

Arduino Memory Game

Ciorea-Cristescu Roxana

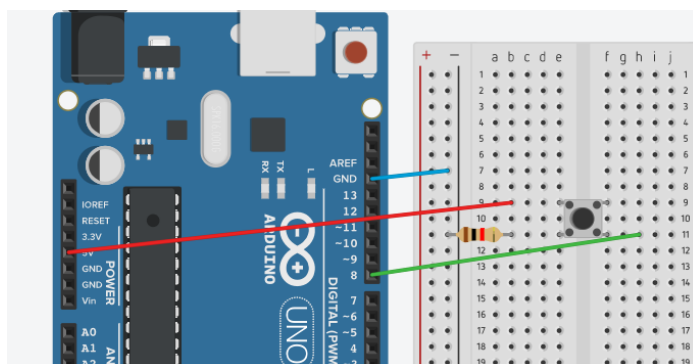
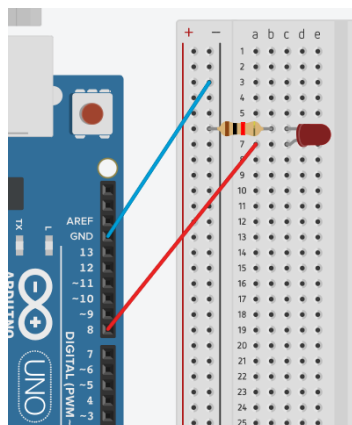
1. Cerință

Implementați un joc de memorie folosind Arduino. Jocul va conține 4 led-uri care se vor aprinde într-o ordine aleatoare conform unei secvențe, pentru a câștiga, jucătorul trebuie să apese cele 4 butoane în aceeași ordine, la fiecare rundă reușită secvența va crește cu 1. În cazul în care jucătorul pierde, secvența actuală va fi afișată pentru a putea ține scorul.

Extra: adăugarea unui buton de start.

2. Cunoștințe necesare

Singurele cunoștințe necesare pentru implementarea acestui joc sunt limbajul Arduino (`delay()`, `digitalWrite()`, `digitalRead()`) și montajele pentru led-uri și pentru butoane, prezentate mai jos:



3. Soluție

Un prim pas este declararea unor variabile:

```
byte sequence[100]; //the random sequence
byte seqLen = 0; //current lenght used of the sequence
byte inputCount = 0; //number of pressed buttons
byte expect = 0; //expected input
byte lastInput = 0;

byte nrPins = 4;
byte Buttonpins[] = {3, 4, 6, 7};
byte Ledpins[] = {8, 9, 11, 12};

bool wait = false; //check if it's the player's turn
bool resetFlag = false;
bool press = false; //check if a button was pressed
bool start = false; //check if the start button was pressed
```

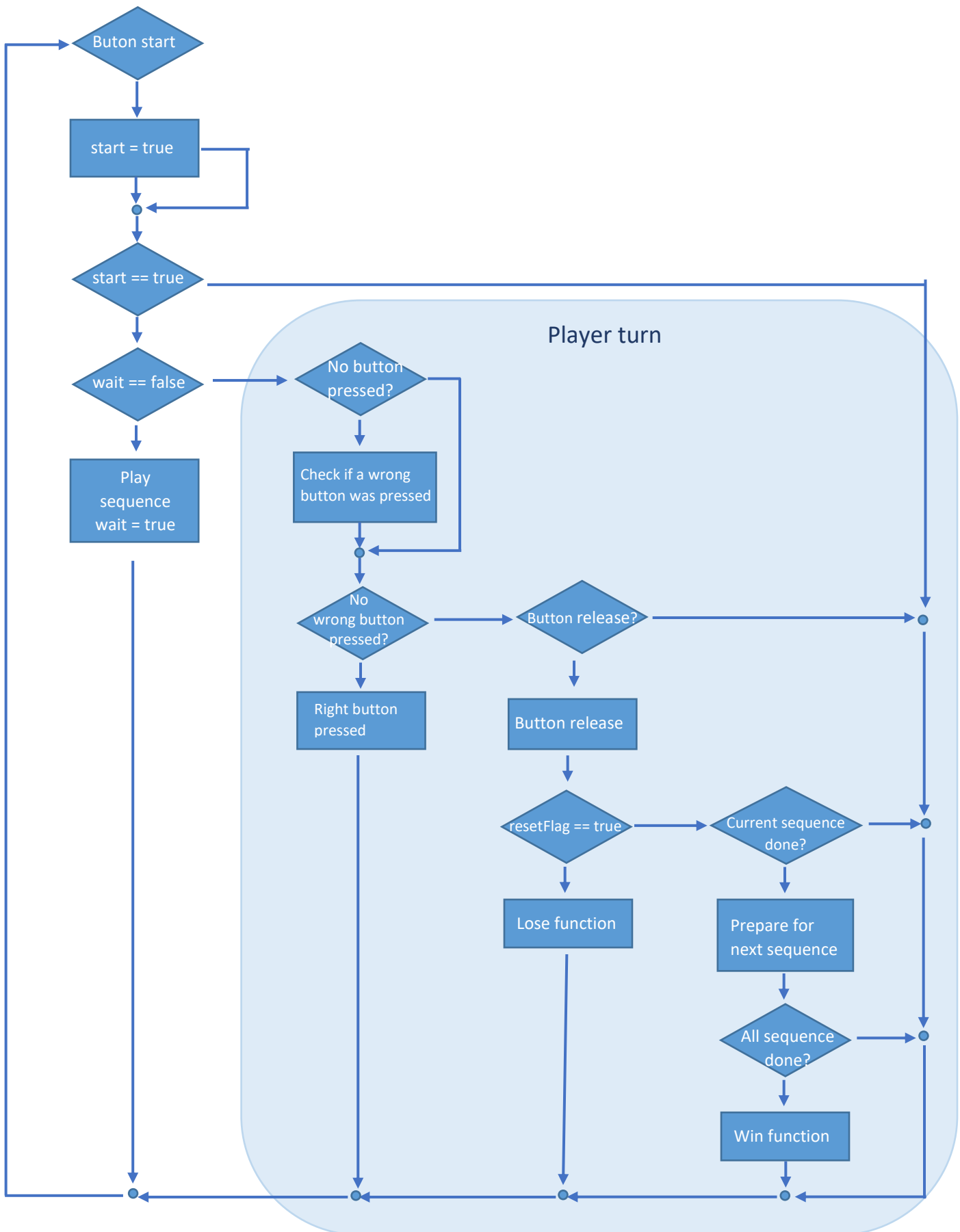
Următorul pas este definirea unor funcții care sunt utilizate in mod repetat, acestea sunt:

- `void flash(byte nr)` care face toate becurile să se aprindă simultan de nr ori
- `void Reset()` care readuce variabilele într-o stare inițială
- `void playSequence()` care afiseaza secventa aleatoare actuala
- `void Lose()` funcție de afișare când jucătorul pierde
- `void Win()` funcție de afișare când jucătorul câștigă (la secvența de 100)

În funcția `void setup()` se inițializează pini (led-urile ca OUTPUT și butoanele ca INPUT) și se apelează funcția de Reset pentru a aduce variabilele în stările adecvate începerii jocului

Cât despre funcția loop aceasta are următoarea structură de if-uri și else-uri:

```
void loop() {
    if(digitalRead(13) == HIGH){ //checking if the start button was pressed
        if(start == true){ //game can begin
            if(wait == false){ //showing the sequence
                else{ //player's turn
                    if(press == false){ //checking if a wrong button was pressed
                        if(digitalRead(Buttonpins[expect]) == HIGH && press == false){ //checking if the right button was pressed
                            else{
                                if(press == true && digitalRead(Buttonpins[lastInput]) == LOW){ //checking if the player released the button
                                    if(resetFlag == true){ //reset the game if the player pressed a wrong button
                                        else{
                                            if(inputCount == seqLen){ //the player complited the curent sequence
                                                if(seqLen == 100){ //the player finished the game
```



4. Schemă

Schematicul este foarte simplu: pe breadboard vom pune 4 led-uri și 5 butoane, conectate conform schemelor prezentate anterior:

- pentru **led**: intrarea de la placuță (pini din Ledpins[]) care merge pe anodul led-ului și o rezistență pusă în serie cu led-ul pentru siguranță, conectată apoi la GND
- pentru **buton**: conectăm primul pin la 5V, o rezistență pusă în serie la pinul alăturat pentru siguranță, conectată apoi la 0V, iar pinul paralel cu cel conectat la rezistență este conectat la placuță (pini din Buttonpins[])

Schema întregului proiect este prezentată mai jos:

