Whac-A-Mole Led Edition

Autor: *Fuiorea Daniela-Andreea, 331CC*

Introducere

Ce Face

Proiectul reprezinta un joc de tipul Whack-A-Mole, unde iluminarea unui LED reprezinta existenta unei cartite, iar pentru a omori-o, jucatorul trebuie sa apese pe butonul din fata LED-ului. Daca este cartita corecta lovita in timp, se aprinde un LED verde si vibreaza scurt buzzerul. Daca timpul cartitei a trecut fara sa fie lovita sau a fost lovita o cartita gresita, se apinde LED-ul rosu si vibreaza mult buzzerul. Pe un ecran LCD se arata timpul ramas din joc, scorul si cate vieti au ramas. Pentru a porni jocul, se apasa pe butonul de start.

Ideea Originala

Ideea provine de la jocul Whack-A-Mole. Din niste gauri ies cartite aleatoriu care trebuie sa fie lovite de catre jucator pana sa se intoarca inapoi in gaura. Daca jucatorul loveste una, scorul sau creste. Jocul se opreste cand timpul se termina.

Exemplu de Whack-A-Mole

Scopul si Utilitatea

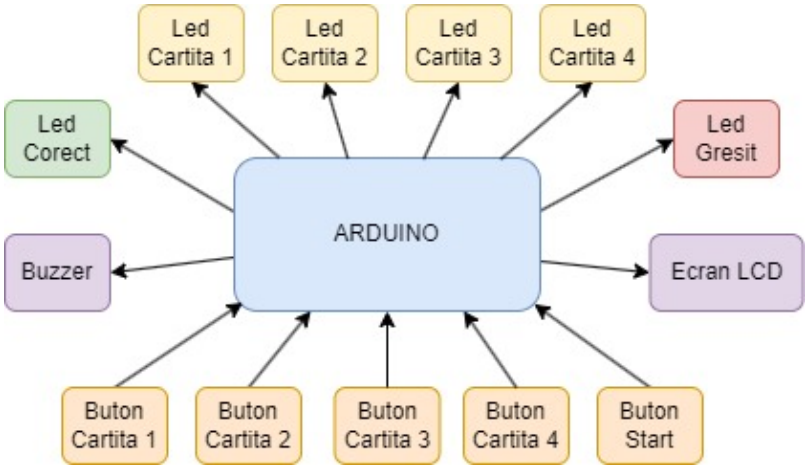
Jocul este o metoda de destindere distractiva. Pe langa acestea, ajuta la coordonarea mana-ochi cat si a reflexelor.

Laboratoare Incluse

- Intreruperi - Apasarea butoanelor pentru omorarea cartitei
- PWM - Intensitatea ledurilor cartita
- I2C - Afisarea stats-urilor jucatorului

Descriere generală

Schema Bloc



Descriere

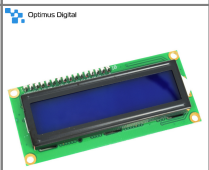
Pentru a porni proiectul se apasa pe butonulde start. O data apasat, jucatorul are 3 vieti, incepe un timer care scade si se aprind un numar intamplator de LED-uri galbene la intervale intamplatoare. Daca butonul corespunzator este apasat, LED-ul se inchide, scorul creste cu 50 de puncte, LED-ul verde se aprinde iar buzzerul vibreaza. Daca se apasa un buton, iar LED-ul corespunzator era inchis, se scade 10 din scor, jucatorul pierde o viata, LED-ul rosu se aprinde, iar buzzerul vibreaza. Scorul nu poate sa scada sub 0. La fiecare mie de puncte, jucatorul isi regenereaza o viata. Exista si cartite false unde LED-ul lumineaza slab, iar daca butonul reprezentativ este apasat, este considerat gresit.

Pe ecranul LCD se afiseaza pe primul rand timpul si scoru, iar pe al doilea rand numarul de vieti. Jocul se termina fie cand timpul ajunge la 00:00, fie cand numarul de vieti ajunge la 0. Daca timpul s-a terminat, pe ecranul LCD, pe primul rand apare mesajul “TIMP EXPIRAT”, iar pe al doilea rand scorul obtinut. Daca numarul de vieti a ajuns la 0, pe ecranul LCD, pe primul rand apare mesajul “AI MURIT”, iar pe al doilea rand scorul obtinut. Pentru a reincepe jocul, trebuie sa se apese iar pe butonul de start.

Hardware Design

Lista Piese

Componenta	Cantitate	Imagine	Descriere

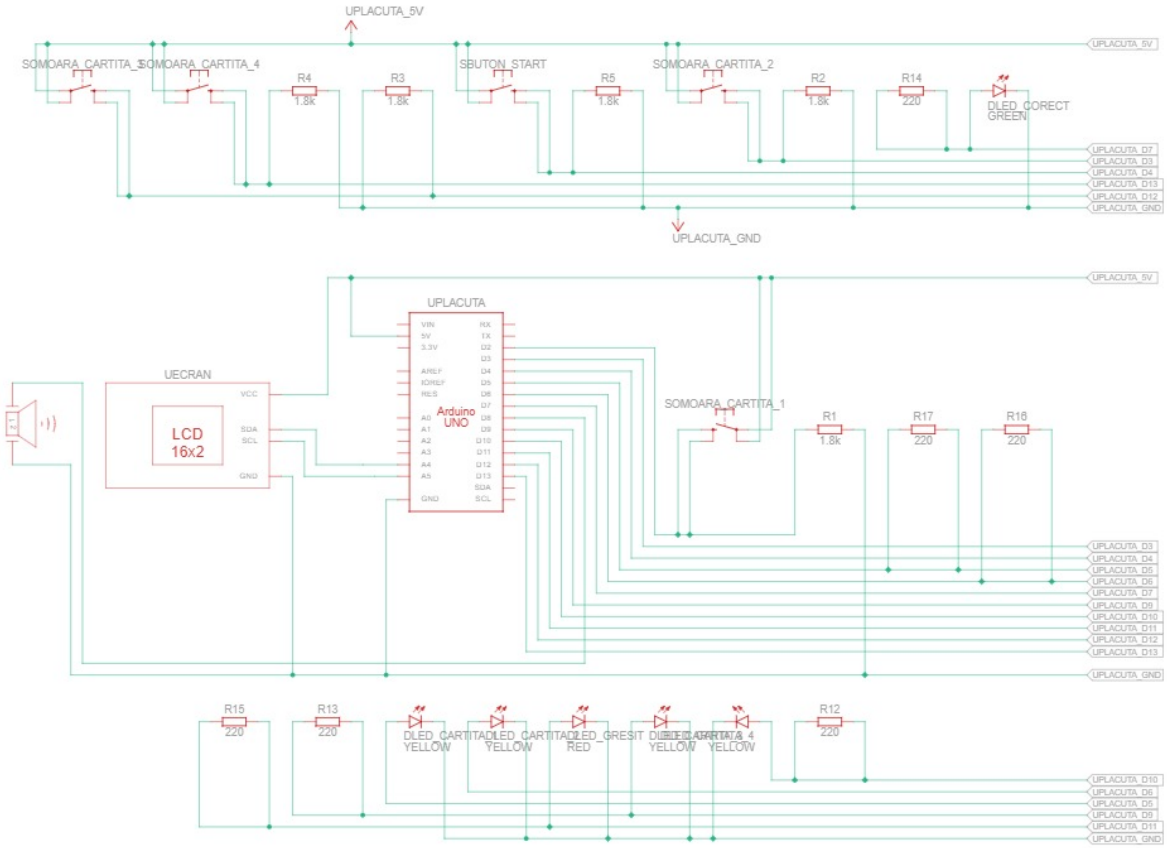
Placuta Arduino Uno R3	1		Placuta care controleaza tot jocul
Ecran LCD 1602 I2C	1		Arata timpul ramas, numarul de vieti si scorul
Buton	5		“Loveste” cartita / Cel de start e pentru oprirea/pornirea jocului
LED Galben	4		Cand e aprins, arata ca acolo se afla o cartita
LED Verde	1		Se aprinde pentru a atentiona jucatorul ca a lovit cartita corecta
LED Rosu	1		Se aprinde pentru a atentiona jucatorul ca a lovit cartita gresita
Buzzer	1		Vibreaza in functie daca jucatorul a lovit cartita corecta sau nu
Rezistor 1.8KΩ	5		Sa functioneze corect butonul
Rezistor 220Ω	6		Sa functioneze corect LED-urile

Piesele au fost comandate de pe Optimus Digital

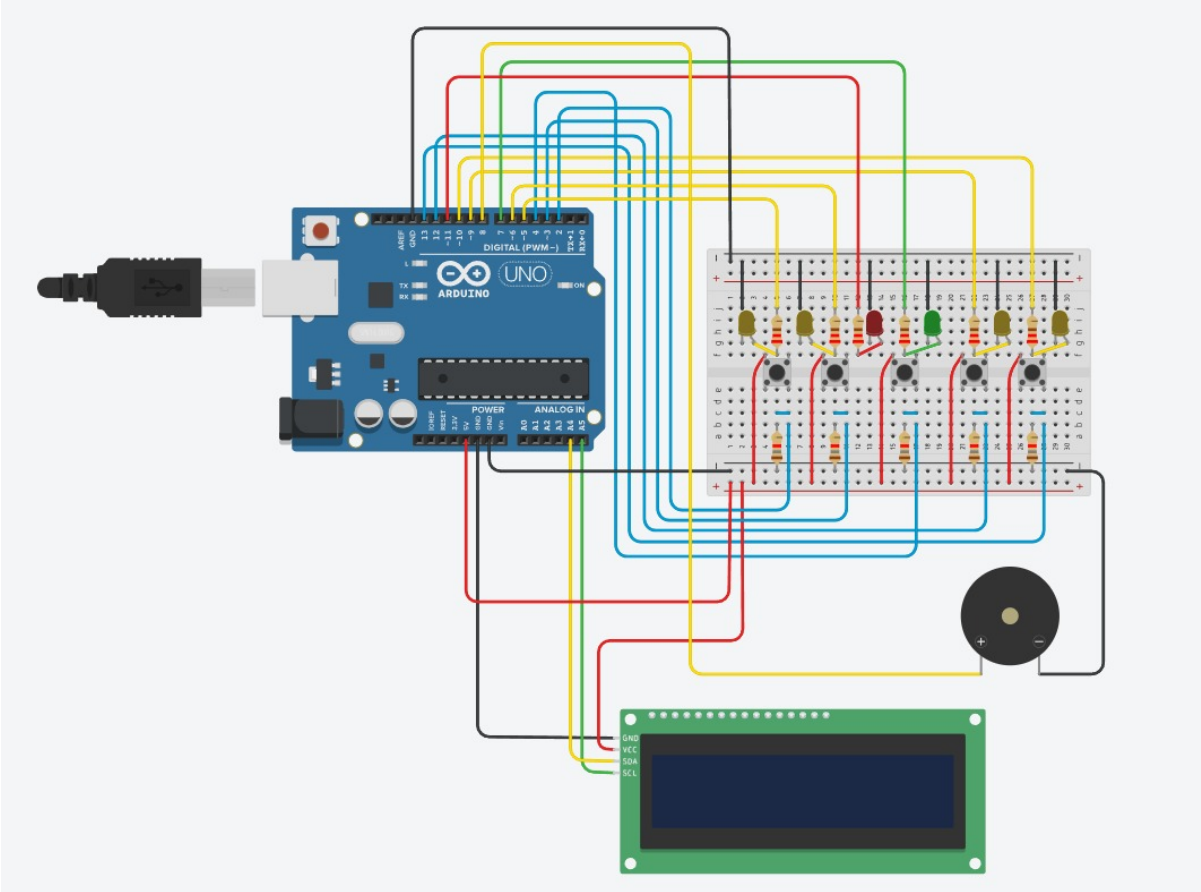
Buzzerul a fost donat ca cel original si-a luat bully UwU

Scheme Electrice

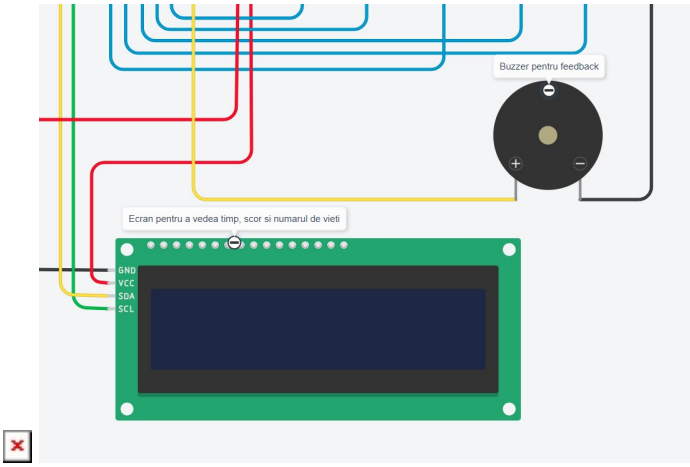
Schema Principala



Schema Fizica



Detalii



Alegerea piniilor

- Pinii 0 si 1 sunt rezervati pentru comunicare seriala, iar eu nu am avut nevoie de ei.
- Pinii 5, 6, 7, 9, 10 si 11 i-am ales pentru LED-uri deoarece accepta PWM (cu exceptia 7 care nu accepta). Initial si LED-ul verde folosea PWM, dar a trebuit sa renunt pentru folosirea pinului pentru intreruperi
- Pinii 2 si 3 sunt pentru butoane cu intreruperi pentru ca sunt singuri pini care accepta intreruperi.
- Pinii 4, 8, 12 si 13 sunt pentru restul butoanelor / buzzer pentru ca doar ei au mai ramas si nu au

nevoie de ceva special.
- Pinii A4 si A5 sunt pentru ecranul LCD. Nu mai aveam loc pe pinii digitali si acestia apareau in datasheet.

Construirea

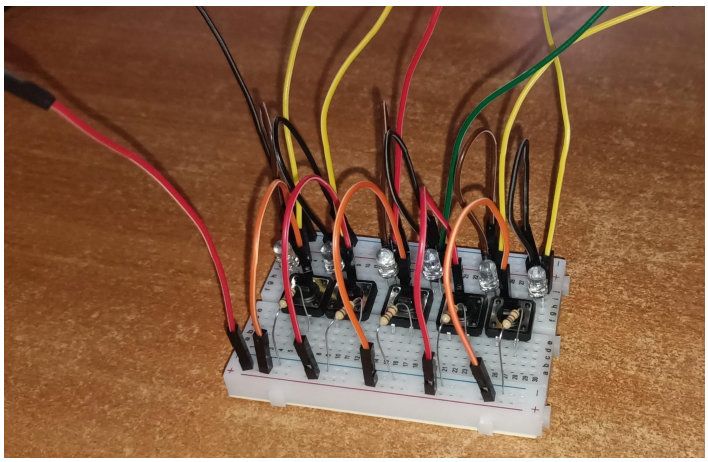
Prototip de Hartie

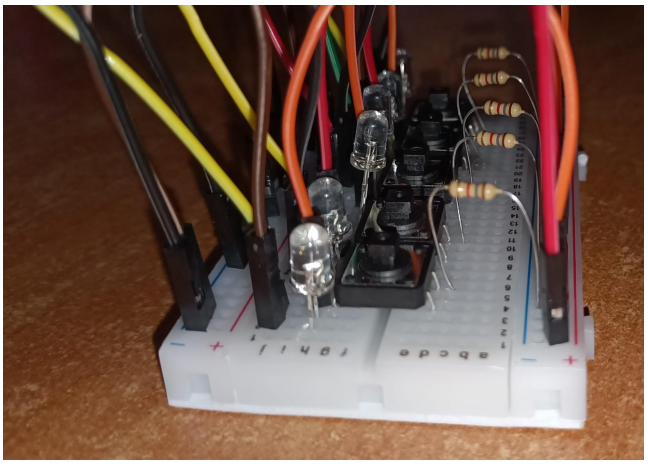
Desi nu are o valoare efectiva prea mare din punct de vedere al hardware-lui, m-a ajutat sa imi dau seama cum ar arata componentele din exterior, de pozitionarea lor. Am vrut sa aiba dimensiuni mici, dar sa si dea impresia de arcade si de o consola retro.


Prototipul pe Breadboard

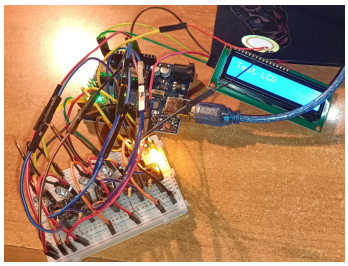
Din cauza unei frici imense de a nu strica ceva, am vrut sa fac initial un prototip pe breadboard, inainte sa lipesc componentele intre ele.

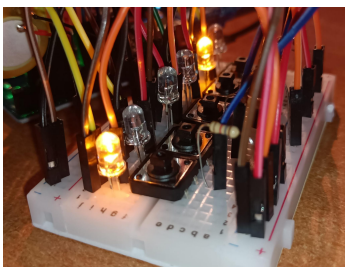
Pentru inceput, am pus butoanele, LED-urile, rezistentele si le-am legat prin fire.

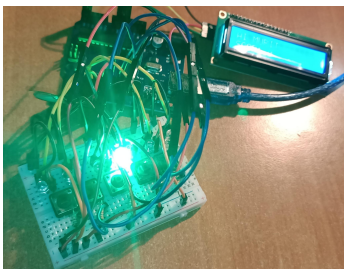


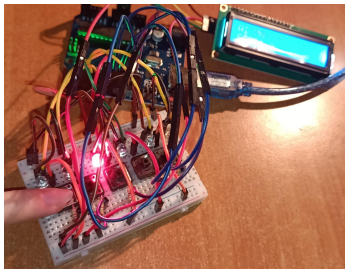


Apoi am conectat si ecranul LCD si buzzerul, luandu-ma o frica ca am firele prea scurte. In final am conectat totul la placuta si m-am rugat sa fie totul bine. (frica legata de firele prea scurte s-a adeverit)







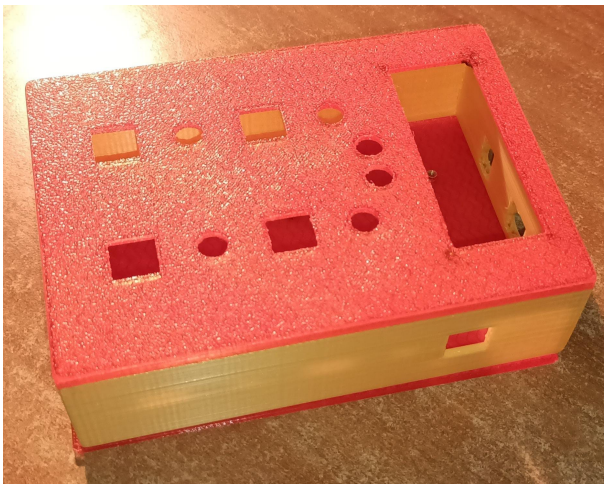


Creearea Carcasei

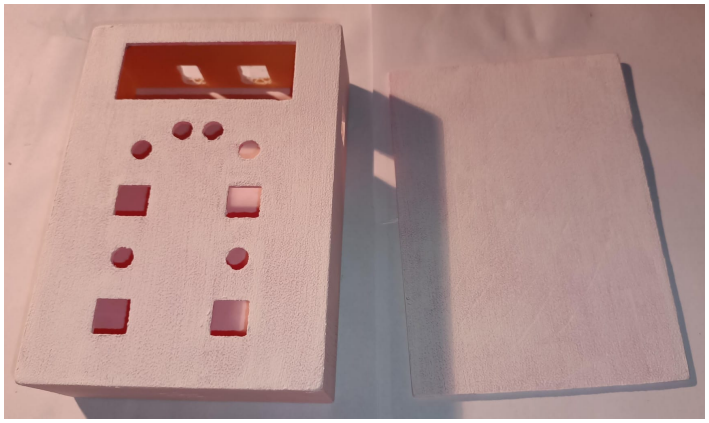
Pentru a face o carcasa in care sa intre componentele si sa fie prinse bine, am cautat dimensiunile lor si am inceput sa proiectez in AutoCAD cum ar trebuii sa fie carcasa. Pentru componentele la care nu am gasit dimensiuni, am masirat cu rigla componentele mele.

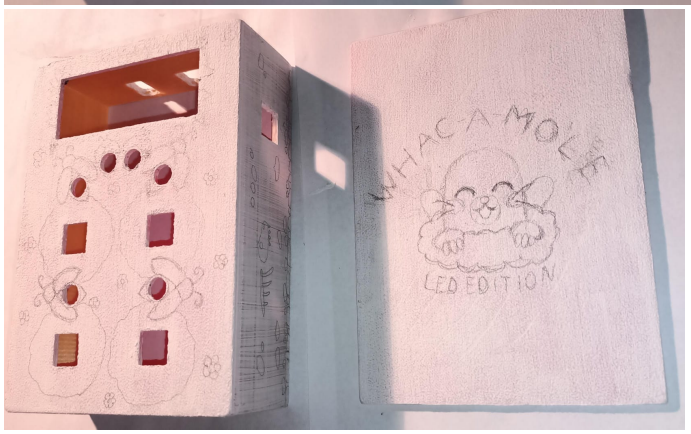
Apoi in Blender am facut modelul 3D. Spatele initial este separat, urmand sa fie lipit in final. I-am facut o usita laterala care se poate trage pentru a vedea interiorul. Usita si carcasa au un mecanism de legare pentru a sta in loc si a nu cadea cand este intoarsa.

Multumesc enorm Victor Stoica pentru printarea 3D a modelului!

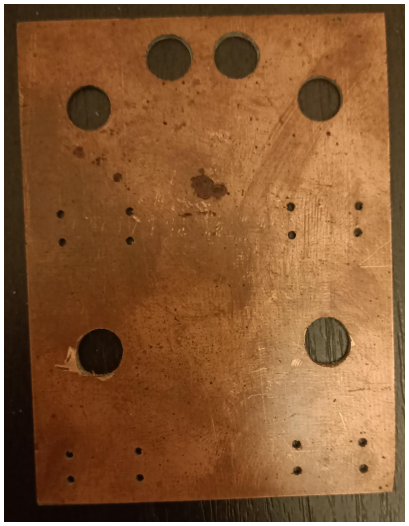
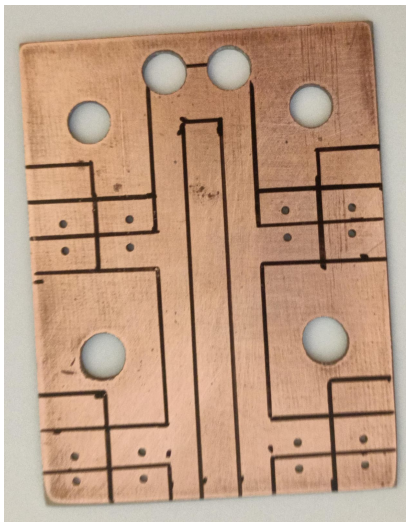
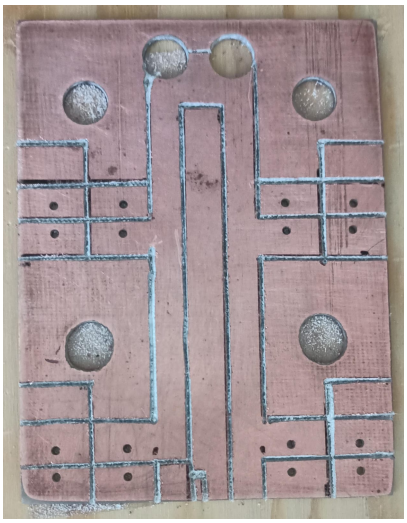
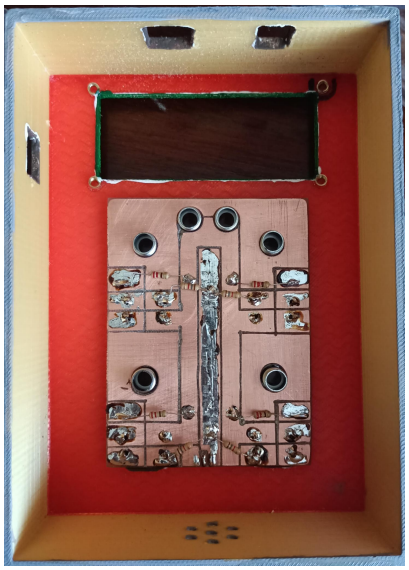
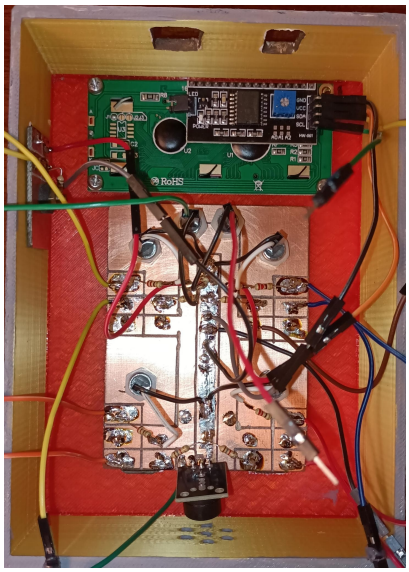
Am inceput sa pictez cu mai multe straturi de alb carcasa pentru a putea face designul. Dupa ce am terminat de pictat cu alb, am facut in creion conturul designului. Dupa am inceput sa pictez carcasa. In final am dat-o cu un lac pentru a sigila vopselele acrilice.



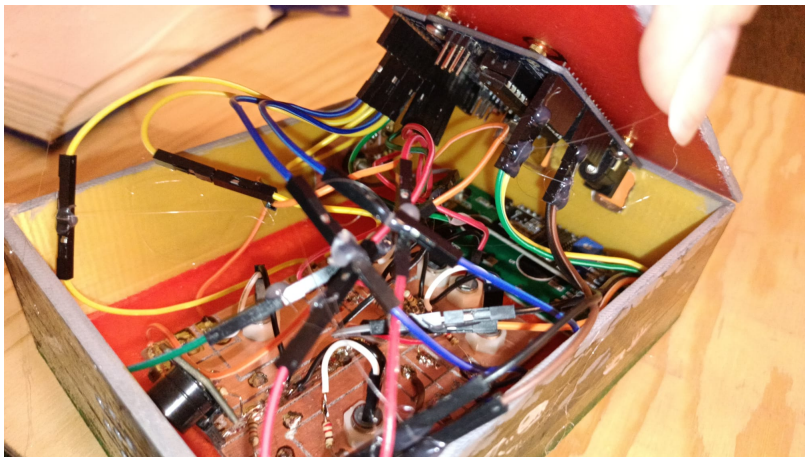


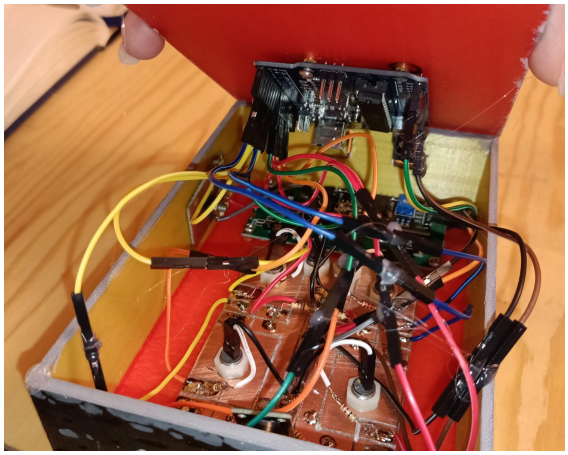
Lipirea Componentelor

Am folosit o placuta de cupru pe care am gaurit-o. Am trasat negativele pe ea pentru a putea conecta componentele si a asigura rezistenta butoanelor. Am lipit firele, rezistentele, buzzerul si butoanele si am conectat LED-urile. Be placa de baza am conectat si ecranul LCD si butonul de start. Pe mijloc este GND-ul, iar VCC-ul este in jurul sau. In rest, sunt bine partitionate componentele. Firele au fost, intradevar, prea scurte.

Carcasa a fost lipita asa ca niste poze ca sa se vada ce e in interior (Dumnezeu cu mila):





Software Design

Mediul de Dezvoltare

Demo-ul proiectului a fost implementat folosind TinkerCAD si poate fi gasit aici.

Programul efectiv a fost facut folosind ArduinoIDE.

Biblioteci Utilizate

Am folosit biblioteca “LiquidCrystal_I2C.h” pentru a putea afisa cu usurinta pe ecranul LCD.

Functii Implementate

Initializarea jocului

Aceasta parte a codului incepe cand se porneste proectul (prima parte) si cand un joc nu este in functiune iar butonul de start a fost apasat (a doua parte). Aici fiecare variabila de care avem nevoie in joc primeste valori necesare pentru a incepe unjoc.

```
void setup() {
    DDRD |= (1 << DDD5) | (1 << DDD6) | (1 << DDD7);
    DDRB |= (1 << DDB1) | (1 << DDB2) | (1 << DDB3);
    DDRB |= (1 << DDB0);
    DDRC |= (1 << DDC0);
    DDRD &= ~((1 << DDD2) | (1 << DDD3));
    DDRB &= ~((1 << DDB4) | (1 << DDB5));
    DDRD &= ~(1 << DDD4);
```

```
      attachInterrupt(digitalPinToInterrupt(butonCartita[0]),
ISR_butonCartita0, CHANGE);
      attachInterrupt(digitalPinToInterrupt(butonCartita[1]),
ISR_butonCartita1, CHANGE);
      randomSeed(analogRead(A0));
      ecran.init();
      ecran.setCursor(0, 0);
      ecran.backlight();
      ecran.display();
  }
```

```
  void incep() {
      joc = true;
      scor = 0;
      vieti = 4;
      nrCartite = 0;
      delayCartite = 0;
      timp = millis() + 180000;
      for (int i = 0; i < 4; i++) {
          timerCartita[i] = 0;
          onCartita[i] = false;
          debounceTimes[i] = 0;
      }
      timerRosu = 0;
      timerVerde = 0;
      onVerde = false;
      onRosu = false;
      debounceStart = millis();
      for(int i = 0; i < 10; i++)
          bonusHP[i] = true;
      ecran.clear();
  }
```

Jocul Efectiv

Aici este bucla unde se apeleaza functiile pentru functionarea jocului.

```
  void loop() {
      if ((PIND & (1 << PIND4)) && !joc && debounceStart + 500 < millis()) {
          incep();
      }
      if (joc && timp > currentMillis && vieti > 0) {
          currentMillis = millis();
          if (scor < 0)
          scor = 0;
          afisare();
          viata();
          logicaCartite();
```

```
          genereazaCartite();
          afisLedVerde();
          afisLedRosu();
          if (PINB & (1 << PINB4))
              lovireCartite(2);
          if (PINB & (1 << PINB5))
              lovireCartite(3);
          if ((joc && vieti == 0) || (timp <= currentMillis))
              final();
          if ((PIND & (1 << PIND4)) && joc && debounceStart + 500 <
millis())
              final();
      }
  }
```

Alegerea Numarului de Cartite, LED-urilor si Timpul de Aprindere

Noi cartite apar atunci cand nu mai sunt cartite vizibile (nici un led nu mai e aprins) si s-a asteptat un timp pentru ai da jucatorului putin timp pentru a se relaxa (foarte putin timp). Pentru asigurarea functionarii corecte, se inchid toate LED-urile si se opreste buzzerul. Se alege un numar aleator intre 0 si 4 si apoi se alege aleator un pin al LED-urilor. Se poate sa fie ales un LED care deja a fost aprins, dar nu are nici un impact asupra jocului. Timpul in care un LED cartita este aprins, este aleatoriu pentru o recujabilitate mai mare, cat si ingreunarea jocului. Dupa se aleg un numar aleator de “cartite false” intre o si 4.

```
  void genereazaCartite() {
      if (nrCartite <= 0 && delayCartite <= currentMillis) {
          PORTD &= ~(1 << PORTD7);
          PORTB &= ~(1 << PORTB3);
          TCCR0A &= ~(1 << COM0B1);
          TCCR0A &= ~(1 << COM0A1);
          TCCR1A &= ~(1 << COM1A1);
          TCCR1A &= ~(1 << COM1B1);
          noTone(buzzer);
          timerVerde = 0;
          timerRosu = 0;
          int nr = random(0, 4);
          for (int i = 1; i <= nr; i++) {
              int ledIndex = random(0, 4);
              if (!onCartita[ledIndex]) {
                  int timp = random(750, 1250);
                  timerCartita[ledIndex] = currentMillis + timp;
                  onCartita[ledIndex] = true;
                  nrCartite++;
              }
          }
          nr = random(0, 4);
          for (int i = 1; i <= nr; i++) {
```

```
            int ledIndex = random(0, 4);
            if (!onCartita[ledIndex]) {
                cartitaFalsa[ledIndex] = true;
            }
        }
    }
}
```

Aprinderea si Stingerea Unui LED

Deoarece modul de alegere este aleatoriu, trebuie sa avem in vedere cand aprindem un LED ca acesta nu este aprins inainte. La aprindere, retinem intr-o variabila ca a fost aprins si de asemenea retinem cat timp trebuie sa fie aprins si creste numarul de cartite.

LED-ul se stinge dupa un anumit timp daca butonul corespunzator nu a fost apasat. Atunci variabila retine ca este inchis, scorul scade cu 10, iar numarul de cartite scade. Daca nu mai sunt cartite, atunci incepe un “delay” pana la aprinderea urmatoarelor. Nu am folosit functia efectiva “delay()” pentru ca aceasta blockeaza tot programul, dar eu am nevoie ca sa considere si in acel timp apasarea butoanelor.

Daca timpul de iluminare al LED-ului nu s-a terminat, acesta lumineaza. Daca LED-ul este “cartita falsa” atunci lumineaza putin pentru a incurca jucatorul.

```
void logicaCartite() {
    for (int i = 0; i < 4; i++) {
        if (timerCartita[i] <= currentMillis && onCartita[i]) {
            if (ledCartita[i] == 5) {
                TCCR0A &= ~(1 << COM0B1);
                PORTD &= ~(1 << PORTD5);
            } else if (ledCartita[i] == 6) {
                TCCR0A &= ~(1 << COM0A1);
                PORTD &= ~(1 << PORTD6);
            } else if (ledCartita[i] == 9) {
                TCCR1A &= ~(1 << COM1A1);
                PORTB &= ~(1 << PORTB1);
            } else if (ledCartita[i] == 10) {
                TCCR1A &= ~(1 << COM1B1);
                PORTB &= ~(1 << PORTB2);
            }
            nrCartite--;
            if (nrCartite == 0) {
                delayCartite = currentMillis + 1000;
                for (int j = 0; j < 4; j++) {
                    if (ledCartita[j] == 5) {
                        PORTD &= ~(1 << PORTD5);
                    } else if (ledCartita[j] == 6) {
                        PORTD &= ~(1 << PORTD6);
                    } else if (ledCartita[j] == 9) {
                        PORTB &= ~(1 << PORTB1);
```

```
                    } else if (ledCartita[j] == 10) {
                        PORTB &= ~(1 << PORTB2);
                    }
                    cartitaFalsa[j] = false;
                }
            }
            onCartita[i] = false;
            scor -= 10;
        }
    }
    for (int i = 0; i < 4; i++) {
        if(cartitaFalsa[i]) {
            if (ledCartita[i] == 5) {
                TCCR0A |= (1 << COM0B1);
                OCR0B = 1;
            } else if (ledCartita[i] == 6) {
                TCCR0A |= (1 << COM0A1);
                OCR0A = 1;
            } else if (ledCartita[i] == 9) {
                TCCR1A |= (1 << COM1A1);
                OCR1A = 1;
            } else if (ledCartita[i] == 10) {
                TCCR1A |= (1 << COM1B1);
                OCR1B = 1;
            }
        }
        if (timerCartita[i] > currentMillis) {
            if (ledCartita[i] == 5) {
                TCCR0A |= (1 << COM0B1);
                OCR0B = 255;
            } else if (ledCartita[i] == 6) {
                TCCR0A |= (1 << COM0A1);
                OCR0A = 255;
            } else if (ledCartita[i] == 9) {
                TCCR1A |= (1 << COM1A1);
                OCR1A = 255;
            } else if (ledCartita[i] == 10) {
                TCCR1A |= (1 << COM1B1);
                OCR1B = 255;
            }
        }
    }
}
```

Apasarea Unui Buton

Daca LED-ul corespunzator este aprins, se intampla ca la stingerea de LED, dar scorul creste cu 50, si LED-ul verde urmeaza sa fie aprins, numarul cartitelor scade si daca e ultima, ca mai sus, incepe un

“delay”. Daca LED-ul era inchis, atunci scade o viata si urmeaza sa se aprinda LED-ul rosu. Si LED-ul rosu, si cel verde il va opri pe celalat, luminand doar unul o data. Va lumina cea mai recenta actiune. Daca o actiune de acelasi tip se repeta pana la terminarea urmatoarei, timpul se reseteaza.

Primele 2 butoane folosesc intreruperi care apeleaza functia de lovire a cartitelor. Ultimele doua o apeleaza direct fara sa foloseasca intreruperi.

```
void ISR_butonCartita0() {
    lovireCartite(0);
}
```

```
void ISR_butonCartita1() {
    lovireCartite(1);
}
```

```
void lovireCartite(int index) {
    if (debounceTimes[index] < currentMillis) {
        debounceTimes[index] = currentMillis + 500;
        if (onCartita[index]) {
            timerCartita[index] = currentMillis;
            nrCartite--;
            if (nrCartite == 0) {
                    delayCartite = currentMillis + 1000;
                    for (int i = 0; i < 4; i++) {
                        if (ledCartita[i] == 5) {
                            PORTD &= ~(1 << PORTD5);
                        } else if (ledCartita[i] == 6) {
                            PORTD &= ~(1 << PORTD6);
                        } else if (ledCartita[i] == 9) {
                            PORTB &= ~(1 << PORTB1);
                        } else if (ledCartita[i] == 10) {
                            PORTB &= ~(1 << PORTB2);
                        }
                        cartitaFalsa[i] = false;
                    }
                }
            onCartita[index] = false;
            if (ledCartita[index] == 5) {
                TCCR0A &= ~(1 << COM0B1);
                PORTD &= ~(1 << PORTD5);
            } else if (ledCartita[index] == 6) {
                TCCR0A &= ~(1 << COM0A1);
                PORTD &= ~(1 << PORTD6);
            } else if (ledCartita[index] == 9) {
                TCCR1A &= ~(1 << COM1A1);
                PORTB &= ~(1 << PORTB1);
            } else if (ledCartita[index] == 10) {
                TCCR1A &= ~(1 << COM1B1);
                PORTB &= ~(1 << PORTB2);
            }
            scor += 50;
```

```
            onVerde = true;
            PORTB &= ~(1 << PORTB3);
            noTone(buzzer);
            onRosu = false;
            timerVerde = currentMillis + 100;
        } else {
            vieti--;
            onRosu = true;
            PORTD &= ~(1 << PORTD7);
            noTone(buzzer);
            onVerde = false;
            timerRosu = currentMillis + 100;
        }
    }
}
```

Cartita Corecta

LED-ul verde se aprinde iar buzzerul vibreaza pe o frecventa inalta pentru ca jucatorul sa primeasca un feedback. Cand timpul de a fi aprins se termina, LED-ul se inchide si buzzerul se opreste.

```
void afisLedVerde() {
    if (timerVerde <= currentMillis && onVerde) {
        PORTD &= ~(1 << PORTD7);
        noTone(buzzer);
        onVerde = false;
    }
    if (onVerde) {
        PORTD |= (1 << PORTD7);
        tone(buzzer, 10000);
    }
}
```

Cartita Gresita

LED-ul rosu se aprinde iar buzzerul vibreaza pe o frecventa joasa pentru ca jucatorul sa primeasca un feedback. Cand timpul de a fi aprins se termina, LED-ul se inchide si buzzerul se opreste.

```
void afisLedRosu() {
    if (timerRosu <= currentMillis && onRosu) {
        PORTB &= ~(1 << PORTB3);
        noTone(buzzer);
        onRosu = false;
    }
    if (onRosu) {
        PORTB |= (1 << PORTB3);
```

```
        tone(buzzer, 500);
    }
}
```

Crestere Numar Vieti

La fiecare mie de punte atinsa de jucatoru, viata lui se regenereaza, crescand cu unu daca viata nu este plina.

```
void viata() {
    if(bonusHP[scor / 1000] && vieti < 4 && scor >= 1000) {
        bonusHP[scor / 1000] = false;
        vieti++;
    }
}
```

Afisarea pe Ecranul LCD

Se afiseaza pe ecran pe primul rand timpul pe care il are jucatoru in minute si secunde si numarul de vieti, unde O reprezinta o viata existenta, iar X reprezinta o viata. Jucatorul are 4 incercari, iar pentru fiecare viata pierduta, un O se transforma in X, iar Pe al doilea rand se afiseaza scorul.

```
void afisare() {
    ecran.setCursor(0, 0);
    ecran.print("0");
    int minute = ((timp - currentMillis) / 1000) / 60;
    int secunde = ((timp - currentMillis) / 1000) % 60;
    ecran.setCursor(1, 0);
    ecran.print(minute);
    ecran.setCursor(2, 0);
    ecran.print(":");
    if (secunde >= 10) {
        ecran.setCursor(3, 0);
        ecran.print(secunde);
    } else {
        ecran.setCursor(3, 0);
        ecran.print("0");
        ecran.setCursor(4, 0);
        ecran.print(secunde);
    }
    ecran.setCursor(5, 0);
    ecran.print("  VIETI:");
    for (int i = 1; i < vieti; i++) {
        ecran.setCursor(12 + i, 0);
        ecran.print("O");
    }
```

```
    for (int i = vieti; i <= 3; i++) {
        ecran.setCursor(12 + i, 0);
        ecran.print("X");
    }
    ecran.setCursor(0, 1);
    ecran.print("SCOR: ");
    ecran.setCursor(6, 1);
    ecran.print(scor);
}
```

Terminarea Jocului

In functie de modul de terminare a jocului, jucatorul o sa primeasca un mesaj diferit urmat de scor. Daca a murit, va primi mesajul “AI MURIT”, iar daca timpul s-a terminat, primeste mesajul “TIMP EXPIRAT”. Toate LED-urile se sting, iar cel verde si cel rosu incep sa alterneze, iar buzzerul sa vibreze pentru atentionarea finalizarii jocului. Dupa terminarea alternarii, se sting, ecranul devine gol iar variabila jocului devine falsa.

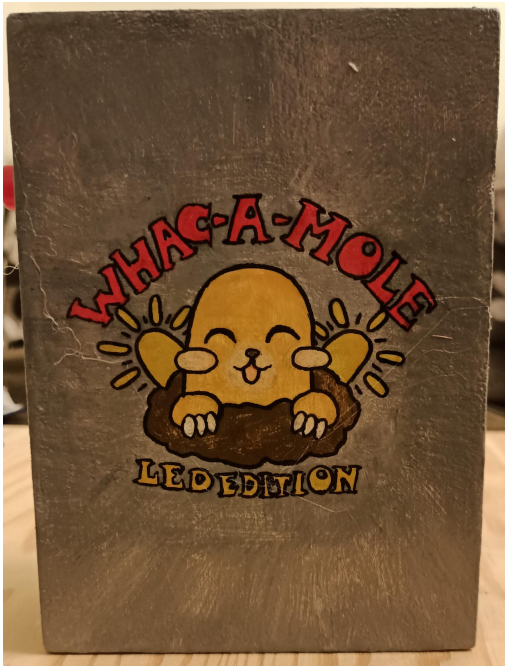
Daca jocul este in derulare, iar butonul de start este apasat, se sting toate LED-URILE si jocul se inchide. Pentru a fi repornit trebuie apasat butonul de start iar.

```
void final() {
    PORTD &= ~(1 << PORTD7);
    PORTB &= ~(1 << PORTB3);
    TCCR0A &= ~(1 << COM0B1);
    TCCR0A &= ~(1 << COM0A1);
    TCCR1A &= ~(1 << COM1A1);
    TCCR1A &= ~(1 << COM1B1);
    if (vieti == 0 || timp <= currentMillis) {
        ecran.clear();
        ecran.setCursor(0, 0);
        ecran.print(vieti == 0 ? "AI MURIT" : "TIMP EXPIRAT");
        ecran.setCursor(0, 1);
        ecran.print("SCOR: ");
        ecran.setCursor(6, 1);
        ecran.print(scor);
        for (int i = 0; i < 10; i++) {
            if (i % 2 == 0) {
                PORTD |= (1 << PORTD7);
                tone(buzzer, 10000);
                delay(500);
                PORTD &= ~(1 << PORTD7);
                noTone(buzzer);
            } else {
                PORTB |= (1 << PORTB3);
                tone(buzzer, 500);
                delay(500);
                PORTB &= ~(1 << PORTB3);
                noTone(buzzer);
```

```
                }
            }
        } else {
            debounceStart = millis();
        }
        joc = false;
        ecran.clear();
    }
```

Rezultate Obţinute

Proiectul functioneaza complet, avand o structura intuitiva si un design atractiv si inovativ.



Concluzii

Acest proiect este o alegorie a nasterii si cresterii unui copil. Initial am fost confuza cu ce tre sa fac, nestiind de unde sa ma apuc. Apoi incetul cu incetul m-am atasat de el, primele sunete, primele luminite, parca mi se facea un drag de el. Apoi a venit partea tumultoasa, imi venea sa imi rup parul din cap, eram satula peste cap de tot, nu mai suportam sa il vad, parca nimic nu merge bine. In final, dupa tot timpul asta, sunt mandra de ce a ajuns.

Workflow-ul meu, in opinia mea, m-a ajutat mai mult decat daca urmam efectiv deadline-urile impuse. Sa creez mai intai un demo pentru a ma asigura ca nu ard componente si apoi sa incep scriu codul, a fost un boost de start foarte bun. Apoi prototiparea pe breadboard (fara a lipi direct componentele) m-a dat siguranta ca toate componentele functioneaza, cat si o testare rapida a modificarilor aduse ulterior codului.

Observatii si Ganduri

- Niciodata sa nu iei prea putine componente ca sunt sanse sa se buleasca.
- DEBOUNCE!
- Trebuie sa testezi de multe ori, mai ales daca e un joc pentru ca pot aparea bugguri oricand.
- Daca merge pe demo, nu inseamna ca merge la fel de bine si pe prototip.

- Surprinzator ca ce inveti la optionale in facultatea asta chiar ajuta (macar la alte materii), am fost placut surprinsa ca mai stiu sa lucrez in AutoCAD dupa 2 ani.
- Cred ca sunt trend setter de tabele lmao
- Trebuia sa ma uit ce pini accepta PWM si intreruperi inainte sa ma apuc...
- Cine a zis ca nu pot sa fac arta la politehnica s-a inselat!
- Daca adaugi apa in vernis poliere devine albicios cu o textura ciudata, dar la uscare e ok si parca se usuca is mai repede
- Inca nu sunt sigura cu ce se dilueaza vernis poliere, dar cu apa e clar ca nu
- Parca mai mergea inca niste timp sa nu ajung sa fac aproape all nighter...
- Nu imi place sa lipesc componente

Obstacole Intalnite

- Am uitat de existenta debounce-ului, neintelegand de ce imi ia de mai multe ori butonul. Acum mi-am adus aminte.
- In timpul creeri demo-ului virtual, simularea din TinkerCAD se misca greu, dand impresia ca ar fi o problema cu codul.
- Pierderea la ce am scris pe OCW pentru ca m-a delogat si nu amm observat...
- Un led era ars, noroc ca am luat mai multe
- Nu mergea ecranul LCD chiar daca gasisem adresa (trebuia sa insurubez/desurubez chestia albastra)
- Buzzerul Piezo avea dorinte sinucigase si nu voia sa stea in fire asa ca a fost pus in breadboard pt prototip
- A trebuit sa modific o parte din codul de la prototip pentru ca nu mergea cum ma asteptam
- Butoanele nu faceau contact bine cu breadboardul, a trebuit sa fac pe chirurgul
- Firele erau prea scurte pentru prototip (chiar nu ma pricep la aproximare)
- Modelul facut in BLender nu avea scalarea buna
- Daca un mecanism merge la mine in cap, nu inseamna ca merge si in realitate
- Nu am luat in considerare grosimea si niste gauri nu erau bine puse
- NU TOTI PINII ACCEPTA PWM SI INTRERUPERI
- E greu sa deschizi sticla de vernis poliere de la atelier
- Nu stiu sa planuiesc inainte si era sa belesc toata placuta de cupru

Download

mole.zip contine fisierul .ino unde a fost facut programul si un fisier upload.sh pentru compilare si bagatul automat pe placuta.

Jurnal

02.05.2024: Am creat pagina proiectului.

04.05.2024: Am creat schema bloc pentru logica proiectului.

05.05.2024: Am creat in TinkerCAD schema electrica a proiectului.

07.05.2024: Am terminat de scris codul pentru demo-ul virtual simulat in TinkerCAD si poate fi gasit aici.

08.05.2024: Am dat comanda de componentele necesare.

09.05.2024: Am preluat componentele si am inceput sa prototipez pe un breadboard.

10.05.2024: Am terminat prototipul si am modificat putin codul (am uitat iar de debounce pentru butonul de start, am modificat timpii si am scos delay-urile din timpul jocului pentru a ma asigura ca se poate apasa un buton cat timp ledul verde sau rosu sunt pornite).

11.05.2024: Am creat in AutoCAD dimensiunile exacte ale carcasei

12.05.2024: Am creat in Blender modelul 3D pentru carcasa folosind dimensiunile din AutoCAD

13.05.2024: Am adus modificari modelului 3D pentru carcasa

15.05.2024: Am primit prima iteratie a carcasei si un capac de buton, dar ulterior am schimbat modelul carcasei pentru una formata din 2 piese. Mi s-a donat un alt buzzer pentru ca sunetul sa se auda bine

16.05.2024: Am primit a doua (si ultima) interatie a carcasei si i s-au pus insertii filetate pentru prinderea placutei si a ecranului.

18.05.2024: Am procurat vopsele acrilice pentru a picta carcasa si am slefuit-o cu un smirghel

19.05.2024: Am pictat cu alb carcasa, am modularizat codul si am adaugat PWM (nu o facusem initial ups) si am modificat cablajul pentru ca nu stiu sa citesc

20.05.2024: Am inceput sa pictez carcasa, am schimbat iar schema electrica pentru a adauga intreruperi (iar nu stiu sa citesc) si am modificat codul pentru a folosii cat mai mult registrii.

21.05.2024: Am adaugat rezistente LED-urilor si am terminat de pictat carcasa si butoanele.

22.05.2024: Am sigilat vopseaua si am gaurit placuta de cupru pentru cablaj.

25.05.2024: Am facut negativele pe placuta de cupru si am inceput sa lipesc componentele.

26.05.2024: Am finalizat lipirea componentelor si a carcasei - proiectul a fost finalizat.

Bibliografie/Resurse

Resurse Software

Medii de Dezvoltare

- TinkerCAD

- ArduinoIDE

Medii de Modelare

- AutoCAD
- Blender
- PrusaSclicer

Resurse Hardware

Distribuitor componente

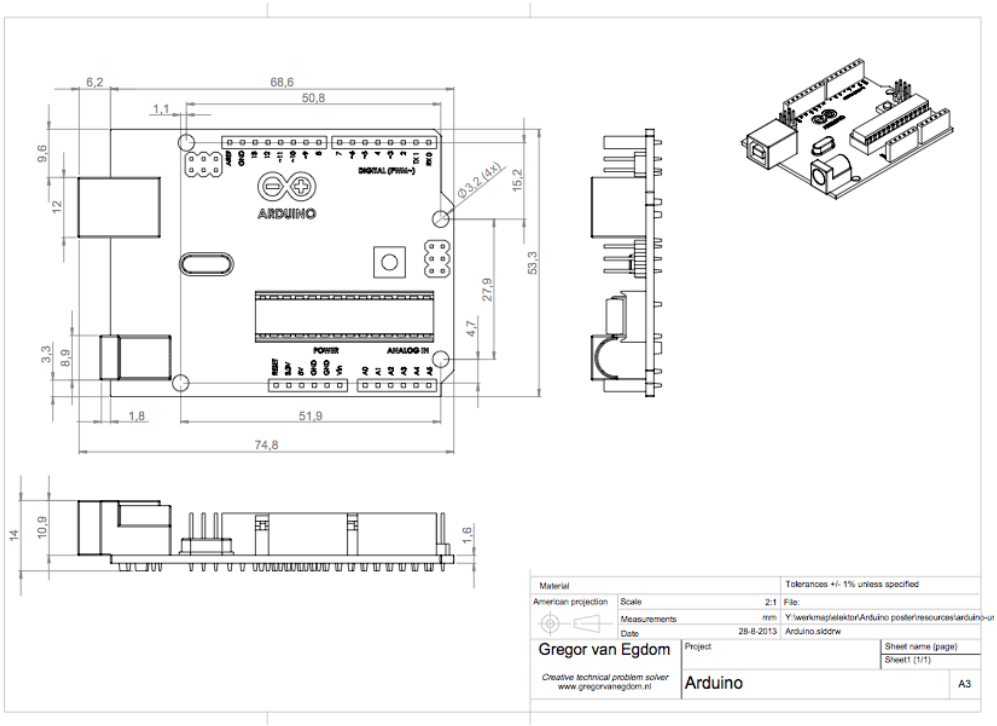
- Optimus Digital

Datasheet-uri

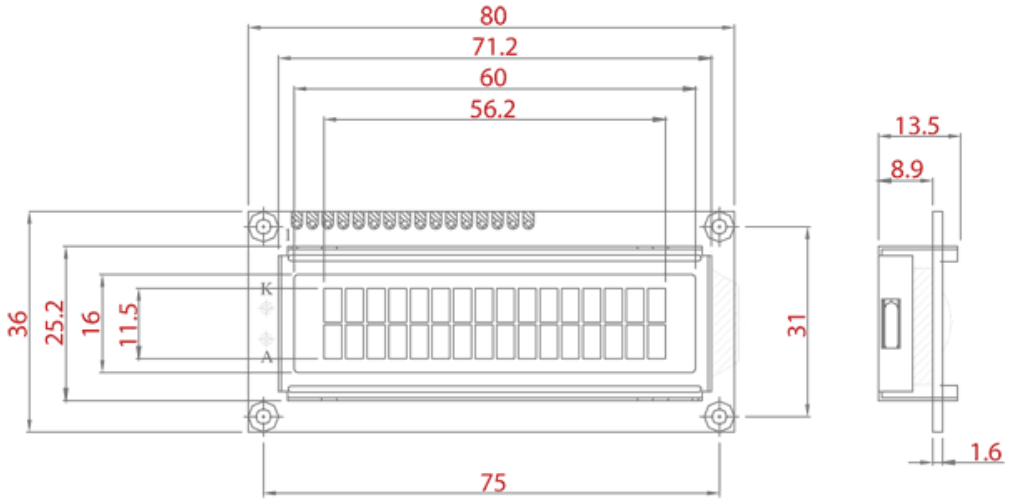
- Arduino UNO R3
- Ecran LCD 1602 I2C

Desene tehnice

- Arduino UNO R3



- Ecran LCD 1602 I2C



Export to PDF

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/daniela.fuiorea

Last update: **2024/05/27 01:24**