



## **Autóipari szoftver- és hardverfejlesztés a gyakorlatban I**

**2020/2021. tanév, 1. félév**

### **2. projektmunka – feladatkiírás**

#### **2/A feladat – 15 pont**

Egészítsd ki az 12. oktatási héthez tartozó, "CS\_12\_3\_Acquiring\_VW\_Frames" nevű mintaprogramot úgy, hogy a program dekódolja a lekérdezett 0x2F és 0x30 azonosítójú keretekben érkező, az alábbiakban részletezett jeleket!

A 0x2F keretből:

BDM\_Spannung\_dyn,  
BDM\_I\_Bereich\_dyn,  
BDM\_Strom\_dyn,  
BDM\_Temp\_dyn.

A 0x30 keretből:

BZE\_SOC\_K20.

A keretek dekódolásához a LIN Definition File (LDF) tartalmazza a szükséges paramétereket. Ezek egy-egy screen-shot formájában megtalálhatók a 7. oktatási héthez feltöltött "VW\_signals" nevű könyvtárban.

A keret tartalmának kijelzése után a program a következő formátumban jelezze ki a dekódolt jeleket:

A feszültségre (BDM\_Spannung\_dyn): "U = 12.312 V", azaz három tizedes jegyen.

Az áramtartomány (BDM\_I\_Bereich\_dyn) kijelzésére nincs szükség, viszont értékét fel kell használnunk az áram értékének kijelzésénél.

Az áramra (BDM\_Strom\_dyn-ból BDM\_I\_Bereich\_dyn-t figyelembe véve):

"I = -3.296 A", ha BDM\_I\_Bereich\_dyn == 0. Itt LSB = 1 mA, így 3 tizedes jegyen.

"I = -52.51 A", ha BDM\_I\_Bereich\_dyn == 1. Itt LSB = 10 mA, így 2 tizedes jegyen.

"I = -881.9 A", ha BDM\_I\_Bereich\_dyn == 2. Itt LSB = 100 mA, így 1 tizedes jegyen.

"I = error", ha BDM\_I\_Bereich\_dyn == 3.

A hőmérsékletre (BDM\_Temp\_dyn): "T = 32 C".

A töltöttségi állapotra (BZE\_SOC\_K20): "SOC = 47 %".

Mindezt természetesen csak akkor, ha a keretek lekérdezésénél sem *Timeout*, sem helytelen ellenőrzőösszeg nem lép föl. Bármely ilyen esetben azt írja ki a program hogy "No signals"!

Az elvárás az, hogy ha a kiegészített programodat letöltöm a saját összeállításomra, akkor az az indítás után megjeleníti a kerettartalmat (ez már megvan a programban), majd a jelek dekódolt értékeit is a fent előírt formátumban.



## Segítség

A feladathoz számos mintakeretet rögzítettem a fejlesztési eszközünkkel, melyekhez a jelek dekódolt értékeit is elmentettem. A mintakereteket a "VW sample frames.txt" szövegfájl tartalmazza. Az egyértelműség kedvéért az amperban kifejezett áramértékeket is odaírtam a keretek után.

A "VW sample frames.txt" szövegfájl egy-egy olyan tömböt is tartalmaz a két lekérdezett kerethez, amellyel az IBS-től lekérdezett kereteket tudjátok szimulálni! Ez megkönnyíti a jelek dekódolását.

## 2/B feladat – 10 pont

Készíts egy minden eddigittől teljesen független, új programot, amely a soros porton fogadja az IBS-nek szánt UDS-parancsot, majd a kapott adatsort kiértékelve keretekre bontja azt. A program „írja ki” az USB portra a generált keretek tartalmát.

A kommunikáció sebessége 9600 baud legyen, a parancsot pedig nem zárjuk le semmiféle extra karakterrel.

Példák:

Command: "22 FD 11"

SF: 47 03 22 FD 11 FF FF FF

Command: "22 FD 11 FD 12"

SF: 47 05 22 FD 11 FD 12 FF

Command: "22 FD 11 FD 12 FD 13"

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"

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

A fentiek alapján tehát a programodnak meg kell állapítania, hogy hány bájtól áll a beérkező parancs, majd ennek függvényében kell a parancsot jelentő adatsort *Single Frame*-re vagy *First Frame* és *Consecutive Frame*-ek sorozatára bontania a helyes protokollinformációkat alkalmazva. Az IBS NAD-ja 0x47 (Node Address).

A bájt sorozat bármennyi bájtól állhat. (Természetesen 30-40 bájt elegendő lesz; ez lehet akár konfigurálható egy makrókonstanssal.) A parancselemző hexadecimális karakternek fogadja el a következőket: a, b, c, d, e, f, A, B, C, D, E, F. A program a parancs fogadása és kiértékelése után nyugtázza vagy elutasítja azt.



Példák és hibatípusok (további hibatípusok származtathatók):

"2A FD 0B"

"Command OK". – A parancs helyes.

"2 FD 0B"

"Invalid command format." – Az egyik szám nem XX formátumú.

"222 FD 0B"

"Invalid command format." – Az egyik szám nem XX formátumú.

"2G FD 0B"

"Invalid command format." – Az egyik szám nem hexadecimális érték.

"@2 FD 0B"

"Invalid command format." – Az egyik szám nem hexadecimális érték.

"03 22 FD 0B"

"Invalid command format." – Több, mint két space áll két szám között.

"03 22FD 0B"

"Invalid command format." – Nincs space két kétjegyű szám között.

## 2/C feladat – 10 pont

Készíts dokumentációt arról, hogy a programod hogyan értékeli ki a parancsot, majd hogyan szedi szét azt keretekre. Elvárt terjedelem 3-4 A4-es oldal (szorítkozzunk a 11-es, 12-es betűméretre!).

A dokumentációban szerepelnie kell részletes folyamatábrának vagy absztrakt automatának vagy állapot-átmenet gráfnak! A szöveges leírás álljon arányban a rajzokkal!

## Beküldés

A beküldési határidő: 2020. XII. 16., 7:00. A két programot és a dokumentációt tömörítsétek, a tömörített fájl neve ezt a mintát kövesse: "Minta\_Mókus\_123ABC.zip". Ebben a fájlnevben tehát még nyugodtan szerepelhetnek ékezetek. Az Arduino az, amelyik nem szereti azokat. Legyetek szívesek, a programjaitoknak a könyvtárát is hagyjátok meg, hogy ne nekem kelljen külön-külön létrehoznom azokat a pucér arduino-kódhoz a fordítás érdekében!

Kráncz Balázs  
Dr. Gömb

Veszprém, 2020. december 3.