ภาคผนวก ข

Soluna Security System's Main program

```
; Main program
; Developer: Soluna Crew2
; It will receive converted data from POCSAG decoder
; and store at memory location $D800

            . EQU       PORTA,   $1000          ; Port  A
            . EQU       PORTB,   $1004          ; Port  B
            . EQU       PORTC    $1003          ; Port  C
            . EQU       DDRC,    $1007          ; Data Direction for Port  C
            . EQU       OPTION,  $1039          ; System Configuration Options
            . EQU       DATA,    $D800          ; Address for keeping data
            . EQU       BUFFER,  $D808          ; Address for keeping dataC
            . EQU       DATAC,   $D600          ; Address for converted data
STRING      . DB            "20002543"
;--------------------------  Start of main program ----------------------------
        . ORG       $C000

        LDAA        OPTION
        ORAA        #00100000B              ; Set  IRQE = 1 (Select Edge Sensitive
        STAA        OPTION                  ; Only)

        LDAA        #00001100B              ; Set  BIT2-BIT3 to be OUTPUT for
        STAA        DDRC                    ; Port  C

; OFF Mode first before ON Mode                   .... SUP FEB 14, 2000
        LDAA        #10000001B              ; Entering OFF mode
        STAA        PORTC                   ; ON = 0,  SK = 0
        JSR         SEC1
;---------------------------------------
        LDAA        #10001001B              ; Entering On mode
        STAA        PORTC                   ; ON = 1,  SK = 0
        JSR         DLY35

; Add delay.  Wait to see 'OL' LED .... SUP  FEB 14, 2000
        JSR         SEC1
        JSR         SEC1
```

```
        JSR     SEC1
;---------------------------------------

        LDAA    #$7E            ; Set vector jump table for IRQ
        STAA    $00EE           ; to jump (7E) to location
        LDX     #$C800          ; C800 (interrupt service routine)
        STX     $00EF

YA      LDX     #DATA
        STX     $D700

        LDAA    PORTB           ; Start signal  00001111
        ORAA    #$01
        STAA    PORTB

        LDAA    #$30            ; Reset the receiver, use Port A BIT3-5
        STAA    PORTA           ; to be 110
        JSR     DLY10

        LDAA    #$10            ; Set the receiver to be TRACK state
        STAA    PORTA
        CLI                     ; Clear Interrupt Mask

; Timer for exit
        KDX     #$3400          ; Timer constant
        STX     $D703
        LDAA    #$FF            ; Flag set to FF --- Not activated
        STAA    $D702

WAIT
        LDAB    PORTB
        ORAB    #$01
        STAB    PORTB
        LDAA    $D702           ; Flag
        BNE     WAIT

MNDLY10
        LDAB    PORTB
        ORAB    #$02
        STAB    PORTB
        LDX     $D703           ; Timer
        DEX
```

```
        STX     $D703
        BNE     MNDLY10
        SEI                             ; INHABIT INTERUPT

        LDAB    PORTB
        ORAB    #$05
        STAB    PORTB
        JMP     CONVERTD
;-------------------------- Start of interrupt service routine ----------------------

        . ORG   $C800
        LDAA    PORTB                   ; Interrupt signal correct
        ORAA    #$01
        STAA    PORTB
        LDX     $D700

        LDAB    PORTC
        STAB    $0, X
        INX
        LDAA    #$00
        STAA    $D702                   ; Start countdown
        LDX     #$3400                  ; 10 ms Timer for main
        STX     $D703

        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        JSR     DLY35
        RTI

;----------------------------- Subroutine DLY35 -----------------------------
; Time delay of approx. 35 us
DLY35
        PSHA                            ; Preserve A
        LDAA    #$0B                    ; Init loop
LDLY35
        DECA
        BNE     LDLY35
```

```
        PULA                                    ; Restore A
        RTS

;------------------------------ Subroutine DLY10 ------------------------------
; Time delay of approx. 10 ms
DLY10
        PSHX
        LDX         #3400
LDLY10
        DEX
        BNE         LDLY10
        PULX
        RTS
;--------------------- SUPER POWER DELAY 10 -1 SEC -------------------------
DLY100
        PSHX
        LDX         #30000
LDLY100
        DEX
        BNE         LDLY100
        PULX
        RTS
;--------------------------- 1 SEC -----------------------------------
SEC1
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    JSR     DLY100
                    RTS
;------------------------------ CONVERTED ------------------------------
CONVERTED           LDX     #BUFFER
                    LDAB    #11111111B          ; End of data pattern

;------------------------------ Start of MASK ------------------------------
; Mask data to be 000X0000
MASK
```

```
        LDAA    $0, X           ; Check if end of data
        CBA
        BEQ     SORT
        ANDA    #00010000B      ; Mask data to be 000X0000

        LSLA                    ; Logical shift right from
        LSLA                    ; 000X0000 to X0000000
        LSLA

        STAA    $0, X
        INX
        BRA     MASK
;------------------------------ End of MASK ------------------------------

;------------------------------ Start of SORT ------------------------------
; Change data pattern to be
; X0000000
; 0X000000
; 00X00000
; 000X0000
; 0000X000
; 00000X00
; 000000X0
; 0000000X

SORT
        LDX     #BUFFER
        LDAB    #11111111B      ; End of data pattern
SORT_SUB
        LDAA    $0, X           ; Check if end of data
        CBA
        BEQ     COMBINE
        INX

        LSR     $0, X           ; X00000000 to 0X000000
        INX

        LSR     $0, X           ; X0000000 to 00X00000
        LSR     $0, X
        INX

        LSR     $0, X           ; X0000000 to 000X0000
```

```
        LSR     $0, X
        LSR     $0, X
        INX

        LSR     $0, X                   ; X0000000 to 0000X000
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        INX

        LSR     $0, X                   ; X0000000 to 00000X00
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        INX

        LSR     $0, X                   ; X0000000 to 000000X0
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        INX

        LSR     $0, X                   ; X0000000 to 0000000X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        LSR     $0, X
        INX
        BRA     SORT_SUB
;------------------------------ End of SORT ------------------------------

;---------------------------- Start of COMBINE ----------------------------
; Combine all of these data from location $D808 to be
; X0000000
; 0X000000
; 00X00000
; 000X0000  - - >   XXXXXXXX    (By oaring them together)
```

```
; 0000X000
; 00000X00
; 000000X0
; 0000000X
; And store them on location $D600

COMBINE
        LDX     #BUFFER
        LDY     #DATAC
        LDAB    #11111111B          ; End of data pattern
COMBINE_SUB
        LDAA    $0, X               ; Check if end of data
        CBA
        BEQ     CONVERT
        INX

        ORAA    $0, X
        INX
        ORAA    $0, X
        INX
        ORAA    $0, X
        INX
        ORAA    $0, X
        INX
        ORAA    $0, X
        INX
        ORAA    $0, X
        INX
        ORAA    $0, X
        INX

        STAA    $0, Y
        INY
        BRA     COMBINE_SUB
;----------------------------- End of COMBINE -----------------------------

;----------------------------- Start of CONVERT -----------------------------
; Change from 8-BIT BLOCK to 4-BIT BLOCK (12345670 to 00004321)
; For more information, please consult the PCF5001
; POCSAG Paging Decoder Data Sheet page 22

CONVERT
```

```
        LDX     #DATAC
        STAB    $0, Y                   ; Store end of data point
CONVERT_SUB
        LDAA    $0, X

        LDAB    #11111111B              ; End of data pattern
        CBA                             ; Check if end of data
        BEQ     BREAK_PT1               ; Can not jump in one time due to
                                        ; limitation of relative address
                                        ; range (-128 - +128)

        LDAB    #00001100B              ; Check if '0'
        CBA
        BEQ     A

        LDAB    #10001100B              ; Check if '1'
        CBA
        BEQ     B

        LDAB    #10001100B              ; Check if '2'
        CBA
        BEQ     C

        LDAB    #10001100B              ; Check if '3'
        CBA
        BEQ     D

        LDAB    #10001100B              ; Check if '4'
        CBA
        BEQ     E

        LDAB    #10001100B              ; Check if '5'
        CBA
        BEQ     F

        LDAB    #10001100B              ; Check if '6'
        CBA
        BEQ     G

        LDAB    #10001100B              ; Check if '7'
        CBA
        BEQ     H
```

```
        LDAB    #10001100B      ; Check if '8'
        CBA
        BEQ     I

        LDAB    #10001100B      ; Check if '9'
        CBA
        BEQ     J

        LDAB    #10001100B      ; Check if '*'
        CBA
        BEQ     K

        LDAB    #10001100B      ; Check if 'U'
        CBA
        BEQ     L

        LDAB    #10001100B      ; Check if ' '
        CBA
        BEQ     M

        LDAB    #10001100B      ; Check if '-'
        CBA
        BEQ     N

        LDAB    #10001100B      ; Check if ']'
        CBA
        BEQ     O

        LDAB    #10001100B      ; Check if '['
        CBA
        BEQ     P

        INX
        LDAB    PORTB           ; Error signal on LEDs 10101010 change to 01
        ORAB    #$01
        STAB    PORTB
        BRA     CONVERT_SUB

BREAK_PT1                       ; Can not jump in one time due to
        BRA     NUM_ASCII       ; limitation of relative address
                                ; range (-128 - +128)
```

```
BREAK_PT2
        BRA         CONVERT_SUB

A
        LDAA        #00000000B
        STAA        $0, X
        INX
        BRA         BREAK_PT2

B
        LDAA        #00000001B
        STAA        $0, X
        INX
        BRA         BREAK_PT2

C
        LDAA        #00000010B
        STAA        $0, X
        INX
        BRA         BREAK_PT2

D
        LDAA        #00000011B
        STAA        $0, X
        INX
        BRA         BREAK_PT2

E
        LDAA        #00000100B
        STAA        $0, X
        INX
        BRA         BREAK_PT2

F
        LDAA        #00000101B
        STAA        $0, X
        INX
        BRA         BREAK_PT2

G
        LDAA        #00000110B
        STAA        $0, X
```

```
        INX
        BRA     BREAK_PT2

H
        LDAA    #00000111B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

I
        LDAA    #00001000B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

J
        LDAA    #00001001B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

K
        LDAA    #00001010B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

L
        LDAA    #00001011B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

M
        LDAA    #00001100B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

N
        LDAA    #00001101B
        STAA    $0, X
```

```
        INX
        BRA     BREAK_PT2

O
        LDAA    #00001110B
        STAA    $0, X
        INX
        BRA     BREAK_PT2

P
        LDAA    #00001111B
        STAA    $0, X
        INX
        BRA     BREAK_PT2
;------------------------------ End of CONVERT ------------------------------

;--------------------------- Start of NUM_ASCII ------------------------------
; Data are now in bit stream, so cut them seven bits each to be
; ASCII format.

; * * * Algorithm * * *
; 00004321      1       00004321        -
; 00001765      2       00000765        MASK
;               3       07650000        <-- 4
;               4       07654321         1 + 3
;               5       00000001        --> 3
; 00005432      6       00054320        <-- 1
;               7       00054321         5 + 6
; 00002176      8       00000076        MASK
;               9       07600000        <-- 5
;               10      07654321         7 + 9
;               11      00000021        --> 2
; 00006543      12      00653200        <-- 2
;               13      00654321        11 + 12
; 00003217      14      00000007        MASK
;               15      07000000        <-- 6
;               16      07654321        13 + 15
;               17      00000321        --> 1
; 00007654      18      07654000        <-- 3
;               19      07654321        17 + 18

NUM_ASCII
```

```
        LDX     #DATAC
        LDY     #DATAC
        LDAB    #11111111B          ; End of data pattern
NUM_ASCII_SUB
        LDAA    $0, X
        CBA
        BEQ     BREAK_PT4
        STAA    $0, Y
        INX

        LDAA    $0, X
        CBA
        BEQ     BREAK_PT4
        ANDA    #00000111B          ; Mask
        LSLA
        LSLA
        LSLA
        LSLA
        ORAA    $0, Y
        STAA    $0, Y
        INY
        LDAA    $0, X
        LSRA
        LSRA
        LSRA
        STAA    $0, Y
        INX

        LDAA    $0, X
        CBA
        BEQ     BREAK_PT4
        LSLA
        ORAA    $0, Y
        STAA    $0, Y
        INX

        LDAA    $0, X
        CBA
        BEQ     BREAK_PT4
        ANDA    #00000011B
        LSLA
        LSLA
```

```
        LSLA
        LSLA
        LSLA
        ORAA    $0, Y
        STAA    $0, Y
        INY
        LDAA    $0, X
        LSRA
        LSRA
        STAA    $0, Y
        INX
        BRA     CONTINUE

BREAK_PT3                       ; Can not jump in one time due to
        BRA     NUM_ASCII_SUB   ; limitation of relative address
                                ; range  (-128 - +128)
BREAK_PT4
        BRA     FINISH

CONTINUE
        LDAA    $0, X
        CBA
        BEQ     FINISH
        LSLA
        LSLA
        ORAA    $0, Y
        STAA    $0, Y
        INX

        LDAA    $0, X
        CBA
        BEQ     FINISH
        ANDA    #00000001B
        LSLA
        LSLA
        LSLA
        LSLA
        LSLA
        LSLA
        ORAA    $0, Y
        STAA    $0, Y
        INY
```

```
        LDAA    $0, X
        LSRA
        STAA    $0, Y
        INX

        LDAA    $0, X
        CBA
        BEQ     FINISH
        LSLA
        LSLA
        LSLA
        ORAA    $0, Y
        STAA    $0, Y
        INY
        INX

        BRA     BREAK_PT3
;------------------------- End of NUM_ASCII ----------------------------

FINISH
                STAB    $0, Y           ; Store end of data point
; HERE          BRA     YA2
;---------------------------------- cutt ----------------------------------
YA2
                LDX     #STRING         ; STRING PASSWORD
                LDAA    $D600           ; DATAPASSWD
                CMPA    0, X            ; COMPARE DATA PASSWD DATASTRING

                BEQ     NEXT
                LMP     YA
NEXT
                LDAA    $D601
                CMPA    1, X
                BEQ     NEXT2
                JMP     YA
NEXT2
                LDAA    $D602
                CMPA    2, X
                BEQ     NEXT3
                JMP     YA
NEXT3
                LDAA    $D603
```

```
                CMPA    3, X
                BEQ     NEXT4
                JMP     YA
NEXT4
                LDAA    $D604
                CMPA    4, X
                BEQ     NEXT5
                JMP     YA
NEXT5
                LDAA    $D605
                CMPA    5, X
                BEQ     NEXT6
                JMP     YA
NEXT6
                LDAA    $D606
                CMPA    6, X
                BEQ     NEXT7
                JMP     YA
NEXT7
                LDAA    $D607
                CMPA    7, X
                BEQ     YA01
                JMP     YA
YA01
                LDAA    $D608
                CMPA    #$31
                BEQ     CUT1
                CMPA    #$32
                BEQ     ON2
                CMPA    #$33
                BEQ     START3
                CMPA    #$34
                BEQ     DOORC
                JMP     YA
;---------------------------- MODE CUTT ENGINE ----------------------------
CUT1
                LDAA    #10000000B
                STAA    PORTB
                JMP     YA
ON2
                LDAA    #$00
                STAA    PORTB
```

```
                JMP     YA
START3
                LDAA    #10000000B
                STAA    PORTB
                JSR     SEC1
                JSR     SEC1
                JSR     SEC1
                LDAA    #00100000B
                STAA    PORTB
                JSR     SEC1
                JSR     SEC1
                JSR     SEC1
                LDAA    #00110000B
                STAA    PORTB
                JSR     SEC1
                JSR     SEC1
                LDAA    #00100000B
                STAA    PORTB
                JMP     YA
DOORC
                LDAA    #01000000B
                STAA    PORTB
                JMP     YA
                . END

;------------------------- End of main program ----------------------------
```