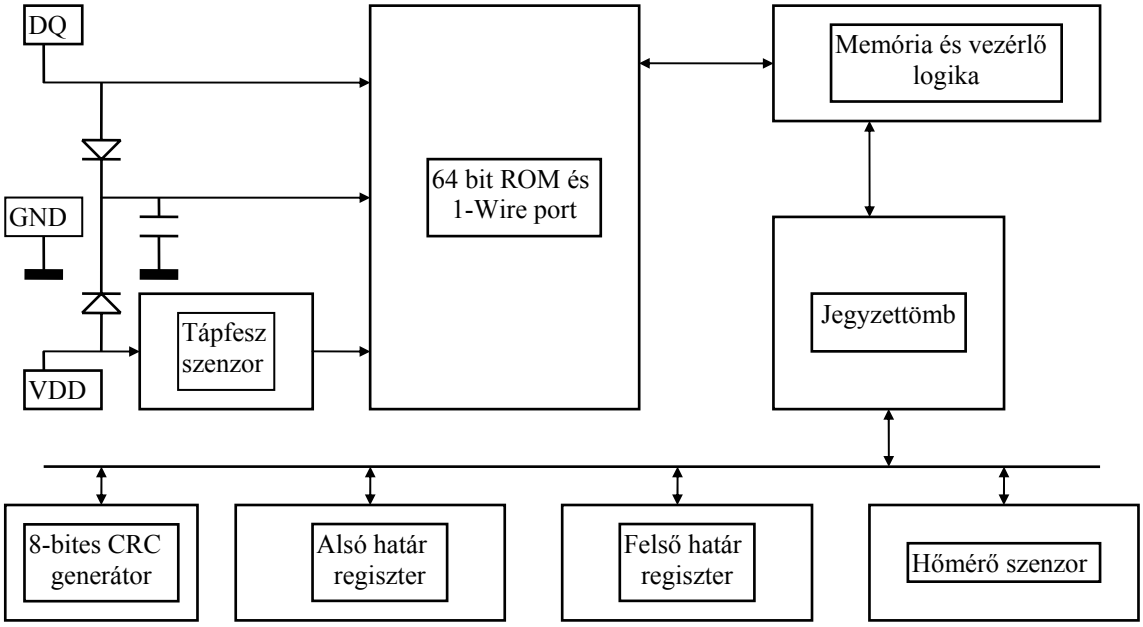
1. DS1820 hőmérő chip leírása

A DS1820 hőmérő chip egy 3 kivezetéssel rendelkező PR35 tokozású eszköz. A főbb jellemzői a következők:

• 1- Wire™ buszra csatlakozik, így csak egy port szükséges a kommunikációhoz.
• nincs szükség az alkalmazásához más külső komponensre.
• ha a működéshez szükséges 1mA áramot fel tudja venni az adatvonalról, akkor elég csak két vezetéket kiépíteni egy eszközhöz.
• hőmérséklettartomány: -55°C-tól +125°C-ig, 9 bites felbontásban.
• a hőmérséklet konverziós idő 500ms.
• a felhasználó által megadható riasztási határok, amelyek a tápfeszültség kikapcsolása után is megmaradnak.

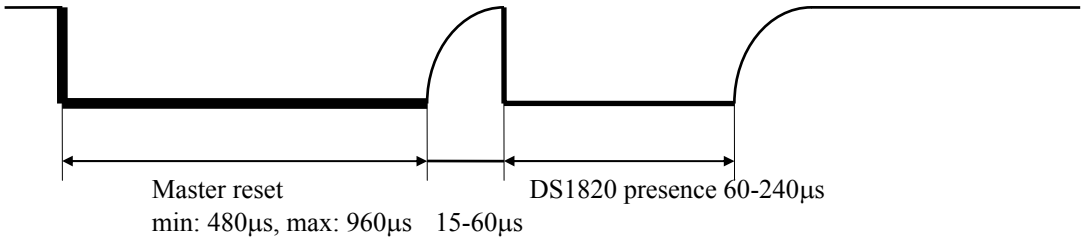
A chip belső felépítése:



A DS1820 főbb részei a fenti ábrán láthatók. A kommunikáció az eszköz és a master egység között a DQ lábon keresztül történik, sorosan az 1-Wire buszon keresztül. Mindig az LSB bit lép ki először. A kommunikációs protokoll a következő:

• Inicializálás
• ROM utasítás
• Memória/vezérlő utasítás
• Adatátvitel

Inicializálás: mielőtt a buszon adatokat vinnénk át, inicializálni kell az alárendelt eszközt. Az inicializálás során a master egység reset jelet ad a buszra, amire a slave egység „jelenlét” (presence) jelet ad ki. Ebből tudja a vezérlő egység, hogy a slave egység készen áll az adatforgalomra.



A presence jel detektálása után a master egységnek az itt leírt ROM utasítás egyikét kell kiadnia (az utasítások 8 bit hosszúak):

1. Read ROM (ROM olvasás) [33h] : ez az utasítás lehetővé teszi, hogy a master egység kiolvassa a DS1820-as 8 bites család-kódját (10h), a 48 bit hosszú egyedi azonosítót (gyári szám) és az előző 56 bitre vonatkozó CRC kódot (szintén 8 bites). Ez az utasítás csak akkor ad „értelmes" eredményt vissza, ha egyetlen DS1820 csatlakozik a buszra!

2. Match ROM (ROM kijelölés) [55h] : az utasítás követő (a master által kiadott) 64 bites adatfolyam kijelöli az aktív eszközt, ha több eszköz csatlakozik a buszra. Az az eszköz lesz aktív, amelyik ROM-jának tartalma megegyezik a kiküldött adatfolyammal. A többi eszköz a következő reset impulzusig inaktív marad.

3. Skip ROM (ROM „átugrás") [CCh] : ez az utasítás időt takarít meg, ha egy eszköz csatlakozik a buszra, rögtön követheti a memória/vezérlő utasítás.

4. Search ROM (ROM keresés) [F0h] : az utasítás lehetővé teszi, hogy több egység esetén is lekérdezze a master a ROM tartalmakat (eszközök azonosítása a buszon).
A ROM keresés a következő 3 lépésből álló eljárás ismétlése:
a, az összes egység kiteszi a buszra a ROM kódjának 0.bitjét, ezt a master beolvassa.
b, a következő olvasási lépésben az összes egység kiteszi a buszra a 0.bit invertáltját, ezt a master beolvassa.
c, a master egység meghatározza az érvényes 0.bitet a beolvasott két bit alapján, és ezt kiírja a buszra. Ezzel kijelöli azokat az eszközöket, amelyekre a további lépések vonatkoznak. Az érvényes bit meghatározása a következő eljárás segítségével történik:

0 0 - a buszon lévő eszközök adott pozíción lévő bitje eltérő, ekkor a master eldönti, hogy melyik értéket tekinti érvényesnek, azt írja ki a buszra, megjegyezve azt, hogy ezen a bitpozíción elágazás van.
0 1 - az összes buszon lévő eszköz adott pozíciójú bitje 0.
1 0 - az összes buszon lévő eszköz adott pozíciójú bitje 1.
1 1 - nincs eszköz a buszon.

A fenti 3 lépés ismétlésével meghatározható egy eszköz ROM kódja. Ezután a master újabb „Search ROM" parancsot ad ki, majd figyelembe véve az elágazásokat, a fenti eljárás ismétlésével megtudja határozni az összes eszköz ROM-kódját.

5. Alarm Search (riasztás keresés) [ECh] : erre az utasításra a eszköz akkor válaszol, ha az utolsó mérésnél a riasztási feltétel igaz. A riasztási feltétel akkor igaz, ha a mért hőmérséklet a TH regiszterben megadottnál nagyobb, vagy a TL regiszterben megadottnál kisebb.

A ROM parancsok után az eszköznek vezérlőutasítást kell küldeni. Az utasítások ismertetése előtt azonban nézzük meg a *scratchpad* (jegyzetömb) felépítését, és a benne lévő byte-ok értelmezését:

Byte	Scratchpad	E^2RAM
0	Hőmérséklet alsó byte (TEMP_L)	-
1	Hőmérséklet felső byte (TEMP_H)	-
2	TH regiszter (felső határ)	TH regiszter
3	TL regiszter (alsó határ)	TL regiszter
4	fenntartva	-
5	fenntartva	-
6	COUNT REMAIN (maradék)	-
7	COUNT PER °C (hányados)	-
8	CRC kód	-

A TEMP_L és TEMP_H regiszterek tartalma adja meg a mért hőmérsékletet. Ez csak akkor érvényes, ha az eszköz által a 0-7 byte-ra generált CRC kód megegyezik a master eszköz által (a megadott képlet szerint kiszámított) CRC kóddal (lásd később). A hőmérsékletet előjeles kettes komplemens számként kell értelmezni, úgy hogy TEMP_H összes bitje az előjelbit, TEMP_L 0.bitje pedig a 1/2 fokos helyi érték. A következő két regiszter az alsó és felső riasztási érték beállítására szolgál. Ez az érték egy utasítással átmásolható egy E^2RAM-ba, ami lehetővé teszi, hogy az eszköz a tápfeszültség kikapcsolása után is emlékezzen ezekre (lásd később). A következő két regiszter fenntartott, értékük olvasáskor FFh. A következő két regiszter lehetővé teszi a hőmérséklet pontosabb meghatározását (jobb felbontás), a következő képlet alapján:

Temp = TEMP_L,H - 0,25°C + (COUNT PER °C - COUNT REMAIN)/COUNT PER °C

Az utolsó byte az előző 8 byte-ból generált CRC kód.

A vezérlő byte-ok tehát a következők:

1. Write Scratchpad („jegyzettömb írása") [4Eh] : az utasítást követő két byte az eszköz TL és TH regiszterének tartalma. Ezzel beállíthatók a riasztási hőmérséklet értékek. Mivel ezek 8 bites regiszterek, ezért csak egész hőmérsékleti értékek adhatók meg riasztási értekként (8 bites, előjeles kettes komplemens számként értelmezve).

2. Read Scratchpad („jegyzettömb olvasása") [BEh] : a *scratchpad* tartalmának beolvasása.

3. Copy Scratchpad („jegyzettömb másolása") [48h] : a *scratchpad* TH és TL regiszterének bemásolása az E^2RAM regiszterekbe, így ez az érték a tápfeszültség kikapcsolása után is megmarad.

4. Convert T („hőmérséklet konvertálás") [44h] : ez a parancs indítja el a hőmérséklet konverzióját. Ha ezalatt olvassuk a buszt 0-t kapunk, ha kész a konverzió, akkor a buszról 1 olvasható. A konverzió ideje kb. 500 ms.

5. Recall E2 („E^2RAM előhívása") [B8h] : ez az utasítás bemásolja a *scratchpad* TH és TL regiszterébe az E^2RAM-ban eltárolt riasztási hőmérséklet értéket. Ez a művelet egyébként

lejátszódik minden tápfeszültség rákapcsoláskor, így az eszköz mindig tud reagálni az „Alarm Search” utasításra.

6. Read Power Supply („tápfeszültség olvasás”) [B4] : az utasítást követő olvasási ciklusban, ha az eszköz 1-t helyez a buszra akkor külső tápfeszültségről (+5V) működik, ha 0-t akkor az adatvonalról nyeri az energiát és azt a belső (parazita) kapacitásban tárolja. Ha az eszközt az adatvonalról kívánjuk energiával ellátni, akkor azt megtehetjük úgy, hogy a master elemnél egy FET 5V-ot kapcsol az adatvonalra 10µs-ra, ezalatt feltöltődik a belső kapacitás. A feltöltést érdemes megtenni minden konverzió előtt. Ha adatvonalról látjuk el az IC-t, akkor a VCC vezetéket testpontra kell kötni. A módszer 100°C-nál magasabb hőmérséklet mérésénél nem használható.

Az egyes utasítások részletes idődiagramját lásd a „ds1820.pdf” file-ban.

1.1 A CRC kód generálása

A CRC (Cyclic Redundancy Check) kódok, szemben az egyszerű check-sum típusú hibaellenőrző kódokkal lehetővé teszik a több bit hibák detektálását is. Az alapgondolat az, hogy az üzenetet (bitfolyam) tekintsük egy óriási nagy bináris számnak, amit elosztunk egy ismert, fix bináris számmal, és az osztás eredményéből képezzünk check sum-ot. A CRC kódok a polinom-kódok családjába tartoznak.

A polinom-kódok azon alapulnak, hogy a bitsorozatokat polinomok reprezentációjának tekintjük, melyekben csupán 0 és 1 együtthatók szerepelnek. Egy *k* bites keretet tekintsünk egy *k* tagú polinom együtthatóinak x^{k-1}-től x^0-ig. Az ilyen polinomot *k-1*-ed fokú polinomnak nevezzük. A legnagyobb helyi értékű bit az x^{k-1} együtthatója, a legkisebb az x^0 együtthatója. A polinom aritmetika modulo 2 végzendő az algebrai terek elmélete szerint. Nincs átvitel az összeadásnál és kivonásnál, melyek a KIZÁRÓ VAGY művelettel azonosak. Az osztást ugyanúgy kell elvégezni, mint a binárist, kivéve, hogy a kivonás modulo 2 értendő. Az osztó megvan az osztandóban, ha az osztandó ugyanannyi bitet tartalmaz, mint az osztó.

Amikor polinomkódot alkalmazunk, az adónak és a vevőnek meg kell egyeznie egy **generátor polinomban.** Jelölje ezt G(x). A generátor legalsó és legfelső bitjének 1-nek kell lennie. Ahhoz, hogy a egy M(x) polinomnak megfelelő *m* bites keret ellenőrző összegét kiszámoljuk, a keretnek hosszabbnak kell lennie, mint a generátor polinom. Az ötlet az, hogy úgy fűzzünk ellenőrző összeget a kerethez, hogy az így kapott keret által reprezentált polinom osztható legyen G(x)-szel. Amikor a vevő megkapja a keretet, megpróbálja elosztani G(x)-szel, ha van maradék, akkor hiba volt az átvitel során.

Az ellenőrző összeg kiszámításának a menete a következő:
1. Legyen *r* G(X) fokszáma. Fűzzünk *r* darab 0 bitet a keret alacsony helyi értékű végéhez, így az *m+r* bitet fog tartalmazni, és az $x^rM(x)$ polinomot fogja reprezentálni.
2. Osszuk el a $x^rM(x)$-hez tartozó bitsorozatot a G(x)-hez tartozó bitsorozattal modulo 2.
3. Vonjuk ki a maradékot (mely mindig r vagy kevesebb bitet tartalmaz) az x M(x)-hez tartozó bitsorozatból modulo 2-es kivonással. Az eredmény az ellenőrző összeggel ellátott, továbbítandó keret.

A DS1820-as IC-ben két helyen alkalmaznak CRC kódot. Első esetben a ROM-kód 7 byte-jából képez a chip CRC kódot, ez lesz a ROM-kód 8. byte-ja. Második esetben a *scratch-pad* 8 byte-jából képez egy CRC kódot, ez lesz a *scratch-pad* 9. byte-ja. A CRC kód és az adatok helyességének ellenőrzése a PC-ben történik