

## Kódgenerálás állapotgépből Arduino platformra

A témalabor során a céloom az volt, hogy megvizsgáljam, hogy milyen lehetőségek vannak a modellvezérelt fejlesztésre Arduino mikrokontrolleren. Elsősorban azt vizsgáltam, hogy hogyan lehet egy állapotgépből olyan kódot generálni, amely futtatható a mikrovezérlőn is.

Az Arduino egy manapság rendkívül népszerű nyílt forrású eszköz, amely segítségével minden korábbinál egyszerűbbé vált az informatika és a fizikai eszközök összekötése. Hírnevét és elterjedtségét főleg alacsony árának, igényekhez igazodó verzióinak (Nano, Uno, Mega stb.) és a kialakult fejlesztői közösség nagyságának köszönheti. Az állapotgépek segítségével egy eseményvezérelt rendszer működését ábrázolhatjuk: mit kell vagy mit nem szabad csinálnia a rendszerünknek egy adott állapotban. A modellalapú fejlesztés sok esetben megkönnyíti a fejlesztők dolgát, hiszen egy jó modellt megalkotva a hibák egy része könnyen elkerülhető, észrevehető vagy javítható, továbbá egy átláthatóbb képet kaphatunk a rendszer működéséről. Biztonságkritikus rendszerek fejlesztése esetén elkerülhetetlen a modellalapú tervezés és megvalósítás.

Három fejlesztőeszköz közül választottam ki a későbbiekben használtat. Az első általam vizsgált a Yakindu Statechart Tools nevű ingyenes, Eclipse alapú eszköz volt, amely csak állapotgépből képes kódot generálni. A második a MagicDraw nevű program, amellyel többféle diagram is készíthető, de a hivatalos fóruma szerint állapotgépből csak Java kódot képes generálni. A harmadik az Enterprise Architect, amely egy profi – és emiatt viszonylag drága – eszköz. Ez képes az UML összes modelljét kezelni, kódot azonban csak egy kiegészítő (SineLaboreRT) segítségével tudtam generálni. A MagicDraw az újabb verzióiban ezt az exportálási lehetőséget nem támogatja, az Enterprise Architect azonban igen. A kód előállítását egy állapotgépből sikerült is, de a három eszköz közül nem ez bizonyult a legegyszerűbb módnak, ezért a választásom a további vizsgálatok elvégzéséhez a Yakindura esett. A Yakindu az Eclipse környezetnek köszönhetően lehetőséget biztosít különböző kiegészítők telepítésére is, ilyen például egy Arduino plugin, amely segítségével a megírt kód ellenőrizhető és rátölthető az Arduinora. Az állapotgép létrehozását is több funkcióval segíti: a hibákat képes jelezni tervezéskor (például ha nem érhető el egy állapot), valamint lehetővé teszi az elkészített modell szimulációját.

Elsőként egy olyan lámpa modelljét hoztam létre, amely egy gombnyomásra felkapcsolódik, majd 5 másodperc múlva lekapcsol, ha addig a gomb nem lett újra megnyomva. Újabb gombnyomás esetén újabb 5 másodperc múlva kapcsol le a lámpa. Ennek sikeres futtatása után egy összetettebb rendszert építettem meg, amely egy gyalogosátkelő lámpáinak vezérlése. A lámpák a Magyarországon megszokott módon működnek: kikapcsolt állapotban az autósoknak sárgán villog, a gyalogosoknak sötét, bekapcsolt állapotban pedig a megszokott módon jelez. A ki- és bekapcsolt állapot között egy gombbal lehet váltani. Ezen felül a gyalogosok egy gombbal jelezhetik átkelési szándékukat.

A Yakindu által generált kód a közlekedési lámpához nagyjából 1600 soros. Ebben megtalálható az állapotgép kódja, amely az állapotok enumerációját egy switch-case struktúrában kezeli és az egyes állapotok esetén az adott bemenetekre reagál és a szükséges lépéseket hajtja végre. Létrehoz továbbá egy projekt\_Required.h fájlt, amelyben azokat a függvényeket definiálja, amelyeket nekünk kell implementálni. Ahhoz, hogy a program az Arduinon is tudjon futni, importálnunk kell a generált fájlokat, és meg kell írunk a minden esetben szükséges kódsorokat. Ilyen például a led és gombok csatlakozóinak pin száma, a setup függvény, ahol ezeket ki- és bemenetként beállítjuk és a loop, ami folyamatosan fut a mikrokontrolleren. Ebben csak a bemeneteket kell beolvasni (meghívni az eseménykezelőt, ha azok változtak) és az idő múlását jelezni a modellnek. Ezen felül példányosítani kell az állapotgépet és implementálni kell a fent már említett interfészen definiált függvényeket.