Digitális óra

Ficza Csaba

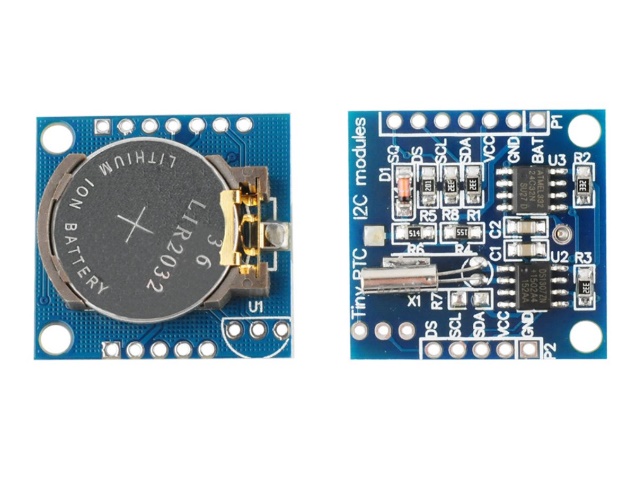
**Bevezetés:**

Az általam elkészített projekt egy digitális óraként funkcionáló olyan Arduino robot, amely az idő, a dátum, a hőmérséklet és a páratartalom folyamatosan változó értekeit tudja megjeleníteni. A projektben felhasználásra került egy Arduino Uno, egy digitális hőmérséklet és páratartalmat mérő szenzor, továbbá egy RTC (real time clock) időmérő modul, melyeknek adatait egy OLED kijelző segítségével jeleníttettünk meg.

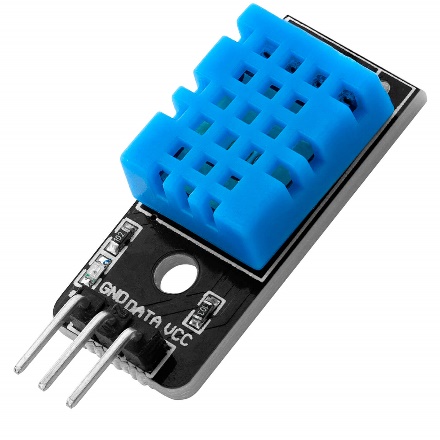
**A robot elkészítéséhez szükséges elméleti háttér:**

A projekt az Arduino Uno típusú modell segítségével készült el, ami felhasználja I2C (Inter-integrated circuit) azaz a master/slave soros kommunikáción alapuló modulokat szinkronizálni képes tulajdonságát, melyet felhasználva a board-ot egyidejűleg összekötjük az RTC modullal és az OLED kijelzővel. Ezen kívül felhasználjuk még az Arduino digitális pinjei által használt PWM (pulse width modulation) amit a hőmérsékletet és páratartalmat mérő szenzor esetében használunk.

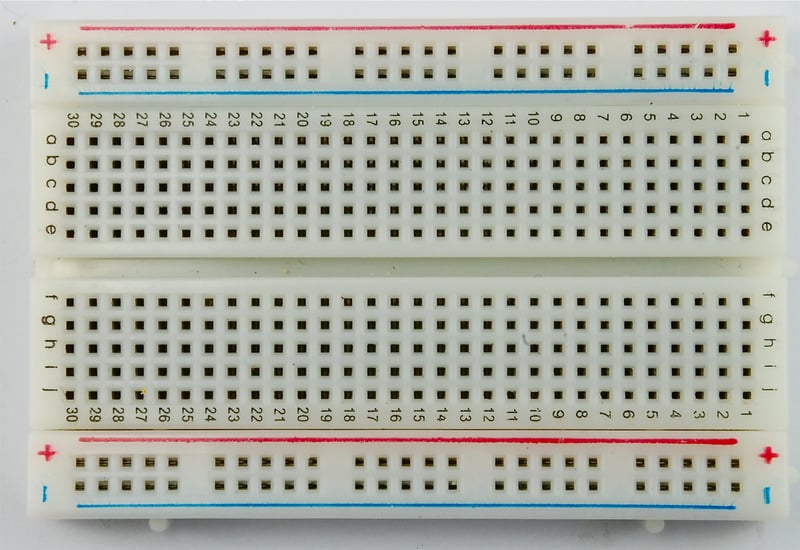
A robot felhasznál egy DS1307 nevezetű RTC (real time clock) modult is, amely egy 3 voltos CR2032 gombelem segítségével működik. Az elem biztosítja azt, hogy a modul még az Arduino kikapcsolt állapotában is számolni tudja a másodperceket, ezáltal amikor ismét bekapcsoljuk az eszközt, az idő számolódott, azaz a modul által az Arduinonak küldött idő mindig pontos lesz. A beépített idő 2000. 01. 01.-től számítódik, de ezt egy egyszerű egysoros paranccsal egyszer kell átállítani, és onnantól kezdve mindig a pontos időt mutatja.

Felhasználunk továbbá egy SH1106 1.3” méretű 132x64 pixel nagyságú OLED kijelzőt is, amelynek feladata az adatok megjelenítése lesz. A képernyőt az U8G2lib könyvtár segítségével lehet programozni.

A hőmérsékletet és páratartalmat mérő szenzorként egy DHT11 nevű digitális érzékelőt használunk fel, mely 10 másodpercenként frissíti az adatokat, és küldi azt a board-nak.



A modulokat egy ’’breadboardra’’, azaz egy próbapanelre helyezve használjuk és M-M típusú kábelekkel kötjük rá az Arduinora.



**A gyakorlati megvalósítás menete:**

Első lépésként magunk elé helyezzük az Arduino board-ot és a próbapanelt. A próbapanel felső két a többitől elkülönített különálló sorát fogjuk a modulok földelésére és áramforrására használni. A felső sor ebben az esetben a földelés szerepét tölti be, egy M-M kábelt a sor bármely részébe beleszúrunk, ezután pedig a másik a végét az Arduino egyik GND pinjébe. Ezután a különálló két sor alsó sorába egy másik kábelt teszünk, melynek másik felének vége az Arduino 5V pinjébe kerül.

Mindezek után biztositanunk kell az I2C csatlakozást. Az I2C kommunikációt 2 kábel biztosítja az SCL (serial clock pin) és a SDA (serial data pin). Az Arduino Uno nem rendelkezik külön I2C-re szánt pinekkel, ezért az A4 és az A5-ös pineket fogjuk erre a célra felhasználni. Az Arduino próbapanelek 5 oszlopos sorokból állnak melyek oszloponként össze vannak kötve, ezek közül kiválasztunk egy tetszőlegeset, és az A4-es és A5-ös panelekből kábelt húzunk valamelyik két egymás mellett lévő oszlop legfelső pinjeibe.

Ebben az esetben az RTC modul 4 pinjére lesz szükségünk. A VCC (azaz az áramforrás) kábelét a modul a próbapanelbe szúrt pinjei feletti közvetlenül első pinbe, másik végét a már fent leírt legfenti rész áramforrásként funkcionáló utolsó előtti sor bármelyik pinjébe tesszük. Ezután a GND (a földelés) pinjét kötjük be, mégpedig egy kábel egyik végét a legfelső rész egy pinjébe, a másikat pedig közvetlenül a modul feletti pinbe. A maradék két pin, azaz az SLC és az SDA pineket először a modul feletti első pinekbe kábelezzük, a kábel másik végét pedig a már bekötött két Arduino A4-es és A5-ös pinjéből származó kábel alá kötjük.

A hőmérséklet és páratartalmat mérő modul nem I2C, ezért az adatok kinyeréséhez digitális pint kell használnunk. A szenzorunknak 3 pinje van, a GND pint a már többször megemlített módon a sorba kábelezzük, azaz egy M-M kábel egyik végét a szenzor ground lába fölötti legelső pinbe tesszük, a másikat pedig a próbapanel fenti részébe. Ezután szenzor áramforrás pinje fölé teszünk kábel, aminek másik vége a már megszokott legfelső rész utolsó előtti áramforrás sorába kerül. Végül az Arduino 2-es digitális pinjéből kábelt kötünk a szenzor adat output pinje fölé.

Legutolsó lépésként az OLED képernyőt kötjük be, az RTC modulhoz nagyon hasonló módon. A GND pint hozzákötjük a próbapanel legfelső sorához, a VCC-t az legfelső utolsó előttihez, az SCL és SDA pineket azokba az oszlopokba, ahova Arduino A4-es és A5-ös pinjeiből származnak a kábelek, és az alá, ahova az RTC modulból származó I2C-s kábeleket kötöttük.

A kódolás menete:

A screen shot of a computer program

Description automatically generated

Első lépésként hozzáadjuk az általunk használni kívánt könyvtárakat, ebben az esetben a ’’Wire.h’’ könyvtárat, hogy lehetséges legyen az I2C kommunikáció, a ’’DHT.h’’ könyvtárat, hogy a DHT típusú hőmérsékletet és páratartalmat mérő szenzor könnyen olvasható legyen, az ’’RTClib.h’’ könyvtárat az RTC modul használatához, és az ’’U8x8lib.h’’ könyvtárat az OLED kijelző használatához.

Ezután a hozzáadott könyvtárakban lévő osztályokból hozunk létre konkrét változókat, mindegyik olyan típusú, mint amilyen maga a modul.

A screen shot of a computer code

Description automatically generated

A setup() függvényben először elindítjuk magát az Arduino Uno-t, utána pedig a fent deklarált változók begin() függvényét hívjuk meg, amelynek segítségével elindítjuk őket.

Az ’’rtc.adjust()’’ függvényt csak egyszer kell meghívnunk, ennek segítségével állítjuk be első használatkor az RTC modul pontos idejét, ami ezek után pontos marad.

A screen shot of a computer program

Description automatically generated

Először a delay segítségével várunk két másodpercet, hogy a DHT szenzornak legyen ideje elindulni, aztán létrehozunk egy RTC ’’DateTime’’ osztályba tartozó ’’now’’ változót, melynek segítségével az RTC műveleteket végezzük.

Két float típusú változóba lementjük a jelenlegi hőmérsékletet és páratartalmat.

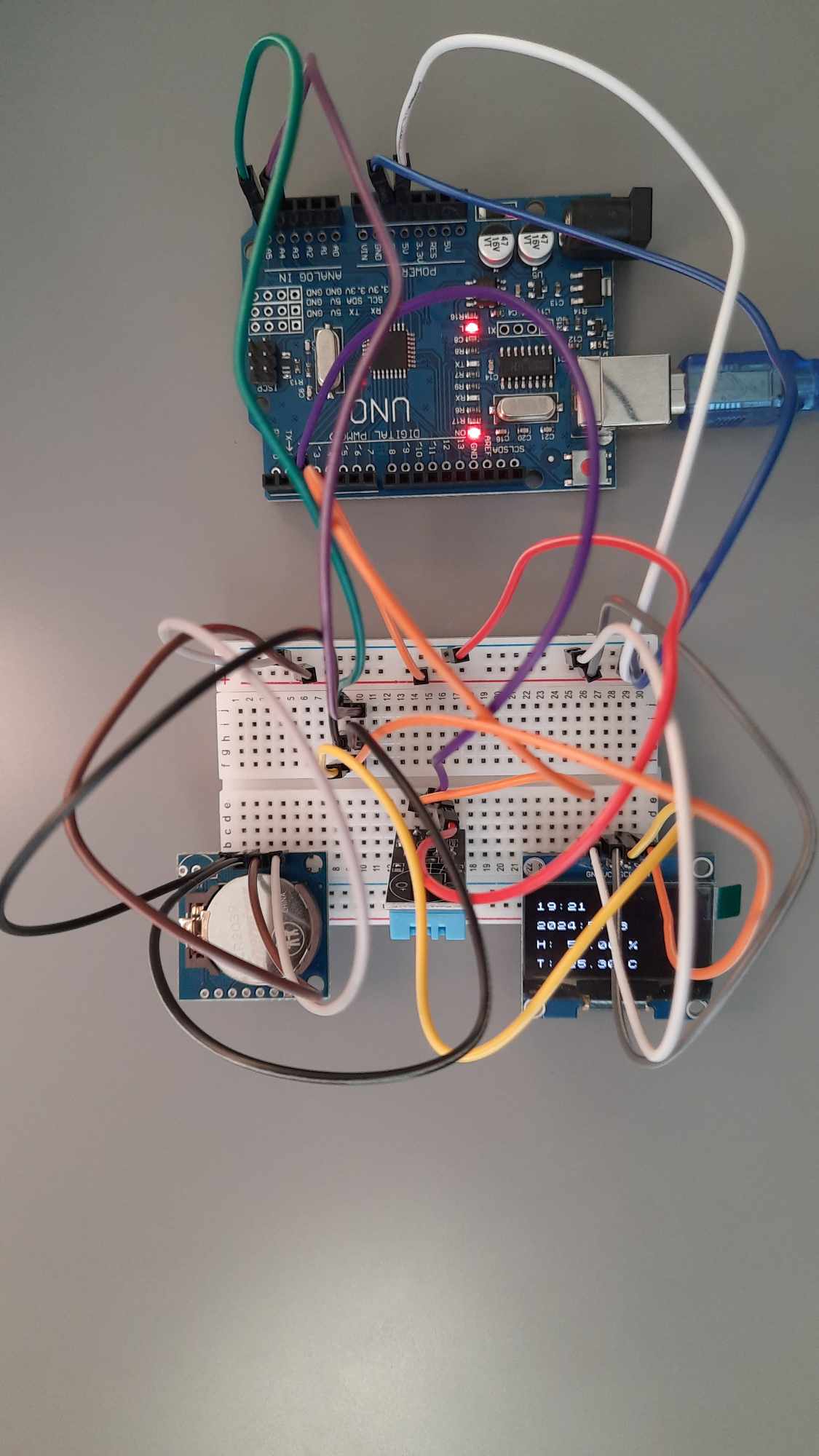
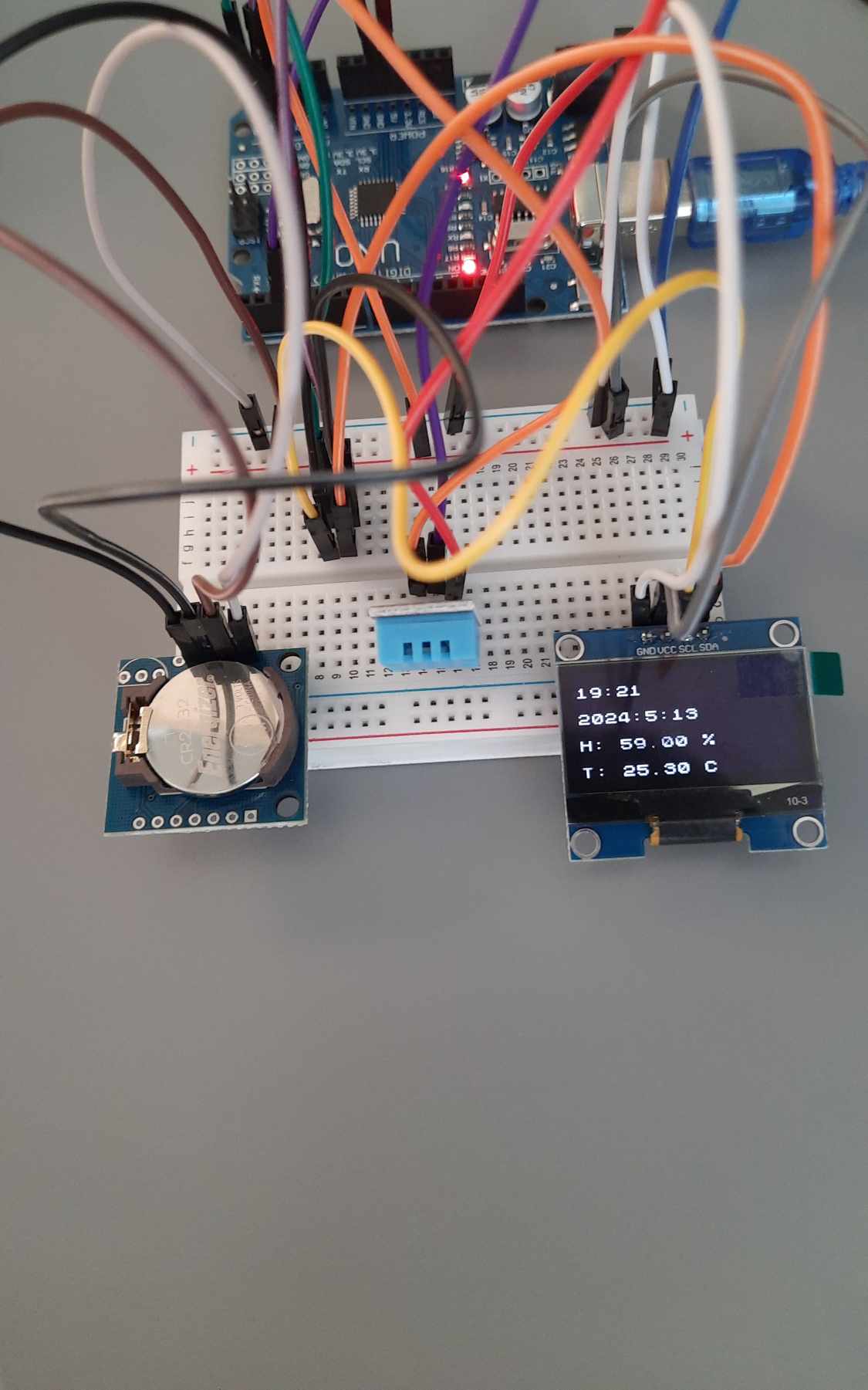
Utána az OLED kijelzővel foglalkozunk, először letörlünk a képernyőről mindent, aztán beállítjuk a betűméretet és stílust, és a kurzort az első sor elejére rakjuk. A létrehozott RTC változó és az OLED print függvényének segítségével kiírjuk az órát, majd rakunk egy kettőspontot. Az if ciklus feladata, hogy ha a percek száma egy számjegyű, rakjon mégegy nullát a szám elé.

A screen shot of a computer program

Description automatically generated

Mindezek után a következő sorba helyezzük a kurzort, és az órához hasonlóan kiírjuk az évet, hónapot, napot, kettőspontokkal elválasztva. Végül ismét áthelyezzük a kurzort és kiírjuk a már leolvasott hőmérséklet és páratartalmat a változók segítségével, és Celsius, valamint százalék jelet teszünk a végére. Ezután a program vár 10 másodpercet és még egyszer lefut a loop függvény, tehát az adatok frissítődnek.

**A működő robotról készült fénykép:**



**Használati útmutató**

A robot használata a beépített moduloknak köszönhetően rendkívül egyszerű, csupán annyi a dolgunk, hogy az Arduino Uno-t áram alá helyezzük, várjunk bár másodpercet és az OLED képernyőn már látjuk is az adatokat.

**A robot felhasználásának és továbbfejlesztésének lehetőségei:**

A robot sokféle módon felhasználható, mivel működése önálló, csak az 5 voltos áramforrásra van szüksége. A hosszabb távon való működése semmilyen számítógépes vagy Arduino IDE-s beavatkozást nem igényel. Felhasználható például szobahőmérőnek, de akár házon kívül is használhatjuk.

A robot által végzett feladatkör a több felhasznált modul miatt a kód minimális változtatásával könnyen változtatható, például átalakítható klasszikus órává, ami nem jelez hőmérsékleti adatokat, vagy átalakítható stopperórává, időzítővé.

Ha bővíteni szeretnénk akkor megtehetjük például egy memóriamodul segítségével, melynek segítségével feljegyezhetjük, hogy mikor mekkora volt a hőmérséklet, buzzer modul segítségével, hogy tudjon az általunk választott esetekben hangot is kiadni, vagy akár felhasználhatunk még több érzékelőt melynek adatait megjeleníthetjük, vagy akár Wifi modult is tehetünk rá, hogy távolról is hozzá tudjunk férni az adatokhoz.