Хранилка за животни

Изготвено от:

Айхан Хасанов, Симона Белчева, Юлиана Иванова

Съдържание

3 стр.

4 стр.

5 стр.

14 стр.

1. Въведение / Ел. схема
2. Блок схема / Съставни части
3. Сорс код
4. Автори и контакти

Въведение:

Pet Feeder e ардуино проект с цел автоматичното хранене на домашен любимец. Пуска храна чрез въртенето на сервомотора, който е контролиран от потребителя като се посочат час и минути след които да се задейства. Проектът има налична симулация.

**Ел. Схема:**

**Diagram

Description automatically generated**

**Блок схема:**

Graphical user interface

Description automatically generated with medium confidence

**Съставни части:**

1. Arduino Uno R3

2. 330Ω Resistor

3. 10 kΩ Resistor

4. Pushbutton

5. Positional Micro Servo

6. LCD 16x2

7. Breadboard

**Сорс Код\*:**

*\*ТОВА Е ЦЕЛИЯТ СОРС КОД (наличен и в github)*

#include <LiquidCrystal.h>

#include <Servo.h>

Servo servomotor;

LiquidCrystal lcd(8,9,4,5,6,7);

int mode = 0;

const int servoPin = 12;

int h = 0; // hours

int m = 0; // minutes

void setup()

{

Serial.begin(9600);

lcd.begin(16,2);

lcd.print("Welcome! ");

lcd.setCursor(0,1);

lcd.print("Pet Feeder V1 ");

pinMode(10,INPUT);

pinMode(11,INPUT);

pinMode(2, INPUT);

pinMode(A0, INPUT);

servomotor.attach(servoPin);

servomotor.write(0);

delay(500);

}

/\*

LEFT BUTTON - navigates through different modes

MIDDLE BUTTON - used for adding 1 hour/minute

RIGHT BUTTON - used for removing 1 hour/minute

\*/

void loop()

{

ChangeMode();

if(mode == 0)

{

lcd.setCursor(0,0);

lcd.print("Welcome!");

lcd.setCursor(0,1);

lcd.print("Pet Feeder V1");

}

if(mode == 1)

{

SetHours();

}

if(mode == 2)

{

SetMinutes();

}

if(mode == 3)

{

Timer();

}

if(mode == 4)

{

CancelTimer();

}

}

/\*

When the most left button ("mode change button") is pressed the mode changes /\*

mode 0 - home screen - "Pet Feeder V1"

mode 1 - set hours

mode 2 - set minutes

mode 3 - start timer

mode 4 - cancel current timer

\*/

void ChangeMode()

{

if(digitalRead(2) == HIGH)

{

mode++;

lcd.clear();

Serial.println("Mode changed to ");

Serial.println(mode);

if(mode >= 5)

{

mode = 0;

h = 0;

m = 0;

}

delay(1000);

}

}

/\*

Sets the hour for the timer

\*/

void SetHours()

{

if(digitalRead(10) == HIGH)

{

if(h <= 24)

{

h++;

if(h == 24)

{

h = 0;

}

Serial.print("Hour Added!\n");

Serial.print(h);

Serial.print("\n");

}

}

if(digitalRead(11) == HIGH)

{

if(h > 0)

{

h--;

Serial.print("Hour removed!\n");

Serial.print(h);

Serial.print("\n");

}

}

Print(1);

delay(150);

}

/\*

Set the minutes for the timer

\*/

void SetMinutes()

{

if(digitalRead(10) == HIGH)

{

if(m <= 59)

{

m++;

if(m==59)

{

m = 0;

}

Serial.println("Minute added!");

Serial.println(m);

}

}

if(digitalRead(11) == HIGH)

{

if(m > 0)

{

m--;

Serial.println("Minute removed!");

Serial.println(m);

}

}

Print(2);

delay(100);

}

/\*

A prototype of a real timer

Uses "delay();" to act as a real timer

Checks

\*/

void Timer()

{

if(m > 0)

{

m--;

Serial.println("A minute has passed! Remaining minutes: ");

Serial.println(m);

}

if(m == 0 && h != 0)

{

h--;

m = 59;

Serial.println("An hour has passed! Remaining hours: ");

Serial.println(h);

}

Print(0);

/\*

1 minute = 60 000 ms = 600 \* 100 -> (iterations \* delay ms)

Using a loop of 600 iterations each of which is delayed by 100 ms

allows the user to cancel the timer in intervals of 100 ms.

If 'delay(60000);' was used the user wouldn't be able to abort the timer

before a minute passes

\*/

for(int i = 0; i < 600; i ++)

{

delay(100);

ChangeMode();

if(mode == 4)

{

CancelTimer();

Serial.println("The timer has been cancelled!");

return;

}

}

}

/\*

Prints message and time in accordance to the given 'print type' code

CODES:

0 - none, print only timer

1 - set hours

2 - set minutes

\*/

void Print(int type)

{

if(type == 1)

{

lcd.setCursor(0,0);

lcd.print("Setting hours:");

PrintTime();

}

if(type == 2)

{

lcd.setCursor(0,0);

lcd.print("Setting minutes:");

PrintTime();

}

if(type == 0)

{

if(h == 0 && m == 0)

{

lcd.clear();

Feed();

lcd.print("Pet is fed");

h = -1;

m = -1;

mode++;

}

else

{

lcd.setCursor(0,0);

lcd.print("Feeding pet in:");

PrintTime();

delay(1000);

}

}

}

/\*

Prints only the time on the second line of the lcd matrix

\*/

void PrintTime()

{

lcd.setCursor(0,1);

if(h<10)

{

lcd.print("0");

lcd.print(h);

}

else

{

lcd.print(h);

}

lcd.print(":");

if(m < 10)

{

lcd.print("0");

lcd.print(m);

}

else

{

lcd.print(m);

}

}

/\*

Rotates servomotors wing allowing the food to drop

\*/

void Feed()

{

servomotor.write(90);

delay(1000);

servomotor.write(0);

Serial.println("Pet has just been fed!");

}

/\*

Cancels the timer

Set hours and minutes to 0

\*/

void CancelTimer()

{

h = 0;

m = 0;

lcd.setCursor(0,0);

lcd.print("Timer is ");

lcd.setCursor(0,1);

lcd.print("Cancelled ");

}#include <Servo.h>

Servo servomotor;

LiquidCrystal lcd(8,9,4,5,6,7);

int mode = 0;

const int servoPin = 12;

int h = 0; // hours

int m = 0; // minutes

void setup()

{

Serial.begin(9600);

lcd.begin(16,2);

lcd.print("Welcome! ");

lcd.setCursor(0,1);

lcd.print("Pet Feeder V1 ");

pinMode(10,INPUT);

pinMode(11,INPUT);

pinMode(2, INPUT);

pinMode(A0, INPUT);

//attachInterrupt(digitalPinToInterrupt(2), changemode, RISING);

servomotor.attach(servoPin);

servomotor.write(0);

delay(500);

}

/\*

LEFT BUTTON - navigates through different modes

MIDDLE BUTTON - used for adding 1 hour/minute

RIGHT BUTTON - used for removing 1 hour/minute

\*/

void loop()

{

ChangeMode();

if(mode == 0)

{

lcd.setCursor(0,0);

lcd.print("Welcome!");

lcd.setCursor(0,1);

lcd.print("Pet Feeder V1");

}

if(mode == 1)

{

SetHours();

}

if(mode == 2)

{

SetMinutes();

}

if(mode == 3)

{

Timer();

}

if(mode == 4)

{

CancelTimer();

}

}

/\*

When the most left button ("mode change button") is pressed the mode changes /\*

mode 0 - home screen - "Pet Feeder V1"

mode 1 - set hours

mode 2 - set minutes

mode 3 - start timer

mode 4 - cancel current timer

\*/

void ChangeMode()

{

if(digitalRead(2) == HIGH)

{

mode++;

lcd.clear();

Serial.println("Mode changed to ");

Serial.println(mode);

if(mode >= 5)

{

mode = 0;

h = 0;

m = 0;

}

delay(1000);

}

}

void SetHours()

{

if(digitalRead(10) == HIGH)

{

if(h <= 24)

{

h++;

if(h == 24)

{

h = 0;

}

Serial.print("Hour Added!\n");

Serial.print(h);

Serial.print("\n");

}

}

if(digitalRead(11) == HIGH)

{

if(h > 0)

{

h--;

Serial.print("Hour removed!\n");

Serial.print(h);

Serial.print("\n");

}

}

Print(1);

delay(150);

}

void SetMinutes()

{

if(digitalRead(10) == HIGH)

{

if(m <= 59)

{

m++;

if(m==59)

{

m = 0;

}

Serial.println("Minute added!");

Serial.println(m);

}

}

if(digitalRead(11) == HIGH)

{

if(m > 0)

{

m--;

Serial.println("Minute removed!");

Serial.println(m);

}

}

Print(2);

delay(100);

}

void Timer()

{

if(m > 0)

{

m--;

Serial.println("A minute has passed! Remaining minutes: ");

Serial.println(m);

}

if(m == 0 && h != 0)

{

h--;

m = 59;

Serial.println("An hour has passed! Remaining hours: ");

Serial.println(h);

}

Print(0);

for(int i = 0; i < 600; i ++)

{

delay(100);

ChangeMode();

if(mode == 4)

{

CancelTimer();

Serial.println("The timer has been cancelled!");

return;

}

}

}

/\*

0 - none, print only timer

1 - set hours

2 - set minutes

\*/

void Print(int type)

{

if(type == 1)

{

lcd.setCursor(0,0);

lcd.print("Setting hour:");

PrintTime();

}

if(type == 2)

{

lcd.setCursor(0,0);

lcd.print("Setting minutes:");

PrintTime();

}

if(type == 0)

{

if(h == 0 && m == 0)

{

lcd.clear();

Feed();

lcd.print("Pet is fed");

h = -1;

m = -1;

mode++;

}

else

{

lcd.setCursor(0,0);

lcd.print("Feeding pet in:");

PrintTime();

delay(1000);

}

}

}

void PrintTime()

{

lcd.setCursor(0,1);

if(h<10)

{

lcd.print("0");

lcd.print(h);

}

else

{

lcd.print(h);

}

lcd.print(":");

if(m < 10)

{

lcd.print("0");

lcd.print(m);

}

else

{

lcd.print(m);

}

}

void Feed()

{

servomotor.write(90);

delay(1000);

servomotor.write(0);

Serial.println("Pet has just been fed!");

}

void CancelTimer()

{

h = 0;

m = 0;

lcd.setCursor(0,0);

lcd.print("Timer is ");

lcd.setCursor(0,1);

lcd.print("Cancelled ");

}

**Автори и контакти**

**Айхан Хасанов**

EMAIL: [**ayhan.hsn.01@gmail.com**](mailto:ayhan.hsn.01@gmail.com)

Github: **https://github.com/AyhanHasanov**

**Симона Белчева**

EMAIL: [**monniebellchewa@gmail.com**](mailto:monniebellchewa@gmail.com)

Github: **https://github.com/monnikyu**

**Юлиана Иванова**

EMAIL: [**julianaivanova04@gmail.com**](mailto:julianaivanova04@gmail.com)

Github: **https://github.com/saltyyul**