

Dunaújvárosi Egyetem Bánki Donát Technikum

Projekt feladat dokumentáció

Tartalom

| | |
|----------------------------|---|
| Az ötlet rövid leírása: | 2 |
| Mi történik a kódban? | 2 |
| Hozzávalók és költségvetés | 2 |
| Működési elv | 3 |
| Kapcsolási rajz | 5 |
| Kód példa | 5 |
| Önreflexió | 6 |

Tantárgy neve: Mikrokontrollerek

Projekt tervezője: Szabó Dávid Róbert

Projekt címe: Tic Tac Toe játék

Osztály: 12.B

Dátum: 2024.05.15.

Az ötlet rövid leírása:

Ez a kód egy egyszerű **Tic Tac Toe** (más néven X-O játék) program, amit egy Arduino alapú rendszerre írtak, OLED kijelzővel és két gombbal a játékosok közötti interakcióhoz. A kód működését részletesebben is kifejtem:

Mi történik a kódban?

A program egy **Arduino Nano** vagy hasonló mikrokontroller használatával készült, és egy OLED kijelzőt használ a játék megjelenítésére. A két gomb a **Move** és az **OK** gombok, amelyek a játékosok lépéseiit vezérlők.

1. Kijelző és gombok:

- Az OLED kijelzőn a játék állása látható, ahol a "0" üres mezőt, "1" a játékos körét (O), "2" pedig a számítógép keresztjét (X) jelzi.
- A játékosok a gombok segítségével navigálhatnak a játéktáblán, és amikor helyet választanak, az **OK** gombbal végezzenek a választásukat.

2. Játék logika:

- A **játéktábla** egy 3x3-as mátrix, amelyet egy egyetlen tömb reprezentál. minden mező értéke 0 (üres), 1 (játékos kör), vagy 2 (számítógép kereszts).
- A program egy egyszerű **AI-t** implementál, amely nem igazán "intelligens", de próbálja megakadályozni, hogy a játékos nyerjen, és ha lehetséges, próbál nyerni.
- A játék körökre oszlik: a számítógép és a játékos felváltva lépnek.

3. Játék menete:

- A kód a **playhuman** és **playcpu** függvényekben kezeli a játékos és a számítógép lépéseiit. A játékos kiválasztja a mezőt a Move gombbal, majd megerősíti az **OK** gombbal.
- A számítógép először ellenőrzi, hogy van-e olyan mező, ahol nyerhet, majd megpróbálja megakadályozni, hogy a játékos nyerjen. Ha nincs ilyen helyzet, akkor véletlenszerűen választ egy üres mezőt.
- A **checkWinner()** függvény folyamatosan ellenőrzi, hogy van-e győztes (három egyforma jel egymás mellett), és ha van, a játék véget ér, és megjelenik a győztes neve.

4. Játék állapota:

- A program állapotát a **gameStatus** és a **winner** változók tárolják. Ha a játék befejeződik (győztes vagy döntetlen), akkor az új játék kezdéséhez visszaállítják a táblát.

Hozzávalók és költségvetés

Alkatrészlista költségvetéssel:

- mikrokontroller
- gombok
- kábelek
- kijelző

További elkészítendő:

- program

További ötletek/fejlesztési lehetőség:

- megfelelő ház/doboz kiválasztása

Működési elv

1. Játék Kezdete:

- A játék indulásakor az Arduino egy üres táblát rajzol ki az OLED kijelzőn. Ez a tábla 9 mezőből áll, és minden mező kezdetben üres (0). Az OLED képernyőn a három sor és három oszlop világosan el van választva, hogy jól látható legyen, melyik mezőben van üres hely.

2. Felhasználói Interakció:

- A felhasználó a BUTTON_MOVE (Move) gombot használja a választott mező mozgatására, és az OK (OK) gombot arra, hogy megerősítse választását.
- A játék során az "?" jel (kérdőjel) villog a kijelzőn, hogy jelezze, melyik mező lesz kiválasztható, ha a felhasználó rákattint a BUTTON_OK gombra.

3. Felhasználói Lépések:

- A felhasználó az BUTTON_MOVE gombot használva mozgatja az aktív mezőt. Ha az aktuális mező már foglalt, a rendszer automatikusan a következő szabad mezőre ugrik.
- A villogó kérdőjel jelzi, hogy melyik mezőt választhatja ki a játékos.
- Amikor a felhasználó az OK gombot lenyomja, a rendszer azt az oszlopot, vagy sort jelöli meg, amelyre a játékos be akar lépni. Az oszlopot vagy sort o karakterekkel tölti ki, jelezve, hogy a felhasználó választotta.

4. CPU Lépései:

- Az egyszerű algoritmusú CPU (számítógép) ellenfelet programozták a játékba. A CPU próbálja megakadályozni a felhasználót abban, hogy nyerjen, és saját nyerő lépésekkel keres.
- A CPU először megvizsgálja, van-e olyan mező, amely a saját nyerő lépése lenne, és ha igen, azt választja.
- Ha a CPU nem tud nyerni, akkor megpróbálja megakadályozni, hogy a felhasználó nyerjen, és ha nincs ilyen lépés, véletlenszerűen választ egy szabad mezőt.

5. A Játék Ellenőrzése:

- Miután minden egyes játékos (felhasználó vagy CPU) lépett, az checkWinner funkció végigellenőrzi az összes lehetséges nyerő kombinációt (3 azonos karakter egy sorban, oszlopban vagy átlóban).
 - Ha egy játékos nyer, akkor az OLED kijelzőn megjelenik a nyertes neve (felhasználó vagy CPU).
 - Ha nincs nyertes, és az összes mező tele van, akkor döntetlene (drafter) nyújt, és ezt is megjeleníti a kijelzőn.
6. **Új Játék Indítása:**
- Miután egy játék véget ér (nyertes vagy döntetlen), a rendszer várakozik, amíg minden gomb fel nem oldódik, hogy új játékot kezdjen. A játékos újra elindíthatja a következő meccset a rendszer automatikus resetjével.

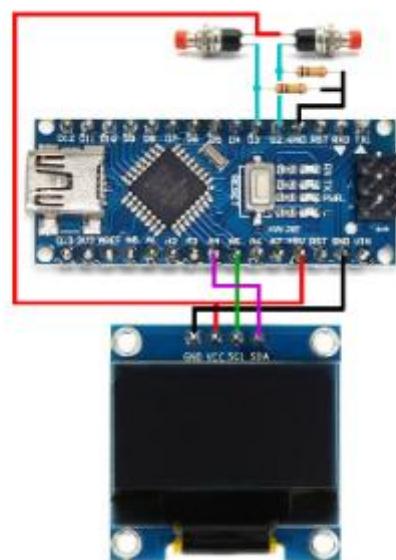
Játék Menete:

1. Az OLED kijelzőn megjelenik az üres tábla.
2. A játékos és a CPU felváltva választanak mezőket.
3. Ha a játékos vagy a CPU nyer, a képernyőn megjelenik a nyertes neve.
4. A játék újraindul, és az újabb kör kezdődik.

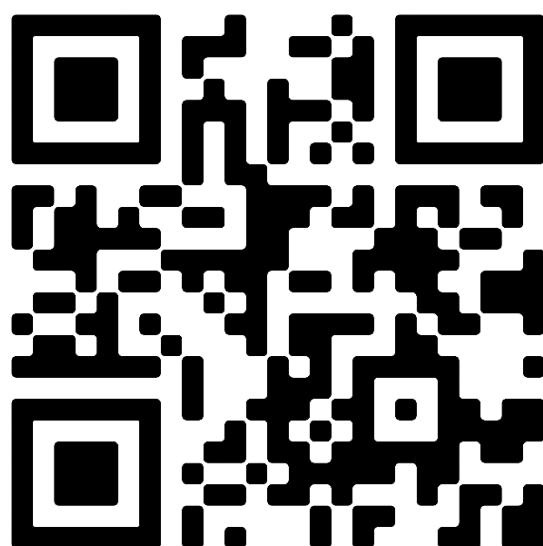
Kódban Az Alábbi Funkciók Valósítják Meg A Játékot:

- **setup():** Inicializálja a rendszert, beállítja a kijelzőt, a gombokat, és eldönti, hogy ki kezd.
- **loop():** A fő ciklus, ami figyeli a gombokat, frissíti a kijelzőt és végrehajtja a játék logikát.
- **playhuman():** A felhasználó lépéseiit kezeli.
- **playcpu():** A CPU lépéseiit kezeli.
- **checkWinner():** Ellenőrzi, hogy van-e győztes.
- **resetGame():** Újraindítja a játékot.
- **boardDrawing():** Kirajzolja a játéktáblát az OLED kijelzőn.
- **charBoard():** Visszaadja a táblán lévő karaktereket a megfelelő helyekhez ("o", vagy "x").

Kapcsolási rajz



Kód példa



Qr-kódban a kódot meg lehet tekinteni a kódot

Önreflexió

Ez a tantárgy valóban egy különleges élményt nyújtott számomra, és mélyebb megértést adott a mikrokontrollerek világába. Az elején, amikor még nem voltam teljesen tisztában a mikrokontrollerek működésével, kissé idegennek tűnt, de ahogy elkezdtem foglalkozni a programozásukkal, gyorsan ráhangolódtam a témaéra.

A tantárgy során végzett gyakorlati feladatok, például a TicTacToe játék készítése, nagyon izgalmas kihívásokat jelentettek, és segítettek abban, hogy a tanulás valós, kézzel fogható projektben valósuljon meg. Az, hogy a különböző elektronikai eszközökkel, például az OLED kijelzőkkel és gombokkal dolgozhattam, egy teljesen új dimenziót nyitott meg a számomra a programozás terén.

A mikrokontrollerek használata lehetőséget adott arra, hogy ne csak elméletben, hanem gyakorlatban is megismerjem a vezérlési logikát, a hardverek és szoftverek közötti interakciókat, és azt is, hogyan építhetünk komplexebb rendszereket egyszerűbb elemekből. A hibakeresés és a problémamegoldás készségeit is jelentősen fejlesztette a tantárgy, hiszen sokszor találkoztam váratlan helyzetekkel, de ezek csak motiváltak abban, hogy tovább dolgozzak a megoldáson.

Mindezek mellett a tantárgy segített abban, hogy rálépjek egy olyan útra, amely számomra valóban érdekes és izgalmas, és úgy érzem, hogy a mikrvezérlő programozásban rengeteg lehetőség van a jövőben való elmélyülésre. Most már teljesen más szemszögből tekintek a különböző elektronikai eszközökre, és egyre inkább látom, hogy miként lehet őket programozás segítségével működésre bírni.

Összességében, a tantárgy valóban jó választás volt számomra. Az eddig megszerzett tudás és tapasztalat egy alapot ad arra, hogy a mikrvezérlő programozásban való jártasságomat tovább mélyítsem, és új projekteken dolgozzak a jövőben. Ráhangolódtam a témaéra, és most már magabiztosabban nyúlok hozzá bármilyen hasonló feladathoz.