

USB hőmérő leírása

Geresdi Attila, Halbritter András

2010. március 8.

1. Bevezetés

A mérési gyakorlat célja egy olyan eszköz megismerése, amely egy egyszerű modellként szolgál összetett számítógépvezérelt mérőrendszerekhez. Ezáltal kezelésén keresztül elsajátíthatóak az automatizált mérésvezérlés és -kiértékelés alapelvei és módszerei.

A vizsgált műszer egy a PC-hez USB-n csatlakozó hőmérő (1. ábra), amely -50°C és 125°C között mér 0.0625°C felbontással. Az eszköz lekérdezése standard soros protokollon keresztül történik. A gyakorlat során áttekintjük az utasításkészlet implementálását Microsoft Visual Studio 2005 C# környezetben, megvizsgáljuk a grafikus megjelenítés és – további felhasználásra – tárolás kérdéseit, valamint becslést adunk az eszköz különböző jellemzőire.

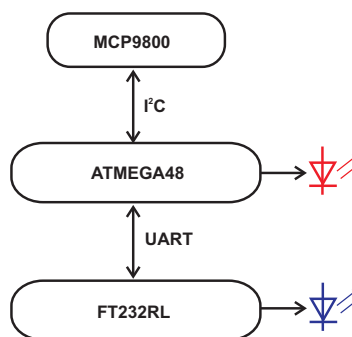


1. ábra.

2. Elektromos felépítés

Az eszköz blokkvázlata a 2. ábrán látható, három egységet különíthetünk el, melyek a nyomtatott

áramköri lapon található három integrált áramkör köré csoportosulnak.



2. ábra.

2.1. Hőmérő IC

Az érzékelő SOT-23-5 tokozású, MCP9800 típusú áramkör [1], amely tartalmaz egy félvezető hőmérsékleti szenzort, valamint elvégzi a szükséges analóg-digitális átalakítást: a hőmérsékleti értékek elméletileg a $-128^{\circ}\text{C} \dots 128^{\circ}\text{C}$ tartományba eshetnek 12 bit felbontással, amely alapján adódik az áramkör 0.0625°C felbontása. A gyakorlatban az eszköz működése csak -50°C és 125°C között garantált. Fontos megjegyezni, hogy a mérés pontossága lényegesen kisebb, mint a digitális felbontás: az adatlap szerint mintegy 1°C hibára számíthatunk a teljes működési tartományban.

Az áramkör egy kétvezetékű, ún. I²C buszon kommunikál a mikrokontrollerrel.

2.2. Mikrokontroller

Az Atmel gyártmányú, ATMEGA48 típusú eszköz [2] több feladatot lát el: egyrészt mintegy 500 ms időközönként lekérdezi a hőmérő IC-től a

hőmérsékletet, másrészt azt binárisról ASCII formátumúvá alakítja, ezzel lehetővé téve, hogy a mérésvezérlő programban közvetlenül felhasználható legyen. A kontroller parancsértelmezőként is funkcionál: a soros porton érkező karaktereket figyelve a megfelelőekre reagál.

A vezérlő további feladata, hogy hardveres hibákra lehetőség szerint reagáljon és értesítse a felhasználót. Esetünkben a hőmérő IC hibás működésére, valamint a soros porti kommunikáció hibájára egy piros LED bekapcsolásával reagál.

Az áramkör a számítógép felé soros (Universal Asynchronous Receiver-Transmitter, UART) porton kommunikál. A felhasznált RxD és TxD vonalak – a feszültség szintektől eltekintve – megfelelnek a számítógép RS-232 portjának hasonló kivezetéseinek.

2.3. Soros-USB átalakító

Mivel a hardveresen megvalósított soros port egyre ritkább a személyi számítógépeken, egy további egység beépítése vált szükségessé: az FTDI FT232RL típusú áramkör egy összetett USB device IC [3]. Mérőeszközünkben átjárót biztosít a mikrovezérlő soros portja és a számítógép USB portja között. A gyártó honlapjáról letölthető driver segítségével az eszköz teljesen transzparensen működik, az operációs rendszer számára egy újabb – virtuális – soros portként látszik (COMx).

Az eszköz a kommunikációt egy kék LED villogtatásával jelzi.

3. Parancskészlet

A mérőeszköznek az alábbi két egykarakteres utasítása van, melyet whitespace karakterrel (új-sor, space, Tab) kell lezárni. A parancsértelmezés nem érzékeny a kis- és nagybetűkre.

- **[R]**ead: A hőmérő visszaadja az aktuális mért hőmérsékletet 4 tizedesjegy felbontással.
- **[I]**nfo: A műszer szoftverének adatait adja vissza.

4. Soros kommunikáció jellemzői

A számítógépen a soros portot a következő beállítással kell használni:

- Baud Rate=38400;
- Data Bits=8;
- Stop Bits=1;
- Paritás nincs.

5. Feladatok

1. Készítsünk mérőprogramot, amely megadott időközönként beolvassa a hőmérsékleti adatokat a műszertől, ezeket grafikusán ábrázolja egy ZedGraphControl objektum segítségével, és elmenti egy szöveges file-ba! A mérőprogram felületén TextBox segítségével jelenítsük meg az aktuális beolvasott értéket, valamint legyen lehetőség a mérési időköz változtatására!
2. Egy 10 hosszúságú mozgóátlag segítségével simítsuk ki a nyers adatokat: Számoljuk ki és ábrázoljuk a legutóbbi 10 beérkezett adat átlagát! Bizonyítsuk be, hogy az átlagolás jó hatással van az adatok szórására: Számítsuk ki és ábrázoljuk a nyers, valamint az átlagolt adatok szórását! Előbbihez felhasználhatjuk az előbbi mozgóátlagban található adatokat, az utóbbihoz a legutóbbi átlagolt adatokból is hozzunk létre egy 10 elemű tömböt!
3. A hőmérő felmelegítésével, majd lehűtésével becsüljük meg az eszköz időállandóját! Ehhez feltehetjük, hogy a hőmérő hőfokkülönbsége a környezethez képest $\Delta T(t) = \Delta T_0 e^{-t/\tau}$ függvény szerint változik. Különbözik-e az eredmény, ha a nyers, illetve a simított adatsorral dolgozunk?

6. Jegyzőkönyv

A jegyzőkönyv szerepe – hasonlóan a korábbi laborokhoz – a mérési feladat megoldásának dokumentálása, valamint a saját eredmények bemu-

tatása. Ennek megfelelően tartalmaznia kell a vizsgált jelenség vagy feladat rövid összefoglalását, részletesen be kell mutatnia a méréshez felhasznált eszközöket, mérőrendszert. A jegyzőkönyv fontos része az adatok kiértékelése és diszkussziója, azaz a nyers adatok további analízise is.

A jegyzőkönyvhöz – külön, betömörítve – csatoljuk a program forráskódját!

Hivatkozások

- [1] <http://www.microchip.com>
- [2] <http://www.atmel.com>
- [3] <http://www.ftdichip.com>