

## **Molnár Dániel (OXOOF):**

### **Első projektfeladat:**

#### **Konfigurációs parancsok értelmezését leíró dokumentáció**

Ebben a dokumentumban be fogom mutatni azt, hogy az Arduino Uno fejlesztőkártyára írt szoftverem miként valósítja meg a konfigurációs parancsok értelmezését. A kommunikációra soros monitor használható, 9600 baud, illetve „Új sor” beállítással. A szoftverem az Arduino Uno board indulásakor, illetve Reset után „Ha konfigurálni szeretne, lépjen a kívánt üzemmódba, majd írja be a parancsot:” szövegű üzenettel fogadja a felhasználót. A fejlesztőkártya az 1. üzemmódban indul el, az aktuális üzemmódot mindig megjeleníti a soros monitoron. A módosítani kívánt üzemmód konfigurálásához a felhasználó köteles abban az üzemmódban lennie, tehát mindig csak az aktuális üzemmód konfigurálható.

A szoftver a konfigurációs parancsok végrehajtását a következő módon valósítja meg: Indítás után a felhasználónak a kívánt üzemmódba kell lépnie, amit a nyomógomb lenyomásával érhet el. Minden alkalommal, amikor a gomb lenyomásra kerül, a szoftver a következő üzemmódba lép. Ha a szoftver a kívánt üzemmódban van, akkor kezdődhet a konfiguráció. A felhasználó a parancsokat a soros monitor szövegmezőjébe írhatja be, és a soros monitor „küldés” gombjával, vagy a billentyűzet ENTER billentyűjének leütésével küldhető el a szoftver számára. A program karakterenként sorban, külön-külön beolvassa az elküldött parancsot egy char típusú „rx\_byte” nevű változóba, minden karakterbeolvasás után az adott karaktert hozzáfűzi -egy parancsbeolvasás kezdetében üres- String típusú „rx\_str” nevű változóhoz. Ezt követően a program az „rx\_str” változó első karakterét eltárolja a String típusú „ini” nevű változóba. Az „rx\_str” változó első karakterét követő összes karaktert pedig Integer típusúvá alakítja, és egy Integer típusú „num” nevű változóba menti, mivel a

parancs formátuma csak olyan lehet, hogy az első karaktere egy betű, az ezt követő összes többi karakter pedig 0 és 9 közötti (a határokat is beleértve) szám. A parancs elküldését követően a szoftver megnézi, hogy hányadik üzemmódban van a felhasználó, illetve megvizsgálja, hogy a parancs első karaktere „ini”, illetve az utána lévő „num” érték megfelel-e az adott mód beállításainak módosításához. Ellenőrzi, hogy a „num” érték a paraméter korlátainak megfelel, vagy pedig nem. Ha minden megfelel, akkor a program az adott paramétert felülírja, illetve az EEPROM memóriába menti el. Sikeres konfigurálás esetén a program "Sikeres konfigurálás!" üzenetet küld, illetve kiírja az adott változó módosított értékét, amit az EEPROM memóriából olvas ki, például: "idoT\_1 értéke: 20". Sikertelen konfiguráció esetén pedig "Érvénytelen konfiguráció!" hibaüzenetet kapunk.

A különböző üzemmódok paraméterei specifikus parancsokkal módosíthatóak. Az első üzemmódban a pulzálást megvalósító jel a hús PWM-fokozat egyedi időtartamának megadásával, USB-paranccsal konfigurálható.

Az első üzemmód konfigurálásához szükséges parancs formátuma: 'Axx', ahol 'xx' 10 és 40 közé eső érték. Példa: 'A10', 'A40'. Az első üzemmódban az időparaméter alapértelmezett értéke 25 milliszekundum/fokozat. Az időparaméter érvényességi tartománya: 10 milliszekundum /fokozat - 40 milliszekundum /fokozat. Ez a parancs a program Integer típusú „idoT\_1” nevű változóját módosítja, illetve az értéket az 5-ös memóriacímre elmenti.

A második üzemmódban az egy perióduson belüli felvillanások száma és az időparaméter USB-paranccsal konfigurálható. Az egy perióduson belüli felvillanások számának alapértelmezett értéke 4. Érvényességi tartománya: 2-6. Konfigurálásához szükséges parancs formátuma: 'Fn', ahol 'n' 2 és 6 közé eső szám. Példa: 'F2', 'F6'. Ez a parancs a program Integer típusú „vill\_2” nevű változóját módosítja, illetve az értéket az 6-os memóriacímre elmenti. A második üzemmód időparaméterének

értékének alapértelmezett értéke 50 milliszekundum. Érvényességi tartománya: 20 milliszekundum - 80 milliszekundum. Konfigurálásához szükséges parancs formátuma: 'Bxx', ahol 'xx' 20 és 80 közé eső szám. Példa: 'B20', 'B80'. Ez a parancs a program Integer típusú „T1\_2” nevű változóját módosítja, illetve az értéket az 7-es memóriacímre elmenti. A harmadik üzemmódban a két vezérlőjel félperiódusát jelentő időparaméter USB-paranccsal, külön-külön konfigurálható. A harmadik üzemmódban a két időparaméter alapértelmezett értéke 240 milliszekundum, ill. 270 milliszekundum. Mindkét paraméter érvényességi tartománya: 100 milliszekundum - 400 milliszekundum. A harmadik üzemmód időparamétereinek konfigurálásához szükséges parancsok formátuma: 'Cxxx' és 'Dyyy', ahol 'xxx' és 'yyy' 100 és 400 közé eső szám. Példa: 'C100', 'C400', 'D100', 'D400'. Ezek a parancsok rendre a program Integer típusú „Tp21\_3”, illetve „Tp22\_3” nevű változóit módosítja, illetve az értékeket a 8-9-10 és a 11-12-13 memóriacímekre elmenti. Ezen kívül, a harmadik üzemmód konfigurálásához egy String típusú „szamjegy” nevű változó is létre lett hozva. Ennek az az oka, hogy az érvényes konfigurációs beállításokat az Arduino Uno EEPROM memóriachipjébe menti a program, és mivel a harmadik üzemmód időparamétereinek akár 400 milliszekundumos értéket is felvehetnek, viszont az egy memóriacímre írható maximális érték 255 (és 400 nagyobb, mint 255) ,ezért ezt a program úgy oldja meg, hogy a parancs kezdő karaktere után lévő karaktereket levágja a parancsot tartalmazó „rx\_str” változóról, és eltárolja a „szamjegy” nevű változóba. Ez után ezt a „szamjegy” változót karakterekre bontja, és azokat Integer típusúvá konvertálja, majd külön külön a fentebb említett memóriacímekre elmenti.

A negyedik üzemmód nem konfigurálható.

Érvénytelen konfigurációs adatok érkezése esetén az adott üzemmód újraindul. Érvényes konfigurálást a szoftver nyugtázó üzenettel jelez vissza. Érvényes konfigurálás után a program az aktuális üzemmódban indul újra. Az érvényes, új konfigurációs értéket a program az EEPROM-ban tárolja. Indulásnál a program kiolvassa az EEPROM-ból az eltárolt konfigurációs értékeket. Reset után, érvénytelen konfigurációs értékek esetén a program az alapértelmezett értékeket alkalmazza. Ezeket a villogási üzemmódok indítása előtt elmenti az EEPROM-ba. A validációs tesztek a program fejlesztése közben, a kérdéses változók a szoftver futása során különböző folyamatok előtti és utáni értékeinek a soros monitorra való kiíratásával, illetve ezeknek az értékeknek az ellenőrzésével valósultak meg. A fejlesztés során az ezeket segítő kiíró metódusok egy része a programkódban kommentként szerepel, de a legtöbb a kódból eltávolításra került.