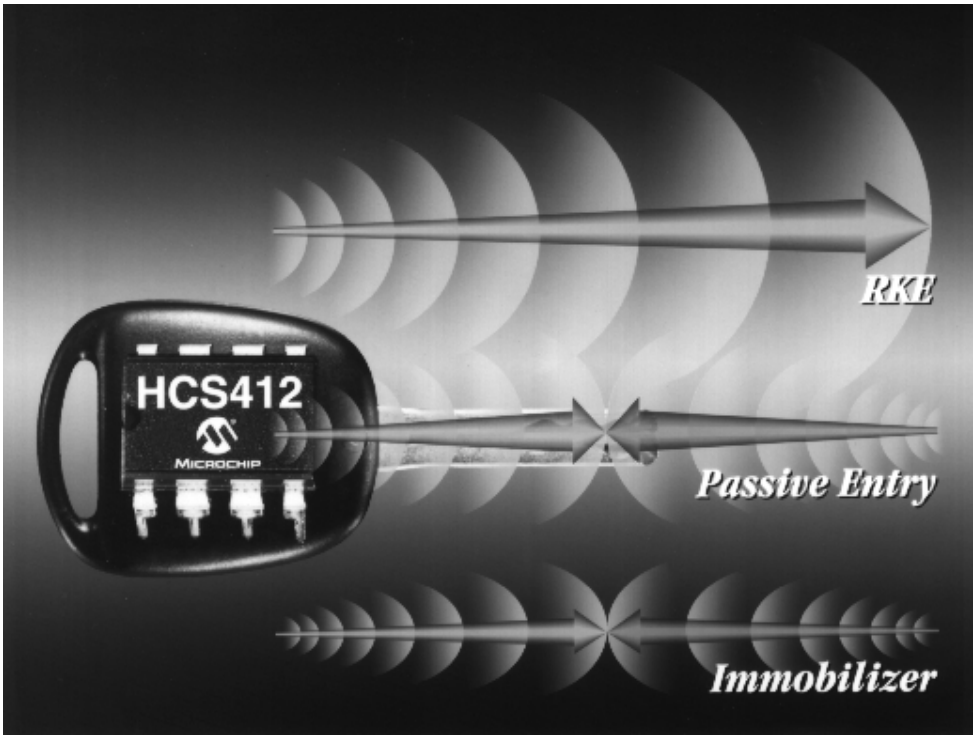
Jakó Attila
A Microchip HCS ugrókódos áramkörei



chipCAD
DISTRIBUTION

1 TARTALOMJEGYZÉK

2 BEVEZETÉS

Napjainkban nagyon népszerűek a rádiófrekvenciás és opto elektronikusan vezérelt távirányításos biztonsági rendszerek. Ilyen biztonsági rendszereket már számos alkalmazásban találhatunk, mint pl. járművek biztonsági rendszerei, vagy automatikus garázsajtó nyitók. A hagyományos távirányításos biztonsági berendezések egyirányú átviteleken alapszanak és a biztonsági szintjük nem a megfelelő. Már több fejlett eszköz (berendezés) alapszik kétirányú átvitelen, azonban nagyon költségesek, számos hátrányaik vannak és nem elég széleskörűen használhatóak.

Napjainkban általában a biztonsági rendszerek két igen fontos hiányossággal rendelkeznek: minden átvitel során állandó kódot küldenek el, és a kód kombinációk lehetőségének a száma relatívan kicsi. Ez utóbbi lehetővé teszi, hogy a kódok találgatásával, próbálgatásával a rendszert hatástalanítani lehet aránylag rövid idő alatt. Könnyen lehet készíteni egy mikrokontrolleres berendezést, amely ezt véghez is viszi. Tipikusan 20 próbálkozás lehet másodpercenként, ami annyit jelent egy 12 bites fix kódos rendszer esetében, hogy a kinyitásuk kevesebb, mint 5 percbe kerül. Ugyanilyen percenkénti próbálgatás mellett egy 16 bites rendszer esetében kevesebb, mint 2.5 órába kerül a berendezés hatástalanítása. A kódpróbálgatás elhárítható azzal, hogy a kód kombinációk lehetőségét kellően nagyra növeljük.

A másik gond általában a biztonsági rendszerekkel, hogy fix kódokkal működnek. Ekkor merül fel a kód rögzíthetőségének és visszajátszásának a problémája. Ahhoz, hogy jobban érthető legyen, lássuk a következőt. Egy normál távirányításos biztonsági rendszernél (pl. autóriasztó) van egy adó, mely elküldi a nyitáshoz szükséges kódot, amit ő generál és van egy vevő, mely az adó által kiadott jeleket veszi, értelmezi. Ezek egy meghatározott frekvencián működnek, melyek közismertek. Így bárkinek adott a lehetőség, hogy építsen egy olyan vevő berendezést, amely az adók jeleit fogja. Gondoljunk bele, hogy illetéktelen személy rendelkezik egy ilyen készülékkel és megvárja azt, hogy pl. egy autótulajdonos hatástalanítsa az autójának a riasztó berendezését a távirányítójával, akkor máris rögzíteni tudta a nyitáshoz szükséges kódot, melyet utána vissza is tud játszani.

Az előbbiek elhárításához a távirányításos biztonsági rendszereknek két alapvető tulajdonsággal kell rendelkezniük, mely tulajdonságokkal a KEELOQ ugró kódos rendszerek már rendelkeznek:

- A kód kombinációk száma kellőképpen nagy legyen.
  A KEELOQ technológiát alkalmazó eszközök 66 bites kódszó hosszal rendelkeznek. Egy 32 bites titkosított részből áll, mely több mint 4 billió kód kombinációt tesz lehetővé. A kódok próbálgatása ez által több mint 17 évbe kerül. Ha a 34 bites állandó részét és figyelembe vesszük, akkor az előbbi idő kb. 5,600 billió évre bővül.
- Sosem szabad ugyanazt a kódot kétszer elküldeni.
  Ez a rendszer olyan titkosító algoritmussal rendelkezik, amelyik lehetővé teszi, hogy ugyanaz a kód egy rendszer életében lehetőleg ne forduljon elő kétszer.

A rendszer azzal a tulajdonsággal rendelkezik még, hogy minden egyes gombnyomás hatására az előzőtől teljesen eltérő kódot küld el. Ez egy kívülálló számára úgy tűnik, hogy a rendszer véletlenszerűen készíti a kódokat. Nincs semmiféle látható összefüggés két egymás után elküldött kód között. Ha egy ilyen rendszert tekintünk, akkor legalább 65000 érvényes kód (üzenet) elküldése lehetséges anélkül, hogy egy is megismétlődne. Ha napi 8-szor működtetjük a rendszerünket, akkor kb. 22 évbe kerül, amíg újra megismétlődik ugyanaz a kód. Továbbá a már használt kódok nem tudják hatástalanítani a biztonsági berendezést.

A hosszú kód hossz és az állandóan változó kód szükségessé teszi az adó és a vevő állandó összehangolását. Ezt nevezzük szinkronizálásnak. A KEELOQ algoritmus is jellemzője a fejlett szinkronizációs technikának. A rendszer akkor is működni fog, ha az adónkat a vevő hatáskörén kívül folyamatosan működtetjük (pl. egy gyerek játszik a távvezérlővel). Ha a távirányító gombját több mint 16-szor lenyomjuk, akkor „elveszti” a szinkronizmusát a rendszer. Azonban ezt követően két egymást követő átvitel helyreállítja a szinkronizmust. Ilyenkor van az, hogy ha az adó működtetésére nem reagál a vevő, akkor a felhasználó természetes reakcióként még egyszer meg nyomja a nyomógombot (pl. autóriasztó), és ebben az esetben áll helyre az

előbb említett módon a szinkronizmus. A felhasználó ilyenkor nincs tudatában annak, hogy a rendszer „kibillent” a szinkronizmusból, majd visszaállt.

Ezek az előnyök egyértelműen észrevehetőek a hagyományos fix kódos rendszerekkel szemben. Továbbá a KEELOQ algoritmus jelentősége; kevés külső komponens szükséges a rendszer működtetéséhez, mivel egy IC-be vannak integrálva a fő részek.

A KEELOQ algoritmus egy 32 bit hosszú blokkon és egy 64 bites kulcson alapszik. Ennek a biztonságát rendkívül fontosnak tartották, azonban mára már nyilvános az algoritmus, mivel mikrokontrollerekkel is megvalósíthatók a dekódolók, ahhoz pedig szükség van, hogy bele programozzuk az üzembe helyezés előtt. Az információk tekintetében az adó adatai és szinkronizációs érték ezzel az algoritmussal kódolva vannak, így az egy kívülálló számára érthetetlen. A dekódoláshoz ugyanarra a 64 bites kulcsra van szükség, amit a kódolásnál használt a kódoló. Következésképpen még akkor is, ha a vevő (amelyik rendelkezik a kulccsal) képes beazonosítani az adót egy kívülálló nem tud mit kezdeni az adó által szolgáltatott információval. Még akkor sem tudja a kapott információkat felhasználni, ha folyamatosan rögzíti azokat, mivel ha a kódolás előtt álló adatok csak egy bitben is térnek el az előzőtől, a következő átvitel során az algoritmusnak köszönhetően teljesen különböző lesz az átvitel.

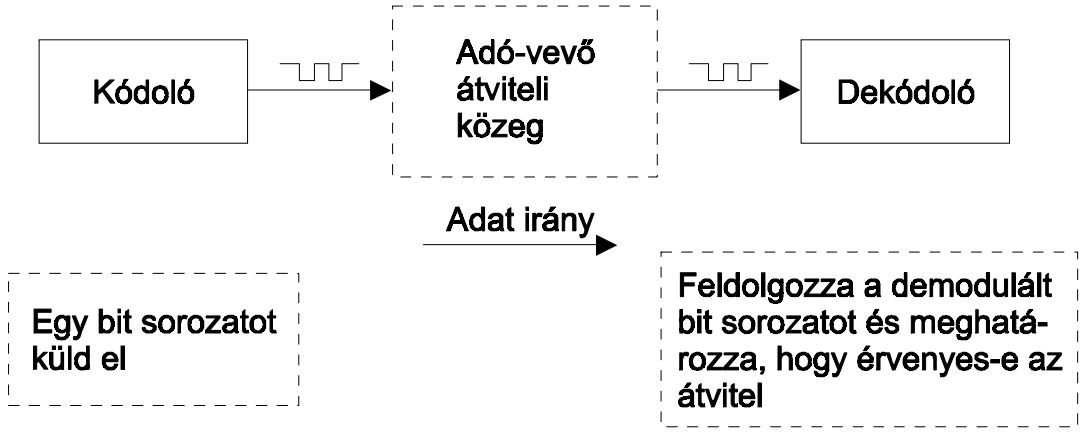
A rendszer működéséhez az adó és vevő eszközeinek egymásról bizonyos információkkal kell rendelkezniük ahhoz, hogy a rendszer működésképes legyen. Ehhez viszont egy úgynevezett tanítási eljárásra alkalmazására van szükségünk, mely igen fontos „szereplője” a rendszernek az üzemeltethetőség szempontjából. Ezen kívül még sok más, sajátosság van a rendszernek, melynek a bemutatását a következőkben láthatjuk.

A dolgozatomban ezt a KEELOQ technológiát és a technológiát megvalósító eszközöket, továbbá a működésükhöz szükséges perifériákat mutatom be részletesen.

3 A KEELOQ ALKOTÓELEMEI

A KEELOQ technológiát alkalmazó rendszerek bizonyos alapvető elemekből épülnek fel. Az átvitelhez feltétlenül szükséges egy adó, amihez csatlakozik a kódoló, továbbá az előző által kiadott jelek fogadására alkalmas vevő és hozzá csatlakozó dekódoló. A kódoló és dekódoló között lévő kapcsolat megteremtésének a legegyszerűbb lehetősége az, ha vezetékkel kötjük

össze a két egységet, de bizonyos esetekben és általában ennél többre van szükségünk, vagyis a vezetékmentes kapcsolatra. Ezt megvalósíthatjuk rádiófrekvenciás vagy optoelektronikus (infra) adatátviteli módszerekkel, ezért van szükség adóra és vevőre. Erre a dolgozatomban a későbbiekben még kitérek. Az előbb leírtakat az **1. ábrán** láthatjuk.



1. Ábra. A KEELOQ alkotórészei

A MICROCHIP által forgalmazott kódoló típusok a következők: HCS200, HCS300/301, HCS360/361 és HCS410. A HCS500 és HCS512 pedig a dekódolók.

A kódoló arra alkalmas, hogy a hozzácsatlakoztatott nyomógomb aktiválásának a hatására egy üzenetet készítsen, a belső memóriájában tárolt értékek és algoritmus segítségével. Ezt követően pedig a perifériaként hozzákötött adó működtetésével elküldje az előbb készített üzenetet a dekódoló számára.

A dekódoló a vevő periféria segítségével veszi a kódoló által elküldött üzenetet, majd azt feldolgozva (dekódolja, a kapott értékeket összehasonlítja a memóriájában tárolt értékekkel) érvényes üzenet érkezése esetében végrehajtja a kívánt feladatot.

4 UGRÓ KÓDOS RENDSZEREKNÉL HASZNÁLT DEFINÍCIÓK

Mielőtt az egész rendszer áttekintésébe kezdenénk szükséges egy-két fogalom, kifejezés értelmezése, magyarázata, melyek ismerete a továbbiak értelmezésének elengedhetetlen feltétele.

4.1 SOROZAT SZÁM

A kódoló, a sorozat számát minden egyes gomb lenyomás hatására elküldi. A sorozat szám az átvitel kódolatlan részének az egyik alkotó eleme, és a dekódolónak nyújt segítséget arra, hogy beazonosítsa, hogy melyik kódolótól kapott utasítást. Ezt a számot továbbá arra használjuk, hogy a tanítási eljárás során ennek a segítségével készítjük el a titkosító kulcsot.

4.2 TITKOSÍTÓ KULCS

A titkosító kulcs egy a gyártás során programozott és létrehozott egyedülálló 64 bites kulcs. A titkosító kulcs vezérli a titkosító algoritmust és tárolódik az elektromosan törölhető és újra programozható memóriában (EEPROM).

4.3 AZONOSÍTÓ ÉRTÉK

Ez szám gyártó által változtatható, programozható, de általában a sorozat szám alsó tíz bitjével egyezik meg. Az azonosító bitek az átvitel során a titkosított információ részét képzik. Azután, hogy a vevő dekódolta a megkapott üzenetet, összehasonlítja ezeket a biteket a tárolt értékkel, hogy a dekódoló eljárás és az üzenet érvényes-e.

4.4 SZINKRONIZÁLÓ SZÁMLÁLÓ

Az átviteli szó tartalmaz egy 16 bites szinkronizáló számláló által szolgáltatott értéket. Ezt az információt is arra használja fel a dekódoló, hogy megállapítsa az átvitelek érvényességét. A már szerepelt kódokat (számláló értékeket) elutasítja az esetleges kódrögzítések kiszűrése miatt. (A HCS300/301-es kódolók túlcsordulást jelző biteket küldenek el mindig, ezáltal a szinkronizáló számláló értékét kibővítik 65536-ról 196608-ra. A HCS360 és HCS361-es kódolók csak egy túlcsordulást jelző bitet küldenek el, amellyel a számláló tartományát csak 131071-re terjeszti ki.)

4.5 GYÁRTÓ KÓD

A gyári kód egy 64 bites szó, mely egyedi lehet minden egyes darabnál. Azonban rendszerenként meg kell egyeznie. Ennek a következménye az egyedülálló titkosító kulcs minden átvitel során. Ezt a kódot a dekódolóba a gyártás (programozás) folyamán kell bele rakni és nem pedig a tanításkor.

4.6 SEED

A Microchip által forgalmazott kódolók mindegyike rendelkezik a seed elküldésének képességével. A seed tulajdonképpen egy véletlenszerűen generált szám, mely értéket akkor kell beprogramoznunk a kódolók memóriájába, amikor először inicializáljuk az eszközt, a számlálóval, a kulccsal, a sorozat számmal és más információkkal együtt. A seed hossza kódolónkként más és más. A HCS200, HCS300 és HCS301 seed értéke 32 bites, míg a HCS360 és HCS361-é 48 bites.

Az előzőeket és egyéb jellemző kifejezések rövid leírását lássuk az **1. táblázatban** összefoglalva:

1. Táblázat. Az ugró kódoknál használt kifejezések

Adó sorozat száma	Minden adó gyártáskor felprogramozható egy egyedülálló 28 vagy 32 bites sorozat számmal. Ez biztosítja minden kódoló számára, hogy egyedi legyen egy rendszeren belül.
Titkosító kulcs	Ez egy 64 bites szám, melyet a kulcskészítő algoritmus generál a 28- vagy 32 bites sorozat számból, vagy a 32- vagy 48 bites seed értékből és a 64 bites gyártó kódból, mint bemenet. A titkosító kulcs nem olvasható és soha nem kerül átvitelre.
Seed	A seed egy 32- vagy 48 bites véletlenszerűen generált érték, melyet beprogramozunk a kódolóba. Ezt az értéket használja a kulcskészítő algoritmus a titkos tanítás esetén. Csak egy speciális nyomógomb kombináció hatására aktiválódik és ekkor kerül átvitelre.
Kulcskészítés	A kulcskészítő funkciót arra használja a kódoló, hogy egyedülálló kulcsot készítsen minden egyes adónak a sorozat számból vagy a seed értékből.
Gyártó kód	A gyártó kódot a vevőnek tartalmaznia kell, ahhoz hogy a titkosító kulcsot elkészítse. A gyártó kódot a gyártás során kell beprogramozni a dekódolóba.
Normál tanítás (sorozat szám forrású)	A vevő ugyanazokat az információkat használja a kulcskészítéshez, mint az adó a normál működés alatt. Ekkor dekódolja az azonosító értéket és a szinkronizáló számlálót, és ezután a vevőben tárolódik a kódoló minden adata.
Titkos tanítás (seed forrású)	Az adó egy speciális nyomógomb kombináció hatására aktiválódik és elküldi az EEPROM-jában tárolt 32- vagy 48 bites értéket (seed), hogy a dekódoló ezt használja a kulcskészítéshez. Ez a funkció letiltódik a tanulás befejeztével.
Azonosító érték	Az azonosító érték a titkosított adatnak egy 10 vagy 12 bites fix részét képzi. Ezt a titkosítás utáni ellenőrzésre használja a rendszer.
Szinkronizáló számláló	Egy 16 bites számláló, amely a kódoló minden egyes aktiválásakor inkrementálódik. Ez az érték a vevő memóriájában is tárolódik és minden egyes átvitelnél az előzően tárolt értékkel van összehasonlítva. Tulajdonképpen ez képzi az ugró kód alapját, emiatt fog minden egyes esetben változni a kód.

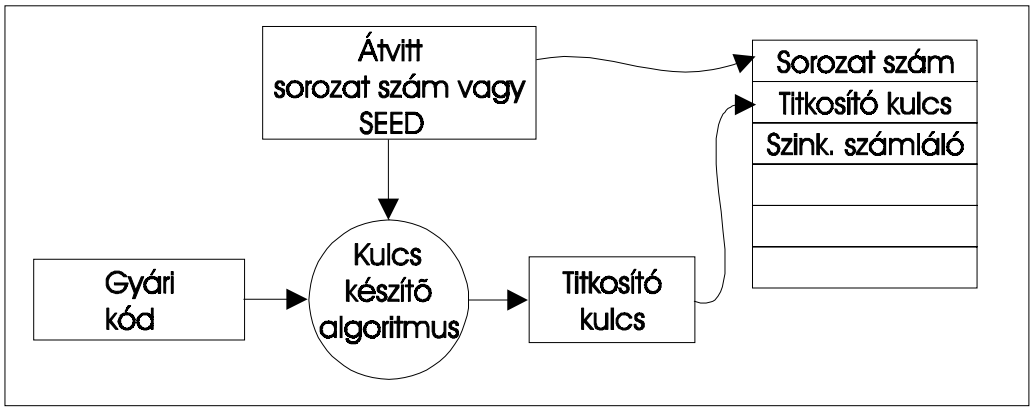
5 A RENDSZER ÁTTEKINTÉSE

A KEELOQ ugró kódos rendszer egy olyan módszert használ fel, mely során minden gombnyomás esetén mindig különböző kódot küld el az adó a vevőnek. A **2. ábrán** látható, hogy az ugró kódos eszközök egy nem túl nagy EEPROM-ot tartalmaznak, melyet a rendszerbe illesztés előtt fel kell programozni néhány, a működéshez szükséges paraméterrel.

Ezek közül a legfontosabbak a következők:

- 28 bites sorozat szám
- titkosító kulcs, mely a gyártás (programozás) során készül el
- 16 bites szinkronozó érték, mely a számláló aktuális állapota és alapból 0-ról indul

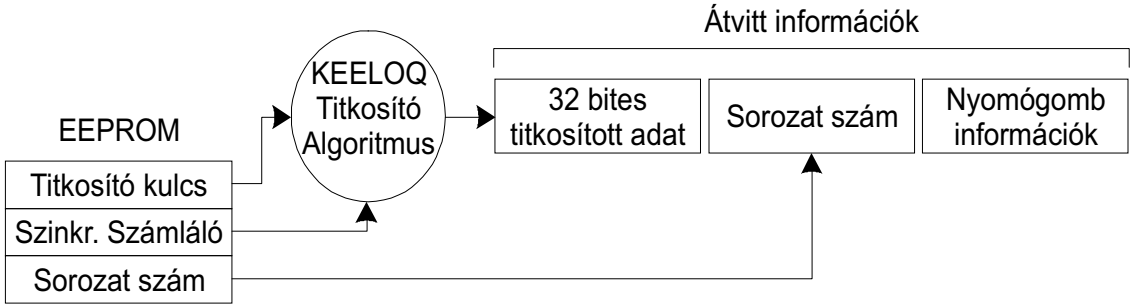
A sorozat számot és a gyártó kódot minden kódoló esetében a gyártás során programozzák be. A titkosító kulcs, a kulcs generáló algoritmus használata során készül (lásd. **2. ábra**) a sorozat számból és a gyártónak a 64 bites kódjából. A 16 bites szinkronozó érték az alapja annak, hogy minden gombnyomásra változni fog az átvitelünk. Igaz, hogy ez a szám csak egy bit-ben tér el az előzőtől, de a titkosító algoritmusnak köszönhetően a következő átviteli kód már sokban fog különbözni.



2. Ábra. Kulcskészítő eljárás és tárolás

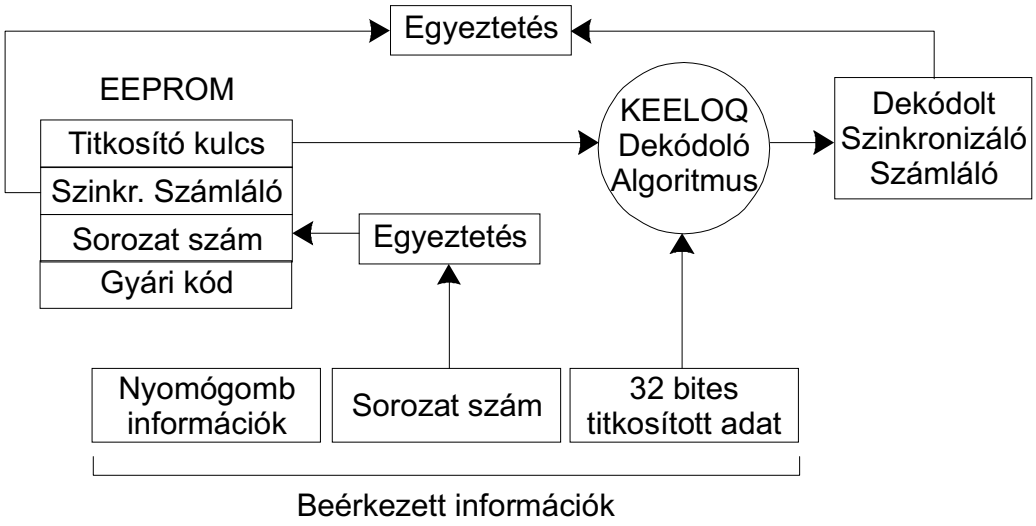
A **3. ábrán** látható az összefüggés az EEPROM-ban tárolt értékek között, és hogy a kódoló hogyan használja fel ezeket. A folyamat úgy történik, hogy először érzékeli a kódoló, hogy

lenyomtak egy gombot, majd megvizsgálja, hogy melyiket és utána felfrissíti a szinkronizáló számláló értékét. Ezt követően a számláló értékéből és a titkosító kulcsból a KEELOQ algoritmus képez egy 32 bites titkosított adatot. Ez lesz a kód szó egyik fele, úgynevezett ugró kódos része. A másik fele (fix rész) a sorozat számból és a lenyomott gombok kombinációjából tevődik össze.



3. Ábra. Az adó alapműködése

A kódoló használata előtt meg kell tanítani a vevőt. Ez azt jelenti, hogy el kell tárolni a vevőnek ugyanazokat az információkat, amelyeket az adó használ. Ezek az értékek az adó sorozat száma, a szinkronizáló számláló értéke, a titkosító kulcs és az azonosító szám. Abban az esetben, ha a dekódolót már megtanítottuk a kódoló paramétereire és vesz egy érvényes üzenetet, akkor először ellenőrzi a sorozat számot, hogy megegyezik-e az EEPROM-ban eltárolt értékkel, majd a dekódoló algoritmus dekódolja a 32 bites kódolt adatot a memóriájában tárolt kódoló kulcs segítségével, melyet előtte már a sorozat számból és a gyártó kulcsból generált. Az így kapott értéket összehasonlítja a saját szinkronizáló számlálójának az értékével. Ha ez az ellenőrzés befejeződik és megegyezik a két szám, akkor megvizsgálja, hogy milyen funkciót kell végrehajtson a lenyomott gombok kombinációjának megfelelően. Az alábbi összefüggések grafikusan láthatóak a **4. ábrán**.

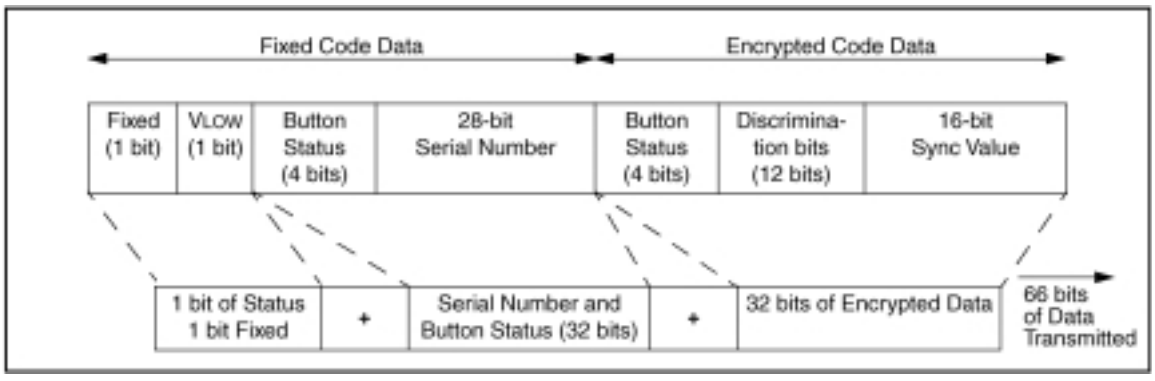


4. Ábra. A dekódoló alapműködése

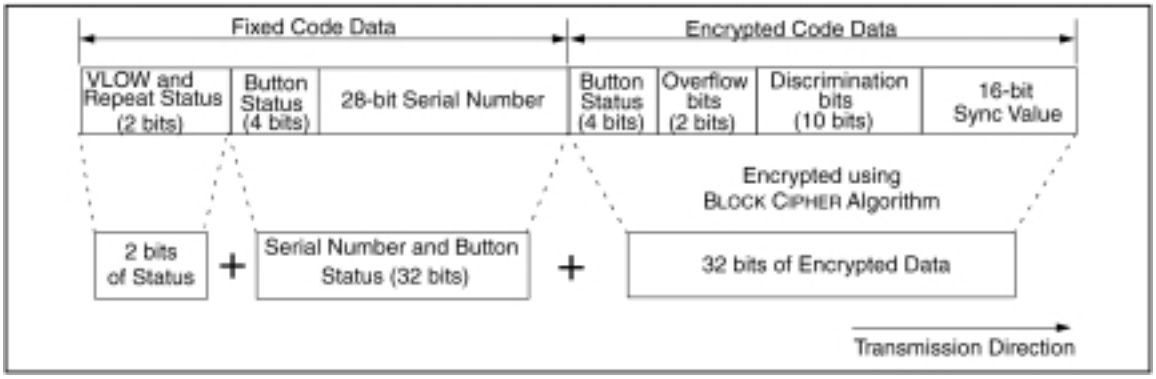
6 AZ ÜZENETEK (ÁTVITELEK) ÖSSZETÉTELE

A kód szó összetételének elkészítése automatikus jellemző, azért hogy biztos legyen, hogy a teljes kód átvitele megtörténik, még abban az esetben is, ha a nyomógombot hamarabb felengedjük, mint ahogy az átvitel befejeződött volna. A kódoló feléled, ha egy nyomógomb lenyomás történt és alapállapotba kerül, ha a parancs végrehajtása befejeződött (ha a használó felengedte közben a gombot). Ha a nyomógomb egy átviteli időn túl nyomva van, akkor többszörös átvitel lesz az eredmény. Ha másik gomb lenyomása történt az átvitel alatt, akkor az éppen aktív átvitel azonnal leáll és az új nyomógomb kombinációnak megfelelő új kód generálódik.

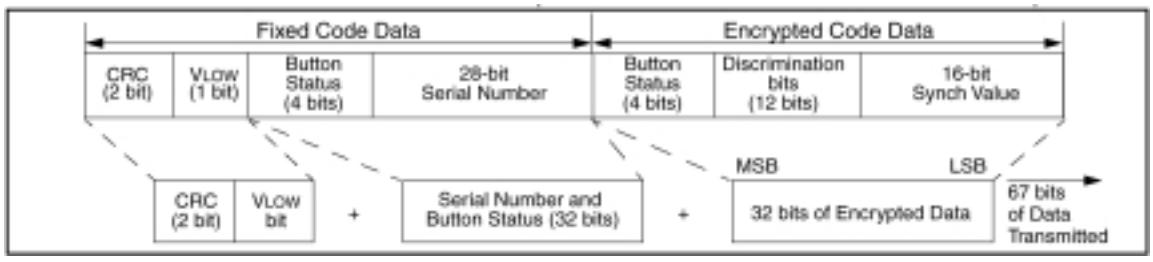
A kódolók által készített 66/67 bites átvitelek (üzenetek) két részből tevődnek össze. Az egyik része mindig változik, amikor az adókat aktiváljuk. Ezt nevezzük az ugró kódos résznek. A másik része tulajdonképpen a kódoló sorozat számából és egyéb bitekből áll, ezt pedig fix résznek nevezzük. Az üzenetek összetételét kódoló típusokra lebontva kell vizsgáljuk, mivel eltérnek egymástól. A **5., 6. és 7. ábrán** láthatjuk grafikusan, hogy kódolónkként miből és hogyan épül fel az üzenet két része.



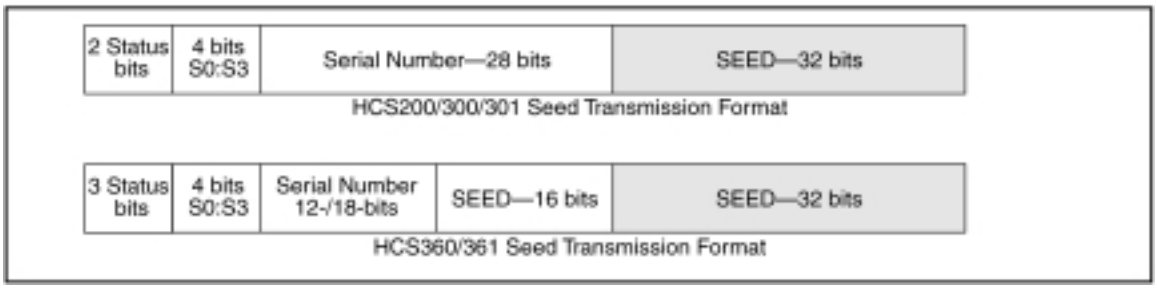
5. Ábra. A HCS200 által készített üzenet összetétele



6. Ábra. A HCS300/301 által készített üzenet összetétele



7. Ábra. A HCS360/361 által készített üzenet összetétele



8. Ábra. A kódolók üzeneteinek összetétele a titkos tanítás során

Az ábrákból kitűnik, hogy a HCS200 és HCS300/301 kódolóknál 66 bites az átvitel, míg a HCS360/361-nél 67 bites. A következőkben pedig részletezzük, hogy miből is épülnek az előbb

említett részek. Az üzenetnek azon részeinek az értelmezésére térnék ki, melyek még nem ismeretesek a számunkra.

6.1 UGRÓ KÓD

Az ugró kódos (titkosított) rész alapvetően tartalmazza végrehajtandó feladat funkció kódjait, az azonosító értéket és a szinkronizáló számláló aktuális értékét. A HCS300/301 kódolóknál kiegészül úgynevezett túlcsordulást jelző bitekkel. Ezeket az adatokat a kódoló minden egyes átvitel előtt lekódolja a titkosító algoritmus alapján (a titkosítás folyamatát a későbbiekben láthatjuk). Ha titkos tanításról beszélünk, akkor egy úgynevezett seed érték képezi teljes mértékben az ugró kódos részét az üzenetnek. Sőt a HCS360/361 kódolóknál a fix kód egy részét is (**8. ábra)**

6.1.1 FUNKCIÓ INFORMÁCIÓK

A kódoló minden egyes átvitel során elküldi a végrehajtandó feladatnak megfelelő nyomógomb kombinációkat, vagyis a funkció kódot. Összesen négy nyomógomb található általában az eszközökön (HCS200-on három van), ami annyit tesz, hogy összesen 15 (HCS200-nál 7) különböző feladatot lehet végrehajtatni a kódolóval. Ez tulajdonképpen 14, mert 1 a seed átvitel aktivizálására szolgál.

6.1.2 TÚLCSORDULÁST JELZŐ BITEK

Ezeket a biteket a szinkronizáló számláló felső határának a kiterjesztésére használja a berendezés. A számláló 16 bites, mely 65536 ciklus lehetőséget biztosít, mielőtt megismétlődnének a számok. A tapasztalatok azt mutatták, hogy a megfelelő biztonság és működési élettartam számára ez nem elegendő és ezért létrehozták ezeket a biteket, hogy kibővíthesse a számlálási tartományt. Ezeknek a biteknek a működése programozó által szabályozható.

6.2 FIX KÓD

A fix kód az a része az üzenetnek, mely kódolatlanul állandóan benne szerepel az átvitelben. Alapvetően a sorozat számból, a funkció kódokból és típusonként változó státusz bitekből tevődik össze. Ezekek a státusz biteknek az értelmezését lássuk a következőkben.

6.2.1 VLOW: Alacsony feszültségszintjelző

A VLOW bit minden egyes kódoló típusnál megtalálható. Értéke egyes lesz, ha a működési feszültség az alacsony feszültség szint alá esik. Ez az érték két érték közül választható, a használt feszültség forráson alapszik. A VLOW jel átvitelre kerül a vevőnek, így „hallható" jelet adhat a használónak, hogy az adó tápforrásának a feszültség szintje alacsony.

6.2.2 RPT: ISMÉTLÉS JELZŐ

Ez csak a HCS300/301-es kódolónál van. Ez a bit az első kód szó átvitel során alacsony szintben lesz. Ha a nyomógombot tovább tartjuk nyomva, mint egy kód szó átvitele, akkor az előbb említett bit állítódik, ezzel jelezve az ismétlődést és megmarad ez az állapot, amíg a gombot nyomva tartjuk.

6.2.3 CRC (Ciklus redundancia ellenőrző) bitek

A CRC biteket az előtte lévő 65 bitre számolja ki. A vevő használja fel, hogy ellenőrizze az adatok integritását azelőtt, hogy elkezdené az üzenet feldolgozását. Ennek a segítségével az egy bitben való eltérést, hibát 100%-osan és a két bitben való hibát 66%-osan képes felismerni. Ez az adottság a HCS360/361-es kódolókban található meg.

6.3 SEED ÁTVITELEK AKTIVIZÁLÁSA

A seed átvitelek során az üzenet összetételéről már szó volt, most lássuk az szükséges az aktivizálásukhoz (**2. táblázatot**).

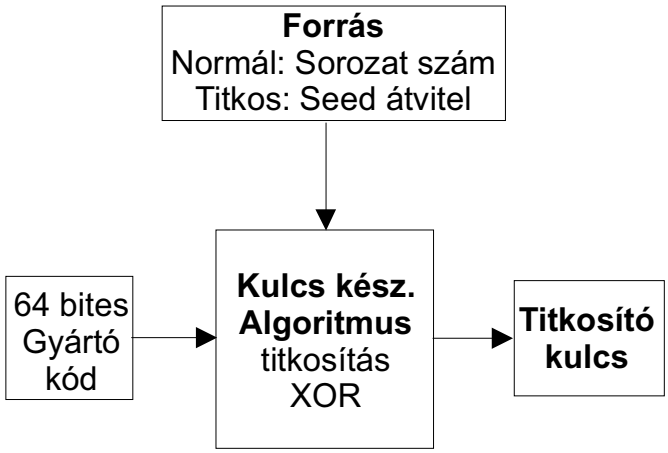
2. Táblázat. A seed átvitelek aktivizálási módjai kódoló típusonként

Kódolók	Seed hossz	Seed átvitel aktiválása
HCS200	32 bit	Seed átvitel azonnal, ha az S0, S1 és S2 egyszerre aktív
HCS300	32 bit	Seed átvitel azonnal, ha az S0, S1, S2 és S3 egyszerre aktív
HCS301	32 bit	Seed átvitel azonnal, ha az S0, S1, S2 és S3 egyszerre aktív
HCS360	48 bit	Seed átvitel azonnal, ha az S0 és S3 egyszerre aktív Seed átvitel 3 másodperc múlva, ha az S0 és S1 egyszerre aktív több mint 3 másodpercig
HCS361	48 bit	Seed átvitel azonnal, ha az S0 és S3 egyszerre aktív Seed átvitel 3 másodperc múlva, ha az S0 és S1 egyszerre aktív több mint 3 másodpercig

7 A KEELOQ KÓDOLÓK ÉS DEKÓDOLÓK TITKOSÍTÓKULCS KÉSZÍTÉSEI

Az előzőekben már szó volt titkosításokról, kódoló algoritmusokról. Most vizsgáljuk meg ezeket a folyamatokat részletesebben.

A KEELOQ technológiát alkalmazó berendezések egy speciális titkosító algoritmust használnak az üzenetek kódolására és dekódolására. Ez az algoritmus a kódolóban és a dekódolókban is egyaránt megtalálható, csak az egyik a paraméterek kódolására (KEELOQ kódoló algoritmus), a másik pedig az üzenetek dekódolására (KEELOQ dekódoló algoritmus) használ egy 64 bites titkosító kulcsot. Ennek a kulcsnak az elkészítéséhez szükség van egy úgynevezett titkosító kulcs elkészítő algoritmusra. Azonban meg kell, hogy jegyezzük, hogy a kulcsot elkészítő algoritmus különbözik az előző algoritmusoktól.



9. Ábra. A KEELOQ eszközök kulcskészítése

A kulcskészítési eljárás három alapelemből épül fel, mint ahogy azt a **9. ábra** is mutatja. Az egyik része, a tulajdonképpeni forrása a kulcskészítésnek. Ez rendszerint egy egyedülálló, pontosabban egyedi minden egyes kódolónál úgy, mint például a sorozat szám vagy a seed érték.

A második része az algoritmus, amely elkészíti a titkosító kulcsot. Végezetül a hármadik alkotóelem a 64 bites gyártó kulcs, amely a másik bemeneti eleme az algoritmusnak.

A kulcskészítő eljárásban a gyártó kód használata kapcsolatot teremt a kódoló sorozat száma és a kódoló/dekódoló kulcs között. Ez biztosítja a gyártónak, hogy olyan kódolókat készítsen, melyeket a konkurencia (illetéktelen személy) nem tud lemásolni. Ezért a gyártó kód biztonsága, titokban tartása egy kritikus pontja a rendszer biztonságának a megőrzésében. Ezért van az is, hogy nem olvashatóak a kódolóban és dekódolóban letárolt adatok.

A KEELOQ termékek két forrást használhatnak a titkosító kulcs elkészítéséhez (**9. ábra**). A sorozat számot abban az esetben használják, hogy ha normál tanításról van szó. Titkos tanítás esetében a HCSXXX kódolóknak van egy olyan képessége, hogy egy fix seed értéket továbbítsanak, ekkor ezt használják a titkosító kulcs generálásához.

A továbbiakban tekintsük egy kicsit a dekódoló oldaláról a titkosító kulcsok különböző elkészítési lehetőségeit.

Kétféle kulcskészítő algoritmus ismeretes. Az egyik egyszerűen úgy nevezik, hogy „titkosító algoritmus”, a másik pedig XOR algoritmus.

Abban az esetben, ha normál tanításról van szó, vagyis a sorozat szám az algoritmus egyik bemeneti eleme, akkor csak az előbb említett „titkosító algoritmust” használhatjuk a kulcs elkészítéséhez. Viszont a titkos tanítás esetén mindkét algoritmust alkalmazhatjuk, választhatjuk.

7.1 TITKOS KULCSKÉSZÍTÉS (SEED A FORRÁS)

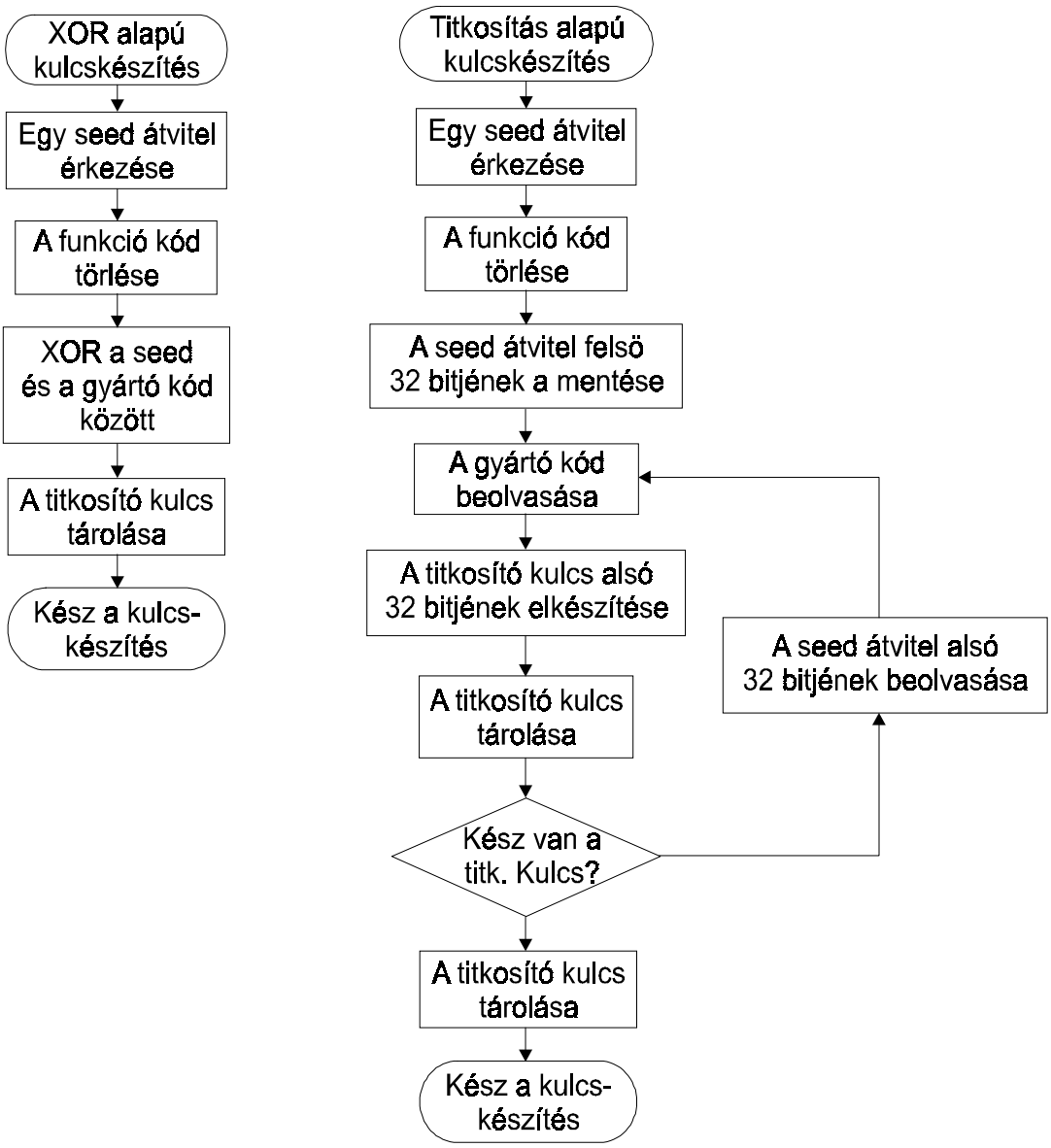
A titkos kulcs generálása a kódolónak a seed átvitelén alapszik, amely az algoritmus forrása és a tanítási folyamat alatt készül. A Microchip HCSXXX kódolók képesek az üzenet ugró kódos részén egy 32 vagy 48 bites seed átvitelére. A seed átvitel során a kód szó tartalmazza magát a seed értékét, a sorozat számát a kódolónak és a kódolatlan funkció kódokat. A véletlenszerűen generált seed érték csak a tanulási procedúra alatt továbbítódik. Ezért utána hiába kapják el az üzeneteket, nem tudják dekódolni azokat, mert nem a sorozat számból készült a titkosító kulcs.

7.1.1 XOR algoritmus használata a kulcskészítéshez

Az algoritmus használata során a funkció kód törlődik az átvitelből, majd az így kapott értékkel és a gyártó kóddal XOR műveletet hajtunk végre. Az eljárás során kapott szám lesz a titkosító (dekódoló) kulcsunk, amivel dekódoljuk a kapott üzenet ugró kódos részét.(**10. ábra**)

7.1.2 „Titkosító" algoritmus használata a kulcskészítéshez

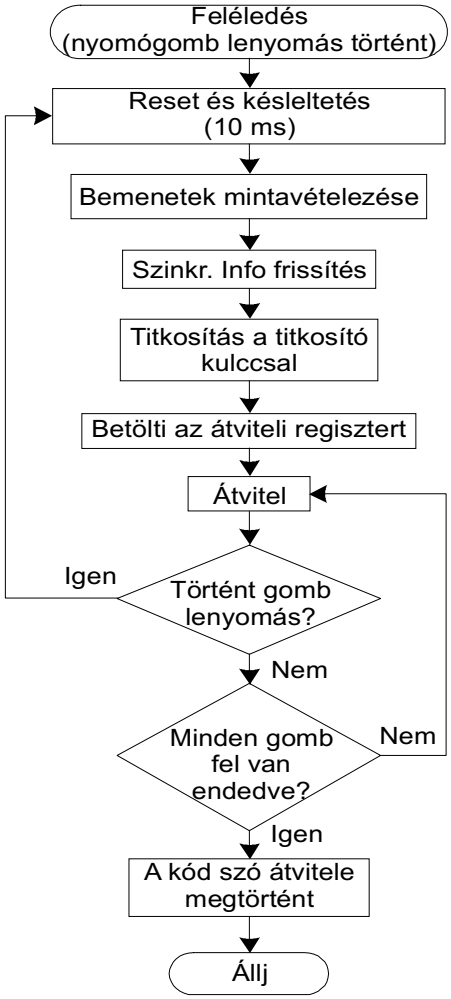
Ennek az algoritmusnak a használata során is az átvitelből a kódolatlan funkció kódok törlődnek. A **10. ábrán** is láthatjuk, hogy utána elmenti az átvitel felső 32 bitjét, majd beolvassa a gyártó kódot. Ezt követően elkészíti a titkosító kulcs alsó 32 bitjét és elmenti azt. Beolvassa az átvitel felső 32 bitjét, amit átmenetileg előzőleg eltárolt és ebből pedig elkészíti a kulcs másik részét.



10. Ábra. XOR és „titkosító” algoritmus használata a kulcskészítéshez

8 A KEELOQ KÓDOLÓK ÜZENETKÜLDÉSÉNEK A FOLYAMATA

Az alkalmazott algoritmus biztonságossá teszi az információt, ami azt jelenti, hogy minden egyes használat után a következő kód totálisan eltér az előzőtől. Statisztika bizonyítja, hogy ha 32 bites string-nél 1 bit változik, akkor kb. 50 %-os változás történik a kódolt átvitelnél. A kódolók gombnyomásra felélednek, majd egy kb. 10 ms-os késleltetés történik (**11. ábra**).

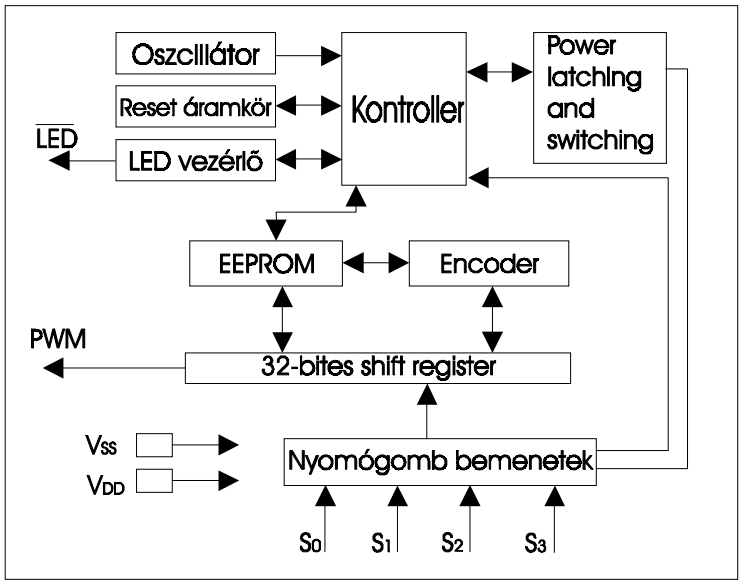


11. Ábra. Az üzenetküldés folyamata

Ezek után beolvassa a bemeneteket, hogy melyik gombot nyomtuk le, majd frissíti (növeli) a számláló értékét. Utána a már ismert paraméterekkel megtörténik a titkosítás. Abban az esetben, ha ugyanazt a gombot nyomtuk meg kétszer egymás után akkor is más információ fog átvitelre kerülni, vagyis akkor is változik a kód. Ha nyomva tartjuk a gombot abban az esetben az történik, hogy ugyanazt a kódot küldi el addig míg abba nem hagyjuk, vagy időtúllépés nem történik. Ha átvitel közben új gombnyomás történik, abban az esetben azonnal alapállapotba áll a rendszer, és a kód szó nem lesz teljes.

9 A KEELOQ KÓDOLÓK

A KEELOQ kódolók belső sematikus felépítését a **12. ábrán** láthatjuk. Ez az ábra a HCS200-as kódoló esetén abban tér el, hogy nincs LED vezérlő kimenet és csak három nyomógomb bemenete van.



12. Ábra. A KEELOQ kódolók blokk diagramja

9.1 ÁLTALÁNOS JELLEMZŐK

Biztonság

- Programozható 28/32 bites sorozat szám
- Programozható 64 bites titkosító kulcs
- Minden átvitel egyedi

- 66/67 bites átviteli kód hossz
- 32 bites ugró kód
- 34/35 bites fix kód (28/32 bites sorozat szám, 4/0 bit funkció kód, Vlow, Repeat/2 CRC bit)
- A titkosító kulcs olvasás védett

Működés

- 3 vagy 4 nyomógomb bemenet – 7 vagy 15 funkciót lehet megvalósítani
- Választható baud rate
- Automatikus power down az átvitel után
- Alacsony feszültség jel küldése a vevőnek
- IR modulációs lehetőség

Egyéb műszaki paraméterek

- Chip-be van építve az EEPROM, oszcillátor és időzítő komponensek, valamint a reset áramkör
- A nyomógomb bemenetek belső lehúzó ellenállásokat tartalmaznak
- LED vezérlő kimenet

A következő két táblázatban az előbb általánosan leírtakat eszközökre szétbontva láthatjuk. A **3. táblázat** az átviteli kódok összetételét, míg a **4. táblázat** a működési jellemzőket mutatja részletesen.

3. Táblázat. Az átviteli kódok összetétele típusonként

	HCS200	**HCS300/301**	**HCS360/361**
Teljes átvitel hossza	66 bit	66 bit	67 bit
Ugró kódos rész (teljes hossz)	32 bit	32 bit	32 bit
Funkció bitek	3 bit	4 bit	4 bit
Azonosító bitek	12 bit	12 bit	12 bit
Szinkronizáló bitek	16 bit	16 bit	16 bit
Fix rész (teljes hossz)	34 bit	34 bit	35 bit
Sorozat szám	28 bit	28 bit	28/32 bit[2]
Funkció bitek	3 bit	4 bit	4/0 bit[2]
Státusz bitek			
Vlow	1 bit	1 bit	1 bit
Repeat	0	1 bit	0
CRC	0	0	2 bit

[2]Felhasználó által megválasztható, hogy 28 vagy 32 bites legyen a sorozat szám és ennek megfelelően változik a funkció bitek száma is.

4. Táblázat. A működési jellemzők kódoló típusonként

Kódolók	HCS200	HCS300	HCS301	HCS360	HCS361
Bemenetek	3	4	4	4	4
Funkciók	7	15	15	15	15
Feszültség	3.5-13V	2.0-6.3V	3.5-13V	2.0-6.6V	2.0-6.6V
Baud rate	2	3	3	2	2
LED kimenet	Nincs	Van	Van	Van	Van
Idő túllépés	Van	Választható	Választható	Választható	Választható
Modulálás	PWM	PWM	PWM	PWM Manchester	PWM VPWM
Egyéb jellemző	Alacsony feszültség szint jelzése	Túlcsordulást jelző bitek, Alacsony feszültség szint jelzése	Ugyanaz, mint a HCS300	IR mód, 2 db önálló számláló, 2 CRC bit	IR mód, 2 db önálló számláló, 2 CRC bit

A továbbiakban a HCS300 és HCS301 kódolót együtt tárgyaljuk, mivel az előző táblázatban is láthatóan csak az üzemeltetési feszültség szintben különböznek egymástól.

9.2 LÁBKIOSZTÁS

9.2.1 HCS200-as kódoló

5. Táblázat. A HCS200-as lábkiosztása

Elnevezés	Láb szám	Leírás
S0	1	Kapcsoló bemenet 0
S1	2	Kapcsoló bemenet 1
S2	3	Kapcsoló bemenet 2/ Órajel láb a programozáskor
V_{SS}	5	Föld pont csatlakozó
PWM	6	Impulzus szélesség modulált kimenet/ Adat láb a programozáskor
V_{DD}	8	Pozitív tápfeszültség pont csatlakozó

9.2.2 HCS300/301 és HCS360/361 kódoló

6. Táblázat. A HCS300/301 és HCS360/361 lábkiosztása

Név	Láb szám	Leírás
S_0	1	Kapcsoló bemenet 0
S_1	2	Kapcsoló bemenet 1
S_2	3	Kapcsoló bemenet 2/ programozáskor óra láb lehet
S_3	4	Kapcsoló bemenet 3/ programozáskor óra láb
V_{SS}	5	Föld pont csatlakozó
PWM	6	Impulzus szélesség modulált kimenet/ programozáskor adat láb
/LED	7	Katód csatlakozású direkt vezérelt LED kimenet az átvitel alatt
V_{DD}	8	Pozitív tápfeszültségű pont csatlakozó

9.3 EEPROM MEMÓRIA KIOSZTÁS

9.3.1 HCS200-as kódoló

HCS200 memóriája 192 bites (12x16 bit (szó), **7. táblázat**). Ezt használjuk a már előzőekben tárgyalt adatok tárolására. A memória további leírását a következőkben tekinthetjük meg. Csak azoknak a táblázatokban található cellák értelmezésével foglalkozunk, amelyiket még nem ismerjük.

7. Táblázat. A HCS200 memória kiosztása

SZÓ CÍM	MNEMONIC	MEGNEVEZÉS
0	KEY_0	64 bites titkosító kulcs (word 0)
1	KEY_1	64 bites titkosító kulcs (word 1)
2	KEY_2	64 bites titkosító kulcs (word 2)
3	KEY_3	64 bites titkosító kulcs (word 3)
4	SYNC	16 bites szinkronozó érték
5	RESERVED	0000H címre beállítás
6	SER_0	Az eszköz sorozat száma (word 0)
7	SER_1	Az eszköz sorozat száma (word 1)
8	SEED_0	Seed érték (word 0)
9	SEED_1	Seed érték (word 1)
10	RESERVED	0000H címre beállítás
11	CONFIG	Konfigurációs szó

Konfigurációs szó

A konfigurációs szó az egy 16 bites EEPROM-ban tárolt érték, melyet az eszköz használ, hogy a titkosító eljárás alatt információkat tároljon, valamint a beállítások állapotát mutatja. Az egyes bitek további értelmezését lássuk a **8. táblázatban**.

8. Táblázat. A HCS200 konfigurációs szavának az összetétele

BIT SZÁM	BIT LEÍRÁS
0	Azonosító bit 0
1	Azonosító bit 1
2	Azonosító bit 2
3	Azonosító bit 3
4	Azonosító bit 4
5	Azonosító bit 5
6	Azonosító bit 6
7	Azonosító bit 7
8	Azonosító bit 8
9	Azonosító bit 9
10	Azonosító bit 10
11	Azonosító bit 11
12	Alacsony feszültség szint választó
13	Baudrate választó bit 0 (BSL0)
14	Nincs használva
15	Nincs használva

Alacsony feszültségszint választó

Az alacsony feszültség szint választó bitet arra használjuk, hogy megmondjuk a HCS200-nak, hogy milyen V_{DD} szintet használjon. Ezt az információt az eszköz arra használja, hogy tudja, hogy mikor kell a vevőnek elküldeni a VLOW jelet. Ha ez a bit 1, akkor feltehetően 9V és 12V között van a V_{DD} szint. Ha ez a bit 0, akkor a V_{DD} szint 6.0V.

Baud rate választó bit

A BSL0 bit segítségével választhatjuk ki az átvitelnek a sebességét. A **9. táblázat** mutatja, hogy hogyan kell a bitet beállítani a különböző baudrate-ek beállításához.

9. Táblázat. Baud rate választás

BSL0	Basic Pulse Element	Kód szavak átvitele
0	400µs	Mind
1	200µs	2-ből 1

9.3.2 HCS300 és HCS301 kódoló

HCS300 és HCS301 memóriája 192 bites (12x16 bit (szó), **10. táblázat**). Ezt használjuk a már előzőekben tárgyalt adatok tárolására. A memória további leírását a következőkben tekinthetjük meg. Csak azoknak a táblázatokban található cellák értelmezésével foglalkozunk, amelyiket még nem ismerjük.

10. Táblázat. A HCS300/301 memória kiosztása

SZÓ CÍM	MNEMONIC	MEGNEVEZÉS
0	KEY_0	64 bites titkosító kulcs (word 0)
1	KEY_1	64 bites titkosító kulcs (word 1)
2	KEY_2	64 bites titkosító kulcs (word 2)
3	KEY_3	64 bites titkosító kulcs (word 3)
4	SYNC	16 bites szinkronozó érték
5	RESERVED	0000H címre beállítás
6	SER_0	az eszköz sorozat száma (word 0)
7	SER_1	az eszköz sorozat száma (word 1)
8	SEED_0	Seed érték (word 0)
9	SEED_1	Seed érték (word 1)
10	EN_KEY	16 bites ellenőrző kulcs
11	CONFIG	konfigurációs szó

Sorozat szám

A sorozat számról van már információkon, azonban egy dolgot meg kell jegyezzünk. Igaz, hogy a táblázat alapján 32 bitet tesz ki, de csak az alsó 28 bitet használjuk az átvitel során. A sorozat szám nem változik, mindig ugyanaz. A legfontosabb ennek a szónak a 31. bitje, mert ezzel állíthatjuk az automatikus lekapcsoló időzítőt.

Automatikus lekapcsoló időzítő választás (időtúllépés)

A sorozat szám 31. bitje felel ennek az időzítőnek a működtetéséről. Ez az időzítő gondoskodik arról, hogy le ne merüljön az akkumlátor, abban az esetben, ha az adónk a zsebünkben van, ráültünk és ezáltal folyamatosan működik. Ez az időtartam kb. 25 másodperc. Ha ez az idő letelt, akkor befejezi az adást a kódoló, és bár egy-két áramkör még aktív marad, de úgynevezett tartalék állapotba kerül. Amennyiben az említett bit egyes, akkor az időzítő működése engedélyezett, ellenkező esetben nem. Az időtartam hossza nem választható.

EN_KEY (ELLENŐRZŐ KULCS)

Ez egy választható opció, mely lehetővé teszi, hogy lekódoljuk az átvitelnek azt a részét, amely az adó sorozat számát tartalmazza. Beállítása a konfigurációs szó megfelelő bitjének állításával lehetséges (lásd **10. táblázat**). Normál esetben a sorozat szám kódolás nélkül kerül átvitelre, azonban a rendszer tervező lehetővé tette a titkosítását. Csak akkor történik meg a titkosítás, ha ez be van állítva. Ez a titkosító algoritmus különbözik a kulcsgeneráló algoritmustól. Az EN_KEY tipikusan egy véletlenszerűen generált szám, és egy rendszeren belül minden adónak ugyanaz.

Konfigurációs szó

A konfigurációs szó az egy 16 bites EEPROM-ban tárolt érték, melyet az eszköz használ, hogy a titkosító eljárás alatt információkat tároljon, valamint a beállítások állapotát mutatja. Az egyes bitek további értelmezését lássuk a **11. táblázatban**.

11. Táblázat. A HCS300/301 konfigurációs szavának az összetétele

BIT SZÁM	BIT LEÍRÁS
0	Azonosító bit 0
1	Azonosító bit 1
2	Azonosító bit 2
3	Azonosító bit 3
4	Azonosító bit 4
5	Azonosító bit 5
6	Azonosító bit 6
7	Azonosító bit 7
8	Azonosító bit 8
9	Azonosító bit 9
10	Túlcsordulás bit 0 (OVR0)
11	Túlcsordulás bit 1 (OVR1)
12	Alacsony feszültség szint választó
13	Baudrate választó bit 0 (BSL0)
14	Baudrate választó bit 1 (BSL1)
15	Envelope titkosítás választás (EENC)

Azonosító érték

Annyiban tér el a HCS200-as kódolótól, hogy itt 10 bites ez a szám.

Túlcsordulás bitek (OVR0 és OVR1)

Ezeket a biteket a szinkronizáló számláló felső határának a kiterjesztésére használja a berendezés. A legyártáskor az OVR0 és OVR1 biteket egybe írják. A kódoló automatikusan fogja törölni az OVR0 bitet a számláló első 0xFFFF-ből 0x0000-ba való átfordulásakor, és az OVR1-et

törli a második hasonló esemény hatására. Az egyszer törlődött OVR0 és OVR1 biteket már nem lehet egybe állítani. Így ha a dekódolót beprogramozzuk, hogy figyelje a túlcsordulást jelző biteket, akkor a szinkronizáló érték 196608-re növelhető.

ENVELOPE Titkosítás (EENC)

Ha ez a bit 1-be van állítva, akkor az átvitelnek a 32 bites állandó része is kódolva lesz. A 16 bites envelope kulcsot és az envelope algoritmust használja a titkosításhoz.

BAUDRATE választó bitek (BSL0, BSL1)

A BSL0 és a BSL1 bitek segítségével választhatjuk ki az átvitelnek a sebességét és a kód szó kivágást. A **12. táblázat** mutatja, hogy hogyan kell a biteket beállítani a különböző baudrate-ek beállításához.

12. Táblázat. Baud rate választás

BSL1	BSL0	Basic Pulse Element	Kód szavak átvitele
0	0	400µs	Mind
0	1	200µs	2-ből 1
1	0	100µs	2-ből 1
1	1	100µs	4-ből 1

Alacsony feszültség szint választó

Ugyanarra szolgál, mint a HCS200-nál sőt teljesen ugyanaz a HCS301-nek is, azonban a HCS300-nál a feszültség szintek miatt a következőképpen alakul. Ha ez a bit 1, akkor feltehetően 5V és 6V között van a V_{DD} szint. Ha ez a bit 0, akkor a V_{DD} szint 3.0V.

9.3.3 HCS360 kódoló

HCS360 memóriája 192 bites (12x16 bit (szó), **13. táblázat**). Ezt használjuk a már előzőekben tárgyalt adatok tárolására. A memória további leírását a következőkben tekinthetjük meg. Csak azoknak a táblázatokban található cellák értelmezésével foglalkozunk, amelyiket még nem ismerjük.

13. Táblázat. A HCS360/361 memória kiosztása

SZÓ CÍM	MNEMONIC	MEGNEVEZÉS
0	KEY_0	64 bites titkosító kulcs (word 0)
1	KEY_1	64 bites titkosító kulcs (word 1)
2	KEY_2	64 bites titkosító kulcs (word 2)
3	KEY_3	64 bites titkosító kulcs (word 3)
4	SYNC_A	16 bites szinkronozó érték
5	SYNC_B/SEED_2	16 bites szinkronozó érték vagy seed érték (word 2)
6	RESERVED	0000H címre beállítás
7	SEED_0	Seed érték (word 0)
8	SEED_1	Seed érték (word 1)
9	SER_0	Az eszköz sorozat száma (word 0)
10	SER_1	Az eszköz sorozat száma (word 1)
11	CONFIG	Konfigurációs szó

SYNC A, SYNC B (szinkronizáló számláló)

Ez két 16 bites szinkronizáló érték, amelyet arra használ, hogy elkészítse az átvitel ugró kódos részét. Minden átvitel során változik. A második szinkronizáló érték arra használható, hogy egy másik dekódolóval legyen szinkronban.

SEED 0, SEED 1 és SEED 2

Ez az érték már ismeretes a számunkra, hogy mi célt szolgál, azonban ennél a kódolónál ez 48 bit. A seed átvitel aktiválási módjától függően lehet azt kiválasztani, hogy 32 bites vagy 48 bites legyen a seed az átvitel során. Azonban ez a sorozat szám csökkenésének a rovására megy (csak a tanítás folyamán).

SER 0, SER 1 (kódoló sorozat száma)

A kettő együtt teszi ki a teljes sorozat számot. Azért 28 vagy 32 bites, mert programozó által választható, hogy használja-e a benne lévő konfigurációs biteket.

Konfigurációs szó

A konfigurációs szó az egy 16 bites EEPROM-ban tárolt érték, melyet az eszköz használ, hogy a titkosító eljárás alatt információkat tároljon, valamint a beállítások állapotát mutatja. Az egyes bitek további értelmezését lássuk a **15. táblázatban**.

LNGRD: hosszú guard idő

LNGRD=1 esetén meghosszabbítja a két kód szó közötti „őrző” időt. Ezt arra használjuk, hogy csökkentsük az átlagos teljesítményt a 100 ms-os intervallumon belül és ezáltal az átvitt csúcsteljesítmény magasabb lehet.

FAST 1, FAST 0 baud rate választó

14. Táblázat. Baud rate választás

Rate (bps)	FAST 1	FAST 0
625	0	0
1250	0	1
1250	1	0
2500	1	1

15. Táblázat. A HCS360 konfigurációs szavának az összetétele

BIT SZÁM	Szimbólum	Bit leírás
0	LNGRD	Hosszú guard idő
1	FAST 0	Baud rate választó
2	FAST 1	Baud rate választó
3	NU	Nem használt
4	SEED	Seed átvitel engedélyezése
5	DELM	Késleltetés engedélyezése
6	TIMO	Idő túlfutás engedélyezése
7	IND	Egyedi mód engedélyezése
8	USRA0	User bit
9	USRA1	User bit
10	USRB0	User bit
11	USRB1	User bit
12	XSER	Kibővített sorozat szám engedélyezés
13	TMPSD	Átmeneti seed átvitel engedélyezés
14	MANCH	Manchester /PWM mód választó
15	OVR	Túlcsordulás bit

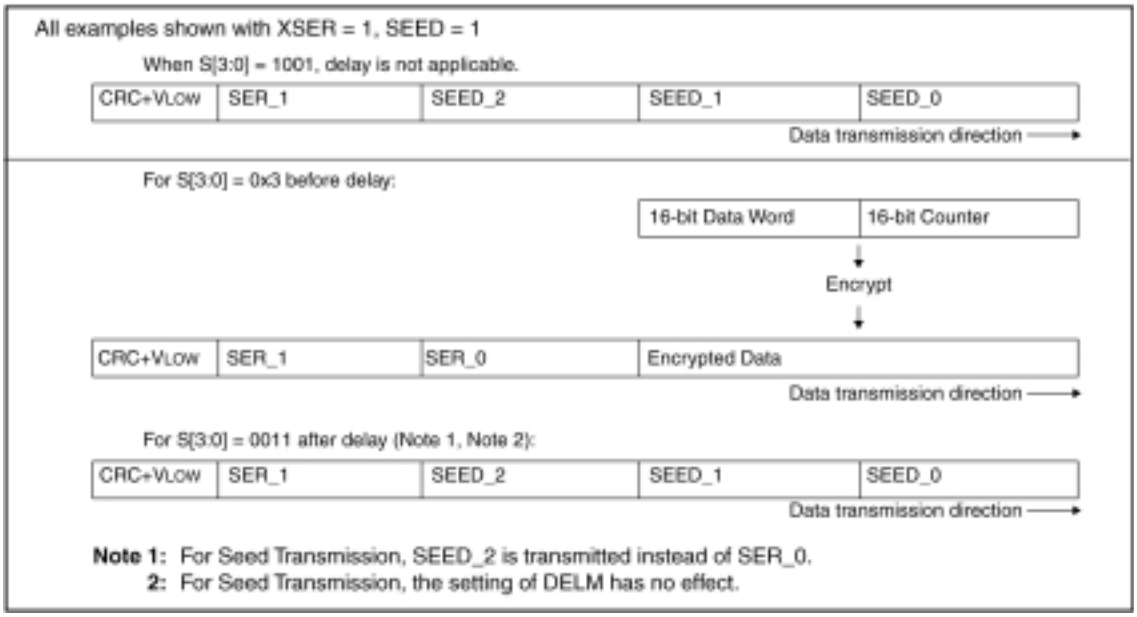
SEED: seed átvitel engedélyezése

Ha a SEED=0, akkor ez az átviteli lehetőség tiltott. Az egyedi számláló üzemmódot csak ilyen tiltott állapot mellett lehet használni, mivel a seed átvitelnél a SEED_2 van a második számláló helyén.

Ha a SEED=1, akkor ez az átviteli mód engedélyezett. A megfelelő nyomógomb kombinációval aktivizálható a seed elküldése. Ekkor a seed érték (SEED_0, SEED_1, SEED_2) és a sorozat szám (SER_1) felső, 12 vagy 16 bitjei lesznek az üzenet ugró kódos része helyén. A seed átvitel az S[3:0] = 1001 és az S[3:0] = 0011 (késleltetett) nyomógomb kombinációk hatására aktiválható. Ez nem veszi figyelembe az IND bit állítását. (**16. táblázat, 13. ábra**)

16. Táblázat. A funkció kódok szerepe a HCS360/361 kódolóknál

	S3	**S2**	**S1**	**S0**	**IND=0**	**IND=1**	**Megjegyzés**
					Számláló		
1	0	0	0	1	A	A	
2	0	0	1	0	A	A	
3	0	0	1	1	A	A	Ha SEED=1, akkor seed átvitel a késleltetés után
4	0	1	0	0	A	A	
5	0	1	0	1	A	A	
6	0	1	1	0	A	A	
7	0	1	1	1	A	A	
8	1	0	0	0	A	B	Ha SEED=1, seed átvitel azonnal
9	1	0	0	1	A	B	
10	1	0	1	0	A	B	
11	1	0	1	1	A	B	
12	1	1	0	0	A	B IR mód	
13	1	1	0	1	A	B IR mód	
14	1	1	1	0	A	B IR mód	
15	1	1	1	1	A	B IR mód	



13. Ábra. A seed átvitelek átviteli formátuma és aktivizálási lehetősége

DELM: késleltetett üzemmód

Ha a DELM=1, akkor ez az üzemmód engedélyezett. A késleltetett üzemmód elsősorban azért van, hogy az előző KEELOQ egységekkel kompatíbilis legyen. Ha DELM=0, akkor tiltott ez az üzemmód (**17. táblázat**).

17. Táblázat. Jellemző késleltetési időtartamok

FAST 1	FAST2	A kód szavak hossza a késleltetett üzemmód előtt	A késleltetett mód előtti idők (MANCH=0)	A késleltetett mód „ref” idői (MANCH=1)
0	0	28	≈2.9s	≈5.1s
0	1	56	≈3.1s	≈6.4s
1	0	28	≈1.5s	≈3.2s
1	1	56	≈1.7s	≈4.5s

TIMO: Idő túllépés

Ez tulajdonképpen ugyanazt a célt szolgálja, mint a HCS300-nál az „*Automatikus lekapcsoló időzítő választó*”. Ha TIMO=1, akkor engedélyezett ez a funkció. Arra lehet használni, hogy a kódoló félbeszakítsa a nem szándékos, folyamatos adást. Ebben az esetben a PWM kimenet alacsonyba áll, a LED kimenet pedig kikapcsolódik. Az áramfogyasztás magasabb lesz, mint stand by módban, azért mert áram fog folyni az aktív bemeneti ellenállásokon. Ezt az állapotot csak úgy lehet kikapcsolni, ha minden bemenet alacsony állapotba kerül (**18. táblázat**).

18. Táblázat. Jellemző időtúllépési időtartamok

FAST 1	FAST2	Maximálisan átvitt kódszavak száma	Időtúllépés ideje (MANCH=0)	Időtúllépés ideje (MANCH=1)
0	0	256	≈26.5 s	≈46.9 s
0	1	512	≈28.2 s	≈58.4 s
1	0	256	≈14.1 s	≈29.2 s
1	1	512	≈15.7 s	≈40.7 s

IND: Egyedi üzemmód

Ezt az üzemmódot akkor használjuk, ha egy kódolót használunk két vevő vezérléséhez. Ehhez két számlálót használ az adó (SYNC_A és SYNC_B). Mint azt a **16. táblázat** is mutatja, hogy 1-7-ig a funkciókódok (nyomógomb kombinációnak megfelelően) a SYNC_A-t használják és 8-15-ig, pedig a SYNC_B-t. Ennek az üzemmódnak az aktiválásával az IR mód is aktiválódik. IR üzemmódban a funkció kódok 12-15-ig a SYNC_B-t használják. Ha IND=0, akkor minden funkciókód hatására csak a SYNC_A-t használja a kódoló. Ha IND=1, akkor pedig a táblázatnak megfelelően használja a számlálókat.

USRA, B: User bitek

Ezek a bitek az azonosító érték részei. A user biteket az IND bittel együtt arra használja, hogy azonosítsa a számlálókat, amikor az egyedi módot alkalmazza.

XSER: A sorozat szám kibővítése

Ha XSER=1, akkor a teljes 32 bites sorozat szám (SER_1 és SER_0) átvitelre kerül. Ha XSER=0, akkor ugyanaz jellemző rá, mint a HCS200/300/301-es kódolókra.

TMPSD: Átmeneti seed átvitel

Ezt az átviteli módot arra használjuk, hogy letiltsuk a tanulást, azután hogy az adót már egy meghatározott alkalommal használtuk. Ez jelentősen megnöveli a rendszer biztonságát. Ha TMPSD=1, akkor a seed átvitel letiltódik egy bizonyos számú ugró kódos átvitel után. Az átvitelek száma a seed átvitel letiltása előtt programozható a szinkronizáló számláló állításával. Lásd a **19. táblázatot**.

19. Táblázat. A szinkronizáló számláló inicializáló értékei

Szinkronizáló számláló értékei	Átvitelek száma
0000H	128
0060H	64
0050H	32
0048H	16

MANCH: Manchester moduláció

Ezzel a bittel lehet kiválasztani, hogy Manchester moduláció legyen vagy PWM. Ha MANCH=0, a PWM módot választottuk ki.

OVR: túlcsordulás bit

Ugyanaz, mint a HCS300-nál, csak itt egy bit van, ami azt jelenti, hogy a számlálót csak 128K-ra bővíti ki

9.3.4 HCS361 kódoló

A memória felépítése ugyanaz, mint a HCS360 kódolóé, viszont a konfigurációs szavának az alkotó részei már mások (**20. táblázat**).

Konfigurációs szó

20. Táblázat. A HCS361 konfigurációs szavának az összetétele

BIT SZÁM	Szimbólum	Bit leírás
0	BACW	Blank Alternate kód szó
1	FAST	Baud rate választó
2	TXWAK	PWM mód: 1/6,2/6 vagy 1/3,2/3 választható VPWM mód: Wakeup engedélyezés
3	SPM	Szinkron pulzus moduláció
4	SEED	Seed átvitel engedélyezése
5	DELM	Késleltetés engedélyezése
6	TIMO	Idő túlfutás engedélyezése
7	IND	Egyedi mód engedélyezése
8	USRA0	User bit
9	USRA1	User bit
10	USRB0	User bit
11	USRB1	User bit
12	XSER	Kibővített sorozat szám engedélyezés
13	TMPSD	Átmeneti seed átvitel engedélyezés
14	VPWM	VPWM mód választó
15	OVR	Túlcsordulás bit

Láthatjuk sok olyan konfigurációs bit szerepel, amely a HCS360-nál is megtalálható. Ezért ezeknek az értelmezésére külön nem térek ki. Csak azokat vizsgálom, amelyek csak a HCS361-re jellemző.

BACW: Blank Alternate kód szó

Ha BACW=1, akkor azt állítottuk be, hogy a kódoló minden második kód szót küldje el. Ezt arra használjuk, hogy csökkentsük az átlagos teljesítményt a 100 ms-os intervallumon belül és ez által az átvihető csúcsteljesítmény magasabb.

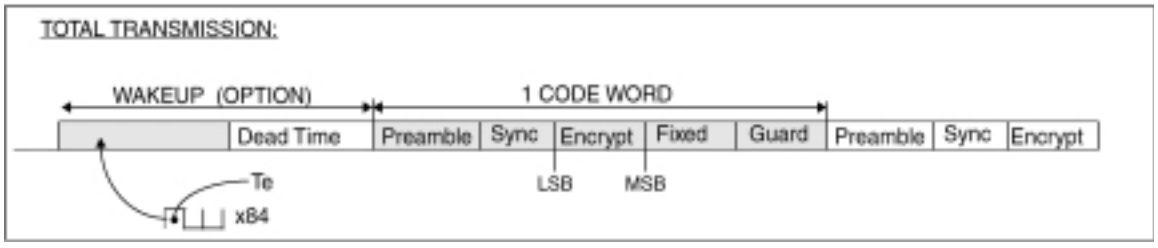
FAST: Gyors átvitel választó

Ezzel lehet kiválasztani a baud rate-tet. Ha a FAST=1, akkor normálisan 1667 bps a baud rate, ellenkező esetben 833 bps.

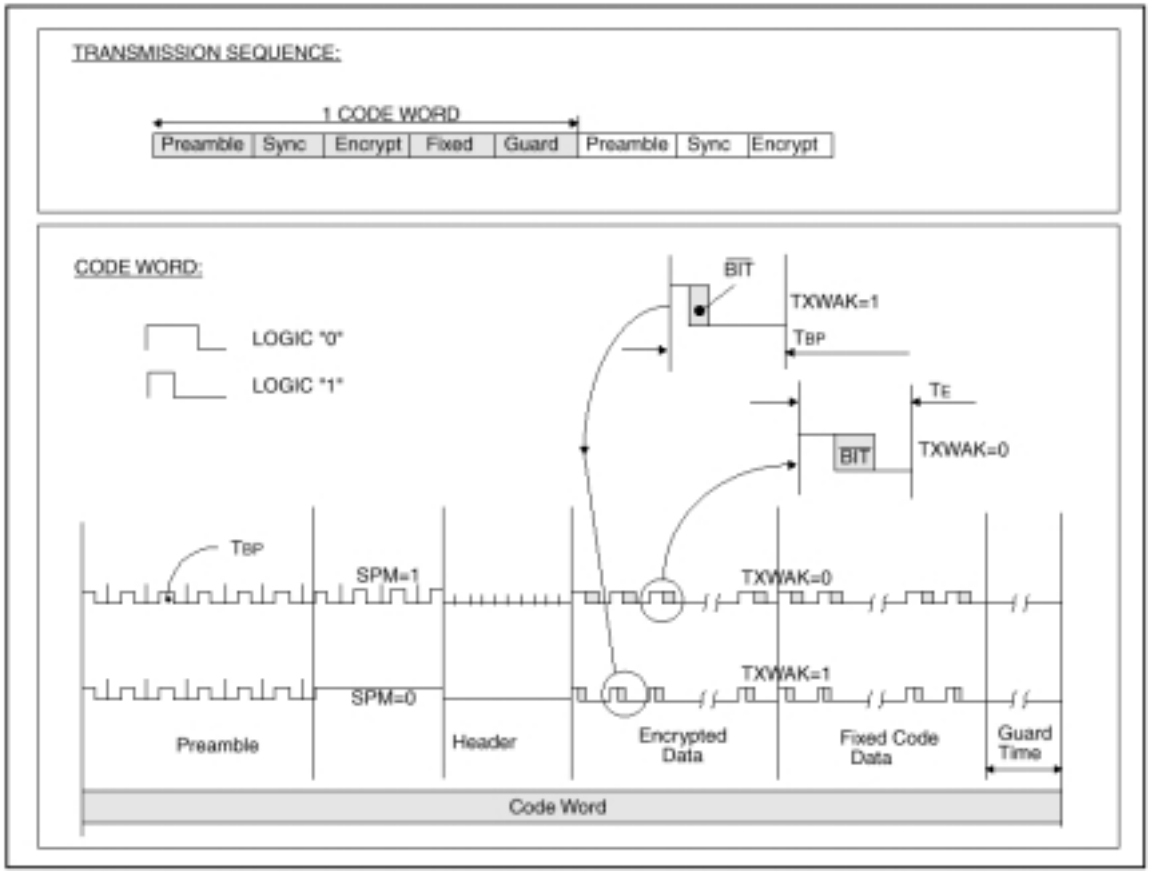
TXWAK: bit formátum választó vagy Wakeup

PWM módban ezzel választhatjuk ki a bit formátumot. Ha TXWAK=1, akkor a PWM impulzus 1/6,2/6 és ha TXWAK=0, akkor 1/3,2/3 (**15. ábra**).

VPWM módban ezzel a bittel lehet engedélyezni a wake-up jelet. Ha TXWAK=1, akkor az átvitelben a wake-up és a „holt idő” szerepelni fog (**14. ábra**). Minden átvitelnek csak az első kód szójában fog szerepelni a wake-up jel.



14. Ábra. Az átviteli szóban a „holt idő” megjelenése



15. Ábra. Az 1/6-2/6 és 1/3-2/3-os PWM formátum választás

DELM: késleltetett üzemmód

Ha a DELM=1, akkor ez az üzemmód engedélyezett. A késleltetett üzemmód elsősorban azért van, hogy az előző KEELOQ egységekkel kompatíbilis legyen. Ha DELM=0, akkor tiltott ez az üzemmód (**21. táblázat**).

21. Táblázat. Jellemző késleltetési időtartamok

TXWAK	FAST2	A kód szavak hossza a késleltetett üzemmód előtt	A késleltetett mód előtti idők (VPWM=0)
0	0	28	≈2.8 s
0	1	56	≈2.9 s
1	0	28	≈2.6 s
1	1	56	≈2.8 s

TIMO: Idő túllépés

Azért emelem ki, mert más bitek határozzák meg az időtartamokat a HCS360-hoz képest (**22. táblázat**).

22. Táblázat. Jellemző időtúllépési időtartamok

TXWAK	FAST2	Maximálisan átvitt kódszavak száma	Időtúllépés ideje (VPWM=0)
0	0	256	≈25.6 s
0	1	512	≈27.2 s
1	0	256	≈23.8 s
1	1	512	≈25.4 s

VPWM: Változtatható impulzus szélesség moduláció

Ezzel lehet kiválasztani, hogy PWM vagy VPWM modulációt használjon a kódoló. Ha a VPWM=1, akkor a VPWM modulációt alkalmazza és ezzel egy időben a következők fognak történni:

1. A TXWAK bit engedélyezésével a WAKEUP átvitel lesz kiválasztva.
2. Kibővül az „őrző” idő.

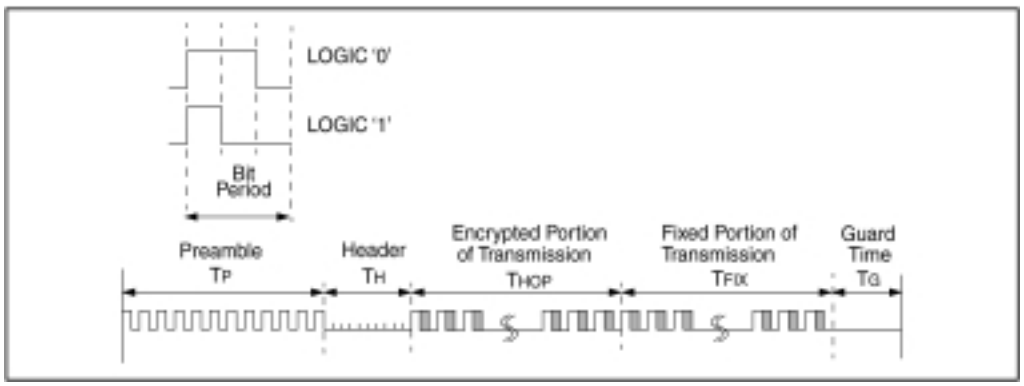
Ha a VPWM=0, akkor a PWM modulációt használja a kódoló.

9.4 A KÓDOLÓK ÁLTAL ALKALMAZOTT MODULÁCIÓS FORMÁK

A kódolók által alkalmazott különböző modulációs formátumokat ismerhetjük meg ebben a részben. Ezek a PWM, VPWM és Manchester moduláció.

9.4.1 HCS200, HCS300/301 kódoló

Ezek a típusú kódolók csak a PWM modulálást ismerik. Az átvitel számos részből épül fel (**16. ábra**). Mindegyik átvitel egy bekezdő jelszakasszal és egy fejléccel kezdődik, aztán következik, a titkosított és a fix adat. Minden átvitelt követ egy biztonsági szakasz mielőtt a következő átvitel elkezdődne.

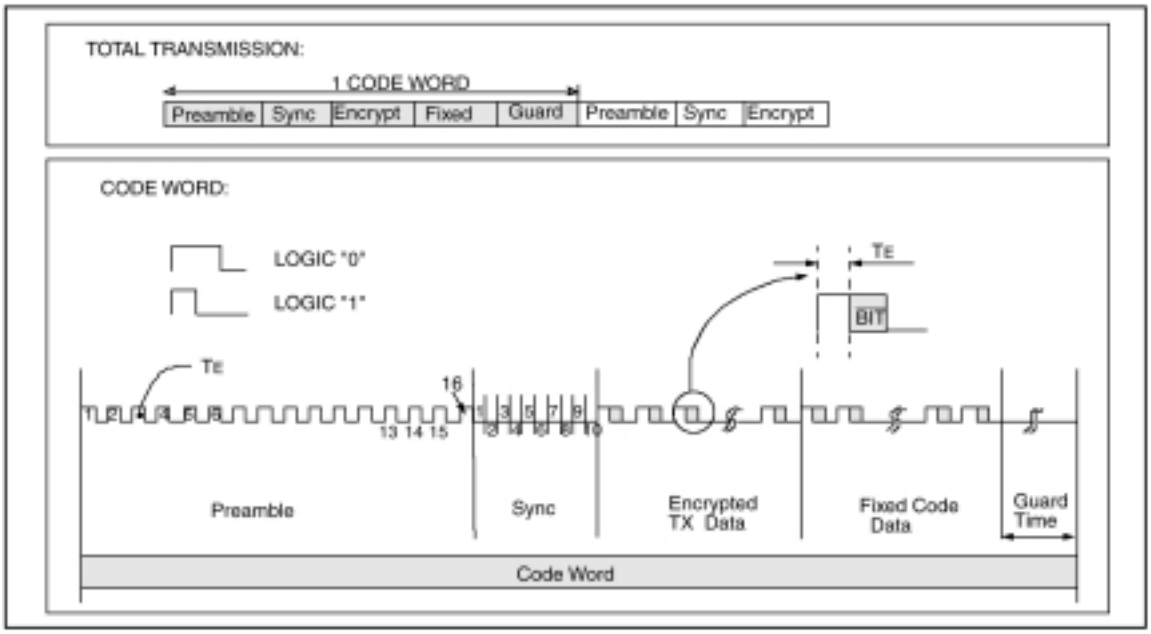


16. Ábra. Az átvitelek összetétele

9.4.2 HCS360 kódoló

Az átvitel számos részből épül fel. Tulajdonképpen, mint az előző kódolóknál itt is mindegyik átvitel egy bekezdő jelszakasszal és egy fejléccel kezdődik, aztán következik, a titkosított és a fix adat. Minden átvitelt követ egy biztonsági szakasz mielőtt a következő átvitel elkezdődne. Azonban itt megkülönböztetünk Manchester és PWM modulálást.

PWM modulálás

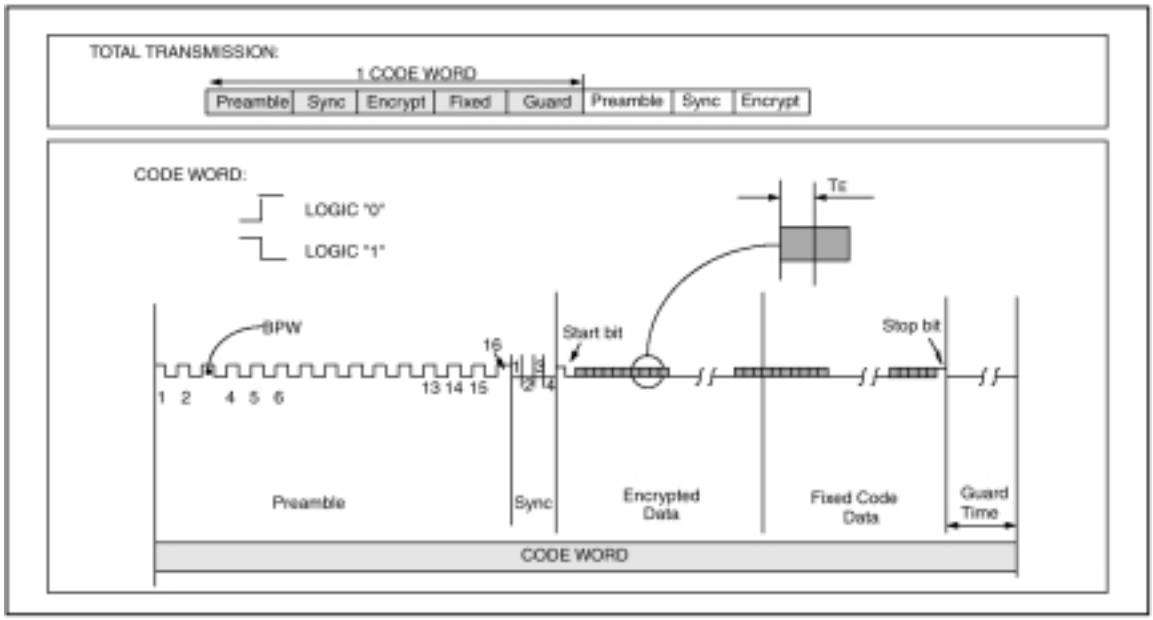


17. Ábra. A PWM modulálás

Az **17. ábrán** is láthatjuk, hogy ez tulajdonképpen az impulzusok szélességének a modulálásával hordozza az információt. A logikai 1 állapotát az jelzi, ha a magas állapot az impulzus 2/3-át teszi ki és az alacsony az 1/3-át és a logikai 0-át pedig az ellenkezője.

Manchester modulálás

Az **18. ábrán** láthatjuk, hogy ennél a modulálási fajtánál az impulzuson belüli él váltás hordozza az információt. Az impulzus kitöltési tényezője 50%-os és pont az impulzus felénél történik meg az él váltás. A felfutó él logikai 0 állapotot jelent, míg a lefutó él logikai 1-et.



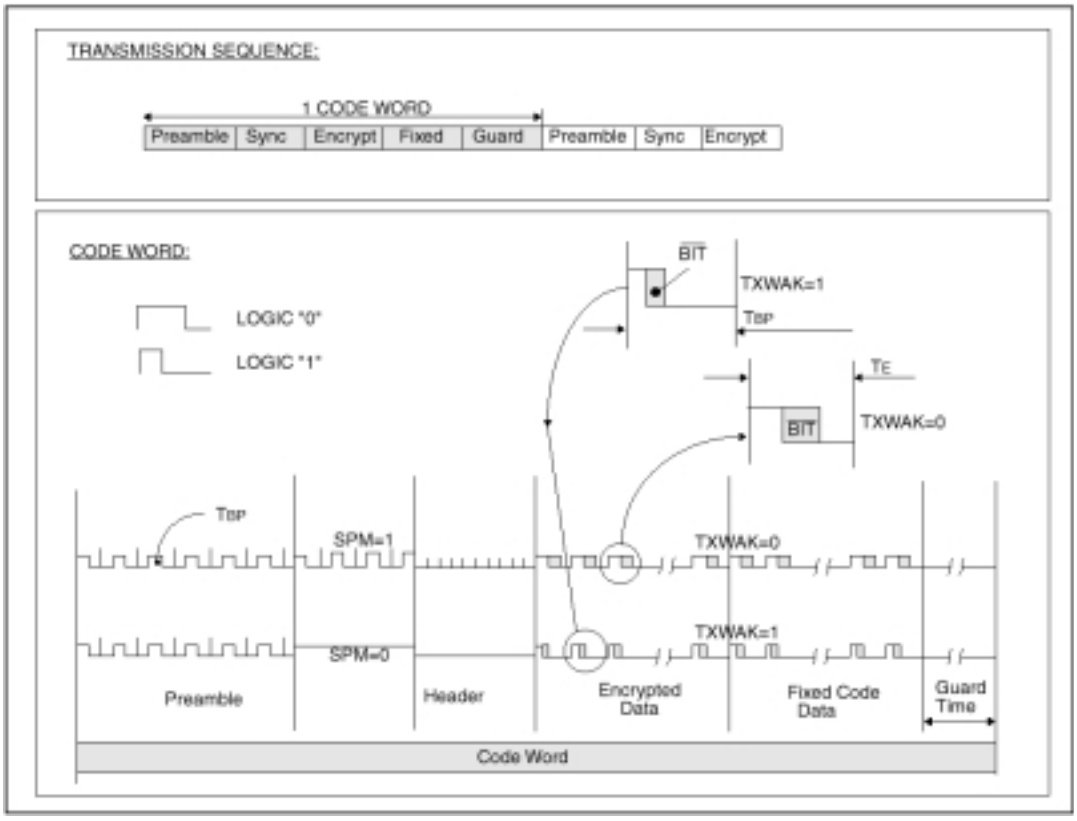
18. Ábra. Manchester modulálás

9.4.3 HCS361 kódoló

Az átvitel itt is számos részből épül fel. Tulajdonképpen ugyanaz, mint a HCS360-as kódoló, azonban itt a bekezdő jelszakasz előtt VPWM modulálás esetében kiválaszthatjuk, hogy legyen-e egy ún. Wakeup szakasz vagy sem. Megkülönböztetünk VPWM és PWM modulálást.

PWM modulálás

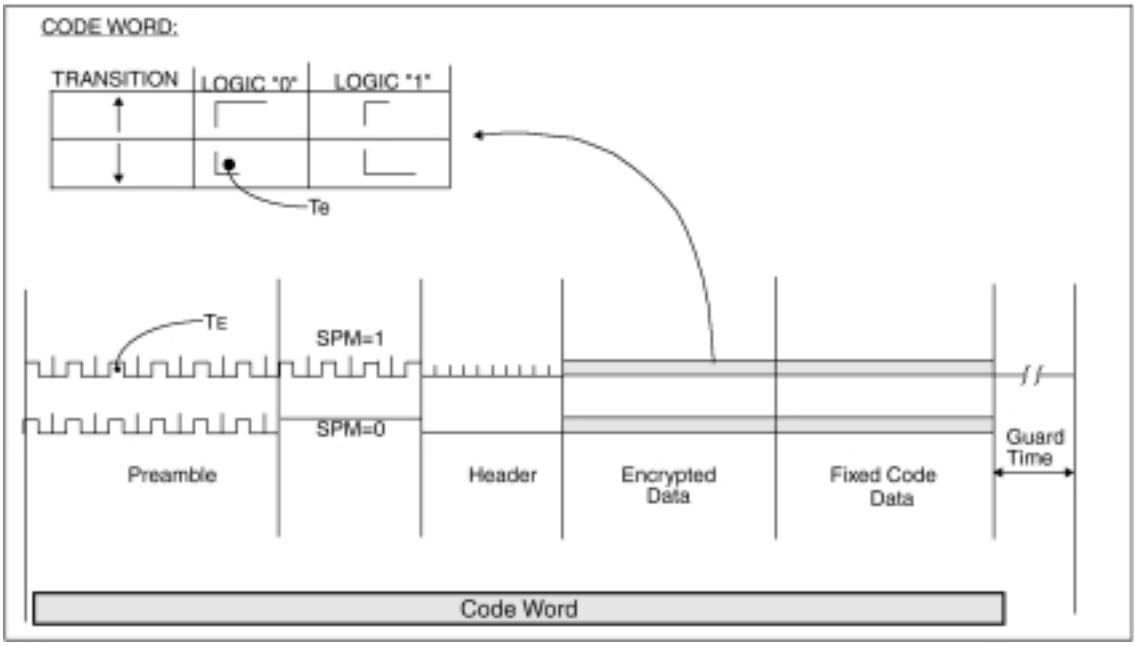
Tulajdonképpen ugyanaz, mint a többi kódolónál azonban azért kell kiemelni, mert a HCS361-nek van egy olyan konfigurációs bitje, amelyikkel a PWM kitöltési tényezőjét változtathatjuk.



19. Ábra. PWM modulálás a kitöltési tényező változtatása

Mint a **19 ábrán** is láthatjuk, hogy a PWM jel kitöltési tényezője egyrészt 2/3-1/3-os lehet, mint az eddigi kódolóknál, vagy itt be lehet állítani, hogy 2/6-1/6-os kitöltési tényezőjű impulzus szélesség modulált jel hordozza az információt.

VPWM modulálás



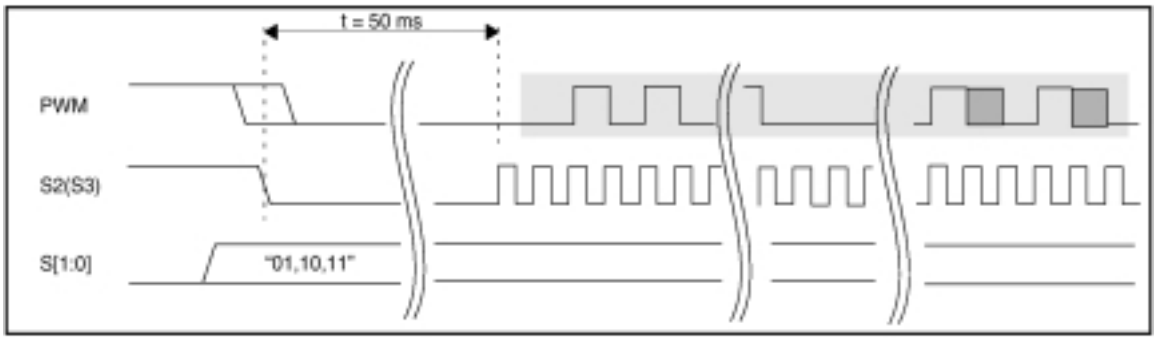
20. Ábra. A VPWM modulálás

Láthatjuk a **20. ábrán**, hogy hogyan történik az információk kódolása a VPWM modulációval. A VPWM tulajdonképpen változtatható szélességű impulzus szélesség modulációt jelent. Két részre bonthatjuk a beérkező információhordozó jeleket. Vagy felfutó, vagy lefutó él „vezéreltek". Abban az esetben, ha egy felfutó él kezdődik és 2 T_E hosszúságú magas állapotú szakasz követ1i addig, míg nem történik él váltás, akkor logikai 0-át jelent, ha csak T_E , akkor logikai 1-et. Ha viszont lefutó élű lesz a következő él váltás és azt T_E hosszúságú alacsony állapotú szakasz követi addig, míg újabb él váltás nem történik, akkor ez jelenti a logikai 0-át, és ha 2 T_E, akkor az, logikai 1-et.

9.5 SZINKRONIZÁLT ADATÁTVITELI MÓD (VEZETÉKES)

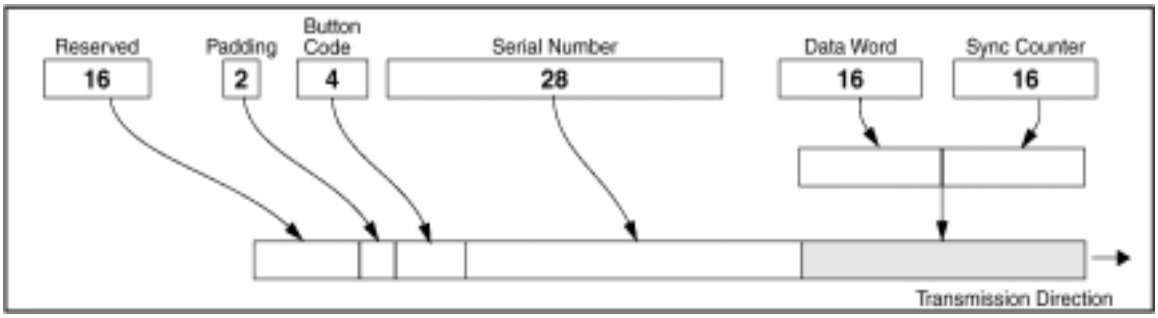
A szinkron átviteli mód egy külső órajel használatával történik. Mindegyik kódoló esetében tekinthetjük egyszerre, mert tulajdonképpen ugyanúgy zajlik le a folyamat. Van egy kis különbség köztük, de erre az áttekintés közben fel fogom hívni a figyelmet.

Azt, hogy hogyan lehet felélesztени ezt a módszert, azt a **21. ábra** mutatja. Ha S1 és S2 közül valamelyik állapota változik az S2 (vagy S3, ami a HCS200-nak nincsen) lefutó elé alatt, akkor az egység belép a szinkron átviteli üzemmódba. Ebben az üzemmódban úgy működik, mint egy normál adó, azzal a kivétellel, hogy a PWM adat string időzítése kívülről vezérelt és a kód szó végén 16 extra bit átvitelére kerül sor. A funkció kód az S0 és S1 aktuális értéke lesz az S2 vagy S3 lefutó élénél. Az előbb említett órajelnek nem szabad 20 KHz-nél nagyobbnak lennie. A kód szó ugyanaz, mint a PWM módban a 16 pótbittel a szó végén, mely pótbitek elhagyhatóak. Szinkron átviteli módban az S2-öt vagy S3-at nem szabad aktiválni nyomógomb által, amíg minden belső feldolgozás be nem fejeződik.



21. Ábra. A szinkronizált átviteli mód

Az átviteli szó formátuma:



22. Ábra. A szinkronizált átviteli mód alatt az átvitel formátuma

9.6 SPECIÁLIS JELLEMZŐK

9.6.1 Auto-shutoff

Ez a funkció a HCS200-at kivéve mindegyik kódolónál megtalálható. Az Auto-shutoff funkció automatikusan lekapcsolja az egységet az átvitelkor, ha egy nyomógomb hosszú ideig nyomva marad. Ez, pl. olyankor történhet meg, amikor a távirányító a zsebben van és egy gomb benyomódik és sok ideig úgy marad. Ez a funkció engedélyezhető és tiltható az Auto-shutoff bit törlésével és írásával. Ha a bit egyben van, akkor a funkció engedélyezett, ellenkező esetben tiltott. Időlefutási periódus kb. 25 másodpercenként ismétlődik.

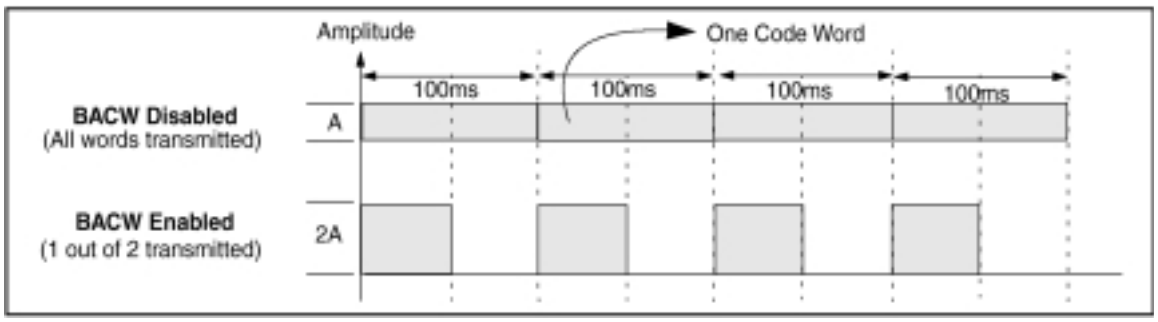
9.6.2 Blank Alternate kód szó (BACW)

A Szövetségi Hírközlési Tanács (FCC) 15. szabálya meghatározza, hogy egy eszköz mekkora teljesítménnyel sugározhat. A teljesítményt a legrosszabb esetben vett közepesen átvitt teljesítményből számolja a 100 ms ablakra nézve. Ezért hasznos minimalizálni az átvitt szó teljesítmény ciklusát. Ez az egyes bitek teljesítmény ciklus alatti minimalizálásával és az egymást követő szavak kivágásával érhető el. A BACW-t arra használjuk, hogy lecsökkentsük az átvitel

átlagos teljesítményét (**23. és 24. ábra**). Ez egy választható tulajdonság. A BACW használata lehetővé teszi a felhasználó számára, hogy nagyobb amplitúdója legyen az átvitelnek amellett, hogy az átvitel hossza rövidebb. Az FCC korlátozásokat tesz az átlagos teljesítményre, amellyel az egység továbbíthatja az üzenetet, és a BACW hatékonyan meggátolja a folyamatos adást azáltal, hogy csak minden második kód szó átvitelét engedélyezi. Ez lecsökkenti az átlagos teljesítményt, és ezzel megfelel az FCC előírásainak.

9.6.2.1 HCS200, HCS360/361 kódoló

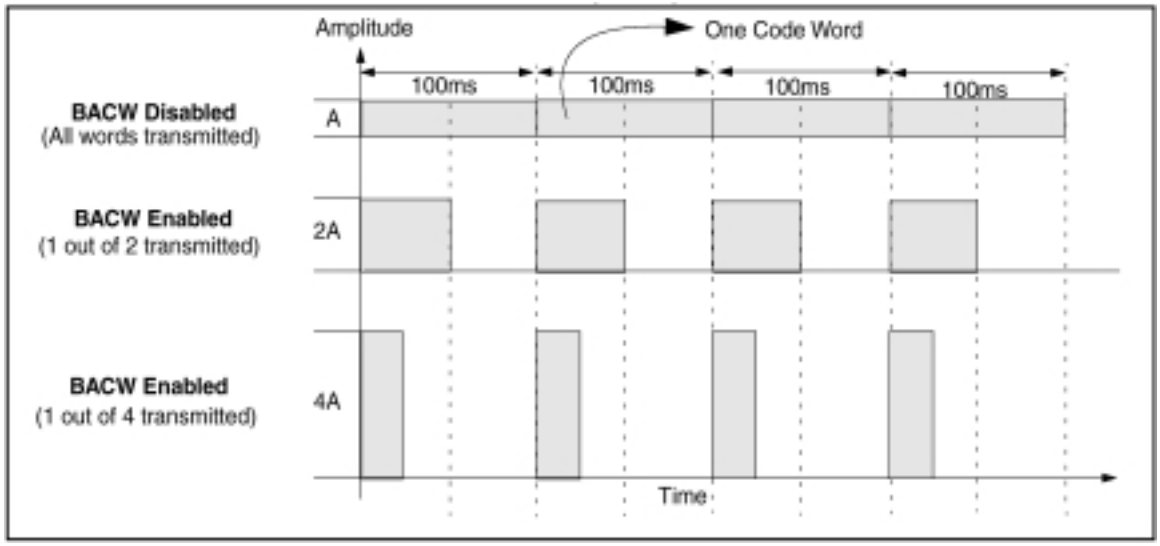
Ezeknél a kódoló típusoknál megegyezik a BACW választási lehetőség (**23. ábra)**.



23. Ábra. BACW választási lehetőségei

9.6.2.2 HCS300/301 kódoló

Ezeknél a kódoló típusoknál megegyezik a BACW választási lehetőség (**24. ábra)**.

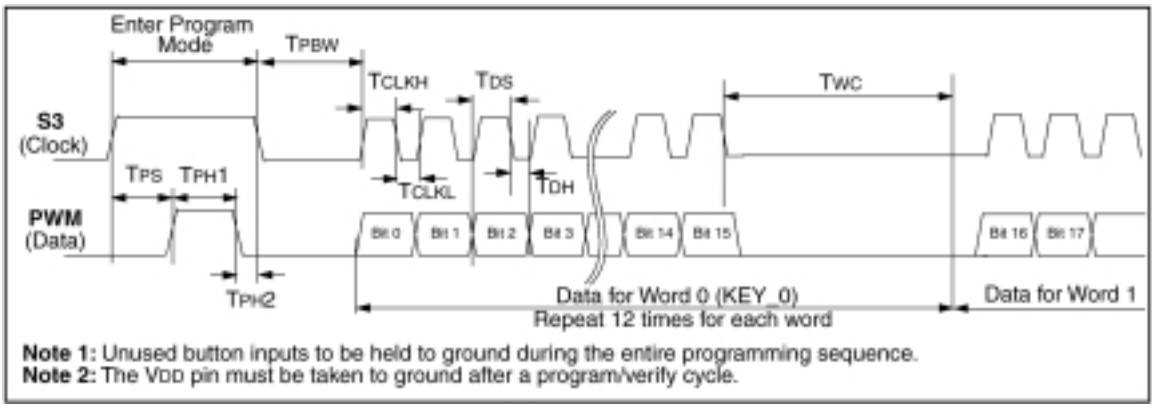


24. Ábra. BACW választási lehetőségei

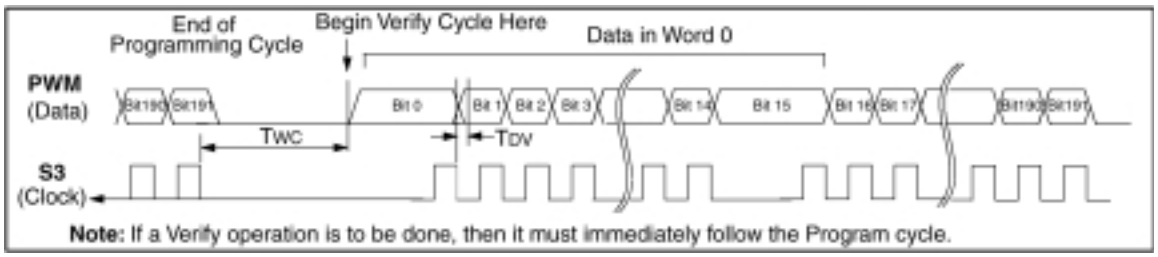
9.7 PROGRAMOZÁS

A HCSXXX kódolók programozásuk során egy 192 bites csomagot kell sorosan beprogramozni a belső EEPROM-ba. A programozást úgy kell kezdeni, hogy az S3 (HCS200-nál S2, HCS300/301-nál S3, HCS360/361-nél S2 vagy S3) magasba emelését követően megadott idő múlva (**23., 24. és 25. táblázat**) a PWM lábat magasba kell emelni a **25. ábrán** is látható módon. Ezután a PWM lábat a meghatározott idő múlva alacsonyba kell állítani és ezt követően az S3-at TPH2 idő elteltével szintén alacsony szintbe kell rakni. Ha a programozási módból visszatért, egy bizonyos késleltetést kell biztosítani a készüléknek, hogy befejezze a belső írási ciklust. Erre azért van szükség, mert ilyenkor az EEPROM-ban mindent a megfelelő helyre tárol el. Az egységet úgy lehet órajel vezérelten 16 bitesével programozni (természetesen sorosan), hogy az S3-at használjuk órajelnek és a PWM lábat pedig az adatoknak. Minden 16 bites szó betöltése után szükség van egy késleltetésre, hogy a belső programciklus befejeződjön. Ez az érték is meg van határozva. A programozás befejeztével ellenőrzésre van szükség (**26. ábra**), hogy

visszaellenőrizzük az EEPROM tartalmát. Az olvasás úgy történik, hogy az S3-ra adjuk az órajelet és a PWM lábon pedig olvashatjuk az adat biteket. A biztonság miatt, az nem lehetséges, hogy ellenőrzést programozás nélkül hajtsunk végre. Az ellenőrzés csak egy programozási ciklust követően egyszer hajtható végre.



25. Ábra. A HCS200 és HCS300/301 programozásának módja



26. Ábra. A HCS200 és HCS300/301 visszaellenőrzésének módja

HCS200

23. Táblázat. Időtartamok a programozáshoz és az ellenőrzéshez a HCS200-nál

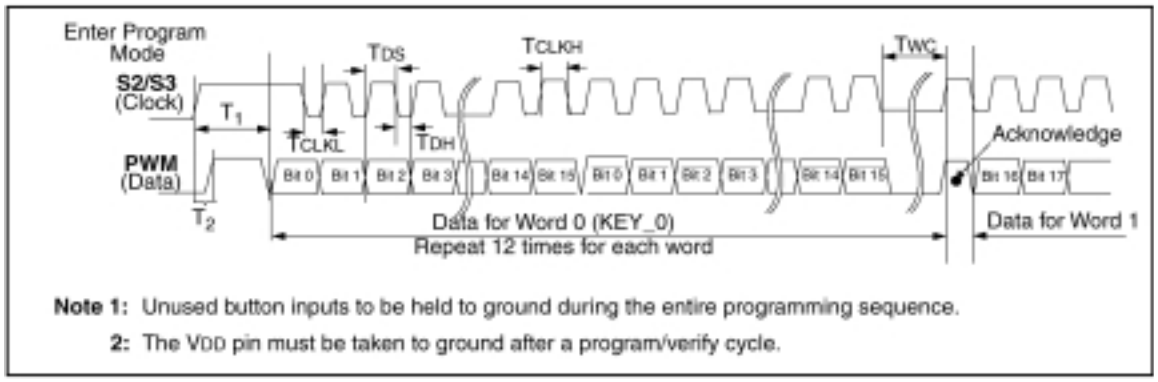
Paraméterek	Jelölések	Min.	Max.	Egység
Program mode setup time	T_{PS}	3.5	4.5	ms
Hold time 1	T_{PH1}	3.5	-	ms
Hold time 2	T_{PH2}	50	-	µs
Bulk Write time	T_{PBW}	-	3.5	ms
Program delay time	T_{PROG}	-	3.5	ms
Program cycle time	T_{WC}	-	36	ms
Clock low time	T_{CLKL}	25	-	µs
Clock high time	T_{CLKH}	25	-	µs
Data setup time	T_{DS}	0	-	µs
Data hold time	T_{DH}	18	-	µs
Data out valid time	T_{DV}	10	24	µs

HCS300/301

24. Táblázat. Időtartamok a programozáshoz és az ellenőrzéshez a HCS300/301-nél

Paraméterek	Jelölések	Min.	Max.	Egység
Program mode setup time	T_{PS}	3.5	4.5	ms
Hold time 1	T_{PH1}	3.5	-	ms
Hold time 2	T_{PH2}	50	-	µs
Bulk Write time	T_{PBW}	-	2.2	ms
Program delay time	T_{PROG}	-	2.2	ms
Program cycle time	T_{WC}	-	36	ms
Clock low time	T_{CLKL}	25	-	µs
Clock high time	T_{CLKH}	25	-	µs
Data setup time	T_{DS}	0	-	µs
Data hold time	T_{DH}	18	-	µs
Data out valid time	T_{DV}	10	24	µs

HCS360/361



27. Ábra. A HCS360/361 programozásának módja

25. Táblázat. Időtartamok a programozáshoz és az ellenőrzéshez a HCS360/361-nél

Paraméterek	Jelölések	Min.	Max.	Egység
Program mode setup time	T_2	0	4.0	ms
Hold time 1	T_1	9.0	-	ms
Program cycle time	T_{WC}	-	30	ms
Clock low time	T_{CLKL}	25	-	µs
Clock high time	T_{CLKH}	25	-	µs
Data setup time	T_{DS}	0	-	µs
Data hold time	T_{DH}	18	-	µs
Data out valid time	T_{DV}	-	24	µs

9.8 HCS410

Ez a kódoló típus a szakdolgozatom írása során még nem volt forgalomban, ezért nem volt lehetőségem a tesztelésre. Ebből kifolyólag csak általánossággal tudok róla írni.

Tulajdonképpen a HCS360 továbbfejlesztett változata, mely kódoló és egyben válaszjeladó (transponder). A következőkben vázlatosan lássuk a jellemzőit.

Biztonság

- Két programozható 64 bites titkosító kulcs
- 16/32 bites kétirányú lekérdezés és válasz az egyik kulcs használatával
- 69 bites átviteli hossz
- 32 bit egyirányú ugró kód, 37 bit titkosítatlan rész
- a titkosító kulcsok olvasás védettek
- Programozható 28/32 sorozat szám
- 60 bites olvasás védett seed a titkos tanuláshoz
- 3 IFF kódoló algoritmus
- Késleltetett növekményes mechanizmus
- Aszinkron válaszadó kommunikáció
- Soros információ átvitel

Működés

- Ütközés védettség több válaszjeladó esetén
- Passzív közelségi aktiválás
- Ellentétes polaritás elleni védettség

Kulcs definíciók

- CH mód: ugró kódos üzemmód. A HCS410 minden egyes aktiválás során egy 69 bites üzenetet visz át, ahol legalább 32 bit változik.
- Kódoló kulcs: Ugyanaz, mint a többi kódolónál
- IFF: Barát vagy ellenség felismerés, mely egy jelzés érvényesítést jelent. A dekóder egy lekérdezést generál és ellenőrzi, hogy a kódoló által kiadott válaszjel érvényes-e.
- KEELOQ titkosító algoritmus

9.8.1 KEELOQ IFF

A HCS410-et IFF válaszadóként „automatikus” azonosításra lehet használni. IFF módban egy megfelelő kulcshitelesítési képességgel rendelkezik, 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 válaszát 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 algoritmusa 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 ismertetettem 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

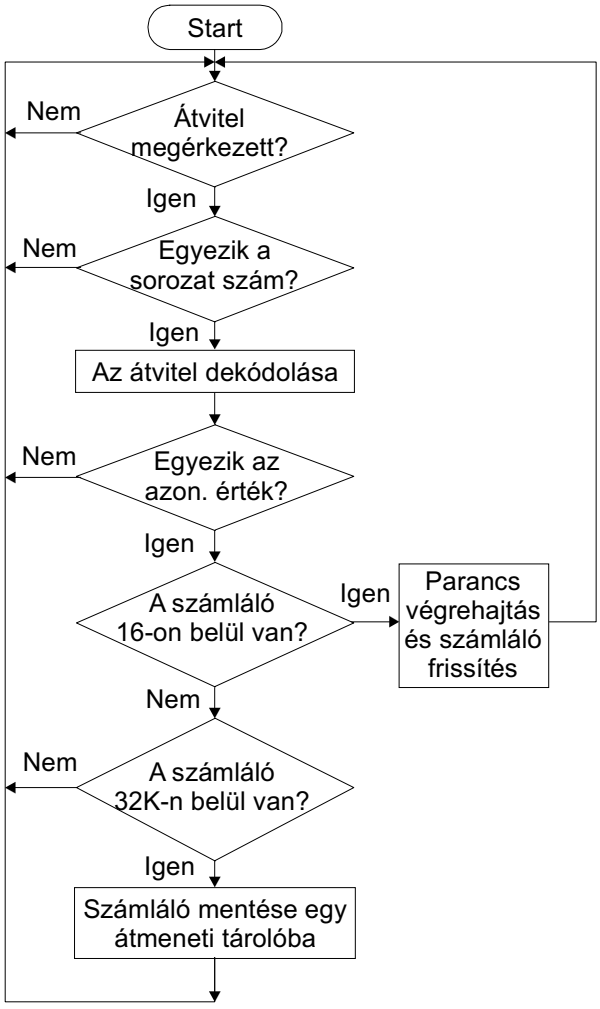
Már ismeretes a számunkra, hogy normál és titkos üzemmódban milyen kulcskészítési algoritmusokat alkalmazhatnak a HCSXXX eszközök. Az üzenetek dekódolása előtt és után úgynevezett érvényesítési eljárásokat hajt végre a berendezés. Ez az jelenti, hogy megvizsgálja, hogy a sorozat szám, az azonosító érték 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 térek 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 kódolók 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 belől 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

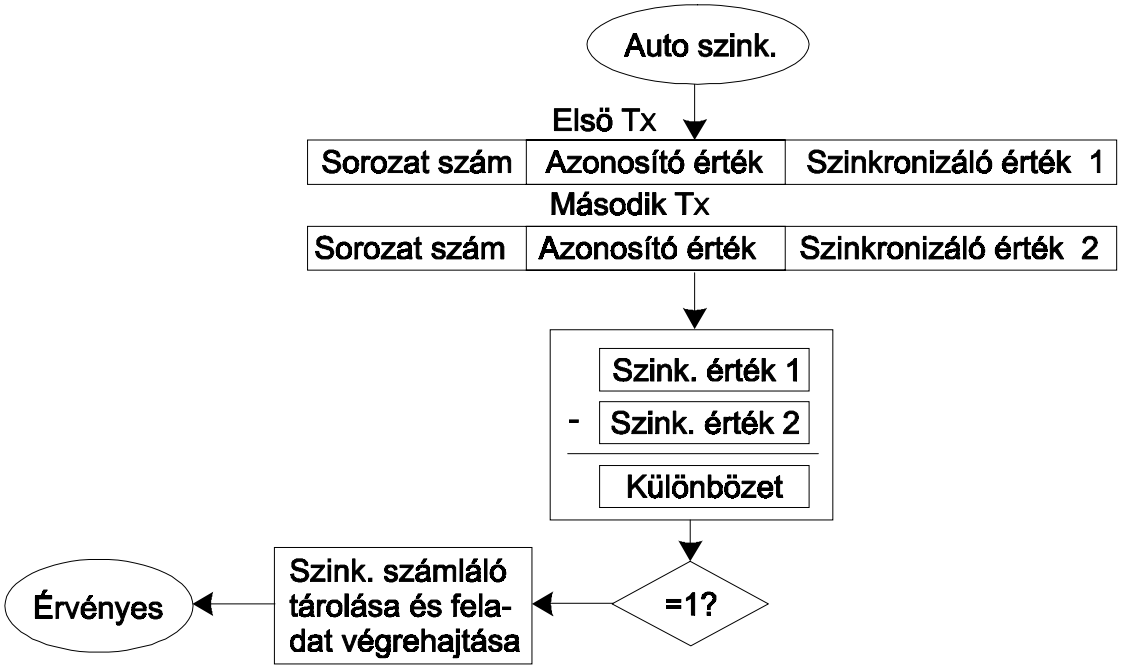
10.2 SZINKRONIZÁLÁS

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**).

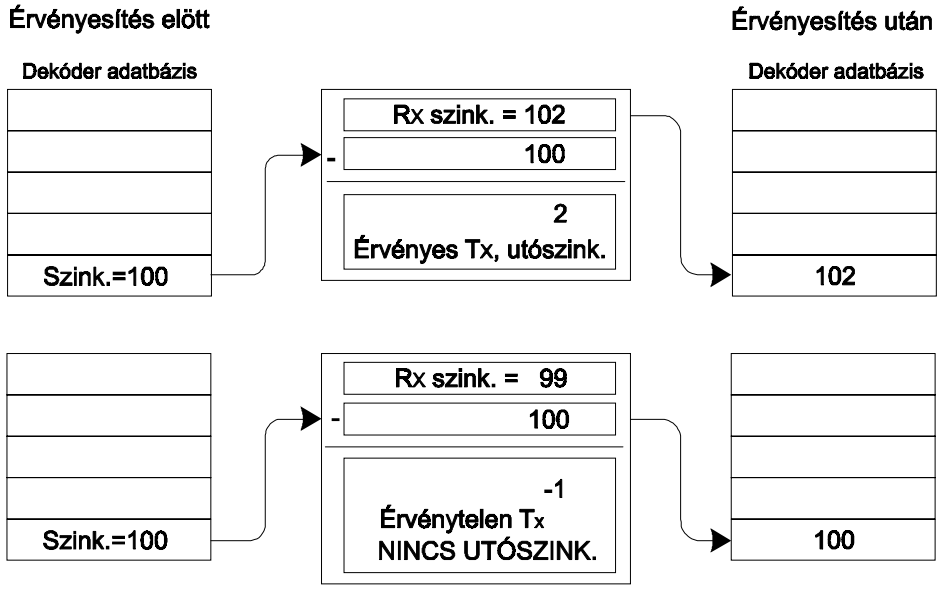


29. Ábra. Automatikus szinkronizáció

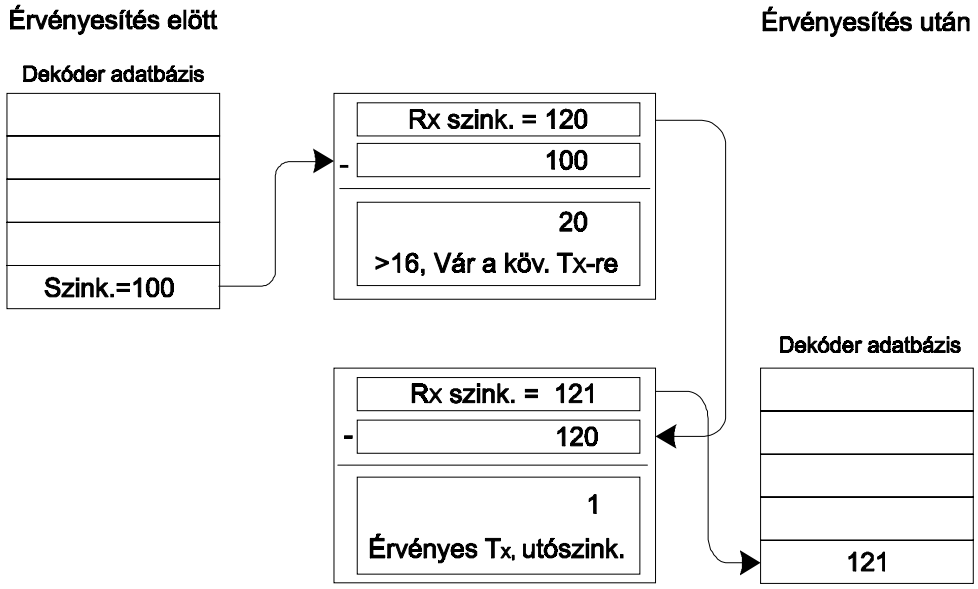
2. 16-nál kevesebb az eltérés: Ekkor az úgynevezett újraszinkronizáló tartományon belül van még a különbség, ami a tartomány nevéből is adódóan azt jelenti, hogy automatikusan

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. Automatikus újra szinkronizáció és elutasítás



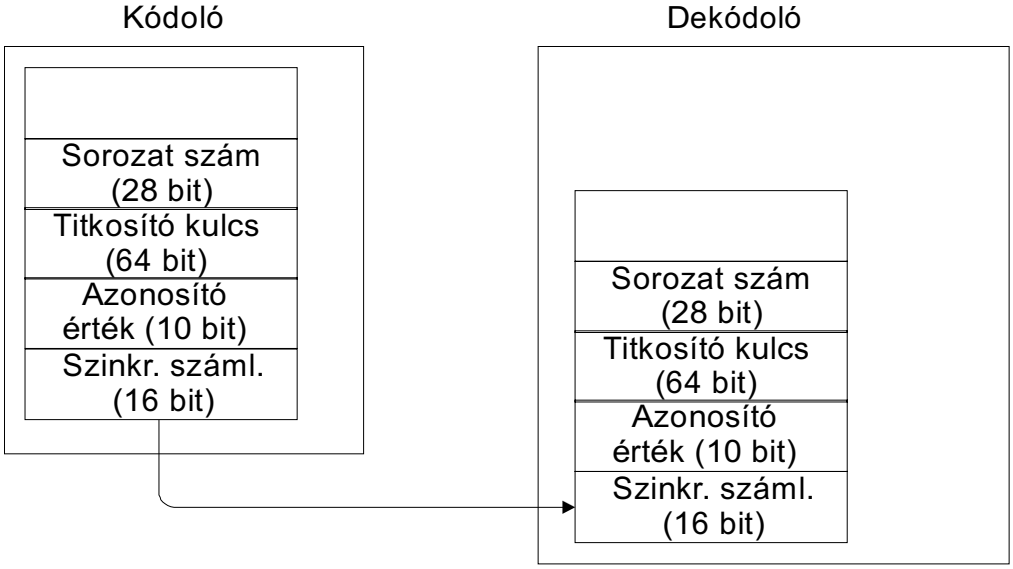
31. Ábra. Újra szinkronizáció

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 felöl 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 programozzunk 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 kétfele folyamata az ismeretes. Az egyik a normál tanítás, a másik, a titkos tanítás. A következőekben tekintsük át ezeket részletesebben.

11.1 NORMÁL TANÍTÁS

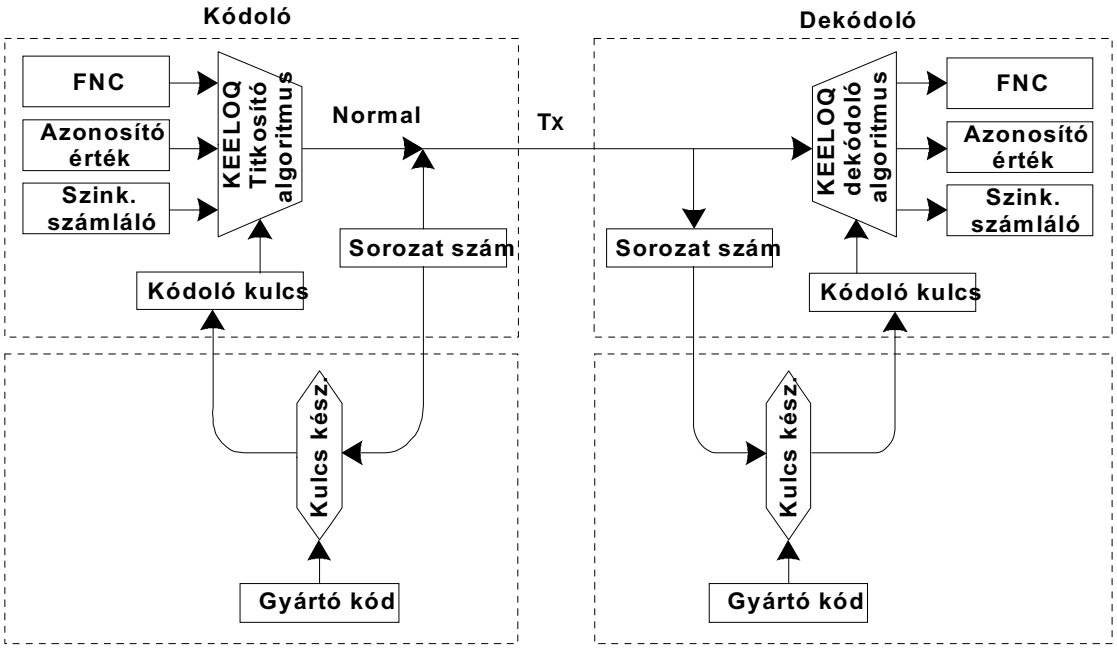
A normál tanítási folyamathoz a kódoló, a kódolt üzenet elkészítéséhez a titkosító kulcsot a normál kulcskészítő algoritmus (sorozat számból és gyártó kódból) alapján készíti el, mint ahogy az ábrán is láthatjuk. Ebben az esetben a kódoló az ugró kódos rész helyén mindig a normál, kódolt információkat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

A következőekben a dekódoló felöl 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 előbbiekben láthattuk, hogy a tanítási folyamat során rendkívül sok ellenőrzésre, egyeztetésre van szükség. Ez esetekben el kell utasítani a tanulási eljárást, és elölről 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…

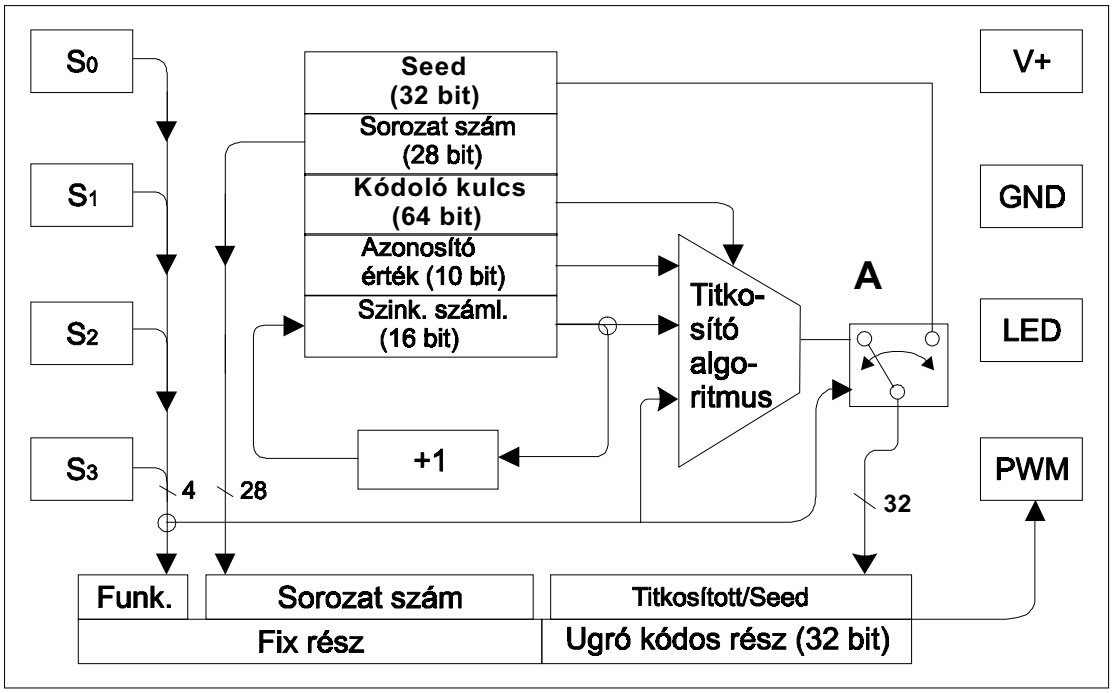
- a támadók nem tudják felhasználni hasznos információnak az átvitelt.
- fizikai behatás kell a kódoló kulcs készítő információért.

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 helyet 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 grafikusan 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.

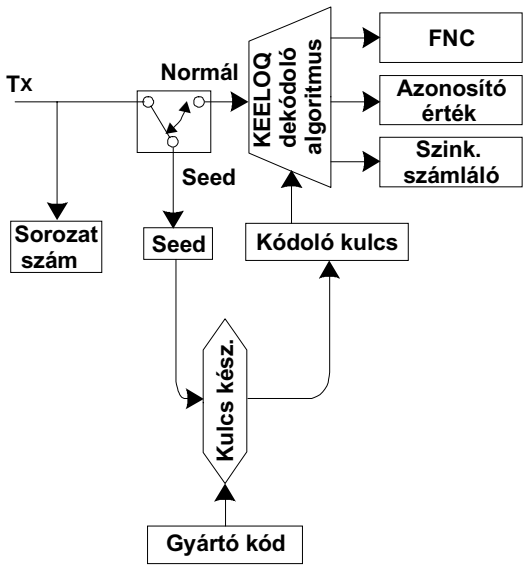


35. Ábra. Titkos tanulás során az üzenetkészítés folyamata

A tanulás folyamata (**36. ábra**):

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

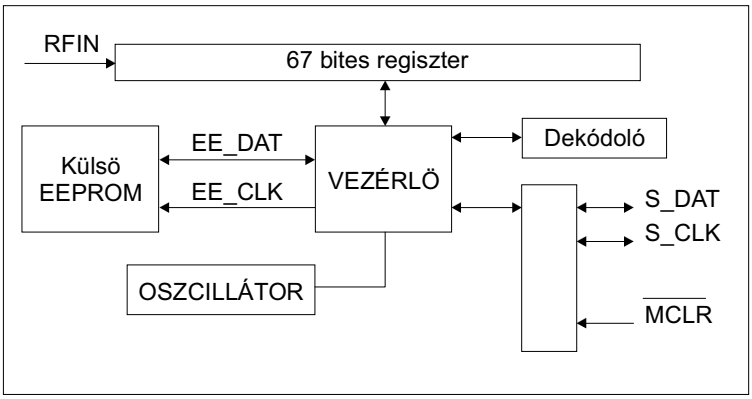
- Figyelni a tanító bemenetet
  – Bekapcsolni a jelző LED-et, amikor a tanítási mód aktív
- Várni a kódoló T_x jelére, amelyik a seed-et tartalmazza.
- A SEED felhasználásával elkészíteni a kódoló kulcsot. Ekkor eltárolja a sorozat számot átmenetileg.
- Ki lehet kapcsolni a LED-et
- Várni az ugró kódos 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.



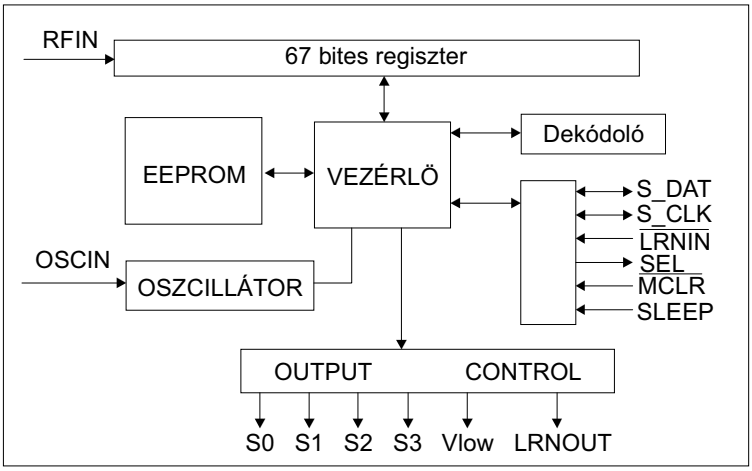
36. Ábra. A titkos tanulás folyamata

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 blokkséméját 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	Akkumlátor 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: 10KΩ 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 lábkiosztása

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^2C EEPROM-hoz
3	EE_DAT	I/O	TTL	Adatláb az I^2C 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 akkumlátor 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	Vlow	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**	*Leírás*
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.

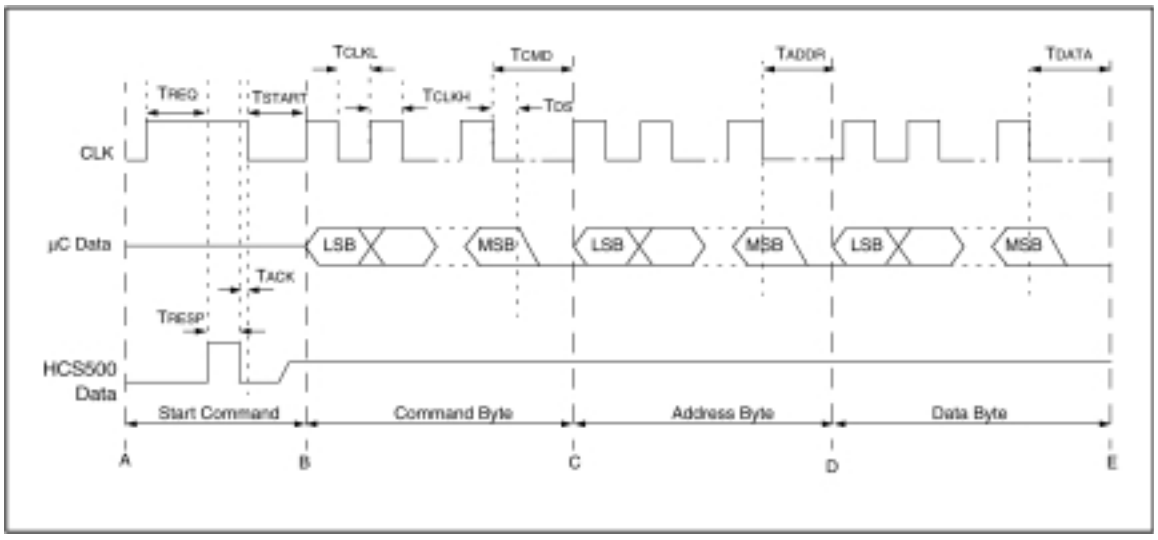
12.5.1.1 Érvényes átviteli üzenet

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 elkezdi 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 kódolók 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 órajel 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

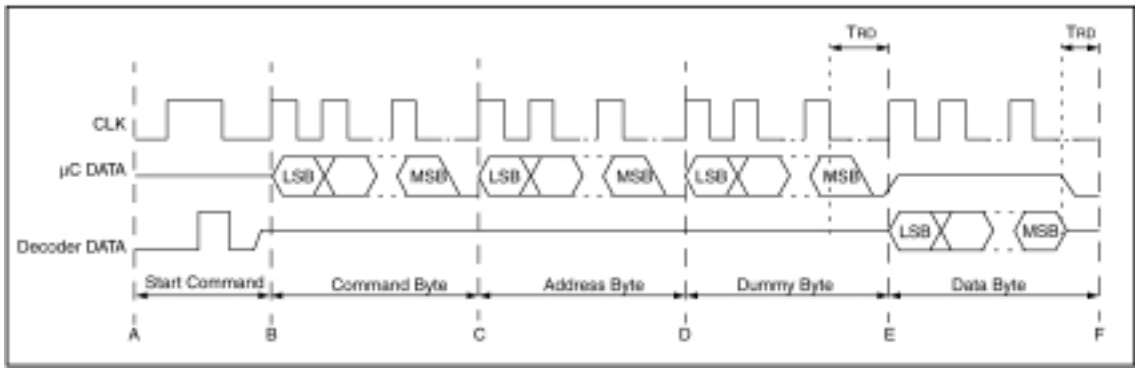
A **32. táblázat** mutatja, hogy milyen utasításokat használhat a vezérlő mikrokontroller.

32. Táblázat. A dekódoló parancsai

Utasítások	*Utasítás bájt*	**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

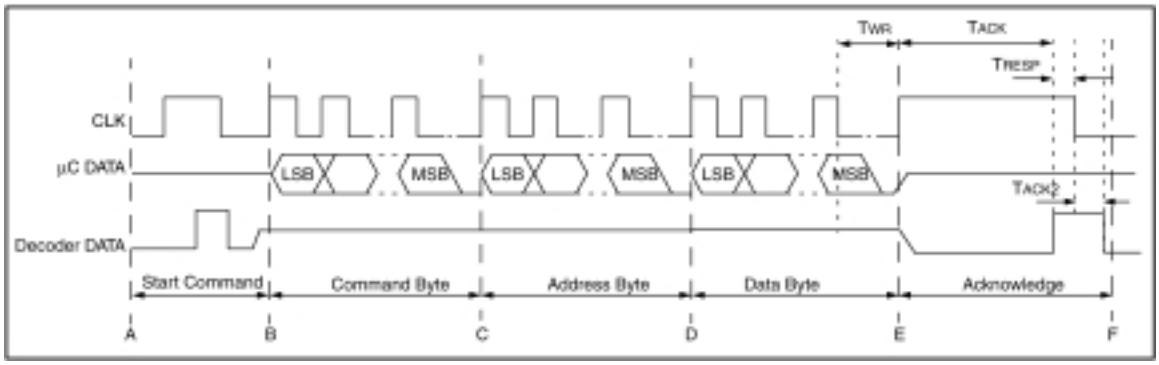
Az EEPROM-ból való olvasás folyamatát a **42. ábrán** láthatjuk. A memóriában való eltolást (offset) a cím bájt határozza meg (lásd ábra C-D). Azután egy látszólagos bájt következik (D-E). A adat bájtok a következő órajel felfutó élére fognak átvitelre kerülni, mégpedig úgy, hogy az alsó biteknél kezdődően (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

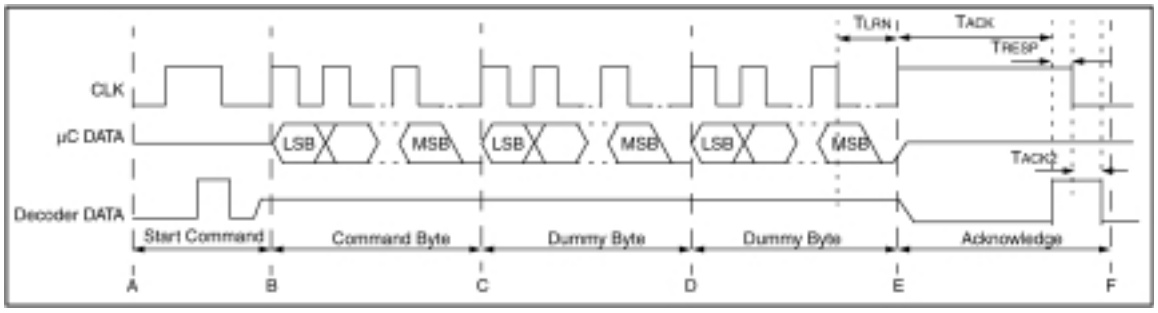
Az EEPROM-ba való írás folyamatát a **43. ábrán** láthatjuk. Először megvizsgálja, hogy milyen címre kell az adatokat írnia és utána az alsó biteknél kezdi az írást. A folyamatos írást is lehetővé teszi a rendszer és ekkor a D-F szakaszt kell ismételgetni. A dekódoló befejezi a parancs végrehajtását, ha 1.2 ms-nál több ideig nem kapja meg az órajelet. A sikeres memóriaírás 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

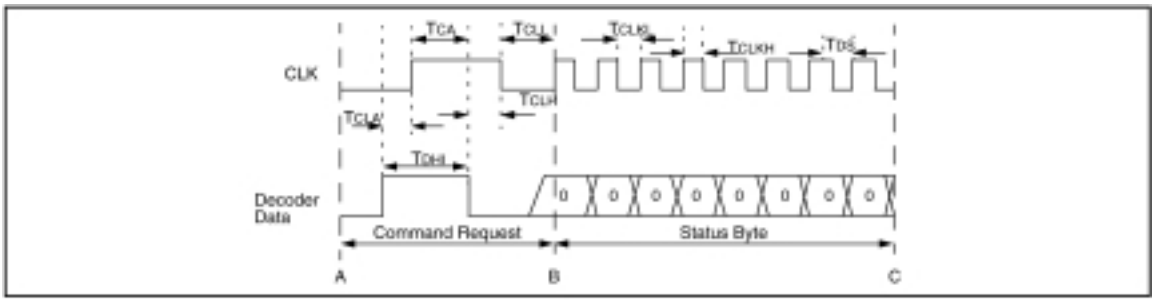
Tanulás aktiválása

A tanulásaktiváló parancsot arra használják, hogy aktiválja a tanulási feladatot a dekódolóban (**44. ábra**) Az utasítás egy parancs üzemmód aktiváló sorozatból, egy parancs bájtból és két látszólagos bájtból 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

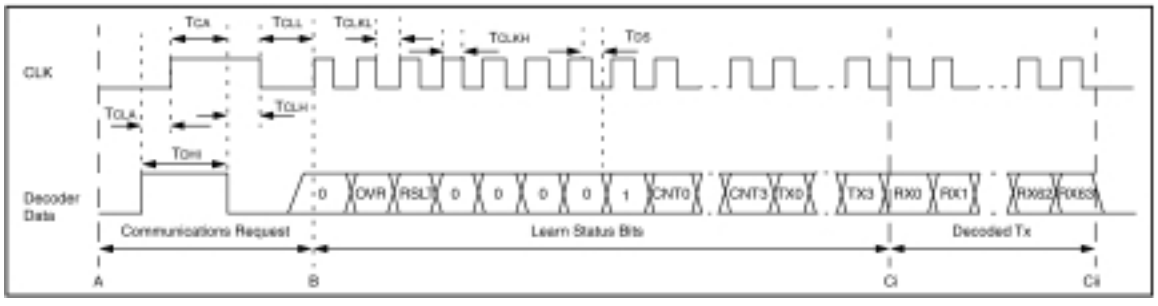
Amint a dekódoló megkapja az első üzenetet, azután egy tanulási állapotot jelző üzenettel válaszol (**45. ábra**).



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:

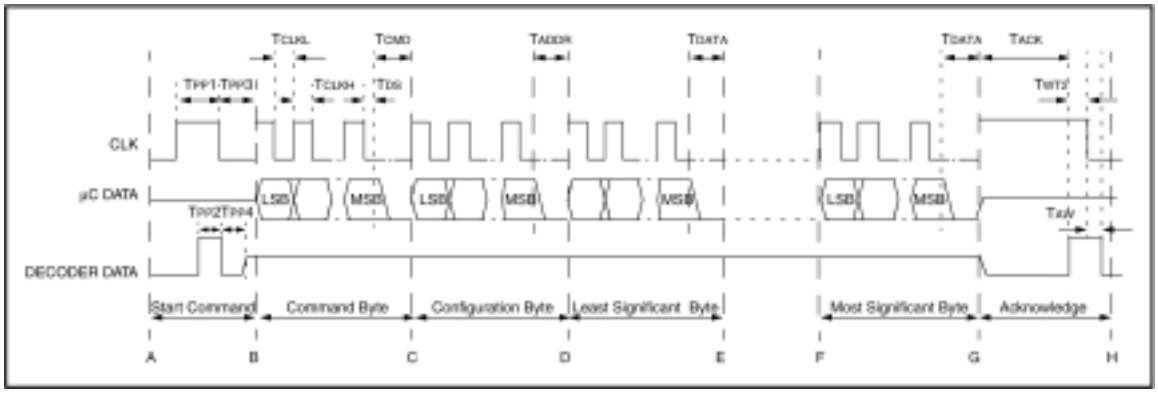
- Titkosító kulcs = gyártó kód
- Sorozat szám = bármilyen érték
- Azonosító bitek = a sorozat szám alsó 10 bitje
- Szinkronizáló számláló értéke = bármilyen érték (nem vesszük figyelembe)

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 bájt (B-C)
- Konfigurációs bájt (C-D)
- A gyártó 8 adat bájtja (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 kódolók 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

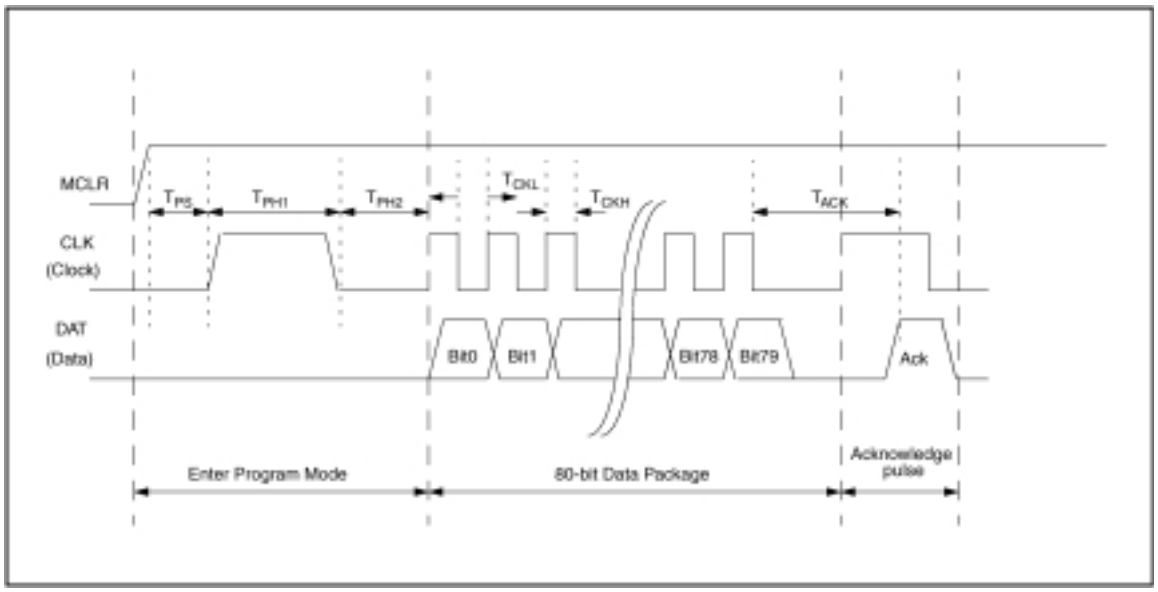
A gyártó kódot kell beprogramozni a HCS512 EEPROM-jába. Ezt a soros interface-n keresztül tehetjük meg a CLK és DATA vonalak használatával.

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 bájtból 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 bájtját összeadja (túlcsorduló biteket levágja) és az eredménynek nullának kell lennie. Magát a checksum bájtot pedig úgy határozzuk meg, hogy összeadjuk az üzenet első 9 bájtját (túlcsorduló bitek levágásával) és az eredményt pedig kivonjuk nullából.

13 ADATÁTVITELI MÓDSZEREK ALKALMAZÁSA AZ UGRÓ KÓDOS RENDSZEREKBEN

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 érzékenységű. Itt elsősorban a nagy infravörös tartalmú világítás (izzólámpák) a mértékadó. Ennek a hatásnak a csökkentésére szoktak használni a fotodióda elé infravörös szűrőt.

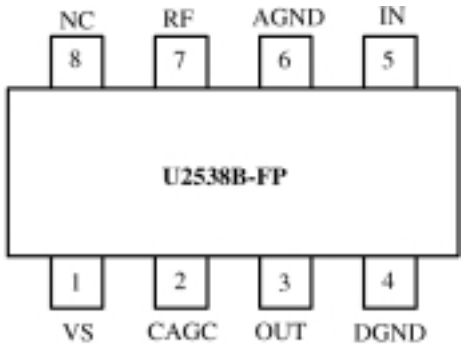
A jelátvitel lényege egy kapcsolójel kiküldése, majd ennek szelektív vétele. A kapcsolójel disszipációs, fényhasznosítási, telepkímélő stb. szempontból szaggatott, és hosszát gyakorlati értékek határolják be alulról és felülről. A kisugárzott energia szempontjából az impulzuscsomagnak minél rövidebbnek 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 bemutatnék 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 értelemszerűen „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 érzéketlenné teszi a vevőt a környezeti fényforrásokkal szemben.
A vivő frekvencia 20-60kHz.

13.1.1.1 Lábkiosztás

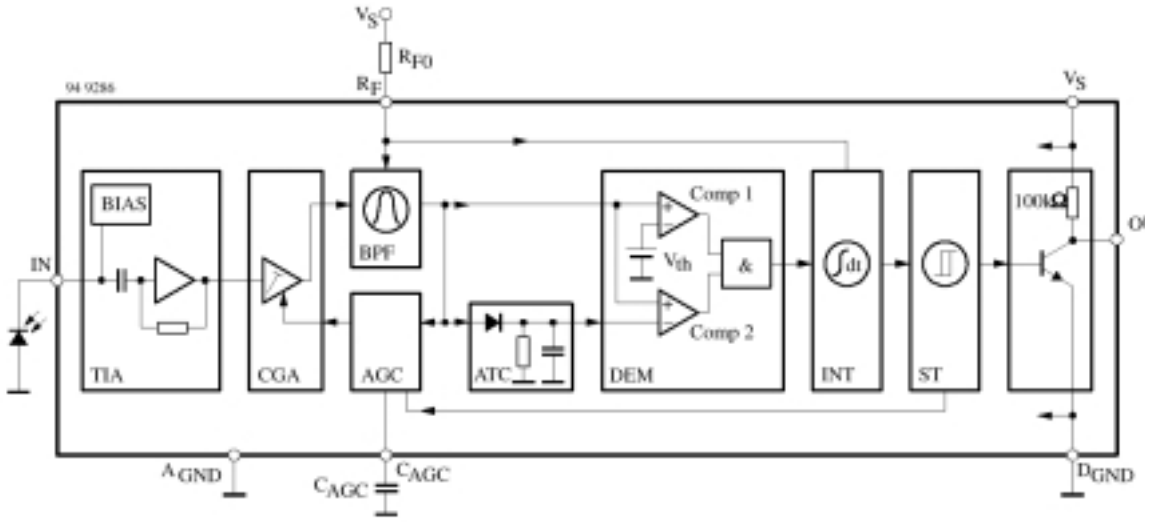


50**. Ábra. A TEMIC által gyártott U2538B lábkiosztása**

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 blokksémája**

TIA	Transimpedance erősítő	**ATC**	Automatikus küszöbérték szabályozó
CGA	Szabályozott teljesítményerősítő	**DEM**	Demodulátor
BPF	Sávátereszt 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<10kΩ). 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} lábnál 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 alapból tartalmaz beintegrált komponenseket. Egy külső ellenállás határozza meg a középfrekvenciát. A következő képlet használatával kiszámíthatjuk az ellenállás értékét (R_{f0}):

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

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, akkot megőrzi az adatokat. Ez abban az esetben történhet meg, hogy ha bementi jel erőssége kétszer nagyobb, mint a minimálisan kimutatható jel intenzitása.

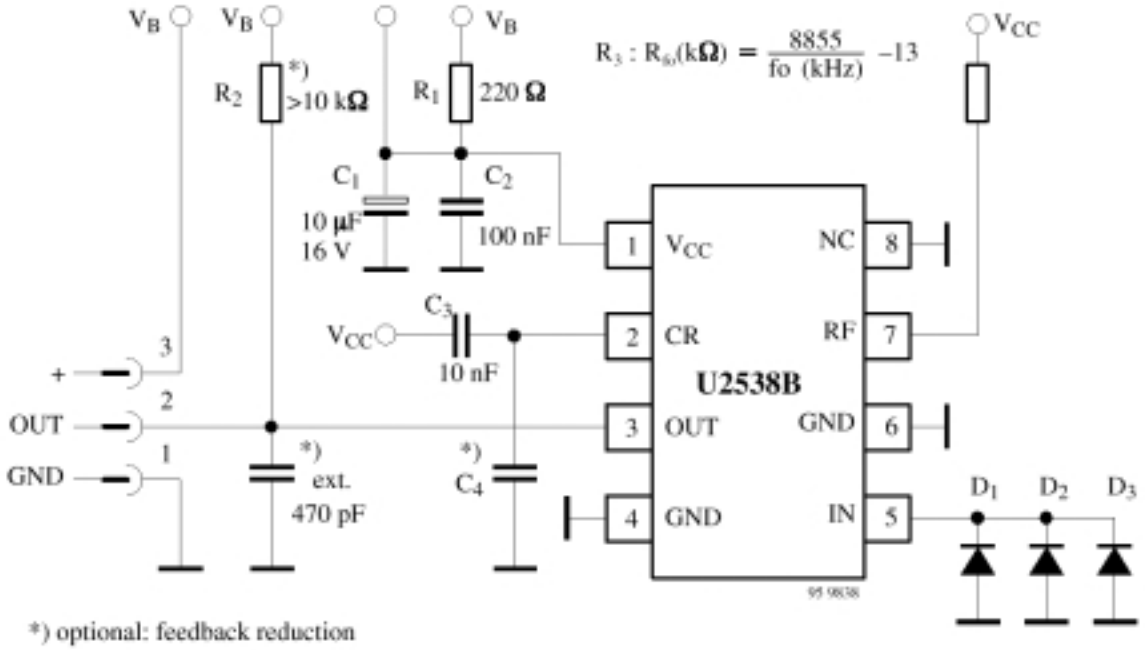
13.1.1.7 Automatikus erősítés szabályozó (AGC)

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 sávátereszt 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 hatásfokú 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.

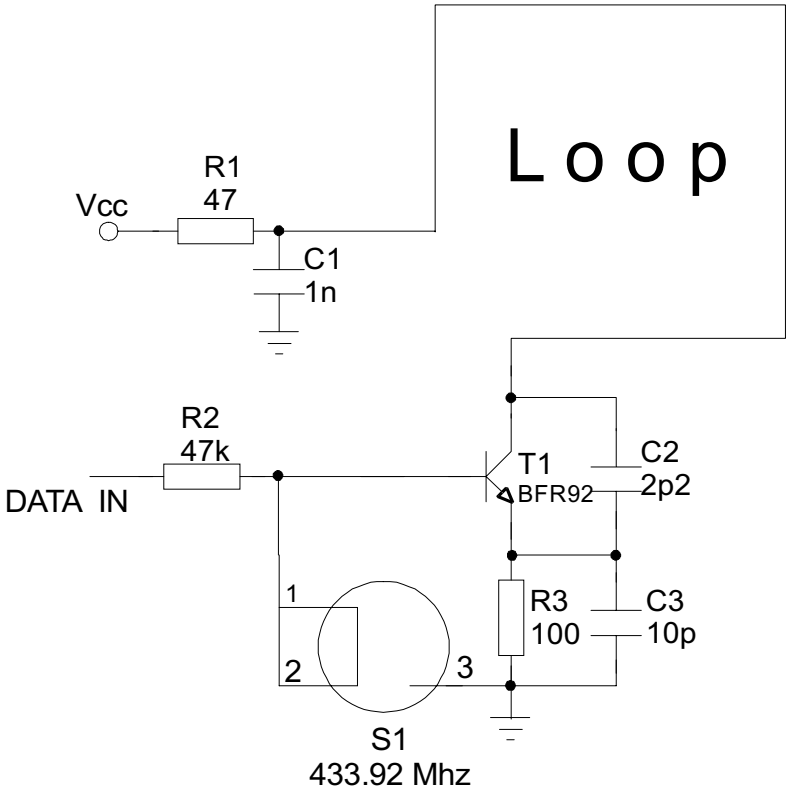
Jelentős az adónál energiatakarékossági szempontból, hogy vivő teljesítmény csak akkor van jelen, amikor a „vezérlő” PWM jelnek magas szintje van, vagyis csak a moduláció ideje alatt működik az adó vivőfrekvencia „generátora”.

Számos megvalósítási lehetőségeket találhatunk és készíthetünk a rádiófrekvenciás adatátvitelekre. A következőkben ismertetném a CHIPCAD által forgalmazott RF adó- és vevő egységet, továbbá bemutatnék más cégek által terjesztett komplett vevő IC-ket 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

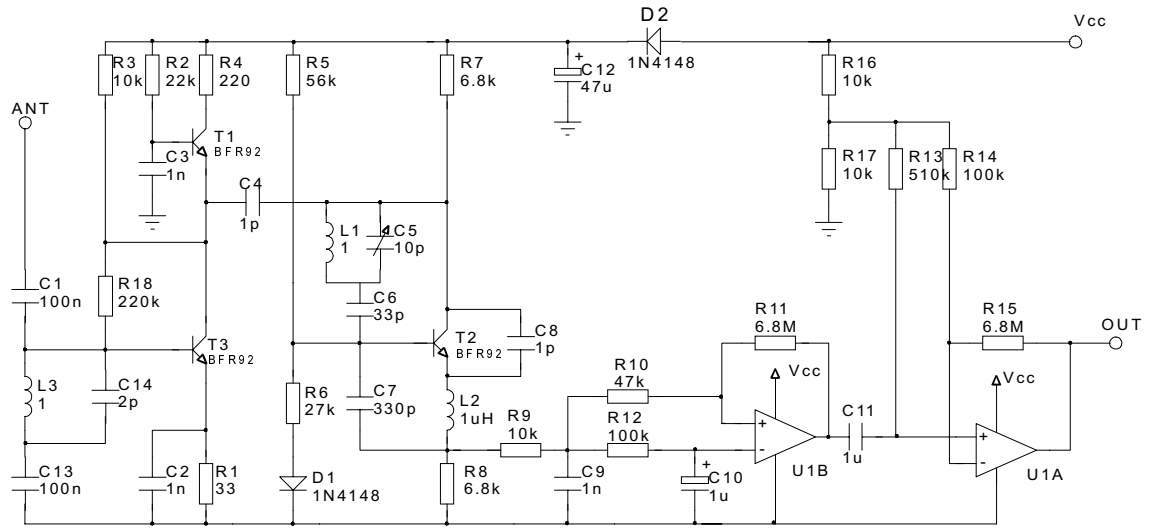


53. Ábra. A CHIPCAD által forgalmazott RF adó

A DATA IN vonalon kapja a kódolótól a PWM modulált üzenetcsomagokat. 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 zavarszűréshez 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 négyszög jelről). Majd ennek a kimenete egy Schmitt-trigger bemenetéhez csatlakozik, mely elkészíti a végleges kimeneti formátumot.



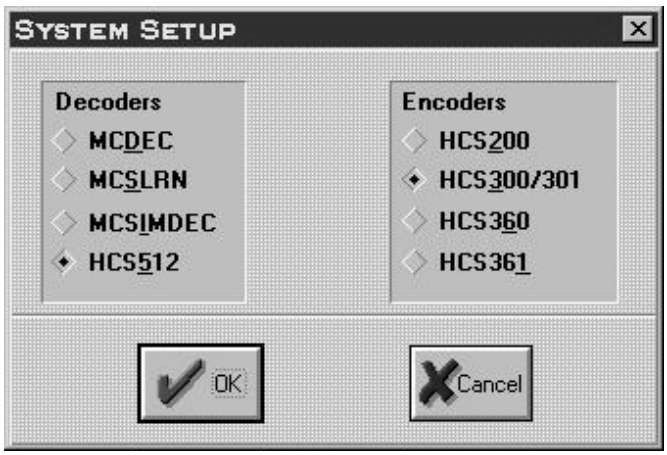
54. Ábra. A CHIPCAD által forgalmazott RF vevő

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:

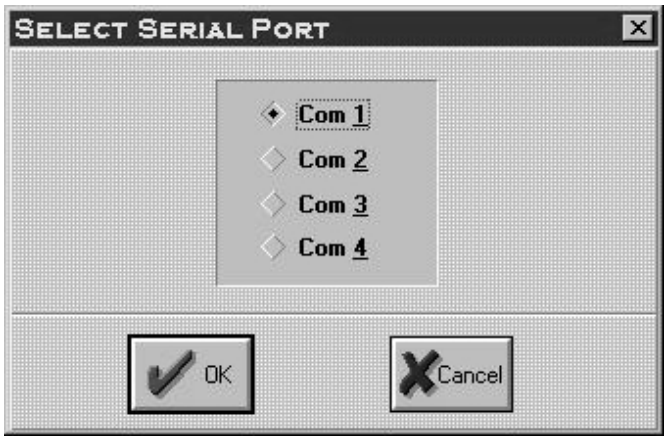
- gyártó kód

- rendszer beállítás



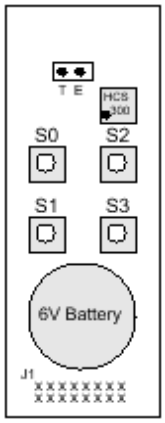
Itt választhatjuk, ki hogy milyen eszközöket akarunk alkalmazni. Mindegyikből csak egyet választhatunk, ami azt jelenti, hogy egy rendszeren belül csak egy féle kódolókat használhatunk. Láthatjuk, hogy a HCS500 dekódolót, a HCS410 kódolót még nem ismeri a program.

- soros port választás



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

14.1 A KÓDOLÓ

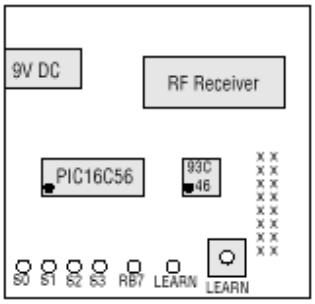


A szett tartalmaz két db négy nyomógombos demonstrációs adót. Ezen az összes funkciógomb kombinációt ki lehet próbálni. Két db 3V elem van benne, azért, hogy lehessen tesztelni az alacsony feszültség jelző funkcióját is a rendszernek. Az ábrán is látható jumper arra jó, hogy az RF adót leválasszuk a 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.

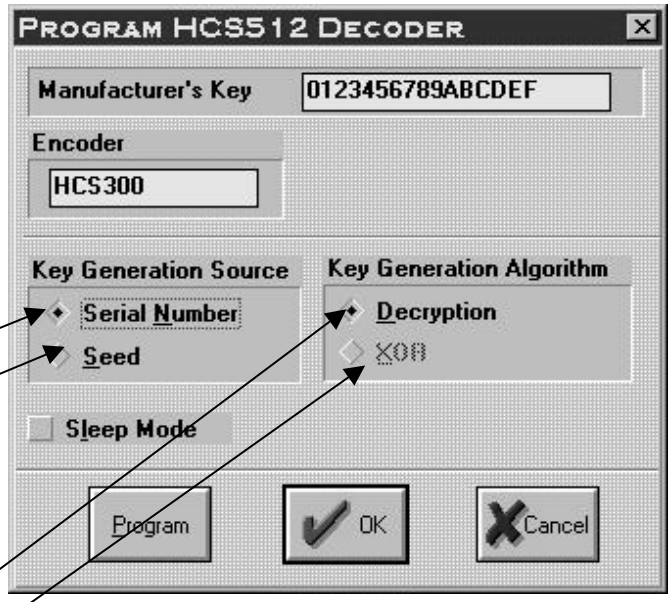


14.3 A DEKÓDOLÓK BEÁLLÍTÁSAI

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

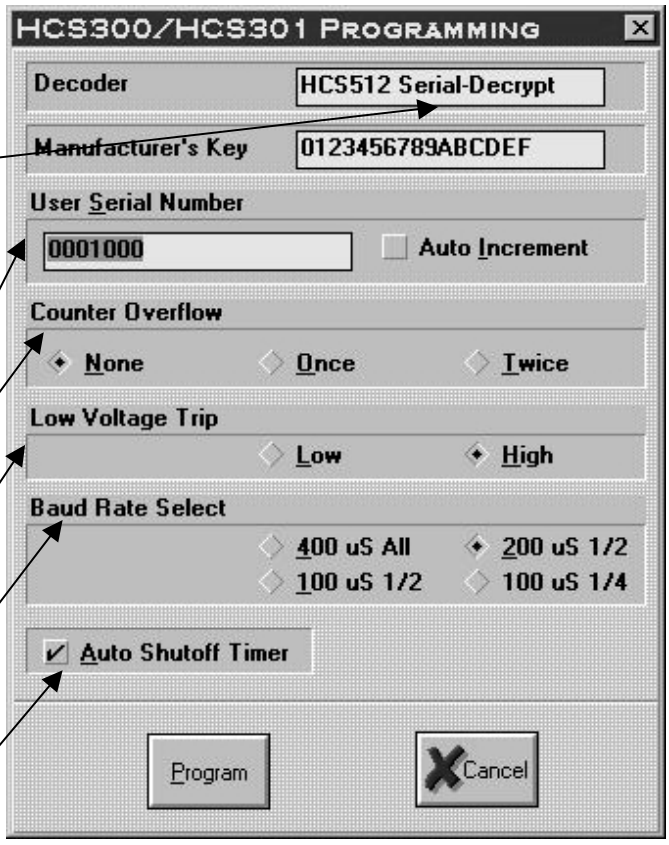
Az XOR algoritmust csak a seed forrás során választható.

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.

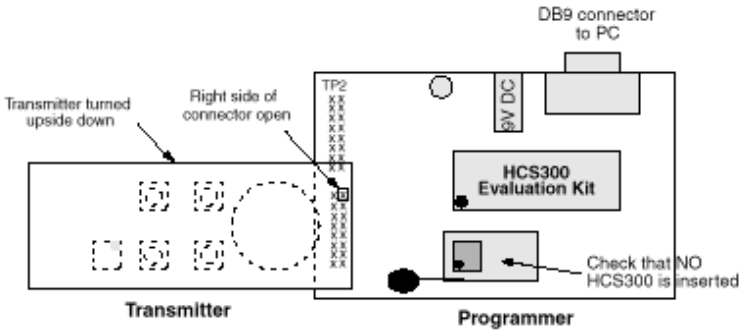


Itt mi állíthatjuk be:

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

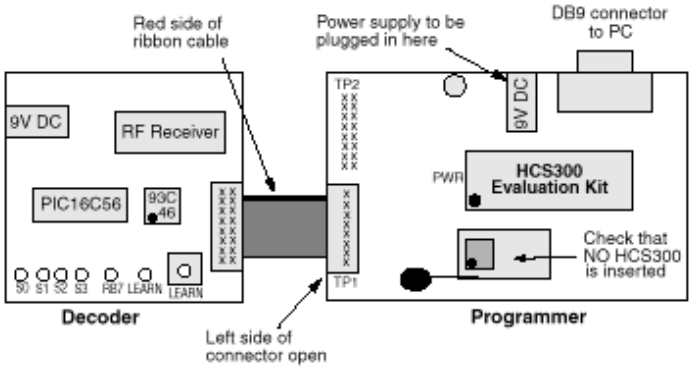
14.5 A KÓDOLÓK PROGRAMOZÁSA

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 kódnak kell szerepelnie az egy rendszerben használt eszközöknek.

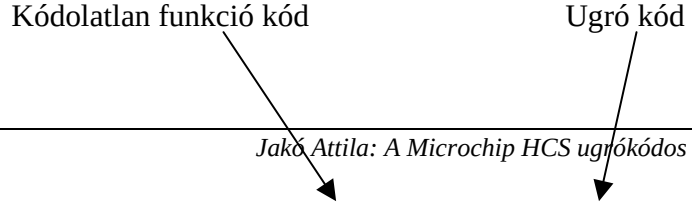
A tanulási folyamat, a már ismertetett módon történik.

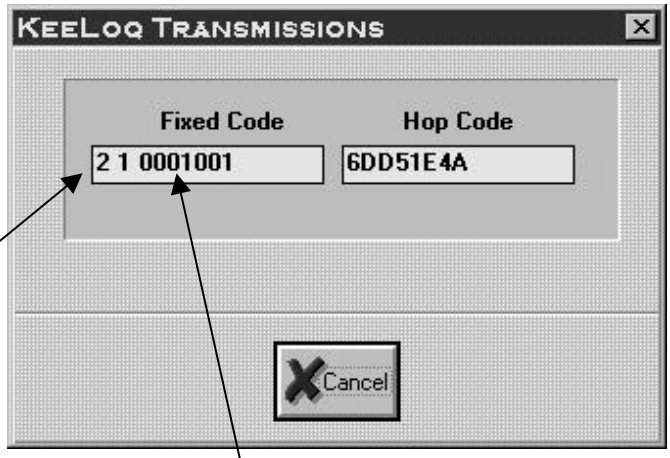
A tanulási eljárás:

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.





Ismétlésjelző és feszültség alacsony szintjét jelző bitek, jelen esetben azt jelzi, hogy ismétlés történt, ha alacsony lenne a feszültség, akkor 0 lenne az értéke

Sorozat szám

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 érvénytelenné válik, és elölről 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 elölről 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

16 TÁBLÁZAT JEGYZÉK

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