

# Járműfedélzeti rendszerek II. projektfeladatok

## 1. Műszerfal grafikus kijelzővel

## Járműfedélzeti rendszerek II., projektfeladat

- Készítsék el a BIGAVR boardon található grafikus LCD kijelző kezeléséhez szükséges függvényeket (inicializálás, adat megjelenítése, karakterek/ikonok rajzolása stb.).
- A mellékelt **.dbc** fájl alapján olvassák be CAN hálózatról a következő információkat: idő, dátum, járműsebesség, ajtók állapota, irányjelző állapota, hibatárolóban található hibák száma.
- Tervezzenek egy egyszerű felhasználói felületet az LCD kijelzőn, amelyen a kiolvasott információk jól láthatóak, rendezettek és esetleg ikonokkal, grafikus elemekkel is segítik az információk áttekinthetőségét.
- Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.

## 2. Pedál és kormányszög szenzor analóg potméterrel

## Járműfedélzeti rendszerek II., projektfeladat

- Készítsenek próbanyákon egy BIGAVR boardra illeszthető pedálszenzor érzékelő hardvert, ami ellenirányú potméterekkel (azonos tengelyen két potméter) méri a fék és gázpedálállást és egy potméterrel a kormányszöget.
- Valósítsanak meg egy brake-throttle override funkciót, amely biztosítja, hogy a fékpedál lenyomása elsőbbséget élvezzen a gázpedállal szemben.
- Generáljon a szoftver hibajelzést, amely figyelmeztet, ha az azonos tengelyhez tartozó két potméter értékei eltérnek egymástól.
- Készítsenek egy kick-down kapcsolót, amely a gázpedál végállásánál aktiválódik.
- 100 Hz frekvenciával küldjék ki az érzékelt adatokat (gázpedál, fékpedál és kormányszög) és a hibajelzést CAN hálózatra a mellékelt **.dbc** fájl alapján.
- Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.

## 3. Járműdinamikai szimulátor

## Járműfedélzeti rendszerek II., projektfeladat

- Készítsenek egy járműszimulátort, amely a gázpedál, fékpedál és kormányszög bemenetek alapján kiszámítja a jármű mozgásállapotát.
- A bemenetek (gázpedál, fékpedál és kormányszög) értékeit CAN hálózatról olvassák be, a járműsebességet és pozíciót 100 Hz frekvenciával küldjék ki CAN hálózatra a mellékelt **.dbc** alapján.
- Járműmodellt a mikrokontrolleren futtatva számítsák ki a jármű aktuális pozícióját és sebességét, állásszögét a bemeneti adatok alapján.
- A járművet elegendő egy egyszerűsített modellel leírni, ami a jármű mozgását geometriai és kinematikai kapcsolatként írja le, figyelmen kívül hagyva az olyan komplexebb erőhatásokat, mint a gumibroncsok csúszása, a jármű tömegének eloszlása vagy a jármű felfüggesztésének dinamikája.
- A járműpozíciót és állásszöget jelenítsék meg az alfanumerikus LCD kijelzőn.
- Fejlesszenek ki egy C# alapú grafikus megjelenítést, amely valós időben 2D környezetben vizualizálja a jármű mozgását a számított pozíció és állásszög alapján. A PC és a mikrokontroller között UART kommunikációt valósítsanak meg.
- Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.

**4. Óra, dátum, hőmérséklet kiolvasás, beállítás****Járműfedélzeti rendszerek II., projektfeladat**

- Készítsék el a BIGAVR boardon található valós idejű óra IC-vel és hőmérséklet szenzorral való kommunikáció megvalósításához szükséges függvényeket.
  - Jelenítsék meg az alfanumerikus LCD kijelzőn a kiolvasott aktuális időt, dátumot és hőmérsékletet, és implementáljanak beállítási funkciókat, amelyekkel a felhasználóbarát módon gombokkal lehet módosítani az időt és dátumot.
  - 1 Hz frekvenciával küldjék el az idő, dátum és hőmérséklet adatokat a CAN hálózatra a mellékelt **.dbc** alapján.
  - Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.
- 

**5. Irányjelző vezérlőegység gombokkal szimulálva****Járműfedélzeti rendszerek II., projektfeladat**

- Készítsenek egy irányjelző vezérlőegységet, amelynél az irányjelző kar gombokkal van szimulálva.
  - Egyszeri gombnyomásra az irányjelző háromszor villanjon.
  - Implementáljanak egy funkciót, amely a CAN hálózatról érkező kormánysszögadatok alapján automatikusan kikapcsolja az irányjelzőt, ha a kormány visszatér az egyenes állásba.
  - Valósítsanak meg egy vészvillogó funkciót.
  - Amikor a jármű hirtelen jelentős lassulást vagy fékezést érzékelt, amely váratlan eseményre utalhat. A járműsebesség változása alapján számolják ki a lassulást.  $7-10 \text{ m/s}^2$  (0,7-1,0 g) közötti lassulás esetén automatikusan aktiválódjon a vészvillogó.
  - Az irányjelző és a vészvillogó állapotát 20 Hz frekvenciával küldjék ki a CAN hálózatra a mellékelt **.dbc** alapján.
  - Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.
- 

**6. Ablakemelő vezérlőegység****Járműfedélzeti rendszerek II., projektfeladat**

- Készítsenek egy ablakemelő vezérlőegységet, amely képes egy ablak teljes körű kezelésére.
  - Implementáljanak potméter használatával helyzetérzékelést, amely folyamatosan méri az ablak aktuális pozícióját.
  - Valósítsanak meg PWM (Pulzusszélesség-moduláció) alapú hajtást az ablak mozgatásához.
  - Ellenőrizték az ablak mozgási sebességét, és implementáljanak elakadásjelzést, amely figyelmeztet, ha az ablak nem tud fel- vagy lemenni. Generáljon hibakódot az ablak elakadása.
  - Az alfanumerikus kijelzőn jelenítsék meg az ablak aktuális állapotát és pozícióját.
  - Implementáljanak egy gyerekszár funkciót, amely megakadályozza az ablak véletlenszerű mozgatását.
  - 10 Hz frekvenciával küldjék ki az ablak állapotát, pozícióját és a hibakódot a CAN hálózatra a mellékelt **.dbc** alapján.
  - Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.
-

**7. Hibatároló vezérlőegység****Járműfedélzeti rendszerek II., projektfeladat**

- Készítsenek egy hibakódtárolót, amely fogadja és eltárolja a jármű vezérlőegységei által generált hibákat.
- Implementálják a km számláló funkciót, amely folyamatosan nyomon követi a jármű által megtett távolságot.
- A hibakódokat és a km számláló értékét a BIGAVR boardon található külső EEPROM memóriába tárolják.
- Valósítsanak meg CAN kommunikációt, amely a mellékelt **.dbc** alapján lehetővé teszi a hibakódok kiolvasását és törlését.
- Készítsenek egy C# diagnosztikai programot, amely grafikus felületen jeleníti meg a hibakódokat, a km számláló állapotát, valamint lehetőséget biztosít a hibák törlésére és a km számláló visszaállítására. A PC és a mikrokontroller között UART kommunikációt valósítsanak meg.
- Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.

**8. GPS feldolgozó****Járműfedélzeti rendszerek II., projektfeladat**

- Készítsenek egy GPS adatfeldolgozó rendszert, amely UART interfészen keresztül kommunikál egy GPS modullal.
- Implementáljanak egy UBX protokoll parser-t, amely képes egy GPS modul által küldött adatokat (koordináták, idő, sebesség, stb.) feldolgozni.
- Biztosítsanak hibakezelést a GPS adatfeldolgozás során, amely hibakódot generál, ha a beérkező adatok nem érvényesek vagy sérültek.
- A GPS adatok feldolgozása legyen folyamatos, és a rendszer reagáljon valós időben a beérkező információkra.
- A feldolgozott GPS adatokat és hibákat küldjék ki a CAN hálózatra a mellékelt **.dbc** alapján.
- Az egyes szoftvermodulokat szervezzék ki külön **.c** fájlba.

Az elkészült fél éves feladatot és prezentációt az utolsó gyakorlaton kell bemutatni 10 percben.

A kb. 6-8 diából álló prezentációnak a következőket kell tartalmaznia:

- Megoldandó feladat rövid ismertetése
- Működés bemutatása
- Szoftver felépítése
- Felmerülő problémák és azok megoldásai