

Elosztott, többprocesszoros beágyazott információs rendszerek fejlesztésének támogatása nyílt fejlesztő rendszer segítségével

Önálló laboratórium feladat összefoglalója (2. félév)

Steiner Mátyás (VXZCZ6)

Konzulens: Balogh András

BME Méréstechnika és Információs Rendszerek Tanszék

Informatikai infrastruktúra tervezése szakirány, 2006/2007. II. félév

Az önálló feladatom, hogy nyílt fejlesztőrendszerhez kiegészítést készíteni, amivel a fejlesztést menetét megkönnyítve egy többprocesszoros vagy többeszközös beágyazott rendszer fejlesztése valósítható meg. Az előző félév során a beágyazott rendszerek fejlesztési lehetőségeit vizsgáltam, irodalomkutatással próbáltam minél teljesebb képet alkotni a fejlesztési lehetőségekről. Ezáltal a kialakítandó fejlesztőkörnyezetről kialakult egy koncepció a megvalósítási funkciókra, kritériumokra. A mostani félévben a rendszer elemei közötti kommunikáció megvalósítása volt a cél, illetve egy újonnan felmerült funkció megvalósíthatósági tervezése, ez a funkció a rendszer működésének helyességvizsgálatát segíti elő, állapotkövetéssel, Statechart diagram segítségével.

A félév során kialakultak a rendszer elemei között végbemenő kommunikációnak a szabályai, és egy tesztkörnyezet kialakítása is megtörtént a működés helyességének vizsgálata érdekében, illetve létrejött az állapotkövető funkció megvalósíthatósági tervének kidolgozása.

A kommunikációs protokoll kidolgozásánál a cél az volt, hogy azon a rendszerelemen, amin keresztül a rendszer programozását végezzük, láthassuk az egész rendszer állapotát. Az alap architektúra, amin a fejlesztés folyt, egy központi elemből, és az abból elérhető alhálózatból áll. A központi elem csak abban az értelemben van kiemelve, hogy a kapcsolat a fejlesztő és a rendszer között rajta keresztül valósul meg.

A protokollal szembeni elvárás, hogy a debug funkciók megvalósítása közben a teljes rendszert a központi egységen keresztül tudjuk vezérelni. Az elosztott rendszer tekintve nem cél a rendszer reprodukálható viselkedése.

A protokoll megvalósításánál két szerepet különböztetünk meg, amelyeket az elosztott rendszer elemire értelmezünk, a központi kontroll egységet és a többi kliens egységeket. A szerepek változhatnak attól függően, hogy a fejlesztő melyik eszközhöz csatlakozik a rendszerben. A központi kontroll elem feladata a protokoll állapotátmeneteinek vezérlése és az eszközök állapotainak lekérése. A kliens feladata követni a központi egység által kiadott utasításokat. Mindkét egység feladata ugyanakkor még a helyes működés figyelése. Ebből következően bármelyik elem hibajelzése azonos súllyal számít a rendszer egészét tekintve. Továbbá a központi kontroll egység a protokoll állapotváltoztatási kérényét a visszaérkező nyugtázással érvényesíti.

A protokoll tesztelését Java alapú implementáción végeztem. A kód követi a célhardver fizikai tulajdonságait, illetve annak programbeli következményeit. A modellben egy magasabb absztrakcióként szerepel a fizikai átviteli közeg a rendszerelemek között (CAN), az üzenetküldési protokoll a fizikai csatornán, illetve a megszakításkezelés és a keresztbe programozás fizikai kapcsolata. A tesztesetekben ellenőrzött funkciók: üzenetek visszatartása, újraküldése, félrecímzése, információk lekérdezése, folyamatos helyes működés.

Rendszer működésének helyességvizsgálata, állapotkövetéssel, Statechart diagram segítségével, funkció kialakításához szükséges volt nyílt, szabadon kiegészíthető, forráskódú UML modellező rendszerre, illetve az illesztés kidolgozására a modellező rendszerekhez. A terv kidolgozásakor két megvalósítás került előtérbe. Az első a Rational Software Architect

olyan szintű kiegészítése, amikor a megjelenítendő modellt leíró állományban (XMI) állítanánk a kívánt elemek tulajdonságait, amit az RSA automatikusan követne, a másik esetben egy nyílt modellező rendszerhez írnánk egy interfészt, amin keresztül tudnánk a modellt változtatni. Mindkét megvalósításban közös az eszközök állapotainak követése, amit aszinkron üzenetekkel valósítana meg a rendszer. Továbbá mindkét megvalósítás előnye, hogy a modellezési rendszert Eclipse környezetben is lehet használni.

A következő félévben a állapotkövetési funkció megvalósítása és a fejlesztőkörnyezet kialakítása lesz a feladat.