

MENJADORA AUTOMÀTICA PER A MASCOTES



CURS DE PROGRAMACIÓ DE PLAQUES ROBÒTIQUES.

CIFO - LA VIOLETA.

MARINA ÁLVAREZ ÁLVAREZ

ÍNDEX

INTRODUCCIÓ	3
--------------------------	----------

COS

Disseny.....	4
Funcionament.....	5
Materials utilitzats.....	6
Codi.....	9
Esquema	14

CONCLUSIONS

Objectius assolits.....	15
Possibles millores	16

INTRODUCCIÓ

L'objectiu del projecte és construir una menjadora automàtica per a mascotes controlada per una placa Arduino UNO.

Des del programa creat amb l'Arduino IDE es podrà escollir les hores a les que volem que l'aparell dispensi el pinso i la quantitat que ha de caure, podent adaptant-nos així a l'horari d'àpats de cada animal i els seus requeriments nutricionals.

Després d'administrar el pinso sonarà una melodia per avisar a la mascota de que el pinso està servit.

DISSENY:

El disseny és simple, consta bàsicament d'una caixa on col·locar i protegir la placa Arduino i la resta de components, un recipient amb tapa per emmagatzemar el pinso i una pestanya, que un motor servo accionarà per permetre el pas del pinso i un bol.



FUNCIONAMENT:

El mòdul RTC «Real Time Clock» és un dispositiu que sincronitza l'hora actual.

El motor servo i el buzzer s'activaran quan el dispositiu RTC arribi a la hora programada, deixant pas al pinso i reproduint una melodia per avisar a la mascota.

La pantalla LCD mostrarà l'hora actual i l'hora de l'àpat següent, així com un missatge mentre el pinso s'estigui servint.

En el següent enllaç es pot veure el funcionament del circuit:

<https://www.screencast.com/t/Vvj3WAoCU>

MATERIALS UTILITZATS:

Software:

- Arduino IDE


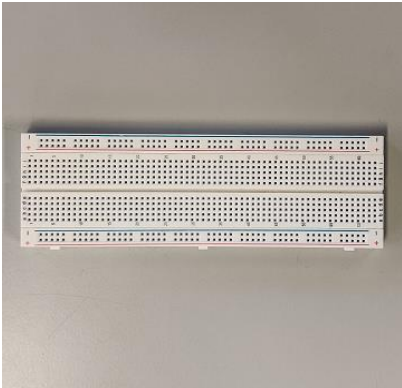
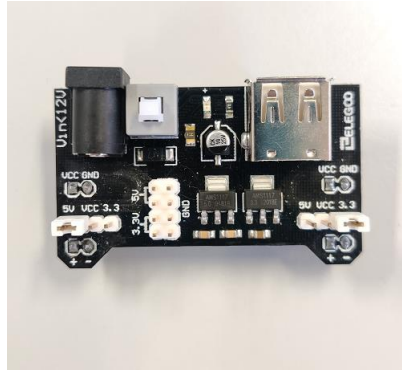
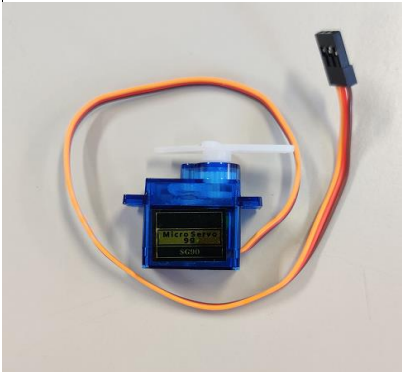

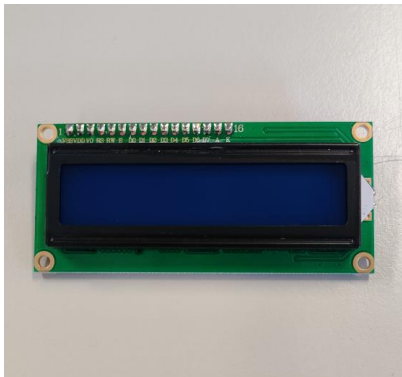



Llibraries Arduino:

- Wire.h
- RTCLib.h
- DS1307RTC.h
- Servo.h
- LiquidCrystal.h

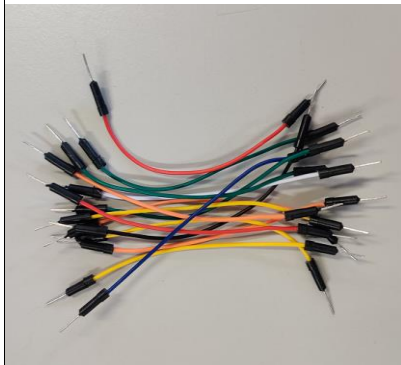
Materials no electrònics:

- Recipient de plàstic apte per emmagatzemar aliments
- Tub de plàstic
- Pestanya de plàstic
- Estructura de cartró
- Bol

Components electrònics:

<p>Placa Arduino UNO</p> 	<p>Protoboard</p> 	<p>Power Supply</p> 
<p>Motor MicroServo SG90</p> 	<p>Mòdul RTC DS1307</p> 	<p>Mòdul LCD1602</p> 
<p>Potenciòmetre 10K</p> 	<p>Buzzer passiu</p> 	<p>Cables pont mascle-femella (12 unitats)</p> 

Cables pont mascle-mascle (12 unitats)



Adaptador 9V1A



Cable USB



CODI:

```
#include <Wire.h>

#include "RTCLib.h"

#include <Servo.h>

#include <LiquidCrystal.h>

#include "DS1307RTC.h"


// declara notes del Buzzer

#define NOTE_DS3 156

#define NOTE_E3 165

#define NOTE_FS3 185

#define NOTE_G3 196

#define NOTE_AS3 233

#define NOTE_D4 294

#define NOTE_DS4 311

double porcentaje = 1.1;


const int pinBuzzer = 2; // PIN buzzer


const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12; //PINS LCD

LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // Declara lcd


Servo myservo; // Declara motor servo

bool state = true;
```

```

int pos = 0;

RTC_DS1307 rtc; //declara real time clock

void setup() {
    Serial.begin(9600);
    myservo.attach(6); //Servo
    lcd.begin(16, 2); //Pantalla LCD

    setSyncProvider(RTC.get); //inicialitza RTC i sincronitza hora actual
    if (timeStatus() != timeSet)
        Serial.println("Fallo de RTC");
    else
        Serial.println("Sincronizado con RTC");
    pinMode(pinBuzzer, OUTPUT); //Buzzer
}

//Imprimeix per pantalla un missatge i activa la funció del Servo.
void Menjar() {
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("A MENJAR!!");
    Serial.print ("a menjar");
    delay(1000);
    ServoOn();
}

//Acciona el Servo perquè caigui el pinso
void ServoOn() {

```

```

for (pos = 0; pos <= 75; pos += 1) {
    myservo.write(pos);
    delay(20);
}
for (pos = 75; pos >= 0; pos -= 1) {
    myservo.write(pos);
    delay(20);
}
delay(1000);
}

void So(){ //Programa una melodia
int melodia[] = {
NOTE_G3, NOTE_G3, NOTE_G3, NOTE_DS3, NOTE_AS3, NOTE_G3, NOTE_DS3, NOTE_AS3,
NOTE_G3, NOTE_D4, NOTE_D4, NOTE_D4, NOTE_DS4, NOTE_AS3, NOTE_FS3, NOTE_DS3,
NOTE_AS3, NOTE_G3, 0};

int tiempo[] = { // temps que dura cada nota (1=1segon)
2, 2, 2, 3, 9, 2, 3, 9, 2, 2, 2, 2, 3, 9, 2, 3, 9, 1 };

int arregloSize = sizeof(melodia) / sizeof(int);

for (int i = 0; i < arregloSize; i++) { // bucle fins la i-èssima nota
    int duracion = 1000 / tiempo[i];

    tone(pinBuzzer, melodia[i], duracion); //Reproduceix cada nota del bucle

    int pausa = duracion * porcentaje; // Pausa 1.1% de la duració de la nota
    delay(pausa);

    noTone(pinBuzzer);
}

```

```

    }
}

void loop() {

    String h1 = "07:00:00"; String h2= "11:00:00"; String h3= "19:00:00"; String h4= "23:00:00" ;
    //Declara hores de menjar tipus String

    char charh1[3]; char charh2[3]; char charh3[3]; char charh4[3];

    h1.toCharArray(charh1, 3); h2.toCharArray(charh2, 3); h3.toCharArray(charh3, 3);
    h4.toCharArray(charh4,3); //Passa els dos primers digits dels String anteriors a array de char

    int inth1 = atoi(charh1); int inth2 = atoi(charh2); int inth3 = atoi(charh3); int inth4 =
    atoi(charh4); // hores de char a int


    DateTime now = rtc.now(); //hora actual

    char buf1[] = "hh:mm:ss"; //format en que volem que imprimeixi l'hora actual

    lcd.setCursor(0,0);

    lcd.print("HORA: ");

    lcd.setCursor(6, 0);

    lcd.print(now.toString(buf1)); //imprimeix hora a la LCD

    //Serial.println(buf1);

    String hora=now.toString(buf1);

    if ((hora == h1) || (hora == h2) || (hora == h3) || (hora == h4)){ //activa la funció Menjar() i
    So() a les hores escollides.

        Serial.println("MEAL TIME!!");

        Menjar();

        delay(500);

        So();    }

    lcd.setCursor(0, 1);

    lcd.print("NEXT MEAL: ");

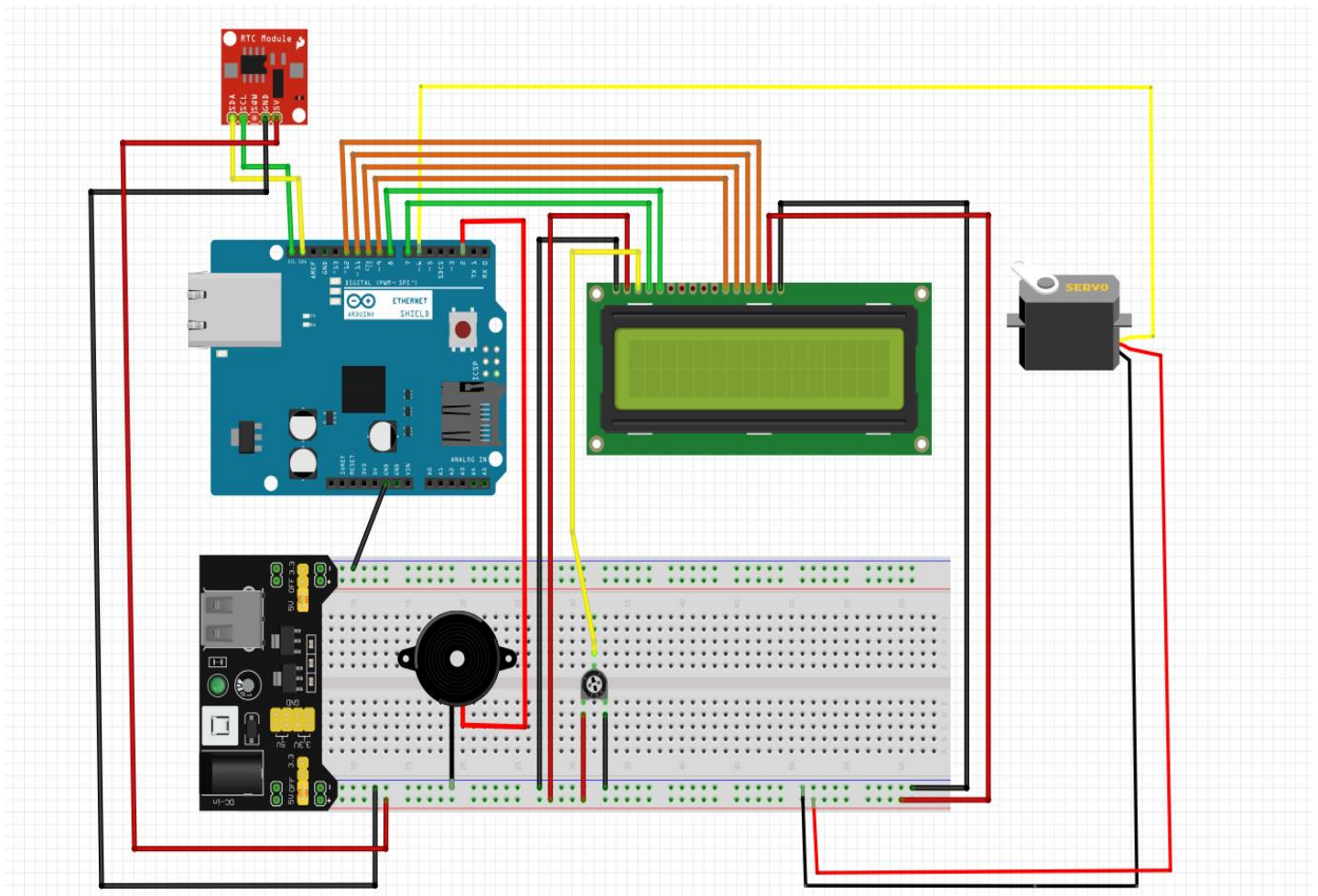
```

lcd.setCursor(11,1); // Quan l'hora actual coincideix amb l'hora de menjar imprimeix a la LCD l'hora del següent àpat.

```
if (hour() < inth1) {  
    lcd.print(h1); }  
else if (hour() < inth2) {  
    lcd.print(h2); }  
else if (hour() < inth3) {  
    lcd.print(h3); }  
else if (hour() < inth4) {  
    lcd.print(h4); }  
  
delay(1000);  
}
```

ESQUEMA:

Realitzat amb el software Fritzing.



CONCLUSIONS:

OBJECTIUS ASSOLITS:

Els objectius principals han sigut assolits. Disposem d'un artefacte que deixa pas al pinso quatre cops al dia.

La quantitat de pinso que surt l'he ajustat variant els graus de moviment del servo i la velocitat del motor, ajustant-los i comprovant, amb una bàscula de cuina, els grams de pinso que queien. El pes que cau no és del tot exacte i depèn en certa part de com de ple estigui el dipòsit del pinso. Actualment cauen de 30 a 40 grams cada cop que s'activa el servo.

Una de les dificultats ha sigut mostrar per pantalla LCD l'hora del següent àpat, ja que el mòdul RTC proporciona l'hora de tipus String. Passant l'String a char array i després a int he aconseguit que detecti quina hora és més gran sense tenir en compte els minuts. Per tant, perquè el programa funcioni correctament les hores dels àpats s'han de posar en punt.

En el següent enllaç es pot visualitzar un vídeo que mostra el funcionament de la menjadora.

<https://www.screencast.com/t/MhQsGjiqC6Hm>

La menjadora posada a la pràctica:

<https://www.screencast.com/t/BbAhHOiy9N>

<https://www.screencast.com/t/NDl6XlqiYi>

POSSIBLES MILLORES:

- Substituir la pantalla LCD1602 per una LCD amb botons integrats que permeti seleccionar opcions i poder modificar l'hora dels àpats sense necessitat d'entrar al codi des de l'ordinador, modificar les hores i pujar-lo a la placa Arduino.
- Programar una funció que permeti escollir els grams de pinso que volem que caiguin. Amb el programa actual si volem modificar la quantitat de pinso hem d'entrar al fitxer de l'Arduino IDE i jugar amb els graus (pos) i la velocitat del servo fins que trobem els paràmetres que ens serveixen.
- Afegir una webcam per comprovar que el pinso cau correctament i que la mascota està menjant.
- Construir una estructura amb materials més resistents i impermeables.
- Construir un sistema on la pestanya que permet que surti el menjar encaixi millor amb el recipient d'emmagatzematge fent-lo més estanc i hermètic per tal de que el pinso es conservi millor.

