

# Arduino UNO vezérlésű digitális óra felhasználói dokumentációja.

## Tartalom

Feladat.....	1
Futási környezet .....	1
Periféria követelmények: .....	1
Működési vázlatok, programok .....	2
Beüzemelés .....	4
Használat .....	4
Fizikai hibák .....	4
Képek és videók a kész óráról: .....	5
Szerző .....	6

## Feladat

Egy digitális óra programozása, és építése.

## Futási környezet

Arduino IDE 2.0.1 futtatására alkalmas operációs rendszer:

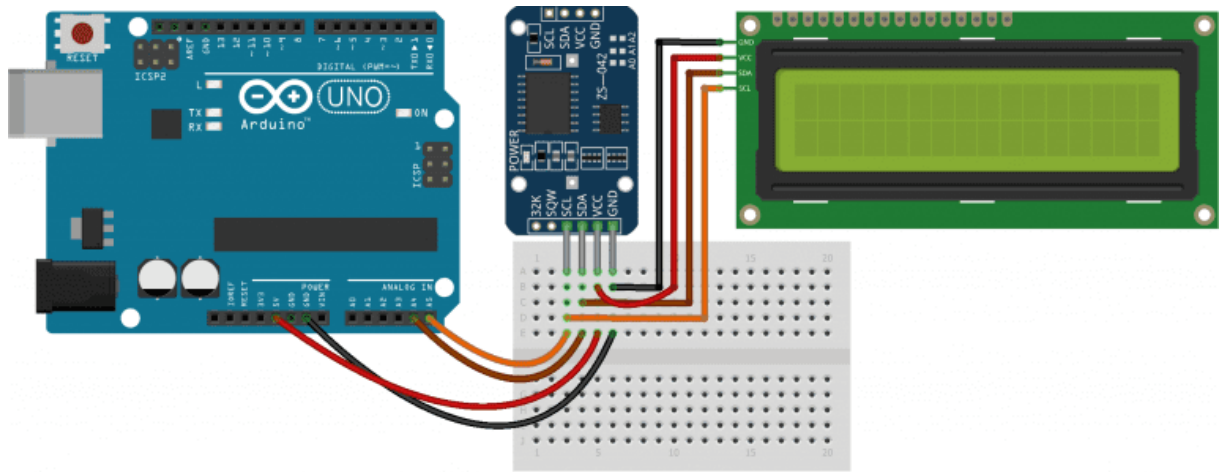
- Windows:
  - Windows 10, vagy újabb
- Linux:
  - Appliance 64 bits (X86-64)
- MacOS:
  - 10.14: "Mojave" or newer, 64 bits

## Periféria követelmények:

- Billentyűzet
- Egér
- Monitor

## Működési vázlatok, programok

Az alábbi kapcsolási rajz alapján kössük össze a modulokat:



Az alábbi kódra lesz szükségünk:

```
#include <Wire.h>           // for I2C communication
#include <LiquidCrystal_I2C.h> // for LCD
#include <RTCLib.h>          // for RTC
LiquidCrystal_I2C lcd(0x27, 16, 2); // create LCD with I2C address 0x27, 16 characters per line, 2
lines
RTC_DS3231 rtc;            // create rtc for the DS3231 RTC module, address is fixed at 0x68
/*
function to update RTC time using user input
*/
void updateRTC()
{
    lcd.clear(); // clear LCD display
    lcd.setCursor(0, 0);
    lcd.print("Szerkesztes...");
    // ask user to enter new date and time
    const char txt[6][15] = { "evet [szam]", "honapot [1~12]", "napot [1~31]",
                                "orat [0~23]", "percet [0~59]", "mp-t [0~59]"};
    String str = "";
    long newDate[6];
    while (Serial.available()) {
        Serial.read(); // clear serial buffer
    }
    for (int i = 0; i < 6; i++) {
        Serial.print("Kerek egy ");
        Serial.print(txt[i]);
        Serial.print(": ");
        while (!Serial.available()) {
            ; // wait for user input
        }
        str = Serial.readString(); // read user input
        newDate[i] = str.toInt(); // convert user input to number and save to array
        Serial.println(newDate[i]); // show user input
    }
}
```

```

// update RTC
rtc.adjust(DateTime(newDate[0], newDate[1], newDate[2], newDate[3], newDate[4],
newDate[5]));
Serial.println("RTC Frissitve!");
}
void updateLCD()
{
  /*
   create array to convert digit days to words:
   0 = Sunday | 4 = Thursday
   1 = Monday | 5 = Friday
   2 = Tuesday | 6 = Saturday
   3 = Wednesday |
  */
  const char dayInWords[7][4] = {"SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT"};
  /*
   create array to convert digit months to words:
   0 = [no use] |
   1 = January | 6 = June
   2 = February | 7 = July
   3 = March | 8 = August
   4 = April | 9 = September
   5 = May | 10 = October
   6 = June | 11 = November
   7 = July | 12 = December
  */
  const char monthInWords[13][4] = {"", "JAN", "FEB", "MAR", "APR", "MAY", "JUN",
    "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"};
  // get time and date from RTC and save in variables
  DateTime rtcTime = rtc.now();
  int ss = rtcTime.second();
  int mm = rtcTime.minute();
  int hh = rtcTime.twelveHour();
  int DD = rtcTime.dayOfTheWeek();
  int dd = rtcTime.day();
  int MM = rtcTime.month();
  int yyyy = rtcTime.year();
  // move LCD cursor to upper-left position
  lcd.setCursor(0, 0);
  // print date in dd-MMM-yyyy format and day of week
  if (dd < 10) lcd.print("0"); // add preceeding '0' if number is less than 10
  lcd.print(dd);
  lcd.print("-");
  lcd.print(monthInWords[MM]);
  lcd.print("-");
  lcd.print(yyyy);
  lcd.print(" ");
  lcd.print(dayInWords[DD]);
  // move LCD cursor to lower-left position
  lcd.setCursor(0, 1);
  // print time in 12H format
  if (hh < 10) lcd.print("0");

```

```

    lcd.print(hh);
    lcd.print(':');
    if (mm < 10) lcd.print("0");
    lcd.print(mm);
    lcd.print(':');
    if (ss < 10) lcd.print("0");
    lcd.print(ss);
    if (rtcTime.isPM()) lcd.print(" PM"); // print AM/PM indication
    else lcd.print(" AM");
}
void setup()
{
    Serial.begin(9600); // initialize serial
    lcd.init();        // initialize lcd
    lcd.backlight();    // switch-on lcd backlight
    rtc.begin();        // initialize rtc
}
void loop()
{
    updateLCD(); // update LCD text
    if (Serial.available()) {
        char input = Serial.read();
        if (input == 'u') updateRTC(); // update RTC time
    }
}

```

## Beüzemelés

A fellebb mellékelt kapcsolódási ábra alapján illesszük össze a modulokat. Letöltjük az Arduino IDE 2.0.1-t. Bedugjuk az eszközbe. Ha az Arduino UNO-n lévő "ON" lámpa, az RTC modulon a "POWER" lámpa és az LCD kijelző világít, akkor helyesen kötöttük össze, jók az érintkezések, áram alatt van a készülék. A kódot illesszük be a programba, majd a Serial Monitor ablakot használjuk

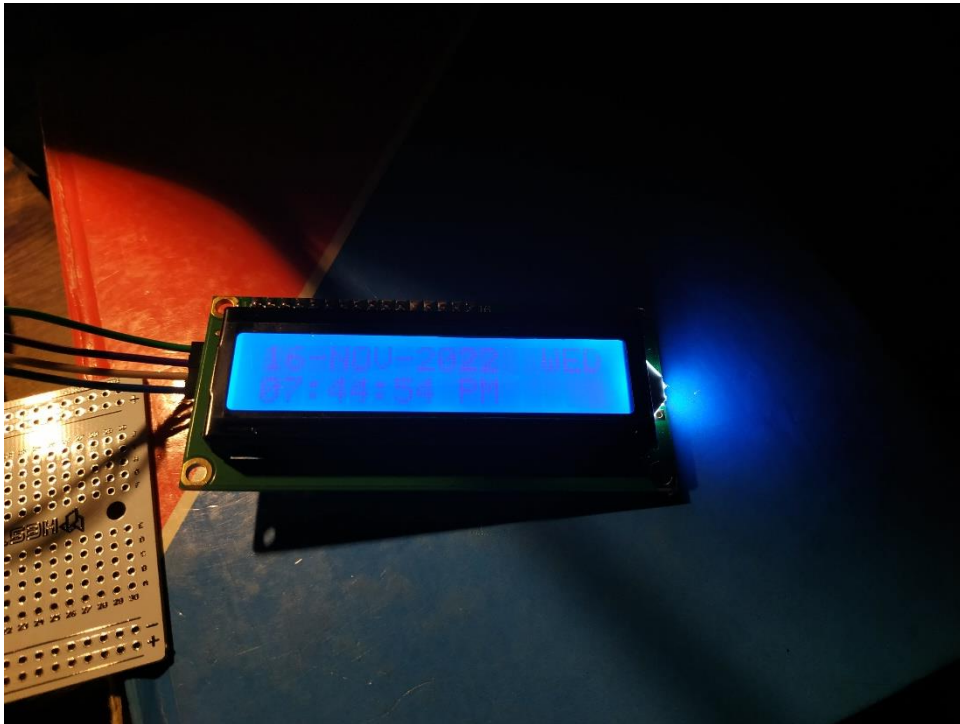
## Használat

A Serial Monitor-on lévő mezőbe írunk egy "u" karaktert. Ezután a program kéri, hogy írjuk be az évet, a hónapot, a napot, az órát, a percet és a másodpercet.

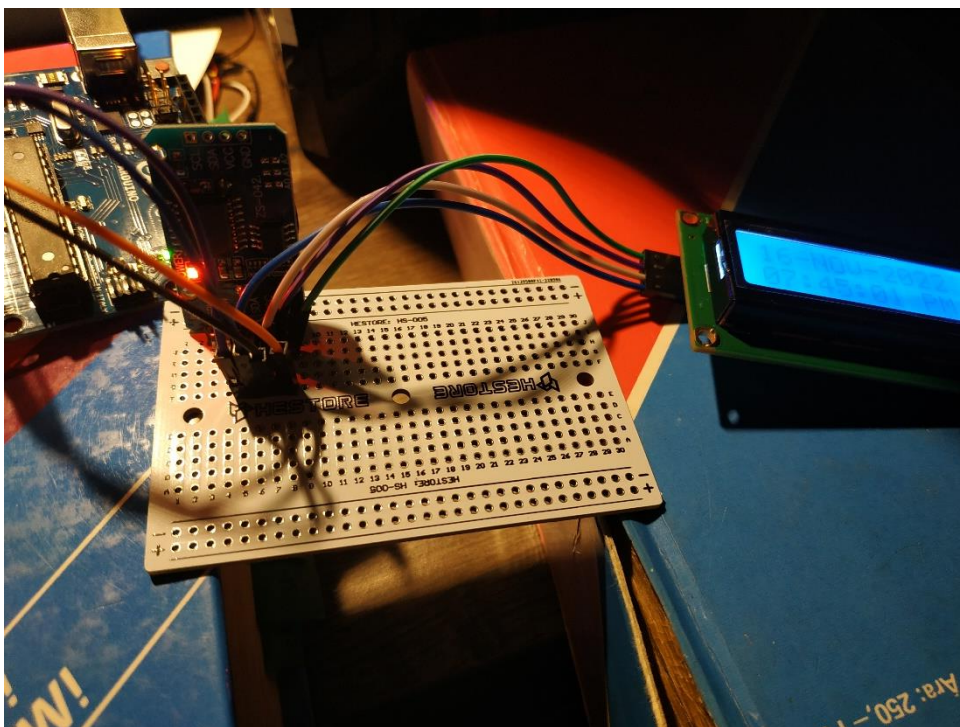
## Fizikai hibák

- Készülék rosszul lett csatlakoztatva a számítógéphez.
- Rosszul lett a kötés létrehozva.
- Kicsúszott egy kábel a nyákból

Képek a kész óráról:  
Az LCD kijelző

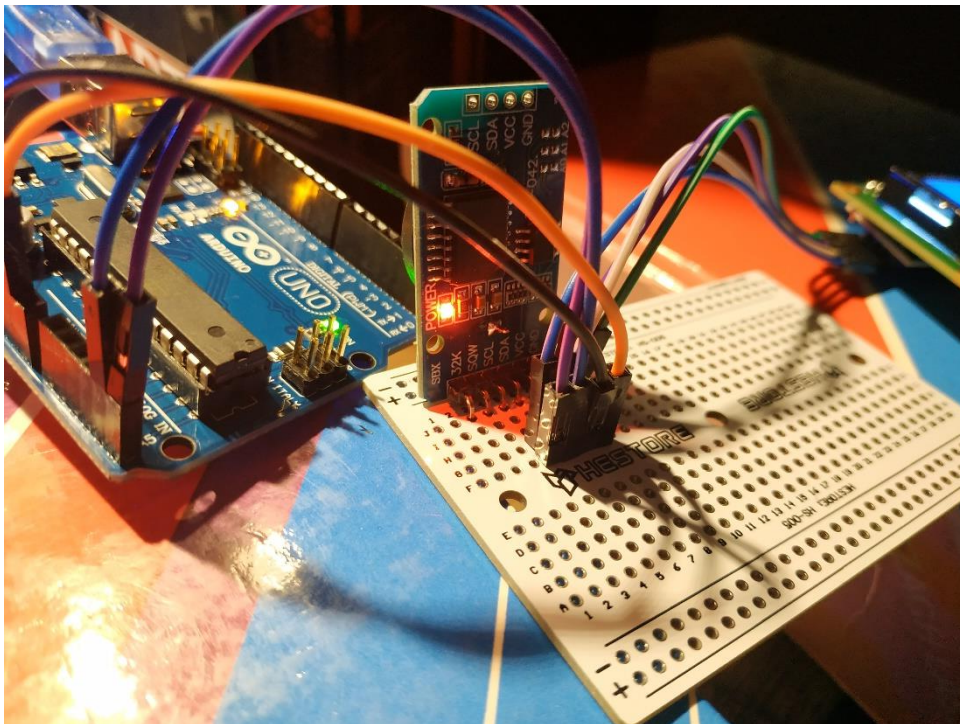


A kapcsolás:

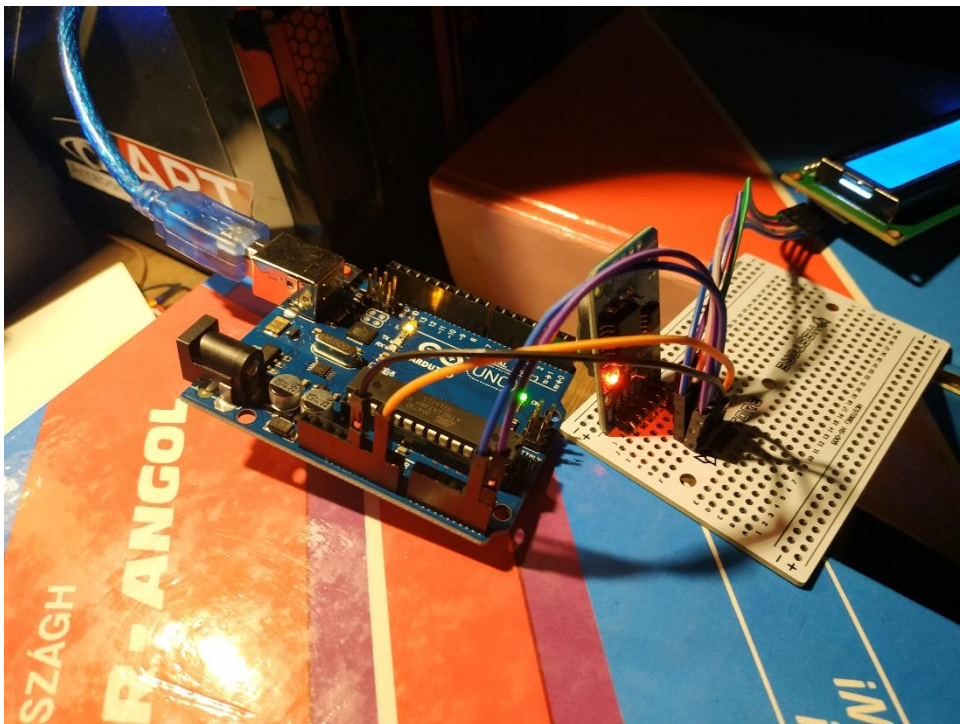




RTC modul:



Arduino UNO:



Szerző

Név: Játékos Ádám Csaba

Neptun-kód: DLKHKIJ

Email: jatekadi@gmail.com

Tantárgy kódja: GKNB\_INTM020

Tantárgy neve: Mikroelektromechanikai rendszerek