



SZAKDOLGOZAT-FELADAT

Sisak Gergely (CPBWO2)

szigorló villamosmérnök hallgató részére

AUTOSAR CAN kommunikációs modulok megvalósítása

A modern gépjárművek biztonságtechnikai és kényelmi funkcióinak megvalósításában, környezetvédelmi jellemzőinek javításában stb. egyre jelentősebb szerepet kapnak a számítástechnikai megoldások. Ma egy prémium személyautó gyártójának közel száz elektronikus vezérlőegységből (ECU) és számos fedélzeti kommunikációs sínből kell kialakítani egy megbízhatóan működő elosztott rendszert, amely komoly algoritmus- és kommunikációtervezési, illetve munkaszervezési kihívást jelent. Az így adódó komplexitás uralására alakultak ki különféle szabványok, pl. a megbízható kommunikáció biztosítására a CAN és FlexRay sínek, a valós idejű feladatok futtatására az OSEK operációs rendszer vagy a futási idejű monitorozást támogató XCP protokollcsalád. A vezető autógyártók által 2002-ben életre hívott AUTOSAR konzorcium célja az, hogy ezen szakterületi szabványokra építve specifikáljon egy (i) *alapvető szolgáltatásstruktúrát*, amely eltakarja a hardver sajátosságait és támogatja az alkalmazási szoftver hordozhatóságát (base software stack, BSW), (ii) egy *modelllezési nyelvet* az ECU-kon futó alkalmazási szoftver szabványos leírására (software component template) és (iii) az alkalmazások és BSW-k ECU-n belüli és ECU-k közti *transzparens kommunikációját* lehetővé tevő elosztott runtime szolgáltatást (RTE):

- A *base software stack* magában foglalja az alacsony szintű eszközmeghajtókat (pl. EEPROM és Flash driverek), az ezeket eltakaró absztrakciós rétegeket (pl. memória absztrakciós felület) és az ezekre ültetett magas szintű funkciókat (pl. perzisztens adattárolás).
- A *modelllezési nyelv* lehetővé teszi, hogy precízen specifikáljuk az adattípusokat, illetve az alkalmazást alkotó komponensek interface-eit és belső felépítését.
- Az *RTE* egy generált glue kód réteg, amely eltakarja az alkalmazáskomponensek elől, hogy az általuk fogadott vagy küldött információ pontosan hogyan jut el a forrástól a célig, potenciálisan ECU-k közötti kommunikációs buszok igénybevételével.

A konzorcium jelentős hangsúlyt fektet az *API-k szabványosítására*, de kifejezetten támogatja a versengést az egyes szolgáltatások *megvalósításában* („Cooperate on standards, compete on implementation”).

A jelölt feladata az AUTOSAR base software stack egy részének megvalósítása a kapcsolódó tervezési, tesztelési és teljesítménymérési feladatok elvégzésével az alábbiak szerint:

- *A szabvány kapcsolódó részeinek megismerése*: (i) ismertesse az AUTOSAR rétegzett BSW struktúráján belül a *hálózati kommunikációért felelős modulok szerepét* és (ii) vázolja ezek együttműködését egy olyan *forgatókönyv bemutatásával*, amelyben két, különböző ECU-n futó alkalmazáskomponens vált üzenetet. A téma bemutatásánál koncentráljon a CAN sínen zajló kommunikáció sajátosságaira.
- *Algoritmustervezés és megvalósítás*: (i) *tervezzen egy-egy algoritmust*, amelyek megvalósítják a CAN stack kiválasztott moduljait (a konkrét modulok felsorolását a dolgozat bevezetője tartalmazza, pl. CAN State Manager (CanSM), CAN Transport Layer (CanTp), stb.) A szabvány a modulok megvalósítását egy *statikus* (kézzel írt, minden konfigurációban azonos) és egy *dinamikus* (konfigurációtól függő, tipikusan generált) részre bontással javasolja és megadja a konfigurációs adatok modelljét egy

osztálydiagrammal. A dinamikus kódrészletek előállítását a szabványos adatmodellből az aktuális megvalósítás feladata úgy, hogy az illeszkedjen a statikus részhez és megfeleljen az alkalmazási terület erőforrás-használatra vonatkozó követelményeinek (pl. minimális ROM használat). Ennek megfelelően (ii) *azonosítsa* a fenti komponensekben a *statikusan megírható részeket*, (iii) *valósítsa meg* ezeket C nyelven, ügyelve az autóiipari környezetben elvárt MISRA kódolási szabályok betartására, majd (iv) adja meg a *dinamikus részek* adatszerkezeteit és (v) javasoljon egy *sablont a kódgeneráláshoz* (vagyis hogyan lehet létrehozni az elvárt kompakt adatstruktúrákat a szabvány szerint rendelkezésünkre álló konfigurációs modellből).

- *A megvalósítás erőforrásigényének vizsgálata:* (i) adjon becslést az Ön által fejlesztett modulok *erőforrásigényére* (tár és futási idő) egy tipikus konfiguráció esetén; (ii) vizsgálja meg, hogy van-e olyan speciális konfiguráció, ahol az Ön által fejlesztett modulok funkciója részben egyszerűsödik (pl. partial networking támogatás kikapcsolása), becsülje meg, hogy mekkora az ilyenkor jelentkező *overhead*.
- *A megvalósítás tesztelése:* a szabvány az API specifikáció mellett megad egy *megfelelőségi teszt készletet* TTCN nyelven. Az Ön által fejlesztett modul helyességének vizsgálatához (i) illessze megvalósítását egy TTCN futtató környezetbe, ennek érdekében valósítsa meg a szükséges adaptereket és függvénycsomópontokat, (ii) futtassa a teszteket és győződjön meg arról, hogy megvalósítása megfelel a szabvány által elvártaknak illetve (iii) szükség esetén javítsa a megvalósítást.

Tanszéki konzulens: Dr. Sujbert László, docens

Külső konzulens: Dr. Pintér Gergely (ThyssenKrupp Presta Hungary Kft.)

Budapest, 2012. október 5.

.....
Dr. Jobbágy Ákos
tanszékvezető