TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Tietokonetekniikan suuntautumisvaihtoehto

Miika Kaatrasalo

GSM-RELE

Työn valvoja: Yliopettaja Mauri Inha
Työn ohjaaja: Yliopettaja Mauri Inha

Tekijä:	Kaatrasalo Miika
Työn nimi:	GSM-rele
Päivämäärä:	15.5.2007
Sivumäärä:	26 sivua ja 17 liitesivua
Hakusanat:	PIC16F870, mikrokontrolleri, DTMF
Koulutusohjelma:	Tietotekniikka
Suuntautumisvaihtoehto:	Tietokonetekniikka
Työn valvoja:	Yliopettaja Mauri Inha
Työn ohjaaja:	Yliopettaja Mauri Inha Tampereen ammattikorkeakoulu, Tampere

Tässä tutkintotyössä suunniteltiin ja rakennettiin matkapuhelimeen kytkettävä laite, GSM-rele, joka mahdollistaa siihen kytkettyjen sähkölaitteiden etäkäyttämisen. Tämä sisältää sekä yksinkertaisia ohjaustoimenpiteitä että tiedonkulun myös takaisinpäin, kuten vaikka takaisinkytkentätietojen saamisen ohjattavista laitteista. Releessä on neljä transistorilähtöä, joita voidaan käyttää laitteen/laitteiden ohjaamiseen. Lisäksi releessä on kaksi hälytystuloa, jotka aiheuttavat hälytyksen käyttäjän valitseman viiveen jälkeen. GSM-rele osaa ilmoittaa käyttäjälle sekä ohjattavien lähtöjen että hälytystulojen/hälytysten tilat. Releen käyttäminen voidaan estää salasanalla. Toimintojen käyttäminen tapahtuu soittamalla GSM-releeseen kytkettyyn matkapuhelimeen ja lähettämällä DTMF-koodattuja merkkejä.

Tämänkaltainen laite voidaan toteuttaa hyvin monella eri tavalla. Tämän työn laite on tehty PIC16F870-mikrokontrollerin ympärille. Mikrokontrolleri on liitetty DTMF-vastaanottimeen, joka vastaanottaa ja dekoodaa linjalla esiintyvät DTMF-merkit.

Tässä työssä kerrotaan laitteen ominaisuuksista ja käydään läpi sekä laitteen sähköinen rakenne että mikrokontrollerille tehty ohjelma. Lisäksi työssä selvitetään mikrokontrollerin toimintaa ja rakennetta ja DTMF-koodauksen teoriaa siinä määrin, kuin on tarpeellista.

Author:	Kaatrasalo Miika
Title:	GSM-relay
Date:	15.5.2007
Number of pages:	26 pages and 17 appendices
Key words:	PIC16F870, microcontroller, DTMF
Program:	Information technology
Specialisation:	Computer engineering
Supervisor:	Senior Lecture Mauri Inha
Instructor:	Senior Lecture Mauri Inha Tampere Polytechnic University, Tampere

In this work, there was designed and built a device, called GSM-relay, which is connected to a mobile phone and gives opportunity to do simple control operations to other electrical devices, and get some information e.g. feedback from these controlled devices. The relay has four transistor outputs, which can be used to control these devices. It has also two inputs, which makes alert after the desired delay. The GSM-relay is able to announce the conditions of the outputs and the inputs to the user. The use of the functions can be prevented by a password. Operations are done by calling to the mobile phone, which is connected to the device, and sending DTMF-coded characters.

There is many ways to implement this kind of device. This device is built around the PIC16F870 microcontroller. The microcontroller is connected with a DTMF receiver, which receives and decodes DTMF-codes on the line.

This thesis is about the program of the microcontroller and properties and electrical structure of the GSM-relay. There is also discussed some theory of DTMF-coding and characteristics of the PIC16F870.

ALKUSANAT

GSM-rele ei yleisesti ottaen ole mikään uusi keksintö. Olin kuullut aiemmin, että vastaavanlaisia laitteita on olemassa ja minulla oli jo pidemmän aikaa kiinnostusta aiheeseen. Niinpä ajattelin, että se olisi hyvä aihe myös tutkintotyöhöni.

GSM-releen suunnittelun aloitin alkusyksystä 2006. Ensimmäiset toimivat versiot olivat valmiina ennen saman vuoden joulukuuta. Viimeistelin työtä lähes sen palauttamispäivään saakka.

Tampereella 15. toukokuuta 2007

_____
Miika Kaatrasalo

SISÄLLYSLUETTELO

KÄYTETYT MERKINNÄT JA TERMIT

AD-muunnos	analogisen signaalin muuttaminen digitaaliseen muotoon
BS	Base Station, tässä yhteydessä, GSM-releeseen kytkettävä, etäkäyttöjärjestelmän ”tukiasema”
heksadesimaalijärjestelmä	Heksadesimaalijärjestelmän on lukujärjestelmä, jonka kantaluku on 16. Järjestelmässä käytetään tavallisesti numeroita 0-9 ja kirjaimia a-f.
dekoodaus	koodauksen purkaminen
EEPROM	Electrically Erasable Programmable Read-Only Memory, haihtumatonta muistia, johon voidaan kirjoittaa uudelleen
FLASH	haihtumatonta muistia, johon voidaan kirjoittaa uudelleen
GSM	Global System for Mobile Communications, joukko matkapuhelinviestintää koskevia standardeja
LED	Light-Emitting Diode, ledi, valoa synnyttävä puolijohdekomponentti
MS	Mobile Station, tässä tapauksessa etäkäyttöjärjestelmän liikkuva asema
PWM	Pulse-Width Modulation, pulssinleveysmodulaatio, käytetään mm. tehonsäädössä
SRAM	Static Random Access Memory, RAM, joka säilyttää datan ilman muistin virkistämistä

1 JOHDANTO

Työn tavoitteena oli suunnitella ja rakentaa GSM-rele, jonka avulla saadaan rakennettua GSM-verkon yli toimiva sähkölaitteiden etäkäyttöjärjestelmä.

Vaihtoehtoisia toteutustapoja oli useita, mutta lähtökohtina oli rakentaa laite PIC-mikrokontrollerin ympärille, säilyttää laitteeseen kytkettävä matkapuhelin koskemattomana, käyttää apuna DTMF-koodausta ja rakentaa laite, jolla voidaan toteuttaa useita eri toimintoja.

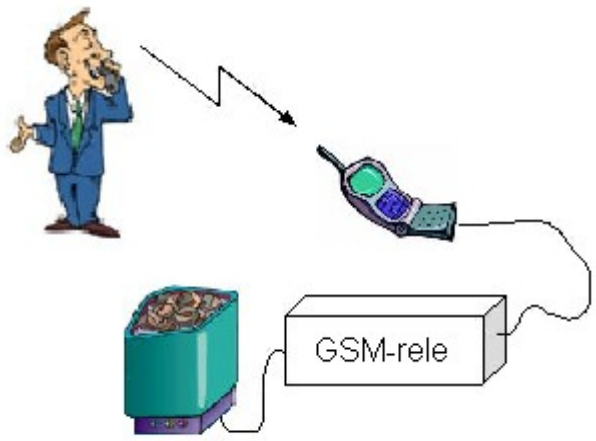
Suurin osa tästä työstä oli ohjelmiston suunnittelua. Ohjelmiston suunnittelu ja kehitys tapahtui ilmaisella, Microchipin valmistamalla MPLAB IDE -ohjelmistolla. Ohjelmointi tehtiin assembly-kielellä. Varsinaisessa elektroniikkasuunnittelussa ei käytetty apuvälineitä, mutta kytkennän piirikaavion piirtämiseen käytettiin ilmaista Eagle Layout Editor -ohjelmaa.

Mikrokontrollerin ohjelmointiin käytettiin Tampereen ammattikorkeakoulun Labtool-48 -ohjelmointilaitetta.

Tarvittavat komponentit hankittiin Elektorilta ja Bebekiltä.

2 ETÄKÄYTTÖJÄRJESTELMÄN RAKENNE JA TOIMINTAPERIAATE

Etäkäyttöjärjestelmä koostuu tutkintotyönä tehdystä GSM-releestä, siihen kytkettävästä matkapuhelimesta, BS, GSM-verkosta ja puhelimesta, MS, jolla käyttäjä antaa ohjauskäskyt GSM-releelle. Kuvassa 1 on esitetty järjestelmän rakenne.



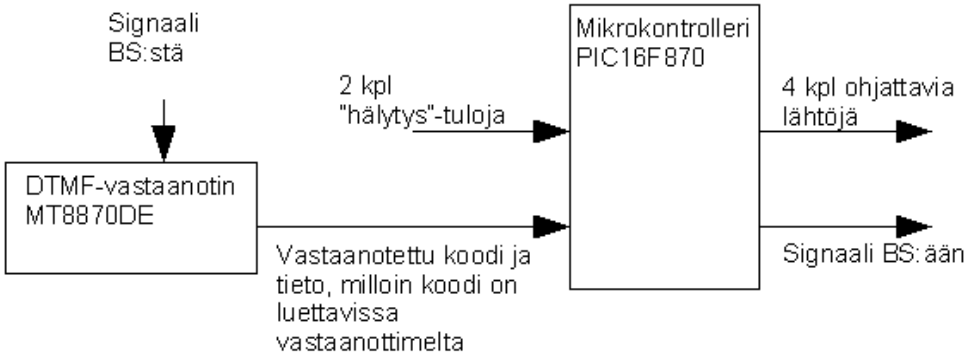
Kuva 1. Etäkäyttöjärjestelmän rakenne

Järjestelmän perusajatuksena on, että saadaan tehtyä vaivattomasti, lähes mistä tahansa, yksinkertaisia ohjaustoimenpiteitä esimerkiksi kodissa tai autossa oleviin laitteisiin. Lisäksi laitteen avulla saadaan tietoa siinä olevien lähtöjen ja tulojen tiloista. Etäkäyttölaitteeseen kytketyn laitteen ohjaaminen tapahtuu soittamalla BS:ään. BS:n asetuksista täytyy olla valittuna automaattinen vastaus.

Yhteyden avauduttua voidaan suorittaa halutut toiminnot lähettämällä DTMF-koodattuja merkkejä.

Käytännössä laitteella ei voi ohjata suoraan saunan kiuasta, kuten kuvassa 1, eikä muitakaan suuritehoisia laitteita. Tosin käyttämällä apureleitä sekin on mahdollista.

3 GSM-RELEEN RAKENNE JA KYTKENTÄ



Kuva 2. GSM-releen lohkokaavio

Itse rele koostuu joukosta elektoniikan peruskomponentteja ja kahdesta mikropiiristä: PIC16F870-mikrokontrollerista ja DTMF-vastaanottimesta. Kuvassa 2 on yksinkertaistettu, kaukokäyttölaitteen rakennetta selkeyttävä lohkokaavio. Kuvassa näkyvät tärkeimmät osat ja signaalit. Laite saa käyttöjännitteensä, 5,5 V:n tasajännitteen, verkkolaitteesta.

3.1 DTMF-vastaanotin

DTMF on koodaustapa, jota käytetään puhelinlaitteiden numeronvalinnassa. Tässä koodaustavassa jokainen numero tai merkki koodataan ääneksi, joka koostuu kahdesta eri äänestä, joiden taajuudet ovat toisiinsa nähden epäharmonisessa suhteessa. /1/

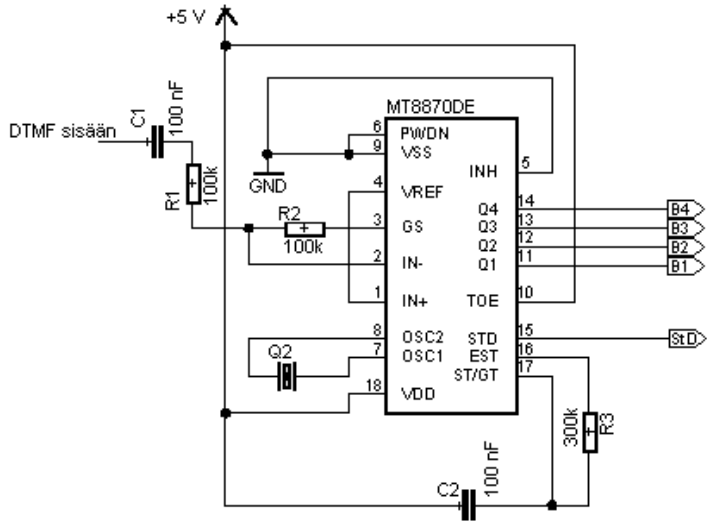
DTMF-vastaanottimen tehtävä on tunnistaa ja dekoodata signaalissa olevat DTMF-koodatut merkit ja ilmoittaa ne mikrokontrollerille. Vastaanottimessa on neljä kolmitilalähtöä, Q1...Q4, joilla se ilmoittaa vastaanotetun merkin binäärilukuna ja yksi lähtö, StD, jolla se ilmoittaa mikrokontrollerille, että merkki on luettavissa vastaanottimen lähdöissä Q1...Q4. Kun signaalissa esiintyy jokin DTMF-koodi, vastaanotin asettaa lähdöt Q1...Q4 vastaanotettua merkkiä vastaavaan tilaan taulukon 1 mukaisella tavalla. Kun lähdöt ovat asettuneet oikeisiin tiloihinsa, vastaanottimen StD-lähtö asettuu.

Taulukko 1. DTMF-merkkien dekoodaus

Vastaanotettu merkki	DTMF-vastaanottimen lähdöt Q4	Q3	Q2	Q1
0	1	0	1	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
*	1	0	1	1
#	1	1	0	0

3.1.1 Vastaanottimen kytkentä

Vastaanottimen kytkentä on kuvassa 3. Vastaanottimen kytkentä on melko yksinkertainen. Signaali, josta koodit etsitään, otetaan BS:n handsfree-liitynnän kaiutinlähdöstä. Signaalin kytkentään tarvitaan 100 nF:n kytkentäkondensaattori, C_1 ja vastukset R_1 ja R_2. Vastusten resistanssien suhde määrää jännitevahvistuksen, jolla sisään tulevaa signaalia vahvistetaan. Vahvistus on R2:n ja R1:n osamäärä. Tässä tapauksessa vahvistusta ei tarvita, joten valitsemalla samankokoiset vastukset, vahvistus on 1. Vastuksien resistansseiksi valittiin 100 kΩ.



Kuva 3. DTMF-vastaanottimen kytkentä

Lisäksi vastaanotin tarvitsee vastuksen R3, kondensaattorin C2 ja 3,579545 MHz:n kiteen, Q2. Kide kytketään suoraan OSC 1 ja OSC 2 -tuloihin. Vastuksen ja kondensaattorin mitoitus määrää vähimmäisajan, jonka DTMF-signaalin on oltava vastaanottimen tulossa. Nokia 6510 -puhelimen lähettämien DTMF-koodien kesto on noin 180 ms. Kun otetaan huomioon vastaanottimessa oleva 10 ms:n viive, ajaksi täytyy asettaa alle 170 ms. Komponenttien arvoiksi valittiin 100 nF ja 300 kΩ, joilla ajaksi tulee noin 20 ms.

Vastaanottimen lähdöt, Q1...Q4 ja StD, on kytketty suoraan, ilman lisäkomponentteja mikrokontrollerin tulonastoihin. Kolmitilalähtöjen korkeaimpedanssista tilaa ei tarvita, joten vastaanottimen TOE-nasta on kytketty käyttöjännitteeseen.

Vastaanotin pystyy vastaanottamaan myös merkit A, B, C ja D, mutta tässä työssä ne eivät ole käytössä. INH-nasta on kytketty nollaan volttiin, mikä estää näiden merkkien tunnistamisen.

Kun PWDN-signaali asetetaan, oskillaattori pysähtyy ja vastaanotin menee virransäästötilaan. PWDN-nasta on kytketty kiinteästi nollaan volttiin.

3.2 Mikrokotrolleri

Kaukokäyttölaitteen ”sydämeksi” on valittu Microchipin valmistama, 8-bittinen, PIC16F870-mikrokontrolleri, jossa on 22 nastaa, jotka voidaan kukin erikseen määritellä joko tuloksi tai lähdöksi. Koska jo pelkästään PIC16-piiriperheeseen kuuluu lähes 100 eri mikrokontrolleria, parhaan mahdollisen kontrollerin etsiminen tähän työhön olisi vaatinut kohtuuttoman suuren vaivan. Kontrollerin valintaan vaikutti lähinnä näiden nastojen riittävä määrä ja PWM-lähdön olemassaolo.

GSM-releessä kontrollerin tehtävänä on hoitaa toimintojen suorittaminen DTMF-vastaanottimelta saamiensa merkkien mukaan.

3.2.1 Osoitustavat

PIC16F870-mikrokontrollerin ohjelmoinnissa voidaan käyttää kahta osoitustapaa, joko suoraa tai epäsuoraa. Käytettäessä suoraa osoitustapaa käskyn parametriksi annetaan suoraan se rekisteri, jota käskyn halutaan käsittelevän. Alla on esimerkki TRISA-rekisterin nollaamisesta suoralla osoitustavalla.

```
clrf   TRISA
```

Epäsuorassa osoitustavassa käytetään kahta apurekisteriä: INDF ja FSR. FSR on rekisteri, joka toimii osoittimena. FSR-rekisteriin ladataan käsiteltävän rekisterin osoite. INDF on taas rekisteri, jota käytetään käskyjen parametrina, kun käsky osoitetaan FSR-rekisterin osoittamaan rekisteriin. Edellisen esimerkin operaatio voidaan toteuttaa myös epäsuoralla osoitustavalla seuraavan esimerkkikoodin mukaan.

```
movlw  0x85   ; ladataan akkuun luku 0x85, joka on TRISA-rekisterin
                osoite

movwf  FSR    ; akun sisältö kopioidaan osoitinrekisteriin

clrf   INDF   ; osoittimen osoittama rekisteri nollataan
```

Tässä työssä käytetään epäsuoraa osoitustapaa vastaanotettujen merkkien käsittelyyn.

3.2.2 Muistit ja rekisterit

PIC16F870-mikrokontrollerissa on kolmiosainen muisti, joka sisältää FLASH- ja data-muistit ja EEPROM:in. FLASH-muisti on kooltaan 2048 sanaa ja yksi sana sisältää 14 bittiä. Suoritettava ohjelma kirjoitetaan FLASH-muistiin. EEPROM on 64-tavuinen. FLASH-muistin ja EEPROM:in sisältöön päästään käsiksi vain epäsuorasti, tiettyjen rekistereiden avulla.

SRAM-teknologialla toteutettu datamuisti on jaettu neljään osaan, joista kukin sisältää 128 8-bittistä rekisteriä. Nämä osat eli pankit on numeroitu nollasta kolmeen. Rekistereiden sisältöä voidaan käsitellä joko suoralla tai epäsuoralla osoitustavalla. Osa rekistereistä on erikoisrekistereitä, joilla on tietty käyttötarkoitus ja osa rekistereistä on yleiskäyttöisiä, joita voidaan käyttää haluttuihin tarkoituksiin ohjelmakoodissa.

3.2.2.1 STATUS-rekisteri

STATUS-rekisteri sisältää lippubittejä ja bitit, joilla valitaan käytössä oleva pankki. Kun käytetään suoraa osoitustapaa, käytössä oleva pankki valitaan RP1- ja RP0-biteillä (bitit 6 ja 5). IRP-bitillä (bitti 7) taas valitaan käytössä oleva pankki, kun käytetään epäsuoraa osoitustapaa. Kun ohjelmassa käsitellään jotain tiettyä rekisteriä, täytyy aina varmistua, että nämä bitit ovat oikeissa tiloissa. STATUS-rekisteri on käsiteltävissä aina, olivatpa nämä bitit missä tiloissa tahansa.

STATUS-rekisterin lippubiteistä tässä työssä käytetään eniten Z-lippua eli nollalippua. Z-lippu asettuu, jos tehdyn aritmeettisen tai loogisen operaation tulos on nolla. Lisäksi yhdessä kohtaa käytetään C- eli Carry-lippua. C-lipun toiminnasta kerrotaan lisää myöhemmin siltä osin, kuin sitä tässä työssä käytetään.

3.2.2.2 W-rekisteri

W-rekisteri eli akku on erillinen, 8-bittinen työrekisteri, joka ei sisälly edellä mainittuun datamuistiin. Suuri osa mikrokontrollerin käskykannan käskyistä kohdistuu akkuun. Esimerkiksi aina, kun käsky kohdistuu kahteen rekisteriin, akku on niistä toinen. Toisin sanoen jonkin rekisterin sisällön kopioimista toiseen rekisteriin ei voi tehdä suoraan yhdellä käskyllä, vaan akkua on käytettävä tiedon välitallennuspaikkana.

3.2.3 Keskeytykset

Keskeytykset ovat yleensä olennainen osa mikrokontrollerisovellusta. Keskeytyksiä voidaan hyödyntää mm. tehtävien priorisointiin ja ohjelman ajoitukseen liittyvissä asioissa. PIC16F870-mikrokontrollerilla on 14 mahdollista keskeytyksen aiheuttajaa. Näitä ovat esimerkiksi ajastimet ja laskurit, AD-muunnin ja tietyt nastat. Ohjelmassa voidaan sallia tai estää eli maskata yksilöllisesti eri aiheuttajien aiheuttamat keskeytykset. Kullakin keskeytyksen aiheuttajalla on lippubitti, joka asettuu, kun aiheuttaja on aiheuttanut keskeytyksen. Heti, kun jonkin sallitun aiheuttajan lippubitti asettuu, siirrytään keskeytyspalveluohjelmaan.

Keskeytyksen aiheuttaja saadaan selville tutkimalla näitä lippubittejä keskeytyspalveluohjelmassa ja tehtävät operaatiot voidaan valita aiheuttajan mukaan.

Keskeytyksiä hallitaan INTCON-, PIE1-, PIR1-, PIE2- ja PIR2-rekistereillä. PIE1- ja PIE2-rekistereitä käytetään maskaukseen. PIR1- ja PIR2-rekisterit taas sisältävät lippubittejä. INTCON-rekisterissä on sekä maskaus- että lippubittejä. Tämä rekisteri sisältää mm. GIE-bitin, jolla voidaan estää kaikki keskeytykset.

3.2.4 Portit

Kontrollerin tulo- ja lähtönastat on ryhmitelty portteihin porta, portb ja portc. Jokaisen portin jokainen nasta on kaksisuuntainen ja se voidaan erikseen määritellä joko tuloksi tai lähdöksi. Tämän lisäksi nastalla voi olla joitain lisätoimintoja, vaikkapa esimerkiksi keskeytyksen aiheuttamiseksi tai AD-muunnoksen tekemiseksi. Kaikki tässä sovelluksessa käytetyt tulot ja lähdöt ovat digitaalisia.

Porttien nastojen suunnat valitaan TRISA-, TRISB- ja TRISC-rekistereillä. Jokaisella nastalla on sitä vastaava bitti jossain näistä rekistereistä. Esimerkiksi portb:n RB4-nastan bitti on TRISB-rekisterin bitti 4. Kun bitti on asetettu, nasta toimii tulona, ja kun bitti on nollattu, nasta toimii lähtönä.

PORTA-, PORTB- ja PORTC-rekistereihin kirjoittamalla voidaan muuttaa sitä vastaavan portin lähtöjen tiloja. Vastaavasti näitä rekistereitä lukemalla saadaan selville kyseisen portin nastojen tilat.

3.2.5 Ajastimet

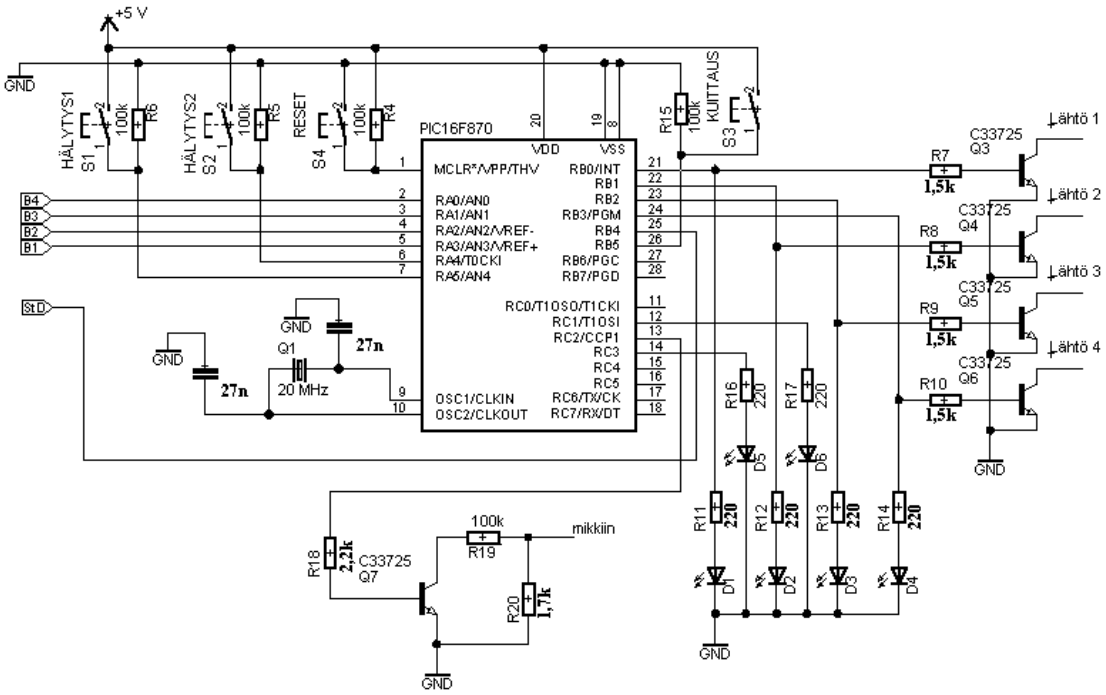
PIC16F870-kotrollerissa on kolme ajastinta, joista tässä työssä käytetään kahta: TIMER1:tä ja TIMER2:ta. TIMER1-ajastinta voidaan käyttää myös laskurina. Näiden ajastimien arvo kasvaa yhdellä joka neljännellä kontrollerin käyttämän kiteen kellopulssilla. Tätä neljän kellopulssin jaksoa kutsutaan myös käskyjaksoksi. Tässä työssä on käytössä 20 MHz:n kide, jolloin käskyjakson pituudeksi tulee 200 ns. Molemmilla ajastimilla on myös esijakaja, jolla voidaan kasvattaa ajastimen arvon kasvamiseen tarvittavaa käskyjaksojen määrää. Esimerkiksi esijakajan arvolla kaksi arvo kasvaa joka toisella käskyjaksolla eli 400 ns:n välein. Ajastimen lukualueen täyttymisen jälkeen tapahtuu ylivuoto ja sen arvo nollaantuu. Ajastimista ja niiden käytöstä tässä työssä kerrotaan lisää myöhemmissä kappaleissa.

3.2.6 PWM-lähtö

Portc:n nastaa 2 on mahdollista käyttää PWM-lähtönä. Mikrokontrolleri voidaan siis määrätä generoimaan tähän nastaan jaksollista signaalia, jonka jaksonaika ja pulssisuhde ovat valittavissa. Signaalin jaksonaika määritellään PR2-rekisterillä ja pulssisuhde CCPR1L-rekisterillä ja kahdella CCP1CON-rekisterin bitillä. PWM-lähtö ajastetaan aina TIMER2-ajastimen mukaan. Näin ollen myös TIMER2:n esijakajan arvo ja käytössä olevan kiteen taajuus vaikuttavat generoitavan signaalin ominaisuuksiin. Tässä työssä PWM-lähtöä käytetään kahden eri taajuisen signaalin generoimiseen.

3.2.7 Mikrokontrollerin kytkentä

Kuvassa 4 on mikrokontrollerin kytkentä. Kide on kytketty kontrollerin OSC1- ja OSC2-nastoihin. Molemmat nastat on kytketty 27 nF:n kondensaattorien kautta nollaan volttiin. Sopivat kondensaattorit valittiin kontrollerin datalehden taulukosta.



Kuva 4. Mikrokontrollerin kytkentä

Hälytystulot ovat nastoissa RA5 ja RA4. Nastat on vedetty alas eli kytketty nollaan volttiin 100 kΩ:n vastuksilla, R5 ja R6. Hälytysten testausta varten kytkentään on lisätty kaksi painiketta, S1 ja S2, joilla hälytystulot saadaan asetettua eli kytkettyä viiteen volttiin.

Hälytysten kuittauspainikkeen kytkentä on täysin vastaava kuin hälytystulojenkin. Kuittauspainike S3 on kytketty nastaan RB5. Resetpainike S4 on kytketty nolla-aktiivisen MCLR-nastan ja nollan voltin välille, ja ylösvetovastus on kytketty viiden voltin ja MCLR-nastan välille.

Ohjattavat lähdöt on kytketty nastoihin RB0...RB3. Jokainen lähtö sisältää merkkivalon, joka ilmaisee lähdön tilan ja transistorilähdön ohjattavan laitteen kytkemistä varten. Lähdön merkkivalo on ledi, joka on kytketty 220 Ω:n etuvastuksen kautta lähtönastaan. Transistorilähdöt ovat avokollektorilähtöjä. Transistorien kannat on kytketty 1,5 kΩ:n vastuksien kautta lähtönastoihin. Jokainen lähtö voi ohjata enintään noin 100 mA:n virtaa. Transistoreilla ohjattava jännite saa olla enintään 45 V.

Kontrollerin PWM-lähtö eli RC2-nasta on kytketty 2,2 kΩ:n vastuksen kautta transistorin kannalle. Transistori toimii kytkimenä, joka kytkee 100 kΩ:n vastuksen

R19, BS:n handsfree-liitännän mikrofonituloon. Mikrofonitulon ja maan välille on kytketty myös 1,7 kΩ:n vastus R20. Tämä kytkentä näkyy BS:lle 1,7 kΩ:n tai noin 1,67 kΩ:n resistanssina RC2-nastan tilan mukaisesti. Tällä kytkennällä generoidaan ääniä, joilla välitetään tietoa etäkäyttäjälle. Äänien käytöstä kerrotaan lisää kappaleessa 4.3.

RC1- ja RC3-nastoihin on etuvastusten R16 ja R17 kautta kytketty ledit, jotka ilmoittavat hälytyksistä.

4 LAITTEEN TOIMINNOT JA OMINAISUUDET

Tässä luvussa kerrotaan kaukokäyttölaitteen ominaisuuksista ja toiminnoista ja niiden käytöstä.

4.1 Lähtöjen ohjaus

GSM-releen neljää transistorilähtöä, voidaan ohjata neljällä tavalla. Asetustoiminnolla voidaan asettaa valittu lähtö. Nollaustoiminto taas nollaa halutun lähdön. Vaihtotoiminto vaihtaa halutun lähdön tilan eli joko asettaa tai nollaa halutun lähdön. Ja viimeinen toiminto on pulssitoiminto, jolla voidaan tehdä valittuun lähtöön pulssi, jonka pituudeksi voidaan asettaa 1...100 sekuntia. Pulssitoiminto vaihtaa valitun lähdön tilan, odottaa asetetun ajan verran ja muuttaa lähdön tilan uudestaan. Jos lähdölle asetetaan pulssitoiminto, aiemmin asetetun pulssin ollessa kesken, aiemmin aloitettu pulssi unohdetaan ja uusi pulssi suoritetaan.

4.2 Hälytystulot

GSM-releessä on kaksi hälytystuloa. Hälytys syntyy, kun tulo on yhtäjaksoisesti asetetun viiveen verran asetettuna. Viiveeksi voidaan asettaa 1...100 sekuntia. Tulojen tilaa tarkastellaan noin 50 ms:n välein, joten lyhyitä nollassa käyntejä ei välttämättä havaita. Viiveen muuttamismahdollisuuden ansiosta voidaan vähentää väärien hälytysten syntymistä. Liian lyhyt viive voi aiheuttaa hälytyksen turhan herkästi, ja viiveen ollessa liian pitkä laite ei välttämättä reagoi silloin kun pitäisi.

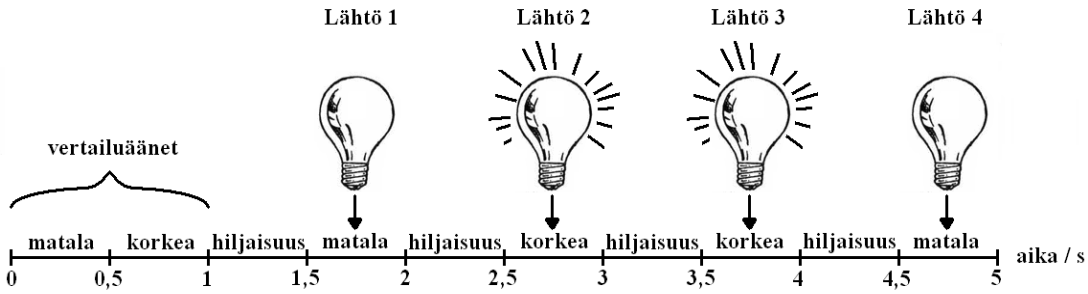
Kun hälytykset kuitataan, hälytysten merkkivalot sammuvat ja molemmat hälytykset kuittaantuvat.

4.3 Lähtöjen ja hälytysten tilojen ilmoitus

Releeltä voidaan kysyä sekä lähtöjen että hälytystulojen tilat. Laite ilmoittaa tilat kaksitaajuuksisella äänikoodilla. Taajuudet on valittu sen mukaan, miten hyvin ne voidaan erottaa korvakuulolla. Tietysti taajuuksien täytyi myös olla sellaisia, että GSM-järjestelmä pystyy välittämään ne. Taajuuksiksi valittiin 1,22 kHz ja 2,44 kHz. Sekä lähtöjen että tulojen tilojen ilmoittamisessa laite antaa aluksi vertailuääniksi puolen sekunnin matalan, 1,22 kHz:n, ja sitten puolen sekunnin korkean, 2,44 kHz:n, äänen.

4.3.1 Lähtöjen tilojen ilmoitus

Ilmoitettaessa lähtöjen tiloja matala ääni tarkoittaa sitä, että lähtö on nollattuna, ja korkea ääni tarkoittaa, että lähtö on asetettuna. Vertailuäänten jälkeen rele ilmoittaa lähtöjen tilat numerojärjestyksessä alkaen lähdöstä 1. Kunkin lähdön tilaa ilmoitettaessa on ensin puolen sekunnin hiljaisuus, ja sitten tulee joko matala tai korkea ääni lähdön tilan mukaan. Kuva 5 helpottaa lähtöjen ilmoitustavan ymmärtämistä. Palava lamppu kuvaa asetettua lähtöä ja sammunut lamppu nollattua lähtöä.



Kuva 5. Lähtöjen ilmoitus

4.3.2 Hälytystulojen tilojen ilmoitus

Tulojen ilmoittaminen tapahtuu samaan tyyliin kuin lähtöjenkin ilmoitus. Vertailuäänten jälkeen laite ilmoittaa, onko hälytystulo 1 aiheuttanut hälytyksen. Korkea ääni tarkoittaa, että hälytys on tapahtunut ja matala, ettei hälytystä ole tapahtunut. Seuraavaksi ilmoitetaan ensimmäisen hälytystulon senhetkinen tila. Korkea ääni tarkoittaa, että tulo on asettunut ja matala, että tulo on nollattu. Sen jälkeen ilmoitetaan toista hälytystuloa koskevat tiedot vastaavalla tavalla.

4.4 Lukitus

GSM-releen toiminnot voidaan lukita 3-merkkisellä salasanalla. Kun lukitus asetetaan päälle, laite ei suorita lukitsemisen jälkeen vastaanotettuja ohjauskoodeja. Muuten laitteen toiminta säilyy normaalina. Salasana voidaan muuttaa syöttämällä ensin '#'-merkki ja sen perään uusi salasana. Salasana muuttuu silloin, kun uusi salasana on syötetty kaksi kertaa edellä mainitulla tavalla. Kun toimintojen käyttö on estetty, MS:lle annetaan sekunnin välein sekunnin mittainen merkkiääni, jonka taajuus on 1,22 kHz.

5 TOIMINTOJEN KÄYTTÄMINEN

Toimintojen käyttäminen tapahtuu 1-, 2- tai 4-merkkisillä ohjauskoodeilla. Ensimmäisen merkin syöttämisen jälkeen käyttäjällä on viisi sekuntia aikaa syöttää koodin loppuosa. Jos merkkien syöttäminen kestää yli viisi sekuntia, syötetyt merkit unohdetaan ja ohjauskoodin vastaanottaminen aloitetaan alusta.

Eri toiminnot vaativat erimittaiset ohjauskoodit. Esimerkiksi hälytysten kuittaukseen riittää yksi merkki, kun taas pulssitoimintoon vaaditaan neljä merkkiä. Koodin ensimmäisellä merkillä valitaan toteutettava toiminto. Lukuun ottamatta salasanan syottämistä toisella merkillä osoitetaan, mihin lähtöön tai hälytystuloon toiminto kohdistuu. Pulssitoiminnossa ja hälytysaikojen asettamisessa kolmatta ja neljättä merkkiä käytetään ajan valitsemiseen. Salasanaa käsittelevissä toiminnoissa toinen, kolmas ja neljäs merkki vastaavat salasanan vastaavia merkkejä. Taulukko 2 esittää toimintoja vastaavat ohjauskoodit.

Taulukko 2. Toimintojen ohjauskoodit

Toiminto	1. merkki	2. merkki	3. merkki	4. merkki
Lähdön nollaus	0	ohjattavan lähdön numero, merkit 1...4	X	X
Lähdön asetus	1	ohjattavan lähdön numero, merkit 1...4	X	X
Hälytysten tilojen ilmoitus	2	X	X	X
Hälytyksen ajan asetus	4	5 -> hälytystulo 1 6 -> hälytystulo 2	Haluttu hälytyksen viive sekunteina, merkit 0...9	
Hälytysten kuittaus	5	X	X	X
Lähtöjen tilojen ilmoitus	6	X	X	X
Pulssitoiminto	7	ohjattavan lähdön numero, merkit 1...4	haluttu pulssin kesto sekunteina, merkit 0...9	
Lähdön tilan vaihtaminen	8	ohjattavan lähdön numero, merkit 1...4	X	X
Lukitus/avaus	*	Salasana, kaikki merkit		
Salasanan vaihto	#	Uusi salasana, kaikki merkit		

6 MIKROKONTROLLERIN OHJELMA

Tässä luvussa tarkastellaan mikrokontrollerin ohjelmaa. Ohjelma koostuu pääohjelmasta, muutamasta aliohjelmasta ja keskeytyspalveluohjelmasta.

6.1 Vakioaikavälikeskeytys

Ohjelman ajastukset on tehty vakioaikavälikeskeytyksen pohjalle. Kaikki aikaan sidotut toiminnot suoritetaan tämän keskeytyksen avulla. Keskeytys luodaan 50 ms:n välein TIMER1-ajastimen ylivuodon tapahtuessa. TIMER1 on 16-bittinen ajastin, joten sen lukualue on 0...65535. Näin ollen keskeytys tulisi 65536:n käskyjakson välein. Kun käskyjakson pituus on 200 ns, keskeytys tulisi noin 13 ms:n välein. 50 ms:n keskeytysvälin saamiseksi täytyy käyttää esijakajaa. Kun esijakajan arvoksi asetetaan neljä, ajastimen arvo kasvaa 800 ms:n välein ja keskeytysväliksi saadaan 52,4288 ms. Tämä väli on siis 2,4288 ms pidempi kuin haluttu keskeytysväli. Keskeytysväliä saadaan lyhennettyä, kasvattamalla ajastimen arvoa keskeytyksen tapahtumisen jälkeen. 2,4288 ms koostuu 3036:sta 800 ns:n jaksosta.

Näin ollen laskurin arvoon on lisättävä luku 3036 eli heksadesimaaliluku BDC tasan 50 ms:n keskeytysvälin saamiseksi.

Keskeytyspalveluohjelma on ohjelmakoodi, joka suoritetaan välittömästi keskeytyksen tapahtuessa. Edellä mainittu ajastimen arvon kasvatus tehdään keskeytyspalveluohjelmassa. Sen lisäksi keskeytyspalveluohjelmassa nollataan keskeytyksen automaattisesti asettama lippubitti ja asetetaan keskeytys-lippu, merkiksi pääohjelmalle siitä, että keskeytys on tapahtunut.

Alla on vakioaikavälikeskeytyksen ohjelmakoodi kommentteineen. Kommentit on erotettu varsinaisesta ohjelmakoodista puolipisteellä. Heksadesimaaliluvut merkitään 0x-etuliitteellä.

Koska keskeytyspalveluohjelmaan voidaan periaatteessa hypätä missä kohtaa tahansa ohjelman suoritusta, on keskeytyspalveluohjelman alussa yleensä hyvä ottaa ainakin akun ja STATUS-rekisterin sisällöt talteen ja palauttaa ne ennen sieltä poistumista. Näin keskeytyspalveluohjelmassa käynti ei aiheuta virhetilanteita ohjelman muussa toiminnassa. Tässä sovelluksessa keskeytyksen tapahtumakohta tiedetään, koska keskeytysten välillä suoritettavan ohjelman suorittaminen kestää perustilanteessa noin 15 µs, minkä jälkeen ei tehdä muuta, kuin odotetaan uutta keskeytystä. Kun ohjelmassa suoritetaan toimintoja tai vastaanotetaan merkkiä, ohjelman ajoaika on tietysti hieman pidempi, mutta se ei koskaan yllä lähellekään 50 ms:a. Kohdassa, jossa odotetaan uutta keskeytystä, ei akun eikä STATUS-rekisterin sisällöllä ole merkitystä, joten tässä tapauksessa tietojen väliaikaisen tallentamisen puuttuminen ei aiheuttaisi virhetilanteita. Seuraavilla käskyillä talletetaan STATUS-rekisterin ja akun sisällöt apumuuttujiin.

```
movwf       akku_temp          ; otetaan akun sisältö talteen
                               ; akku_temp -muuttujaan

movf        STATUS, 0          ; luetaan STATUS-rekisteri akkuun

movwf       status_temp        ; akun sisältö (STATUS) talteen
                               ; status_temp -muuttujaan
```

Seuraavaksi tehdään lippubittien päivitykset. Ensin nollataan keskeytyksen asettama lippubitti PIR1-rekisteristä. Sen jälkeen asetetaan keskeytys-lippu merkiksi pääohjelmalle, että keskeytys on tapahtunut.

```
bcf         PIR1, 0            ; keskeytyksen asettaman lipun
                               ; nollaus

bsf         keskeytys          ; keskeytys-lipun asettaminen
```

TIMER1-ajastimen arvoa kasvatetaan luvulla 0xBDC. Koska ajastin on 16-bittinen, sen arvo sijaitsee kahdessa rekisterissä. Arvon vähemmän merkitsevä osa on TMR1L-rekisterissä ja enemmän merkitsevä osa TMR1H-rekisterissä. Arvon kasvatus täytyy siis tehdä kahdessa osassa. Ensin TMR1L-rekisteriin lisätään luku 0xDC ja sitten TMR1H-rekisteriin lisätään luku 0xB.

```
movlw       0xDC            ; ladataan akkuun luku 0xDC

addwf       TMR1L,1         ; lisätään akussa oleva luku TIMER1:n
                            ; arvoon


movlw       0x0B            ; ladataan akkuun 0x0B

addwf       TMR1H,1         ; TIMER1:n (kokonais)arvoon lisätään
                            ; luku 0x0B00
```

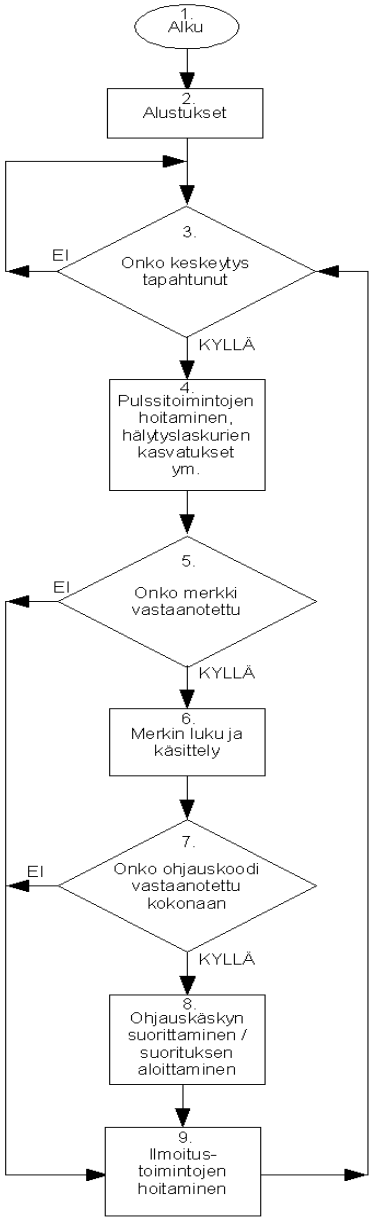
Lopuksi palautetaan talteen otetut STATUS-rekisterin ja akun arvot ja poistutaan keskeytyspalveluohjelmasta. Akun arvon palauttamisessa käytetään swapf-käskyä, koska se ei muuta STATUS-rekisterin lippubittien arvoja. Swapf-käsky vaihtaa parametrinä olevan rekisterin neljän eniten merkitsevien ja neljän vähiten merkitsevien bittien paikat. Tulos voidaan tallentaa joko alkuperäiseen rekisteriin tai akkuun. Käsky joudutaan siis suorittamaan kaksi kertaa, joista jälkimmäisellä tulos talletetaan akkuun..

```
movf   status_temp, 0
; vanha STATUS akkuun

movwf  STATUS
; STATUS-rekisterin palautus

swapf  akku_temp, 1
; bittien vaihto

swapf  akku_temp, 0
; bittien vaihto ja akun vanhan
; sisällön palauttaminen akkuun

retfie
; paluu keskeytyspalveluohjelmasta
```

6.2 Pääohjelma

Kuvassa 6 on yksinkertaistettu vuokaavio pääohjelmasta. Ensimmäiseksi pääohjelmassa tehdään alustuksia, jotka tehdään vain kerran mikrokontrollerin käynnistyessä. Alustuksella tarkoitetaan kontrollerin ja ohjelman toimintaan vaikuttavien rekistereiden sisältöjen muuttamista sopiviksi. Alustuksiin sisältyy nastojen toimintaan liittyvät asetukset, keskeytysten maskaaminen, ajastimien asetukset ja tiettyjen, ohjelmassa käytettävien rekistereiden arvojen alustaminen.

Suurimman osan ajasta mikrokontrolleri ei tee oikeastaan mitään hyödyllistä, vaan odottaa pelkästään keskeytyksen tapahtumista. Pääohjelmassa siis odotetaan, että keskeytys-lippu asetetaan.



Kuva 6. Pääohjelman vuokaavio

Vasta kun tämä tapahtuu, lähdetään pääohjelman koodia suorittamaan kokonaisuudessaan.

Alustusten jälkeen vuokaavion kolmannessa lohkossa jäädään odottamaan vakioaikavälikeskeytystä. Alla on ohjelmakoodi, jolla havaitaan keskeytyksen tapahtuminen.

```
ALKU   btfss  keskeytys

       goto   ALKU

       bcf    keskeytys
```

Ensimmäinen käsky selvittää keskeytys-lipun tilan ja suorittaa tarvittaessa hypyn seuraavan käskyn ohi. Niin kauan, kun tämä muuttuja on nollattuna, ohjelmassa suoritetaan vain kahta ensimmäistä tässä olevaa käskyriviä. Kun keskeytys tapahtuu ja keskeytys-lippu asetetaan keskeytyspalveluohjelmassa, hypätään goto-hyppykäskyn ohi, ja ohjelmaa lähdetään suorittamaan sitä seuraavasta käskystä, jolla nollataan keskeytys-lippu.

6.2.1 Neljännen lohkon toiminta

Neljännessä lohkossa tehdään useita eri asioita. Näitä ovat merkkivalon vilkutus, hälytystulojen valvominen, lähtöjen pulssien suorittaminen loppuun ja ohjauskoodin vastaanottamiseen kuluvan ajan mittaaminen.

6.2.1.1 Ledin vilkuttaminen

Portc:n nelosnastaan on kytketty ledi, jota vilkutetaan merkkinä ohjelman suorittamisesta. Seuraavalla, heti neljännen lohkon alussa olevalla ohjelmakoodilla tehdään tämän ledin sytytys tai sammutus.

```
movlw  0x10          ; akkuun ladataan luku 0x10

xorwf  PORTC, 1      ; RC4-lähdön tila vaihdetaan
                     ; xor-operaatiolla
```

6.2.1.2 Hälytystulojen valvonta

Seuraavaksi tehdään hälytysten kuittaus, mikäli kuittauspainiketta painetaan. Kuittaus tehdään sammuttamalla hälytysledit ja nollaamalla hälytyslaskurit.

```
btfss  KUITTAUSPAINIKE     ; testataan, painetaanko
                           ; RB5-nastaan kytkettyä
                           ; kuittauspainiketta

goto   sivuutus      ; jos ei paineta, hypätään sivuutus-
                     ; kohtaan

bcf    HÄLYTYSLEDI1 ; ledin sammutus

bcf    HÄLYTYSLEDI2 ; ledin sammutus

clrf   hälytys1      ; nollataan hälytys1
```

```
                clrf   hälytys11     ; nollataan hälytys11

                clrf   hälytys2      ; nollataan hälytys2

                clrf   hälytys22     ; nollataan hälytys22
```

Tämän jälkeen suoritetaan kaksi toisiaan vastaavaa koodinpätkää. Ero koodien välillä on se, että ensimmäinen pätkä koskee ensimmäistä hälytystuloa ja toinen pätkä toista hälytystuloa. Seuraava koodinpätkä koskee ensimmäistä hälytystuloa.

Koodissa lasketaan aikaa, jonka ensimmäinen hälytystulo eli RA4-nasta on ollut asettuneena yhtäjaksoisesti. Kun aika saavuttaa hälytyksen viiveeksi asetetun arvon, joka on talletettu aika5-rekisteriin, kyseisen tulon merkkivalo sytytetään. Hälytys11-rekisteriin lasketaan 50 ms:n jaksot, jotka tulo on ollut asettuneena. Koska aika5-rekisterissä olevan ajan yksikkö on sekunti, tarvitaan avuksi hälytys1-rekisteri, jonka arvoa kasvatetaan aina, kun 50 ms:n jaksoja on ollut 20 kappaletta. Näin ollen hälytys1-rekisterissä on sisältönä asettuneenaoloaika sekunteina.

Ensin katsotaan onko hälytystulo1 asettuneena. Jos tulo ei ole asettuneena, tehdään laskureiden nollaukset.

```
sivuutus        btfsc  HÄLYTYSTULO1 ; onko hälytystulo1 asettuneena

                goto   passaus      ; jos on hypätään passaus-kohtaan

                clrf   hälytys1     ; jos ei, nollataan hälytys1

                clrf   hälytys11    ; nollataan hälytys11

                goto   passaus2     ; hypätään passaus2-kohtaan
```

Sitten päivitetään laskureiden arvot ja verrataan kulunutta aikaa aika5-rekisterissä olevaan asetusarvoon.

```
passaus         incf   hälytys11, 1 ; kasvatetaan hälytys11:n arvoa

                movlw  0x14         ; ladataan akkuun luku 0x14 eli
                                    ; desimaaliluku 20

                subwf  hälytys11, 0 ; vähennetään akun arvo halytys11:stä

                btfss  nollalippu   ; katsotaan onko tulos nolla

                goto   passaus2     ; jos ei ole, hypätään passaus2-
                                    ; kohtaan

                incf   hälytys1, 1  ; jos on, kasvatetaan hälytys1:tä

                clrf   hälytys11    ; nollataan hälytys11

                movf   aika5, 0     ; ladataan aika5 akkuun

                subwf  hälytys1, 0  ; vähennetään akun arvo hälytys1:stä,
                                    ; tulos akkuun

                btfss  nollalippu   ; tutkitaan, onko tulos nolla

                goto   passaus2     ; jos ei ole, hypätään passaus2-
                                    ; kohtaan
```

```
                bsf    HÄLYTYSLEDI1 ; jos on, sytytetään merkkivalo

passaus2        ...                 ; tästä alkaa hälytystulo2:ta
                                    ; koskeva, vastaava, ohjelmakoodi
```

6.2.1.3 Pulssien suorittaminen loppuun

Lähtöihin tehtävät pulssit laitetaan alulle kahdeksannessa lohkossa mutta niiden keston mittaaminen ja lopettaminen tehdään tässä kohtaa. Otetaan esimerkiksi lähtö1 ja katsotaan miten nämä asiat tehdään. Ensin katsotaan onko lähdössä keskeneräinen pulssi. Jos pulssia ei ole, hypätään tarkastelemään lähdön 2 pulssitilannetta.

```
passaus4        btfss  pulssikesken1       ; onko lähdössä1 pulssi kesken

                goto   ohi1                ; jos ei ole, hypätään ohi1-
                                           ; kohtaan
```

Puolikymmenys1-laskuriin lasketaan pulssin kestoa 50 ms:n jaksoissa. Seuraavaksi kasvatetaan puolikymmenys1-laskurin arvoa. Pulssikesken1-lippu ilmaisee keskeneräisen pulssin. Kun jaksoja on 20, vähennetään aika1-muuttujan arvosta yksi sekunti.

```
                incf   puolikymmenys1, 1   ; jos on, kasvatetaan laskuria

                movf   puolikymmenys1, 0   ; ladataan laskurin arvo akkuun

                sublw  0x14                ; vähennetään akun arvo
                                           ; desimaaliluvusta 20

                btfss  nollalippu          ; testataan, onko tulos nolla

                goto   ohi1                ; jos ei ole, hypätään
                                           ; tarkastelemaan lähdön 2
                                           : laskuria

                clrf   puolikymmenys1      ; jos on, nollataan laskuri

                decf   aika1, 1            ; vähennetään jäljellä olevasta
                                           ; ajasta sekunti
```

Seuraavaksi katsotaan onko pulssi kestänyt määrätyn ajan. Jos on, lähdön 1 tila vaihdetaan.

```
                btfss  nollalippu          ; onko aikaa jäljellä

                goto   ohi1                ; jos on, hypätään ohi1-kohtaan

                bcf    pulssikesken1       ; jos ei ole, nollataan
                                           ; pulssikesken1-lippu

                movlw  0x01                ; ladataan akkuun luku 0x01

                xorwf  PORTB, 1            ; vaihdetaan lähdön 1 tila

ohi1            ...
```

6.2.1.4 Ohjauskoodin syöttöajan valvonta

Mikäli käyttäjä, syöttäessään ohjauskoodia, huomaa tehneensä virheen, hän saattaa vielä välttyä väärän toiminnon suorittamiselta odottamalla muutaman sekunnin. Neljännen lohkon lopussa tarkastellaan kuinka kauan käyttäjä on syöttänyt ohjauskoodia. Mikäli koodin syöttäminen kestää yli viisi sekuntia, siihen mennessä syötetty ohjauskoodi unohdetaan ja koodin vastaanottaminen aloitetaan alusta.

Aina, kun vastaanottimelta luetaan merkki, vastaanottomenossa-lippu asetetaan. Jos vastaanotto on käynnissä, vahtikoira-laskurin arvoa kasvatetaan.

```
ohi          btfsc  vastaanottomenossa  ; onko ohjauskoodin vastaanotto
                                        ; käynnissä

             incf   vahtikoira, 1       ; jos on, laskuria kasvatetaan
```

Sitten verrataan vahtikoira-laskurin arvoa lukuun 100, mikä vastaa viittä sekuntia.

```
             movf   vahtikoira, 0       ; vahtikoiran arvo ladataan
                                        ; akkuun

             sublw  0x64                ; vähennetään akun arvo
                                        ; desimaaliluvusta 100

             btfss  nollalippu          ; testataan, onko 5 sekuntia
                                        ; kulunut 1. merkin
                                        ; vastaanottamisesta

             goto   merkinluku          ; jos ei ole, hypätään
                                        ; merkinluku-kohtaan
```

Jos viisi sekuntia on kulunut, laskuri nollataan ja ohjauskoodin vastaanotto keskeytetään.

```
             clrf   vahtikoira          ; jos on, laskuri nollataan

             movlw  0x40                ; ladataan akkuun 0x40

             movwf  FSR                 ; osoittimen arvoksi ladataan
                                        ; akun sisältö

             bcf    vastaanottomenossa  ; nollataan vastaanottomenossa-
                                        ; lippu

merkinluku   ...
```

6.2.2 Merkin luku ja käsittely

Viidennessä lohkossa selvitetään onko DTMF-vastaanotin vastaanottanut uuden merkin. DTMF-vastaanottimen StD-signaalin seuraamiseen käytetään kahta lippubittiä, joihin talletetaan signaalin tila nykyisellä hetkellä ja sen tila, kun tarkastelu tehtiin edellisen kerran. Aina, kun StD-signaalissa esiintyy nousureuna, uusi merkki on vastaanotettu ja vastaanottimelta luetaan merkki. Seuraavaksi käydään läpi ohjelmakoodi, jolla tämä selvitys tehdään.

Ensin nollataan vanhastd-lippu. Tämän jälkeen asetetaan vanhastd-lippu mikäli std-lippu on asetettu.

```
merkinluku   bcf    vanhastd     ; nollataan vanhastd-lippu

             btfsc  std          ; katsotaan, onko std-lippu asetettu

             bsf    vanhastd     ; jos oli, vanhastd-lippu asetetaan
```

Seuraavilla käskyillä päivitetään std-lippu StD-signaalia vastaavaan tilaan.

```
             bcf    std          ; nollataan std-lippu

             btfsc  STDSIGNAALI  ; tutkitaan, onko DTMF-vastaanottimen
                                 ; StD-signaali asetettu

             bsf    std          ; jos oli, std-lippu asetetaan
```

Sitten katsotaan onko StD-signaalissa nousureuna. Jos nousureunaa ei ole, ohjelman suorituksessa hypätään ilmoitus-kohtaan yhdeksänteen lohkoon. Jos merkki on vastaanotettu, jatketaan kuudenteen lohkoon.

```
             btfsc  vanhastd     ; onko vanhastd 0

             goto   ilmoitus     ; jos ei, hypätään ilmoitus-kohtaan

             btfss  std          ; onko std 1

             goto   ilmoitus     ; jos ei ole, hypätään ilmoitus-
                                 ; kohtaan
             ...
```

Kuudennessa lohkossa luetaan vastaanotettu merkki DTMF-vastaanottimelta. Luetuille merkeille on varattu neljän tavun kokoinen ”taulukko” rekisterialueelta. Vastaanotetut ohjauskoodin merkit talletetaan järjestyksessä taulukkoon, jonka alkioita ovat rekisterit 0x41…0x44. Ensimmäisenä vastaanotettu merkki talletetaan taulukon ensimmäiseen alkioon eli rekisteriin 0x41. Merkkien luku voidaan suorittaa kätevästi epäsuoralla osoitustavalla.

Kun yhtään merkkiä ei ole luettu, osoittimen arvo on 0x40. Ensimmäiseksi siis kasvatetaan osoittimen arvoa yhdellä ja luetaan akkuun porta:n bitit, joihin sisältyy DTMF-vastaanottimen vastaanottama ja dekoodaama merkki.

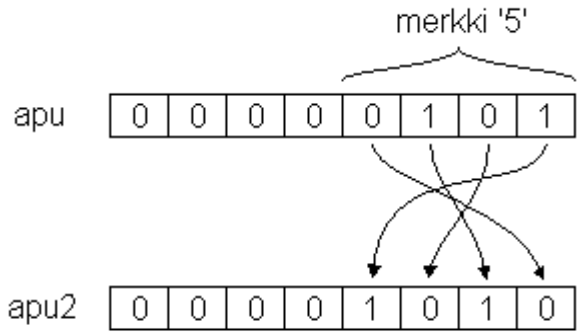
```
             incf   FSR, 1       ; kasvatetaan osoittimen arvoa

             movf   PORTA, 0     ; luetaan porta:n bitit akkuun
```

Koska vastaanotettu merkki on nyt apu-rekisterin neljässä vähiten merkitsevässä bitissä, apurekisterin neljä eniten merkitsevää bittiä maskataan eli nollataan. Sitten luettu tieto siirretään apu-rekisteriin jatkokäsittelyä varten.

```
             andlw  0x0f         ; maskaus

             movwf  apu          ; merkki kopioidaan apu-rekisteriin
```

Vastaanotettu merkki on nyt apu-rekisterissä, mutta fyysisen kytkennän vuoksi merkin bitit ovat käänteisessä järjestyksessä. Rekisterin sisältöä käsitellään vielä siten, että se vastaa vastaanotettua merkkiä, kuten taulukossa 1 on esitetty. Merkin käsittely tehdään seuraavalla ohjelmakoodilla. Käsittelyyn tarvitaan avuksi apu2-rekisteri. Kuva 7 havainnollistaa merkille tehtävää käsittelyä.

```
clrf   apu2          ; apu2-rekisterin nollaus

btfsc  apu, 0        ; testataan, onko apu-rekisterin
                     ; bitti 0 asetettu

bsf    apu2, 3       ; jos on, apu2-rekisterin
                     ; bitti 3 asetetaan

btfsc  apu, 1        ; testataan, onko apu-rekisterin
                     ; bitti 1 asetettu

bsf    apu2, 2       ; jos on, apu2-rekisterin
                     ; bitti 2 asetetaan

btfsc  apu, 2        ; testataan, onko apu-rekisterin
                     ; bitti 2 asetettu

bsf    apu2, 1       ; jos on, apu2-rekisterin
                     ; bitti 1 asetetaan

btfsc  apu, 3        ; testataan, onko apu-rekisterin
                     ; bitti 3 asetettu

bsf    apu2, 0       ; jos on, apu2-rekisterin
                     ; bitti 0 asetetaan

movf   apu2, 0       ; apu2-rekisteri ladataan akkuun

movwf  INDF          ; merkki kopioidaan taulukkoon
```



Kuva 7. Merkin käsittely

Akussa edelleen olevan merkin käsittelyä jatketaan vielä hieman. DTMF-vastaanotin esittää vastaanotetun '0'-merkin lukuna 0xA. Seuraavalla koodilla muutetaan luku 0xA nollaksi.

```
sublw  0x0A              ; vähennetään akun sisältö
                         ; luvusta 0x0A

btfsc  nollalippu        ; testataan, oliko tulos 0

clrf   INDF              ; jos oli, osoittimen osoittama
                         ; alkio nollataan
```

Viimeiseksi asetetaan vastaanottomenossa-muuttuja.

```
bsf    vastaanottomenossa        ; vastaanottomenossa-lippu
                                 ; asetetaan
```

6.2.3 Ohjauskoodin tarkastelu

Ohjauskoodin ensimmäinen merkki määrää suoritettavan toiminnon ja näin ollen myös ohjauskoodin pituuden. Lohkossa seitsemän tutkitaan montako merkkiä on vastaanotettu eli onko ohjauskoodi vastaanotettu kokonaan. Jos ohjauskoodi on vastaanotettu kokonaan, hypätään suorittamaan sen osoittama toiminto. Seuraavaksi käydään läpi koodia seitsemännen lohkon alusta.

Osoittimen arvon perusteella lasketaan vastanotettujen merkkien määrä. Osoittimen arvoa verrataan taulukon ensimmäisen alkion osoitteeseen eli lukuun 0x41. Jos arvot ovat samat, vastaanotettuja merkkejä on yksi. Tällöin hypätään hyppykäskyn ohi. Jos luettuja merkkejä on enemmän kuin yksi, suoritetaan hyppykäsky ja siirrytään ohtus-kohtaan, missä testataan onko vastaanotettuja merkkejä kaksi.

```
        movf   FSR, 0       ; ladataan osoitin akkuun

        sublw  0x41         ; vähennetään akun arvo luvusta 0x41

        btfss  nollalippu   ; katsotaan onko tulos 0

        goto   ohtus        ; jos tulos ei ollut 0, hypätään
                            ; ohtus-kohtaan
```

Nyt kun tiedetään, että on vastaanotettu yksi merkki, tutkitaan mikä tämä merkki on. Vastaanotettua merkkiä verrataan vuorotellen merkkeihin, joiden osoittaman toiminnon ohjauskoodi on yksimerkkinen. Taulukon kaksi mukaisesti nämä merkit ovat '2', '5' ja '6'. Jos merkin määräämän toiminnon vaatima ohjauskoodi on yksimerkkinen, voidaan hypätä suoraan toiminnon suorittamiseen lohkoon kahdeksan. Jos toiminto vaatii pidemmän ohjauskoodin, hypätään yhdeksänteen lohkoon.

Merkeille '2' ja '6' on samankaltaiset koodit kuin alla mutta nyt käydään läpi vain koodi, millä verrataan vastaanotettua merkkiä merkkiin '5'.

```
        ...

        movf   0x41, 0      ; ohjauskoodin 1. merkki
                            ; kopioidaan akkuun

        sublw  0x05         ; akun sisältö vähennetään luvusta
                            ; 0x05 (vastaa merkkiä '5')

        btfsc  nollalippu   ; katsotaan, oliko laskun tulos 0

        goto   kuittaus     ; jos ei ollut, hypätään
                            ; kuittaus-kohtaan, missä suoritetaan
                            ; hälytysten kuittaus
```

Jos ensimmäinen merkki ei ollut mikään edellä mainituista, hypätään yhdeksänteen lohkoon.

```
        goto   ilmoitus     ; jos ohjauskoodia ei ole vielä
                            ; vastaanotettu kokonaan, hypätään
                            ; lohkoon 9
ohtus   ...
```

Ohjauskoodin toinen merkki ilmaisee usein lähdön tai hälytystulon, johon toiminto kohdistuu. Kun on varmistettu ettei kyseessä ole salasanaan liittyvä toiminto, ohjauskoodin toisen merkin vastaanottamisen jälkeen suoritetaan alla oleva koodi, joka vähentää erityisesti lähtöjen ohjaukseen liittyvän ohjelmakoodin määrää. Aluksi nollataan lähtö-muuttuja ja asetetaan STATUS-rekisterissä oleva carry-lippu.

```
clrf   lähtö        ; nollataan lähtö-rekisteri

bsf    STATUS, 0    ; asetetaan carry-lippu
```

Tämän jälkeen kopioidaan merkki taulukon toisesta alkiosta apu2-rekisteriin ja siirrytään suorittamaan seuraavaa silmukkaa, kunnes apu2-rekisterin sisältö on nolla. Tässä kohtaa nollalippu on päivitetty apu2-rekisterin sisällön mukaan, joten jos rekisterin sisältö on nolla, alla olevaa silmukkaa ei suoriteta lainkaan. Silmukan yhdellä kierroksella shiftataan eli siirretään koko lähtö-rekisterin sisältöä vasemmalle yhden bitin verran. Shiftattaessa vasemmalle carry-lipun tila siirtyy rekisterin vähiten merkitseväksi bitiksi ja eniten merkitsevän bitin tila siirtyy carry-lipun tilaksi.

```
silmukka     btfsc  nollalippu   ; onko apu2-rekisterin sisältö 0

             goto   eteen        ; jos on, hypätään ulos silmukasta

             rlf    lähtö, 1     ; shiftaus vasemmalle

             decf   apu2, 1      ; pienennetään apu2:n sisältöä
                                 ; yhdellä

             goto   silmukka     ; hypätään silmukan alkuun

eteen        ...                 ; tästä alkaa koodi, jossa tutkitaan
                                 ; onko ohjauskoodi vastaanotettu
                                 ; kokonaan
```

Näin lähtö-rekisterin sisällöksi saadaan nollia ja yksi ykkönen. Tämä ykkönen on ohjattavan lähdön, PORTB-rekisterissä olevan ohjausbitin vastinbitti. Nyt lähtöjen ohjaustoimintoja suoritettaessa ei enää tarvitse selvittää vertailuoperaatioilla ohjattavaa lähtöä, vaan lähdön ohjaus voidaan tehdä jollakin, PORTB- ja lähtö-rekisterin välisellä loogisella operaatiolla. Seuraavassa kappaleessa on esimerkki lähdön ohjauksesta.

6.2.4 Toimintojen suoritus

Kahdeksannessa lohkossa suoritetaan ohjauskoodien mukaiset toiminnot. Jos vastaanotettu ohjauskoodi on virheellinen, mitään toimintoa ei suoriteta ja ohjelman suoritus jatkuu normaalisti.

Lähtöjen- ja hälytystenilmoitustoimintoja ja pulssitoimintoa ei voida suorittaa kertaluontoisesti, vaan niiden suorittaminen kokonaan vie tietyn ajan. Ilmoitustoimintojen tapauksessa asetetaan joko hälynilmoitus- tai lähdönilmoitus-lippu kyseessä olevan toiminnon mukaan. Ilmoitustoimintojen samanaikainen käyttö on estetty. Näiden toimintojen varsinainen suorittaminen tehdään yhdeksännessä lohkossa. Pulssitoiminnon kohdalla asetetaan kyseisen lähdön pulssikesken-muuttuja ja vaihdetaan lähdön tila.

Neljännessä lohkossa lasketaan aikaa pulssin aloittamishetkestä ja asetetun ajan kuluessa kyseisen lähdön tila vaihdetaan uudestaan.

Muut toiminnot voidaan suorittaa kokonaan tässä lohkossa. Alla on esimerkki kahdeksannessa lohkossa olevasta lähdön asettamisesta. Ensimmäisillä kahdella käskyllä ”nollataan” osoitin eli ladataan sen arvoksi 0x40. Ne eivät siis liity varsinaisen toiminnon suoritukseen.

```
Asetus      movlw  0x40
            movwf  FSR
```

Sitten nollataan muuttujat, joita käytetään ohjauskoodin vastaanottamiseen kuluvan ajan mittaamiseen.

```
            clrf   vahtikoira
            bcf    vastaanottomenossa
```

Seuraavaksi katsotaan onko lukitus päällä. Jos lukitus on päällä, toimintoa ei suoriteta ja hypätään yhdeksänteen lohkoon ilmoitus-kohtaan.

```
            btfsc  lukossa
            goto   ilmoitus
```

Seuraava koodi suorittaa varsinaisen toiminnon eli asettaa lähdön. Valittu lähtö asetetaan PORTB- ja lähtö-rekisterin välisellä or-operaatiolla, jonka tulos sijoitetaan PORTB-rekisteriin.

```
            movf   lähtö, 0     ; lähtö-rekisteri akkuun
            iorwf  PORTB, 1     ; asetetaan haluttu lähtö
            goto   ilmoitus     ; hypätään 9. lohkoon
```

6.2.5 Aikamuunnos-aliohjelma

Asetettaessa hälytyksen viivettä tai suoritettaessa pulssitoimintoa ohjauskoodin kaksi viimeistä merkkiä kertovat viiveen tai pulssin keston sekunteina. Aikamuunnos-aliohjelma muuntaa, 2-numeroisena desimaalilukuna syötetyn ajan heksadesimaaliluvuksi ja tallentaa tuloksen kyseessä olevalle lähdölle tai tulolle varattuun rekisteriin. Syötetyn ajan ykköset on siis rekisterissä 0x44 ja kymmenet rekisterissä 0x43. Ohjelmassa on silmukka, jossa tätä heksadesimaalilukua kasvatetaan luvulla 0x0A jokaista kymmentä sekuntia kohti. Ykkösiä ei tarvitse muokata mitenkään, koska ne ovat jo valmiiksi oikeassa muodossa. Lopuksi tämä luku lisätään ykkösiin ja näin saatu luku tallennetaan asianmukaiseen paikkaan. Seuraavaksi käydään läpi aikamuunnos-aliohjelman ohjelmakoodi.

Aluksi ladataan akkuun luku 0x0A. Sitten päivitetään nollalippu, jotta saadaan selville onko rekisterin 0x43 sisältö nolla.

```
aikamuunnos  movlw  0x0A         ; ladataan akkuun desimaaliluku 10
             movf   0x43, 1      ; päivitetään nollalippu
```

Tästä alkaa varsinainen silmukka. Ensin tarkastellaan onko rekisterin 0x43 sisältö nolla. Jos sisältö on nolla, tavoiteltu heksadesimaaliluku on jo valmiina rekisterissä 0x44, eikä silmukkaa tarvitse suorittaa kertaakaan. Tällöin hypätään ulos-kohtaan.

```
silmu       btfsc  nollalippu   ; testataan, onko kymmenet 0

            goto   ulos         ; jos on, hypätään ulos silmukasta
```

Silmukan sisällä ykkösiin lisätään luku 0x0A ja vähennetään kymmenistä luku 1. Sen jälkeen hypätään silmukan alkuun, missä taas katsotaan onko kymmeniä enää jäljellä.

```
            addwf  0x44, 1      ; lisätään akun sisältö, eli 0x0A,
                                ; rekisteriin 0x44

            decf   0x43, 1      ; vähennetään kymmenistä 1

            goto   silmu        ; hypätään silmukan alkuun
```

Kun aika on saatu muunnettua, se luetaan akkuun rekisteristä 0x44. Asetetun ajan tulee olla yhden ja 99 sekunnin välillä. Jos ajaksi on annettu nolla, se muutetaan sadaksi sekunniksi.

```
ulos        movf   0x44, 0      ; saatu aika akkuun

            btfsc  nollalippu   ; katsotaan, onko saatu aika 0

            movlw  0x64         ; jos aika on 0, ajaksi laitetaan
                                ; 100 sekuntia eli 0x64
```

Sitten tutkitaan lähtö-rekisterin perusteella, mille lähdölle tai tulolle aika on osoitettu. Alla oleva ohjelmakoodi koskee lähtöä 1. Muita lähtöja ja tuloja koskevat koodit ovat periaatteessa samanlaisia. Ensin katsotaan onko lähtö-rekisterin bitti 0 asetettu. Jos bittiä ei ole asetettu, hypätään muun1-kohtaan, missä taas tutkitaan onko bitti 1 asetettu.

```
            btfss  lähtö, 0     ; onko aika lähdölle 1

            goto   muun1        ; jos ei ole, hypätään muun1-kohtaan
```

Jos aika on osoitettu lähdölle 1, se tallennetaan aika1-rekisteriin ja pulssikesken1-lippu asetetaan merkiksi neljännelle lohkolle. Lopuksi nollataan laskuri, johon lasketaan pulssin kestoa ja palataan pääohjelmaan.

```
            movwf  aika1              ; saatu aika kopioidaan aika1-
                                      ; rekisteriin

            bsf    pulssikesken1      ; lipun asetus

            clrf   puolikymmenys1     ; laskurin nollaus

            return                    ; paluu pääohjelmaan

muun1       ...
```

6.2.6 Ilmoitustoiminnot

Viimeisessä lohkossa hoidetaan PWM-lähdön ohjaaminen mikäli joko lähtöjen- tai hälytystenilmoitustoiminto tai toimintojen lukitus on käytössä. Lohkon ohjelma

koostuu neljästä osasta: hälytysten ilmoitus, lähtöjen ilmoitus, laskurien kasvatus ja lukituksen merkkiäänen tekeminen. Tässä kappaleessa perehdytään kahteen ensimmäiseen osaan, jotka toimivat samalla periaatteella toisiinsa nähden ja joiden ohjelmakooditkin ovat lähes samanlaisia. Lisäksi katsotaan laskureiden kasvatus. Ensin käydään kuitenkin läpi kuinka PWM-lähdön varsinainen ohjaus tapahtuu ja sitten katsotaan lähtöjen ilmoitukseen liittyvää ohjelmaa.

6.2.6.1 PWM-lähdön asetukset

Lähdön ohjaukseen tehtiin kaksi aliohjelmaa. Pienif-aliohjelma alustaa PWM-lähdön ohjaukseen liittyvät rekisterit siten, että generoitavan signaalin taajuus on 1,22 kHz. Suurif-aliohjelma taas alustaa rekisterit 2,44 kHz:n signaalille. Seuraavaksi katsotaan, miten rekisterit on alustettava 2,44 kHz:n signaalille.

Alla olevaa kaavaa soveltamalla voidaan laskea PR2-rekisteriin sijoitettava luku halutun jaksonajan saamiseksi.

$$jaksonaika = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (TIMER2:n\ esijakajan\ arvo)$$

Kaava 1. Jaksonajan kaava

409,6 µs:n jaksonaika eli noin 2,44 kHz:n taajuus saadaan esijakajan arvolla 16 ja PR2:n arvolla 0x7F. Signaalin asettuneenaoloaika eli työjakso määritellään 10-bittisellä luvulla, jonka kahdeksan eniten merkitsevää bittiä sijoitetaan CCPR1L-rekisteriin ja kaksi vähiten merkitsevää bittiä CCP1CON-rekisteriin. Sijoitettava luku lasketaan seuraavan kaavan avulla.

$$työjakson\ pituus = (10 - bittinen\ luku) \cdot T_{OSC} \cdot (TIMER2:n\ esijakajan\ arvo)$$

Kaava 2. Työjakson kaava

Pulssisuhteeksi on valittu 50 %. Asettuneenaoloajan tulee siis olla puolet jaksonajasta eli 204,8 µs. Kaavaa käyttäen sijoitettavaksi luvuksi saadaan 0x100.

Seuraavaksi katsotaan suurif-aliohjelman toiminta. Koska PR2-rekisteri on pankissa 1, täytyy STATUS-rekisterin bitti 5 asettaa.

```
suurif      bsf    STATUS, 5    ; valitaan pankki 1
```

Sitten asetetaan jaksonaika eli PR2:n arvoksi ladataan 0x7F. Tämän jälkeen pankinvalintabitit asetetaan vastaamaan pankkia 0.

```
            movlw  0x7f         ; ladataan akkuun luku 0x7F
            movwf  PR2          ; kopioidaan akun sisältö PR2:een
            bcf    STATUS, 5    ; valitaan pankki 0
```

Seuraavaksi asetetaan pulssisuhde. Kun luku 0x100 jaetaan kahteen osaan rekistereiden vaatimalla tavalla, 8-bittiseksi luvuksi tulee 0x40 ja 2-bittiseksi luvuksi 0x0. Tämä 2-bittinen osa on sama molemmilla käytettävillä taajuuksilla, joten sen

alustaminen tehdään vain kerran, toisessa lohkossa. 8-bittisen luvun alustaminen tehdään näissä aliohjelmissa. Seuraava koodi lataa CCPR1L-rekisteriin luvun 0x40 ja tekee paluun pääohjelmaan.

```
movlw  0x40          ; ladataan akkuun luku 0x40

movwf  CCPR1L        ; kopioidaan kun sisältö CCPR1L:ään

return               ; palataan pääohjelmaan
```

Pienif-aliohjelma on vastaavanlainen mutta PR2-rekisteriin ladataan luku 0xFF ja CCPR1L-rekisteriin luku 0x80.

6.2.6.2 PWM-lähdön ohjauksen ajoitus

Kuvassa 3 on esimerkki lähtöjen ilmoituksesta, joten otetaan se esimerkiksi tässäkin kohtaa. Kuvasta 3 nähdään, että ilmoitusääni koostuu puolen sekunnin jaksoista. Kun kahdeksannessa lohkossa on asetettu lähdönilmoitus-lippu, yhdeksännessä lohkossa lähdetään laskemaan aikaa ilmoitusajastin2-rekisteriin. Lasketun ajan perusteella suoritetaan puolen sekunnin välein PWM-lähdön ohjaus.

Ohjelmassa katsotaan ensin onko lähtöjen ilmoitus -toiminto käytössä eli onko lähdönilmoitus-lippu asetettu. Jos toiminto ei ole käytössä, hypätään siihen liittyvänohjelman ohi passaus6-kohtaan.

```
passaus5    btfss  lähdönilmoitus     ; onko lähdön ilmoitus
                                      ; -toiminto käynnissä

            goto   passaus6           ; jos ei ole, hypätään
                                      ; passaus6-kohtaan
```

Tämän jälkeen on 11 koodinpätkää, jotka hoitavat PWM-lähdön ohjauksen. Alla oleva pätkä hoitaa puolen sekunnin jakson 2,5 sekunnin kohdalta eteenpäin. Kuvan 3 mukaisesti tässä jaksossa ilmoitetaan lähdön 2 tila. Ensin verrataan ilmoitusajastin2:n arvoa lukuun 0x32 eli desimaalilukuun 50. 50 kappaletta 50 ms:n jaksoja vastaa 2,5 sekuntia.

```
movlw  0x32                ; ladataan akkuun luku 0x32

subwf  ilmoitusajastin2, 0 ; vähennetään akun arvo
                           ; ilmoitusajastin2:sta, tulos
                           ; akkuun

btfss  nollalippu          ; onko tulos 0
```

Jos ilmoitustoiminto on kestänyt 2,5 sekuntia, tutkitaan lähdön 2 tila ja asetetaan PWM-lähdön asetuksen sen mukaisella tavalla. Muuten hypätään next6-kohtaan.

```
goto   next6         ; jos ei ollut, hyppy next6-kohtaan

btfss  LÄHTÖ2        ; testataan, onko lähtö 2 nollattuna

call   pienif        ; jos on, asetetaan pieni taajuus
                     ; kutsumalla pienif-aliohjelmaa

btfsc  LÄHTÖ2        ; testataan, onko lähtö 2 asetettuna
```

```
                call   suurif       ; jos on, asetetaan suuri taajuus
                                    ; kutsumalla suurif-aliohjelmaa
```

Koska PWM-lähtö generoi signaalia TIMER2-ajastimen mukaan, ajastin täytyy käynnistää asettamalla T2CON-rekisterin bitti 2.

```
                bsf    PWMon        ; käynnistetään TIMER2

                goto   passaus6     ; hypätään kohtaan, jossa
                                    ; kasvattamaan laskureita

next6           ...
```

Lopuksi sekä lähtöjen että hälytysten ilmoitustoimintojen koodien jälkeen kasvatetaan näiden toimintojen laskureita mikäli kyseinen toiminto on käytössä.

```
passaus6        btfsc  hälynilmoitus       ; ilmoitetaanko hälytysten
                                           ; tiloja

                incf   ilmoitusajastin1, 1 ; jos ilmoitetaan, kasvatetaan
                                           ; laskuria

                btfsc  lähdönilmoitus      ; ilmoitetaanko lähtöjen tiloja

                incf   ilmoitusajastin2, 1 ; jos ilmoitetaan, kasvatetaan
                                           ; laskuria
```

6.2.7 Lukituksen merkkiääni

Yhdeksännen lohkon, ja koko pääohjelman viimeisessä osassa hoidetaan lukituksesta kertovan merkkiäänen tuottaminen. Kun lukitus otetaan käyttöön, lukossa-lippu asetetaan kahdeksannessa lohkossa. Ja päinvastoin, kun lukitus avataan, kyseinen lippu nollataan. Ensin testataan onko lukitus päällä.

```
                btfss  lukossa             ; onko lukitus käytössä

                goto   ALKU                ; jos ei ole, hypätään
                                           ; lohkoon 2
```

Käytössä on lukitusääni-rekisteri, jota käytetään 50 ms:n jaksojen laskemiseen. Rekisteriin ladataan luku 0x15, joka vastaa sekunnin mittaista aikaa. Rekisterin sisältöä pienennetään joka ohjelmakierroksella, kunnes sen sisällöksi tulee nolla. Tämän tapahtuessa PWM-lähdön tila vaihdetaan eli se joko käynnistetään tai pysäytetään.

```
                movf   lukitusääni, 0      ; laskurin arvo akkuun

                btfss  nollalippu          ; onko aika 0

                goto   sivuun              ; jos ei ole, hypätään sivuun-
                                           ; kohtaan

                movlw  0x15                ; jos on, ladataan akkuun 0x15

                movwf  lukitusääni         ; kopioidaan akun sisältö
                                           ; lukitusääni-rekisteriin

                movlw  0x04                ; ladataan akkuun luku 0x04
```

```
                xorwf  T2CON, 1          ; pysäytetään/käynnistään PWM
sivuun          decf   lukitusääni, 1    ; pienennetään laskurin arvoa
                goto   ALKU              ; hypätään 2. lohkoon
```

7 YHTEENVETO

Tämän työn vaativin ja eniten aikaa vievä osuus oli mikrokontrollerin ohjelman tekeminen. Koska mikrokontrollereiden ohjelmoinnista ei oikeastaan ollut aiempaa kokemusta, pelkästään ohjelmoinnin alkuun pääsemiseen vaadittiin jonkin verran selvitystyötä. Tietysti MPLAB IDE -ohjelmistoon tutustuminenkin vei oman aikansa. Valmiin ohjelman rakenne on melko yksinkertainen eikä ohjelman teossa ollut suurempia ongelmia.

Työn toisen osan eli fyysisen laitteen suunnittelu ja rakentaminen oli huomattavasti helpompaa kuin ohjelman, koska kontrolleri ja DTMF-vastaanotin tarvitsivat hyvin vähän ulkoisia komponentteja. Suunnittelussa pärjäsi hyvin perustason elektroniikkasuunnittelutaidoilla. Ensimmäinen versio laitteesta rakennettiin koekytkentäalustalle. Seuraavaksi kytkentä siirrettiin reikälevylle ja lopulta piirilevylle. Myös työn tämä osa sujui ilman merkittäviä ongelmia.

Laitetta voidaan periaatteessa käyttää monenlaisissa eri kohteissa, mistä johtuen yleiskäyttöisen laitteen tekeminen on vaikeaa. Koska tämän työn laitetta ei suunniteltu mihinkään tiettyyn käyttökohteeseen, se on toteutettu yhdellä tavalla monien vaihtoehtoisten tapojen joukosta niin fyysisen laitteen kuin ohjelmankin osalta. Tavat, joilla laite on toteutettu, voivat rajoittaa sen käytettävyyttä sellaisenaan.

Laitteessa käytettyjen komponenttien ominaisuudet mahdollistavat laitteen käytön melko laajalla lämpötila-alueella. Laitetta ei ole testattu ääriolosuhteissa mutta normaaleissa olosuhteissa laitteen pitäisi toimia luotettavasti. Ohjelman osalta laitteen toimivuutta on pyritty testaamaan mahdollisimman kattavalla tavalla. Testauksien aikana laitteen toiminnassa ei ole ilmennyt virheitä.

LÄHDELUETTELO

Painamattomat lähteet

1. http://fi.wikipedia.org/wiki/DTMF

LIITELUETTELO

12.0 INSTRUCTION SET SUMMARY

Each PIC16F870/871 instruction is a 14-bit word, divided into an OPCODE, which specifies the instruction type, and one or more operands, which further specify the operation of the instruction. The PIC16F870/871 instruction set summary in Table 12-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 12-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 12-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
TO	Time-out bit
PD	Power-down bit

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles, with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 12-2 lists the instructions recognized by the MPASM™ assembler.

Figure 12-1 shows the general formats that the instructions can have.

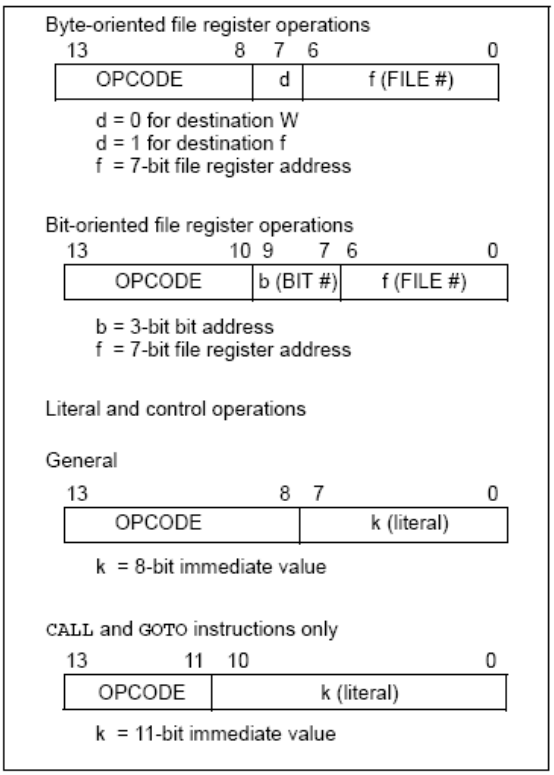
Note:	To maintain upward compatibility with future PIC16F870/871 products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 12-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

TABLE 12-2: PIC16F870/871 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00 0001 lfff ffff	Z	2
CLRW	-	Clear W	1	00 0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff	Z	1,2
DECF	f, d	Decrement f	1	00 0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff		1,2,3
INCF	f, d	Increment f	1	00 1010 dfff ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff	Z	1,2
MOVF	f, d	Move f	1	00 1000 dfff ffff	Z	1,2
MOVWF	f	Move W to f	1	00 0000 lfff ffff		
NOP	-	No Operation	1	00 0000 0xx0 0000		
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff		1,2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff		3
LITERAL AND CONTROL OPERATIONS						
ADDLW	k	Add literal and W	1	11 111x kkkk kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11 1001 kkkk kkkk	Z	
CALL	k	Call subroutine	2	10 0kkk kkkk kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00 0000 0110 0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10 1kkk kkkk kkkk		
IORLW	k	Inclusive OR literal with W	1	11 1000 kkkk kkkk	Z	
MOVLW	k	Move literal to W	1	11 00xx kkkk kkkk		
RETFIE	-	Return from interrupt	2	00 0000 0000 1001		
RETLW	k	Return with literal in W	2	11 01xx kkkk kkkk		
RETURN	-	Return from Subroutine	2	00 0000 0000 1000		
SLEEP	-	Go into Standby mode	1	00 0000 0110 0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11 110x kkkk kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11 1010 kkkk kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

```
        list        p=16f870
        #include    <p16f870.inc>      ; kontrollerikohtaiset rekisterimääritykset

        __CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_OFF & _HS_OSC &
        _WRT_ENABLE_ON & _LVP_OFF & _CPD_OFF


;                   - MUUTTUJIEN NIMEÄMINEN -

apu                         equ 0x20
apu2                        equ 0x21

liput1                      equ 0x22
liput2                      equ 0x23

ilmoitusajastin1            equ 0x24    ; hälytysten ilmoitukseen
ilmoitusajastin2            equ 0x25    ; lähtöjen ilmoitukseen

puolikymmenys1              equ 0x26    ; lähdön 1 pulssin tekemiseen
puolikymmenys2              equ 0x27    ; lähdön 2 pulssin tekemiseen
puolikymmenys3              equ 0x28    ; lähdön 3 pulssin tekemiseen
puolikymmenys4              equ 0x29    ; lähdön 4 pulssin tekemiseen
aika1                       equ 0x2A    ; laskuri lähdön 1 pulssille
aika2                       equ 0x2B    ; laskuri lähdön 2 pulssille
aika3                       equ 0x2C    ; laskuri lähdön 3 pulssille
aika4                       equ 0x2D    ; laskuri lähdön 4 pulssille
aika5                       equ 0x2E    ; asetusarvo, hälytystulo 1
aika6                       equ 0x2F    ; asetusarvo, hälytystulo 2
vahtikoira                  equ 0x30    ; laskuri ohj.koodin syöttöajan mittaamiseen
lukitusääni                 equ 0x31    ; ajastin lukituksen merkkiääntä varten
hälytys1                    equ 0x32    ; laskuri, kasvaa, kun häl.1-pinni = 1
hälytys2                    equ 0x33    ; laskuri, kasvaa, kun häl.2-pinni = 1
hälytys11                   equ 0x34    ; laskuri, jolla saadaan 0,05s -> sek.
hälytys22                   equ 0x35    ; laskuri, jolla saadaan 0,05s -> sek.
salasana1                   equ 0x36    ; salasanan tallennuspaikka, 1. merkki
salasana2                   equ 0x37    ; salasanan tallennuspaikka, 2. merkki
salasana3                   equ 0x38    ; salasanan tallennuspaikka, 3. merkki
muutos1                     equ 0x39    ; apumuuttujat salasanan muuttamiseen
muutos2                     equ 0x3A
muutos3                     equ 0x3B
muutos4                     equ 0x3C
muutos5                     equ 0x3D
muutos6                     equ 0x3E
lähtö                       equ 0x3F
akku_temp                   equ 0x40
status_temp                 equ 0x45

        #define     HÄLYTYSTULO1        PORTA, 4
        #define     HÄLYTYSTULO2        PORTA, 5

        #define     LÄHTÖ1              PORTB, 0
        #define     LÄHTÖ2              PORTB, 1
        #define     LÄHTÖ3              PORTB, 2
        #define     LÄHTÖ4              PORTB, 3
        #define     STDSIGNAALI         PORTB, 4
        #define     KUITTAUSPAINIKE     PORTB, 5

        #define     HÄLYTYSLEDI1        PORTC, 1
        #define     HÄLYTYSLEDI2        PORTC, 3

        #define     nollalippu          STATUS,2
        #define     PWMon               T2CON, 2

        #define     keskeytys           liput1, 0
        #define     pulssikesken1       liput1, 1
        #define     pulssikesken2       liput1, 2
        #define     pulssikesken3       liput1, 3
        #define     pulssikesken4       liput1, 4
        #define     vastaanottomenossa  liput1, 5
```

```
        #define     lukossa             liput1, 6
        #define     muutosyrit          liput1, 7

        #define     hälynilmoitus       liput2, 0
        #define     lähdönilmoitus      liput2, 1
        #define     std                 liput2, 2
        #define     vanhastd            liput2, 3


                ORG     0x000       ; ohjelmamuistin alkuosoite
                goto    main        ; hyppy pääohjelman alkuun


;               - VAKIOAIKAVÄLIKESKEYTYSPALVELUOHJELMA -

                ORG     0x004       ; keskeytyspalveluohjelman alkuosoite

                movwf   akku_temp   ; otetaan akun sisältö talteen akku_temp-muuttujaan
                movf    STATUS, 0   ; luetaan STATUS-rekisteri akkuun
                movwf   status_temp ; akun sisältö (STATUS) talteen status_temp:iin

                bcf     PIR1, 0     ; keskeytyksen asettaman lipun nollaus

                bsf     keskeytys   ; keskeytys-lipun asettaminen

                movlw   0xDC        ; ladataan akkuun luku 0xDC
                addwf   TMR1L,1     ; lisätään akussa oleva luku TIMER1:n arvoon
                movlw   0x0B        ; ladataan akkuun luku 0x0B
                addwf   TMR1H,1     ; TIMER1:n (kokonais)arvoon lisätään luku 0x0B00

                movf    status_temp, 0      ; vanha STATUS akkuun
                movwf   STATUS              ; STATUS-rekisterin palautus
                swapf   akku_temp, 1        ; bittien vaihto
                swapf   akku_temp, 0        ; bittien vaihto ja akun vanhan sisällön
                                            ; palauttaminen akkuun
                retfie                      ; paluu keskeytyspalveluohjelmasta


;********************** PÄÄOHJELMAN ALKU ************************

;                       ************************
;                       *****  2. LOHKO   *****
;                       ************************

;               - ALUSTUKSET -

main            bcf     STATUS, 6   ; valitaan pankki 1
                bsf     STATUS, 5   ; -''-
                movlw   0x3F        ; 0b00111111
                movwf   TRISA       ; TRISA -> pinnien suunnat (in/out)
                movlw   0xF0        ; 0b11110000
                movwf   TRISB       ; TRISB -> pinnien suunnat
                clrf    TRISC       ; portti C:n pinnit lähdöiksi
                movlw   0x06        ; 0b00000110
                movwf   ADCON1      ; ADCON1 -> kaikista pinneistä digitaalisia I/O

                movlw   0x80
                movwf   OPTION_REG


;               - KESKEYTYSTEN MASKAUS -

                movlw   0xC0
                movwf   INTCON      ; GIE = 1, PEIE = 1
                movlw   0x01
                movwf   PIE1        ; TIMER1:n ylivuotokeskeskeytyksen sallinta
                movlw   0x00
                movwf   PIE2
```

```
;           - MUITA ALUSTUKSIA -

            bcf   STATUS, 7    ; epäsuoran osoituksen pankeiksi 0 ja 1
            bcf   STATUS, 5    ; valitaan pankki 0

            movlw 0x40
            movwf FSR          ; indeksiksi 40h

            movlw 0x21
            movwf T1CON        ; T1CON: esijakajaksi 4, ajastin käy

            movlw 0x0C
            movwf CCP1CON      ; PWM mode

            movlw 0x02
            movwf T2CON        ; TIMER2: prescaler = 16, timer ei käy

            clrf  PORTB        ; PORTB:n lähtönastojen nollaus
            clrf  PORTC        ; PORTC:n lähtönastojen nollaus


;           - MUUTTUJIEN NOLLAUS -

            clrf  liput1
            clrf  liput2

            clrf  aika5
            bsf   aika5, 0     ; hälytys1:n viiveeksi sekunti

            clrf  aika6
            bsf   aika6, 0     ; hälytys2:n viiveeksi sekunti


;           ************************
;           *****  3. LOHKO   *****
;           ************************

;           - ONKO KESK. TAPAHTUNUT -

ALKU        btfss keskeytys    ; testataan, onko keskeytys-lippu asetettu
            goto  ALKU         ; jos ei ollut, hypätään ALKU:un
            bcf   keskeytys    ; jos oli, nollataan keskeytys-lippu


;           ************************
;           *****  4. LOHKO   *****
;           ************************

;           - LEDIN VILKUTUS -

            movlw 0x10         ; akkuun ladataan luku 0x10
            xorwf PORTC, 1     ; RC4-lähdön tila vaihdetaan xor-operaatiolla


;           - KUITTAUS PAINIKKEELLA -

            btfss KUITTAUSPAINIKE     ; testataan, painetaanko RB5-nastaan
                                      ; kytkettyä kuittauspainiketta
            goto  sivuutus            ; jos ei paineta, hypätään sivuutus-kohtaan
            bcf   HÄLYTYSLEDI1        ; ledin sammutus
            bcf   HÄLYTYSLEDI2        ; ledin sammutus
            clrf  hälytys1            ; nollataan hälytys1
            clrf  hälytys11           ; nollataan hälytys11
            clrf  hälytys2            ; nollataan hälytys2
            clrf  hälytys22           ; nollataan hälytys22
```

```
;           - HÄLYTYSLASKURI1:N KASVATUS -

sivuutus    btfsc  HÄLYTYSTULO1 ; onko hälytystulo1 asettuneena
            goto   passaus      ; jos on, hypätään passaus-kohtaan
            clrf   hälytys1     ; jos ei nollataan hälytys1
            clrf   hälytys11    ; nollataan hälytys11
            goto   passaus2     ; hypätään passaus2-kohtaan
passaus     incf   hälytys11, 1 ; kasvatetaan hälytys11:n arvoa
            movlw  0x14         ; ladataan akkuun luku 0x14, eli desimaaliluku 20
            subwf  hälytys11, 0 ; vähennetään akun arvo hälytys11:stä
            btfss  nollalippu   ; katsotaan, onko tulos nolla
            goto   passaus2     ; jos ei ole, hypätään passaus2-kohtaan
            incf   hälytys1, 1  ; jos on, kasvatetaan hälytys1:tä
            clrf   hälytys11    ; nollataan hälytys11

            movf   aika5, 0     ; ladataan aika5 akkuun
            subwf  hälytys1, 0  ; vähennetään akun arvo hälytys1:stä, tulos akkuun
            btfss  nollalippu   ; tutkitaan, onko tulos nolla
            goto   passaus2     ; jos ei ole, hypätään passaus2-kohtaan
            bsf    HÄLYTYSLEDI1 ; jos on, sytytetään merkkivalo


;           - HÄLYTYSLASKURIN2:N KASVATUS -

passaus2    btfsc  HÄLYTYSTULO2 ; sama toimintaperiaate, kuin edellisessä osassa
            goto   passaus3
            clrf   hälytys2
            clrf   hälytys22
            goto   passaus4
passaus3    incf   hälytys22, 1
            movlw  0x14
            subwf  hälytys22, 0
            btfss  nollalippu
            goto   passaus4
            incf   hälytys2, 1
            clrf   hälytys22

            movf   aika6, 0
            subwf  hälytys2, 0
            btfss  nollalippu
            goto   passaus4
            bsf    HÄLYTYSLEDI2


;           - PULSSILASKURIEN KASVATUS -

;                 - Lähtö 1 -
passaus4    btfss  pulssikesken1       ; onko lähdössä 1 pulssi kesken
            goto   ohi1                ; jos ei ole, hypätään ohi1-kohtaan
            incf   puolikymmenys1, 1   ; jos on, kasvatetaan laskuria
            movf   puolikymmenys1, 0   ; ladataan laskurin arvo akkuun
            sublw  0x14                ; vähennetään akun arvo desimaaliluvusta 20
            btfss  nollalippu          ; testataan, onko tulos nolla
            goto   ohi1                ; jos ei ole, hypätään tarkastelemaan lähdön
                                       ; 1 laskuria
            clrf   puolikymmenys1      ; jos on, nollataan laskuri
            decf   aika1, 1            ; vähennetään jäljellä olevasta ajasta
                                       ; sekunti
            btfss  nollalippu          ; onko aikaa jäljellä
            goto   ohi1                ; jos on, hypätään ohi1-kohtaan
            bcf    pulssikesken1       ; jos ei ole, nollataan pulssikesken1-lippu
            movlw  0x01                ; ladataan akkuun luku 0x01
            xorwf  PORTB, 1            ; vaihdetaan lähdön 1 tila

;                 - Lähtö 2 -
ohi1        btfss  pulssikesken2       ; kuten edellinen lähtö
            goto   ohi2
            incf   puolikymmenys2, 1
            movf   puolikymmenys2, 0
```

```
            sublw  0x14
            btfss  nollalippu
            goto   ohi2
            clrf   puolikymmenys2
            decf   aika2, 1
            btfss  nollalippu
            goto   ohi2
            bcf    pulssikesken2
            movlw  0x02
            xorwf  PORTB, 1

;                  - Lähtö 3 -
ohi2        btfss  pulssikesken3       ; kuten edelliset lähdöt
            goto   ohi3
            incf   puolikymmenys3, 1
            movf   puolikymmenys3, 0
            sublw  0x14
            btfss  nollalippu
            goto   ohi3
            clrf   puolikymmenys3
            decf   aika3, 1
            btfss  nollalippu
            goto   ohi3
            bcf    pulssikesken3
            movlw  0x04
            xorwf  PORTB, 1

;                  - Lähtö 4 -
ohi3        btfss  pulssikesken4       ; kuten edelliset lähdöt
            goto   ohi
            incf   puolikymmenys4, 1
            movf   puolikymmenys4, 0
            sublw  0x14
            btfss  nollalippu
            goto   ohi
            clrf   puolikymmenys4
            decf   aika4, 1
            btfss  nollalippu
            goto   ohi
            bcf    pulssikesken4
            movlw  0x08
            xorwf  PORTB, 1


ohi         btfsc  vastaanottomenossa
            incf   vahtikoira, 1
            movf   vahtikoira, 0
            sublw  0x64                ; 100dec
            btfss  nollalippu          ; onko 5 sekuntia kulunut 1. merkin
                                       ; vastaanottamisesta
            goto   merkinluku
            clrf   vahtikoira
            movlw  0x40                ; indeksi = 0
            movwf  FSR
            bcf    vastaanottomenossa  ; vastaanottomenossa = 0


;           ************************
;           *****  5. LOHKO   *****
;           ************************

;           - ONKO MERKKIÄ VASTAANOTETTU -

merkinluku  bcf    vanhastd            ; nollataan vanhastd-lippu
            btfsc  std                 ; katsotaan, onko std-lippu asetettu
            bsf    vanhastd            ; jos oli, vanhastd-lippu asetetaan
```

```
                bcf    std                 ; nollataan std-lippu
                btfsc  STDSIGNAALI         ; tutkitaan, onko DTMF-vastaanottimen StD-
                                           ; signaali asetettu
                bsf    std                 ; jos oli, std-lippu asetetaan

                btfsc  vanhastd            ; onko vanhastd 0
                goto   ilmoitus            ; jos ei, hypätään ilmoitus-kohtaan
                btfss  std                 ; onko std 1
                goto   ilmoitus            ; jos ei ole, hypätään ilmoitus-kohtaan


;               ************************
;               *****  6. LOHKO   *****
;               ************************

;               - MERKIN LUKU JA KÄSITTELY -

                incf   FSR, 1        ; kasvatetaan osoittimen arvoa
                movf   PORTA, 0      ; luetaan porta:n bitit akkuun
                andlw  0x0F          ; maskaus
                movwf  apu           ; merkki kopioidaan apu-rekisteriin

                clrf   apu2          ; apu2-rekisterin nollaus
                btfsc  apu, 0        ; testataan, onko apu-rekisterin bitti 0 asetettu
                bsf    apu2, 3       ; jos on, apu2-rekisterin bitti 3 asetetaan
                btfsc  apu, 1        ; testataan, onko apu-rekisterin bitti 1 asetettu
                bsf    apu2, 2       ; jos on, apu2-rekisterin bitti 2 asetetaan
                btfsc  apu, 2        ; testataan, onko apu-rekisterin bitti 2 asetettu
                bsf    apu2, 1       ; jos on, apu2-rekisterin bitti 1 asetetaan
                btfsc  apu, 3        ; testataan, onko apu-rekisterin bitti 3 asetettu
                bsf    apu2, 0       ; jos on, apu2-rekisterin bitti 0 asetetaan

                movf   apu2, 0       ; apu2-rekisteri ladataan akkuun
                movwf  INDF          ; merkki kopioidaan taulukkoon
                sublw  0x0A          ; vähennetään akun sisältö luvusta 0x0A
                btfsc  nollalippu    ; testataan, oliko tulos 0
                clrf   INDF          ; jos oli, osoittimen osoittama alkio nollataan

                bsf    vastaanottomenossa  ; vastaanottomenossa-lippu asetetaan


;               ************************
;               *****  7. LOHKO   *****
;               ************************

                ; Ensimmäisen merkin vastaanottamisen jälkeen

                movf   FSR, 0        ; ladataan osoitin akkuun
                sublw  0x41          ; vähennetään akun arvo luvusta 0x41
                btfss  nollalippu    ; katsotaan, onko tulos 0
                goto   ohtus         ; jos tulos ei ollut 0, hypätään ohtus-kohtaan

                movf   0x41, 0       ; 1. merkki akkuun
                sublw  0x02
                btfsc  nollalippu    ; halutaanko kysyä hälytykset
                goto   kysyhälytys   ; mennään jos merkki oli 2

                movf   0x41, 0       ; 1. merkki akkuun
                sublw  0x06
                btfsc  nollalippu    ; halutaanko kysyä lähdöt
                goto   kysylähtö     ; mennään jos merkki oli 6

                movf   0x41, 0       ; ohjauskoodin 1. merkki kopioidaan akkuun
                sublw  0x05          ; akun sisältö vähennetään luvusta 0x05 (vastaa
                                     ; merkkiä '5')
```

```
                btfsc  nollalippu   ; katsotaan, oliko laskun tulos 0
                goto   kuittaus     ; jos ei ollut, hypätään kuittaus-kohtaan, missä
                                    ; suoritetaan hälytysten kuittaus
                goto   ilmoitus

                ; Toisen merkin vastaanottamisen jälkeen

ohtus           movf   FSR, 0
                sublw  0x42
                btfss  nollalippu   ; nollalipun tarkistus -> onko indeksi 42
                goto   ohitus2      ; jos ei ole, hypätään ohitus2-kohtaan

                movf   0x41, 0      ; 1. merkki akkuun
                sublw  0x0B         ; akku vähennetään luvusta, joka vastaa '*'-merkkiä
                btfsc  nollalippu   ; onko tulos 0
                goto   ilmoitus     ; mennään, jos 1. merkki oli *

                movf   0x41, 0      ; 1. merkki akkuun
                sublw  0x0C         ; akku vähennetään luvusta, joka vastaa '#'-merkkiä
                btfsc  nollalippu   ; onko tulos nolla
                goto   ilmoitus     ; mennään, jos 1. merkki oli #


                ; halutun lähdön valinta -->

                clrf   lähtö        ; nollataan lähtö-rekisteri
                bsf    STATUS, 0    ; asetetaan carry-lippu
                movf   0x42, 0
                movwf  apu2

silmukka        btfsc  nollalippu   ; WHILE-silmukka, onko merkki 0
                goto   eteen
                rlf    lähtö, 1     ; shiftaa vasemmalle
                decf   apu2, 1      ; pienennä lukua
                goto   silmukka     ; hypätään silmukan alkuun

eteen           movf   0x41, 0      ; 1. merkki taulukosta akkuun
                btfsc  nollalippu   ; halutaanko nollata lähtö
                goto   nollaaminen  ; jos merkki on 0, hypätään

                sublw  0x01
                btfsc  nollalippu   ; halutaanko asettaa lähtö
                goto   asetus       ; jos merkki on 1, hypätään

                movf   0x41, 0      ; 1. merkki akkuun
                sublw  0x08
                btfsc  nollalippu   ; halutaanko vaihtaa lähdön tilaa
                goto   vaihto       ; jos merkki on 8, hypätään

                goto   ilmoitus


                ; Neljännen merkin vastaanottamisen jälkeen

ohitus2         movf   FSR, 0
                sublw  0x44
                btfss  nollalippu   ; nollalipun tarkistus -> onko indeksi 44
                goto   ilmoitus

                movf   0x41, 0      ; 1. merkki akkuun
                sublw  0x04
                btfsc  nollalippu   ; halutaanko asettaa hälytysaika
                goto   hälaika      ; mennään, jos 1. merkki oli 4

                movf   0x41, 0      ; 1. merkki akkuun
                sublw  0x07
                btfsc  nollalippu   ; halutaanko suorittaa pulssitoiminto
                goto   pulssikohta  ; mennään, jos 1. merkki oli 7
```

```
                movf   0x41, 0             ; 1. merkki akkuun
                sublw  0x0B
                btfsc  nollalippu          ; halutaanko suorittaa lukitus/avaus
                goto   lukitus             ; mennään, jos 1. merkki oli *

                movf   0x41, 0             ; 1. merkki akkuun
                sublw  0x0C
                btfsc  nollalippu          ; halutaanko suorittaa salasanan muutos
                goto   salamuutos          ; mennään, jos 1. merkki oli #

                clrf   vahtikoira          ; jos virheellinen koodi syotetty, nollataan
                                           ; vahtikoira
                bcf    vastaanottomenossa  ; ja vastaanottomenossa
                goto   ilmoitus


;               ************************
;               *****  8. LOHKO   *****
;               ************************

;               - TOIMINTOJEN SUORITUS -

lukitus         movlw  0x40
                movwf  FSR          ; indeksin nollaus
                clrf   vahtikoira
                bcf    vastaanottomenossa

                movf   0x42, 0      ; 2. merkki akkuun
                subwf  salasana1, 0
                btfss  nollalippu   ; oliko salasanan 1. merkki oikein
                goto   ilmoitus     ; jos ei ollut, hypätään

                movf   0x43, 0      ; 3. merkki akkuun
                subwf  salasana2, 0
                btfss  nollalippu   ; oliko salasanan 2. merkki oikein
                goto   ilmoitus     ; jos ei ollut, hypätään

                movf   0x44, 0      ; 4. merkki akkuun
                subwf  salasana3, 0
                btfss  nollalippu   ; oliko salasanan 3. merkki oikein
                goto   ilmoitus     ; jos ei ollut, hypätään

                movlw  0x40         ; jos salasana oli oikein, jatketaan tästä
                xorwf  liput1, 1    ; lukossa-lipun kääntö, aseta/poista lukitus
                bcf    PWMon        ; pysäytä PWM
                call   pienif
                btfss  lukossa
                goto   ilmoitus             ; hypätään, jos lukitus ei ole päällä
                bcf    lähdönilmoitus       ; jos lukitus on päällä,
                                            ; ilmoitustoiminnot lopetetaan
                bcf    hälynilmoitus
                clrf   ilmoitusajastin1
                clrf   ilmoitusajastin2
                goto   ilmoitus

salamuutos      movlw  0x40
                movwf  FSR          ; indeksin nollaus
                clrf   vahtikoira
                bcf    vastaanottomenossa
                btfsc  lukossa
                goto   ilmoitus     ; jos lukitus on päällä suoritusta ei tehdä

                movlw  0x80
                xorwf  liput1, 1    ; lipun kääntö
                btfss  muutosyrit
                goto   kakkonen
```

```
                movf   0x42, 0      ; kopioi syötetty salasana apumuuttujiin talteen
                movwf  muutos1
                movf   0x43, 0
                movwf  muutos2
                movf   0x44, 0
                movwf  muutos3
                goto   testaus

kakkonen        movf   0x42, 0      ; kopioi syötetty salasana apumuuttujiin talteen
                movwf  muutos4
                movf   0x43, 0
                movwf  muutos5
                movf   0x44, 0
                movwf  muutos6

testaus         movf   muutos1, 0   ; testataan, onko apumuuttujiin talletettu
                                    ; samat salasanat
                subwf  muutos4, 0
                btfss  nollalippu   ; onko ekat merkit samat
                goto   ilmoitus

                movf   muutos2, 0   ; onko tokat merkit samat
                subwf  muutos5, 0
                btfss  nollalippu
                goto   ilmoitus

                movf   muutos3, 0   ; onko kolmannet merkit samat
                subwf  muutos6, 0
                btfss  nollalippu
                goto   ilmoitus

                movf   muutos4, 0   ; jos salasanat olivat samat, salasana talletetaan
                                    ; uudeksi salasanaksi
                movwf  salasana1
                movf   muutos5, 0
                movwf  salasana2
                movf   muutos6, 0
                movwf  salasana3
                goto   ilmoitus

nollaaminen     movlw  0x40
                movwf  FSR                  ; indeksin nollaus
                clrf   vahtikoira
                bcf    vastaanottomenossa
                btfsc  lukossa              ; jos lukitus on päällä suoritusta ei tehdä
                goto   ilmoitus

                comf   lähtö, 1     ; komplementoi lähtö-rekisteri
                movf   lähtö, 0     ; lähtö akkuun
                andwf  PORTB, 1     ; nollataan haluttu lähtö
                goto   ilmoitus

asetus          movlw  0x40
                movwf  FSR                  ; indeksin nollaus
                clrf   vahtikoira
                bcf    vastaanottomenossa
                btfsc  lukossa              ; jos lukitus on päällä suoritusta ei tehdä
                goto   ilmoitus

                movf   lähtö, 0             ; lähtö-rekisteri akkuun
                iorwf  PORTB, 1             ; asetetaan haluttu lähtö
                goto   ilmoitus

kysyhälytys     movlw  0x40
                movwf  FSR                  ; indeksin nollaus
                clrf   vahtikoira
                bcf    vastaanottomenossa
                btfsc  lukossa              ; jos lukitus on päällä suoritusta ei tehdä
                goto   ilmoitus
```

```
             btfsc  lähdönilmoitus      ; jos lähdönilmoitus on jo päällä,
                                        ; hälytysten ilmoitusta ei tehdä
             goto   ilmoitus

             bsf    hälynilmoitus
             goto   ilmoitus

hälaika      movlw  0x40
             movwf  FSR                 ; indeksin nollaus
             clrf   vahtikoira
             bcf    vastaanottomenossa
             btfsc  lukossa             ; jos lukitus on päällä suoritusta ei tehdä
             goto   ilmoitus

             call   aikamuunnos         ; kutsutaan aikamuunnos-aliohjelmaa
             goto   ilmoitus

kuittaus     movlw  0x40
             movwf  FSR                 ; indeksin nollaus
             clrf   vahtikoira
             bcf    vastaanottomenossa
             btfsc  lukossa             ; jos lukitus on päällä suoritusta ei tehdä
             goto   ilmoitus

             bcf    HÄLYTYSLEDI1        ; ledin sammutus
             bcf    HÄLYTYSLEDI2        ; ledin sammutus
             clrf   hälytys1            ; nollataan hälytys1
             clrf   hälytys11           ; nollataan hälytys11
             clrf   hälytys2            ; nollataan hälytys2
             clrf   hälytys22           ; nollataan hälytys22
             goto   ilmoitus

kysylähtö    movlw  0x40
             movwf  FSR                 ; indeksin nollaus
             clrf   vahtikoira
             bcf    vastaanottomenossa
             btfsc  lukossa             ; jos lukitus on päällä suoritusta ei tehdä
             goto   ilmoitus
             btfsc  hälynilmoitus
             goto   ilmoitus

             bsf    lähdönilmoitus
             goto   ilmoitus

pulssikohta  clrf   vahtikoira
             btfsc  lukossa             ; jos lukitus on päällä suoritusta ei tehdä
             goto   ilmoitus

             call   aikamuunnos

vaihto       movlw  0x40
             movwf  FSR                 ; indeksin nollaus
             clrf   vahtikoira
             bcf    vastaanottomenossa
             btfsc  lukossa             ; jos lukitus on päällä suoritusta ei tehdä
             goto   ilmoitus

             movf   lähtö, 0            ; apu akkuun
             xorwf  PORTB, 1            ; asetetaan haluttu lähtö


;            ************************
;            *****  9. LOHKO   *****
;            ************************

;            - HÄLYTYSTEN ILMOITUS -

ilmoitus     btfss  hälynilmoitus
             goto   passaus5
```

```
                movf   ilmoitusajastin1, 1
                btfss  nollalippu
                goto   next
                call   pienif
                bsf    PWMon                    ; käynnistetään PWM
                goto   passaus5

next            movlw  0x0A                     ; 0,5 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfsc  nollalippu
                call   suurif

                movlw  0x14                     ; 1 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfsc  nollalippu
                bcf    PWMon                    ; pysäytä PWM

                ; Hälytys1:n ilmoitus

                movlw  0x1E                     ; 1,5 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfss  nollalippu
                goto   next2
                btfss  HÄLYTYSLEDI1
                call   pienif
                btfsc  HÄLYTYSLEDI1
                call   suurif
                bsf    PWMon
                goto   passaus5

next2           movlw  0x28                     ; 2 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfsc  nollalippu
                bcf    PWMon                    ; pysäytä PWM

                ; Hälytys1-pinnin tilan ilmoitus

                movlw  0x32                     ; 2,5 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfss  nollalippu
                goto   next3
                btfss  HÄLYTYSTULO1
                call   pienif
                btfsc  HÄLYTYSTULO1
                call   suurif
                bsf    PWMon
                goto   passaus5

next3           movlw  0x3C                     ; 3 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfsc  nollalippu
                bcf    PWMon                    ; pysäytä PWM

                ; Hälytys2:n ilmoitus
                movlw  0x46                     ; 3,5 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfss  nollalippu
                goto   next4
                btfss  HÄLYTYSLEDI2
                call   pienif
                btfsc  HÄLYTYSLEDI2
                call   suurif
                bsf    PWMon
                goto   passaus5

next4           movlw  0x50                     ; 4 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfsc  nollalippu
                bcf    PWMon                    ; pysäytä PWM
```

```
                ; Hälytys2-pinnin tilan ilmoitus

                movlw  0x5A                     ; 4,5 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfss  nollalippu
                goto   nexti
                btfss  HÄLYTYSTULO2
                call   pienif
                btfsc  HÄLYTYSTULO2
                call   suurif
                bsf    PWMon
                goto   passaus5

nexti           movlw  0x64                     ; 5 sekunnin päästä
                subwf  ilmoitusajastin1, 0
                btfss  nollalippu
                goto   passaus5
                bcf    PWMon                    ; pysäytä PWM
                clrf   ilmoitusajastin1
                bcf    hälynilmoitus


;               - LÄHTÖJEN ILMOITUS -

passaus5        btfss  lähdönilmoitus
                goto   passaus6

                movf   ilmoitusajastin2, 1
                btfss  nollalippu
                goto   nextt
                call   pienif
                bsf    PWMon                    ; käynnistetään PWM
                goto   passaus6

nextt           movlw  0x0A                     ; 0,5 sekunnin päästä
                subwf  ilmoitusajastin2, 0
                btfsc  nollalippu
                call   suurif

                movlw  0x14                     ; 1 sekunnin päästä
                subwf  ilmoitusajastin2, 0
                btfsc  nollalippu
                bcf    PWMon                    ; pysäytä PWM

                movlw  0x1E                     ; 1,5 sekunnin päästä
                subwf  ilmoitusajastin2, 0
                btfss  nollalippu
                goto   next5
                btfss  LÄHTÖ1
                call   pienif
                btfsc  LÄHTÖ1
                call   suurif
                bsf    PWMon
                goto   passaus6

next5           movlw  0x28                     ; 2 sekunnin päästä
                subwf  ilmoitusajastin2, 0
                btfsc  nollalippu
                bcf    PWMon                    ; pysäytä PWM

                movlw  0x32                     ; 2,5 sekunnin päästä
                subwf  ilmoitusajastin2, 0
                btfss  nollalippu
                goto   next6
                btfss  LÄHTÖ2
                call   pienif
                btfsc  LÄHTÖ2
                call   suurif
                bsf    PWMon
```

```
            goto   passaus6

next6       movlw  0x3C                    ; 3 sekunnin päästä
            subwf  ilmoitusajastin2, 0
            btfsc  nollalippu
            bcf    PWMon                   ; pysäytä PWM

            movlw  0x46                    ; 3,5 sekunnin päästä
            subwf  ilmoitusajastin2, 0
            btfss  nollalippu
            goto   next7
            btfss  LÄHTÖ3
            call   pienif
            btfsc  LÄHTÖ3
            call   suurif
            bsf    PWMon
            goto   passaus6

next7       movlw  0x50                    ; 4 sekunnin päästä
            subwf  ilmoitusajastin2, 0
            btfsc  nollalippu
            bcf    PWMon                   ; pysäytä PWM

            movlw  0x5A                    ; 4,5 sekunnin päästä
            subwf  ilmoitusajastin2, 0
            btfss  nollalippu
            goto   next8
            btfss  LÄHTÖ4
            call   pienif
            btfsc  LÄHTÖ4
            call   suurif
            bsf    PWMon
            goto   passaus6

next8       movlw  0x64                    ; 5 sekunnin päästä
            subwf  ilmoitusajastin2, 0
            btfss  nollalippu
            goto   passaus6
            bcf    PWMon                   ; pysäytä PWM
            clrf   ilmoitusajastin2
            bcf    lähdönilmoitus


passaus6    btfsc  hälynilmoitus
            incf   ilmoitusajastin1, 1

            btfsc  lähdönilmoitus
            incf   ilmoitusajastin2, 1


;           - LUKITUKSEN MERKKIÄÄNI -

            btfss  lukossa
            goto   ALKU

            movf   lukitusääni, 0          ; päivitä nollalippu
            btfss  nollalippu              ; onko aika 0
            goto   sivuun

            movlw  0x15
            movwf  lukitusääni
            movlw  0x04
            xorwf  T2CON, 1                ; pysäytä/käynnistä PWM

sivuun      decf   lukitusääni, 1

            goto   ALKU
```

```
;******************** PÄÄOHJELMAN LOPPU *********************


            ; AIKAMUUNNOS-ALIOHJELMA
            ; jakaa syötetyn sekuntimäärän 0,05 sekunnilla

aikamuunnos movlw  0x0A                     ; akkuun 10 Dec
            movf   0x43, 1
silmu       btfsc  nollalippu
            goto   ulos
            addwf  0x44, 1
            decf   0x43, 1
            goto   silmu
ulos        movf   0x44, 0                  ; saatu aika akkuun

            btfsc  nollalippu
            movlw  0x64                     ; jos aika on 0 -> ajaksi laitetaan
                                            ; 100 sekuntia

            btfss  lähtö, 0                 ; pulssi lähdölle 1
            goto   muun1
            movwf  aika1
            bsf    pulssikesken1
            clrf   puolikymmenys1
            return

muun1       btfss  lähtö, 1                 ; pulssi lähdölle 2
            goto   muun2
            movwf  aika2
            bsf    pulssikesken2
            clrf   puolikymmenys2
            return

muun2       btfss  lähtö, 2                 ; pulssi lähdölle 3
            goto   muun3
            movwf  aika3
            bsf    pulssikesken3
            clrf   puolikymmenys3
            return

muun3       btfss  lähtö, 3                 ; pulssi lähdölle 4
            goto   muun4
            movwf  aika4
            bsf    pulssikesken4
            clrf   puolikymmenys4
            return

muun4       btfss  lähtö, 4                 ; viive hälytystulolle 1
            goto   muun5
            movwf  aika5
            return

muun5       btfsc  lähtö, 5                 ; viive hälytystulolle 2
            movwf  aika6
            return


;           - SUUREN TAAJUUDEN ASETUKSET -

suurif      bsf    STATUS, 5                ; pankki 1
            movlw  0x7F                     ; -> f = 2,44 kHz
            movwf  PR2                      ; jaksonaika
            bcf    STATUS, 5                ; pankki 0
            movlw  0x40
            movwf  CCPR1L                   ; pulssisuhde 50%

            return
```

```
;            - PIENEN TAAJUUDEN ASETUKSET -

pienif       bsf    STATUS, 5                ; pankki 1
             movlw  0xFF                     ; -> f = 1,22 kHz
             movwf  PR2                      ; jaksonaika
             bcf    STATUS, 5                ; pankki 0
             movlw  0x80
             movwf  CCPR1L                   ; pulssisuhde 50%

             return


             END
```