GND, I/O atakak,..)

Arduino UNOaren 2, 3, 4, 5, 6 eta 7 ataka digitalak (irteera bezala konfiguratu) LCD-aren kontrolerako.

- *I/O digitalak*: 7.pina (RS erregistroen aukeraketa); 6.pina ( *Enable*); 2, 3, 4, 5 pinak ( data *nibble-en* kontrolerako.)
- Sarrera analogikoak: A0. LCD shield-ak integratzen dituen pultsagailuak (*switch-ak*) ataka honetara mapaturik daude.

## **DC motorea (ERM)**



1.5. Irudia. ERM motorea

### **Ezaugarriak**

*Tentsio operatiboa: 2.5-3.8V*

*Atari tentsio balioa: 2.3V*

*Korrante nominala: 80mA*

*Biraketa abiadura nominala: 12000 rpm*

*Karga erresistiboa: 75  $\Omega$*

*Bibrazioaren anplituadea (G): 0.8*

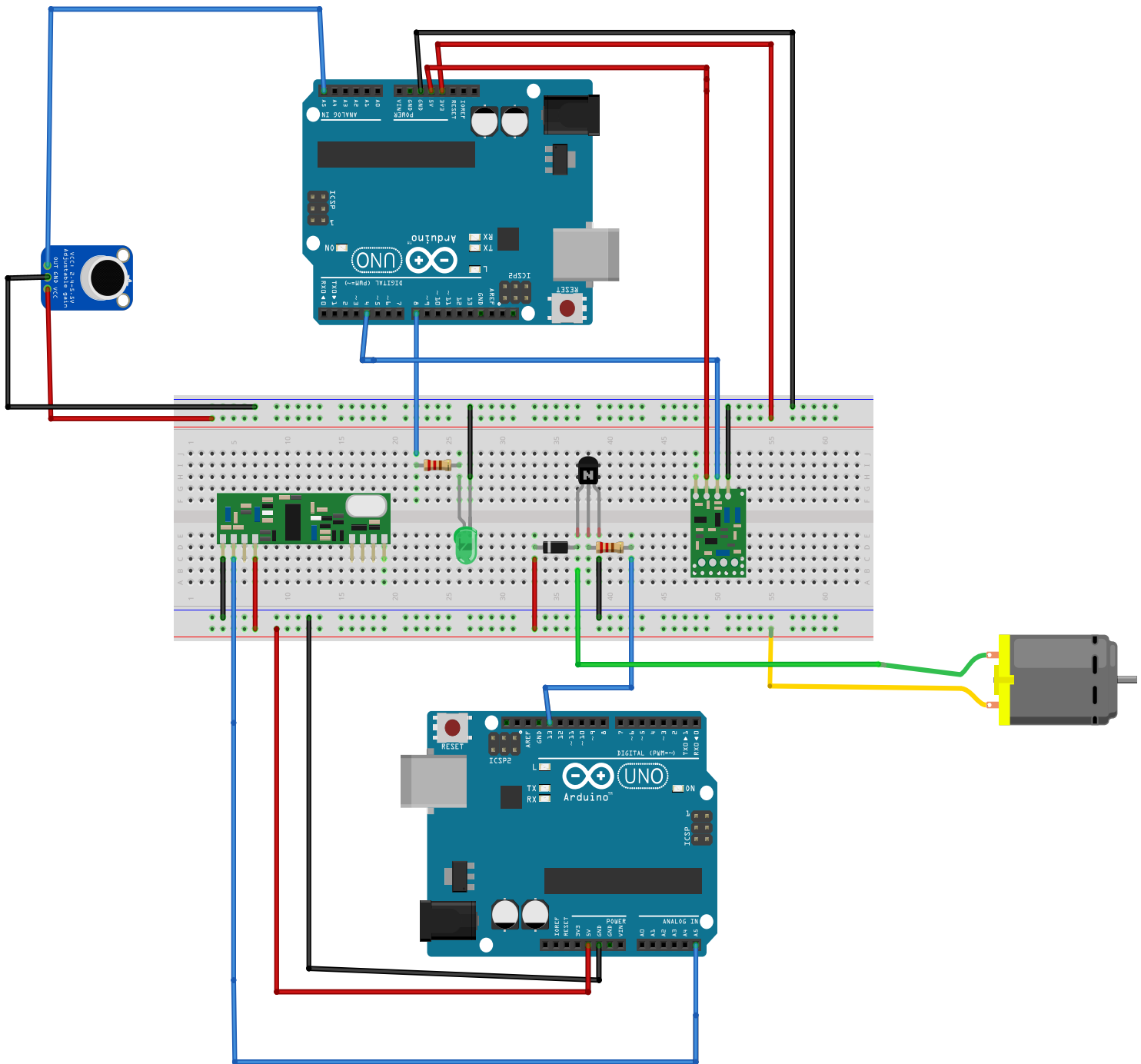
*Dimentsioak: bastidorearen diametroa (3.4mm), zama (1.2g)*

### **Diseinua osatzeko bestelako osagaiak**

Erresistore zeramikoak, LED-ak (funtzionamendu egokiaz ohartarazteko solik) ,  
pultsagailuak (*switch push button*), bakelitazko aurre estainatutako plakak, protoboard-  
ak, kuprezko harilak.



## 2. MUNTAIA FISIKOA/ PROTOTIPOAK



fritzing

### 2.1. Irudia. Muntaiaren eskema\*

\***Oharra:** Goiko irudiak, bukaerako diseinuaren prototipo bat erakusten du. Muntaiak definitiboak ez du konfigurazio berbera edukiko. (ERM motorraren kitzikapenerako ez da beharrezkoa irteerako seinalearen amplifikazioa egitea, goiko kasuan ez bezela).

### 3. SOFTWAREA

Arduino plataformak, ohiko sistema eragile nagusietan (*Windows, Linux, MacOS*) exekutatu daitezkeen berezko IDE-a (garapen ingurune integratua) eskeintzen du. Software askea den aldetik, IDE-aren azken bertsio eguneratua Arduinoren webgune ofizialetik eskuratu daiteke; eta kodea irekia denez, behin ordenagailu baten instalatuta, hura arakatu eta moldatu daiteke.

Garapen ingurunea bera Java lengoaian eraikita dago, nahiz eta aurretik diseinatutako *Processing* garapen ingurunean oinarritzen den. IDE-aren antzera *sketch*-ak (erabiltzaileak idatzitako programak) idazteko erabiliko den Arduino lengoia ere *Processing*-en oinarritzen da, baina aldiz, C/C++ -an oinarrituta.

*Sketch*-ak aipatzean aproposa dirudi arduino lengoian idatzitako programek jarraitzen duten eskema laburki azaltzea.

Hurrengo irudiak, arduino *sketch* baten egitura orokorra irudikatzen du:

```
#include <XXYY.h> // liburutegiak kargatu
#define xxyz 4 // konstanteak definitu
const

int XXYYZZ; //aldagien deklarazioak
boolean

void setup()
{
  // uC -aren AURRE KONFIGURAZIOA
  // parametroen balio lehenetsiak hasieratu
  // komunikazioak hasieratu, moduluak gaitu
  // sarrera-irteerak definitu...
}
void loop ()
{
  // BEGIZTA NAGUSIA.
  // programaren exekuzio ziklikoa hemen idatziko da
}
}
```

#### 3.1. Irudia. *Sketch* baten egitura

## **Liburutegiak**

Liburutegiak zeregin bereziak dituzten kode konplexuen garapena ikaragarri errazten duten instrukzio multzo ordenatuak dira.

Edozein programa baten betekizun zehatzak edo funtzio bereziak kudeatu behar badira (hala nola: modulu zehatz bat edo gehiagoren kontrola, modulu horiekin komunikazioa, algoritmo konplexuen erabilera programan integratzeko, memoriaren optimizazioa, etab.) liburutegien erabilera erabat lagungarria izan ohi da.

Liburutegiak programa konplexuen idazketa errazten dute, kontrolagailuaren ezaugarri guztiak sakontasunean aztertzeak suposatuko lukeen lan-karga arintzen dutelako (kontrolagailuen parametro guztiak kudeatzea eta modulu bakoitzarekin duten elkarrekintzaren eagitza zehatza izateak, ikerketa lan nahikoa eta denbora dedikazioa eskatzen du). Logikoaenez, liburutegiak *sketch*-etan integratuz errekurtsoak optimizatzen dira.

Arduinoren IDE-ak zenbait liburutegi ofizial aurrez integraturik dakartza, eta hauekin batera instalatuko da. Liburutegi hauek (motore edo serbomotoreen kontrola, memoriaren idazketa, komunikazio protokolo ezagunak,... eta antzeko betekizunak kudeatzeko baliogarriak dira). Bestelaz ere, badira liburutegi ez ofizialak eta hauek erabiltzeko aukera dago, betiere behin IDE-ak (eta konpilatzaileak) ulertu ditzan, beharrezko instalaziorako pauso batzuk jarraituta.

Orokorrean liburutegiak osatzen dituzten fitxategi motak *.ccp* (iturburu kodeak) eta *.h* (goiburuak) dira.

Edozein sketch baten liburutegi bat kargatzeko hurrengo direktibaren eredu jarraituko da: `#include <liburutegiizena.h>`

Gure lanaren programazioari dagokionez liburutegi bi erabiliko dira: LCD moduluen eta RF moduluen kudeaketarako, *LiquidCrystal* eta *VirtualWire* hurrenez hurren.

## Sarrera/Irteerak\*

Atal honetan proiektuaren programazioa eta aldagaiak definitzeko, kontutan izango diren sarrera/ irteerak deskribatuko dira.

### Sarrerak

#### ANALOGIKOAK

- **A0 ataka:** *unsigned int* sample; (mikrofonoaren seinalea jaso, 1. arduino plakan)
- **A0 ataka:** *int* alegoera; (alarma egoera ahalbidetzeko pultsagilua, 2. arduino plakan)

#### DIGITALAK

- **D11 ataka:** ( RF modulu hartzailetik datua jaso, 2.arduino plaka)

### Irteerak

#### DIGITALAK

- **D12 ataka:** (modulu igorleari datua bidali, 1go arduino plaka)
- **D8 ataka:** *int* led; (mezua bidalita. Frogatzeko led-a piztu, 1go arduino plaka)
- **D2, D3, D4, D5, D6, D7 atakak:** (LCD-a kudeatzeko, 2.arduino plaka)\*\*
- **D13 ataka:** *int* led 13 (alarma dago. motorra bibratuarazi, 2.arduino plaka)

### Oharrak

\* Sarrera/ irteerako aldagaiak gure kodean erabilitakoak dira; noski, beste batzuk aukera zitezkeen.

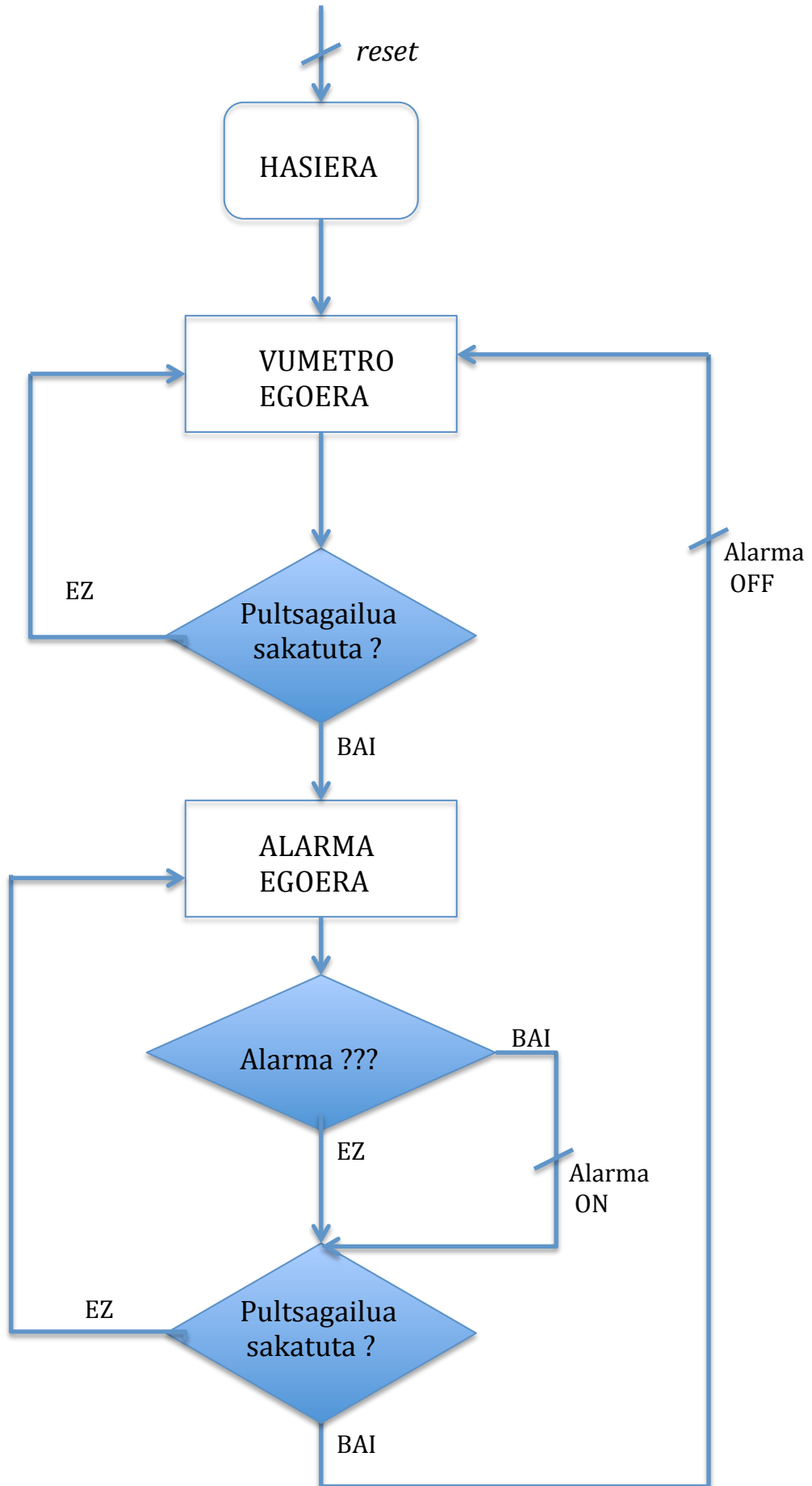
\*\* LCD shield-aren ezaugarri orrietan zehaztuta

## **Funtzionamendua**

Bukaerako sistemaren funtzionamendua bi egoeratan banandu daiteke. Lehenengoa, eta itzulezina (hau da, behin sistema hasieratu edo berrabiarazte bat "*Reset*" gertatzen denean, begizta itxi batean uneoro exekutatzen egongo dena), VUMETRO EGOERA da.

Bigarren egoera, ALARMA EGOERA litzateke. Egoera hau itzulgarria da. Pulsagailu batekin egoera hau gaitu eta ezgaitu daiteke. Bakarrik bigarren egoera honetan gaudelarik (eta baldin eta soilik baldin egoera honetan gaudenean) "alarma gertaera" batek motorra bibratuaraziko du. Motorra gelditzeko ALARMA EGOERA ezgaitu egingo da.

Aurreko azalpena modu intuitiboagoan ulertzeko, bisualagoa alegia, funtzionamendudigrama batez lagunduko gara. (Hurrengo orrialdean ikusgai)



## Kodeak

```

// IGORLEA
#include <VirtualWire.h> // RF (VirtualWire) liburutegia hasieratu

const int sampleWindow = 10; // Laginketa leihoaren zabalera (ms-tan)

// *****ALDAGAIEN DEKLARAZIOA*****K //
unsigned int sample;
int led=8; // LEDa 13 pinera konektatuko da
char* msg;
char mezuzabal[2];
unsigned int mean[5];
int i=0;
int k=0;

void setup() // Set up konfigurazioa (ireerak, sarrerak, serial komunikaz
{
Serial.begin(9600);
pinMode(led, OUTPUT);

vw_set_ptt_inverted(true); // RF modulua abiarazteko egiaztapena
vw_setup(2000); // Konexio abiadura bps-tan
vw_set_tx_pin(12); // Modulu igorleari datuak transmititzeko 12 pinera konekt
Serial.println("Igorlea hasieratuta");
}

void loop() //***** BEGIZTA NAGUSIA *****//
{
unsigned long startMillis=millis(); // Laginketa leihoa hasieratzeko

unsigned int peakToPeak = 0; // Peak to Peak maila
unsigned int peakToPeakmean = 0;
unsigned int result = 0;
unsigned int batazb=0;

unsigned int signalMax = 0;
unsigned int signalMin = 1024;

while (millis() - startMillis < sampleWindow) // Datuen bilketa hasi
{
sample =analogRead(0); // Sesntsorearen irteera, Arduinoaren A0 sarrera analog
if (sample < 1024)
{
if (sample > signalMax)
{
signalMax = sample; // Laginketa tartean, seinale balio maximoa
}
else if (sample < signalMin)
{
signalMin = sample; // Laginketa tartean, seinale balio minimoa
}
}
}
peakToPeak = signalMax - signalMin; // Max - min = Peak to Peak anplitudea
peakToPeak=map(peakToPeak, 0, 900, 1, 8); // Peak to Peak aldagaiaren mapeoa

```

```
    mean[i]=peakToPeak;//          Array-a      bete
    i=i+1;

    if (i>4){// Array-a beteta dagoenean, mezua bidali
        for (k = 0; k< 5; k++){// Array-tik irakurketa egin
            peakToPeakmean=mean[k];
            result= peakToPeakmean+result;}
            batazb=result/5;// batazbestekoa
            Serial.println (batazb);
            mezuzabal[0] =(char) batazb;

            // ***** Mezua bidaltzeko: *****//
            char *msg = mezuzabal;
            vw_send((uint8_t *)msg, strlen(msg));
            vw_wait_tx(); // Mezua bidali arte itxaron
            Serial.println(msg[0]); // Mezua Serial monitorean bistaratu
            delay(100); // Hurrengo mezua bidaltzeko itxaron
            i=0;
        }
    } // ***** BEGIZTA NAGUSIA BUKATU *****//
```



```

// HARTZAILEA
// RF modulu eta LCDaren kudeaketarako liburutegien hasieratzea//
#include <VirtualWire.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);// LCD-a hasieratu

const char vum1[2]= {0xFF};
const char vum2[3]= {0xFF, 0xFF};
const char vum3[4]= {0xFF, 0xFF, 0xFF};
const char vum4[5]= {0xFF, 0xFF, 0xFF, 0xFF};
const char vum5[6]= {0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
const char vum6[7]= {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
const char vum7[8]= {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
const char vum8[9]= {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};

//***** Aldagaien deklarazioak *****//
char zbk[2];
char maila;
int led=13;
byte hasieratu=0;
int k,j =0;
int batura=0;
int alarma[5]={0,0,0,0,0};
int alarmamaila=0;
int batazb=0;
int alegoera=0;
int ekont=0;
int aurrepultsa=0;
int pultsa=0;

void setup() // Set up konfigurazioa (ireerak, sarrerak, serial komunikazioa)
{
Serial.begin(9600); // Ordenagailuarekin serie konexioa konfiguratu
vw_set_ptt_inverted(true); // RF modulu abiarazteko egiaztapenaF
vw_setup(2000); // Konexio abiadura bps-tan
vw_set_rx_pin(11); //Modulu hartzailearen 'data' ataka 11 pinera konektatu
vw_rx_start(); // Hartzailea hasieratu
pinMode(led, OUTPUT);

lcd.begin(16,          2);// Set up, LCDaren errenkada eta zutabe kop.

}

void loop() // *****BEGIZTA NAGUSIA*****//
{
uint8_t buf[VW_MAX_MESSAGE_LEN]; // defektuz 30 short-int matrizea
uint8_t buflen =VW_MAX_MESSAGE_LEN;
if (vw_get_message(buf, &buflen)// Dataren jasoera konprobatzeko
{
int i,g;
// Mezuaren egiaztapen egokia

```

```
for (i = 0; i < buflen; i++)
{
Serial.write(buf[i]); // Jasotako datuak buffer-ean gorde

zbk[i]=buf[i];
maila= zbk[i];
lcd.setCursor(0, 1); //LCDan kurtsorea posizionatu

switch (maila){ // ***** SWITCH CASE *****//
  case 1:{
    lcd.print(vum1);
    lcd.print(" ");
  }
  break;
  case 2:{
    lcd.print(vum2);
    lcd.print(" ");
  }
  break;
  case 3:{
    lcd.print(vum3);
    lcd.print(" ");
  }
  break;
  case 4:{
    lcd.print(vum4);
    lcd.print(" ");
  }
  break;
  case 5:{
    lcd.print(vum5);
    lcd.print(" ");
  }
  break;
  case 6:{
    lcd.print(vum6);
    lcd.print(" ");
  }
  break;
  case 7:{
    lcd.print(vum7);
    lcd.print(" ");
  }
  break;
  case 8:{
    lcd.print(vum8);
    lcd.print(" ");
  }
  break;
} // ***** end SWITCH CASE *****

} //for amaitu

k=k-j;
alarma[k]=maila;
```

```
k=k+1;
j=0;

    if (hasieratu!=0){// ***** EGOERA EGONKORRA *****
        batura=0;
        for (g = 0; g < 5; g++){
            alarmamaila =alarma[g];
            batura= alarmamaila+batura;}
        batazb= batura/5;
        Serial.println(batazb);}

alegoera= analogRead(0);
if (alegoera<300){
    pultsa=1;}
else{ pultsa=0;}

if (pultsa != aurrepulsas){// alarma piztu-itzali errutina
    if (pulsas==1){
        ekont=ekont+1;}
}
aurrepulsas=pulsas;

    if (ekont%2==0 ){
        digitalWrite(led, LOW);
        lcd.setCursor(0, 0);
        lcd.write ("vumetro ");
    }
    else{
        if (batazb>5){
            digitalWrite(led, HIGH);}
        lcd.setCursor(0, 0);
        lcd.write ("alarma egoera");
    }

Serial.println();

    if (k>4)  { // abio baldintza
        hasieratu= 1;
        j=5;
    }

    // if begizta nagusiaren amaiera
} // ***** loop-aren amaiera *****
```

## Erabiltzailearentzako gida

Kode guztiaren programazioa, erabiltzaileak nahi beste doiketak egiteko aukera izan dezan, malgutasuna eskeiniz pentsatu da. Horren adibide, ondoren malgutasun hori eskeintzen duten aldagai eta parametro aipagarrienak, batzuk behintzat eta betiere proiektuaren funtzionalitateari lotuta, aurkeztuko dira:

### ALDAGAIAK\*

*sample*: laginketa periodoa /leihoaren tamaina moldatzeko. Honekin kalkulurako erabilitako guztizko laginen batzbesteko denbora handitu edo txikitu daiteke. Honela datuen transmisio erritmoa moldatuko da ere.

*batazb*: alarma-pizte atari balioa aukeratzeko.

\*Erabiltzaileak hauek eraldatu ditzake kasu partikularren beharretara egokitzeko

### FUNTZIOAK

*map*:  $\text{map}(X, Y, n_0, n)$ . Gure *vumetro*-rako aukeratutako (1-etik, 8-rako)  $n$  mailak,  $X$  eta  $Y$  balioen bitarteko balioekin tipifikatuko ("mapatuko") dira,  $X$  eta  $Y$  parametroen arabera.

Parametro hauekin jokatuz *vumetro*ko maila bakoitzaren sentikortasuna aldatu daiteke.

Hobeto ulertzeko:

Hurrengo formulak,  $|X - Y|$  tartearen zenbateko balio aldaketa beharrezkoa den hurrengo  $n$  mailara pasatzeko erakusten du.

$$\text{Sentikortasun (Se)} = \frac{|X-Y|}{(n-n_0)+1} \quad \text{Non, } X < Y \text{ eta } n_0 < n \text{ beti.}$$

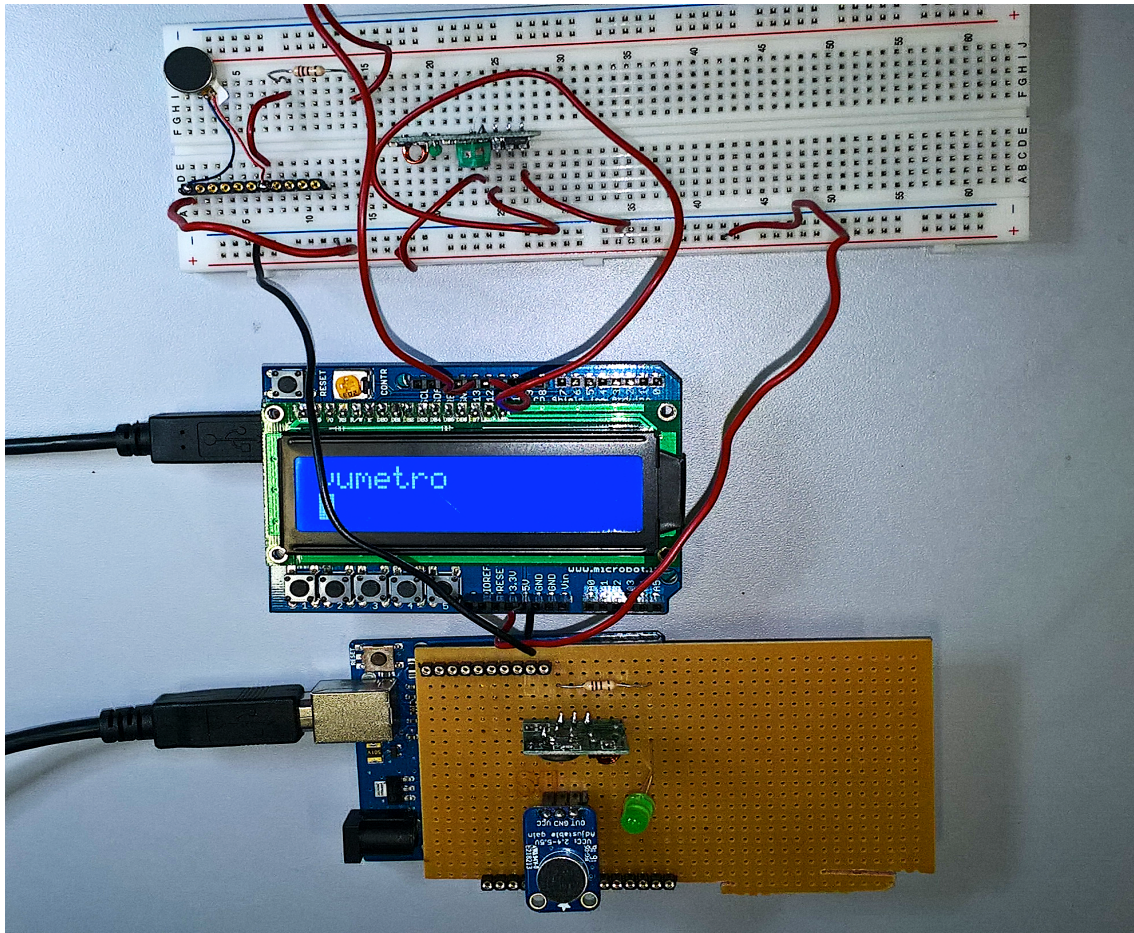
**adib 1.:**  $\text{map}(0, 800, 1, 8)$ , goiko formula aplikatuz erraz ikusten da  $n$  maila bakoitzari 100 baliotako tartea dagokiola.  $\text{Se}_1 = 100$

**adib 2.:**  $\text{map}(0, 400, 1, 8)$ , orain  $n$  maila bakoitzari 50 baliotako tartea dagokio.

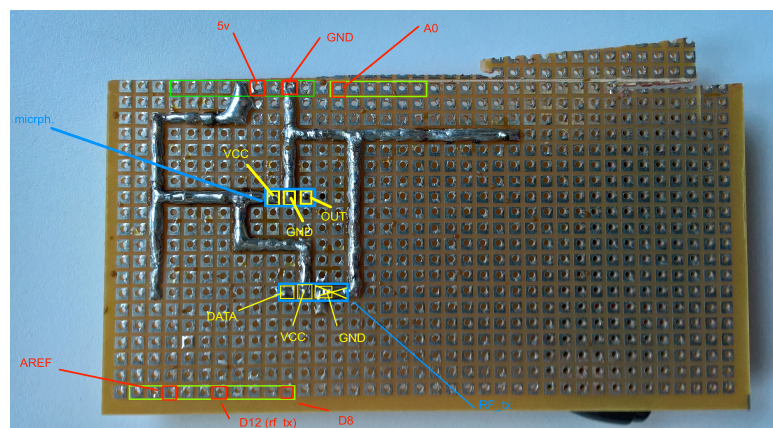
$$\text{Se}_2 = 50$$

Hortaz,  $\text{Se}_1 < \text{Se}_2$  (sentikortasun hobea 2. adibidean)

## 4. BUKAERAKO DISEINUA



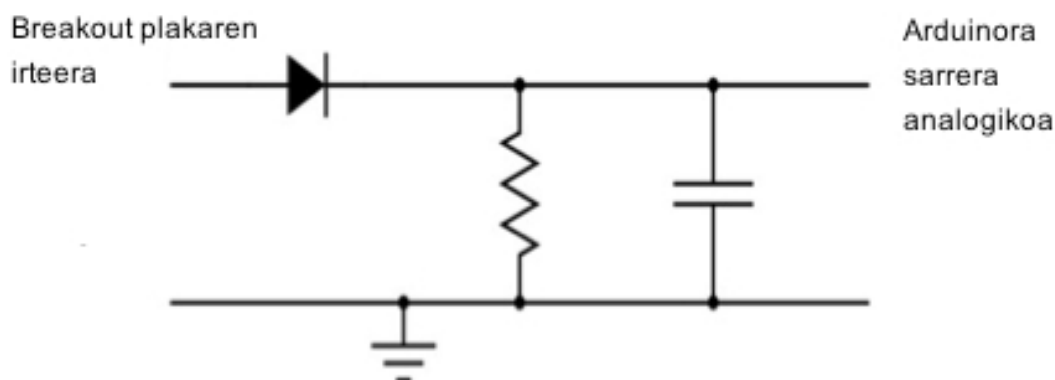
4.1. Irudia. *Bukaerako muntaia*



4.2. Irudia. *Bakelitazko plakaren soldaketa*

## 5. HOBEKUNTZA PROPOSAMENAK

- Mikrofono modulu anplifikatzailearen irteerako seinalea tratamendu analogikoa aplikatu dakiok, esaterako, zarata puntualak jarraitasuna duten soinetatik bereizteko. Aurre tratamendu honek gerorako kodearen sinplifikazioa suposa lezake gainera.



5.1. Irudia. *Envelope detector circuit*

Goiko zirkuituak seinalearen balio positiboak iragazteaz gain *Peak to Peak* balioen arteko seinale "leundua" eraikitzen du (kondentsadorearen karga eta deskarga denborak  $R$ ,  $C$  parametroen menpe egongo dira).

Jadanik ezaguna dugu Atmega328P-ak eskaini dezakeen laginketa abiadurak, bere mugak dituela. Hori dela eta maiztasun altuko tontorrak galdu egiten dira.

Aztertzeke uzten da, proposamen honen irteera ahotsaren ezaugarri bereziak dituen seinale bati aplikatzean.

- RF moduluen irismena hobetzeko antenen erabilera estudiatu daiteke, hauek gaineratu behar balira.
- DC motorra 2. arduino plakatik bereiz zitekeen, sistemaren portaera errealistago bat simulatzeko. Honetarako proposatzen diren aukeren artean, lehenengoa 3.

Arduino shield baten eskutik etorriko litzake. Plaka berriak RF modulu hartzaile berriaren kudeaketaz eta motorren kitzikapenaz, arduratzeaz baino besterik egin behar ez lukeenez, Arduino Lilypad, Arduino Nano edota Arduino Mini bezalako dispositiboek baloratu zitezkeen proposamena burutzeko.

Nolanahi ere, RF modulu pare berri baten eskuraketa nahitaezkoa da, komunikazioan interferentziak ekiditeko, beste maiztasun batean lan egingo duena (315 MHz-eko bandan adibidez). Beste aukera bat komunikazio protokolo eta modulo ezberdinak frogatzean oinarritu daiteke (Wifi, Bluetooth...).

Betelaz ere, bidalitako informazioa bitarra den eanean (alarma BAI, alarma EZ) RF modulu hartzaile berriaren irteera erabil zitezkeen soilik, atonduta, bibrazio motorra kitzikatzeko; inongo kontrolagailu edo Arduino plakaren beharrik gabe. Funtzionaltasun bakar hori (bibrazioa) mugatuta, betiere.