

3. Hőmérséklet és páratartalom mérése és kiírása DHT11-es szenzor segítségével

A következő projekt lényege a DHT11-es hőmérséklet és páratartalom mérésére képes szenzor értékeinek a kiolvasása és megjelenítése egy I2C LCD kijelző segítségével. Természetesen itt is érdekünk, hogy a megjelenített adatok mindig a legfrissebbek legyenek ezért ennek a feltételnek megfelelően fogunk eljárni.

Szükséges eszközök: Számítógép, Arduino IDE, Arduino Uno, egy db 2x16 karakteres LCD kijelző I2C adapterrel, egy db DHT11-es szenzor, USB kábel és vezetékek.

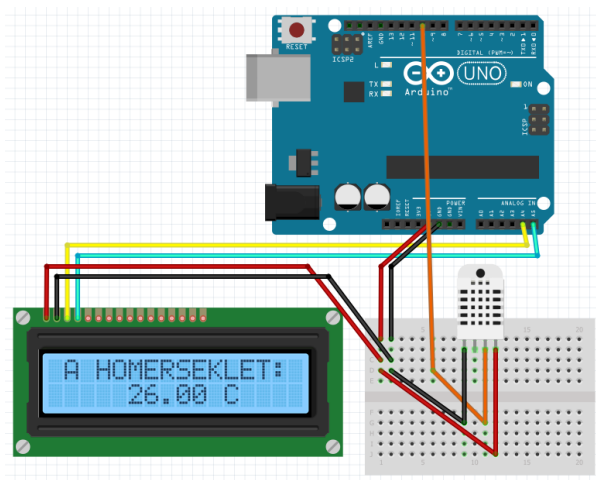
Szokáshoz híven a folyamatot a programkóddal indítjuk. Nem szabad megfeledkeznünk a könyvtárak beszállásáról. Itt már eggyel bővül a lista mivel a DHT11-es szenzor használatához egy külön könyvtárra lesz szükségünk. Ez a könyvtár a *dht.h* ennek a beszerzése a már említett módon megvalósítható. A további *Wire.h* és *LiquidCrystal_I2C.h* könyvtárakat is mellékeljük a programunkba.

```
#include <dht.h> //A DHT11-es szenzor használatához szükséges könyvtár
#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
dht DHT;
const int dht11 = 10; //konstans globális integer típusú változó mely a DHT11 szenzor pin-jét tárolja
const int led = 12; //konstans globális integer típusú változó mely a LED dióda pin-jét tárolja
void setup()
{
    lcd.init(); //Az LCD kijelző inicializálása
    lcd.backlight(); //Az LCD kijelző háttérvilágításának bekapcsolása
    pinMode(dht11, INPUT); //A DHT11 által használt Pin bemenetté alakítása
    pinMode(led, OUTPUT); //A LED diódát tartalmazó Pin kimenetté alakítása
    Serial.begin(9600); //A soros porton történő kommunikáció bitrátája
}
void loop() //ciklus
{
    delay(2000); //Várakozás 1 másodpercig
    int chk = DHT.read11(dht11); //A DHT11 értékének a kiolvasása és segédváltozóban tárolása
    Serial.print("Hőmérséklet = ");
    Serial.println(DHT.temperature); //A Hőmérséklet értékének a soros porton való megjelenítése
    lcd.setCursor(1, 0); //Kurzor pozicionálás ez esetben 1. karakter a 0. sorban
    lcd.print("A HŐMÉRSEKLET:"); //Megadott karakterlánc kiírása
    lcd.setCursor(5, 1); //Kurzor pozicionálás ez esetben 5. karakter az 1. sorban
    lcd.print(DHT.temperature); //A Hőmérséklet értékének kiírása
    lcd.setCursor(11,1); //Kurzor pozicionálás ez esetben 11. karakter az 1. sorban
    lcd.print("C"); //Megadott karakter kiírása
    delay(5000); //Várakozás 5 másodpercig
    lcd.clear(); //Az LCD kijelző tartalmának a törlése
    Serial.print("Páratartalom = ");
    Serial.println(DHT.humidity); //A Páratartalom értékének a soros porton való megjelenítése
    lcd.setCursor(1, 0); //Kurzor pozicionálás ez esetben 1. karakter a 0. sorban
    lcd.print("A PÁRATARTALOM:"); //Megadott karakterlánc kiírása
    lcd.setCursor(5, 1); //Kurzor pozicionálás ez esetben 5. karakter az 1. sorban
    lcd.print(DHT.humidity); //A Páratartalom értékének kiírása
    lcd.setCursor(11,1); //Kurzor pozicionálás ez esetben 11. karakter az 1. sorban
    lcd.print("%"); //Megadott karakter kiírása
    delay(5000); //Várakozás 5 másodpercig
}
```

1.ábra Hőmérséklet és páratartalom mérése programkód
(forrás: saját szerkesztés)

A könyvtárak és az LCD kijelző paramétereinek megadását követően szükséges egy típus definiálás a szenzorunkhoz *dht DHT*; ennek a segítségével tudunk hivatkozni a *dht.h* könyvtárban lévő bizonyos adatokra melyekre majd szükségünk lesz az értékek kiírásánál. Ha ezt megoldottuk következhet a `void setup()` szekció, itt a szokásos eljárásokat vesszük véghez, mint például az lcd inicializálása és háttérvilágítása, valamint a DHT11-es szenzor által használt 10-es pin bemenetté alakítása. Természetesen, ha szükségünk van a soros port megjelenítő felületére, vagyis, ha szeretnénk bizonyos adatokat megjeleníteni a porton akkor a kommunikáció bitrátáját is állítsuk be a megszokott 9600-as értékre. A program `void loop ()` ciklusát egy két másodperces késleltetéssel indítjuk, ez azért szükséges, hogy a szenzor valós értéket mutasson ez idő alatt az beáll a helység hőmérsékletére, valamint páratartalmára. Ha szeretnénk több időt is megadhatunk. A késleltetést követően egy értékadási művelet következik, ahol a *chk* nevű integer típusú segédváltozó kerül deklarálásra majd a DHT11-es szenzor adat pinje értékének a tárolása is végbe megy. Ezt követően a soros porton megjelenítjük a jelenlegi hőmérséklet értékét. Majd kiírjuk azt a kijelzőre is. A *DHT.temperature* parancs a *dht.h* könyvtárban tárolt *temperature* változó értékét adja vissza, ebben a változóban tárolódik a szenzortól kapott adat mely a hőmérséklet értékére mutat. A kijelzőn megjelenített értékhez hozzá rendelünk egy "C" karaktert melyet helyes pozicionálunk a `setCursor()` funkció segítségével. Ezt követően egy öt másodperces várakozás következik, ami egyben azt is jelenti, hogy az imént kiíratásra került információ az indulástól számítva öt másodpercig lesz látható, aztán a szekvenciában sorra következő parancs, amely az `lcd.clear()` törli az lcd kijelző tartalmát. Ennek a funkciónak a használata azért szükséges, hogy az előzőleg megjelenített információ ne maradjon a kijelzőn mivel, ha nem kerül törlésre és a következő kurzor pozicionálást a mutató olyan helyre mutat, ahol karakter található akkor felül írja azt viszont csak a szükséges hosszúságban. Továbbiakban a páratartalom értékének a módja következik. Elsőként az értéket a soros porton jelenítsük meg, ez segítséget tud adni abban, hogy ha nem tudjuk biztosra, hogy egy változó hány karakter nagyságú jelet ad vissza akkor először tanácsos azt megjeleníteni a soros porton és az után a kijelzőn. A páratartalmat szintén a *dht.h* könyvtárból ezúttal a *DHT.humidity* hivatkozás segítségével tudjuk kiolvasni, majd a megfelelő helyre kiírni. Ahogyan a hőmérséklet kiírásánál itt is szükséges hozzárendelnünk egy karaktert a kiíratott értékhez, ez esetben a karakter egy "%". Végül nincs más dolgunk, mint meghatározni, hogy mennyi ideig legyen látható a megjelenített információnk. A mi esetünkben ez öt másodperc

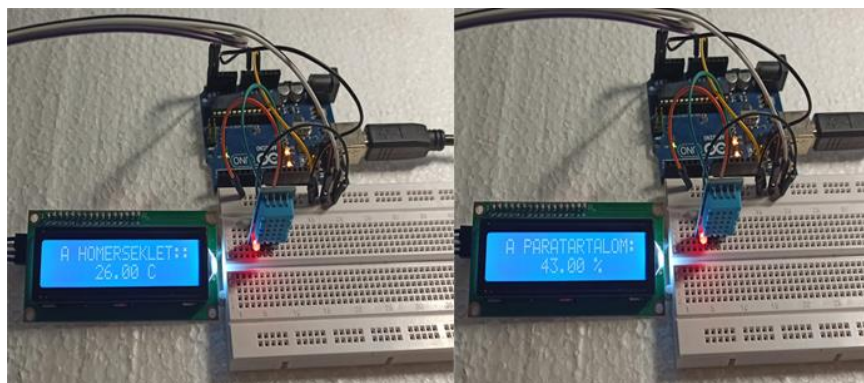
lesz. Vagyis a kijelzőnk az indítást követően öt másodpercenként frissíti a mért adatokat és a megjelenített karaktereket.



2. ábra Hőmérséklet és páratartalom mérése bekötési rajz
(forrás: saját szerkesztés)

Ami az áramkör bekötését illeti segítségünkre lesz a 2. ábra. Mely szerint az LCD kijelző bekötése nem módosul, csupán annyi a teendőnk, hogy a DHT11-es szenzort a megfelelő helye kössük. A szenzor négy csatlakozóval rendelkezik, mi csak hármat használunk. Mint tudjuk található rajta analóg és digitális kimenet is egyaránt, ahogyan már a programkódban láthattuk nekünk a szenzor digitális kimenetét kell használnunk. A másik két csatlakozóra pedig csatlakoztatjuk az állandó 5 volt feszültséget és a GND pont.

Amennyiben mindent a helyére csatlakoztattunk feltölthetjük a programkódot. Az indulást követően a kijelző tartalma a következő (3.ábra).



3. ábra Hőmérséklet és páratartalom mérése működés közben
(forrás: saját szerkesztés)