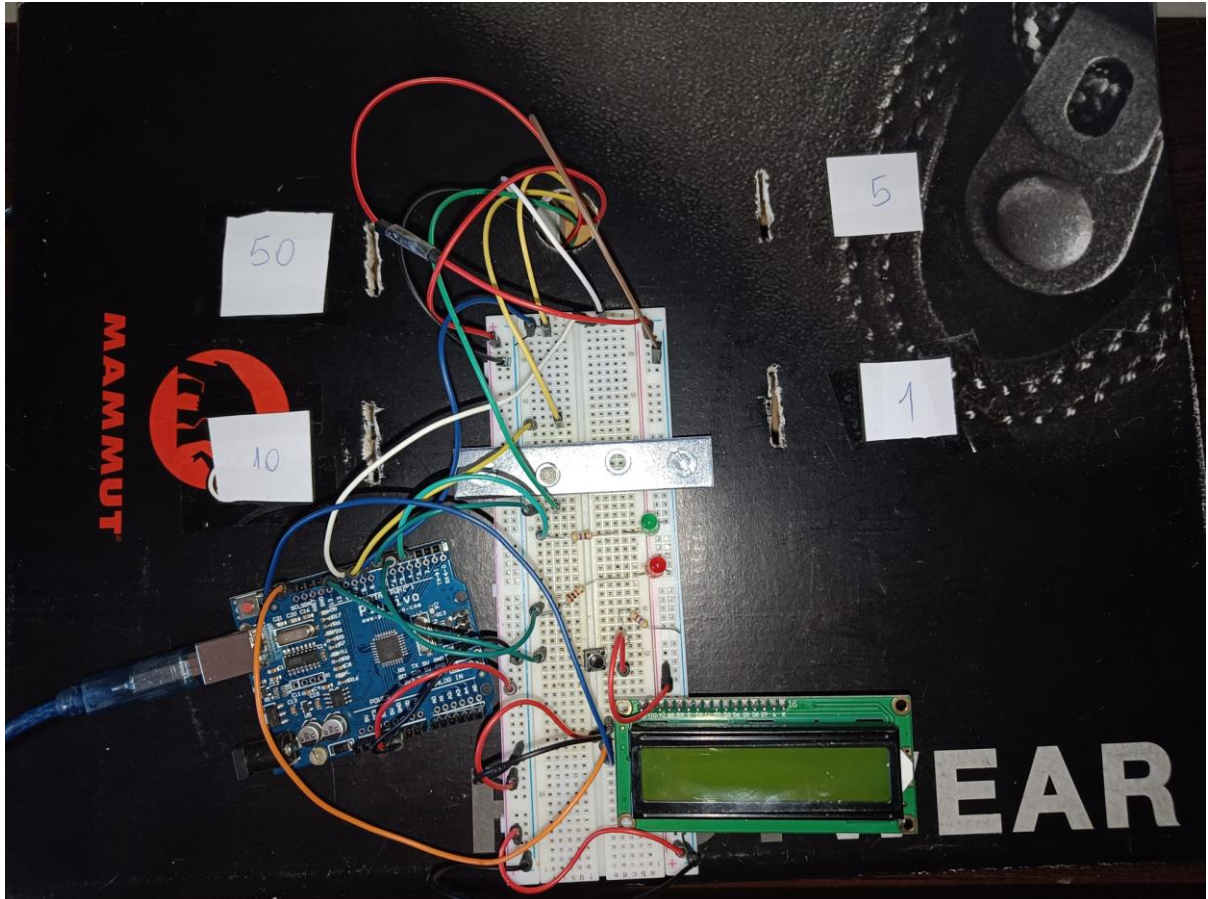


Contor monede digital

Contorul de monede digital reprezinta o inovatie practica si eficienta pentru gestionarea banilor, acesta fiind implementat pentru a simplifica procesul de numarare a monedelor.



Dispozitivul are la baza 4 senzori pentru fiecare tip de moneda(50, 10, 5, 1). In momentul in care o moneda este introdusa in cutie , pe un ecran va fi adaugata suma respectiva. Acest proces se repeata pana la atingerea unei sume maxime. Cat timp este posibila adugarea monedelor se va observa ca un led verde este aprins. Atunci cand suma maxima este atinsa, ledul verde se va opri, se va porni unul rosu, iar pe ecran apare un mesaj care indica golirea cutiilor. De asemenea, pe ecran mai apare si indicatia, ca dupa ce cutiile sunt golite, sa apasati pe buton pentru reset. Dupa apsarea butonului, ledul rosu se va opri, va porni cel verde, iar procesul de adugare al monedelor se repeat, incepand iar cu suma de la zero.

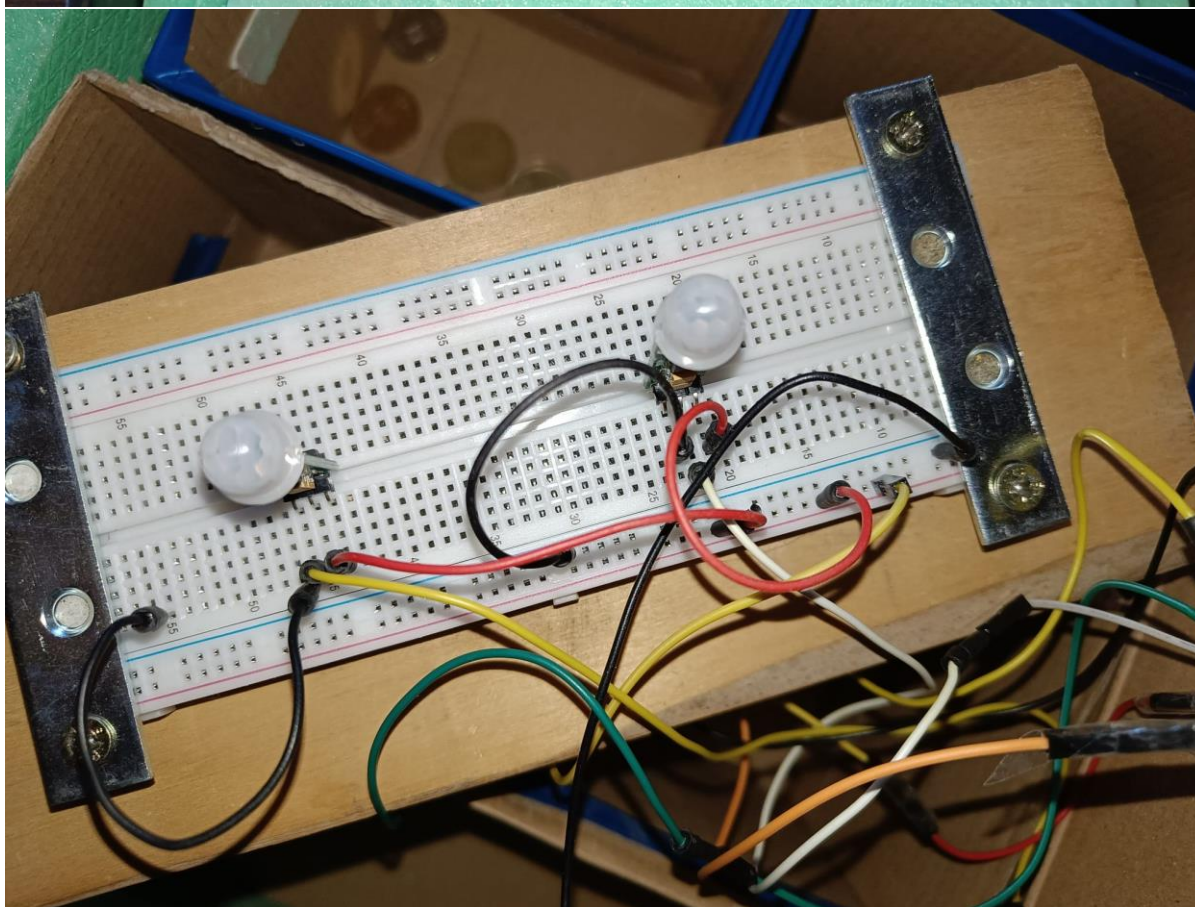
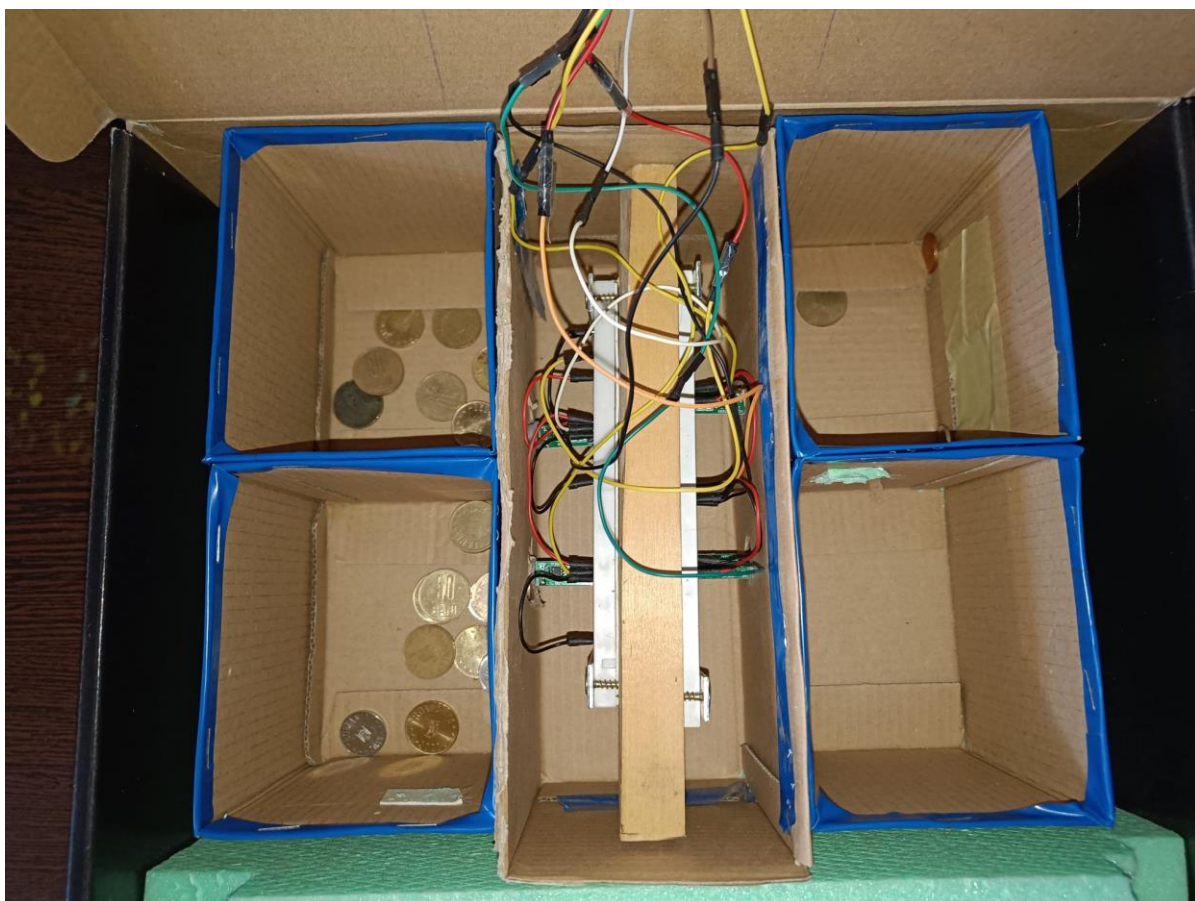
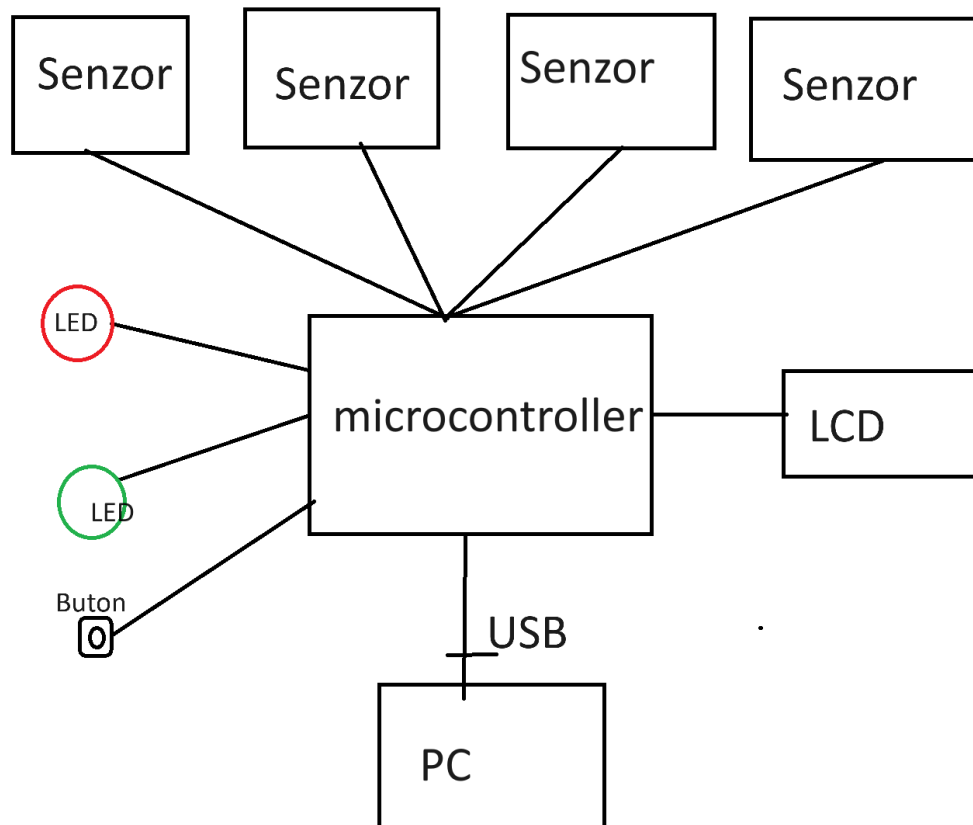


Diagrama bloc simplificata:



Modul de functionare si componentele utilizate sunt aratate in cele doua video uri postate.

Componente:

- microcontroller
- 2 leduri
- 1 buton
- 4 senzori PIR
- LCD I2C
- fire
- USB AM to BM

Pentru cod, avem doua clase, una pentru senzor(care initiaza senzorul si verifica daca este activat), iar alta pentru suma de bani(unde se fac calculele).(codul este atasat in folderul cu videouri).

```
// C++ code
//

#include <LCD_I2C.h>

LCD_I2C lcd(0x3F);
class Sum
{
    int currentSum_bani;
    int currentSum_lei;
public:
    Sum();
    void add_amount(int n);
    int getBani();
    int getLei();
    void baniToLei();
    void reset();
};

Sum::Sum()
{
    currentSum_bani=0;
    currentSum_lei=0;
}

void Sum::add_amount(int n)
{
    currentSum_bani=currentSum_bani+n;
}

int Sum::getBani()
{
    return currentSum_bani;
}

int Sum::getLei()
{
    return currentSum_lei;
}

void Sum::baniToLei()
{
    if(currentSum_bani>99)
    {
        currentSum_lei+=1;
    }
}
```

```

        currentSum_bani-=100;
    }
}

void Sum::reset()
{
    currentSum_bani=0;
    currentSum_lei=0;
}

int maxSum = 3;
class Sensor
{
protected:
    int pin;
    int add;
public:
    Sensor(int n, int a);
    void init();
    int getPin();
    bool On();
    int getAdd();
};

Sensor::Sensor(int n, int a)
{
    pin=n;
    add=a;
}

int Sensor::getPin()
{
    return pin;
}

void Sensor::init()
{
    pinMode(pin, INPUT);
}

bool Sensor::On()
{
    return digitalRead(pin)==HIGH;
}

int Sensor::getAdd()
{
    return add;
}

```

```

}

Sensor s1(6,1);
Sensor s5(7,5);
Sensor s10(8,10);
Sensor s50(9,50);
Sum s;

void setup()
{
    s1.init();
    s5.init();
    s10.init();
    s50.init();
    pinMode(5, OUTPUT);
    pinMode(10, INPUT);
    pinMode(3, OUTPUT);
    Serial.begin(9600);
    lcd.begin();
    lcd.setCursor(0,0);
    lcd.print("Suma(lei):");
    delay(500);
}

void loop()
{
    digitalWrite(5, LOW); // ledul rosu oprit
    if(s.getLei()<maxSum) // daca nu am atins suma maxima
    {
        delay(4000);
        digitalWrite(3,HIGH); //ledul verde pornit
        if(s50.On()){
            s.add_amount(s50.getAdd()); //adaug 50 daca seznorul de 50 este
activat
        }
        else if(s10.On()){
            s.add_amount(s10.getAdd()); //adaug 10 daca seznorul de 10 este activat
        }
        else if(s5.On()){
            s.add_amount(s5.getAdd()); //adaug 5 daca seznorul de 5 este activat
        }
        else if(s1.On()){
            s.add_amount(s1.getAdd()); //adaug 1 daca seznorul de 1 este activat
        }

        //afisez suma de bani cat sa corespunda cu numerotarea corecta
    }
}

```

```

s.baniToLei();
    if(s.getBani()<10){
        lcd.setCursor(14, 1);
        lcd.print("0");
        lcd.print(s.getBani());
    }else{
        lcd.setCursor(14, 1);
        lcd.print(s.getBani());
    }
    lcd.setCursor(13,1);
    lcd.print(".");
    if(s.getLei()<10){
        lcd.setCursor(12,1);
        lcd.print(s.getLei());
    } else{
        lcd.setCursor(11,1);
        lcd.print(s.getLei());
    }
    delay(2000);
}else{
    //daca am atins suma maxima
    digitalWrite(3,LOW); //opresc ledul verde
    digitalWrite(5,HIGH); //pornesc ledul rosu
    //afisez mesaje
    lcd.setCursor(0,0);
    lcd.print("Suma maxima ");
    lcd.setCursor(0,1);
    lcd.print("atinsa");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Goliti cutia");
    lcd.setCursor(0,1);
    lcd.print("de monede");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Tineti apasat pe");
    lcd.setCursor(0,1);
    lcd.print("buton pt reset");
    delay(2000);
    lcd.clear();
    //daca butonul este apasat se da reset
    if(digitalRead(10)==HIGH){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Suma(lei):" );
        s.reset();
    }
}

```

```
    delay(2000);  
}
```

```
}
```

```
}
```

```
int measureDistance(int a[]) {  
    pinMode(a[1], OUTPUT);  
    digitalWrite(a[1], LOW);  
    delayMicroseconds(2);  
    digitalWrite(a[1], HIGH);  
    delayMicroseconds(10);  
    digitalWrite(a[1], LOW);  
    pinMode(a[0], INPUT);  
    long duration = pulseIn(a[0], HIGH, 100000);  
    return duration / 29 / 2;  
}
```