5. Folyadékszint mérése Water Sensor segítségével

Ezen projekt lényege a folyadékszint mérés. Az Arduino Upgraded Learning kit tartalmaz egy erre szánt szenzort mely tökéletesen megfelel az elvárásainknak. A feladat ez esetben a következő, készítünk egy áramkört mely az imént említett folyadékszint mérő szenzor segítségével segít megjeleníteni egy lcd kijelzőn a mérésre szánt folyadék szintjét. A folyadék szintjeit egymásba ágyazott feltétel vizsgálatokkal fogjuk vizsgálni.

Szükséges eszközök: Számítógép, Arduino IDE, Arduino Uno, egy db 2x16 karakteres LCD kijelző I2C adapterrel, egy db Water Sensor, USB kábel és vezetékek.

Ebben az esetben nem kell a szokásos I2C, valamint az LCD kijelző használatához szükséges könyvtárakon kívül egyéb könyvtárakat beillesztenünk, ezt a szenzort anélkül is tökéletesen tudjuk használni.

```
#include<Wire.h> //Wire könyvtár beillesztése az I2C busz használatához
#include<LiquidCrystal I2C.h> //Az I2C Folyékony kristályos LCD kijelző kezelő könyvtára
LiquidCrystal I2C lcd(0x27, 16, 2); //Az általunk használt kijelző karakterkészlete 16 karakter és 2 sor
int ertek = 0:
const int sensor = A0; //konstans globális integer típusú változó mely a folyadékszint méró szenzor pin-jét tárolja
void setup()
£
 lcd.init(); //Az LCD kijelző inicializálása
 lcd.backlight(); //Az LCD kijelző háttérvilágításának bekapcsolása
  lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("FOLYADEKSZINT:");
  Serial.begin(9600); //A soros porton történő kommunikáció bitrátája
void loop() //ciklus
 lcd.setCursor(0, 0);
 lcd.print("FOLYADEKSZINT:");
 ertek = analogRead(sensor); //A szenzor értékének beolvasása és változóban tárolása
  Serial.println(ertek); //A beolvasott és változóban tárolt érték kiíratása a soros portra
  if (ertek <= 100) //Feltétel vizsgálat, ha a beolvasott érték kisebb vagy egyenlő, mint 100 akkor:
  {
   Serial.println("EMPTY"); //Megadott karakterlánc kiíratása
   lcd.print(" "); //Üres karakterlánc kiíratása
   lcd.setCursor(0, 1); //Kurzor pozicionálás ez esetben 0. karakter a 1. sorban
   lcd.print("URES"); //Megadott karakterlánc kiíratása
  else if (ertek > 130 & ertek <= 250) //Feltétel vizsgálat:
  Ł
   Serial.println("LOW"): //Megadott karakterlánc kiíratása
   lcd.print(" "); //Üres karakterlánc kiíratása
   lcd.setCursor(0, 1); //Kurzor pozicionálás ez esetben 0. karakter a 1. sorban
   lcd.print("ALACSONY"); //Megadott karakterlánc kiíratása
  else if (ertek > 260 & ertek <= 310) //Feltétel vizsgálat:
    Serial.println("MEDIUM"); //Megadott karakterlánc kiíratása
   lcd.print(" "); //Üres karakterlánc kiíratása
   lcd.setCursor(0, 1): //Kurzor pozicionálás ez esetben 0, karakter a 1, sorban
   lcd.print("KOZEPES"); //Megadott karakterlánc kiíratása
  1
  else if (ertek > 315) //Feltétel vizsgálat:
    Serial.println("HIGH"); //Megadott karakterlánc kiíratása
    lcd.print(" "); //Üres karakterlánc kiíratása
    lcd.setCursor(0, 1); //Kurzor pozicionálás ez esetben 0. karakter a 1. sorban
    lcd.print("MAGAS"); //Megadott karakterlánc kiíratása
  delay(500); //A szenzor olvasásának intervalluma 0.5 másodperc
  lcd.clear(); //Az LCD kijelző tartalmának a törlése
3
```

1. ábra Folyadékszint mérés programkód (forrás: saját szerkesztés)

Ezek után a program megírását egy globális integer típusú változó deklarálásával folytathatjuk. Ez a változó a szenzortól kapott érték tárolására fog szolgálni. Majd hozzunk létre egy konstans szintén globális és szintén integer típusú változót melyben a szenzorunk által használt bemenetet tároljuk. Azért állítjuk az ilyen változókat konstansra mivel az áramkör megépítése alatt nem tervezzük ezen pinek megváltoztatását. Ha a deklaráció kész akkor következhet a program *void setup()* szekciója. Itt szokásosan az lcd kijelzővel kapcsolatos eljárásokat kell végrehajtanunk, valamint a soros porton történő kommunikáció egy bizonyos paraméterének a beállítását. Viszont,

mi szeretnénk egy megadott karakterláncot kiíratni a kijelzőnkre már ebben a szekcióban. Ez az üzenet a következő "FOLYADÉKSZINT:". Ez az üzenet azért a void setup() részben került kiíratásra mert nem szeretnénk, hogy a ciklus minden egyes ujra indulásánal megjelenjen, ezt az üzenetet nem szükséges minden alkalommal törölni és ujra kiíratni. Tovább haladva a void loop() ciklusunkba lépünk. Itt elsőként egy érték adó műveletet hozunk létre, ahol a már előre deklarált globális integer típusú érték nevezetű változónk megkapja a szenzorunk által olvasott és továbbított jel értékét. Ezen jel kiolvasását az analogRead() funkció segítségével végezhetjük el. Majd ezt az értéket megjelenítjük a soros porton is a Serial.println(ertek) funkció használatával. A következő lépés egy feltétel vizsgálat mi szerint, ha az olvasott jel kisebb vagy egyenlő mint 100 akkor, a soros portra a következő üzenetet jelenítjük meg "ÜRES". Az LCD kijelzőre pedig egy üres karakterláncot íratunk ki, ez egy módszer a kijelző egy bizonyos részén lévő tartalom törlésére, ezután a kijelző kurzorát állítsuk az 1-es sor 0-ik karakterére, majd jelenítsünk meg egy általunk megadott helyzetjelző üzenetet, mint például "ÜRES". A feltétel vizsgálatból kilépve az egyébként ágra ugrunk, ahol egy újabb feltételt hozunk létre mi szerint, ha a mért érték nagyobb, mint 100 és kisebb vagy egyenlő, mint 250 akkor, szintén egy üzenet kerül a soros portra és az lcd kijelzőre is egyaránt ez esetben ez az üzenet az "ALACSONY" lesz. Ahogy észrevehettük ebben a feltétel vizsgálatban már logikai operatárt is használtunk, ami nem mást volt, mint a logikai "ÉS" mely akkor igazolja a feltételt, ha mindkét oldali összehasonlító művelet igaz. Ez a feltétel az Arduino által használt programnyelvben így demonstrálható, *if(ertek > 130 & ertek <= 250)*. Tovább haladva a szekvencián egy újabb feltétel vizsgálat következik, ahol, ha a mért érték nagyobb, mint 260 és kisebb vagy egyenlő, mint 310 akkor a "KÖZEPES" karakterlánc kerül kiíratásra az lcd kijelzőre, valamint a soros portra is egyaránt. A feltétel vizsgálat ezen ágából kilépve eljutunk az utolsó feltételünkhöz mely szerint, ha az érték nagyobb, mint 315 akkor, a folyadék szintje elérte a magas szintet, tehát a "MAGAS" üzenet is kiíratásra kerülhet mindkét helyre. Végül, hogy a ciklus ütemét egy kicsit irányítsuk beiktatunk egy 500 milliszekundumos késleltetést majd ezt követően töröljük a kijelzőnk tartalmát, mely a ciklus újra indulását követően ismét a legfrissebb értékeket fogja megjeleníteni.



2. ábra Folyadékszint mérés bekötési rajz (forrás: saját szerkesztés)

Az áramkörünk bekötése roppant egyszerű (2.ábra) mivel a Water Sensor csak 3 csatlakozóval rendelkezik, ezért a pozitívval jelzett csatlakozóját az Arduino Uno 5 volt egyenfeszültségére az S-el vagyis Signal-lal jelzett pinjét az Arduino A0 sorszámú analóg bemenetére kötjük, és végül a GND pint pedig a közös GND pinhez kapcsoljuk.

Amennyiben figyeltünk a részletekre és mindent megfelelően csatlakoztattunk, jöhet a programkód feltöltése. S ha az Arduino fejlesztői környezete az Arduino IDE a *Done uploading* üzenetet jeleníti meg a folyamat ablakában akkor a következő működést tapasztalhatjuk.



3. ábra Folyadékszint mérése működés közben (forrás: saját szerkesztés)