

Feline Feeder

Introducere

Proiectul Feline Feeder este un dispozitiv automatizat care asigură hrănirea regulată a pisicilor tale. Acesta stochează o cantitate generoasă de hrană într-un recipient și eliberează porțiuni controlate în mod automat, la orele programate. Scopul principal al lui este acela de a facilita și de a face mai convenabilă hrănirea pisicilor, în special pentru proprietarii ocupați sau cei care călătoresc des.

Ideea a pornit de la pisicuța mea Luna, care, de multe ori, a rămas fără mâncare atunci când eu eram plecată la facultate sau în alte locuri pentru o perioadă mai lungă de timp.

Feline Feeder este util atât pentru noi, cât și pentru animăluțele noastre de companie. Pentru proprietari, acest dispozitiv oferă liniștea că pisicile lor sunt hrănite regulat chiar și atunci când sunt ocupați sau plecați de acasă. De asemenea, ne oferă satisfacția că putem contribui la sănătatea și bunăstarea lor.

Descriere generală

Dispozitivul constă într-un recipient mai mare în care poți stoca hrana preferată a pisicii tale. Folosind un servomotor, Feline Feeder eliberează automat porțiuni din mâncare la orele programate de tine. Setezi cu ușurință ora de hrănire folosind o tastatură matriceală, iar informațiile sunt afișate clar pe un ecran LCD.

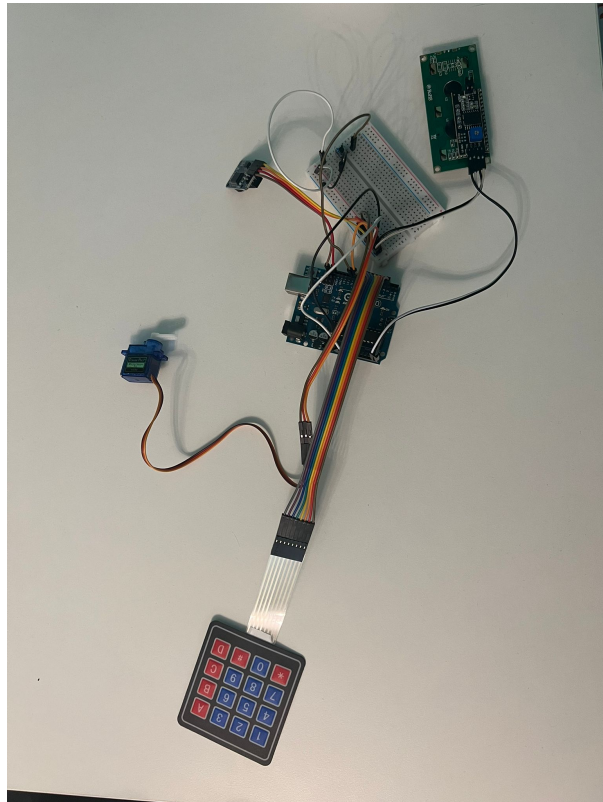
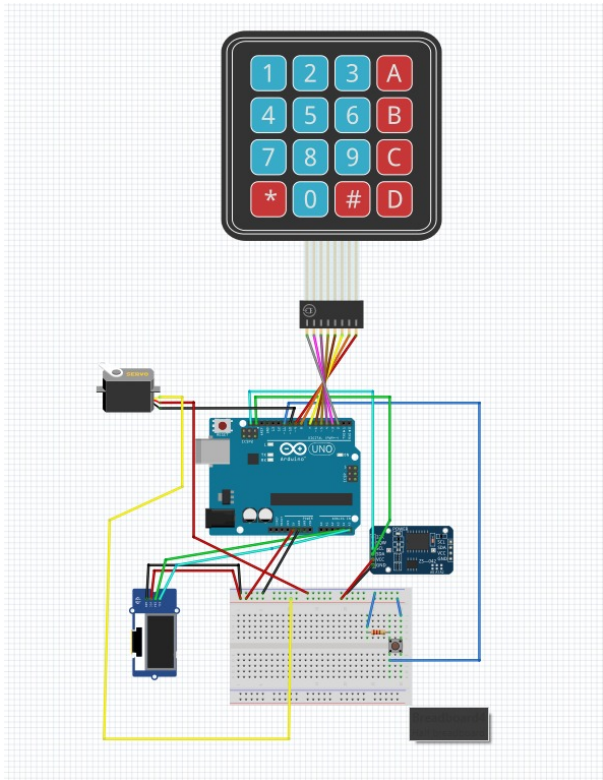


Hardware Design

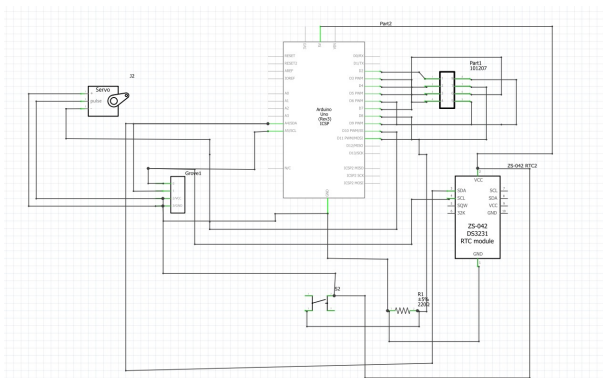
Piese:

- Arduino UNO
- ecran LCD
- buton
- servomotor
- modul RTC DS3231
- tastatura matriceala

Design Circuit



Schemă Electrică



Software Design

Mediu de Dezvoltare

- Arduino IDE

Librării și surse 3rd-party

- **Wire.h**: Biblioteca standard Arduino pentru comunicarea I2C.
- **RTClib.h**: Biblioteca pentru manipularea modulelor RTC, cum ar fi DS3231.
- **LiquidCrystal_I2C.h**: Biblioteca pentru controlul afișajului LCD I2C.
- **Servo.h**: Biblioteca pentru controlul servo motoarelor.
- **Keypad.h**: Biblioteca pentru interfațarea cu un keypad matricial.

Algoritmi și structuri implementate

1. Inițializare și configurare hardware:

- Configurarea modulelor RTC, LCD și servo.
- Configurarea interfeței cu keypad-ul.
- Configurarea unui timer pentru controlul duratei de activare a servo-ului.
- Configurarea unui buton pentru intrarea în modul de setare a orei de hrănire.

2. Funcționalitate principală:

- Afișarea timpului curent pe afișajul LCD.
- Detectarea și intrarea în modul de setare a timpului de hrănire când se apasă butonul.
- Permite utilizatorului să introducă ora de hrănire utilizând keypad-ul.
- Stocarea și afișarea orei de hrănire setate.

3. Controlul hrănirii automate:

- Compararea orei curente cu ora de hrănire setată.
- Activarea servo-ului pentru a deschide compartimentul de hrană la ora setată.
- Folosirea unui timer pentru a menține servo-ul activat pentru o perioadă specificată de timp.

Surse și Funcții Implementate

1. Configurarea hardware:

void configure_timer2(): Configurarea timerului 2 pentru generarea unei întreruperi la fiecare 1 ms.

```
void configure_timer2()
{
  TCCR2A = 0;
  TCCR2B = 0;
  TCNT2 = 0;
  OCR2A = 249;
  TCCR2A |= (1 << WGM21);
  TCCR2B |= (1 << CS22);
}
```

```
}
```

void init_timer2(): Inițializarea timerului 2.

```
void init_timer2()
{
    TIMSK2 |= (1 << OCIE2A);
}
```

void configure_button(): Configurarea butonului de intrare în modul de setare a orei de hrănire.

```
void configure_button()
{
    DDRD&=~(1<<PD2);
    PORTD|=(1<<PD2);
    PCICR |= (1 << PCIE2);
    PCMSK2 |= (1 << PCINT18);
}
```

2. Funcții ISR (Interrupt Service Routine):

ISR(TIMER2_COMPA_vect): Funcția de întrerupere pentru Timerul 2 care controlează durata de activare a servo-ului.

```
ISR(TIMER2_COMPA_vect)
{
    if (timer2Flag)
    {
        unsigned long currentMillis = millis();
        if (currentMillis - previousMillis >= interval)
        {
            servo.write(0);
            timer2Flag = false;
        }
    }
}
```

ISR(PCINT2_vect): Funcția de întrerupere pentru butonul de setare a orei de hrănire.

```
ISR(PCINT2_vect)
{
    if (!(PIND & (1 << PD2)))
    {
        setTimeMode = true;
    }
}
```

3.Setarea timpului de hrănire:

DateTime getDateTime(): Funcția care permite utilizatorului să introducă ora de hrănire folosind keypad-ul și returnează un obiect DateTime cu ora setată.

```
DateTime getDateTime()
{
    DateTime now = rtc.now();
    int hour = 0;
    int minute = 0;
    int h = 0;
    int m = 0;
    while (1)
    {
        char key;
        if (key = keypad.getKey())
        {
            if (key >= '0' && key <= '9')
            {
                if(h == 0)
                {
                    hour += (key - '0');
                    h = 1;
                }
                else if(h == 1)
                {
                    hour *= 10;
                    hour += (key - '0');
                    h = 2;
                }
                else if(m == 0)
                {
                    minute += (key - '0');
                    m = 1;
                }
                else if(m == 1)
                {
                    minute *= 10;
                    minute += (key - '0');
                    m = 2;
                }
                lcd.setCursor(0, 1);
                if(hour<10 && h==2)
                {
                    lcd.print("0");
                }
                lcd.print(hour);
                lcd.print(":");
                if(minute<10 && m==2)
                {
                    lcd.print("0");
                }
                lcd.print(minute);
            }
        }
    }
}
```

```
    }
    else if (key == '*')
    {
        hour = 0;
        minute = 0;
        h = 0;
        m = 0;
        lcd.setCursor(0, 1);
        lcd.print("      ");
        lcd.setCursor(0, 1);
        lcd.print("00:00");
    }
    else if (key == '#')
    {
        if (hour < 24 && minute < 60)
        {
            break;
        }
        else
        {
            hour = 0;
            minute = 0;
            h = 0;
            m = 0;
            lcd.clear();
            lcd.print("The time is");
            lcd.setCursor(0, 1);
            lcd.print("incorrect.");
            delay(2000);
            lcd.clear();
            lcd.print("Set feeding time:");
            lcd.setCursor(0, 1);
            lcd.print("      ");
            lcd.setCursor(0, 1);
            lcd.print("00:00");
        }
    }
}
}
}
return DateTime(now.year(), now.month(), now.day(), hour, minute, 0);
}
```

4. Funcții de inițializare:

void setup(): Funcția de setup pentru inițializarea tuturor componentelor hardware și setarea inițială a RTC-ului.

5. Funcția principală de loop:

void loop(): Funcția principală care rulează în buclă infinită, afișând ora curentă și verificând dacă este timpul de hrănire.

Rezultate Obținute

Demo: https://youtube.com/shorts/5oI52AkM5Os?si=_Y6OJJ0E64xwd0GT

Concluzii

Mi-a placut foarte mult sa lucrez la acest proiect. Ma asteptam sa fie mai usor putin dar a fost super interesant sa lucrez la partea de hardware si chiar sa imi iasa :))). Poate voi testa proiectul si cu pisica mea.

Download

Descarcare arhiva: [felinefeeder.zip](#)

Bibliografie/Resurse

<https://www.youtube.com/watch?v=BiWoA81fgTE&list=WL&index=48&t=1s>

<https://circuitdigest.com/microcontroller-projects/automatic-pet-feeder-using-arduino>

<https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/iconstantinescu1206>



Last update: **2024/05/27 02:44**