6. Szervomotor vezérlése és pozíciójának állítása potenciométer segítségével

Ebben a projektben egy szervomotor vezérlését fogjuk bemutatni. Az általunk használt szervomotor egy Tower Pro Micro Servo SG90 típusú motor mely megtalálható az Arduino Upgraded Learning Kit-ben. A projekt lényege, hogy egy változtatható ellenállás segítségével az Arduino Uno egy analóg bemenetén arányosan változtatjuk a jelet majd ezt a jelet egy érték illesztő eljárás segítségével fokokra alakítjuk melyeknek az értékét a szervomotor pozícióként fogja értelmezni.

Szükséges eszközök: Számítógép, Arduino IDE, Arduino Uno, egy db 2x16 karakteres LCD kijelző I2C adapterrel, egy db Tower Pro Micro Servo SG90 típusú szervomotor, egy db potenciométer, USB kábel és vezetékek.

Az előző projekttel ellentétben itt már szükségünk lesz egy új könyvtárra ahhoz, hogy a szervomotorunk mozgásra, késztessük. Ez a könyvtár nem mást, mint a *Servo.h* ezt a már említett módszerek segítségével tudjuk a programkódunkba implementálni. Ha ezt megtettük és a szükséges további könyvtárakat *Wire.h* és *LiquidCrystal_I2C.h* is beillesztettük akkor haladhatunk tovább.

```
#include<Wire.h> //Wire könvvtár beillesztése az I2C busz használatához
#include<LiguidCrystal I2C.h> //Az I2C Folvékony kristálvos LCD kijelző kezelő könvvtára
LiquidCrystal I2C lcd(0x27, 16, 2); //Az általunk használt kijelző karakterkészlete 16 karakter és 2 sor
#include<Servo.h> //A szervomotor vezérléséhez szükséges könyvtár beillesztése
Servo servo; //Servo típusú változó mely a szervomotorra mutat
int p; //A pozíciót tartalmazó változó
const int potmeter = A0; //konstans globális integer típusú változó mely a potenciométer vezetőjének pin-jét tárolja
void setup()
  lcd.init(); //Az LCD kijelző inicializálása
  lcd.backlight(); //Az LCD kijelző háttérvilágításának bekapcsolása
  lcd.clear(); //Az LCD kijelző tartalmának a törlése
  servo.attach(9); //A szervomotor csatlakoztatása a 9-es pinre
  servo.write(0); //A szervomotor 0-dik pozícióba állítása
  delay(3000); //Várakozás 3 másodpercig
  lcd.setCursor(0,0); //Kurzor pozicionálás ez esetben 0. karakter a 0. sorban
  lcd.print("-----MOTOR-1----"); //Megadott karakterlánc kiíratása
void loop() //ciklus
 lcd.setCursor(0,1); //Kurzor pozicionálás ez esetben 0. karakter a 1. sorban
lcd.print("POZICIO:"); //Megadott karakterlánc kiíratása
p = analogRead(potmeter); //A potenciométer értékének beolvasása és a p változóban tárolása
p = map (p, 0, 1023, 0, 180); //A kapott érték intervallumának motorpozícióként értelmezhető értékké alakítása
lcd.setCursor(9,1); //Kurzor pozicionálás ez esetben 9. karakter a 1. sorban
 lcd.print(p); //A változó értékének kiíratása
lcd.print(" "); //Üres karakterlánc kiíratása
 servo.write(p); //A szervomotor pozicionálása a változó értékének megfelelően
lcd.setCursor(13,1): //Kurzor pozicionálás ez esetben 13. karakter a 1. sorban
lcd.print("FOK"); //Megadott karakterlánc kiíratása
 delay(10); //Várakozás 10 mikroszekundum ideig
```

1. ábra Szervomotor vezérlés programkód (forrás: saját szerkesztés)

A könyvtárak beszúrását követően egy típus definiálást kell elvégezni, ahol, a szervomotorunkra mutató változó is létrejön, a Servo jelenti a változó típusát a servo pedig maga a szervomotorunk. Majd létrehozunk egy globális integer típusú változót melyet p-vel jelöltünk, ez a változó fogja tartalmazni a motor pozíciójának az értékét. A deklarálás szekcióban végül létrehozunk a már korábban említett okból folyóan egy konstans változót mely a potenciométer vezetőjére csatlakoztatott pin sorszámát hordozza, ez esetben ez a pin az A0 sorszámú analóg bemenet. El is érkeztünk a void setup () részünkhöz. Ebben a programban nem szeretnénk használni a soros port kommunikációs lehetőségeit, ezért a Serial.begin(9600) parancs nem kerül beillesztésre. Viszont a szükséges lcd műveleteket mindenképp elkell végeznünk. Valamint a szervomotor csatlakozását melyet a servo.attach() funkcióval tudunk véghez vinni, itt paraméterként mindössze azt kell megadnunk, hogy a szervomotorunk az Arduino melyik pinjére van csatlakoztatva. Fontos tudni, hogy a pin melyet használni szeretnénk PWM jel kibocsátására alkalmas legyen. Szerencsére ezek a pinek jelölve vannak egy "~" karakterrel. Ha a szervomotor csatlakozása megtörtént érdemes azt ún. nulladik pozícióba állítani, ezt egyszerűen a servo.write(0) parancs segítségével tudjuk elvégezni. A következő lépés egy három másodperces késleltetést követő kurzor pozicionálás és egy megadatott karakterlánc kiíratása. Ezt követően a kijelző 0-ik sorában a következő üzenet jelenik meg "-----MOTOR-1-----". Ezzel a void setup () szekciót el is hagyhatjuk és tovább léphetünk a void loop () szekcióba. Itt egy mutató pozicionálással kezdünk melynek paraméterei (0, 1) vagyis a 0-ik karakter az 1-es sorban. Itt megjelenítjük a "POZICIO:" karakterláncot, majd ezt egy értékadó művelet elvégzése követi, ahol a p változó tartalmának az A0 analóg bemeneten potenciométer által szabályozott érték felel meg. Az érték olvasását szintén az analogRead() funkció segítségével végezzük. Ahogy az érték tárolásra került meg kell határoznunk, hogy a bemenet értéke milyen motor pozíciónak felel meg. Ezt a *map* funkció segítségével a következőképpen valósítjuk meg, p = map (p, 0, 1023, 0, 180); vagyis, ha az analóg bemeneten mért érték 0 akkor a p változó értéke is 0, viszont, ha bemenet értéke 1023 akkor a p értéke 180-al lesz egyenlő. Mivel a motorunk csak 180 fokos mozgást tud végezni ezért a felső érték 180. Ha az értékillesztést véghez vittük, kiírathatjuk a kijelzőre a p változó értékét, vagyis az aktuális motorpozíciót is egyben. Előtte a kurzort az 1-es sor 9-ik karakterére helyezzük. A servo.write(p) funkciót használva pedig a motornak megadjuk azt a pozíciót melyet éppen a p változónk tárol. A kijelzőn megjelenített értékhez hozzárendeljük a "FOK" karakterláncot, és végül egy 10 milliszekundumos késleltetést is alkalmazunk, ez a potenciométer értékének olvasása és a motor pozíciójának az állítása közti késleltetést állítja. Végül bezárjuk a *void loop()* ciklust és el is készült a programkód.



2. ábra Szervomotor vezérlés bekötési rajz (forrás: saját szerkesztés)

A bekötéshez szükséges 2.ábra szerint járhatunk el a megépítésben. Nincs más dolgunk, mint csatlakoztatni a szervomotorunkat az Arduino Uno 9-es pinjéhez majd a motor tápellátását is biztosítani mely 5 volt egyenfeszültség, ezt követi a potenciométer csatlakoztatása. Ahogy előzőleg a potenciométer középső csatlakozója, vagyis a vezető az Arduino Uno A0 bemenetére csatlakozik. A maradék két csatlakozót az Arduino állandó 5 voltot biztosító pinjére és a közös földelésre (GND) kell kötnünk.

Elkészült az áramkörünk, ellenőrizzük le, hogy minden a helyére került-e és ha mindent rendben találunk csatlakoztathatjuk az eszközt a számítógépünkre és megkezdhetjük a programkód feltöltését. Ha a feltöltés sikeresen végbe ment akkor az áramkör a következőképpen működik (3.ábra).



3. ábra Szervomotor vezérlés működés közben (forrás: saját szerkesztés)