Dokumentáció a 2. beadandó 2/B feladatához

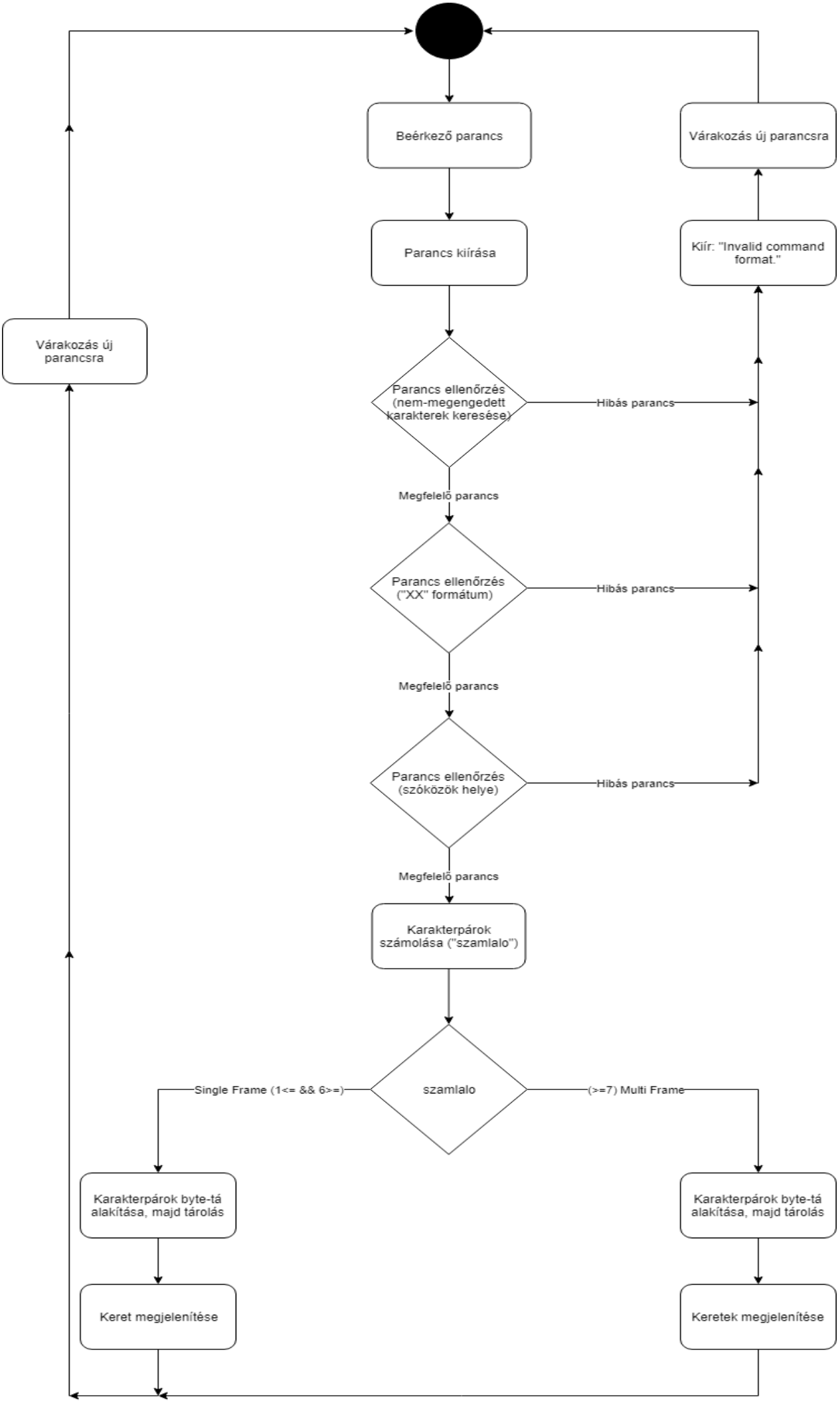
Készítette: Molnár Dániel (OXOOBF)

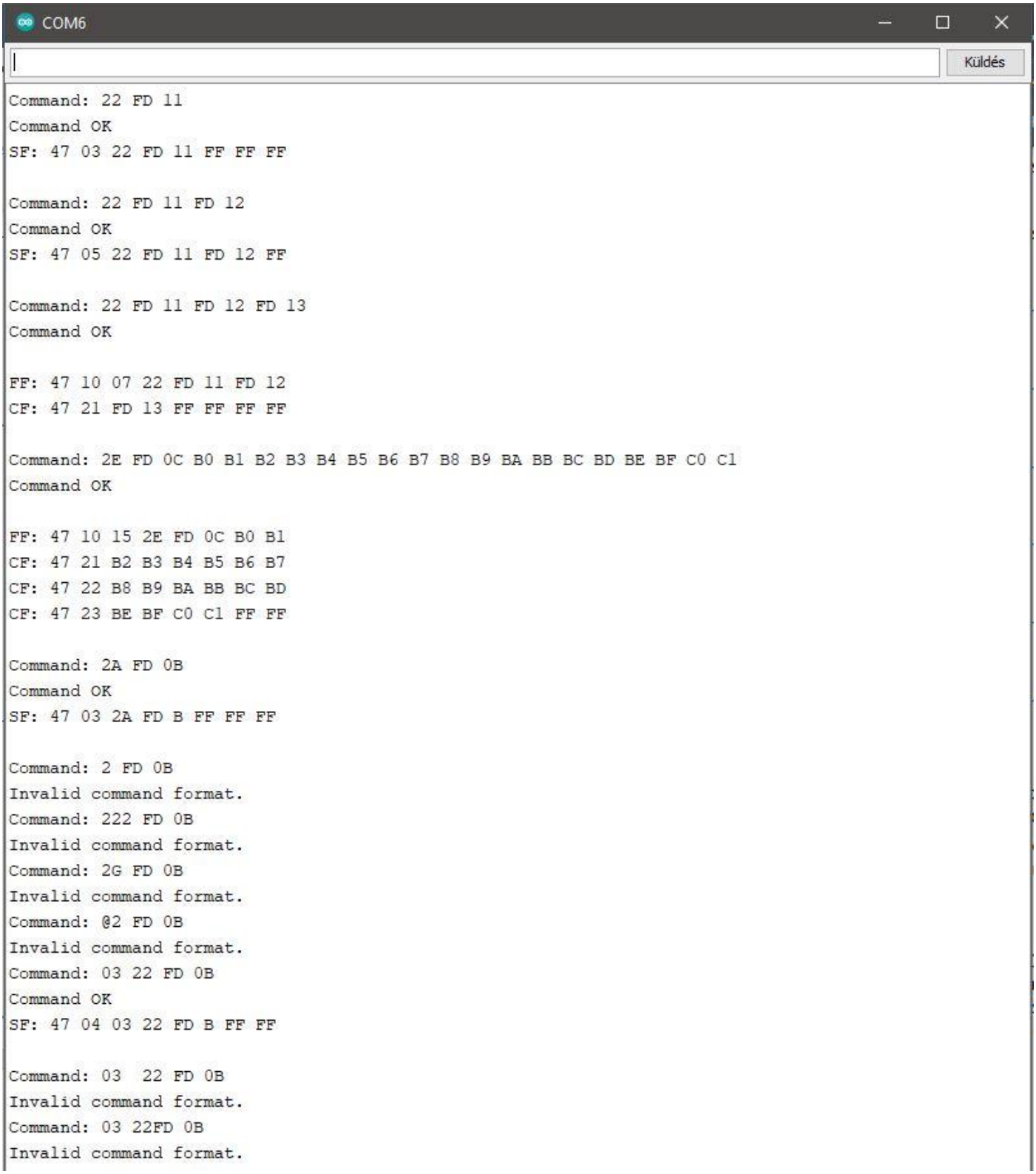
Ebben a dokumentumban be fogom mutatni az általam készített program működését, azon belül azt, hogy hogyan értékeli ki a kapott parancsot, majd hogyan szedi szét azt keretekre.

Az Arduino Uno fejlesztőkártyára írt program soros porton fogadja az IBS-nek szánt UDS-parancsot, majd a kapott adatsort kiértékelve keretekre bontja azt. A program kiírja az USB portra a generált keretek tartalmát. A kommunikáció sebessége 9600 baud, a parancsot pedig nem zárjuk le semmiféle extra karakterrel.

A program működése a következő:

Kezdésként meg van adva egy karakter tömb, „AllowedChars” néven. Ez tartalmazza az engedélyezett karaktereket (0-9;a-f;A-F), illetve a szóköz karaktert is. A program a soros portra várja felhasználótól a parancsot. Ha a felhasználó elküldte a parancsot, akkor a szoftver azt egy String változóba(„recievedString”) menti el. Ezután a program megvizsgálja és elmenti a String hosszát egy unsigned int típusú változóba(„recievedCharArray_LENGTH”). Ezt követően a létrehoz egy karaktertömböt megfelelő mérettel és karakterenként belemásolja a parancs karaktereit, majd kiírja a parancsot. Megvizsgálja a program, hogy a bevitt parancs csak olyan karaktereket tartalmaz-e, amelyek elfogadottak és az „AllowedChars” tömbben benne vannak. Ha igen, akkor ellenőrzi, hogy a parancs formátuma megfelelő-e, azaz „XX XX XX XX …” formátumú-e. Ekkor kiírja, hogy „Command OK”. Ha pedig a parancs nem megfelelő, akkor a kovetkezőt írja ki: „Invalid command format.”. Amennyiben a parancs megfelelt, a program megszámolja, hogy hány kettő egymás utáni karaktert(„XX”) tartalmaz a bevitelt tartalmazó karaktertömb. Ezt a számot egy unsigned int típusú, „szamlalo” nevű változóba menti. Ez alapján dől el, hogy a parancsot jelentő adatsort Single Frame-re vagy First Frame és Consecutive Frame-ek sorozatára kell-e bontania a programnak. Akár az előbbi, akár az utóbbi folyamat kerül megvalósításra, mindkét esetben az a következő lépes, hogy a karaktereket kettesével(„XX”) hexadecimális értékké alakítja, majd ezt az értéket a keretnek megfelelő byte típusú tömbbe (Single_Frame [FRAME_LENGTH]), vagy mátrixba (Multi_Frame[NUMBER_OF_FRAMES][FRAME_LENGTH]) menti, a megfelelő indexű helyre. Multi Frame esetében maximálisan 47 bájt hosszúságú parancsot tud értelmezni. A szamlalo értéke alapján, ha a szamlalo értéke kisebb hatnál vagy egyenlő hattal, akkor Single Frame-ként valósul meg a parancs, ha pedig hétnél nagyobb, vagy héttel egyenlő, akkor Multi Frame-ként, azaz First Frame és Consecutive Frame-ek sorozatára lesz bontva a parancs. Miután ez megtörtént, a program kiírja a soros monitorra az így kapott keretet a „display_byte_in_hexa” nevű függvény segítségével.


Beérkező parancs
Parancs kiírása
Parancs ellenőrzés (nem-megengedett karakterek keresése)
Hibás parancs
Kiír: "Invalid command format."
Várakozás új parancsra
Megfelelő parancs
Parancs ellenőrzés ("XX" formátum)
Hibás parancs
Megfelelő parancs
Parancs ellenőrzés (szóközök helye)
Hibás parancs
Megfelelő parancs
Karakterpárok számolása ("szamlalo")
szamlalo
Single Frame (1<= && 6>=)
(>=7) Multi Frame
Karakterpárok byte-tá alakítása, majd tárolás
Keret megjelenítése
Karakterpárok byte-tá alakítása, majd tárolás
Keretek megjelenítése
Várakozás új parancsra


COM6
Küldés
Command: 22 FD 11
Command OK
SF: 47 03 22 FD 11 FF FF FF
Command: 22 FD 11 FD 12
Command OK
SF: 47 05 22 FD 11 FD 12 FF
Command: 22 FD 11 FD 12 FD 13
Command OK
FF: 47 10 07 22 FD 11 FD 12
CF: 47 21 FD 13 FF FF FF FF
Command: 2E FD 0C B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C1
Command OK
FF: 47 10 15 2E FD 0C B0 B1
CF: 47 21 B2 B3 B4 B5 B6 B7
CF: 47 22 B8 B9 BA BB BC BD
CF: 47 23 BE BF C0 C1 FF FF
Command: 2A FD 0B
Command OK
SF: 47 03 2A FD B FF FF FF
Command: 2 FD 0B
Invalid command format.
Command: 222 FD 0B
Invalid command format.
Command: 2G FD 0B
Invalid command format.
Command: @2 FD 0B
Invalid command format.
Command: 03 22 FD 0B
Command OK
SF: 47 04 03 22 FD B FF FF
Command: 03 22 FD 0B
Invalid command format.
Command: 03 22FD 0B
Invalid command format.