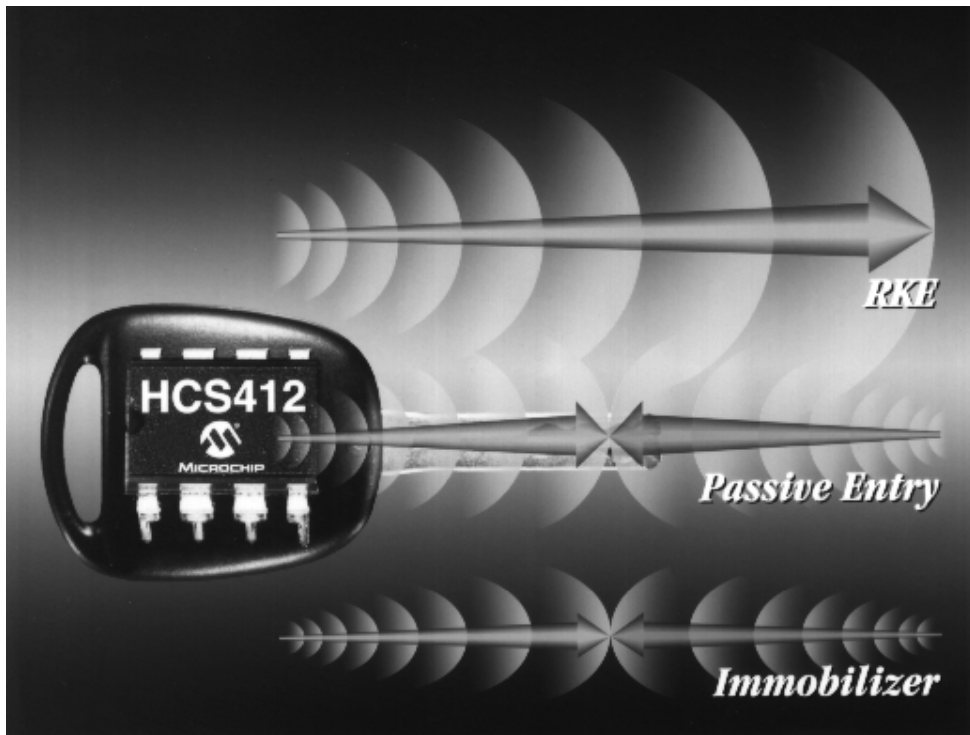


Jakó Attila

A Microchip HCS ugrókódos áramkörei



1 TARTALOMJEGYZÉK

1	TARTALOMJEGYZÉK	1
2	BEVEZETÉS	4
3	A KEELOQ ALKOTÓELEMEI.....	6
4	UGRÓ KÓDOS RENDSZEREKNÉL HASZNÁLT DEFINÍCIÓK.....	8
4.1	SOROZAT SZÁM	8
4.2	TITKOSÍTÓ KULCS.....	8
4.3	AZONOSÍTÓ ÉRTÉK	8
4.4	SZINKRONIZÁLÓ SZÁMLÁLÓ	8
4.5	GYÁRTÓ KÓD.....	9
4.6	SEED.....	9
5	A RENDSZER ÁTTEKINTÉSE	10
6	AZ ÜZENETEK (ÁTVITELEK) ÖSSZETÉTELE.....	14
6.1	UGRÓ KÓD	16
6.1.1	<i>FUNKCIÓ INFORMÁCIÓK.....</i>	<i>16</i>
6.1.2	<i>TÚLCSORDULÁST JELZŐ BITEK.....</i>	<i>16</i>
6.2	FIX KÓD	17
6.2.1	<i>VLOW: Alacsony feszültség szintjelző.....</i>	<i>17</i>
6.2.2	<i>RPT: ISMÉTLÉS JELZŐ.....</i>	<i>17</i>
6.2.3	<i>CRC (Ciklus redundancia ellenőrző) bitek.....</i>	<i>17</i>
6.3	SEED ÁTVITELEK AKTIVIZÁLÁSA.....	18
7	A KEELOQ KÓDOLÓK ÉS DEKÓDOLÓK TITKOSÍTÓKULCS KÉSZÍTÉSEI.....	19
7.1	TITKOS KULCSKÉSZÍTÉS (SEED A FORRÁS).....	20
7.1.1	<i>XOR algoritmus használata a kulcskészítéshez.....</i>	<i>21</i>
7.1.2	<i>„Titkosító” algoritmus használata a kulcskészítéshez.....</i>	<i>21</i>
8	A KEELOQ KÓDOLÓK ÜZENETKÜLDÉSÉNEK A FOLYAMATA.....	23
9	A KEELOQ KÓDOLÓK	25
9.1	ÁLTALÁNOS JELLEMZŐK.....	25
9.2	LÁBKIOSZTÁS.....	28
9.2.1	<i>HCS200-as kódoló.....</i>	<i>28</i>
9.2.2	<i>HCS300/301 és HCS360/361 kódoló.....</i>	<i>28</i>
9.3	EEPROM MEMÓRIA KIOSZTÁS	29
9.3.1	<i>HCS200-as kódoló.....</i>	<i>29</i>
9.3.2	<i>HCS300 és HCS301 kódoló.....</i>	<i>31</i>
9.3.3	<i>HCS360 kódoló.....</i>	<i>34</i>
9.3.4	<i>HCS361 kódoló.....</i>	<i>42</i>
9.4	A KÓDOLÓK ÁLTAL ALKALMAZOTT MODULÁCIÓS FORMÁK.....	46
9.4.1	<i>HCS200, HCS300/301 kódoló.....</i>	<i>46</i>
9.4.2	<i>HCS360 kódoló.....</i>	<i>46</i>
9.4.3	<i>HCS361 kódoló.....</i>	<i>48</i>

9.5	SZINKRONIZÁLT ADATÁTVITELI MÓD (VEZETÉKES).....	51
9.6	SPECIÁLIS JELLEMZŐK.....	52
9.6.1	Auto-shutoff.....	52
9.6.2	Blank Alternate kód szó (BACW).....	52
9.7	PROGRAMOZÁS.....	54
9.8	HCS410.....	57
9.8.1	KEELOQ IFF.....	59
9.8.2	Készülék működése.....	59
10	A KEELOQ DEKÓDOLÓK ÜZENETDEKÓDOLÓ ELJÁRÁSA.....	61
10.1	ÉRVÉNYESÍTÉS.....	61
10.2	SZINKRONIZÁLÁS.....	62
11	TANÍTÁS, TANULÁS.....	65
11.1	NORMÁL TANÍTÁS.....	67
11.2	TITKOS TANÍTÁS.....	69
12	A KEELOQ DEKÓDOLÓK.....	72
12.1	JELLEMZŐK.....	73
12.2	LÁBKIOSZTÁS.....	74
12.2.1	HCS512.....	74
12.2.2	HCS500.....	75
12.3	MŰKÖDÉSI LEÍRÁSOK.....	75
12.4	KONFIGURÁCIÓS BÁJT.....	76
12.4.1	HCS500.....	76
12.4.2	HCS512.....	78
12.5	A HCS500 EGYÉNI JELLEMZŐI.....	79
12.5.1	Mikrokontrollerhez való csatlakozás.....	79
12.5.2	Programozása.....	86
12.6	A HCS512 EGYÉNI JELLEMZŐI.....	87
12.6.1	Adó tesztelése.....	87
12.6.2	Programozása.....	87
12.6.3	Checksum.....	89
13	ADATÁTVITELI MÓDSZEREK ALKALMAZÁSA AZ UGRÓ KÓDOS RENDSZEREKBE	90
13.1	INFRA ADATÁTVITEL.....	90
13.1.1	TEMIC U2538B.....	91
13.1.2	TEMIC U2535B-FP.....	94
13.2	RÁDIÓFREKVENCIÁS ADATÁTVITEL.....	95
13.2.1	CHIPCAD által forgalmazott RF adó, vevő.....	95
14	A CHIPCAD ÁLTAL FORGALMAZOTT HCS PROGRAMOZÓ SZETT RÖVID BEMUTATÁSA	98
14.1	A KÓDOLÓ.....	99
14.2	A DEKÓDOLÓ.....	100
14.3	A DEKÓDOLÓK BEÁLLÍTÁSAI.....	100
14.4	A KÓDOLÓK BEÁLLÍTÁSAI.....	101
14.5	A KÓDOLÓK PROGRAMOZÁSA.....	102
14.6	A DEKÓDOLÓ PROGRAMOZÁSA.....	102
14.7	TANÍTÁSI ELJÁRÁS.....	102
14.8	KEELOQ ÁTVITELEK FIGYELÉSE.....	103
14.9	TAPASZTALATOK.....	104
15	ÁBRAJEGYZÉK.....	106

16	TÁBLÁZAT JEGYZÉK.....	108
17	IRODALOMJEGYZÉK.....	109

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 távirányítós 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 adókat a vevő hatáskörén kívül folyamatosan működtetjük (pl. egy gyerek játszik a távvezérlővel). Ha a távirányító gombját több mint 16-szor lenyomjuk, akkor „elveszti” a szinkronizációt a rendszer. Azonban ezt követően két egymást követő átvitel helyreállítja a szinkronizációt. 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.

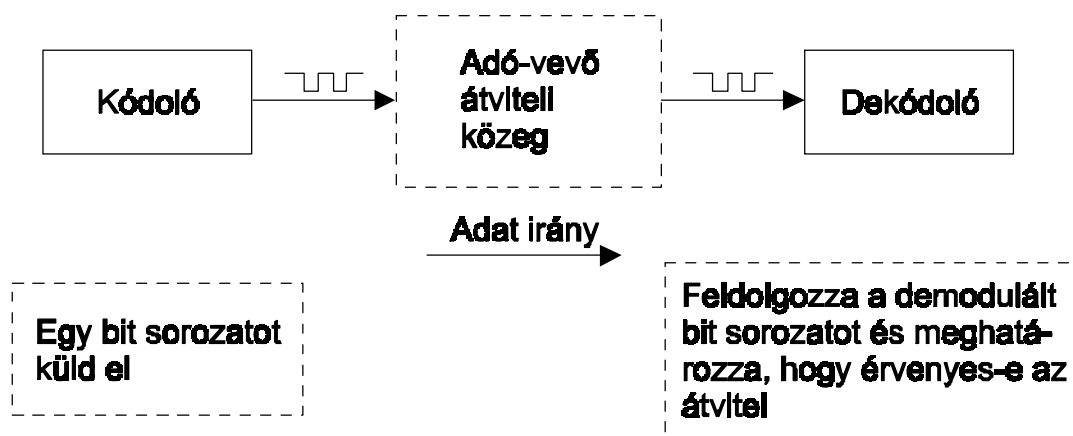
A rendszer működéséhez az adó és vevő eszközeinek egymásról bizonyos információkkal kell rendelkezniük ahhoz, hogy a rendszer működésképes legyen. Ehhez viszont egy úgynevezett tanítási eljárásra alkalmazására van szükségünk, mely igen fontos „szereplője” a rendszernek az üzemeltethetőség szempontjából. 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 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 feldolgozza (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 képezik. 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 túlsordulást jelző biteket küldenek el mindig, ezáltal a szinkronizáló számláló értékét kibővítik 65536-ról 196608-ra. A HCS360 és HCS361-es kódolók csak egy túlsordulást 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 kódolónként 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 képezi. 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 képezi az ugró kód alapját, emiatt fog minden egyes esetben változni a kód.

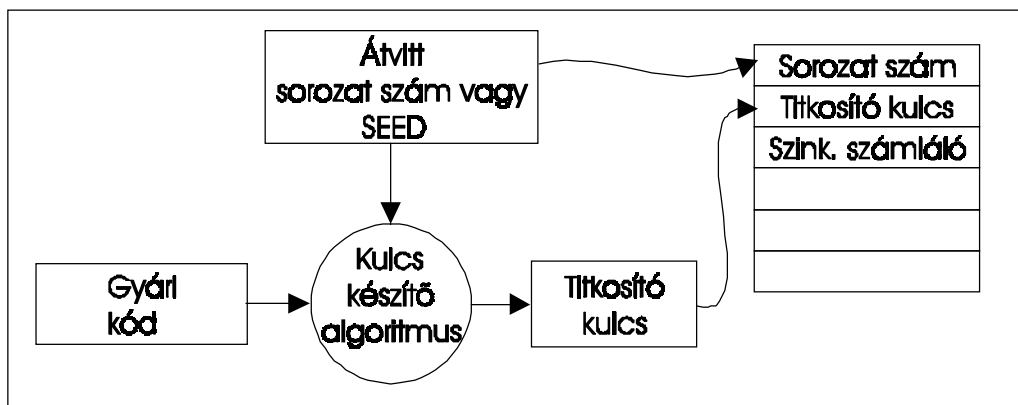
5 A RENDSZER ÁTTEKINTÉSE

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

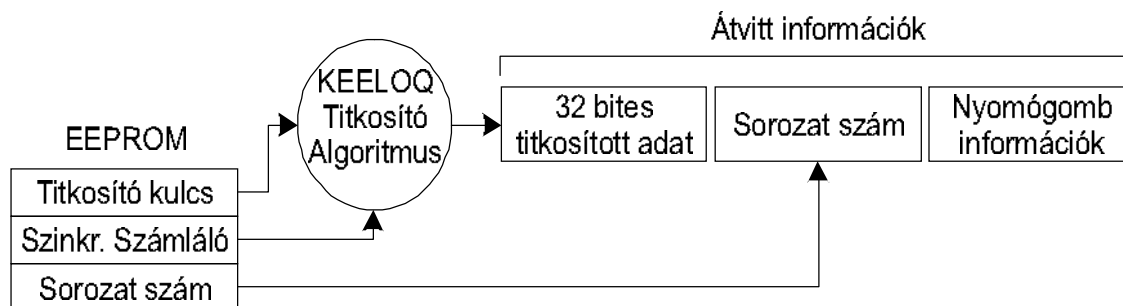
A sorozat számot és a gyártó kódot minden kódoló esetében a gyártás során programozzák be. A titkosító kulcs, a kulcs generáló algoritmus használata során készül (lásd. **2. ábra**) a sorozat számból és a gyártónak a 64 bites kódjából. A 16 bites szinkronozó érték az alapja annak, hogy minden gombnyomásra 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

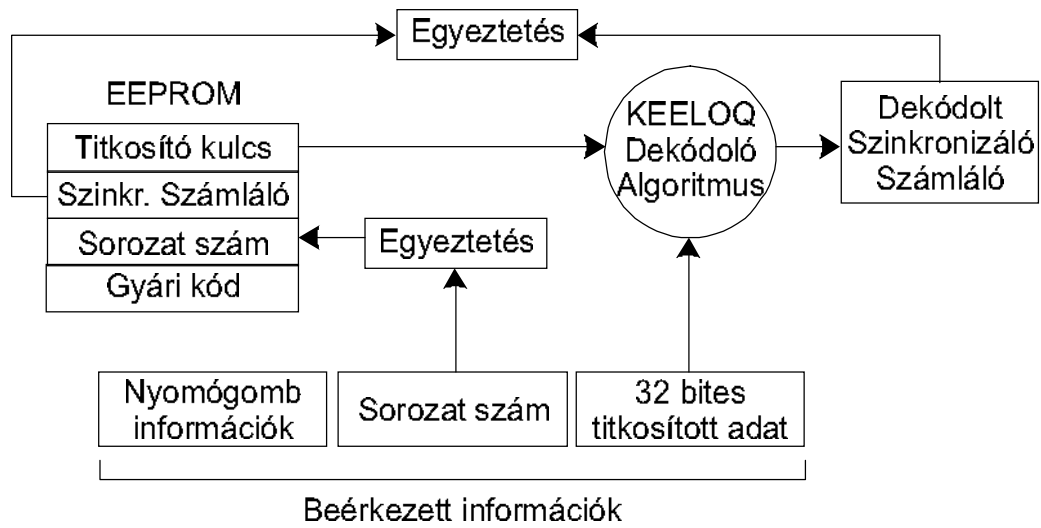
A **3. ábrán** látható az összefüggés az EEPROM-ban tárolt értékek között, és hogy a kódoló hogyan használja fel ezeket. A folyamat úgy történik, hogy először érzékeli a kódoló, hogy

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 alaplóműködése

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 grafikusán láthatóak a **4. ábrán**.

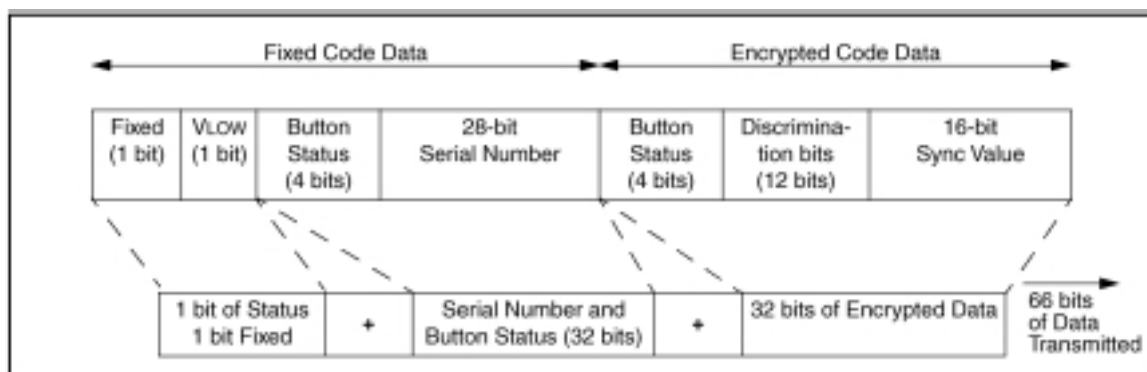


4. Ábra. A dekódoló alapl működése

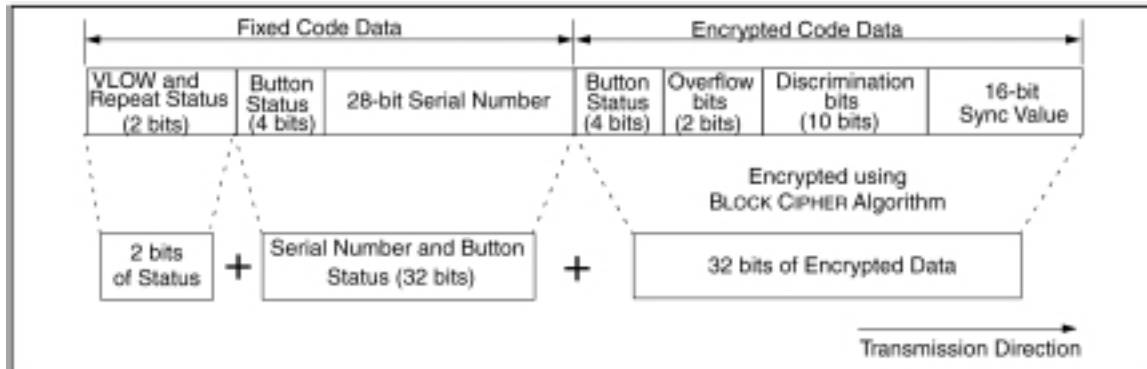
6 AZ ÜZENETEK (ÁTVITELEK) ÖSSZETÉTELE

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.

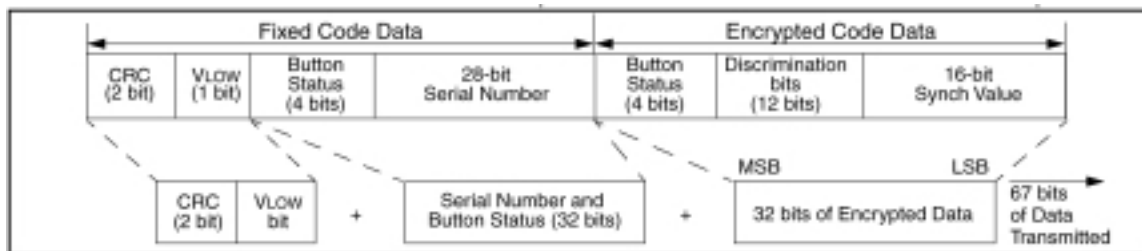
A kódoló által készített 66/67 bites átvitelek (üzenetek) két részből tevődnek össze. Az egyik része mindig változik, amikor az adókat aktiváljuk. Ezt nevezzük az ugró kódos résznek. A másik része tulajdonképpen a kódoló sorozat számából és egyéb bitekből áll, ezt pedig fix résznek nevezzük. Az üzenetek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



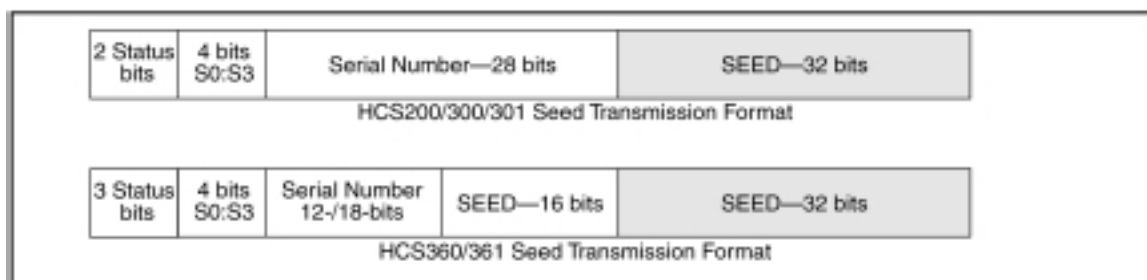
5. Ábra. A HCS200 által készített üzenet összetétele



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



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



8. Ábra. A kódoló 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 térnek 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 túlcordulást 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 végrehajtani 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. Ezeknek a státusz biteknek az értelmezését lássuk a következőkben.

6.2.1 VLOW: Alacsony feszültség szintjelző

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 kódolóknál 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**).

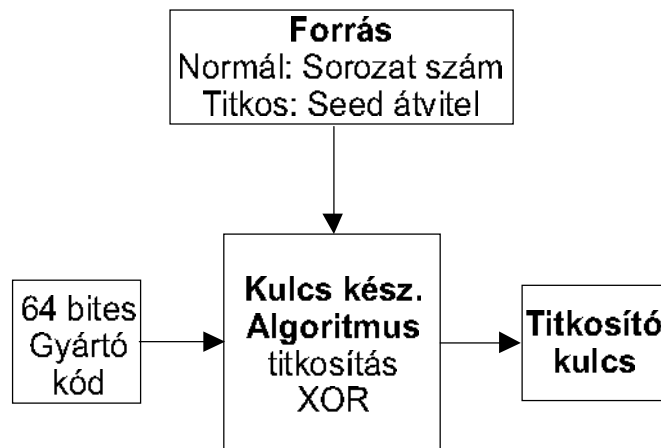
2. Táblázat. A seed átvitelek aktivizálási módjai kódoló típusonként

Kódoló	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 dekódolóban 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 harmadik 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)

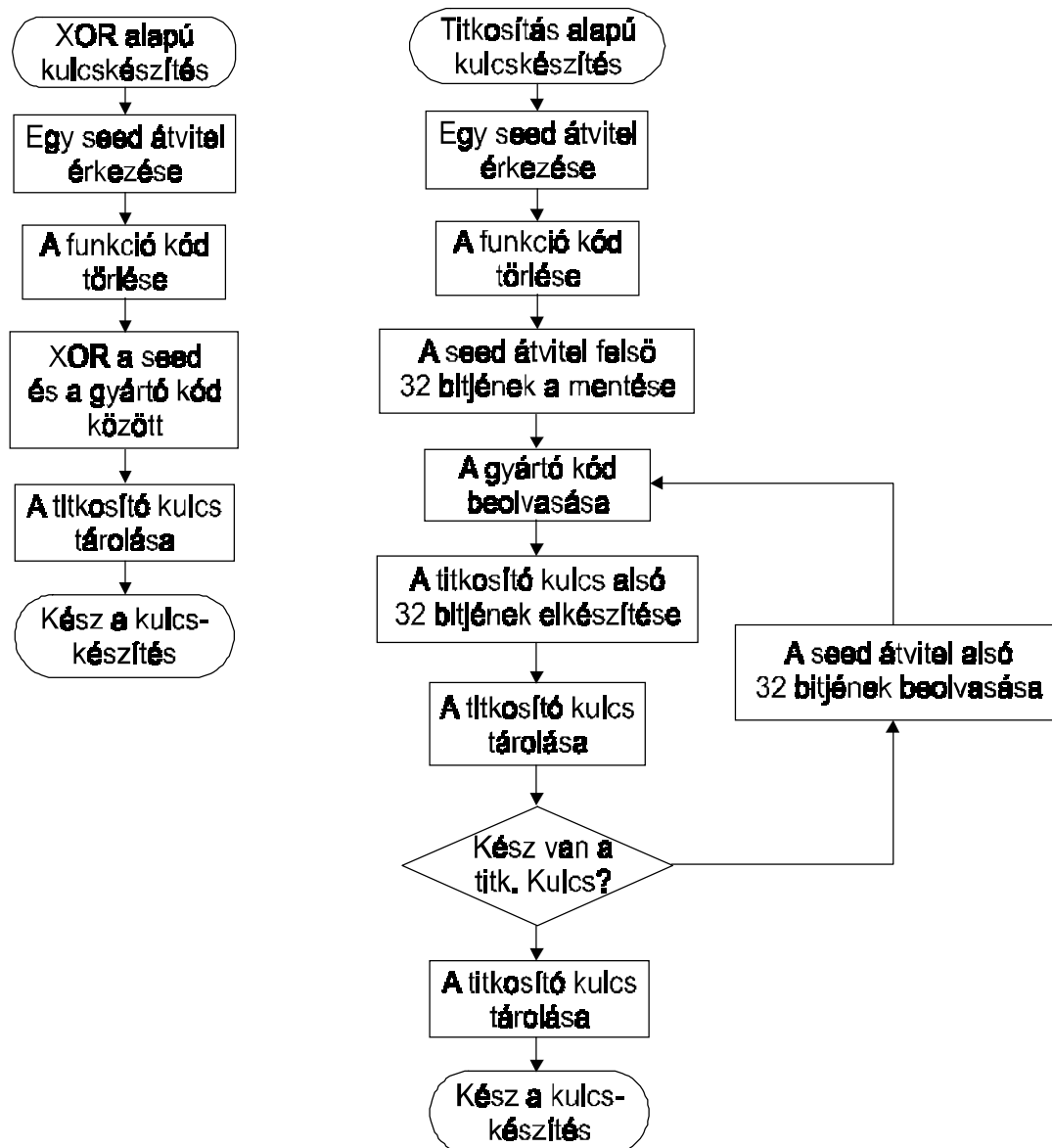
A titkos kulcs generálása a kódolónak a seed átvitelén alapszik, amely az algoritmus forrása és a tanítási folyamat alatt készül. A Microchip HCSXXX kódolók képesek az üzenet ugró kódos részén egy 32 vagy 48 bites seed átvitelére. A seed átvitel során a kód szó tartalmazza magát a seed értékét, a sorozat 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

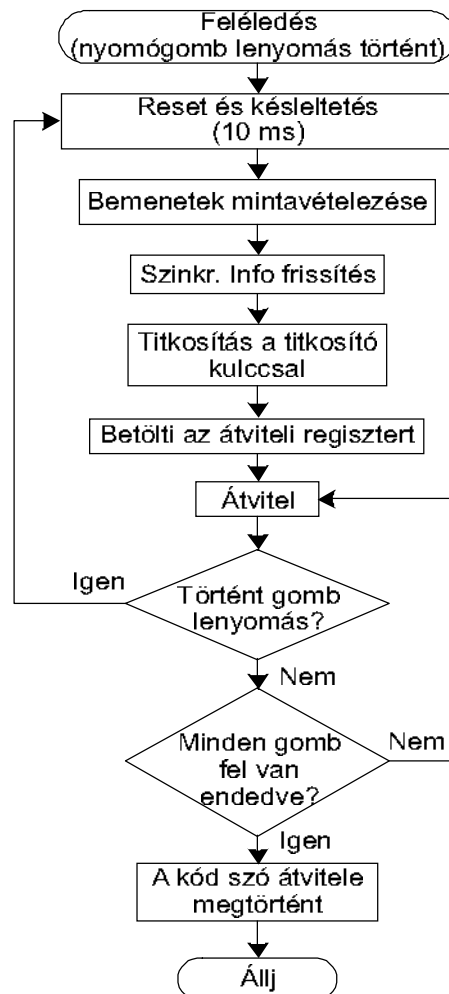
Ennek az algoritmusnak a használata során is az átvitelből a kódolatlan funkció kódok törlődnek. A **10. ábrán** is láthatjuk, hogy utána elmenti az átvitel felső 32 bitjét, majd beolvassa a gyártó kódot. Ezt követően elkészíti a titkosító kulcs alsó 32 bitjét és elmenti azt. Beolvassa az átvitel felső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

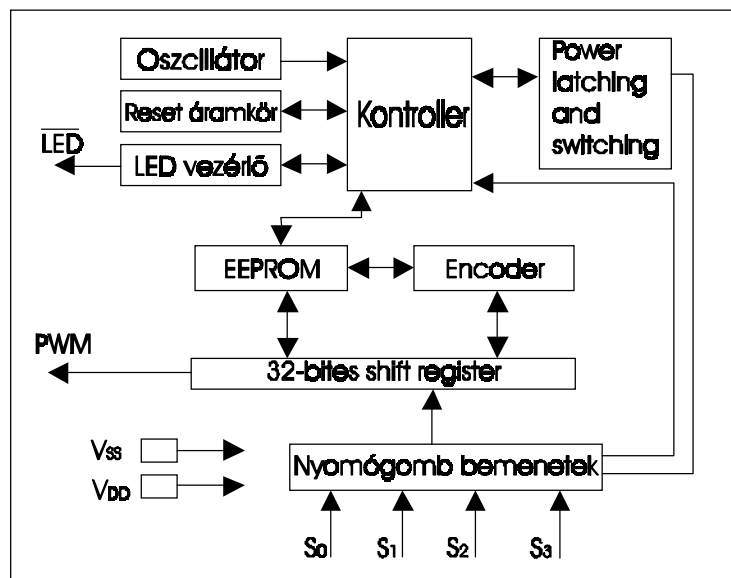


11. Ábra. Az üzenetküldés folyamata

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 kódolóknak 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 kódolóknak 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

²Felhasználó 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

Kódoló	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	Túlsordulást 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 feszültség szint 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	400 μ s	Mind
1	200 μ s	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 akkumulátor, 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 mutassa. 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	Túlsordulás bit 0 (OVR0)
11	Túlsordulás 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

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

Túlsordulás 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 túlsordulást 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	400 μ s	Mind
0	1	200 μ s	2-ből 1
1	0	100 μ s	2-ből 1
1	1	100 μ s	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

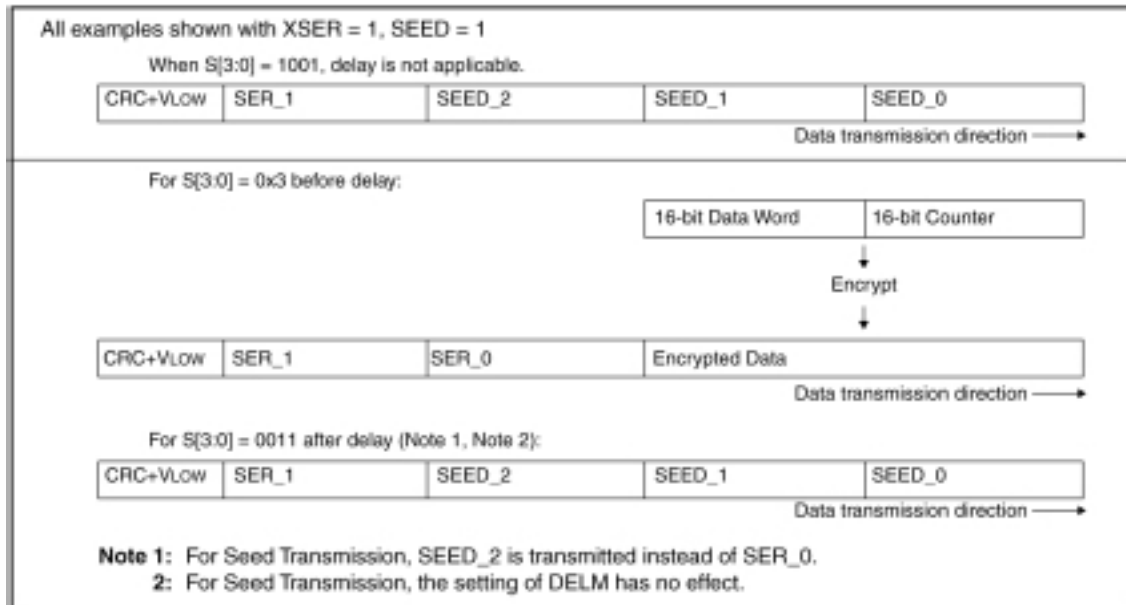
SEED: seed átvitel engedélyezése

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 kompatibilis 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: túlsordulás 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	Túlsordulás 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.

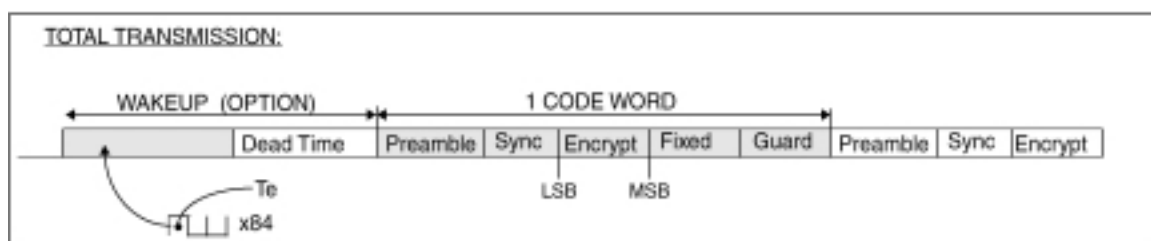
FAST: Gyors átvitel választó

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

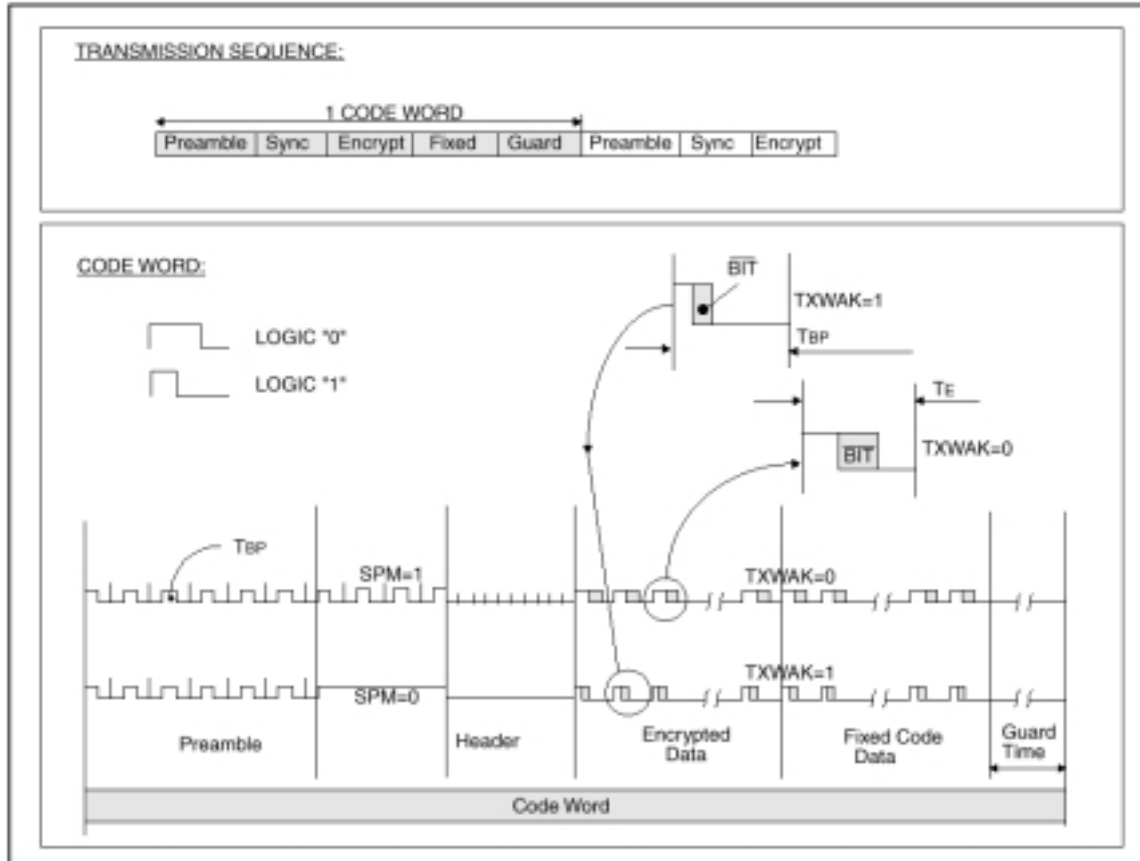
TXWAK: bit formátum választó vagy Wakeup

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 kompatibilis 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.

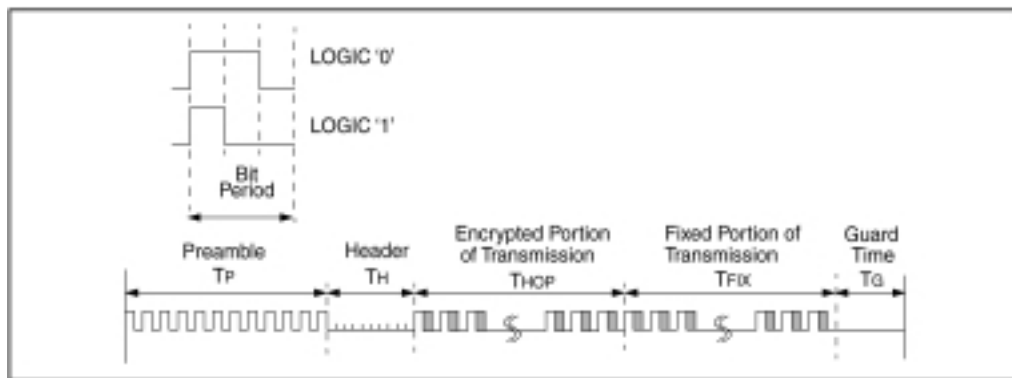
Ha a VPWM=0, akkor a PWM modulációt használja a kódoló.

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.

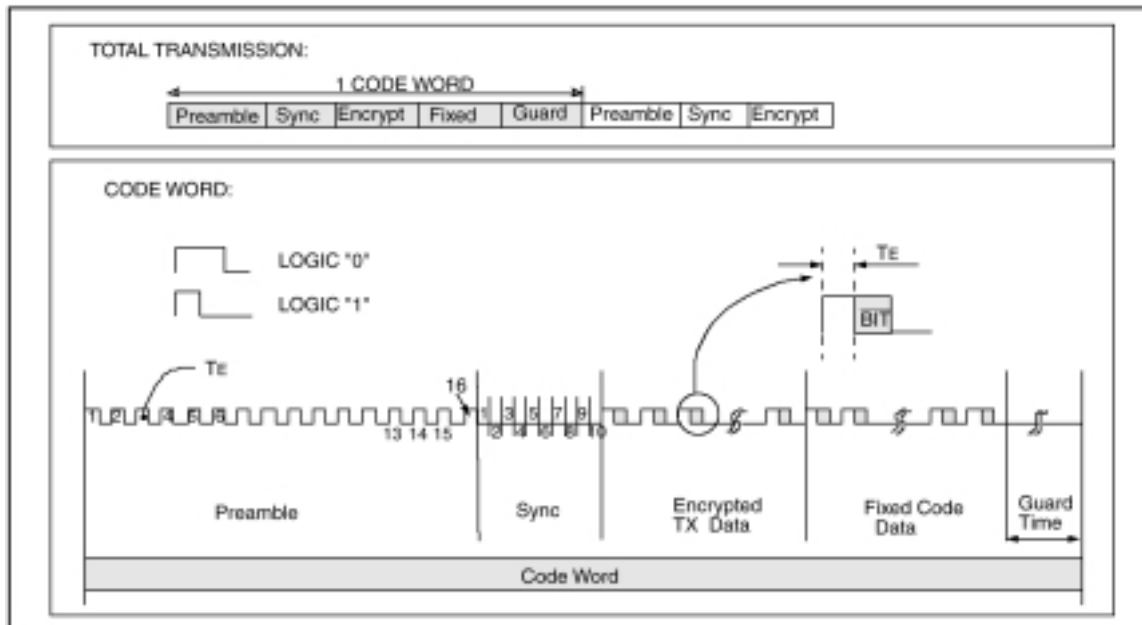


16. Ábra. Az átvitelek összetétele

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

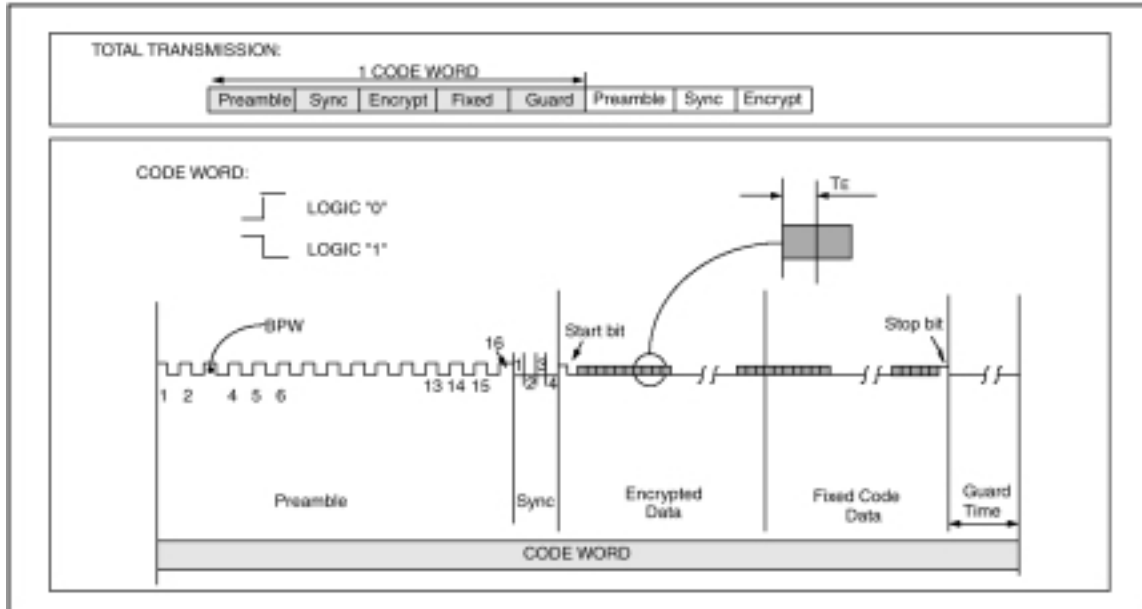


17. Ábra. A PWM modulálás

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.



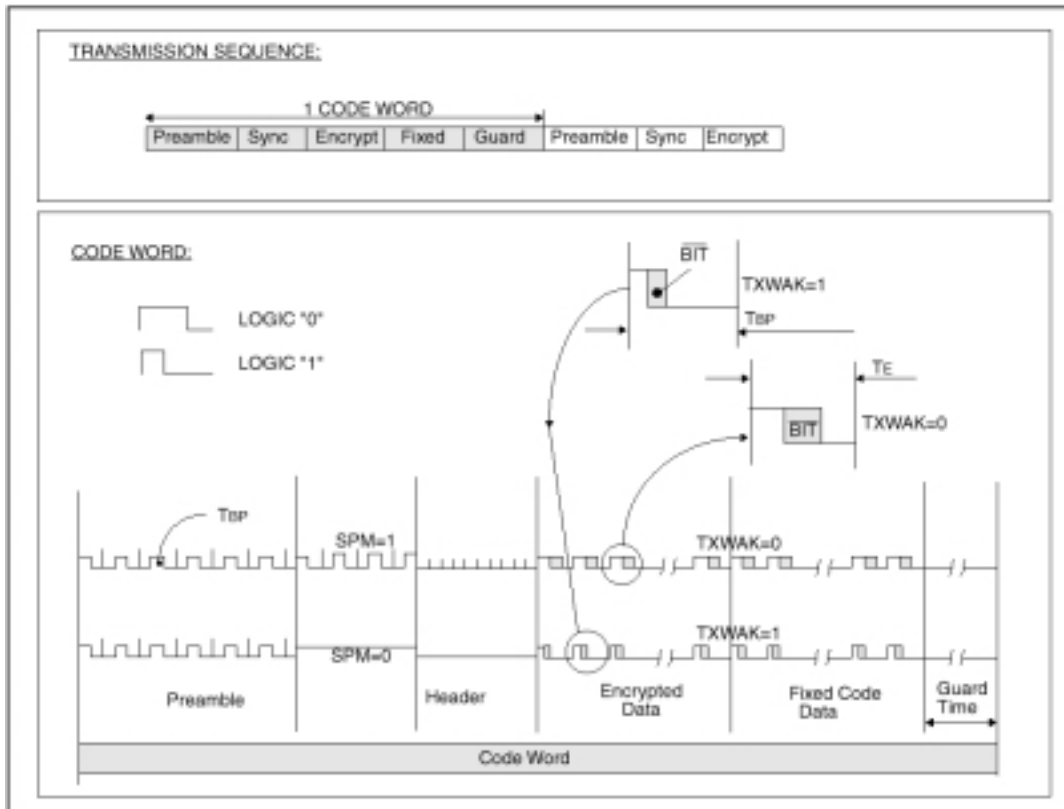
18. Ábra. Manchester modulálás

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

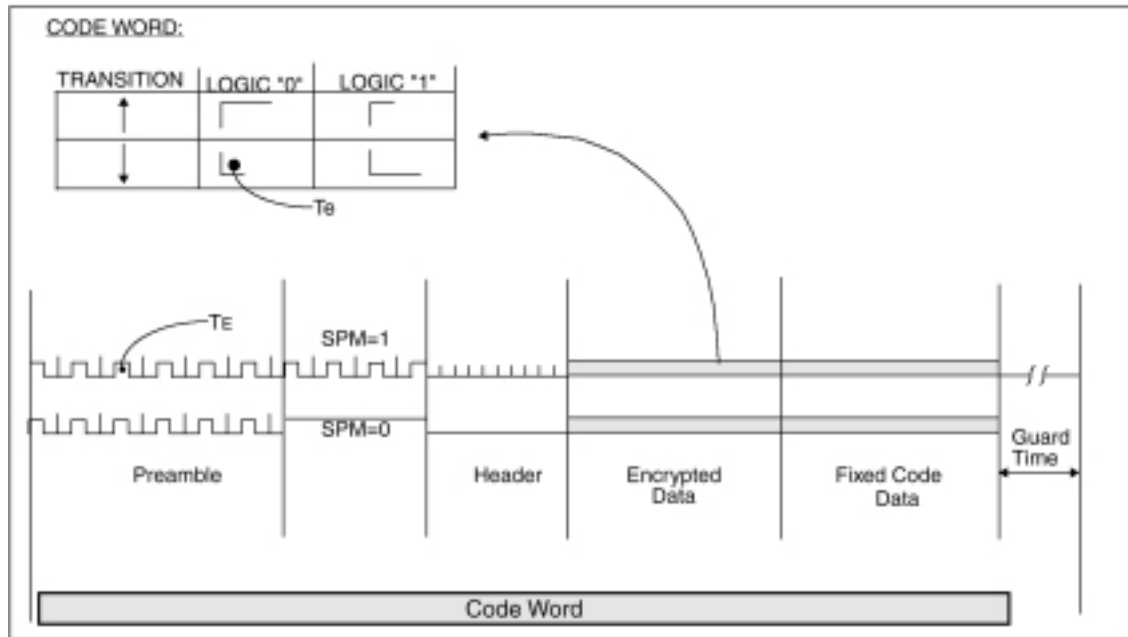
Tulajdonképpen ugyanaz, mint a többi kódolónál azonban azért kell kiemelni, mert a HCS361-nek van egy olyan konfigurációs bitje, amelyikkel a PWM kitöltési tényezőjét változtathatjuk.



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 kódolónál, vagy itt be lehet állítani, hogy $2/6$ - $1/6$ -os kitöltési tényezőjű impulzus szélesség modulált jel hordozza az információt.

VPWM modulálás



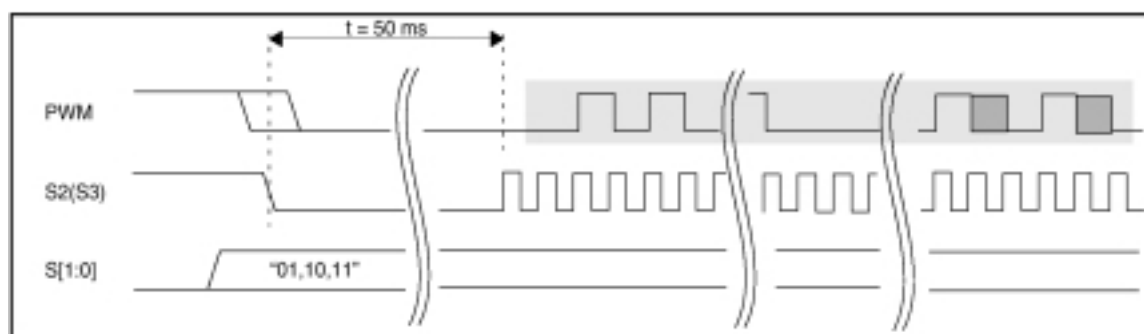
20. Ábra. A VPWM modulálás

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 követi 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)

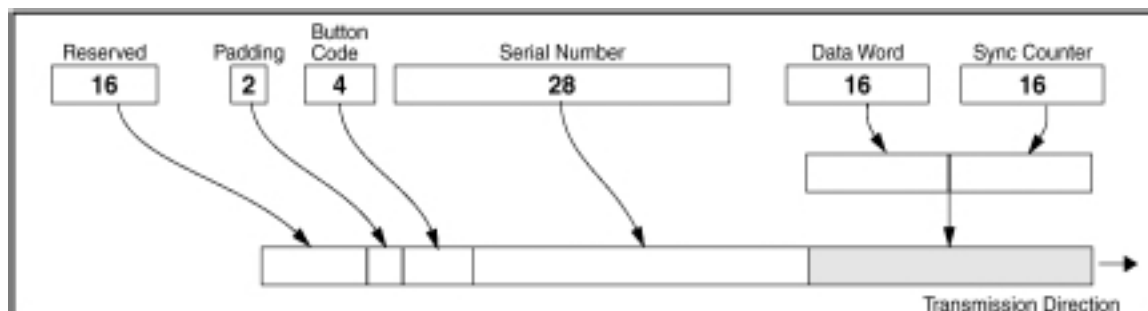
A szinkron átviteli mód egy külső órajel használatával történik. Mindegyik kódoló esetében tekinthetjük egyszerre, mert tulajdonképpen ugyanúgy zajlik le a folyamat. Van egy kis különbség köztük, de erre az áttekintés közben fel fogom hívni a figyelmet.

Azt, hogy hogyan lehet feléleszteni 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 nagyobbak lennie. A kód szó ugyanaz, mint a PWM módban a 16 pótbittel a szó végén, mely pótbitok 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.

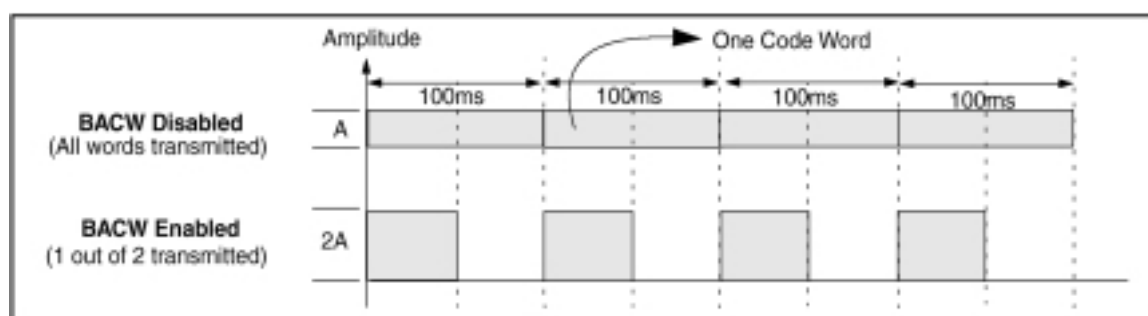
9.6.2 Blank Alternate kód szó (BACW)

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 amellet, 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

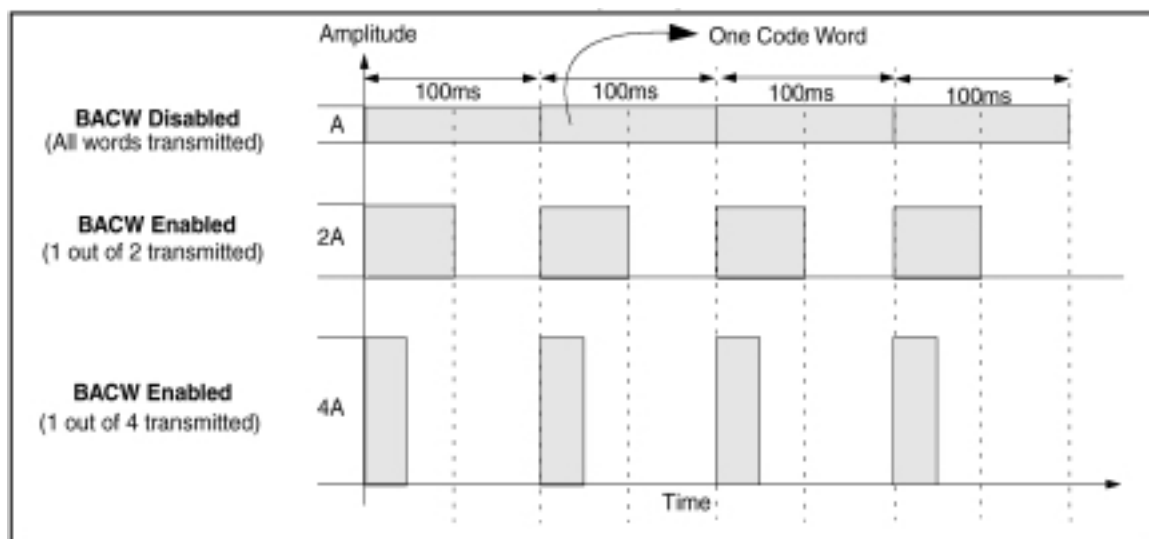
Ezeknél a kódoló típusoknál megegyezik a BACW választási lehetőség (23. ábra).



23. Ábra. BACW választási lehetőségei

9.6.2.2 HCS300/301 kódoló

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

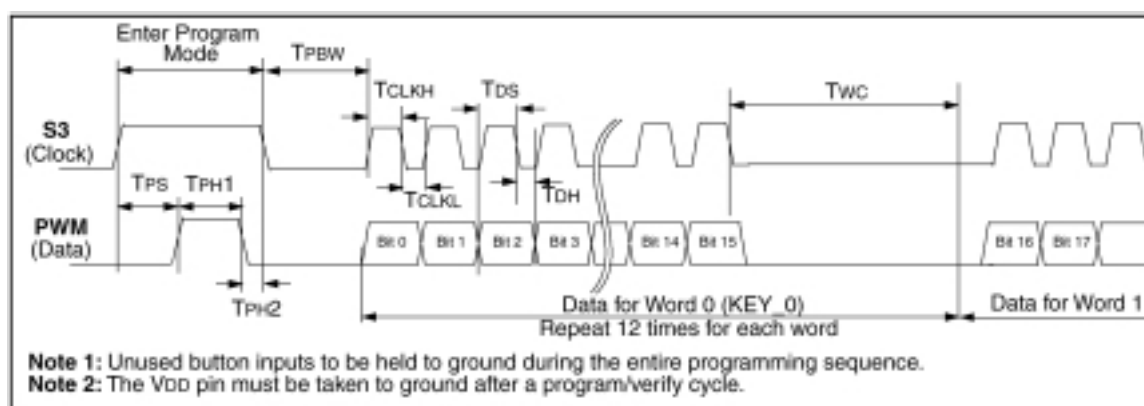


24. Ábra. BACW választási lehetőségei

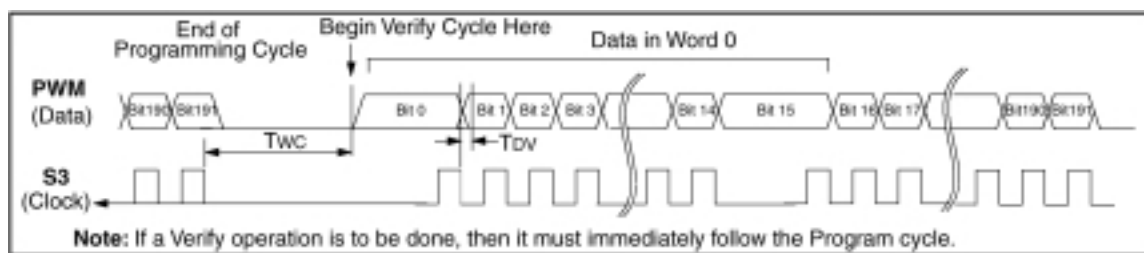
9.7 PROGRAMOZÁS

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 lábón 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

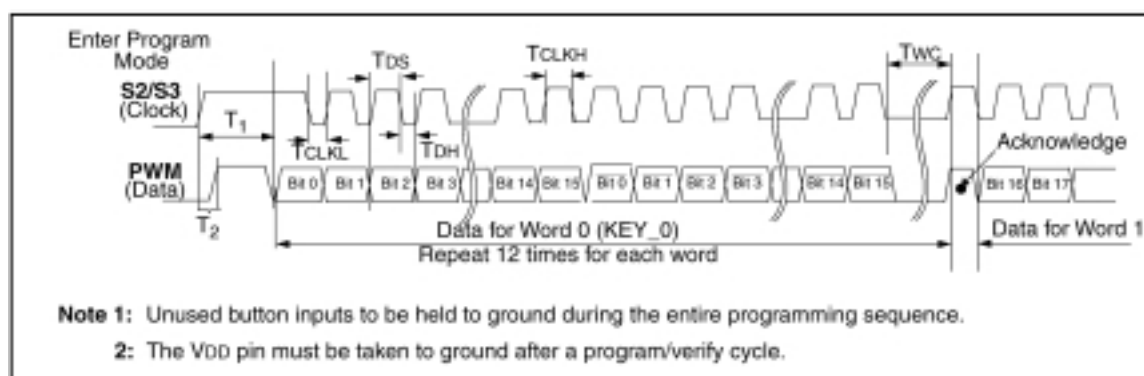
Paraméterek	Jelölések	Min.	Max.	Egység
Program mode setup time	T _{PS}	3.5	4.5	ms
Hold time 1	T _{PH1}	3.5	-	ms
Hold time 2	T _{PH2}	50	-	μs
Bulk Write time	T _{PBW}	-	3.5	ms
Program delay time	T _{PROG}	-	3.5	ms
Program cycle time	T _{WC}	-	36	ms
Clock low time	T _{CLKL}	25	-	μs
Clock high time	T _{CLKH}	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