

Az Arduino fejlesztőrendszer

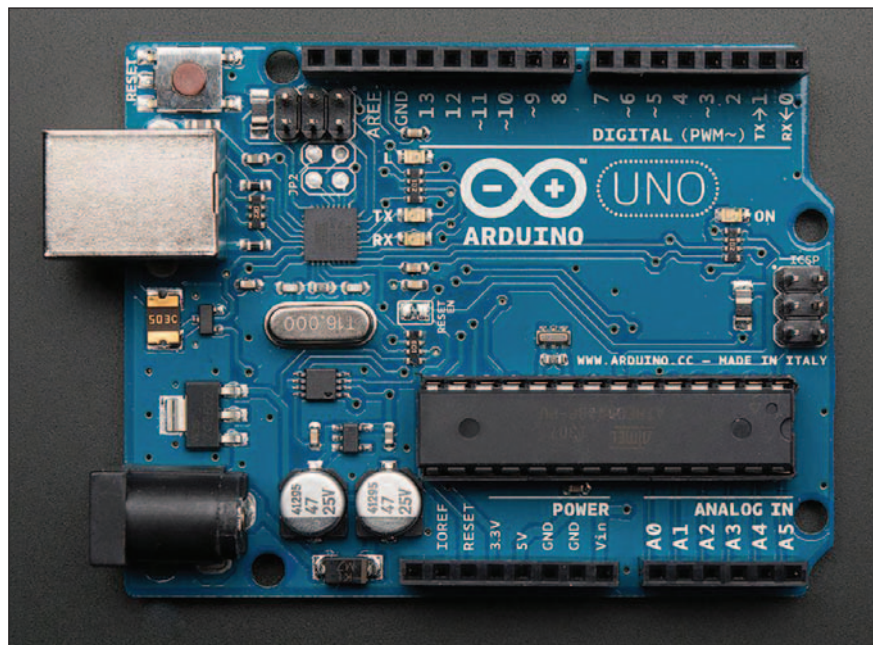
Kovács Tamás okl. villamos üzemmérnök, ERICSSON Hungary, kovtam112@gmail.com

Gyorsuló világunkban egy adott projektben a hardver- és szoftverfejlesztésre szánt időt is minimalizálnunk kell, hogy lépést tudjunk tartani az egyre dinamikusabban fejlődő technológiával, illetve, hogy lerövidítsük az ötlettől a konkrét megvalósításig ráfordított időt. Gyors és látványos eredményeket érhetünk el az Arduino mikrokontrolleres fejlesztőrendszer használatával a rádióamatőr, ill. hobby alkalmazásokban egyaránt. Az Arduino valójában egy egész fejlesztőrendszer-család, ami a megoldandó feladat típusához kínál különböző felszereltségű paneleket. Általában a több perifériát és ezzel együtt több I-O portot tartalmazó panelekben ezzel arányosan nagyobb teljesítményű mikrokontroller is került beépítésre. Ez alkalommal az Arduino-család legkisebb tagját fogjuk részletesebben megismerni, az Arduino UNO-t.

Ez a panel az 1. ábrán látható. Kicsi ugyan, de minden fontos elemet tartalmaz ahhoz, amivel a legtöbb amatörgyakorlatban előforduló mikrokontrolleres feladat megoldható. Nézzük át mit is tartalmaz a panel! Az áramkör lelke egy AT328 (Atmel) 8 bites mikrokontroller, ami egy egyszerűen, de meglepően jól felépített hardver környezetbe van beépítve, ami szinte azonnali használatra kész. A panelen 14 digitális I-O portot találunk (Digital 0-13), valamint 6 (A0-A5) analóg ki-bemenetet. A panel támogatja a legfontosabb kommunikációs perifériákat is, úgymint USART, I²C

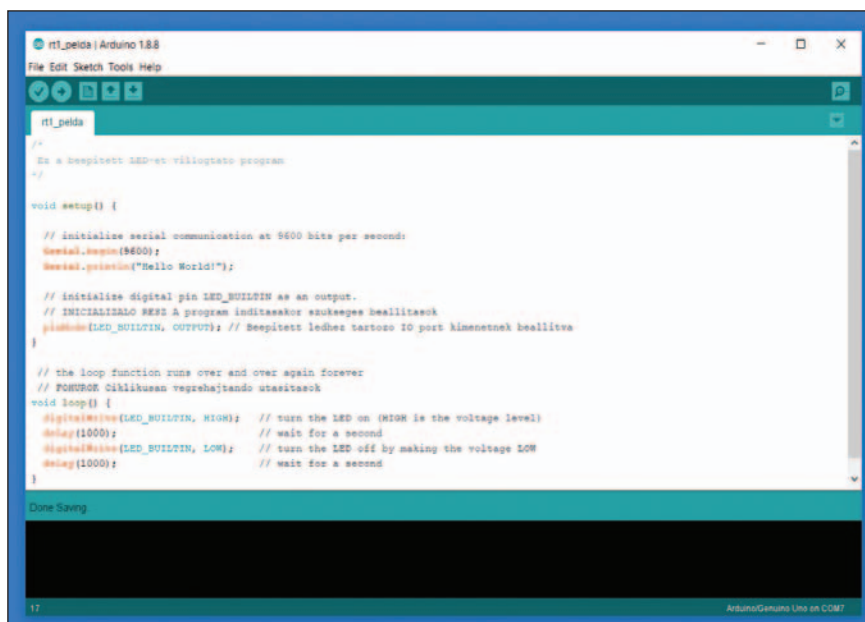
és SPI. A mikrokontroller (továbbiakban: MCU Micro Controller Unit) az előbb felsorolt perifériáknak, a hatékony és gyors működés érdekében ún. hardveres támogatást is biztosít, ezért ezeket csak a megfelelő, az adott perifériához rendelt I-O lábakon keresztül tudjuk használni. Hogy melyik periféria melyik I-O lábhoz van hozzárendelve azt a dokumentáció tartalmazza, de majd itt is kitérünk a gyakorlati példánknál erre. Egyelőre itt elég annyit megjegyezni, hogy a USART port, ami egy TTL szintű, soros interfész (RS232) a 0 és az 1 es I-O lábokat használja RXD, TXD (1.

ábra). A soros interfészt egy beépített USB konverteren keresztül használhatjuk hibakeresésre és programunk monitorozására is. Ebből látható, hogy a soros interfész egyidejűleg csatlakozik az USB konverter IC-re és a kimeneti csatlakozóra. Amennyiben olyan perifériát használunk, amely a TTL szintű soros portra (USART) csatlakozik, ez csak megosztva használható az USB-re csatlakoztatott, rendszerint számítógéppel. Ilyen eset lehet például USART-on csatlakoztatott GSM mobilhálózati modem. Az Arduino panel kettős tápellátással rendelkezik, megtáplálható az USB porton csatlakoztatott számítógépről, illetve egy 7...15 V-os dugasztápról is. A tápfeszültség meglétét az ON feliratú LED is jelzi. A panelen az MCU I-O lábai egy raszteres normál hüvelysorra vannak kivezetve. Ugyancsak kivezetésre került 5 V-os és 3,3 V-os tápfeszültség. Figyelem, itt fontos megjegyezni, hogy az MCU 5 V-os TTL szintekkel működik, ezért, ha pl. 3,3 V-os eszközt csatlakoztatunk egy kimenetére akkor szintillesztő használatára van szükség, különben a csatlakoztatott periféria-eszköz károsodhat! Az MCU 13-as I-O lábára az L jelű LED van kötve, így azt rögtön felhasználhatjuk egyszerű indikálásra programunkban. Első példalkalmazásunkban mi is ezt fogjuk majd használni. A panelen találunk még két, 6 tagú túske-



1. ábra

sorból álló csatlakozási pontot (ICSP1, ICSP2). Az ICSP1 jelű szolgálat az MCU ún. közvetlen programfeltöltésére, ehhez bizonyos célhardverre is szükség van. Alapesetben nem szükséges használni, de a szerepéről néhány szót érdemes szólni. Mint látni fogjuk az Arduino fejlesztőrendszer, nagyon komfortos munkakörnyezetet biztosít a fejlesztő számára. Ez lényegében azt jelenti, hogy a fejlesztő rácsatlakoztatja a panelt a számítógépe USB csatlakozójára, utána megnyitja az Arduino IDE (Integrated Development Environment, Integrált Fejlesztői Környezet) szoftvert és utána magasszintű programnyelven írt programját egy kattintással feltölti a MCU-ba, és azonnal ki is próbálhatja, ill. szükség esetén a támogatott hibakeresést is elvégezheti. Ahhoz, hogy a szoftver feltöltése az USB csatlakozón keresztül megvalósítható legyen, az MCU-t erre elő kell készíteni, mivel alapesetben ez csak az előzőleg említett ICSP csatlakozón keresztül volna lehetséges, eléggé körülményes módon, külön célhardver segítségével és viszonylag csak hosszabb idő alatt. Az MCU-ba először egy ún. programbetöltést segítő segédprogramot (bootloader) töltenek be, ezt természetesen csak az ICSP csatlakozón keresztül lehet megtenni, ami tartósan az MCU egy memóriaszegmensében marad, és lehetővé teszi számunkra az USB-n, illetve az ehhez tartozó soros interfészen keresztül történő kényelmes későbbi programozást. Ezt a segédprogramot a fejlesztőpanelben levő MCU-ba gyárilag betöltik, így ezzel azonnal elkezdhetjük a munkát. Az Arduino hardver tartalmaz még egy ICSP2 jelű csatlakozót is. Erre azért van szükség, mert a panelen a soros USB konvertert egy másik MCU-val oldották meg, amit ezen keresztül lehet felprogramozni. Ezt is gyárilag előre felprogramozzák, alapesetben nekünk nincs teendők vele. Végül még találunk egy RESET gombot, amivel az



2. ábra

aktuális program futását indíthatjuk újra. Ezzel a fejlesztőpanel hardver-áttekintését befejeztük, itt még összefoglaljuk a főbb paramétereket.

Arduino UNO mikrokontrolleres fejlesztőpanel hardverjellemzők:

- Beépített mikrokontroller típusa: ATMEGA 328P,
- MCU paraméterek: 8 bites, Flash memory: 32 kB (0,5 kB-ot a bootloader foglal), SRAM: 2 kB, EEPROM: 1 kB, Órajel: 16 MHz,
- Tápfeszültség: Dugasztáp 7...15 V DC, USB porton keresztül 5 V,
- Támogatott periféria protokollok: USART, I²C, SPI,
- Debugger csatlakozás típusa: USB 2.0,
- I-O portok száma: 14 digitális, 6 analóg/digitális,
- I-O portok logikai (TTL) feszültség szintjei: 0...5 V

Az Arduino Integrált fejlesztő környezet (Arduino IDE /Integrated Development Environment/)

Az Arduino fejlesztőrendszer használatát két példaalkalmazáson keresztül fogjuk megismerni.

Az első feladatban mindössze arra vállalkozunk, hogy a pane-

len található L jelű LED-et villogtassuk, illetve kiírassuk a terminálra a már híressé vált „Hello World!” mondatot. Első lépésként töltsük le az IDE szoftvert a Arduino hivatalos weboldaláról: www.arduino.cc/en/Main/Software

A letöltött `Arduino_1_8_8_windows.zip` file-t csomagoljuk ki egy adott könyvtárba, majd az `arduino.exe` programmal el is tudjuk indítani az alkalmazást. Ezt követően csatlakoztassuk az Arduino panelt a számítógép USB csatlakozójára! Optimális esetben a fejlesztőrendszer felismeri automatikusan a csatlakoztatott hardvert. Ez a fejlesztőrendszer fő ablakának jobb alsó sarkában megjelenő feliratból is látszik. Pl. Arduino/Genuino Uno on COM7 (2. ábra). Ha az automatikus csatlakozás nem sikerülne, kézzel is beállíthatjuk. Ehhez elsőként meg kell keresni, melyik virtuális soros portra csatlakozik a panel. Ezt a Windows eszközközkezelő menüjében tudjuk megnézni. Pl. COM7. Az Arduino IDE Tools főmenü, Port almenüjében tudjuk beállítani manuálisan. A Tools főmenü, Board almenüjében pedig ellenőrizzük, az Arduino/Genuino Uno van-e kijelölve, ha nem akkor ezt is állítsuk be kézzel!