

Beágyazott és Ambiens Rendszerek

2. gyakorlat tematikája

A gyakorlat során megvalósítandó feladat, hogy egy nyomógombot megnyomva egy LED kigyullad, a gombot elengedve pedig elalszik. A feladatot kétféle módszerrel oldjuk meg:

- nulláról kezdve, regiszterszintű beállításokkal (itt elegendő, ha egy LED kigyullad),
- a fejlesztői környezet hardverkonfigurátorát használva.

A fenti alapszintű feladatokat fejlesztés közben általában elfedik előlünk az eszközökhöz rendelt függvénykönyvtárak. A feladat megoldása azonban segíti megérteni a háttérfolyamatokat, illetve esetleges nem várt viselkedés esetén jártasságot ad a hasonló jellegű programkódok értelmezésében, debugolásában.

1. Periféria programozása regiszterszinten

- A kapcsolási rajz alapján derítsd ki, hogy a LED-ek és a nyomógombok melyik port melyik bemenetére vannak kötve (EFM32GG-BRD2200A-A03-schematic.pdf).
- Hozz létre egy üres projektet.
- Gyűjtsd össze a szükséges regisztereket. A regiszterek a processzor dokumentációjában találhatók (EFM32GG-RM.pdf). A következő tudnivalók és feladatok vannak:

- A regiszterek memóriába ágyazott regiszterek. A regiszterek címét a következő módon számíthatod. Minden funkciócsoporthoz tartozik egy báziscím, amit a 17. oldal 5.2. ábráján láthatunk. Itt a GPIO és a CMU (Clock Management Unit) báziscímekre lesz szükség. A báziscímhez képest az egyes funkcióregiszterek eltolása a dokumentációban adott az alábbiak szerint:

- CMU regiszterek: 11.4. fejezet, 136. oldal
- GPIO regiszterek: 32.4. fejezet, 764. oldal

A fenti fejezetekben megtalálhatók az egyes regiszterek funkciói, báziscímhez képesti eltolásai, illetve, hogy melyik bitmezőkben milyen paramétereket lehet beállítani.

Megjegyzés: ha ismert egy cím (pl. 0x400c8000+0x044), akkor a C nyelv szerint erre a következő módon lehet hivatkozni 32 bites változóként:

`*(volatile long unsigned int *) (0x400c8000+0x044)`

Érdemes elnevezni a regiszterhivatkozásokat, például:

`#define REG_A (*(volatile long unsigned int *) (0x400c8044))`

amelyekre például így hivatkozhatunk (a 13. bit 1-esbe állítása a REG_A regiszterben):

`REG_A |= 1<<13;`

- Engedélyezni kell a GPIO periféria órajelét (órajelelosztó hálózat: 128. oldal):
 - GPIO órajel: CMU_HFPERCLKEN0 regiszter GPIO biteje.
- A nyomógombokhoz tartozó lábakat bemenetként kell beállítani (INPUT).
 - A bemeneti értékeket az adott porthoz tartozó xx_DIN regiszterből lehet kiolvasni.
- A LED-ekhez tartozó lábakat kimenetként kell beállítani (PUSH_PULL).

- A kimeneti értéket az adott porthoz tartozó `xx_DOUTS` regiszterrel lehet közvetlenül írni, a `xx_DOUTSET` regiszterrel lehet magas szintre állítani, a `xx_DOUTCLR` regiszterben lehet alacsony szintre állítani (1-est írva az adott helyre).
- Írd meg a programot:
 - Engedélyezd a GPIO órajelet,
 - Inicializáld a LED-eket,
 - Állítsd be magas szintre legalább az egyik LED állapotát.
 - Kiegészítő feladat:
 - Inicializáld a nyomógombokat,
 - Egy végtelen ciklusban kérdezd le a nyomógomb(ok) értékét és állítsd a LED(ek) állapotát a gomboknak megfelelően.

2. Periféria programozása hardverkonfigurátor segítségével

- A hardverkonfigurátor az 5.2.2-es verzióval használandó.
- Valósítsd meg az 1-es pontban adott feladatot a fejlesztői környezet hardverkonfigurátorának segítségével.
- Hozz létre egy új projektet (Simplicity Configurator Program).
- Az 1-es pontban meghatározott portokat és lábakat konfiguráld ki- és bemenetnek a funkciótól függően.
- A „Custom pin name” mezőben nevezd el a lábakat a funkciónak megfelelően.
- Az `InitDevice.h` és `InitDevice.c` fájlokban nézd meg, hogy a konfiguráció eredményeként milyen változók és konfigurációs beállítások jöttek létre.
- A feladat megoldása során felhasználható parancsok: `GPIO_PinOutSet(Port, Pin)`, `GPIO_PinOutClear(Port, Pin)`, `GPIO_PinInGet(Port, Pin)`.