

Beágyazott és Ambiens Rendszerek

1. gyakorlat tematikája

1. Fejlesztői kártya megismerése

- A fejlesztői kártya rövid ismertetése, csatlakoztatása a PC-hez.
- A fejlesztői környezet indítása, megismerése.

2. Fejlesztői környezet nézeteinek megismerése

A blink projekten keresztül a fejlesztői környezet különböző nézeteinek rövid bemutatása:

- A blink (STK3700_blink) nevű mintaprojekt megnyitása, megismerése.
- Simplicity IDE (fejlesztés)
- Debug (program letöltése, lejtetése)
- Energy profiler (áramfelvétel mérése, áramfelvétel nyomon követése függvény szinten)
- Configurator (röviden, későbbi gyakorlaton lesz róla szó)

3. Mintaprogram áttekintése, API függvények bemutatása

API-k (Application programming interface) szerepének rövid ismertetése.

BSP: Board Support Package

- CHIP_Init(): errata korrekció
- BSP_TraceProfilerSetup(): tracer inicializálás
- SysTick_Config: system timer konfigurálása
- BSP_LedsInit(): LED-ek inicializálása
- BSP_LedToggle(): LED-ek állapotainak váltása

4. Debug lehetőségek

- Függvények és definíciók visszakövetése
- Változók és regiszterek értékeinek megtekintése (lokális változókat csak akkor látjuk, ha a függvényen belül vagyunk)
- Feladat: a blink projektben kövessük végig, hogy a BSP_LedToggle(...) milyen regiszter melyik bitjeit állítja, és a Registers ablakban nézzük meg, hogy hogyan tükröződik a LED-ek állapota a regiszter bitjeiben.
- Nézzük meg röviden a disassembly kódot. Disassembly jelentősége.
- Mintakód átalakítása: a LED-ek villogtatása számláló jelleggel, és így nézni az áramfelvételt.

5. Energy profiling

- Energy profiler elindítása, rövid ismerkedés.
- A `CMU_HFRCOBandSet(cmuHFRCOBand_**MHz)` függvény segítségével állítani a frekvenciát (lehetséges értékek: 1MHz, 7MHz, 11MHz, 14MHz, 21MHz, 28MHz).
A LED-ek állapota legyen konstans (pl. mindegyik legyen bekapcsolva).
Az áramfelvétel ábrázolható a következő Python kód segítségével
http://home.mit.bme.hu/~orosz/BAMBI_gyak/gyak_1/fogyasztas.py
Letöltve a fájlt és az áramfelvétel értékeket beírva, a fájlra kétszer kattintva megjelenik a grafikon (újabb megjelenítés előtt be kell zárni).

6. LCD kijelző kezelése

- Hozzuk létre az `STK3700_lcd` projektet.
- Nézzük át a mintaprojektet, milyen az LCD kijelzőt kezelő API-k találhatóak.
- Időzítés magyarázata: a belső időzítő 1 ms gyakorisággal ad megszakítást, és ezt számoljuk. Amíg le nem telik az idő, a processzor várakozik.
 - Előny: egyszerű módszer
 - Hátrány: erőforrás-pazarló
- SysTick timer: speciális, magához a processzormaghoz kötődő, mérsékelt funkcionalitással rendelkező időzítő egység, nem tipikus timer periféria. Használható pl. operációs rendszerek, különféle ütemezők esetén.
- Feladat: a gombok kezelését lehetővé tévő fájlok hozzáadása a projekthez (a Main könyvtárba bemásolandók). A gombok segítségével különböző feladatok elkészítése:
 - felfelé/lefelé számlálunk
 - knight rider
 - írjuk ki a kijelzőre, hogy jobb vagy bal oldali nyomógomb van-e benyomva
 - számoljuk a gombnyomásokat (lefutó vagy felfutó élre)

7. Startup kód

- A `startup_gcc_efm32gg.s` fájl bemutatása
 - stack és heap definiálása
 - megszakításvektor-tábla definiálása
 - inicializált és nem inicializált változók feltöltése