

Az első Arduino-sketch

Most előtted kell lennie egy Arduino Unónak, egy USB-kábelnek és egy számítógépnek, amelyen a választott operációs rendszered (Windows, macOS, Linux, vagy a webes szerkesztő) fut. A következő szakaszban arról lesz szó, hogy mit csinálhatsz ezzel a kis eszközzel.

Keressük meg a Blink sketch-et

Annak ellenőrzésére, hogy az Arduino-szoftver és a hardver megfelelően együttműködik-e, fel kell töltened egy sketch-et. (A továbbiakban a sketch, program és vázlat neveket felváltva azonos értelemben használjuk, ahogy ezt korábban már jeleztük). Az Arduino világában a *vázlat*, azaz a *sketch* egy olyan rövid program, amely az adott ötlet vagy funkció kipróbálására szolgál. Az Arduinót olyan eszközként hozták létre, amellyel a felhasználók gyorsan tesztelhetik elképzeléseiket, ötleteiket és mintákat készíthetnek ötleteikről. A „vázlat” elnevezés arra utal, hogy ezek a programok ugyanúgy egy ötlet gyors megfogalmazásai, mint amikor valaki papíron rajzol, vagy ír egy vázlatot.

Innen ered az Arduinóra írt programok elnevezése: vázlat, sketch. Bár a kiindulópont a gyors mintakészítés volt, az Arduino-eszközöket egyre összetettebb műveletekhez használják. Szóval a vázlat, sketch elnevezés alapján ne gondold, hogy az Arduino-programok maguktól értetődőek lennének.

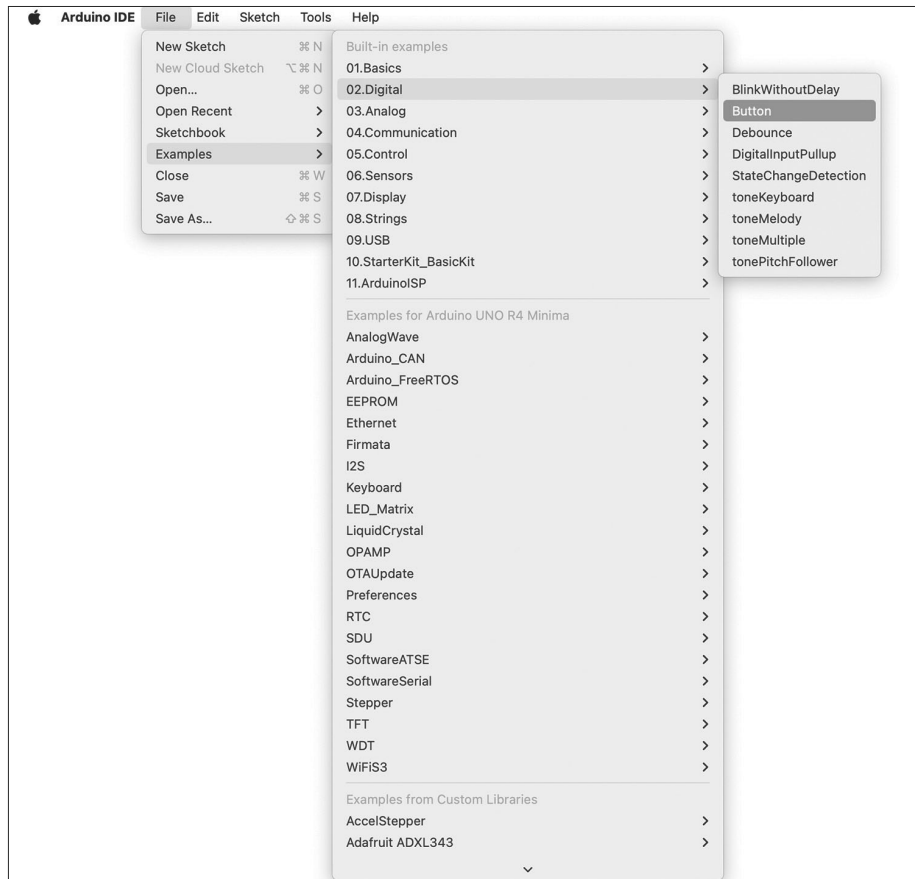
A konkrét sketch, amelyet itt használni szeretnénk, a Blink (Villogás) nevet viseli. Ez a legegyszerűbb sketch, amit csak megírhatsz, egyfajta „Helló, világ!” az Arduinóhoz. Kattints az Arduino-szoftver ablakába. A menüsorban válaszd a File ⇨ Examples (Példák) ⇨ 01.Basics (01.Alapok) ⇨ Blink (Villogás) lehetőséget (lásd a 3.1. ábrát).

Egy új ablak nyílik meg az üres eod előtt, amely a 3.2. ábrához hasonló.

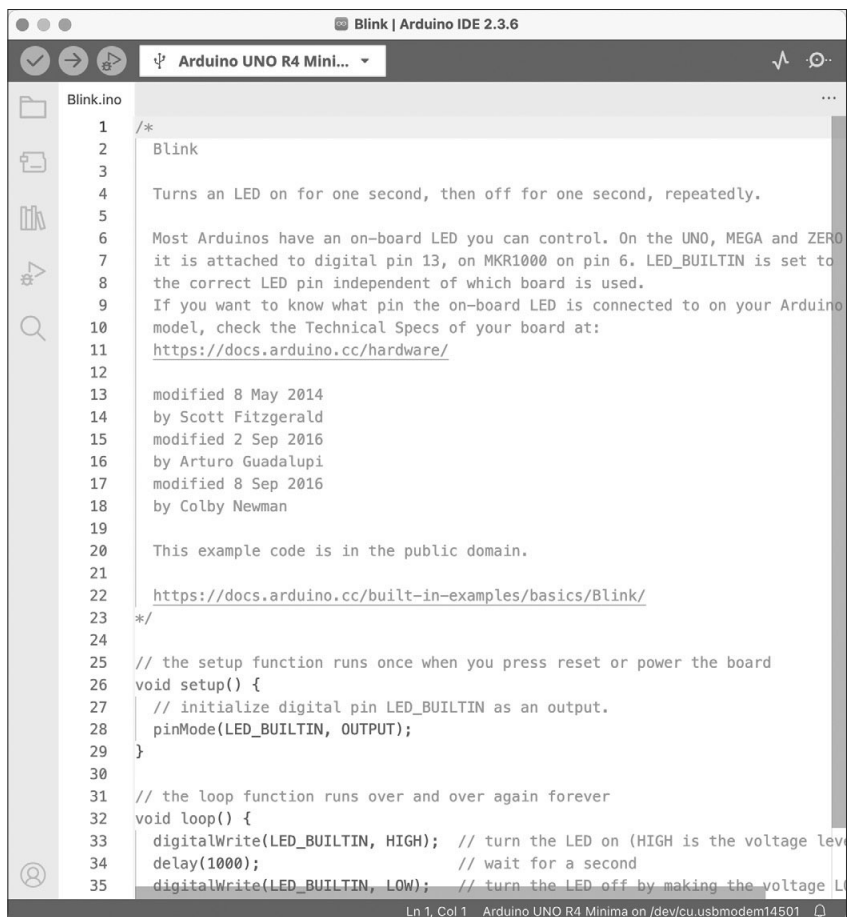
A paneled azonosítása

Ahhoz, hogy feltölthesd a sketch-et, ellenőrizned kell néhány dolgot. Először is meg kell erősítened, hogy milyen paneled is van. Ahogy arról a 2. fejezetben már olvashattál, rengeteg Arduino-eszköz és többféle USB-s panel közül választhatsz. Az USB-s panelek – a könyv írásakor – legújabb generációja az Uno R4 Minima, ezért a következő példákban ezt a panelet fogjuk használni. Ha régebbi paneled, például egy Uno R3-ad van, az alapvető funkciók nagyon hasonlóak lesznek, de ügyelned kell arra, hogy a példákat a saját paneledhez igazítsd.

3.1. ábra.
Keressük meg
a Blink
sketch-et

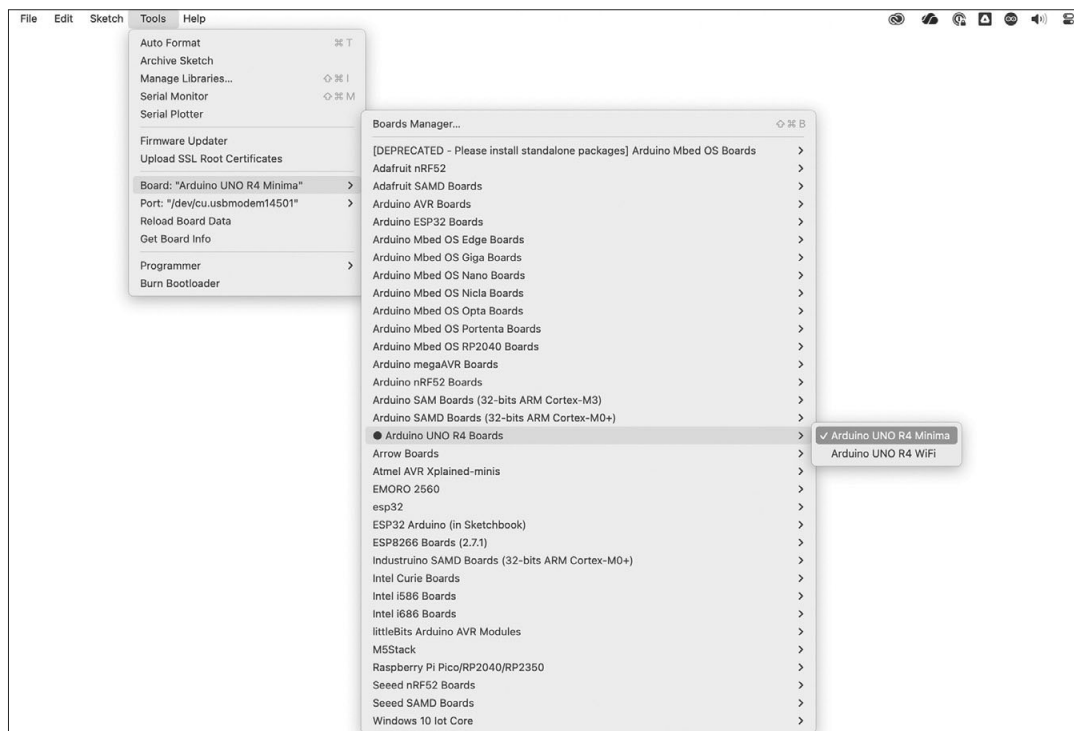


3.2. ábra.
A Blink sketch



Konfiguráljuk a szoftvert

Paneled típusának azonosítása után, az információt meg kell adnod a szoftvernek. Az Arduino fő menüsorában (amelyet Windowson az Arduino-ablak tetején, macOS-en a képernyő tetején találsz) válaszd a Tools (Eszközök) ⇨ Board (Panel) menüpontot. Ekkor megjelenik az Arduino-szoftver által támogatott különböző paneltípusok listája. Válaszd ki a listából a paneledet, ahogy azt a 3.3. ábrán is láthatod.

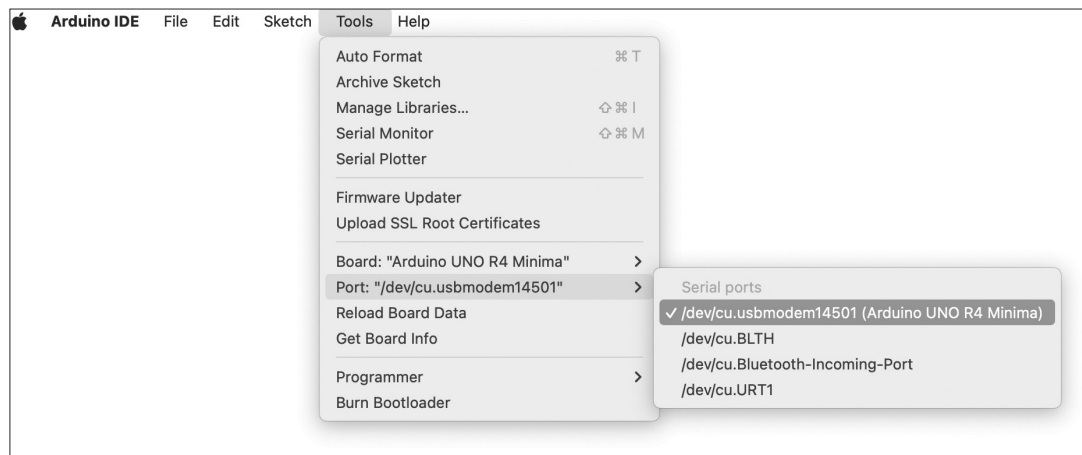


3.3. ábra.
Válaszd az
Arduino UNO
R4 Minima
lehetőséget
a Board
menüben

Ezután kell kiválasztanod a soros portot. A *soros port* az a kapcsolat, amelyen keresztül a számítógép és az Arduino-eszköz kommunikálni tud egymással. A *soros* kifejezés arra utal, hogy az adatok bitenként (0 vagy 1 értékenként) kerülnek továbbításra. A *port* a fizikai csatlakozó (más néven interfész), ebben az esetben egy USB-csatlakozó. A soros kommunikációról bővebben a 6. fejezetben olvashatsz.

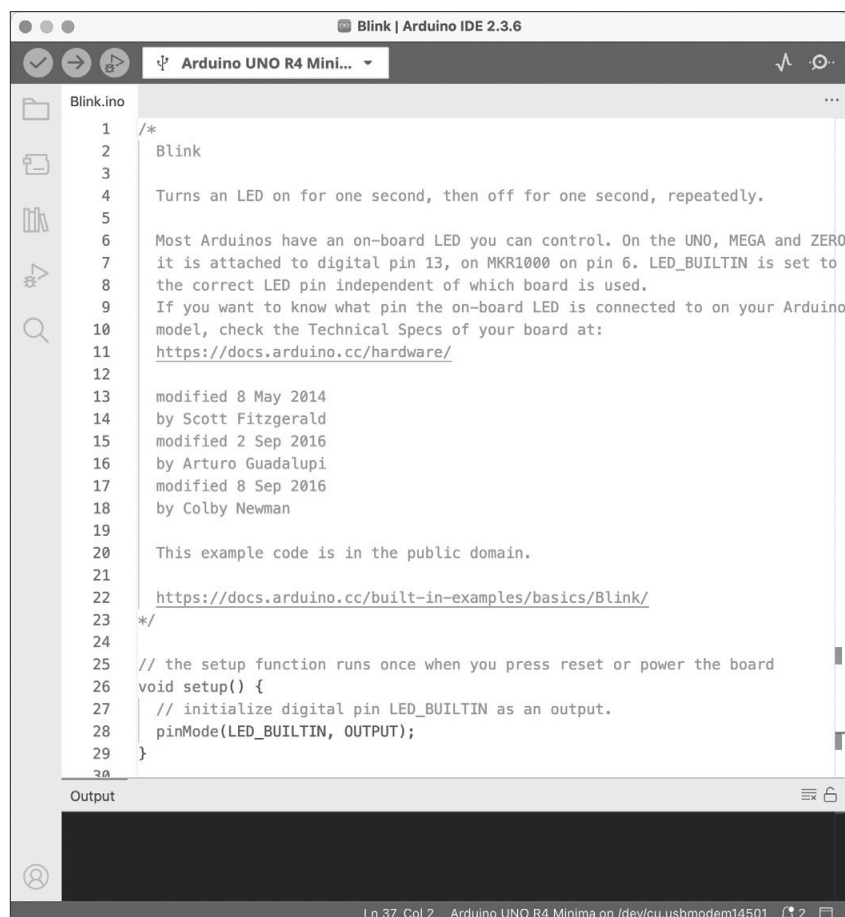
A soros port meghatározásához válaszd a Tools ⇨ Port menüpontot. Ekkor megjelenik a számítógépedhez csatlakoztatott eszközök listája (lásd a 3.4. ábrát). Ezen a listán minden olyan eszköz szerepel, amely soros kommunikációra képes, de minket jelenleg csak az érdekel, hogy megtaláljuk az Arduinót. Ha épp most telepítetted és csatlakoztattad az Arduinót, akkor annak a lista tetején kell lennie. A macOS-felhasználók számára az Arduino /dev/cu.usbmodemXXXXXX vagy /dev/cu.usbmodemXXXXXX néven jelenik meg (ahol az XXXXXX egy véletlenszerűen hozzárendelt szám). Windowson is ugyanez a helyzet, csak a soros portok neve COM1, COM2, COM3 és így tovább. A legmagasabb szám általában a legújabb eszközt jelöli.

3.4. ábra.
Az Arduino-
szoftver szá-
mára elérhető
soros kapcso-
latok listája



Miután megtaláltad a soros portodat, válaszd ki. A portnak az Arduino grafikus felhasználói felületének (GUI-jának) jobb alsó sarkában kell megjelennie, a kiválasztott panellel együtt (lásd a 3.5. ábrát).

3.5. ábra.
A panel és a
port az Ardui-
no GUI-jában

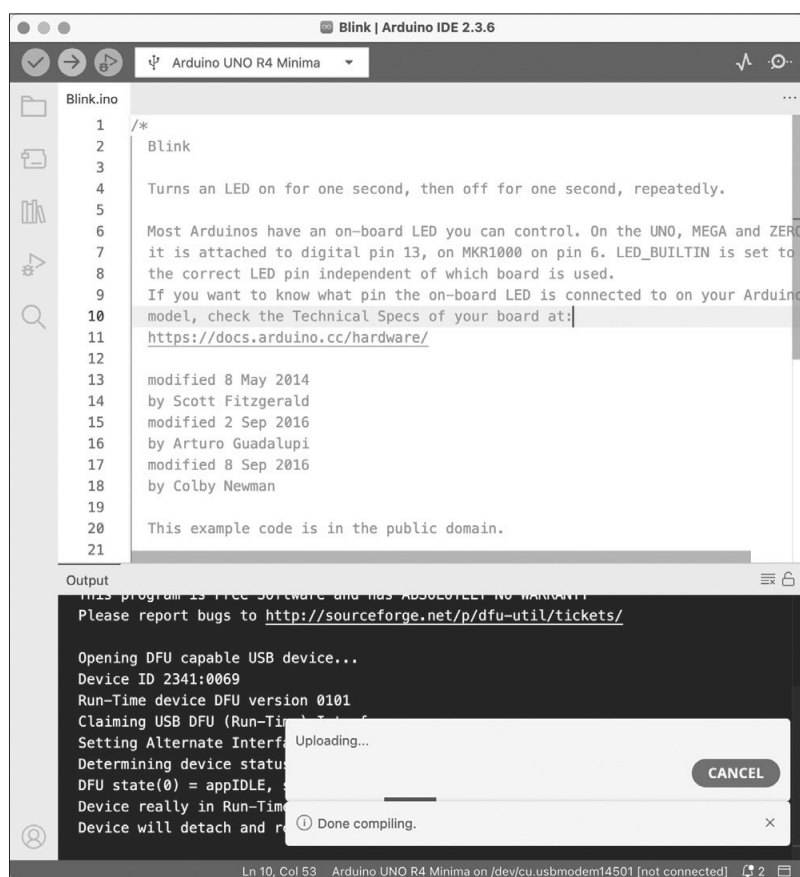


A sketch feltöltése

Most, hogy megadtad az Arduino-szoftvernek, hogy milyen panellel kommunikálsz, és melyik soros portot használsz, feltöltheted a Blink sketch-et, amelyet a fejezet elején megkerestünk.

Először kattints a Verify (Ellenőrzés, ✓) gombra. A Verify gombbal elindított ellenőrzés megnézi, hogy a kód biztosan értelmes-e, vagyis hogy a szintaxis olyan módon van-e megírva, hogy az Arduino megértse azt (lásd a 2. fejezetet). Ez a folyamat azonban nem feltétlenül jelenti azt, hogy a kódod azt fogja csinálni, amit vársz.

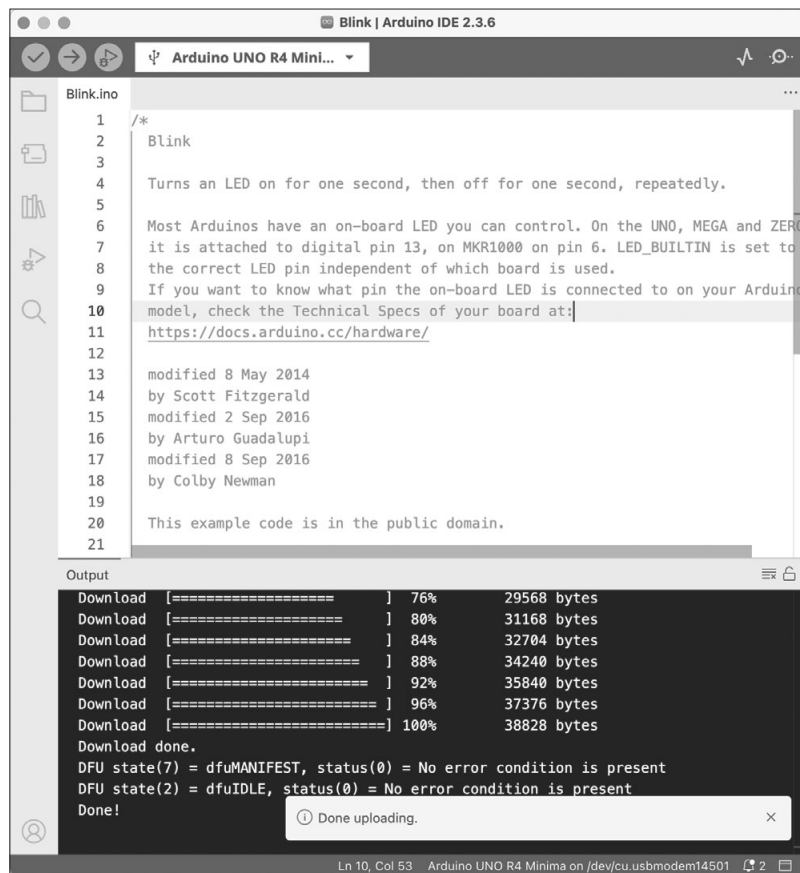
Elvileg néhány másodpercre meg kell hogy jelenjen egy folyamatjelző sáv és egy *Compiling Sketch* üzenet, majd amikor a folyamat befejeződött, a *Done Compiling* (Fordítás befejezve) üzenet (lásd a 3.6. ábrát).



3.6. ábra.
A folyamatjelző sáv mutatja, hogy a sketch fordítása folyamatban van

Ha a sketch fordítása sikeresen befejeződött, akkor rákattinthsz a Verify gomb melletti Upload (Feltöltés, →) gombra. Ekkor megjelenik egy folyamatjelző sáv, és a paneleden az RX és a TX LED-en (melyekről a 2. fejezetben olvashattál) élénk aktivitást láthatsz. Ezek jelzik, hogy az Arduino adatokat küld és fogad. Néhány másodperc múlva az RX és a TX LED abbahagyja a villogást, és az Arduino-ablak alján megjelenik a *Done Uploading* (Feltöltés befejezve) üzenet (lásd a 3.7. ábrát).

3.7. ábra.
Az Arduino-
szoftver sike-
resen feltöltöt-
te a sketch-et



Gratulálj magadnak!

Ekkor azt kell látnod, hogy az L jelölésű LED megnyugtatóan villog: egy másodpercig világít, egy másodpercig nem. Ha így van, akkor veregesd meg a saját válladat! Most töltötted fel az első Arduino-kódodat, és beléptél a fizikai számítástechnika világába!

Ha nem látod, hogy az L villog, fúsd át újra az előző szakaszokat. Győződj meg róla, hogy megfelelően telepítetted az Arduinót, majd próbáld meg még egyszer. Ha még mindig nem látod, hogy az L villog, akkor lehet, hogy a paneled sérült vagy hibás. Az Arduino ügyfélszolgálatával a <https://www.arduino.cc/en/contact-us/> címen veheted fel a kapcsolatot (angol nyelven), vagy esetleg ahhoz a forgalmazóhoz is fordulhatsz, akitől a paneledet vettél.

Ez történt

A legkisebb erőfeszítés nélkül feltöltötted az első sketch-et egy Arduinóra. Gratulálunk!

Csak hogy összefoglaljuk, már túl vagy a következőkön:

- » Csatlakoztattad az Arduinódat a számítógépedhez
- » Megnyitottad az Arduino szoftvert
- » Beállítottad a panelt és a soros portot
- » Megnyitottad a Blink sketch-et az Examples (Példák) mappából, és feltöltötted a panelre

A következő részben végigvesszük az imént feltöltött sketch különböző részeit.

A részletek

Ebben a részben kicsit részletesebben is átnézzük a Blink sketch-et. Mi is történik valójában? Amikor az Arduino-szoftver beolvassa a rövid programot, gyorsan végighalad rajta, sorról sorra. Szóval úgy értheted meg a legjobban a kódot, ha ugyanúgy végighaladunk rajta, csak lassan.

Az Arduino a C programozási nyelvet használja, amely az egyik legszélesebb körben használt nyelv. Rendkívül hatékony és sokoldalú nyelv, de beletelik egy kis időbe, mire megszokja az ember.

Ha követted az útmutatást az előző szakaszban, akkor már a képernyődön kell lennie a Blink sketch-nek. Ha nem így van, akkor a File ⇨ Examples ⇨ 01.Basics ⇨ Blink menüponttal érheted el (lásd a 3.1. ábrát).



VIGYÁZAT

A sketch megnyitásakor valami ilyesmit kell látnod angolul: (A könyvben a sketch-ekben szereplő leírások magyar fordításait olvashatod, hogy még jobban megértsd a kódok működését, de természetesen amikor betöltöd a példákat, az eredeti, angol nyelvű változatot fogod látni. Az egyértelműség kedvéért magát a kódot a könyvben is eredetiben láthatod.)

```
/*
```

Villogás

Egy másodpercre bekapcsol egy LED-et, majd egy másodpercre kikapcsolja, és ezt ismételteti.

A legtöbb Arduino rendelkezik egy beépített LED-del, amelyet vezérelhetsz. Az UNO, a MEGA és a ZERO esetében ez a 13-as digitális pinhez, az MKR1000-nél pedig a 6. pinhez van csatlakoztatva. A LED_BUILTIN (beépített LED) érték a használt paneltől függetlenül a megfelelő LED-pinre van állítva.

Ha szeretnéd tudni, hogy a te Arduino modelleden a beépített LED melyik pinhez van csatlakoztatva, a következő oldalon ellenőrizheted a panel műszaki adatait:

<https://docs.arduino.cc/hardware/>

módosította 2014. május 8-án

Scott Fitzgerald

módosította 2016. szeptember 2-án

Arturo Guadalupi

módosította 2016. szeptember 8-án

Colby Newman

Ez a példakód nyilvánosan hozzáférhető.

<https://docs.arduino.cc/built-in-examples/basics/Blink/>

*/

// A setup (beállítás) függvény egyszer fut le, amikor megnyomod

// a Reset gombot, vagy amikor bekapcsolod a paneledet:

```
void setup() {
```

```
    // A LED_BUILTIN digitális pint inicializáljuk kimenetként:
```

```
    pinMode(LED_BUILTIN, OUTPUT);
```

```
}
```

// A loop (ciklus) függvény végtelenszer lefut újra és újra:

```
void loop() {
```

```
    digitalWrite(LED_BUILTIN, HIGH);          // bekapcsoljuk a LED-et
```

```
        // a feszültségszintet HIGH (MAGAS) értékre állítva
```

```
    delay(1000);          // várunk egy másodpercet
```

```
    digitalWrite(LED_BUILTIN, LOW);          // kikapcsoljuk a LED-et
```

```
        // a feszültségszintet LOW (ALACSONY) értékre állítva
```

```
    delay(1000);          // várunk egy másodpercet
```

```
}
```

A sketch kódsorokból áll. Ha a kódot egészében nézzük, három különálló szakaszt azonosíthatunk:

- » Megjegyzések
- » void setup
- » void loop

Olvass tovább a részletekért.

Megjegyzések

A kód első szakaszában a következőket láthatod:

```
/*  
Villogás  
Egy másodpercre bekapcsol egy LED-et, majd egy másodpercre kikapcsolja,  
és ezt ismételteti.  
A legtöbb Arduino rendelkezik egy beépített LED-del, amelyet  
vezérelhetsz. Az UNO, a MEGA és a ZERO esetében ez a 13-as digitális  
pinhez, az MKR1000-nél pedig a 6. pinhez van csatlakoztatva. A LED_  
BUILTIN (beépített LED) érték a használt paneltől függetlenül a  
megfelelő LED-pinre van állítva.  
Ha szeretnéd tudni, hogy a te Arduino-modelleden a beépített LED  
melyik pinhez van csatlakoztatva, a következő oldalon ellenőrizheted a  
panel műszaki adatait:  
https://docs.arduino.cc/hardware/  
  
módosította 2014. május 8-án  
Scott Fitzgerald  
módosította 2016. szeptember 2-án  
Arturo Guadalupi  
módosította 2016. szeptember 8-án  
Colby Newman  
  
Ez a példakód nyilvánosan hozzáférhető.  
  
https://docs.arduino.cc/built-in-examples/basics/Blink/  
*/
```

Többsoros megjegyzés

Ne feledd, hogy a kódsorokat `/*` és `*/` szimbólumok között kell elhelyezni. Ezek a szimbólumok jelzik a *többsoros megjegyzés* kezdetét és végét. A megjegyzéseket közérthető, emberi nyelven írják, és – ahogy a nevük is sugallja – magyarázatot nyújtanak vagy megjegyzéseket tartalmaznak a kódhoz. A szoftver a megjegyzéseket figyelmen kívül hagyja, amikor lefordítja és feltölti a sketch-et. Így a megjegyzések úgy tartalmazhatnak hasznos információkat a kódról, hogy a kód futását nem befolyásolják.

Ebben a példában a megjegyzésben egyszerűen meg van adva a sketch neve, le van írva, hogy mit csinál, és arra vonatkozóan is tartalmaz egy megjegyzést, hogy ez a példakód nyilvánosan hozzáférhető. A megjegyzések gyakran egyéb részleteket is tartalmaznak, például a szerző vagy a szerkesztő nevét, a kód írásának vagy szerkesztésének dátumát, a kód működésének rövid leírását, a projekt URL-címét, de akár a szerző elérhetőségeit is.

Egysoros megjegyzés

A sketch további részében, a `setup` és a `loop` függvényen belül is ugyanolyan szürke színű szöveget láthatsz a képernyőn, mint a fenti megjegyzések. Ez a szöveg szintén megjegyzés. A `//` szimbólumok nem többsoros, hanem *egysoros megjegyzést* jelölnek. A rendszer figyelmen kívül hagy minden kódot, ami ezek után a perjelek után szerepel az adott sorban. Ebben az esetben a megjegyzés az utána következő kód-részletet írja le:

```
// A setup (beállítás) függvény egyszer fut le, amikor megnyomod
// a Reset gombot, vagy amikor bekapcsolod a paneledet:
```

Függvények

A következő két szakasz egy-egy függvény, és mindkettő a `void` szóval kezdődik: `void setup` és `void loop`. A *függvény* egy olyan kódrészlet, amely egy meghatározott feladatot hajt végre, és ez a feladat gyakran ismétlődő jellegű. Ahelyett, hogy újra és újra leírnád ugyanazt a kódot, használhatsz egy függvényt, amellyel utasíthatod a kódot arra, hogy újból hajtsa végre a feladatot.

Gondolj csak az IKEA-bútorok összeszerelésének általános folyamatára. Ha ezeket az általános utasításokat kódban írnád le egy függvény segítségével, az nagyjából így nézne ki: