

Beágyazott mérési rendszerek tervezése és fejlesztése

Embedded measurement systems development and implementation

Prof. SZÁSZ Csaba

Kolozsvári Műszaki Egyetem, Villamosmérnöki Kar

Memorandumului út 28, 400114, Kolozsvár

email: Csaba.Szasz@emd.utcluj.ro

Abstract

This paper introduces a Field Programmable Gate Array-based (FPGA-based) embedded hardware topology, specially conceived for multifunctional and flexible measurement systems development and implementation. The physical core of this architecture is a ZyboZ7-10 powerful development board that embeds Vivado Design Suite technology-based software modules. The system enables the flexible and user-friendly interconnection to a large scale of various sensor types (such as temperature, humidity, pressure, ambient light, etc.) performing real-time measurements with high accuracy. Additionally, ensure fast prototyping and implementation by using the latest reconfigurable hardware technology combined with distributed computing of efficient real real-time code sources in C language. The paper discusses in details the specific design and development steps of this versatile and high performance embedded system. Finally, its full operation mode has been experimented on a laboratory prototype setup.

Keywords: *embedded system, FPGA processor, reconfigurable hardware technology, distributed system, sensor module*

Kivonat

A dolgozat egy Field Programmable Gate Array-alapú (FPGA-alapú) beágyazott felépítésű hardware rendszert mutat be, kimondottan többfunkciós és rugalmas mérőrendszerek fejlesztésére és kivitelezésére. A fizikai magja ennek a felépítésnek egy ZyboZ7-10-típusú korszerű fejlesztőrendszer, amelybe Vivado Design Suite-alapú software technológia van telepítve. Ez a rendszer lehetővé teszi a rugalmas és fejlesztőbarát összekapcsolást igen széleskörű és sokféle szenzorral (úgy, mint, hőmérséklet-, nedvesség-, nyomás-, fényerősség szenzorok, stb.), és nagy pontosságú méréseket végez valós időben. Továbbá, biztosítja a különböző gyors fejlesztési prototípusok létrehozását és kivitelezését - felhasználva a legkorszerűbb rekonfigurálható hardware technológiát, kombinálva a megosztott feldolgozású és valós időben végrehajtott algoritmusokkal C forráskódban. A dolgozat bemutatja ennek a jellegzetes és nagy feldolgozási teljesítményű beágyazott rendszernek a részletes tervezési és kivitelezési lépéseit. A végén pedig a teljes rendszer működése van bemutatva egy kísérleti laboratóriumi prototípus segítségével.

Kulcsszavak: *beágyazott rendszer, FPGA processzor, rekonfigurálható hardware technológia, megosztott rendszer, szenzor modul.*

1. BEVEZETÉS

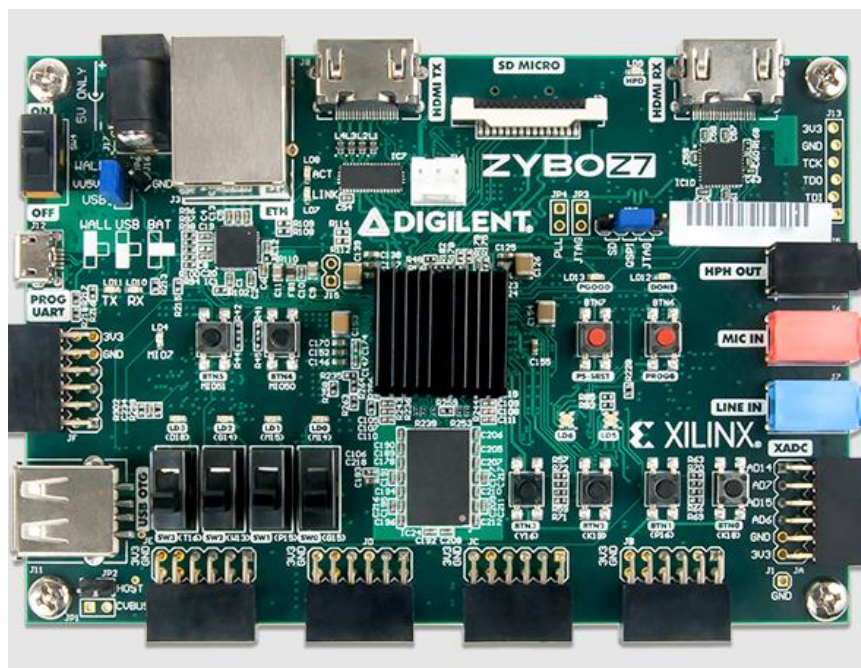
A beágyazott rendszereknek többféle meghatározása létezik a tudományos szakirodalomban. Van ahol a beágyazott rendszert a számítógépes hardware és software jellegzetes kombinációjaként tartják számon, konkrét funkcióra tervezve [1]. Más meghatározás szerint, a beágyazott rendszer nem más, mint egy mikroprocesszor-alapú hardware architektúra beágyazott software-el egy jól meghatározott funkció végrehajtására valós időben, úgy, mint egy független rendszer, vagy éppen egy nagyobb rendszer szerves részeként [2]. Olyan definíció is létezik, amely azt tartja, hogy egy beágyazott rendszer három részből áll: a fizikai hardware rész, a logikai software, valamint egy valós időben működő operációs rendszer. Ez a három rész pedig alkothat egy kis független rendszert, vagy részei lehetnek egy nagy kombinált rendszernek [3]. A beágyazott rendszereket olyan beágyazott számítógépeknek is tekintik, amelyek dedikált feladatok elvégzésére lettek megtervezve és ahol fontos szempont a kis fizikai méret és súly, valamint a kis energiafogyasztás [4]. Egy ugyancsak más megközelítés szerint a beágyazott rendszer nem más, mint egy

félvezető alapú mikroprocesszorra tervezett számítógép integrált software-el, mely jellegzetes feladatok végrehajtására van kitalálva. Ez tud működni függetlenül és önállóan, vagy más bonyolult rendszerek részeként, valós idejű számításokat végrehajtva [5]. A legtöbb meghatározás egy számítógép-alapú rendszerként értelmezi, amelyben megtalálható legalább egy processzor, a memória egység, és a ki/bemeneti interfész áramkörök, melyek célzott működést hajtanak végre mechanikai- vagy elektronikai rendszerekben.

Természetesen, ennek a dolgozatnak nem célja mindezeknek a meghatározásoknak az elemzése, vagy egy sokkal alaposabb áttekintése a nemzetközi szakirodalomban használt sokféle, beágyazott rendszerértelmezéseknek. De nagy általánosságban kijelenthető, hogy nem téves, ha a beágyazott rendszer úgy van meghatározva, mint egy olyan mikroprocesszor-alapú kisméretű hardware architektúra beágyazott software-el, amelyik célzott funkciókat hajt végre, vagy úgy, mint egy független rendszer, vagy mint részeként egy más nagy együttes rendszernek. Egy ilyen rendszer teljes megtervezését és gyakorlati kivitelezését tűzte célul a dolgozat, kimondottan nagy pontosságú mérési alkalmazásokra célzottan. A tervezési lépések magába foglalják, mint a hardware architektúra megtervezését, mint pedig a megfelelő software megírását és beágyazását a rendszerbe.

2. A HARDWARE RENDSZER MEGTERVEZÉSE

A javasolt beágyazott mérési rendszer fizikai alapegységének az 1-es ábrán látható ZyboZ7-10 típusú, Xilinx Zynq-7000 FPGA processzorra alapozott fejlesztőrendszer lett kiválasztva [6].



1. ábra – A ZyboZ7-10 fejlesztőrendszer

A Zynq család magába foglal egy kétmagos 667MHz sebességű ARM Cortex-A9 processzort, valamint egy Xilinx 7000-sorozat FPGA áramkört. Renделkezik DDR3L memória vezérlővel 8 DMA csatorna segítségével, 4 darab AXI3 kimeneti kapuval, 1GHz sebességű Ethernet-, USB2.0-, és SDIO csatornákkal, valamint SPI, UART, CAN és I2C interfész vezérlőkkel. A processzort lehet programozni JTAG, Quad-SPI-flash protokoll és microSD kártya segítségével.

Maga a fejlesztőrendszer rendkívül gazdag és sokféle multimédia- és ki/bemeneti csatlakoztató interfész áramkörökkel rendelkezik, amelyek egy igen erős számítógéppé alakítják a ZyboZ7-10 kártyát. Ide tartoznak az audió- videó eszközök csatlakoztatására szolgáló Pcam és HDMI kimeneti, HDMI bemeneti konnektorok, valamint az Ethernet PHY, USB-UART és USB 2.0 csatlakozók. Ugyanakkor a Zynq processzor mellett a lapon még megtalálható egy 1 GB nagyságú 533MHz sebességen működő 32 bites DDR3L memória blokk és egy 16MB Quad-SPI Flash memória. Továbbá, lehetséges egy microSD memória kártyához való direkt kapcsolás is. Az egész rendszert külső 5V egyenfeszültséggel, vagy USB interfészen keresztül lehet táplálni [6].

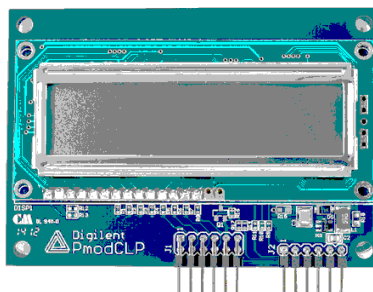
A ZyboZ7-10 sokoldalúsága és hatékonysága a ki/bemeneti egységein keresztül valósul meg igazából. Ezek közül meg kell említeni a kártyán elhelyezett 5 LED kijelzőt, a 6 nyomógombot, a 4 csúszókapcsolót, valamint a 2 RGB LED diódát. A külső hardware eszközök sokaságát pedig a 6 Pmod-típusú csatlakozón keresztül lehet igen könnyen a rendszerhez illeszteni. Példaként, a Pmod konnektorok lehetővé teszik a Digilent Co. cég több mint 70 interfész áramkörének direkt csatlakoztatását, ezek között sokféle szenzor, digitális kijelzők, motorvezérlők, vagy más interfész eszközök. Ugyanakkor, a 6 Pmod csatlakozó összesen 40 FPGA ki/bemenetet, 8 processzor ki/bemenetet, valamint 4 analóg-digitális bemenetet is jelent XADC konverterekhez [6].

A beágazott mérési rendszer felépítése érdekében sokféle szenzor-típust lehet kiválasztani a ZyboZ7-10 fejlesztőrendszerhez. A már meglévő 6 darab Pmod csatlakozó megléte miatt igencsak ajánlott a Digilent Co. szenzorjai közül választani. Ebben az esetben a Pmod HYGRO [7] és a Pmod ALS szenzor [8] lett kiválasztva, melyek a 2-es ábrán láthatóak.



2. ábra – A Pmod HYGRO és a Pmod ALS szenzorok

A Pmod HYGRO tulajdonképpen egy nagy pontosságú integrált hőmérséklet- és nedvesség szenzor. A modulba integrált HDC1080 áramkör segítségével 14 bites pontossággal lehet meghatározni a relatív nedvességet a környezetnek, $\pm 2\%$ hibahatáron belül. A hőmérséklet értékét pedig ± 0.2 °C pontossággal lehet megmérni. A mért adatok továbbítása az *SCL* és *SDA* jelek segítségével történik a feldolgozó egység felé [7]. A Pmod ALS szenzor a környezeti fényerősséget méri, 8 bit felbontással, felhasználva az ADC081S021 analóg-digitális konvertert, valamint a TEMA6000X01 integrált áramkört. Ez a Pmod az *SCK* és *SDO* jeleken keresztül továbbítja a mért információt [8].



3. ábra – 2x16 karakteres Pmod LCD kijelző

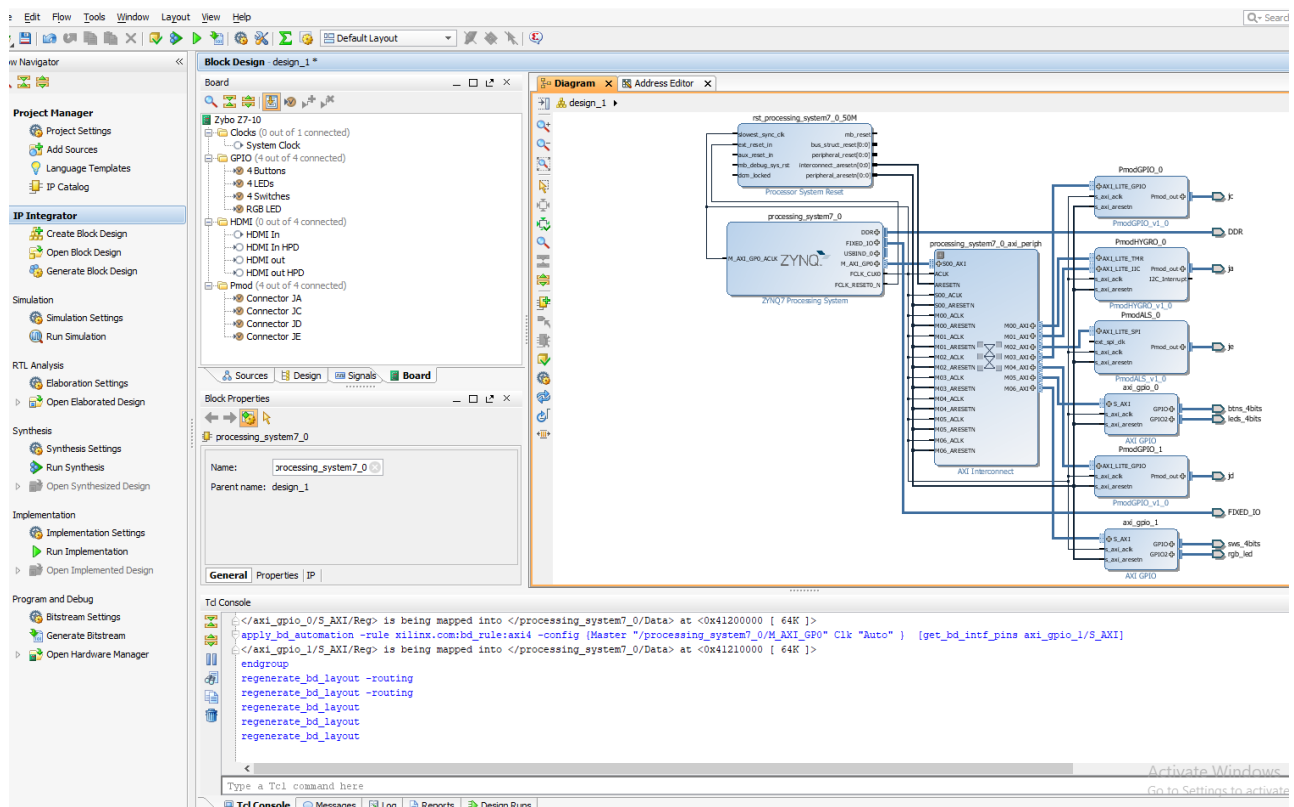
Maga a ZyboZ7-10 fejlesztőrendszer nem rendelkezik a feldolgozott adatok vizuális megjelenítésére alkalmas kijelzővel, ezért szükséges lenne egy olyan eszközre, amely megfelelően be tudná tölteni ezt a feladatot. Figyelembe véve a még meg lévő szabad Pmod csatlakozókat, az adatok kijelzésére igen alkalmas a 3-as ábrán látható Pmod LCD, amely egy 2x16 karakteres kijelző [9].

A ZyboZ7-10 rendszer másik nagy előnye, hogy teljesen kompatibilis a *Vivado Design Suite* integrált software fejlesztői rendszerrel (IDE – Integrated Development Environment) [10]. Ez a software csomag tulajdonképpen egy olyan új technológiát jelent, amely kimondottan az FPGA-alapú hardware rendszerek könnyen áttekinthető és gyors programozását teszi lehetővé. Maga a fejlesztői program két fontos alapegységből tevődik össze. Az első a *Vivado IP Integrator* modul, amely lehetővé teszi, hogy a fejlesztő a gyors tervezés és kivitelezés érdekében az úgynevezett *IP* (Intellectual Property) meglévő könyvtárakat használja. Az *IP*-blokk, vagy *IP*-mag, egy funkcionális logikai cella vagy integrált áramkörti egység, amely a létrehozója szellemi tulajdonát képezi. Ezek lehetnek sokféle interfész áramkör modulok, jól ismert standard áramkör típusok, a tervező által létrehozott bármilyen újszerű integrált áramkör, vagy éppen egy egész mikroprocesszor egység. Mindezek megfelelő könyvtárakban vannak tárolva, és tetszőlegesen

felhasználhatja bármelyik fejlesztő, hogy megtervezze saját jellegzetes hardware struktúráját, a maga projektjének megfelelően.

Köztudott, hogy az FPGA áramkörök használatának egyik legfontosabb előnye a rekonfigurálható hardware technológia alkalmazása. Tulajdonképpen ez a technológia csak az FPGA áramkörökre jellemző, minden más processzor esetében a hardware konfiguráció teljesen merev, vagyis egyáltalán nem változtatható, csak abban a formában használható, ahogyan azt a gyártó előzőleg megadta (legyártotta). Az FPGA-alapú rendszerek esetében a hardware felépítés tetszőlegesen változtatható (akár ennek működése közben is), beleértve itt a sínrendszerek és memória blokkok bit-szélességét, regiszterek nagyságát-hosszúságát, a ki/bemeneti lábak számát, vagy akár bármilyen új áramkör típust lehet létrehozni. Mindezekből kifolyólag, a programozó a legváltozatosabb digitális rendszereket állíthatja össze, majd tetszőlegesen változtathatja saját tervezési céljainak megfelelően. Ha még azt is figyelembe vesszük, hogy mindezen FPGA-alapú rendszereken a programok és ennek moduljai megosztott fizikai részekben futhatnak párhuzamosan (ugyanaz az órajel segítségével), ugyanazon a szilícium mag belsejében, akkor még rendkívülien nagy végrehajtási sebességeket is lehet elérni más hagyományos digitális rendszerekhez viszonyítva. A rekonfigurálható hardware technológia és az igen nagysebességű megosztott párhuzamos végrehajtás együttesen pedig egyedi képességeket biztosít az FPGA-alapú fejlesztőrendszereknek.

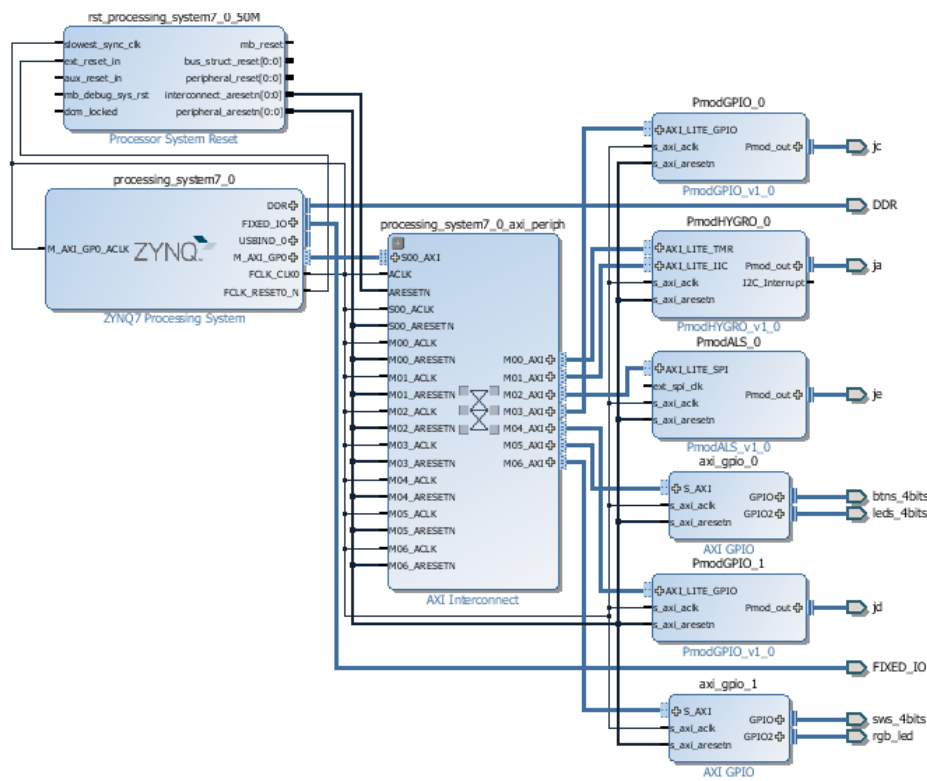
A 4-es ábrán a *Vivado Design Suite* software fejlesztői környezet *IP Integrator* modul menürendszerre van bemutatva [10]. A baloldali oszlopban (*Project Manager*) láthatóak mindazok a műveletek, amelyeket a hardware architektúra fejlesztése közben el lehet végezni (hardware blokk létrehozás, megnyitás, szintézis, program futtatás, kezelés, stb.).



4. ábra – A Vivado Design Suite IP Integrator modul

Onnan jobbra (*Block Design*) látható, hogy a ZyboZ7-10 rendszer melyik hardware elemei voltak becsatolva az adott alkalmazásba. A jelen esetben használva lesz a 4 nyomógomb, a 4 csúszókapcsoló és 4 LED dióda, valamint egy RGB kijelző dióda. Alább az is jól észrevehető, hogy mely Pmod csatlakozók lesznek kiválasztva a beágyazott mérési rendszer megtervezéséhez. Ezek szerint a *JC* és *JD* Pmod konnektorok, mint általános felhasználású ki/bemeneti portoknak lesznek konfigurálva (*PmodGPIO_0* és *PmodGPIO_1*), ezekre lesz illesztve a Pmod LCD kijelző. A *JA* kimeneteire a környezeti nedvesség és hőmérséklet mérésére használt *PmodHYGRO_0* interfész modul van kapcsolva, a *JE*-re pedig a fényerősség mérésére használt *PmodALS_0* szenzor. Mindezeknek az *IP* áramköröknek az összecsatolása a 4-es ábra jobb

oldalán (*Diagram*) van bemutatva. Ezek együttesen fogják alkotni a beágyazott mérési rendszer hardware (fizikai) architektúráját.



5. ábra – Az FPGA-alapú beágyazott mérési rendszer hardware felépítése

A *Vivado Design Suite IP Integrator* által elkészített áramkört összekapcsolás nagyított tömbrajza az 5-ös ábrán látható. Könnyen észrevehető, hogy a rendszer alapegysége a *ZYNQ* processzor modul, amelyhez az *AXI* (Advanced eXtensible Interface) interfész áramkör segítségével vannak csatlakoztatva a megfelelő sínrendszereken keresztül az összes többi integrált áramkör és ki/bemeneti egység [10]. Ennek a módszernek a rendkívülien nagy előnye az, hogy a hardware rendszer igen könnyen változtatható (rekonfigurálható hardware technológia), a legkülönbözőbb felhasználói alkalmazásnak megfelelően. Tehát a fizikai egységek nem megváltoztatatlanul vannak összekapcsolva, hanem program segítségével tetszőlegesen átköthetőek a felhasználó igényei szerint.

```

while(*Temp_amb)
{
    Lcd_char(*Temp_amb);
    Temp_amb++;
    GPIO_delay(&Lcd_control, 200000);
}

int main()
{
    init_platform();
    Xgpio switches, leds, RGB_led;
    float temperature, humidity;
    char Temp_amb[5], Hum_amb[5];

    GPIO_begin(&Lcd_data, XPAR_PMODGPIO_0_AXI_LITE_GPIO_BASEADDR, 0x00, 10000000);
    GPIO_begin(&Lcd_control, XPAR_PMODGPIO_1_AXI_LITE_GPIO_BASEADDR, 0x00, 10000000);

    HYGRO_begin(&HYGRO, XPAR_PMODHYGRO_0_AXI_LITE_IIC_BASEADDR, 0x40, XPAR_PMODHYGRO_0_AXI_LITE_TMR_BASEADDR, XPAR_PMODHYGRO_0_AXI_LITE_IIC_BASEADDR, 0x40, XPAR_PMODHYGRO_0_AXI_LITE_IIC_BASEADDR, 0x40); // Chip address of PmodHYGRO IIC, XPAR_PMODHYGRO_0_AXI_LITE_IIC_BASEADDR, 0x40);

    Xgpio_Initialize(&switches, XPAR_AXI_GPIO_0_DEVICE_ID); //Initialize input Xgpio variable
    Xgpio_SetDataDirection(&switches, 1, 0xFFFFFFFF); //set first channel three-state buffer to input
    Xgpio_Initialize(&leds, XPAR_AXI_GPIO_0_DEVICE_ID); //initialize output Xgpio variable
    Xgpio_SetDataDirection(&leds, 2, 0x00000000); //set second channel three-state buffer to output
    Xgpio_Initialize(&RGB_led, XPAR_AXI_GPIO_1_DEVICE_ID);
    Xgpio_SetDataDirection(&RGB_led, 1, 0x00000000);
    //Sleep(2000);

    Init_Lcd();
    Lcd_char("T");
    Lcd_char("e");
    Lcd_char("m");
    Lcd_char("p");
    Lcd_char("e");
    Lcd_char("r");
    Lcd_char("a");
    Lcd_char("t");
    Lcd_char(" ");
}

```

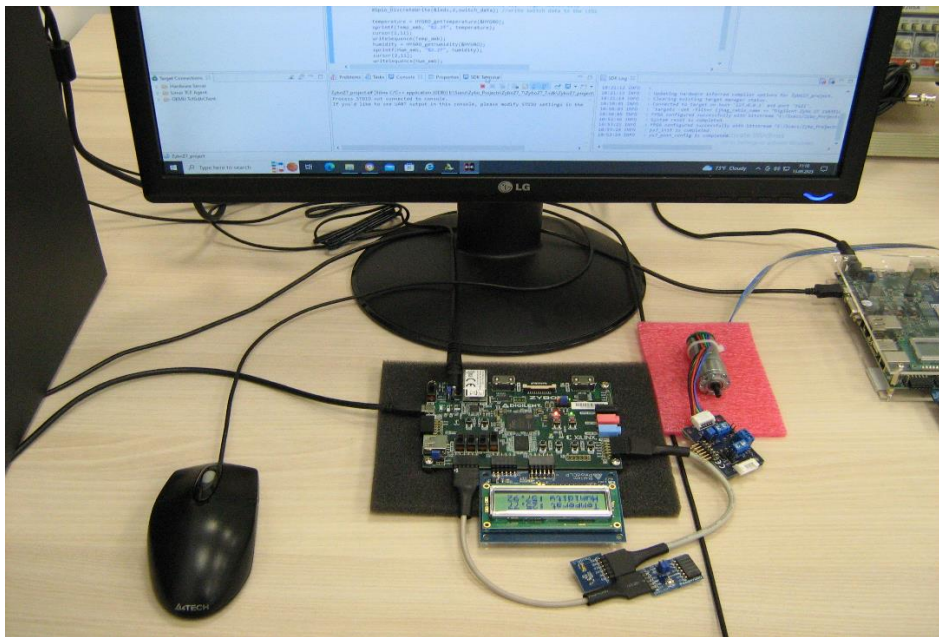
6. ábra – A Vivado ISE Design Tool menürendszere

A *Vivado Design Suite* másik igen fontos modulja a *Vivado ISE Design Tool*, ami a az előbb megtervezett *IP*-alapú hardware struktúrák programozását biztosítja *C* vagy *C++* fejlesztői környezetben [10]. A *Vivado ISE Design Tool* menürendszere a 6-os ábrán van bemutatva. A kép bal oldalán (*Project Explorer*) látszik az a hardware platform, amely programozva lesz - ebben az esetben a *design_1_wrapper_hw_platform_0*, valamint az, hogy milyen komponensekre épül ez a platform. A lenti sorokban az is észrevehető, hogy az *ISE* modul létrehozta a *ZyboZ7_project*-et *C* forráskódban, a platformot vezérlő program (*Zybo_project.c*) pedig az *src* alkönyvtárban van tárolva. A menürendszer közepén látható a felhasználó által kifejlesztett programcsomag a beágyazott mérési rendszer működtetésére. Ebben a konkrét esetben a Pmod LCD kijelző programozása látható a mért adatok megjelenítésére, aztán pedig a Pmod HYGRO szenzor mérési függvényeinek a meghívása az adott könyvtárból, valamint a mért adatok beolvasása és kijelzése a *ZyboZ7-10* kártyán.

A 6-os ábra jobb oldalán pedig leolvashatóak mindazon függvények és forráskód programok, amelyek még szükségesek a rendszer pontos működtetésére. A *Vivado ISE Design Tool* software fejlesztő komponens alapos ismerete után következhet a teljes mérési rendszer tesztelése, valamint konkrét mérések elvégzése az adott környezetben.

3. A BEÁGYAZOTT MÉRÉSI RENDSZER LABORATÓRIUMI PROTOTÍPUSA ÉS GYAKORLATI MÉRÉSEK

A *ZyboZ7-10* típusú, Xilinx Zynq-7000 FPGA processzorra alapozott beágyazott mérési rendszer laboratóriumi prototípusa a 7-es ábrán van bemutatva.



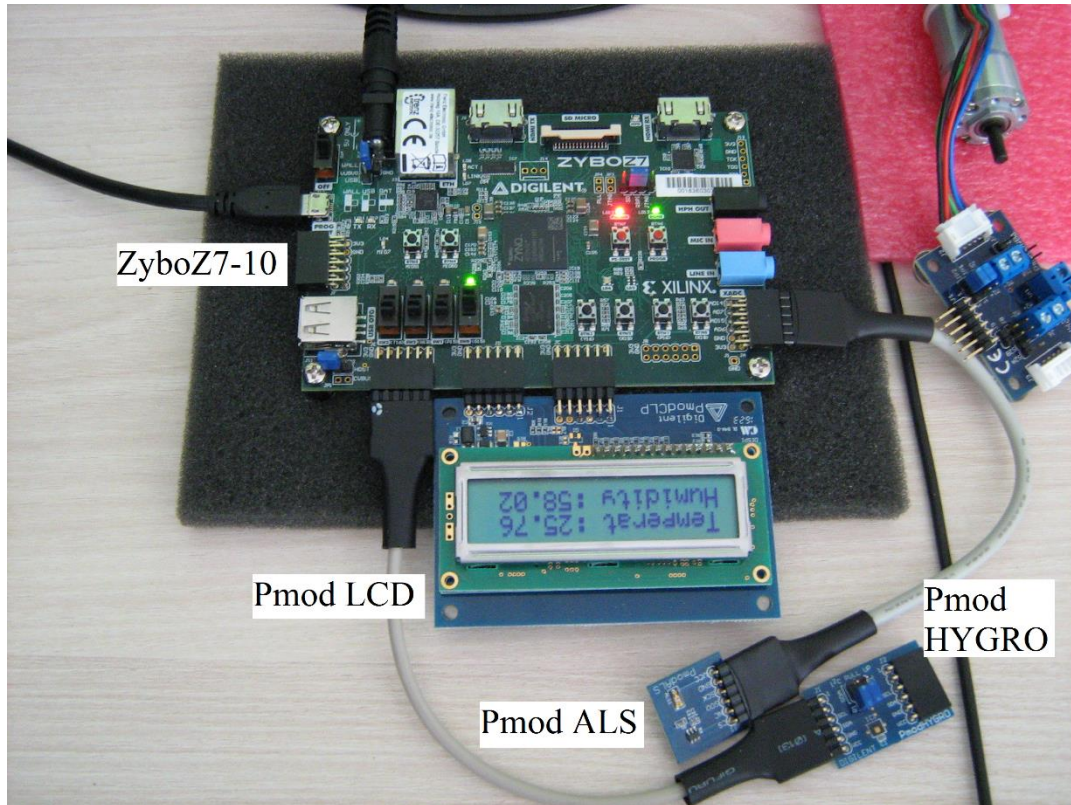
7. ábra – A beágyazott mérési rendszer – laboratóriumi fejlesztőrendszer.

A képen látható a *ZyboZ7-10* kártya a hozzácsatolt interfész áramkörökkel, valamint a háttérben a számítógépen futtatott *Vivado Design Suite* software fejlesztőrendszer. A 8-as ábrán néhány mérési eredmény látható: a környezeti hőmérséklet $25,77\text{ }^{\circ}\text{C}$, a nedvesség $58,02\%$, a fényerősség pedig 70 cd .



8. ábra – Mérési eredmények

Egy közelebbi képről (9. ábra) jobban meg lehet különböztetni a felhasznált szenzorokat és más interfész egységeket. Itt külön meg van jelölve a ZyboZ7-10 fejlesztőrendszer, a csatlakoztatott digitális Pmod LCD kijelző, valamint a Pmod ALS és Pmod HYGRO szenzorok a fényerősség-, hőmérséklet-, és nedvesség mérésére.



9. Ábra – A beágyazott mérési rendszer hardware felépítése

A beágyazott mérési rendszer működésének programozása C forráskódban történt, egy rövid programsor részlet a 10-es ábrán látható, ahol a fényerősség mérése van programozva.

```
XGpio_DiscreteWrite(&RGB_led,1,0x02);

//temp = TMP3_getTemp(&Temperature);
//sprintf(Temp_amb, "%2.2f", temp);

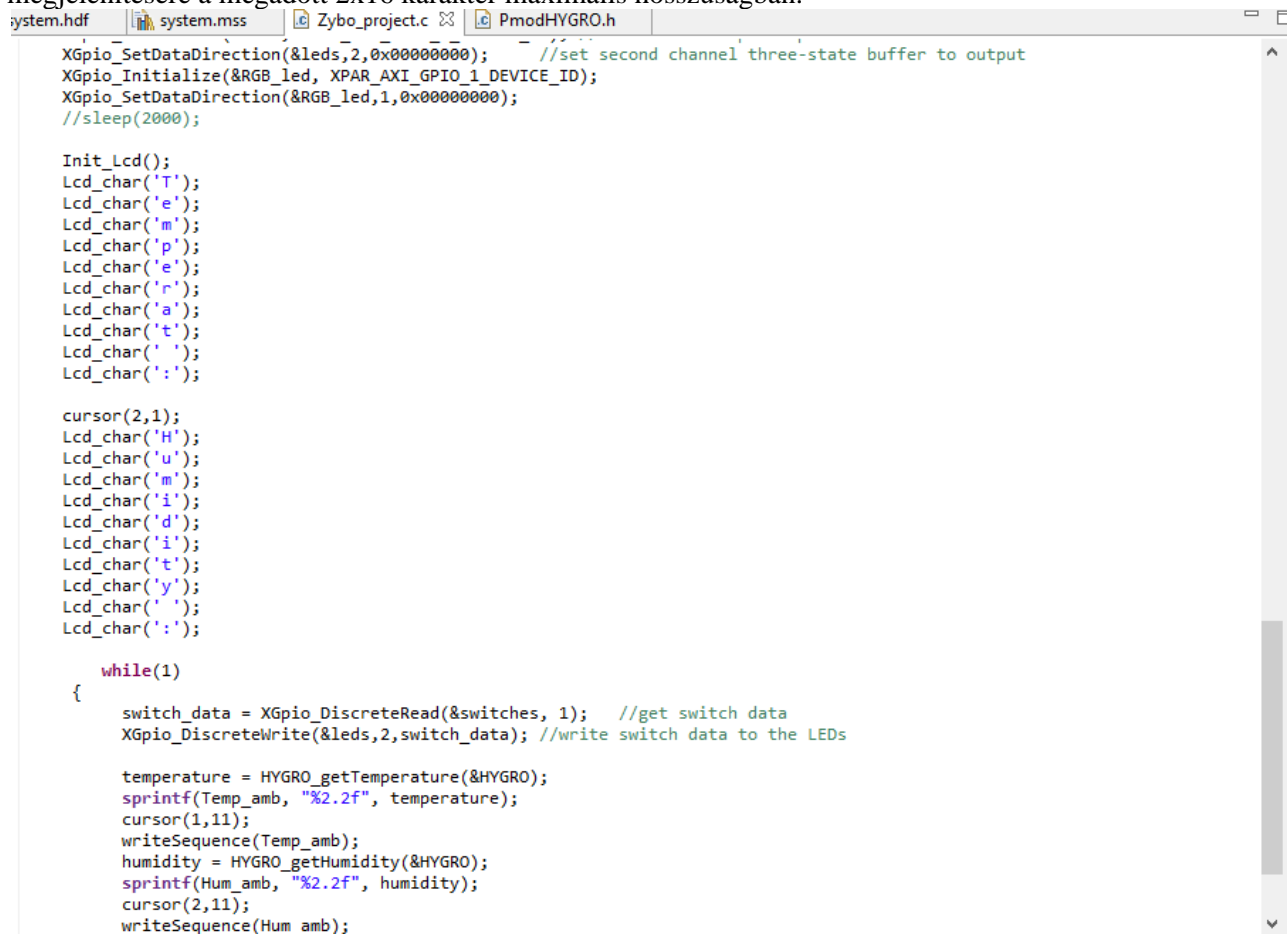
while(1)
{
    switch_data = XGpio_DiscreteRead(&switches, 1); //get switch data
    XGpio_DiscreteWrite(&leds,2,switch_data); //write switch data to the LEDs

    Lightamb = ALS_read(&Light);
    sprintf(Light_amb, "%2.2f", Lightamb);
    writeSequence(Light_amb);
    cursor(2,7);
}
```

10. Ábra – A fényerősség mérésének programozása – forráskód részlet

Ugyancsak egy rövid forráskód részlet látható a 11-es ábrán is, annak szemléltetésére, hogy miként kell programozni a hőmérséklet és a nedvesség mérését a Pmod HYGRO szenzor segítségével. Itt követhető, ahogyan a *HYGRO_getTemperature(&HYGRO)* és *HYGRO_getHumidity(&HYGRO)* mérési függvények vannak meghívva a programban, majd hogyan kell az eredményeket megfelelőképpen kijelezni.

Természetesen, előtte a Pmod LCD kijelző is be lett programozva a megfelelő szövegüzenetek megjelenítésére a megadott 2x16 karakter maximális hosszúságban.



```

system.hdf | system.mss | Zybo_project.c | PmodHYGRO.h
XGpio_SetDataDirection(&leds,2,0x00000000); //set second channel three-state buffer to output
XGpio_Initialize(&RGB_led, XPAR_AXI_GPIO_1_DEVICE_ID);
XGpio_SetDataDirection(&RGB_led,1,0x00000000);
//sleep(2000);

Init_Lcd();
Lcd_char('T');
Lcd_char('e');
Lcd_char('m');
Lcd_char('p');
Lcd_char('e');
Lcd_char('r');
Lcd_char('a');
Lcd_char('t');
Lcd_char(' ');
Lcd_char(':');

cursor(2,1);
Lcd_char('H');
Lcd_char('u');
Lcd_char('m');
Lcd_char('i');
Lcd_char('d');
Lcd_char('i');
Lcd_char('t');
Lcd_char('y');
Lcd_char(' ');
Lcd_char(':');

while(1)
{
    switch_data = XGpio_DiscreteRead(&switches, 1); //get switch data
    XGpio_DiscreteWrite(&leds,2,switch_data); //write switch data to the LEDs

    temperature = HYGRO_getTemperature(&HYGRO);
    sprintf(Temp_amb, "%2.2f", temperature);
    cursor(1,11);
    writeSequence(Temp_amb);
    humidity = HYGRO_getHumidity(&HYGRO);
    sprintf(Hum_amb, "%2.2f", humidity);
    cursor(2,11);
    writeSequence(Hum_amb);
}

```

11. Ábra – Forráskód a hőmérséklet és a nedvesség mérésére (program részlet)

ÖSSZEFOGLALÓ

A dolgozat alapvető célja bemutatni egy igen korszerű és nagy feldolgozási teljesítményű, ZyboZ7-10 rendszerre alapozott beágyazott mérési rendszer teljes tervezési és fejlesztési fázisait. Mindezen lépések alatt rendkívülien korszerű hardware- és software fejlesztői eszközök vannak felhasználva. A megoldás kiemelkedik rugalmasságával, alkalmazkodva a legkülönbözőbb fejlesztői igényeknek megfelelően, valamint nagy pontosságával. A leírt mérési rendszer kompakt- és moduláris felépítésű, viszonylag olcsón és nehézség nélkül megvalósítható, működése biztonságos és hatékony.

V. SZAKIRODALOM

- [1] <https://www.techtarget.com/iotagenda/definition/embedded-system>
- [2] <https://www.heavy.ai/technical-glossary/embedded-systems>
- [3] <https://www.javatpoint.com/embedded-system-tutorial>
- [4] <https://www.trentonsystems.com/blog/what-are-embedded-systems>
- [5] <https://externlabs.com/blogs/what-is-embedded-system-definition-types-and-use/>
- [6] <https://digilent.com/shop/zybo-z7-zynq-7000-arm-fpga-soc-development-board/>
- [7] <https://digilent.com/shop/pmod-hygro-digital-humidity-and-temperature-sensor/>
- [8] <https://digilent.com/shop/pmod-als-ambient-light-sensor/>
- [9] <https://digilent.com/reference/pmod/pmodclp/start>
- [10] <https://www.xilinx.com/products/design-tools/vivado.html>