
9.8.1 KEELOQ IFF

A HCS410-et IFF válaszadóként „automatikus” azonosításra lehet használni. IFF módban egy megfelelő kulcshitelesítési képességgel rendelkeznek, mielőtt kikapcsolná a biztonsági rendszert a dekódoló. Amikor egy slusszkulcsot behelyezzük a kulcslyukba, akkor induktívan lekérdezi a dekóder a kódolót (kulcsérvényesítőt) mielőtt kikapcsolná az immobilizert.

Az IFF jel érvényesítés abból áll, hogy elküld egy véletlen felhívást a dekóder a kódolónak. Mire az egy válaszjelet generál a felhívásra és visszaküldi a választ a dekódernek. Azután a dekóder kiszámítja a várt válaszjelet és összehasonlítja a kódolótól kapott válaszjellel. Ha a jelek egyeznek, akkor az visszajelzést érvényesnek találja és a dekóder a megfelelő feladatot végrehajtja.

A HCS410 16 és 32 bites IFF-re is alkalmas. Továbbá a két kódoló algoritmus is van, amit arra használ, hogy válaszjelet generáljon a felhívásra. Sőt akár még két kódoló kulcsot is használhat.

IFF módban a HCS410 egy parancsra vár a bázis állomástól és arra válaszol. Ez lehet egy olvasás/írás a felhasználói EEPROM-tól vagy pedig egy IFF felhívás válasz. Egy adott 16/32 bites kihívás egy egyedülálló 16/32 bites válaszjelet generál, ami az IFF kulcson és IFF algoritmuson alapszik.

9.8.2 Készülék működése

A HCS410 normál ugró kódos adóként egy vagy két IFF kulccsal vagy tisztán IFF válaszjel adóként két IFF kulccsal funkcionálhat. Amikor ugró kódos adóként használjuk, akkor csak gombok és RF áramkör hozzáadása szükséges, hogy működhessen. Ha válaszjeladó funkciót is el akar látni az adó, akkor csak egy tekercsre és két kondenzátorra van szüksége.

9.8.2.1 Ugró kódos üzemmód (CH mód)

Erre az üzemmódra ugyanaz jellemző, amit már ismerttettem az előző kódoló típusoknál, azzal a legfőbb különbséggel, hogy itt 69 bites az üzenet hossza, mely egy 32 bites titkosított és egy 37 bites állandó részből áll.

9.8.2.2 IFF MÓD

Az IFF mód lehetővé teszi a dekódoló számára, hogy egy IFF érvényesítést hajtson végre, ami által írhat és olvashat a felhasználói EEPROM-ból. A dekódoló mindenkori működése egy utasításkód elküldéséből és a HCS410-től kapott válaszból áll.

Kétféle IFF mód létezik: IFF1 és IFF2. Az IFF1 csak egy IFF kulcs használatát engedélyezi, amíg az IFF2 módban már két kulcs használható. Amikor az IFF2 mód engedélyezve van, akkor a seed átvitelek le vannak tiltva. Ez lehetővé teszi, hogy a HCS410 egy IFF válaszjel adóként használjuk. Ekkor a kapcsolattartáshoz nem szükséges mágneses tér jelenléte. A dekódolót a HCS410-hez, az adat vonalán keresztül csatlakoztathatjuk. A HCS410 ekkor az adat vonalon keresztül kapja a tápfeszültséget, mint normál válaszjeladó üzemmódban. A kommunikáció módja ekkor megegyezik a válaszjeladó üzemmódban használtéval.

10 A KEELOQ DEKÓDOLÓK ÜZENETDEKÓDOLÓ ELJÁRÁSA

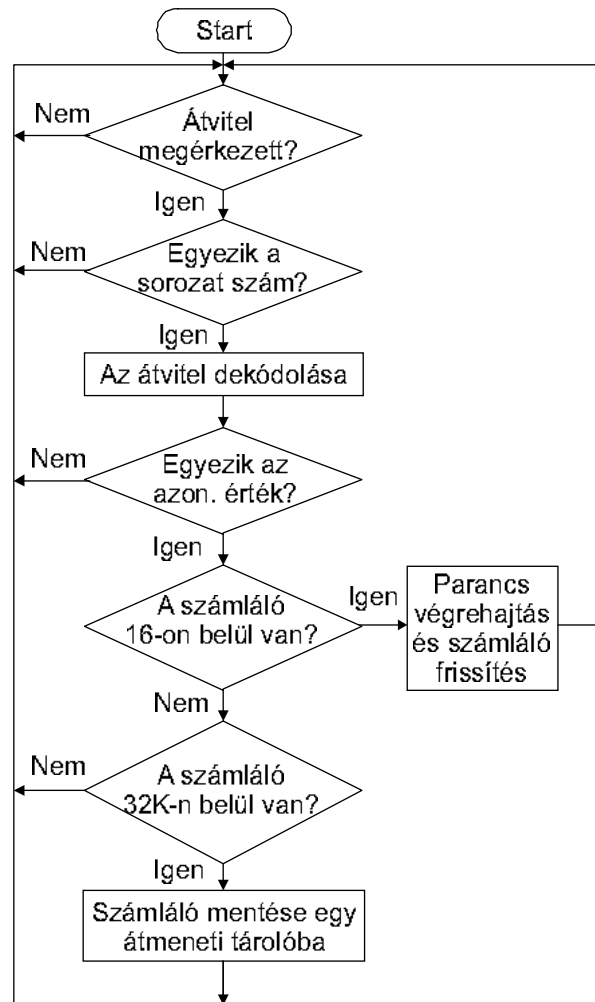
Már ismeretes a számunkra, hogy normál és titkos üzemmódban milyen kulcskészítési algoritmusokat alkalmazhatnak a HCSXXX eszközök. Az üzenetek dekódolása előtt és után úgynevezett érvényesítési eljárásokat hajt végre a berendezés. Ez az jelenti, hogy megvizsgálja, hogy a sorozat szám, az azonosító érték megegyezik-e a memóriájában tárolt értékekkel, és hogy a szinkronizáló számláló szekvenciális-e. Az első kettő az úgynevezett érvényesítési folyamatokhoz tartozik, míg a harmadik (számláló) pedig a szinkronizáláshoz. Ez utóbbira azért van szükségünk, mert gondoljunk bele, hogy mi történik abban az esetben, ha egy adót úgy működtetünk, hogy a vevő hatáskörén kívül van. Ekkor az adó folyamatosan küldi a jeleket, növekvő számláló értékek mellett, ami azt jelenti, hogy nem lesz szekvenciális a szinkronizáló érték (nem egy lesz a különbség az elküldött számláló értéke és a dekódolóban tárolt érték között). Először nézzük meg, hogy hogyan történik az üzenetek érvényesítése, majd utána térünk vissza a szinkronizálásra.

10.1 ÉRVÉNYESÍTÉS

Amikor egy komplett üzenetet vesz a dekódoló a kódolótól, akkor először feltétlenül szükséges a kívánt feladat végrehajtása előtt az érvényesség ellenőrzése (**28. ábra**). Az érvényesítés a következő lépésekből áll:

1. Fogadja a beérkezett üzenetet.
2. Ellenőrizni kell a sorozat számot a memóriában tárolt kódoló sorozat számával.
3. Dekódolni kell a kapott üzenetet.

4. Össze kell hasonlítani a dekódolt üzenetből vett azonosító értéket a memóriában tárolt értékkel.
5. Ellenőrizni kell a szinkronizáló értéket, hogy az úgynevezett újraszinkronizáló tartományon belül van-e.
6. Ellenőrizni kell, hogy a szinkronizáló érték az úgynevezett automatikus újraszinkronizálás tartományán belül van-e. Ha nem, akkor újra kell szinkronizálni.
7. Ha újraszinkronizálás szükséges, akkor várni kell a második átvitelt a kódolótól a közvetlen következő szinkronizáló értékkel.
8. Frissíteni kell az EEPROM-ban tárolt szinkronizáló értéket.
9. Állítani kell a megfelelő kimeneteket.



28. Ábra. Érvényesítés folyamata

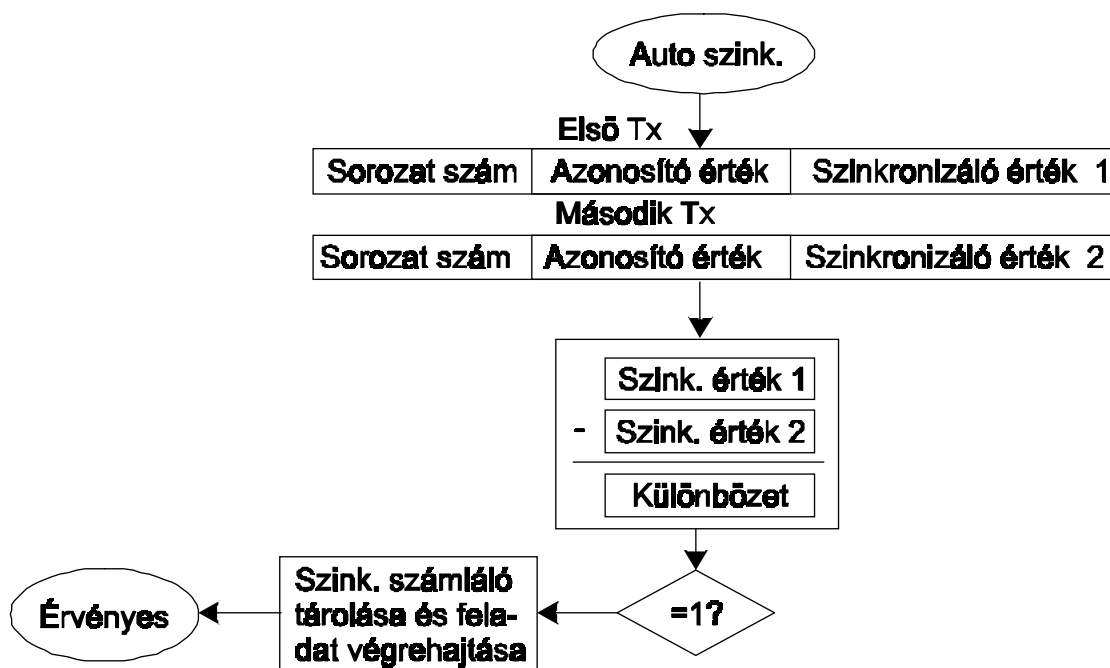
10.2 SZINKRONIZÁLÁS

A KEELOQ technológia jellemzője, hogy egy fejlett szinkronizációs technikával rendelkezik, amelyhez nincs szükség számításokra és a következő kódok tárolására. Abban az esetben, ha az adó számlálójának az értéke – amelyet a dekódoló megkap és dekódol – és a

dekódoló memóriájában tárolt számláló érték között az eltérést vizsgáljuk, akkor három különféle eshetőséget állapíthatunk meg:

1. Egy a különbség
2. 16-nál kevesebb az eltérés
3. 32K-nál kevesebb az eltérés
4. 32K-nál több az eltérés

1. Egy a különbség: Ez az eset akkor áll fenn, ha normál üzemmódban működik a berendezés, vagyis a két üzenet pontosan egymást követi (29. ábra).

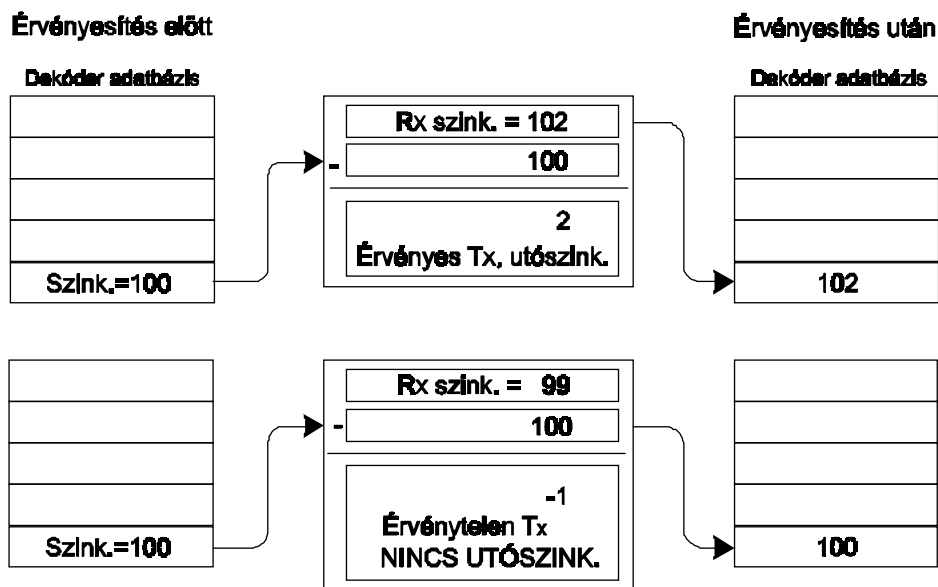


29. Ábra. Automatikus szinkronizáció

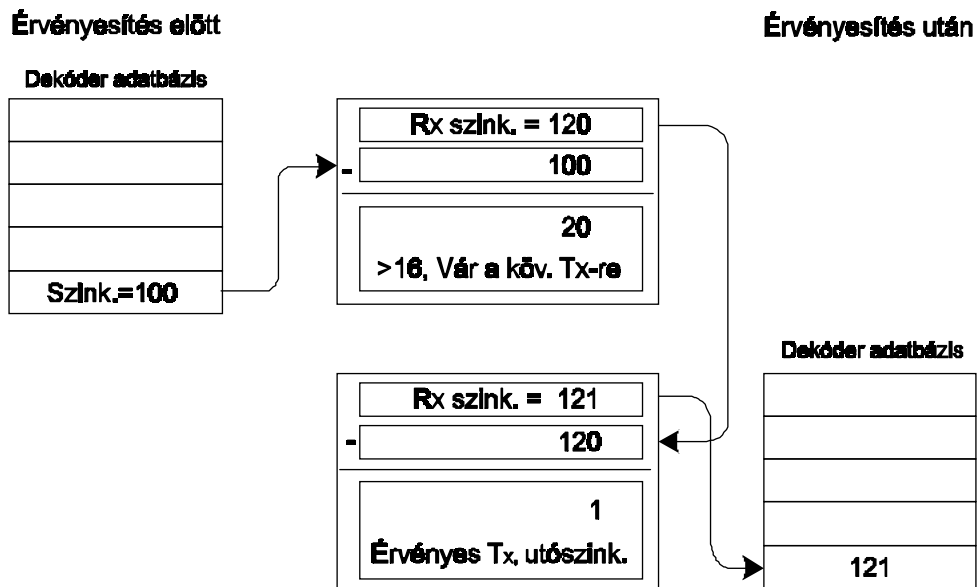
2. 16-nál kevesebb az eltérés: Ekkor az úgynevezett újraszinkronizáló tartományon belül van még a különbség, ami a tartomány nevéből is adódóan azt jelenti, hogy automatikusan

újraszinkronizálódik a dekódoló. Az újonnan kapott érték tárolódik, és a parancs végrehajtódik (**30. ábra**).

3. 32K-nál kevesebb az eltérés: Ha ez az érték 16-nál nagyobb és 32K-nál kisebb, abban az esetben a szinkronizáló szám átmenetileg tárolódik és a dekódoló vár egy újabb átvitelre. Amikor megérkezik a következő átvitel, akkor összehasonlítja az átmenetileg tárolt számmal, és ha azok egymást követik (1. eset áll fenn) akkor azt jelenti, hogy megtörtént a szinkronizáció és az új érték tárolódik, majd az utasítást végrehajtja a berendezés (**31. ábra**).
4. 32K-nál több az eltérés: Ebben az esetben az történik, hogy meghaladja a különbség a 32K-t, ami pedig azt jelenti, hogy nem fog működni a rendszerünk és ez esetben újra tanítást kell alkalmazni.



30. Ábra. Automatikusan újraszinkronizáció és elutasítás



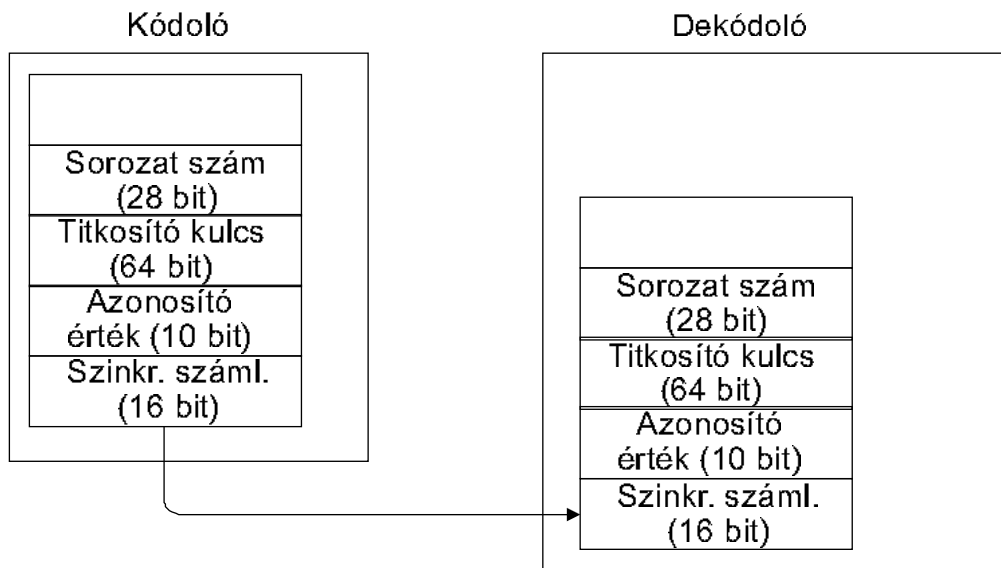
31. Ábra. Újra szinkronizáció

Abban az esetben ha a dekódolt átvitelből megkapott számláló értéke kisebb, mint a memóriában tárolt szám, akkor elutasítja az üzenetet, vagyis érvénytelen lesz az átvitel (30. ábra alsó része).

11 TANÍTÁS, TANULÁS

Már eddig sokat hallottunk az előző részekben olyanról, hogy tanítás, tanulás. Miért is van szükség tulajdonképpen ezekre az eljárásokra? Nos, ha a gyártó arra szánja el magát, hogy a KEELOQ technológiára alapuló biztonsági berendezéseket készítsen, és e célból megvásárolja magának a szükséges alkatrészeket, akkor nem elég, ha egy megadott kapcsolási rajz alapján összerakja az áramkört, hanem ezeket az eszközöket először fel kell programozni, vagy úgy is

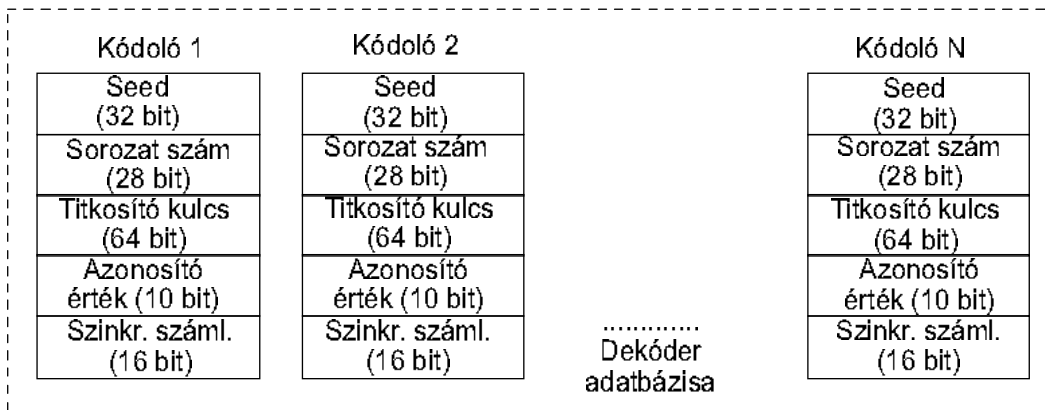
nevezhetném, hogy meg kell ismertetni egymással a kódolókat és dekódolókat. Ezt nevezzük tanításnak, vagy ha a dekódoló felől nézzük, tanulásnak (**32. ábra**).



32. Ábra. A kódoló és dekódoló memóriája közötti összefüggés

Azért használtam a többes számot, mivel arról van szó, hogy egy dekódoló több kódolónak a parancsait tudja végrehajtani, értelmezni. Vagyis egyszerre több kódolónak az adatait képes a memóriájában letárolni (**33. ábra**). Ez dekódoló típusonként változik, hogy hány adót tud felismerni. Másrészt gondoljunk bele, hogy ha már rendelkezésünkre áll egy kész működő berendezés (kódoló, dekódoló felprogramozva; tanítás megtörtént; üzemszerűen működik), mi történik abban az esetben, ha elveszítjük az adó egységünket. Induljunk ki először abból, hogy egy adónk és egy vevőnk van. Ebben az esetben, ha elvesztettük az adónkat (kódolót), akkor veszünk egy másik adót, amelyet fel kell programoznunk ugyanazzal a gyártó kóddal, amivel a már meglévő vevőnk (dekódoló) rendelkezik és ha ez megtörtént, azután meg kell tanítanunk – meg kell ismertetnünk – a dekódoló egységünket az új kódoló adataival. Vegyük most azt a lehetőséget, ha több adónk van egy vevőnk. Ekkor már kicsit nehezebb a helyzetünk, mivel a dekódolónk úgy működik, hogy ha pl. csak 4 adó adatait képes eltárolni és mi elvesztjük az

egyiket, utána veszünk helyette egy másikat és ennek az adatait akarjuk megtanítani vele. Ebben az esetben a dekódoló véletlenszerűen fogja kiválasztja az egyik előző – esetleg még meglévő – adó adatait és kicseréli velük az új kódoló paramétereit. Lehetséges, hogy egy olyan adót tesz „hatástalanná”, amelyiket nem hagytuk el. Ekkor az eljárás az, hogy az egész rendszerünket újra kell programoznunk, és sorba minden adót fel kell ismertetni a vevővel. Ezek a felismertetések vagy inkább tanítások lehetnek úgynevezett normál vagy titkos tanítások.



33. Ábra. A dekódoló memóriája

A tanításnak kétféle folyamata az ismeretes. Az egyik a normál tanítás, a másik, a titkos tanítás. A következőekben tekintsük át ezeket részletesebben.

11.1 NORMÁL TANÍTÁS

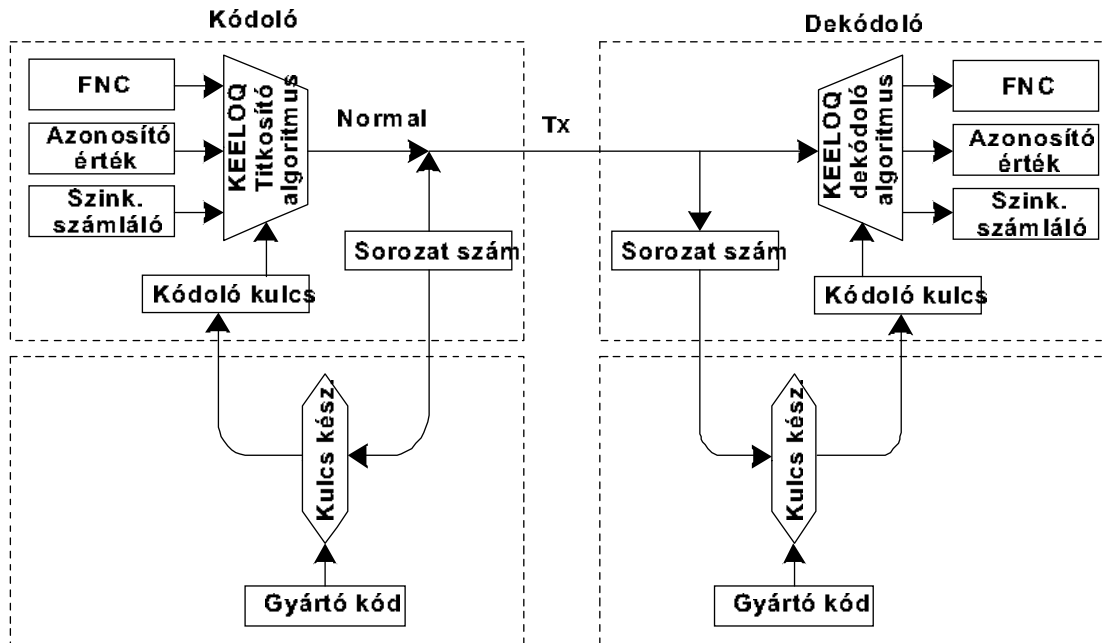
A normál tanítási folyamathoz a kódoló, a kódolt üzenet elkészítéséhez a titkosító kulcsot a normál kulcskészítő algoritmus (sorozat számból és gyártó kódból) alapján készíti el, mint ahogy az ábrán is láthatjuk. Ebben az esetben a kódoló az ugró kódos rész helyén mindig a normál, kódolt információkat (ugró kódot) küldi el.

A következőekben a dekódoló felől vizsgáljuk meg a tanulás folyamatát, abból indulunk ki, hogy a kódolónk elkészítette és már el is küldte az üzenetet. (A leírás a **34. ábra** értelmezése, magyarázata is egyben.)

A tanulás folyamata:

- Figyelni a tanító bemenetet
 - Bekapcsolni a jelző LED-et, amikor a tanítási mód aktív
- Várni a kódoló T_x jelére
- Ha megérkezett a jel, létrehozni a titkosító kulcsot a sorozat szám felhasználásával
- Dekódolni a T_x jel ugró kódos részét az előbb generált titkosító kulccsal
- Ellenőrizni az azonosító számot
 - Jellemző, hogy az azonosító számot egyenlővé teszik a sorozat számmal – ha a kapott azonosító szám megegyezik a sorozat szám alsó bitjeivel, akkor elfogadhatjuk:
 - Ha az előbbi teljesül, megállapíthatjuk, hogy a kiszámított titkosító kulcs megfelelő
- Ideiglenesen tárolhatóak a dekódolásból kapott információk az EEPROM-ban
- Ki kell kapcsolni a LED-et.
- Ezek után várni kell a második T_x jelére a dekódolónak
- Fogadni és dekódolni kell a második T_x jelet, az első T_x jelből kapott titkosító kulccsal.
- Ellenőrizni a második T_x jelnek az azonosító számát.
- Össze kell hasonlítani a második T_x jelben lévő szinkronizáló számláló értéket az első T_x jelben lévő szinkronizáló számláló értékkel. Egy számban kell, hogy eltérjenek (szekvenciális).
- Tárolni kell a kódoló adatait a dekódoló adatbázisába.
- Villogtatni kell a LED-et.

Az előbbieken láhattuk, hogy a tanítási folyamat során rendkívül sok ellenőrzésre, egyeztetésre van szükség. Ez esetekben el kell utasítani a tanulási eljárást, és előlről kell kezdeni az egész folyamatot.



34. Ábra. Normál tanulás (tanítás)

11.2 TITKOS TANÍTÁS

Tekintsük át egy kicsit a következő gondolatokat:

Mi történik, hogyha mi...

- kivesszük a normális átvitelből a kulcskészítési információkat?
- egy speciális nyomógomb kombinációval speciális kulcskészítő információt küldünk?

Akkor...

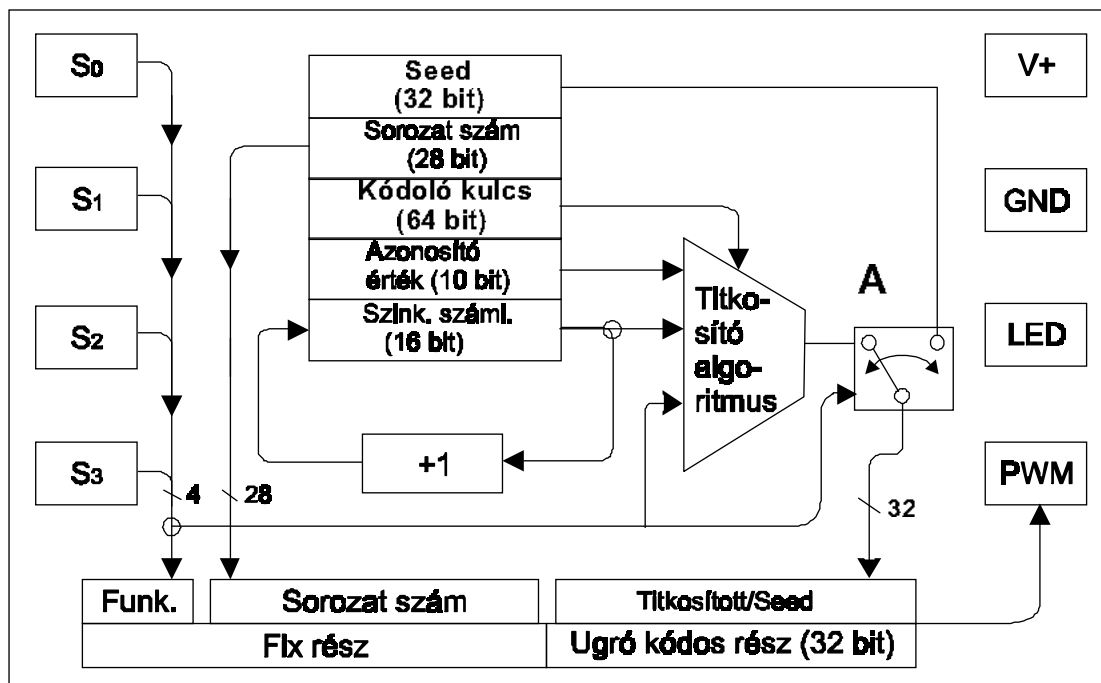
- a támadók nem tudják felhasználni hasznos információnak az átvitelt.
- fizikai behatás kell a kódoló kulcs készítő információiért.

Eredmény, hogy hatékonyan...

- csökkenthetjük a gyári kód hangsúlyát a rendszer védettsége érdekében.
- növelhetjük a biztonságot.

Vagyis ezt a módszert azért találták ki a tervezők, hogy növeljék a rendszer biztonságát, amihez a vevőnek ismernie kell a titkos tanítási eljárást. Ehhez a folyamathoz a kódoló EEPROM-jában tárolt seed értékre van szükség, amely csak egy bizonyos nyomógomb kombináció hatására küld el. Ez a bizonyos kombináció kódoló típusonként változik. A normál kulcskészítő módszer helyett a titkosító kulcskészítő algoritmust használják az eszközök ekkor, ami azt jelenti, hogy egy kreált titkosító kulcs az algoritmus forrása (seed érték). Nincs semmiféle matematikai összefüggés a sorozat szám és a seed érték között. Csak a tanításkor kerül átvitelre ez az érték.

Az üzenet ugró kódos részén ilyenkor a seed értéket küldi el a kódoló először, hogy ebből a vevő elkészítse a titkosító kulcsot, majd a következő normál átvitel során, a kódolt információkat dekódolja az elkészített kulcs segítségével. Ezt a számot az EEPROM-jából veszi. Az elmondottakat grafikusán is végig követhetjük a **35. ábrán**. Láthatjuk, hogy az „A” virtuális kapcsoló szabályozza, hogy a seed érték kerüljön átvitelre, vagy a titkosító algoritmus alapján készített információ.

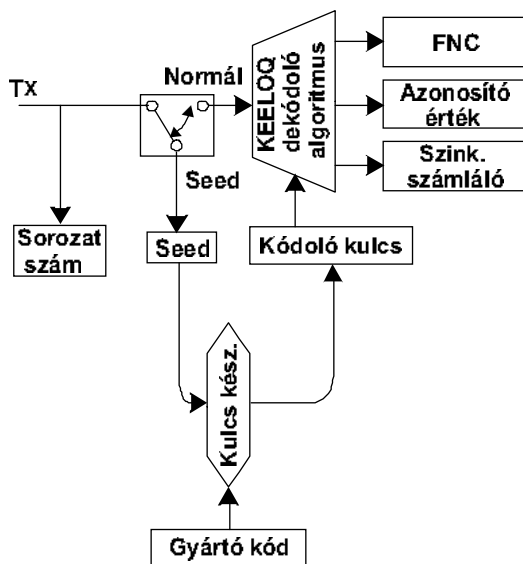


35. Ábra. Titkos tanulás során az üzenetkészítés folyamata

A tanulás folyamata (36. ábra):

Itt is abból indulunk ki, hogy a kódoló már elkészítette és elküldte az üzenetet.

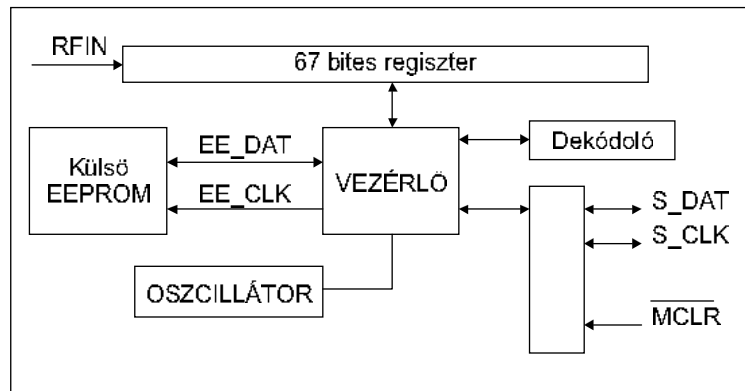
- Figyelni a tanító bemenetet
 - Bekapcsolni a jelző LED-et, amikor a tanítási mód aktív
- Várni a kódoló T_x jelére, amelyik a seed-et tartalmazza.
- A SEED felhasználásával elkészíteni a kódoló kulcsot. Ekkor eltárolja a sorozat számot átmenetileg.
- Ki lehet kapcsolni a LED-et
- Várni az ugró kódos T_x jelre.
- Dekódolni az ugró kódos részét.
- Összehasonlítani a sorozat számot és az azonosító értéket.
- Tárolni a szinkronizáló számláló értéket a kódoló adatainak egy részével együtt.
- Villogtatni kell a LED-et. Ezzel jelezni, hogy a tanulás sikeres.



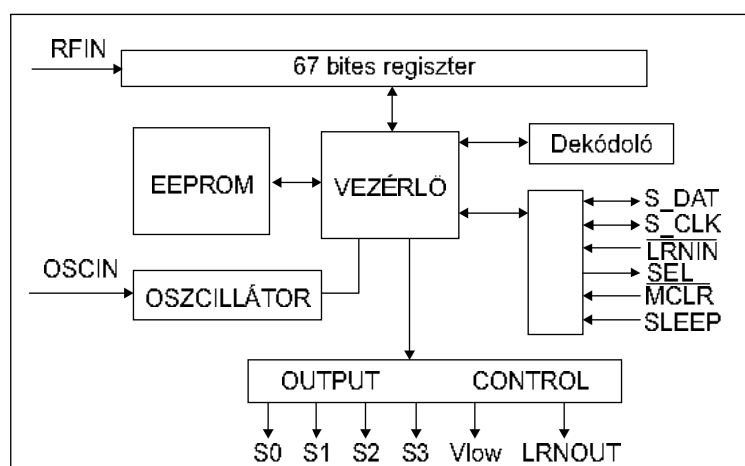
36. Ábra. A titkos tanulás folyamata

12 A KEELOQ DEKÓDOLÓK

A MICROCHIP két különböző dekódoló típust forgalmaz, amelyek a KEELOQ algoritmust használják. Ezek a HCS500-as és a HCS512-es. Rajtuk kívül azonban még mikrokontrollerrel is meg lehet valósítani a dekódoló egységet. A 37. és 38. ábrán a HCS500/512 blokk sémáját láthatjuk.



37. Ábra. A HCS500 blokk diagramja



38. Ábra. A HCS512 blokk diagramja

12.1 JELLEMZŐK

A **26. táblázatban** a dekóder típusok összehasonlítását láthatjuk biztonsági, működési és egyéb szempontok alapján.

26. Táblázat. A KEELOQ dekódolók összehasonlítása

	HCS512	HCS500
Funkció kimenetek	4	1
Funkciók	15	15
Soros csatlakozás	Van	Van
Feszültség	3.0-6.0V	3.0-5.5V
Titkos tanítás	Van	Van
Felhasználói memória	Nincs	Van
Baud rate felismerés	Van	Van
Vlow kimenet	Van	Nincs
Modulálás	PWM	PWM
Belső EEPROM	Van	Nincs
Külső EEPROM	Nincs	Van
Fogadható bitek hossza	67 bit	67 bit
Felismerhető kódolók száma	4	7
Lábak száma	18	8
Kompatibilitás	HCS200, HCS300, HCS301, HCS360, HCS410	HCS200, HCS300, HCS301, HCS360, HCS410

12.2 LÁBKIOSZTÁS

12.2.1 HCS512

27. Táblázat. A HCS512 lábkiosztása

Láb	Dekóder funkció	I/O ⁽¹⁾	Buffer típus	Leírás
1	/LRNIN	I	TTL	Tanító bemenet – tanítás inicializálása, 10K-s felhúzó ellenállást tartalmaz a bemenet
2	LRNOUT	O	TTL	Tanító kimenet – jelzi a tanítást
3	NC	-	TTL	Nincs használva
4	/MCLR	I	ST	Master clear bemenet
5	Ground	P	-	Föld csatlakozás
6	S0	O	TTL	Kapcsoló 0
7	S1	O	TTL	Kapcsoló 1
8	S2	O	TTL	Kapcsoló 2
9	S3	O	TTL	Kapcsoló 3
10	Vlow	O	TTL	Akkumulátor alacsony szint jelző kimenet
11	SLEEP	I	TTL	RFIN-hez csatlakozás, hogy engedélyezze az éledést
12	CLK	I/O	TTL/ST ⁽²⁾	Órajel a programozási módban és a szinkronizálás módban
13	DATA	I/O	TTL/ST ⁽²⁾	Adat láb a programozási módban és a szinkronizálás módban
14	V _{DD}	P	-	Power csatlakozás
15	NC	-	-	Nincs használva
16	OSCIN (4 MHz)	I	ST	Órajel bemenet – javasolt érték: 10K Ω és 10pF
17	NC	-	-	Nincs használva
18	RFIN	I	TTL	RF bemenet a vevőtől

Megjegyzés: 1: P=power, I=bemenet, O=kimenet, ST=Schmitt trigger bemenet

2: A 12-es és 13-as lábnak két rendeltetése van.

12.2.2 HCS500

28. Táblázat. A HCS500 láb kiosztása

LÁB	Dekóder funkció	I/O ⁽¹⁾	Buffer típus ⁽¹⁾	Leírás
1	V _{DD}	P	-	Tápfeszültség csatlakozási pont
2	EE_CLK	O	TTL	Órajel az I ² C EEPROM-hoz
3	EE_DAT	I/O	TTL	Adatláb az I ² C EEPROM-hoz
4	/MCLR	I	ST	Master clear bemenet
5	S_DAT	I/O	TTL	Szinkron üzemmódhoz adatláb
6	S_CLK	I	TTL	Szinkron üzemmódhoz órajel
7	RFIN	I	TTL	RF bemenet a vevő egységtől
8	GND	P	-	Föld csatlakozási pont

Megjegyzés 1: P=Power; I= bemenet; O=kimenet; és ST= Schmitt Trigger bemenet

12.3 MŰKÖDÉSI LEÍRÁSOK

A HCS512-es párhuzamos és soros illesztőegységgel rendelkezik. Amikor egy érvényes üzenetet kap a dekódoló, akkor az S0, S1, S2 és S3 kimeneteit kb. 500 ms-ig aktiválja. Ha ez idő alatt egy ismételt kód érkezik, abban az esetben az előbb említett idő kibővül még kb. 500 ms-mal.

A dekódoló PWM/ szinkronizált interface kapcsolati lehetőséggel rendelkezik, amellyel mikrokontrollerhez lehet csatlakozni. A dekódoló, amikor egy érvényes üzenetet kap, akkor egy kimeneti adat sorozatot generál. Ez tartalmaz egy start bitet, négy funkció bitet, egy akkumulátor állapotát jelző bitet, egy ismétlést jelző bitet, két státusz bitet és egy stop bitet (**39. ábra**). A DATA és CLK vonalat használja, hogy elküldje a szinkronizált üzenetet.

START	S3	S2	S1	S0	V _{low}	REPEAT	TX1	TX0	STOP
-------	----	----	----	----	------------------	--------	-----	-----	------

39. Ábra. Az adat kimeneti formátuma

A tanulás második fázisában a dekódoló egy speciális státusz üzenetet küld el. Ez által tudja meg a mikrokontroller, hogy sikeres volt-e a tanulás vagy sem (Result=1) és azt is, hogy a tanulás során egy előző kódoló adatait írta-e át (Overwrite=1). A státusz üzenet formátumát a **40. ábrán** láthatjuk.

START	0	0	0	0	RESULT	OVRWR	TX1	TX0	STOP
-------	---	---	---	---	--------	-------	-----	-----	------

40. Ábra. A státusz üzenet formátuma

A **29. táblázat** azt mutatja, hogy a TX1 és TX0 bitek állapota hogyan alakul attól függően, hogy a dekódoló hány adó adatait ismeri.

29. Táblázat. Státusz bitek jelentése

TX1	TX2	Adók száma
0	0	Egy
0	1	Kettő
1	0	Három
1	1	Négy

12.4 KONFIGURÁCIÓS BÁJT

12.4.1 HCS500

A dekódoló egy 2K-s 24LC02B soros EEPROM-ot használ az adatok tárolására. A memória fel van osztva rendszer memóriára, ahol az adó adatait tárolja (olvasás védett) és használói memóriára (olvasás/írás lehetséges). A használói memóriának a használatához szükséges utasításokat a későbbiekben ismertetem.

Fontos megjegyezni, hogy:

- a memória részek olvasás védettek és csak programutasítások által írhatóak akkor, ha az eszköz feléled.
- a rendszer memória tartalma egy egyedi 64 bites kulccsal van titkosítva, és ez a kulcs a HCS500-ban van tárolva.
- a rendszer memória inicializálásához a HCS500 programutasításait kell használni.
- az EEPROM és a HCS500 egymáshoz van rendelve és együtt kell használni őket.

A **30. táblázatban** a konfigurációs bájt egyes bitjeinek a szerepét láthatjuk.

30. Táblázat. A HCS500 konfigurációs bájtjának az összetétele

BIT	Mnemonic	Leírás
0	LRN_MODE	Tanulási mód választás LRN_MODE=0 – Normál tanulás LRN_MODE=1 – Titkos tanulás
1	LRN_ALG	Algoritmusválasztás LRN_ALG=0 – Keeloq dekódoló algoritmus LRN_ALG=1 – XOR algoritmus
2	REPEAT	Ismételt átviteli mód engedélyezése 0= tiltva 1= engedélyezve
3	Nem használt	Fenntartott
4	Nem használt	Fenntartott
5	Nem használt	Fenntartott
6	Nem használt	Fenntartott
7	Nem használt	Fenntartott

LRN_MODE

LRN_MODE arra szolgál, hogy a kétféle tanulási mód közül választhatunk. Mint a **30. táblázatban** is láthatjuk, hogy ha LRN_MODE=0, akkor a normál módot választjuk, ahol a

sorozat számot használja a kulcskészítéshez. LRN_MODE=1 beállítással a titkos tanítási módot választjuk.

LRN_ALG

LRN_ALG bit állításával választhatjuk ki a dekódoló algoritmus típusát. Ha LRN_ALG=0, akkor a KEELOQ dekódoló algoritmust választjuk ki és LRN_ALG=1 esetében az XOR algoritmust használja az üzenetek dekódolására.

REPEAT

A HCS500-as beállítható, hogy jelezzen abban az esetben, ha ismételt átvitelt kapott.

12.4.2 HCS512

A **31. táblázatban** a HCS512-es dekódoló konfigurációs bájtjának a beállítási lehetőséget láthatjuk.

31. Táblázat. A HCS512 konfigurációs bájtjának az összetétele

BIT	Mnemonic	<i>Leírás</i>
0	LRN0	Tanulási algoritmusválasztás
1	LRN1	Nem használt
2	SC_LRN	Titkos tanulásengedélyezés (1=engedélyezve)
3	SLEEP	Sleep engedélyezve (1=engedélyezve)
4	RES1	Nem használt
5	RES2	Nem használt
6	RES3	Nem használt
7	RES4	Nem használt

Az LRN1 bit ugyanazt a feladatot látja el, mint a HCS500-nál az LRN_MODE bit és az SC_LRN bit pedig a LRN_ALG bitnek felel meg. Továbbá ugyanazok a beállítás módjai ezeknek a biteknek is.

12.5 A HCS500 EGYÉNI JELLEMZŐI

12.5.1 Mikrokontrollerhez való csatlakozás

A HCS500-at szinkronizált soros illesztőn keresztül lehet a mikrokontrollerhez csatlakoztatni. Ekkor a mikrokontroller a CLK és a DATA vonalat vezérli a kommunikáció megteremtéséhez. Az adat üzeneteknek két csoportja van. Az első üzenet a dekódolótól származik, melyet akkor küld el, amikor egy érvényes üzenetet kap. Ezt a DATA vonal magasba emelésével jelzi (max. 500ms). Ez után a mikrokontroller az órajellel vezérli a dekódolót, hogy az adat stringet elküldje. Az adat string tartalmazza a funkció kódokat, a státusz biteket és a blokkmutatókat. A második üzenetet a mikrokontrollertől származik, mely a definiált parancs beállításokat tartalmazza.

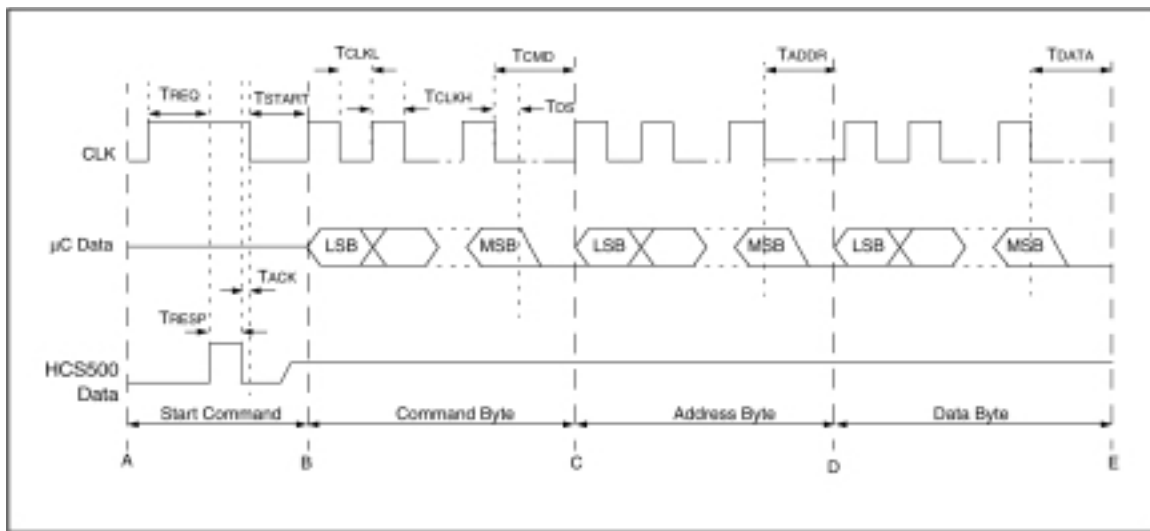
12.5.1.1 Érvényes átviteli üzenet

A dekódoló a DATA vonal kb. 500 ms-ig történő magasban tartásával értesíti a mikrokontrollert, hogy érvényes üzenetet kapott. Ezt a kontroller a CLK vonal magasba emelésével nyugtázza. Ezek után a kódoló a DATA vonalat alacsonyba állítja, mire a kontroller elkezd az órajelet küldeni, hogy megkapja az adat sorozatot. Az adat sorozat egy start bitből (mely 0), 2 státusz bitből (REPEAT, Vlow), 4 funkció bitből (S0, S1, S2, S3), négy jelző bitből, hogy melyik blokkot használja (TX3...TX1), 4 kódoló darab számát jelző bitből (CT3...CT0) és egy 64 bites üzenetből áll, mely tartalmazza az ugró kódot. A dekódoló az adat sorozat átvitele során bármikor félbeszakítja a folyamatot abban az esetben, ha az órajelel 1 ms-nál több ideig alacsony állapotban van. Ennek következtében a mikrokontroller csak az előírt (megszabott) biteket olvashatja ki. Ez csak maximum 80 bitből állhat.

12.5.1.2 Parancs üzemmód

Mikrokontroller parancs üzemmód aktiválása

A mikrokontroller parancsa négy részre osztható. Az első aktivizálja ezt az üzemmódot, a második tartalmazza az aktuális parancsot, a harmadikban a cím található és a negyedik rész tartalmazza az adatot. A mikrokontroller a parancsot az CLK vonal több mint 500 ms-ig tartó magasba emelésével kezdi. A dekódoló ezt azzal nyugtázza, hogy a DATA vonalat magasba emeli. Erre a mikrokontroller, azután hogy a dekódoló a DATA vonalat már alacsonyba rakta, a CLK vonalat alacsonyba állítja. A DATA vonal háromállapotú. Az adatot az órajel felmenő élénél kell összerakni és a lemenő élénél kell mintavételezni (41. ábra).



41. Ábra. A HCS500 parancs üzemmódjának az aktiválása

Ütközésfigyelés

A HCS500 ütközésfigyelést használ, hogy meggátolja a dekódoló és a mikrokontroller összeütközését. Amikor a dekódoló egy érvényes üzenetet kap, akkor a következő sorrendet követi:

- Először ellenőrzi, hogy a CLK vonalat magasban látja-e. Ha igen, akkor az üzenet fogadást félbe hagyja és a mikrokontroller utasításának tesz eleget.
- A dekódoló a DATA vonalat magasba emeli és ellenőrzi, hogy a CLK vonal magasba került-e 50µs-on belül. Ha igen, akkor mindent félbeszakít és a parancs üzemmódot szolgálja ki.
- Ha a CLK vonal 50µs-nál később és 500ms-nál hamarabb került magasba, akkor ezt a dekódoló nyugtázza a DATA vonal alacsonyba állításával.
- A mikrokontroller csak 80 bit olvasására képes.

Dekódoló parancsok

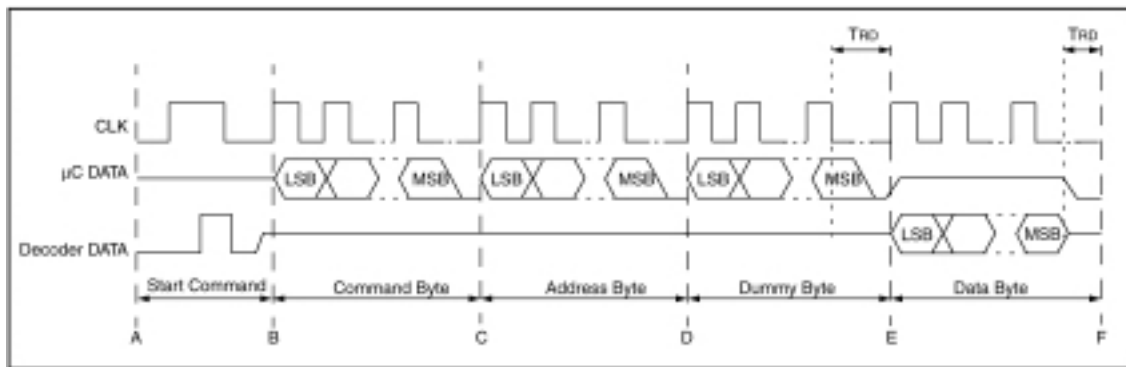
A **32. táblázat** mutatja, hogy milyen utasításokat használhat a vezérlő mikrokontroller.

32. Táblázat. A dekódoló parancsai

Utasítások	<i>Utasítás bájt</i>	Működés
READ	F0 ₁₆	Egy bájtot olvas a használói EEPROM-ból
WRITE	E1 ₁₆	Egy bájtot ír a használói EEPROM-ba
ACTIVATE_LRN	D2 ₁₆	Aktivizálja a tanulás sorrendjét a dekódolón
ERASE_ALL	C3 ₁₆	Aktivizálja a mindent törlő funkciót a dekódolón
PROGRAM	B4 ₁₆	A gyártó kódot és a konfigurációs bájtot programozza

Bájt(ok) olvasása a használói EEPROM-ból

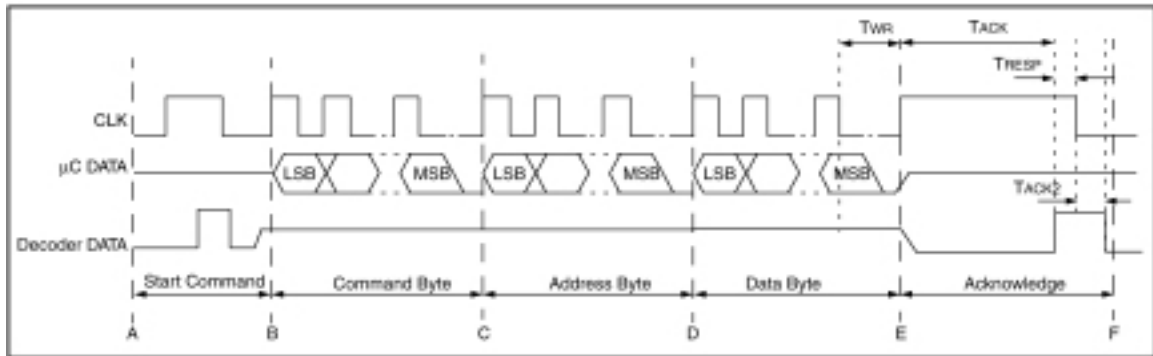
Az EEPROM-ból való olvasás folyamatát a **42. ábrán** láthatjuk. A memóriában való eltolást (offset) a cím bájt határozza meg (lásd ábra C-D). Azután egy látszólagos bájt következik (D-E). A adat bájtok a következő órajel felfutó élére fognak átvitelre kerülni, mégpedig úgy, hogy az alsó biteknél kezdődően (E-F). Folyamatos adatolvasás lehetséges, ha az előző bájt utolsó (MSB) bitjének lefutó éle után 1 ms-on belül megismételjük az E-F szakaszt. A dekódoló befejezi a parancs végrehajtását, ha 1.2 ms-nál több ideig nem kapja meg az órajelet.



42. Ábra. Bájtok olvasása az EEPROM-ból

Bájt(ok) írása a használói EEPROM-ba

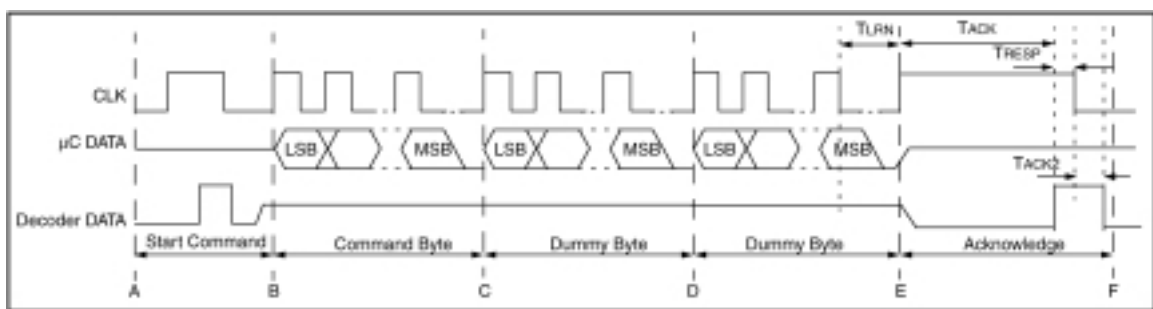
Az EEPROM-ba való írás folyamatát a **43. ábrán** láthatjuk. Először megvizsgálja, hogy milyen címre kell az adatokat írnia és utána az alsó biteknél kezdi az írást. A folyamatos írást is lehetővé teszi a rendszer és ekkor a D-F szakaszt kell ismételtetni. A dekódoló befejezi a parancs végrehajtását, ha 1.2 ms-nál több ideig nem kapja meg az órajelet. A sikeres memóriairás után a dekódoló nyugtázó jelet küld úgy, hogy a DATA vonalat magasban tartja, amíg a CLK vonal alacsonyba nem kerül.



43. Ábra. Bájtok írása az EEPROM-ba

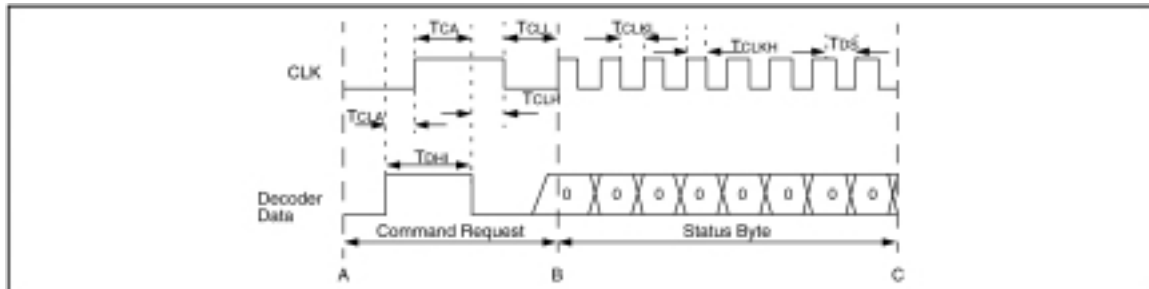
Tanulás aktiválása

A tanulásaktiváló parancsot arra használják, hogy aktiválja a tanulási feladatot a dekódolóban (**44. ábra**) Az utasítás egy parancs üzemmód aktiváló sorozatból, egy parancs bájtól és két látszólagos bájtól áll. A dekódoló a DATA vonal magasba állításával nyugtázza a parancs érvényességét és azt, hogy a tanulási mód aktivizálva lett.



44. Ábra. Tanulás aktiválás a HCS500 esetében

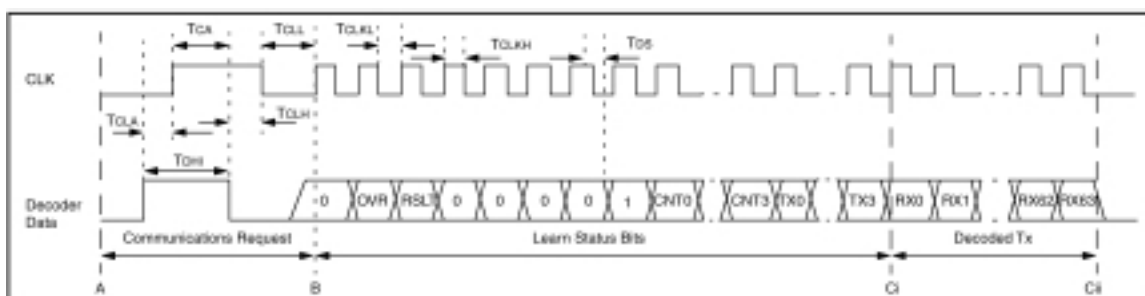
Amint a dekódoló megkapja az első üzenetet, azután egy tanulási állapotot jelző üzenettel válaszol (**45. ábra**).



45. Ábra. Az első tanulási állapotot jelző üzenet

Tanulás alatt a dekódoló az első üzenetet átvételét azzal nyugtázza, hogy a DATA vonalat 60 ms-ig magasba emeli. Ezután elküld egy státusz bájtot, ami arra szolgál, hogy tájékoztassa a kontrollert, hogy megtudja, hogy a dekódolónál time-out történt vagy pedig megkapta az első üzenetet. A mikrokontrollernek biztosítania kell, hogy a CLK vonal 60 ms-on belül ne kerüljön magasba az után, hogy a DATA vonalon a jelnek lefutó elé volt.

Amint a dekódoló megkapja az második üzenetet, azután szintén egy tanulási állapotot jelző üzenettel válaszol (**46. ábra**).



46. Ábra. Az második tanulási állapotot jelző üzenet

Ez az üzenet a következőkből áll:

- 1 start bit

- A funkció kód [S3:S0] az üzenetben nulla, mellyel az jelzi, hogy ez egy tanulási állapotot jelző string.
- Az RESULT bit jelzi a tanulás eredményét. Ha sikerült, abban az esetben beállítódik, egyébként törlődik.
- Az OVR bit akkor jelez, ha valamelyik létező adó adatai felül lettek írva. Ezt a bit magasba állításával teszi meg.
- A [CNT3...CNT0] bitek jelzik a megtanult kódolók számát.
- A [TX3...TX0] bitek a blokk számot jelzik, melyet a kódoló adatainak a tanulása alatt használt.

12.5.1.3 Mindent törlő parancs

A törlő parancsnak a két verzióját és értelmezését a **33. táblázatban** láthatjuk.

33. Táblázat. A mindent törlő parancs értelmezési

Utasítás bájt	Subcommand bájt	Jellemző
C3 ₁₆	00 ₁₆	Minden kódoló adatát törli a memóriájából.
C3 ₁₆	01 ₁₆	Minden kódoló adatát törli a memóriájából, kivéve egyet. Még pedig az első helyen lévőét nem törli.

A 01-es subcommand-ot arra használja a mikrokontroller, hogy megjelölje annak a kódolónak az adatait, melyet sohasem fog törölni.

12.5.1.4 Tesztmód

A gyártó kód beprogramozása vagy minden kódoló adatainak a törlése után, a dekódoló egy speciális tesztmódot aktivizál. Ez a dekódoló tesztelésére használható azelőtt, mielőtt

megtanítanánk vele egy kódoló adatait. Anélkül lehet tesztelni így, hogy időt fordítanánk a tanulási folyamatra. Ez az állapot azonnal megszűnik, amikor egy sikeres tanulási folyamatot véghez viszünk. Ilyenkor az adó a következő adatokkal rendelkezik:

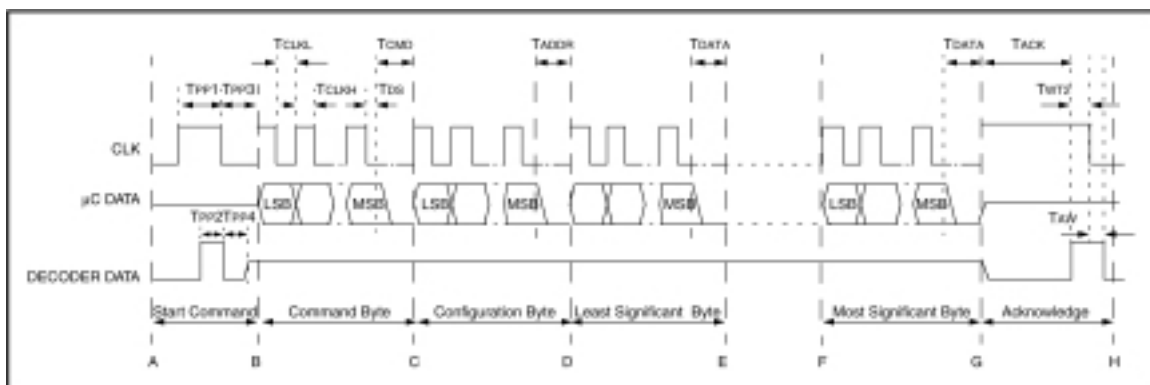
- Titkosító kulcs = gyártó kód
- Sorozat szám = bármilyen érték
- Azonosító bitek = a sorozat szám alsó 10 bitje
- Szinkronizáló számláló értéke = bármilyen érték (nem vesszük figyelembe)

Mivel tesztmódban nem vesszük számításba a szinkronizáló számláló értékét, bármennyi kódolót használhatunk attól függetlenül, hogy mennyi a számlálójuknak az értéke.

12.5.2 Programozása

A HCS500 programozása a következő szakaszokra osztható (lásd 47. ábra):

- Felszólító szakasz (A-B)
- Utasítás bájtt (B-C)
- Konfigurációs bájtt (C-D)
- A gyártó 8 adat bájttja (D-G)
- Aktivizáló és nyugtázó szakasz (G-H)



47. Ábra. A HCS500 programozásának módja

Ilyenkor egy 80 bites adat stringet töltünk fel a dekódolóba. Először a 8 bites utasítás kódot programozzuk be, majd azt követi a 8 bites konfigurációs bájt és a 64 bites gyártó kód. A gyártó kódot az alsó bitjeinél kezdve programozzuk. Miután megtörtént a feltöltése a dekódolónak, azután azonnal titkosítja a gyártó kódot a 64 bites EEPROM titkosító kulccsal. Az EEPROM-ba programozás befejeztével a dekódoló a folyamatot a DATA vonal magasba emelésével nyugtázza (G-H). Ha a CLK vonal magasba emelése után 30 ms-on belül történik a DATA vonal magasba emelése, akkor a programozás érvénytelen lesz.

12.6 A HCS512 EGYÉNI JELLEMZŐI

12.6.1 Adó tesztelése

A HCS512 dekódoló, a kódoló adatainak a törlése után automatikusan egy tesztadót ad a rendszerhez. A tesztadó úgy van definiálva, mint egy normál adó, csak a sorozat száma nulla. Egy ilyen törlés után az adó mindig tanulás nélkül működik és nem ellenőrzi a kódoló szinkronizáló számlálójának az értékét. Egy új kódoló adatainak a megtanítása a dekódolóval törli a tesztadót.

Meg kell jegyezzük, hogy a nulla sorozat számú kódoló nem tanítható meg a dekódolóval. Ha mégis megpróbáljuk, akkor azt tapasztaljuk, hogy az első átvitel után a tanulási folyamat leáll.

12.6.2 Programozása

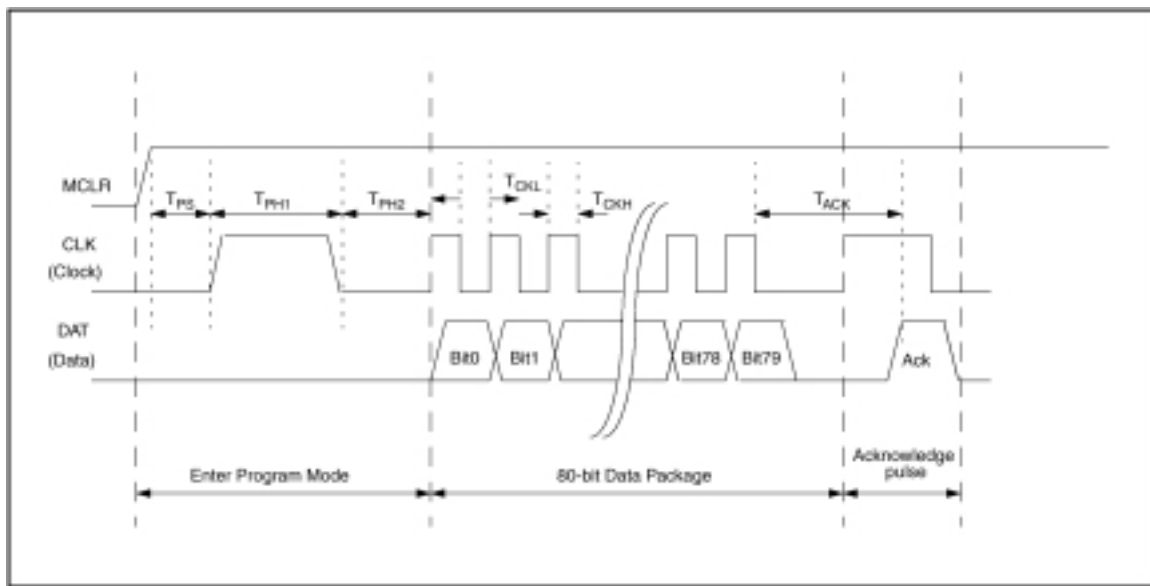
A gyártó kódot kell beprogramozni a HCS512 EEPROM-jába. Ezt a soros interface-n keresztül tehetjük meg a CLK és DATA vonalak használatával.

A programozást úgy aktivizálhatjuk (lásd **48. ábra**), hogy a CLK vonalat legalább 1 ms-ig alacsony állapotba tartjuk és azután az eszköz feléledését követően 64 ms-on belül magasba emeljük legalább 8, de nem több mint 128 ms-ig úgy tartjuk. Ezután történik meg a 80 bites üzenet programozása, mely a 64 bites gyártó kódból, a 8 bites konfigurációs bájtól és a 8 bites

checksum bájtból áll (49. ábra). Ha a programozás sikeresen befejeződik, akkor a dekódoló egy nyugtázó jelet küld.

Miután a gyártó kódot beprogramoztuk a dekódolóba, azután automatikusan kitörli minden előzőleg az EEPROM-jában tárolt kódolónak az adatát.

A programozáshoz szükséges időtartamokat a 34. táblázat tartalmazza.



48. Ábra. A HCS512 programozásának módja

Byte 9	Byte 8	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Check-sum	Config	Man Key 0	Man Key 1	Man Key 2	Man Key 3	Man Key 4	Man Key 5	Man Key 6	Man Key 7

Byte 0, right-most bit downloaded first. →

49. Ábra. A letöltött adat összetétele

34. Táblázat. A programozáshoz szükséges időtartamok

12.6.2.1.1.1.1 Parameter	Symbol	Min.	Max.	Units
Program mode setup time	TPS	1	64	ms
Hold time 1	TPH1	8	128	ms
Hold time 2	TPH2	0.05	320	ms
Clock High Time	TCKH	0.05	320	ms
Clock Low Time	TCKL	0.050	320	ms
Acknowledge Time	TACK	-	80	ms

12.6.3 Checksum

Hibaellenőrzésre használja a HCS512-es dekódoló. Ezt úgy vizsgálja meg, hogy a kapott üzenet 10 bájtyát összeadja (túlcsorduló biteket levágja) és az eredménynek nullának kell lennie. Magát a checksum bájtot pedig úgy határozzuk meg, hogy összeadjuk az üzenet első 9 bájtyát (túlcsorduló bitek levágásával) és az eredményt pedig kivonjuk nullából.

13 ADATÁTVITELI MÓDSZEREK ALKALMAZÁSA AZ UGRÓ KÓDOS RENDSZEREKBE

13.1 INFRA ADATÁTVITEL

Az infra adatátvitel során optoelektronikus úton történik az átvitel. A vevő az adó impulzusait kb. 8...25 m távolságból üzem biztosan veszi attól függően, hogy a fotodióda közvetlenül vagy gyűjtőlencsén keresztül kapja az adó jelét. Szabad térben az adónak és a vevőnek egymással szembe kell állnia, vagyis rálátást kell egymásra biztosítani, míg zárt térben azonban a reflexiónak is nagy jelentősége van. Így pl. mennyezetre, illetve oldalra vagy esetleg ellenkező irányba sugározva is működik a vevő, azonban a reflexió is nagymértékben függ a reflektáló felület minőségétől. Világos, sík felületek jól reflektálnak, sötét és tört felületek abszorbeálják a sugárzás nagy részét. Nem közömbös a helység világítása sem. A vevő ugyanis – a túlvezérlés elkerülésére – a vett jel egyértelmű kiértékelhetősége érdekében erősítésszabályozott. Így erős környezeti fényben csökkentett érzékenységgű. Itt elsősorban a nagy infravörös tartalmú világítás (izzólámpák) a mértékadó. Ennek a hatásnak a csökkentésére szoktak használni a fotodióda elé infravörös szűrőt.

A jelátvitel lényege egy kapcsolójel kiküldése, majd ennek szelektív vétele. A kapcsolójel disszipációs, fényhasznosítási, telepkímélő stb. szempontból szaggatott, és hosszát gyakorlati értékek határolják be alulról és felülről. A kisugárzott energia szempontjából az impulzuscsomagnak minél rövidebbnek kell lennie (telepkímélés), zavarvédelmi szempontból pedig minél hosszabbnak, hiszen a hosszú idejű, állandó frekvenciájú zavarok előfordulási gyakorisága a természetben rendkívül kicsi. Egyéb távirányítós készülékek jelei tekinthetőek zavarójelnek.

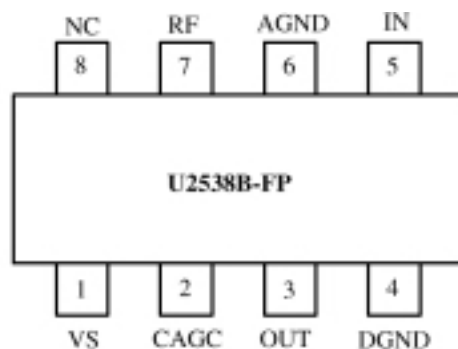
A következőkben bemutatnák infra adó és vevő berendezéseket.

13.1.1 TEMIC U2538B

Ez az IC az infrás adatátvitelhez egy komplett vevő. Egy belső speciális áramkörrel szétválasztja a hasznos bemenő jeleket és egy erősítő felerősíti ezeket a jeleket. A sáváteresztő szűrő pedig értelemeszerűen „elnyomja” a nem kívánt jeleket. A jel detektor egy modulátorból, egy integrátorból és egy Schmitt-trigger-ből áll, mely a bemenő jeleket felhasználható kimenő jellé formálja és lehetővé teszi, hogy pl. egy mikrokontrollerhez, vagy egy dekódolóhoz csatlakoztassuk az IC-t. A vevő érzékenységét az AGC és ATC áramkör vezérli és érzéketlené teszi a vevőt a környezeti fényforrásokkal szemben.

A vivő frekvencia 20-60kHz.

13.1.1.1 Lábkiosztás

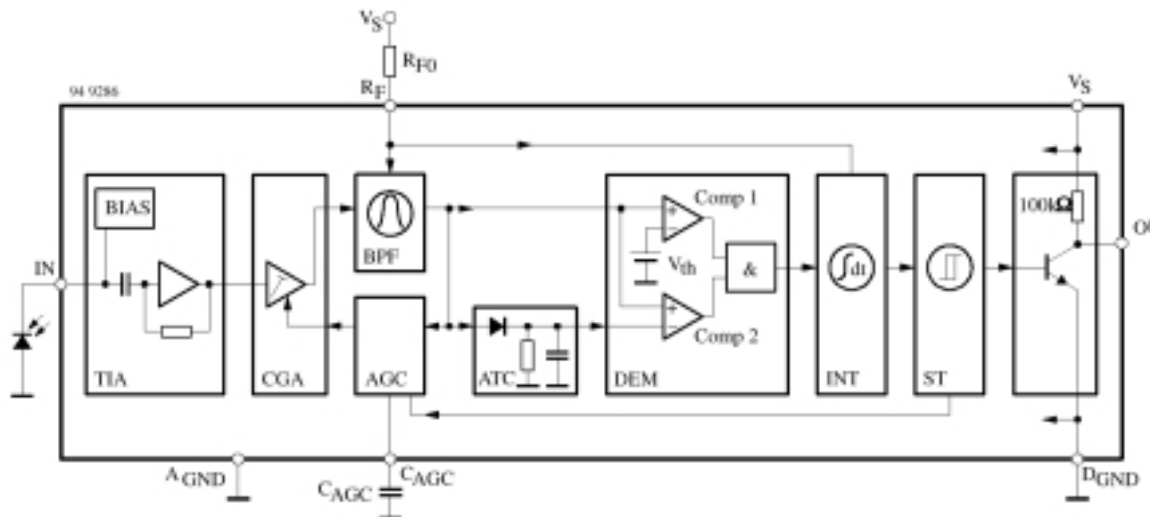


50. Ábra. A TEMIC által gyártott U2538B lábkiosztása

35. Táblázat. Az U2538B lábainak értelmezése

Pin	Symbol	Function
1	VS	Supply voltage
2	CAGC	AGC capacitor
3	OUT	Data output
4	DGND	GND – DEM/INT/ST
5	IN	Input pin diode
6	AGND	GND amplifier
7	RF	Frequency determination
8	NC	Not connected

13.1.1.2 Blokk digram



51. Ábra. A TEMIC U2538B belső blokk-sémája

TIA	Transimpedance erősítő	ATC	Automatikus küszöbérték szabályozó
CGA	Szabályozott teljesítményerősítő	DEM	Demodulátor
BPF	Sáváteresztő szűrő	INT	Integrátor
AGC	Automatikus erősítés szabályozó	ST	Schmitt-trigger

13.1.1.3 Bemeneti szakasz (TIA)

A fotodiódának szükséges előfeszítésről és a hasznos jelek szétbontásáról gondoskodik. Ez a folyamat két speciális részre bontható: az előfeszítésre (BIAS) és a transimpedance erősítő áramkörre (TIA). A feszültség előfeszítő áramkörterhelő ellenállásként működik a fotodiódára nézve. A bemenő jelnek az „ac” része elegendően alacsony bemeneti ellenállással táplálja az invertáló erősítőt ($Z_i < 10\text{k}\Omega$). Ha a bemeneti ellenállás túl nagy lenne, akkor a bemeneti jelek elvesznének.

13.1.1.4 Szabályozott teljesítményerősítő (CGA)

A feszültségerősítésből ennek a résznek van a legnagyobb jelentősége és a C_{AGC} lábánál szabályozható. Az erősítés szabályozására azért van szükség, hogy biztosítsuk a detektor nagyfrekvenciás zajszűrését. A határfrekvencia kb. 20KHZ.

13.1.1.5 Sáváteresztő szűrő (BPF)

A sáváteresztő szűrő alaplól tartalmaz beintegrált komponenseket. Egy külső ellenállás határozza meg a középfrekvenciát. A következő képlet használatával kiszámíthatjuk az ellenállás értékét (R_{f0}):

$$R_{f0}(k\Omega) = \frac{8855}{f_0(kHz)} - 13 \quad \text{ahol : } 20 \text{ kHz} < f_0 < 60 \text{ kHz}$$

13.1.1.6 Automatikus küszöbérték szabályozó (ATC)

A bejövő üzenet fogadásakor az ATC lecsökkenti a demodulátor érzékenységét, hogy megállapítsa a lehető legnagyobb jel-zaj viszonyt (hányadost) a jel erősség beszámításával. Ez elhárítja a nagyfrekvenciás zavarokat, amelyek a bemeneten keletkeznek az átvitel fogadásakor. Ennek az áramkör az előnye az, hogy ha a kimeneti feszültség túllépi a V_{th-t} , akkor megőrzi az adatokat. Ez abban az esetben történhet meg, hogy ha bementi jel erőssége kétszer nagyobb, mint a minimálisan kimutatható jel intenzitása.

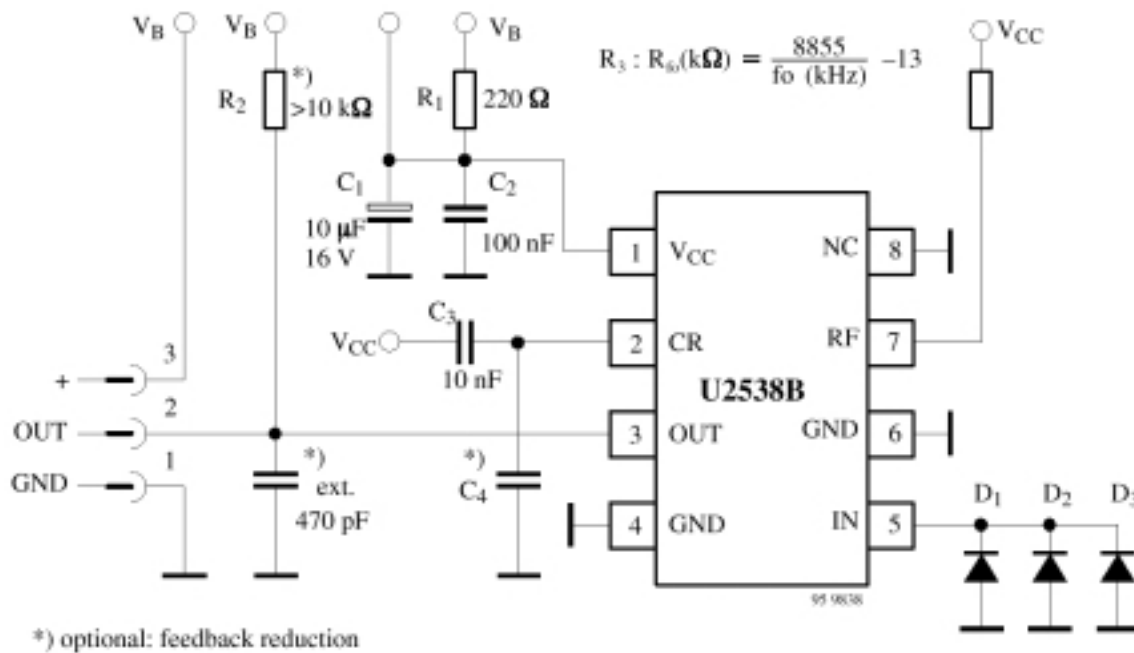
13.1.1.7 Automatikus erősítés szabályozó (AGC)

Az AGC javítja a kör ellenállását a nagyfrekvenciás zavarokkal szemben. Fokozatosan lecsökkenti az érzékenységet, de csak akkor, ha a teljesítmény ciklus túllép egy meghatározott értéket. Ha a teljesítmény ciklusokat nagyobb értékkel használjuk, mint az előző érték, akkor a kapacitás, C_{AGC} , gondoskodik egy meghatározott időre az érzékenységről. A nagyobb kapacitás a hosszabb átviteli időt teszi lehetővé.

13.1.1.8 Jelérzékelő

A sáváteresztő szűrő kimeneti jelét egy referencia értékkel (Comp1) és az ATC által generált jellel hasonltja össze (Comp2). A komparátor kimenetén egy nagyobb küszöb feszültséggel szabályozza az integrátort. Arra használjuk az integrátort, hogy a kimeneti jelet „megtisztítsa” a rövid ideig tartó nagyfrekvenciás zavaroktól. Az integrátor vezérli tulajdonképpen a Schmitt-trigger-t. A belső felhúzó ellenállások néhány alkalmazásban külső ellenállással helyettesíthetők.

13.1.1.9 Áramköri alkalmazása



52. Ábra. A TEMIC U2538B egy áramköri alkalmazási lehetősége

13.1.2 TEMIC U2535B-FP

Hasonló a felépítése az U2538B-hez, azzal a különbséggel, hogy a vivő frekvencia itt 20-100kHz lehet.

13.2 RÁDIÓFREKVENCIÁS ADATÁTVITEL

A rádiófrekvenciás adatátvitel egy vivőjel segítségével történik. A vevő az adó jeleit kb. 10-100 m távolságból képes üzem biztosan venni. Az infrás átvitellel szemben, ennél rálátás nem szükséges az adó és a vevő között, mely tulajdonképpen a legnagyobb előnye. Nagyobb hatáskörü átvitelt tesz lehetővé, ami azt jelenti, hogy kisebb teljesítménnyel nagyobb távolságban is képes működni és nem túl költséges. Hátránya, hogy árnyékolható, mely miatt sikeres átvitel nem jöhet létre. Ha az adó frekvenciáján, vagy annak al- ill. felharmonikusán üzemel a közelben egy nagyobb adó, akkor ez szintén zavarhatja, illetve sikertelenné teheti az adatátvitelt.

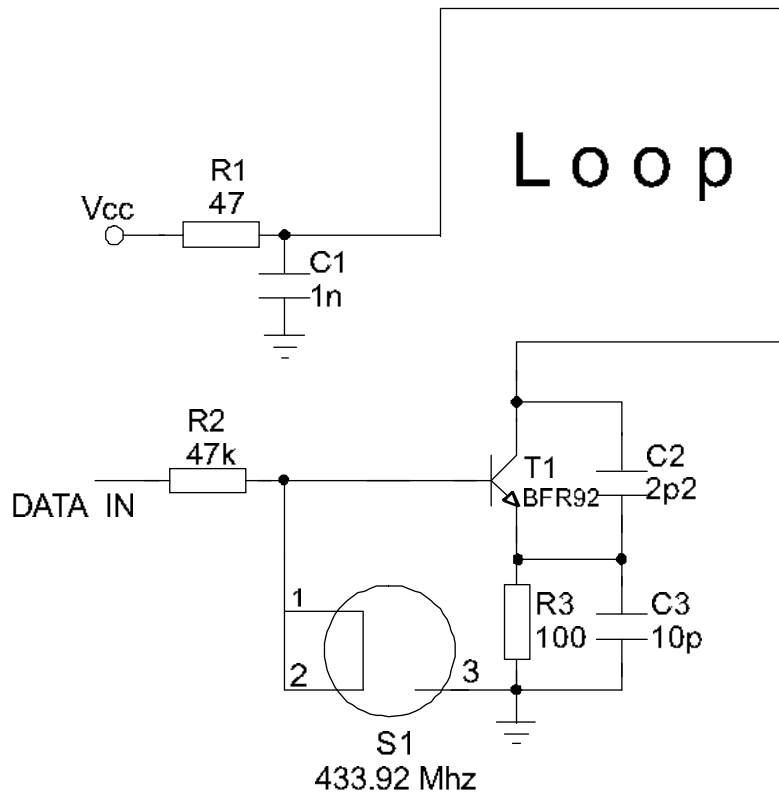
Jelentős az adónál energiatakarékossági szempontból, hogy vivő teljesítmény csak akkor van jelen, amikor a „vezérlő” PWM jelnek magas szintje van, vagyis csak a moduláció ideje alatt működik az adó vivőfrekvencia „generátora”.

Számos megvalósítási lehetőségeket találhatunk és készíthetünk a rádiófrekvenciás adatátvitelre. A következőkben ismertetném a CHIPCAD által forgalmazott RF adó- és vevő egységet, továbbá bemutatnék más cégek által terjesztett komplett vevő IC-eket illetve modulokat. Azért van a hangsúly inkább a vevőkön, mert azok a bonyolultabbak.

13.2.1 CHIPCAD által forgalmazott RF adó, vevő

Az áramköri megvalósítását az adónak az **53. ábrán**, a vevőnek az **54. ábrán** láthatjuk.

ADÓ

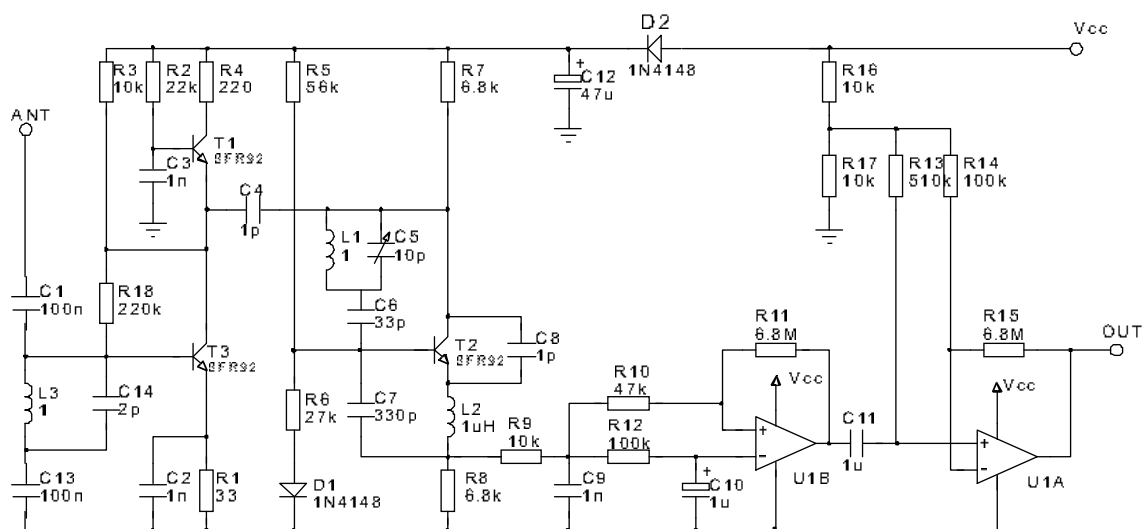


53. Ábra. A CHIPCAD által forgalmazott RF adó

A DATA IN vonalon kapja a kódolótól a PWM modulált üzenetsomagokat. Az S1 oszcillátor biztosítja a vivőfrekvenciát, mely csak a PWM jel magas szintjénél „engedélyezi” a rezonátor működését. Az R1 ellenállás a munkaponti ellenállása a rezgőkörnek. A C1 kondenzátort zavaraszűréshez használja az adó. Az R3 ellenállás a tranzisztor munkapontjának a beállításához szükséges, míg a C3 kondenzátor pedig hidegítésre szolgál. A T1 tranzisztor pedig mentesíti az oszcillátort a terheléstől, és vezérli a rezgőkört a bejövő adatoknak megfelelően.

VEVŐ

Az **54. ábrán** látható kapcsolási rajz működését tekintve láthatjuk, hogy a T3-as tranzisztor köre egy sáverősítő, mely a 433.92 MHz középsávjának egy részét erősíti. A T3 kollektor körében lévő dinamikus ellenálláson (T1 tranzisztoron) keresztül állítjuk be a központi frekvenciát. Ezután a jel egy további jelerősítő fokozatra kerül, melyet a T2-es tranzisztor képvisel, ahol a jel szintjének (még nem logikai szintek) erősítése történik. Következik az U1B fokozat, mely a jel tisztítását végzi (zavarokat szűri le, de még nem a négyzög jelről). Majd ennek a kimenete egy Schmitt-trigger bemenetéhez csatlakozik, mely elkészíti a végleges kimeneti formátumot.

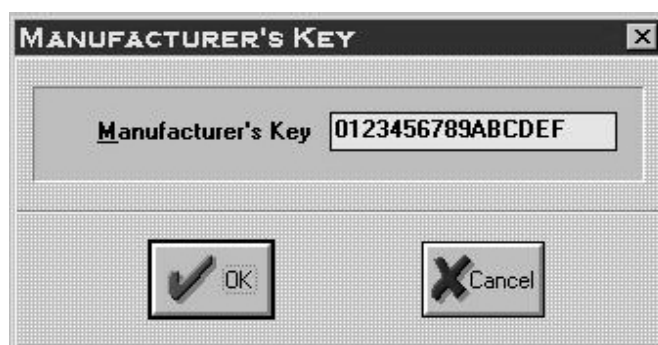


54. Ábra. A CHIPCAD által forgalmazott RF vevő

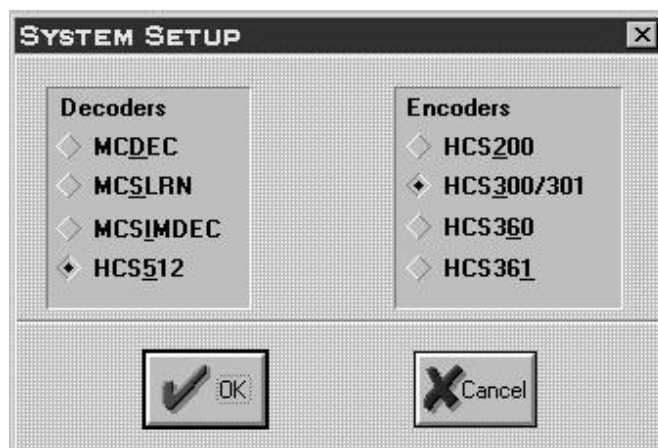
14 A CHIPCAD ÁLTAL FORGALMAZOTT HCS PROGRAMOZÓ SZETT RÖVID BEMUTATÁSA

A programozások megkezdése előtt be kell állítani bizonyos paramétereket. A programozóhoz adott szoftverben van egy ún. setup menüpont, ahol a következőket lehet beállítani:

- gyártó kód

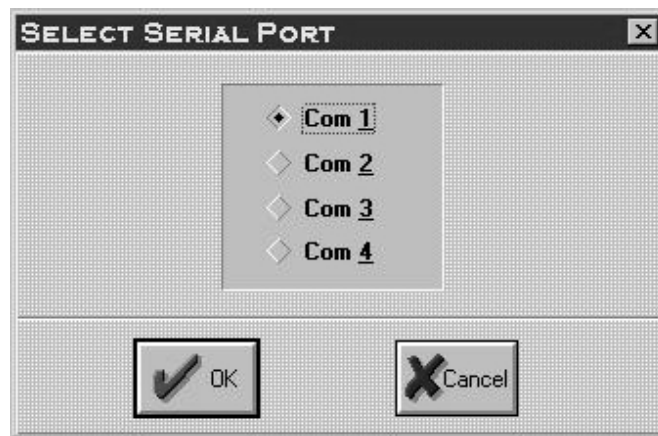


- rendszer beállítás



Itt választhatjuk, ki hogy milyen eszközöket akarunk alkalmazni. Mindegyikből csak egyet választhatunk, ami azt jelenti, hogy egy rendszeren belül csak egy féle kódolókat használhatunk. Láthatjuk, hogy a HCS500 dekódolót, a HCS410 kódolót még nem ismeri a program.

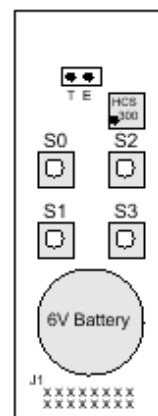
- soros port választás



Az aktuális soros portot választhatjuk ki, amelyiken keresztül programozzuk az eszközöket.

14.1 A KÓDOLÓ

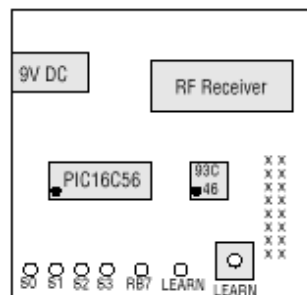
A szett tartalmaz két db négy nyomógombos demonstrációs adót. Ezen az összes funkciógomb kombinációt ki lehet próbálni. Két db 3V elem van benne, azért, hogy lehessen tesztelni az alacsony feszültség jelző funkcióját is a rendszernek. Az ábrán is látható jumper arra jó, hogy az RF adót leválasszuk a kódolóról és így rácsatlakozva figyelhessük a kimenetet. Az RF adó 433.9 MHz-en működik. A kódoló programozásához 4 vezetékes csatlakozás szükséges.



14.2 A DEKÓDOLÓ

Az ábra és a valóság annyiban eltér, hogy még egy HCS512-es is található rajta.

Ha adást vesz a dekódoló (de nem biztos, hogy érvényes, vagy ismeri egyáltalán a kódolót, amelyik küldte a jelet), akkor a LEARN sűrűn villog. A LEARN gomb megnyomása esetén lehet megtanítani a kódoló(k) adataira. Ha nyomva tartjuk ezt a gombot kb. 8 sec-ig, akkor törlődik a memóriája a dekódolónak.



14.3 A DEKÓDOLÓK BEÁLLÍTÁSAI

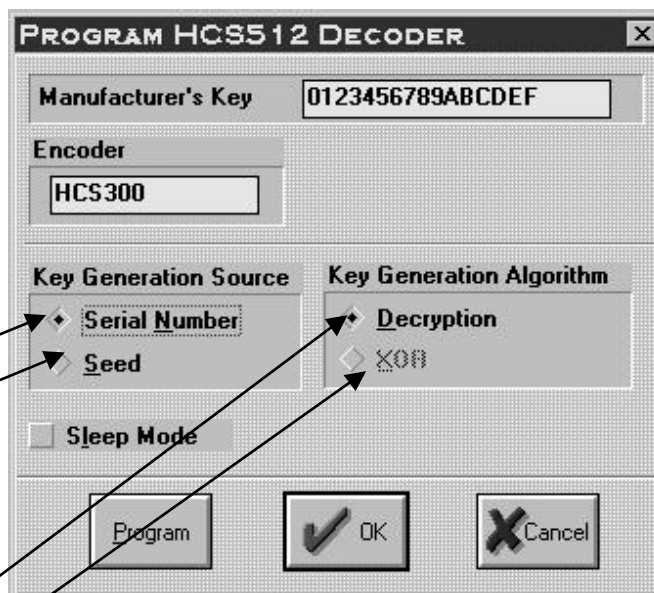
Ugyebár, mint említettem, hogy sajnos csak HCS512-es dekódolót lehet ezzel a programmal programozni, ezért csak ennek a beállításait tekinthetjük meg.

Továbbá itt választhatjuk meg, hogy mi legyen a forrása a kulcskészítő algoritmusunknak:

- sorozat szám
- seed érték

Ugyanakkor az alkalmazott algoritmus fajtáját is itt határozhatjuk meg:

- „titkosítás”
- XOR



Az XOR algoritmust csak a seed forrás során választható.

14.4 A KÓDOLÓK BEÁLLÍTÁSAI

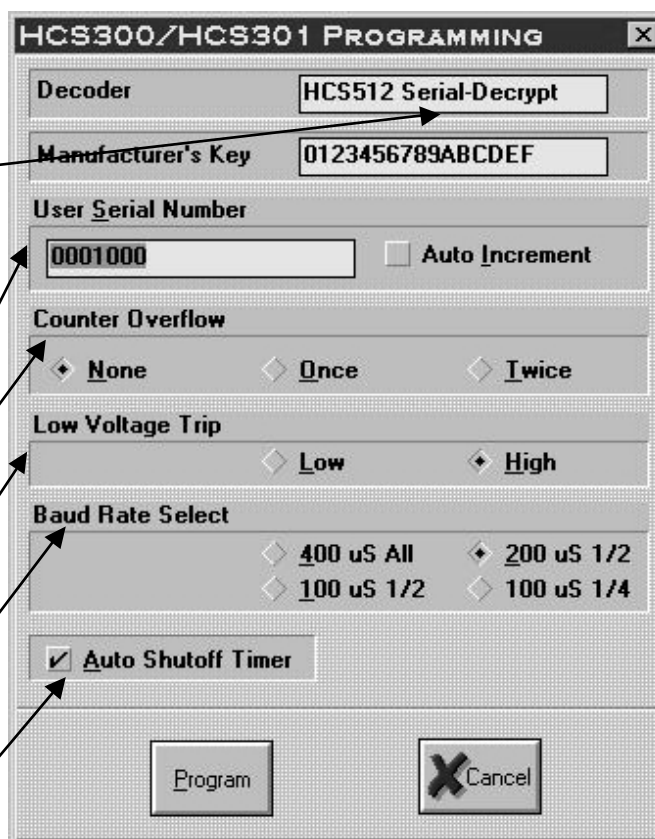
A kódolónál már többféle lehetőség van ellentétben a dekódolókkal, és mint tudjuk ezek beállítási lehetőségei is különböznek egymástól.

Ezek különbségeit nem részletezném. Ezt már az előbbiekben megtettem. Nézzük a HCS300-as egységet példának.

Először a dekódoló kell programozni és akkor már láthatjuk, hogy beállítja, hogy a dekódoló milyen algoritmust és forrást használ a kulcskészítéshez.

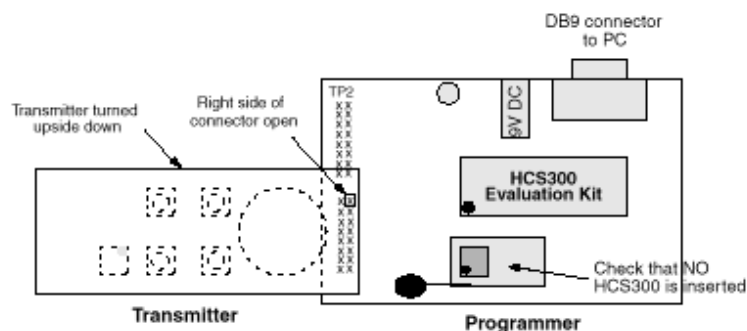
Itt mi állíthatjuk be:

- a sorozat számot
- a számláló túlsordulást
- alacsony feszültség szintet
- baud rate-tet
- auto shutoff timer-t



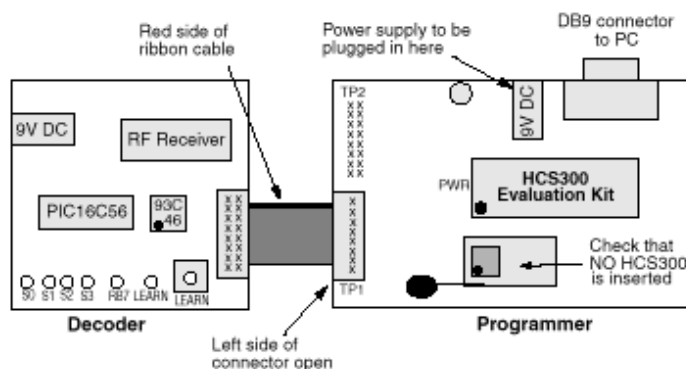
14.5 A KÓDOLÓK PROGRAMOZÁSA

A kódolók programozásához az egész adó egységet csatlakoztatnunk kell a programozóhoz az ábrának megfelelő módon. Persze külön is lehet DIP tokozású HCSXXX-t programozni.



14.6 A DEKÓDOLÓ PROGRAMOZÁSA

Ehhez a folyamathoz pedig az ábrának megfelelő módon kell a dekódolót a programozóhoz csatlakoztatni egy kábel segítségével.



14.7 TANÍTÁSI ELJÁRÁS

A dekódolók tanítási eljárásának a jellemzője, hogy lehetővé teszi a rendszer számára, hogy új adók paramétereit lehet megtanítani a vevővel anélkül, hogy újra programoznánk a

berendezést. A megtanítások után automatikusan felismeri, hogy melyik adótól jött az utasítás a feladat végrehajtására. Meg kell jegyezni, hogy ha egy dekódolót titkos tanulásra (seed átvitelre) és azon belül is pl. XOR kulcskészítő algoritmus alkalmazására programoztuk fel, abban az esetben a csak ezekkel a beállításokkal (lásd „A kódoló beállításai” c. fejezetet) felprogramozott kódoló adatait lehet megtaníttatni vele. Továbbá nagyon fontos, hogy ugyanannak a gyártó kódoknak kell szerepelnie az egy rendszerben használt eszközöknek.

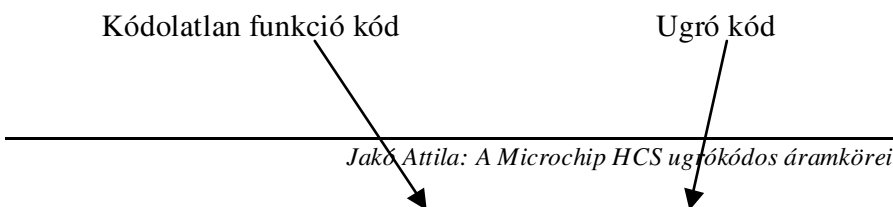
A tanulási folyamat, a már ismertetett módon történik.

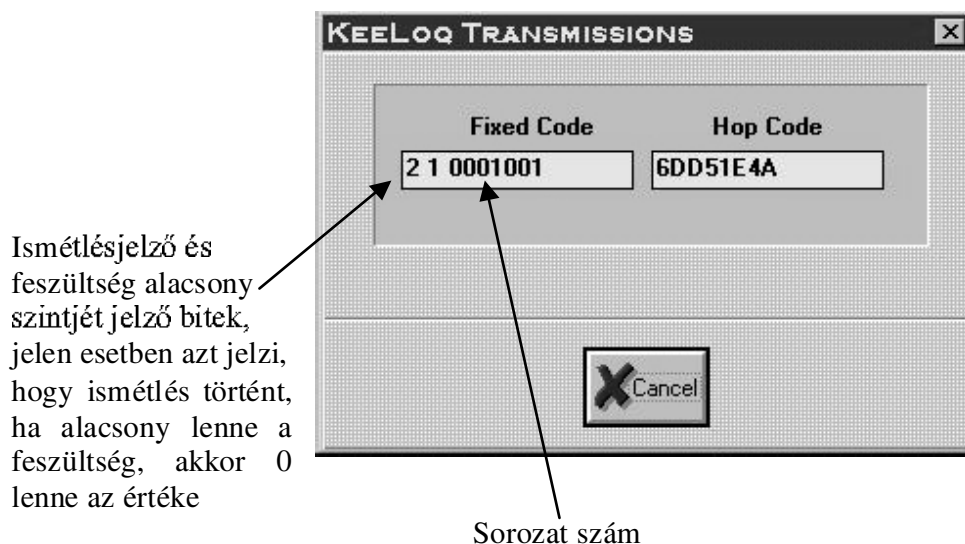
A tanulási eljárás:

1. Meg kell nyomni és fel kell engedni a LEARN gombot. Ekkor LEARN LED folyamatosan világítani fog, jelezve, hogy tanulási üzemmódban vagyunk.
2. Meg kell nyomni egy gombot az adón. A LED nem világít tovább.
3. Még egyszer meg kell nyomni egy gombot az adón. Ha sikeres a tanulás, akkor a LED villogni fog.
4. Az 1-3 lépéseket kell megismételni a többi adó megtanításához.
5. A tanítás sikertelen lesz, ha a két számláló értéke nem szekvenciális, vagy a második üzenet nem érkezik meg 33,6 másodpercen belül. A sikertelen tanítást a LED egy másodpercig való működése jelzi.
6. Az összes adó adatának a törlését a LEARN gomb kb. 8 másodpercig való nyomva tartásával érhetjük el. Ha megtörtént a törlés, akkor a LED kialszik.

14.8 KEELOQ ÁTVITELEK FIGYELÉSE

A program lehetővé teszi, hogy figyelemmel kísérjük az átvitelek változását. Ez úgy lehetséges, hogy a dekódolót csatlakoztatni kell a programozóhoz és a programban ki kell választani az aktuális menü pontot. Majd a kódokat megvizsgálva értelmezhetjük a látott információkat.





14.9 TAPASZTALATOK

A CHIPCAD cég által forgalmazott szett tanulmányozása és tesztelése során kis eltéréseket tapasztaltam az elméleti leírások és a gyakorlat között.

Ezek a következők:

- Titkos tanítás esetében először kell az ugró kódos információt tartalmazó üzenetet elküldeni, és csak utána kell a seed értéket.
- A funkció kódokat is ellenőrzi a normál tanításnál. Ez azt jelenti, hogy abban az esetben, ha a második üzenet elküldését nem ugyanazzal a nyomógombbal aktiváljuk, mint az első üzenet esetén, akkor a tanítás érvénytelené válik, és előlről lehet kezdeni az egész folyamatot.
- Az irodalom azt mondja, hogy a normál tanításnál a második üzenetben lévő számláló értékének és az előzőleg átmenetileg letárolt értéknek szekvenciálisnak kell lennie, ellenkező esetben a tanítás érvénytelen és előlről lehet kezdeni a tanítást. Azonban a gyakorlat azt mutatja, hogy ha az előbb említett két érték közötti különbség nem több mint 16, vagyis még az automatikus újraszinkronizáló állapotban van, akkor az automatikus újraszinkronizálás aktivizálódik és érvényes lesz a tanítás.

15 ÁBRAJEGYZÉK

1. Ábra. A KEELOQ alkotórészei	7
2. Ábra. Kulcskészítő eljárás és tárolás	11
3. Ábra. Az adó alapműködése	12
4. Ábra. A dekódoló alapműködése.....	13
5. Ábra. A HCS200 által készített üzenet összetétele.....	14
6. Ábra. A HCS300/301 által készített üzenet összetétele.....	15
7. Ábra. A HCS360/361 által készített üzenet összetétele.....	15
8. Ábra. A kódolók üzeneteinek összetétele a titkos tanítás során	15
9. Ábra. A KEELOQ eszközök kulcskészítése.....	19
10. Ábra. XOR és „titkosító” algoritmus használata a kulcskészítéshez.....	22
11. Ábra. Az üzenetküldés folyamata.....	23
12. Ábra. A KEELOQ kódoló blok diagramja	25
13. Ábra. A seed átvitelek átviteli formátuma és aktivizálási lehetősége.....	39
14. Ábra. Az átviteli szóban a „holt idő” megjelenése	43
15. Ábra. Az 1/6-2/6 és 1/3-2/3-os PWM formátum választás.....	44
16. Ábra. Az átvitelek összetétele.....	46
17. Ábra. A PWM modulálás	47
18. Ábra. Manchester modulálás	48
19. Ábra. PWM modulálás a kitöltési tényező változtatása	49
20. Ábra. A VPWM modulálás.....	50
21. Ábra. A szinkronizált átviteli mód.....	51
22. Ábra. A szinkronizált átviteli mód alatt az átvitel formátuma.....	52
23. Ábra. BACW választási lehetőségei.....	53
24. Ábra. BACW választási lehetőségei.....	54
25. Ábra. A HCS200 és HCS300/301 programozásának módja	55
26. Ábra. A HCS200 és HCS300/301 visszaellenőrzésének módja.....	55
27. Ábra. A HCS360/361 programozásának módja	57
28. Ábra. Érvényesítés folyamata.....	62
29. Ábra. Automatikus szinkronizáció	63
30. Ábra. Automatikus újra szinkronizáció és elutasítás	64
31. Ábra. Újra szinkronizáció.....	65
32. Ábra. A kódoló és dekódoló memóriája közötti összefüggés.....	66
33. Ábra. A dekódoló memóriája	67
34. Ábra. Normál tanulás (tanítás).....	69
35. Ábra. Titkos tanulás során az üzenetkészítés folyamata.....	70
36. Ábra. A titkos tanulás folyamata	71
37. Ábra. A HCS500 blokk diagramja.....	72
38. Ábra. A HCS512 blokk diagramja.....	72
39. Ábra. Az adat kimeneti formátuma	75
40. Ábra. A státusz üzenet formátuma.....	76

41. Ábra.	A HCS500 parancs üzemmódjának az aktiválása.....	80
42. Ábra.	Bájtok olvasása az EEPROM-ból.....	82
43. Ábra.	Bájtok írása az EEPROM-ba.....	83
44. Ábra.	Tanulás aktiválás a HCS500 esetében.....	83
45. Ábra.	Az első tanulási állapotot jelző üzenet.....	84
46. Ábra.	Az második tanulási állapotot jelző üzenet.....	84
47. Ábra.	A HCS500 programozásának módja.....	86
48. Ábra.	A HCS512 programozásának módja.....	88
49. Ábra.	A letöltött adat összetétele.....	88
50. Ábra.	A TEMIC által gyártott U2538B lábkiosztása.....	91
51. Ábra.	A TEMIC U2538B belső blokk-sémája.....	92
52. Ábra.	A TEMIC U2538B egy áramköri alkalmazási lehetősége.....	94
53. Ábra.	A CHIPCAD által forgalmazott RF adó.....	96
54. Ábra.	A CHIPCAD által forgalmazott RF vevő.....	97

16 TÁBLÁZAT JEGYZÉK

1. Táblázat.	Az ugró kódoknál használt kifejezések.....	9
2. Táblázat.	A seed átvitelek aktivizálási módjai kódoló típusonként.....	18
3. Táblázat.	Az átviteli kódok összetétele típusonként.....	27
4. Táblázat.	A működési jellemzők kódoló típusonként.....	27
5. Táblázat.	A HCS200-as láb kiosztása.....	28
6. Táblázat.	A HCS300/301 és HCS360/361 láb kiosztása.....	28
7. Táblázat.	A HCS200 memória kiosztása.....	29
8. Táblázat.	A HCS200 konfigurációs szavának az összetétele.....	30
9. Táblázat.	Baud rate választás.....	31
10. Táblázat.	A HCS300/301 memória kiosztása.....	31
11. Táblázat.	A HCS300/301 konfigurációs szavának az összetétele.....	33
12. Táblázat.	Baud rate választás.....	34
13. Táblázat.	A HCS360/361 memória kiosztása.....	35
14. Táblázat.	Baud rate választás.....	37
15. Táblázat.	A HCS360 konfigurációs szavának az összetétele.....	37
16. Táblázat.	A funkció kódok szerepe a HCS360/361 kódolóknál.....	38
17. Táblázat.	Jellemző késleltetési időtartamok.....	39
18. Táblázat.	Jellemző időtúllépési időtartamok.....	40
19. Táblázat.	A szinkronizáló számláló inicializáló értékei.....	41
20. Táblázat.	A HCS361 konfigurációs szavának az összetétele.....	42
21. Táblázat.	Jellemző késleltetési időtartamok.....	45
22. Táblázat.	Jellemző időtúllépési időtartamok.....	45
23. Táblázat.	Időtartamok a programozáshoz és az ellenőrzéshez a HCS200-nál.....	56
24. Táblázat.	Időtartamok a programozáshoz és az ellenőrzéshez a HCS300/301-nél.....	56
25. Táblázat.	Időtartamok a programozáshoz és az ellenőrzéshez a HCS360/361-nél.....	57
26. Táblázat.	A KEELOQ dekódolók összehasonlítása.....	73
27. Táblázat.	A HCS512 láb kiosztása.....	74
28. Táblázat.	A HCS500 láb kiosztása.....	75
29. Táblázat.	Státusz bitek jelentése.....	76
30. Táblázat.	A HCS500 konfigurációs bájtnak az összetétele.....	77
31. Táblázat.	A HCS512 konfigurációs bájtnak az összetétele.....	78
32. Táblázat.	A dekódoló parancsai.....	81
33. Táblázat.	A mindent törölő parancs értelmezési.....	85
34. Táblázat.	A programozáshoz szükséges időtartamok.....	88
35. Táblázat.	Az U2538B lábainak értelmezése.....	91

17 IRODALOMJEGYZÉK

- [1] KEELOQ® HCS300 Evulation kit – User`s guide
1996 Microchip kiadás
- [2] Introduction to KEELOQ®
1996 Microchip kiadás
- [3] Advanced KEELOQ®
1996 Microchip kiadás
- [4] HCS200/300/301/360/361 KEELOQ® Code Hopping Encoder
1996 Microchip kiadás
- [5] HCS500/512 KEELOQ® Code Hopping Decoder
1996 Microchip kiadás
- [6] Secure Data Products Handbook
1997/1998 Microchip kiadás
- [7] Rádiótechnika Elektronikai folyóirat 94/7, 94/8
- [8] Karel Novák: Rádióamatőrök barkácskönyve
Műszaki könyvkiadó, Budapest 1982
- [9] Magyar Béla: Rádiótechnikai zsebkönyv
Műszaki könyvkiadó, Budapest 1982
- [10] Elektronikai Szakismeretek 1-2 Híradástechnika
B+V Lap- és Könyvkiadó Kft., 1994