The HT95R5x/6x FSK Decoder Functions

D/N：HA0228E

Introduction

A CID phone is used for Caller Identification, also known as caller ID, and is a telephone service provided by telecommunication companies to called parties. The CID system sends out information regarding the telephone number, caller ID and calling time to the called party for storage and number reviews.

The usual way of implementing a caller ID function is to use a caller program control phone switch to transmit the related information such as the caller number to the called party and sending the caller information using FSK (Frequency-shift Keying) or DTMF (Dual Tone Multi-Frequency) between the first and second ring intervals to the called party. The China telecommunication industry is using FSK to provide a caller ID service for users to subscribe.

Holtek's CID Phone 8-Bit MCUs, the HT95R5x/HT95R6x, includes an internal FSK decoder. The following content describes how to implement a Caller ID Identification with an FSK decoder in the HT95R5x/HT95R6x.

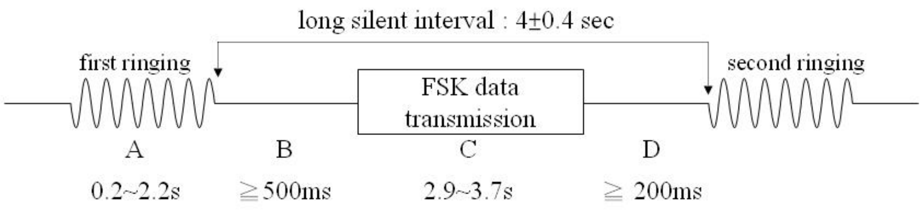
Operating Principles

The CID contains various specifications such as Bell 202 FSK and ETSI V.23 FSK, in which the Bell 202 is the FSK CID protocol made by Bell Laboratory and is the one mainly used in the China, Singapore and the US regional areas. The following gives an introduction to the Bell 202 FSK standard.

The parameters of the CID data transmitted by the program controlled switch are as follows.

Modulation Type：BFSK (Binary Frequency-Shift Keying)
Mark (Logic 1): 1200 ± 12Hz
Space (Logic 0): 2200 ± 22Hz
Transmission Speed : 1200 ± 12bps

A CID data transmission is divided into two types, an on-hook transmission and an off-hook transmission. CID data transmission in the on-hook status can also be classified by ringing and non-ringing types. The figure below is the CID data transmission timing of the ringing type in the on-hook status.



The FSK data includes Channel Seizure Signal, Mark Signal, Message and Checksum data.

- Channel Seizure Signal

  The first signal to be sent when transmitting the CID data, is composed of one group of 300 continuous ″0″ and ″1″ in turn with the first bit as a ″0″ and the last bit as a ″1″.

- Mark Signal

  Composed of 180 ″1″.

- Message

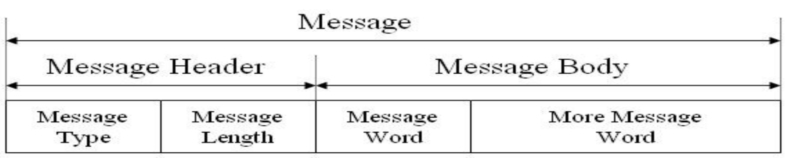
  There are two Message types, the SDMF (Single Data Message Format) and MDMF (Multiple Data Message Format.) The data format that the program controlled switch transmits to the recipient is the 8-bit ASCII.

- Checksum

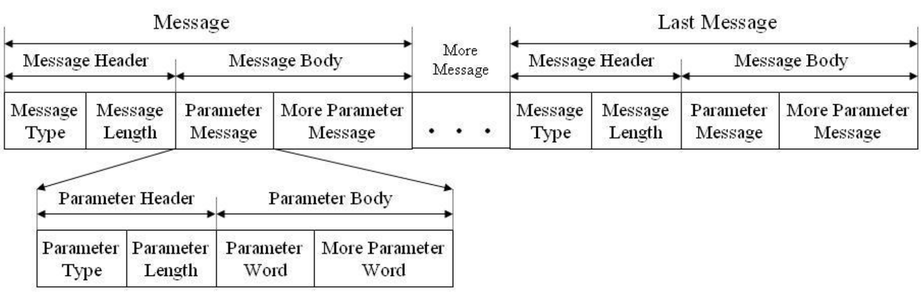
  One byte of data. The last byte of the result of all data from the Message plus with Checksum is ″0″.

The SMDF and MDMF data formats are shown as follows.

SDMF message



MDMF message



The following is one group of the FSK SDMF data:

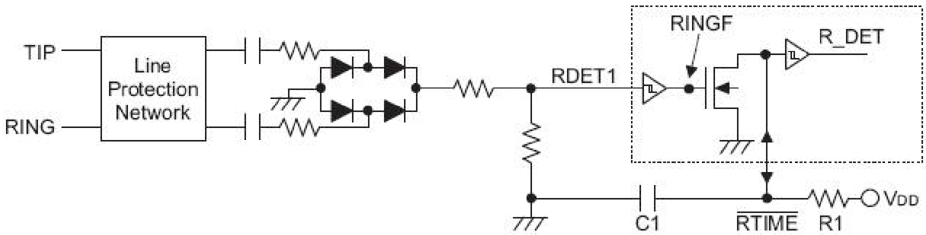
04H, 0FH, 30H, 31H, 32H, 33H, 31H, 36H, 35H, 39H, 35H, 36H, 33H, 31H, 39H, 39H, 39H, 39H, D8H.

In the above data:

- 04H means the Message type, the SDMF.
- 0FH means the Message length.
- 30H, 31H, 32H, 33H mean the date of 23rd January.
- 31H, 36H, 35H, 39H mean a time of 16:59.
- 35H, 36H, 33H, 31H, 39H, 39H, 39H, 39H means a telephone number: 5631999.
- D8H is a Checksum, used for CID data correcting.

The HT95R5x/HT95R6x includes an internal FSK decoder which supports CID specifications, the Bell 202 and ETSI V.23 with Ring and Line Reversal detection functions. The FSK decoder supports four interrupt sources to the peripheral interrupt vector, which are FSK raw data falling edge, ring detect or line reversal detect, FSK carrier detect and FSK packet data.

When a ring or line reversal occurs on the line, the internal signal R_DET is low. When an R_DET falling edge is detected by the device, the RDETF flag in the FSKS register will be set to ″1″. When an FSK carrier signal is detected by the device, the CDETF flag will set to ″1″. The reference circuit is shown below.



Registers relevant to the HT95R5x/6x and FSK are FSKC, FSKS, FSKD, PERIC.

FSKC Register (b7 … b0)

b7							b0
—	—	FSKSEL	CMSK	RMSK	FMSK	—	F_PWDN

FSKSEL: select FSK packet data source with ″1″ as DOUTC and ″0″ as DOUT.

CMSK: FSK carrier detect interrupt control bit. Set ″1″ to disable and ″0″ to enable.

RMSK: interrupt control bit for Ring or Line Reversal detect. Set ″1″ to disable and ″0″ to enable.

FMSK: interrupt control bit for FSK packet data. Set ″1″ to disable and ″0″ to enable.

F_PWDN: FSK decoder power control bit. Set ″1″ for power off and ″0″ for the operating mode.

FSKS Register (b7 … b0)

b7							b0
—	RINGF	FSKF	—	DOUTC	DOUT	CDETF	RDETF

RINGF: ringing signal flag, read only, cannot be modified by software.

FSKF: FSK packet data interrupt flag with ″1″ meaning ready and ″0″ meaning not ready.

DOUTC: FSK decoder COOK data output, read only, cannot be modified by software.

DOUT: FSK raw data output, read only, cannot be modified by software.

CDETF: FSK carrier detect interrupt flag, with ″1″ meaning detection of the ″0″ and ″1″ legal FSK signals, or ″0″ meaning illegal FSK signals.

RDETF: Ring or Line Reversal detect interrupt flag, with ″1″ meaning detection of Ring or Line Reversal and ″0″ meaning no detection.

b7							b0	
—	—	—	—	—	—	—	—	**FSKD Register**

FSKD is to store the FSK packet data.

b7							b0	
—	—	FSKDF	DTMFF	—	—	EFSKDI	EDTMFI	**PERIC Register**

FSKDF: FSK raw data falling edge interrupt flag, which will be set to ″1″ when an interrupt occurs.

DTMFF: DTMF receiver interrupt flag.

EFSKDI: FSK raw data falling edge interrupt enable bit. Set to ″1″ to enable and ″0″ to enable.

EDTMFI: DTMF receiver interrupt enable bit. Set to ″1″ to enable and ″0″ to disable.

Note that when EFSKDI is enabled, the device will automatically disable RMSK, CMSK and FMSK. If an interrupt is allowable, FSKDF, RDETF, CDETF and FSKF will all enable the external interrupt located at the 10H entry address. It will not be cleared to ″0″ by hardware after exiting from the interrupt subroutine and needs to be cleared by software instead before entering the next interrupt.

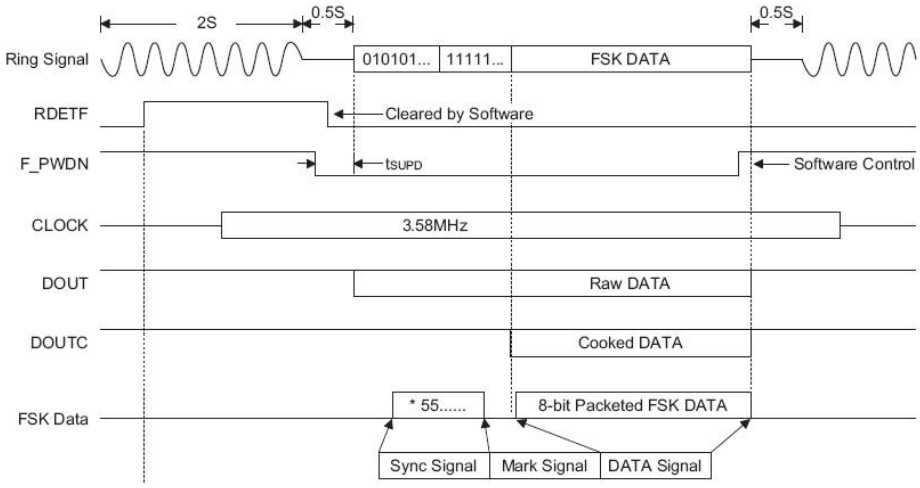
The FKS signals on the module analysis telephone lines, TIP and RING, of the FSK decoder can generate two data types, serial data and 8-bit packet data. To ensure normal operation of the FSK decoder, it is necessary to clear the F_PWDN in the FSKC register to ″0″. The serial FSK data is shown either by RAW or COOK data type monitored by the DOUT or DOUTC flags. When the decoder is in the operating mode, the DOUT flag outputs the decoder output data including Channel Seizure Signal (set by ″0″ and ″1″ in turn) and Message; the DOUTC flag means the decoder output which is similar to DOUT with Channel Seizure Signal excluded. If no FSK data is detected, both DOUT and DOUTC should remain at a high level. Therefore users can implement an FSK decoder function using the DOUT flag and a timer software.

In addition to the serial data type, the decoder also provides FSK Packet data. When the decoder receives an FSK signal, it will transform 10-bit data into an 8-bit packet data by neglecting the first and the tenth bits of the 10-bit data. The effective 8-bit packet data will be stored in the FSKD bit in the FSK register and the FSK packet interrupt flag FSKF will be set to ″1″. If the FMSK bit in the FSKC register is zero and the external interrupt is allowable, a peripheral interrupt will occur. The FSK decoder can packet the data in the

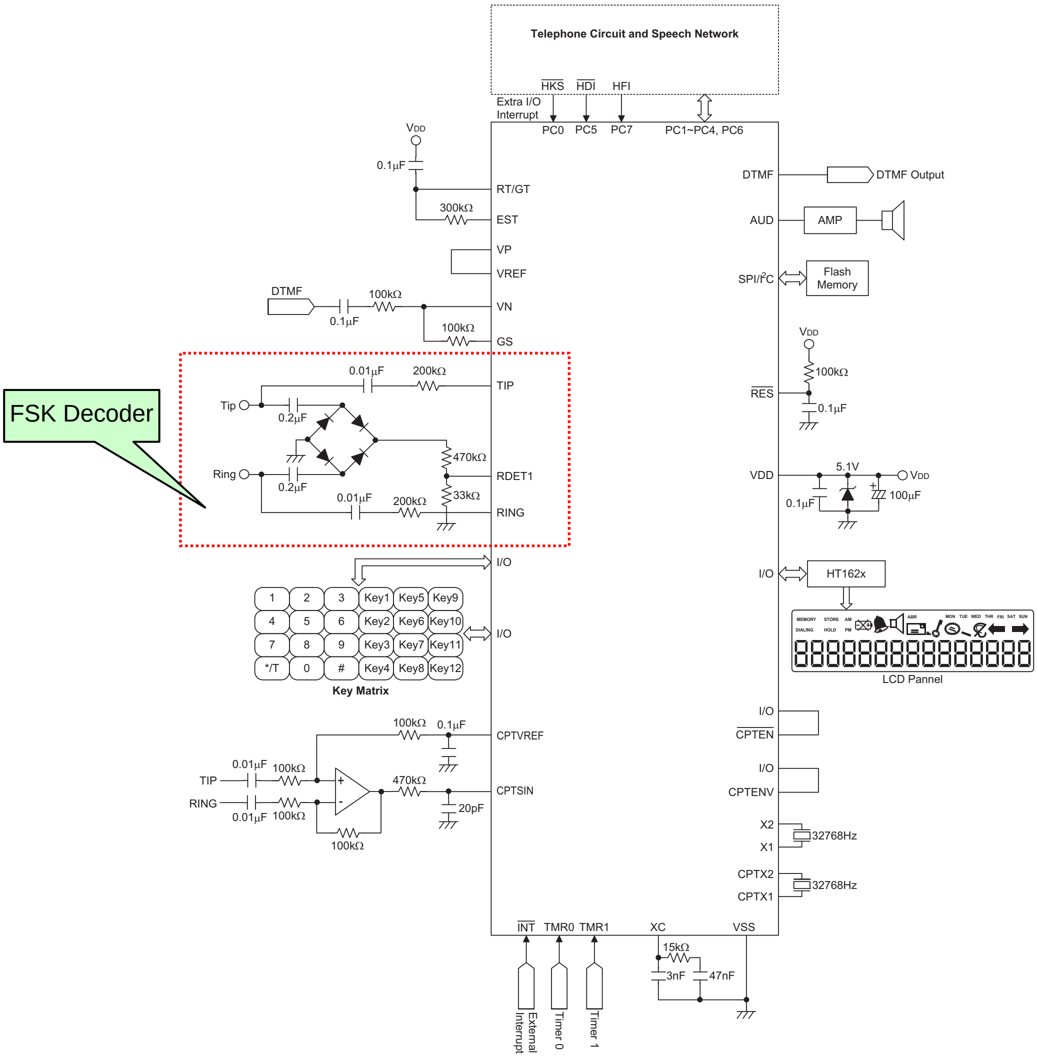
DOUT and DOUTC by setting the FSKSEL bit in FSKC. It should be noted that the first bit of the 10-bit data to be packeted should be zero as the Mark Signal data will not be packeted.

The device must be in the normal mode in order to detect FSK carriers, analyze serial data and transform 10-bit data into an 8-bit data packet. When in the green and sleep modes, the FSK decoder will analyze wrong data while the Ring and Line Reversal detect functions are still effective.

The FSK decoder Timing Figure is shown below.

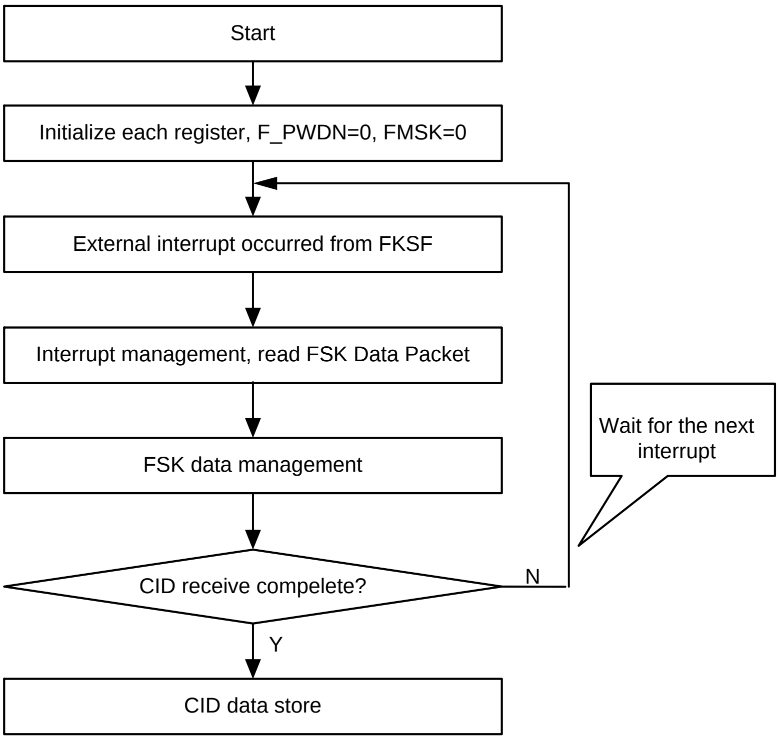


Application Circuits



(Refer to HT95R5x data sheet)

S/W Flowchart



Program Description

The program will receive CID data in SDMF and MDMF formats in the on hook status and store it in RAM Bank1 after processing. The SDMF data contains the time (ex. March 17th, 16:18) and a telephone number (ex. 5551212). The MDMF data contains the time, a telephone number and a name (ex. John Smith.)

Program Example

```
; Option: WDT Disable
include HT95R55.inc
ds .section 'data'
VAR0   DB   ?
RX     DB   40DUP(0)        ;save CID information
COUNT0 DB   ?
COUNT1 DB   ?
COUNT2 DB   ?
LENGTH DB   ?               ; CID Length from CID
LENGTH1  DB ?               ; LENGTH1=LENGTH+3,(Compared with LENGTH, Add message type,
                            ; message length and checksum)
FLAG_S DBIT                 ; Receive Single Data Message Format Flag
FLAG1  DBIT                 ; Finish receiving flag
FLAG0  DBIT                 ; Receive channel seizure signal 55H
FLAG_M DBIT                 ; Receive Multiple Data Message Format Flag
cs .section 'code'

ORG    0000H
JMP    MAIN_START
```

```
ORG    0004H
RETI
ORG    0008H
RETI
ORG    000CH
RETI
ORG    0010H
JMP    FSK_INT
ORG    0014H
RETI
ORG    0018H
RETI

ORG    0020H

MAIN_START:
  SET  UPEN
  CALL DELAY_20MS
  SET  MODE1
  CLR  F0
  CLR  F1                ; System Clock: HCLK 3.58MHz
  CALL CLEAR_RAM0
  MOV  A, 01H
  MOV  BP, A
  CALL CLEAR_RAM1        ; Clear RAM bank0 and bank1
  MOV  A, OFFSET RX
  MOV  MP0, A
  MOV  A, 40H
  MOV  MP1, A

  CLR  F_PWDN            ; FSK decoder work at operation mode
  CLR  CMSK              ; enable Carrier detect interrupt
  CLR  RMSK              ; enable Ring or line reversal detect interrupt
  CLR  FMSK              ; enable FSK packet data interrupt
  CLR  EFSKDI            ; disable FSK RAW data falling edge interrupt
  CLR  FSKSEL            ; select FSK packet data source DOUT
  SET  EMI
  SET  EPERI             ; enable peripheral interrupt
L0:
  SZ   FLAG_S
  JMP  L1
  SZ   FLAG_M
  JMP  L1
  JMP  L0
L1:
  SNZ  FLAG1
  JMP  L0
  CLR  EPERI
  MOV  A, 01H
  MOV  BP, A
  MOV  A, OFFSET RX
  MOV  MP0, A
L2:
  MOV  A, IAR0
  MOV  IAR1, A           ; store the CID from RX to ram bank1
  INC  MP0
  INC  MP1
  INC  COUNT0
  MOV  A, LENGTH1
  SUB  A, COUNT0
  SNZ  C
```

```
    JMP  L3
    MOV  A, COUNT0
    XOR  A, LENGTH1
    SNZ  Z
    JMP  L2
    INC  COUNT2
L3:
    CLR  FLAG_S
    CLR  FLAG_M
    CLR  FLAG1
    CLR  FLAG0
    CLR  COUNT0
    CLR  COUNT1
    CLR  LENGTH
    CLR  LENGTH1
    MOV  A, OFFSET RX
    MOV  MP0, A
    SET  EPERI
    JMP  L0

FSK_INT:
    CLR  EPERI
    SZ   CDETF              ; check FSK carrier detect interrupt flag
    CLR  CDETF
    SZ   RDETF              ; check ring or reversal detect interrupt flag
    CLR  RDETF
    SNZ  FSKF               ; check FSK packet data interrupt flag
    JMP  EXIT_INT
    CLR  FSKF
    SZ   FLAG1
    JMP  EXIT_INT
    CALL DECODE_SUB

EXIT_INT:
    CLR  PERF
    CLR  FSKDF
    CLR  RDETF
    CLR  CDETF
    CLR  FSKF
    SET  EPERI
    RETI

DECODE_SUB:
    SZ   FLAG0
    JMP  SUB0
    MOV  A, FSKD
    XOR  A, 55H
    SNZ  Z
    RET
    SET  FLAG0              ; detect FSK channel seizure signal 55H
SUB0:
    SZ   FLAG_S
    JMP  M1
    SZ   FLAG_M
    JMP  M1
    MOV  A, FSKD
    XOR  A, 04H             ; check SDMF message type
    SZ   Z
    JMP  S0
    MOV  A, FSKD
    XOR  A, 80H             ; check MDMF message type
```

```
    SZ   Z
    JMP  M0
    RET
S0:
    SET  FLAG_S
    JMP  M1

M0:
    SET  FLAG_M

M1:
    INC  COUNT1
    MOV  A, COUNT1
    XOR  A, 02H
    SNZ  Z
    JMP  M2
    MOV  A, FSKD
    MOV  LENGTH, A            ; get message length
    ADD  A, 03H
    MOV  LENGTH1, A

M2:
    MOV  A, FSKD
    MOV  IAR0, A              ; save the CID information to RX
    INC  MP0
    MOV  A, COUNT1
    SUB  A, 3
    SNZ  C
    RET
    MOV  A, COUNT1
    XOR  A, LENGTH1
    SZ   Z
    SET  FLAG1                ; check if Receive action finish
    RET

CLEAR_RAM0:
    MOV  A, 40H
    MOV  MP0, A
CLEAR0:
    CLR  IAR0
    MOV  A, MP0
    XOR  A, 0FFH
    SZ   Z
    RET
    INC  MP0
    JMP  CLEAR0

CLEAR_RAM1:
    MOV  A, 40H
    MOV  MP1, A
CLEAR1:
    CLR  IAR1
    MOV  A, MP1
    XOR  A, 0FFH
    SZ   Z
    RET
    INC  MP1
    JMP  CLEAR1

DELAY_20MS:
    MOV  A, 165
```

```
   MOV  VAR0, A
DELAY:
   SDZ  VAR0
   JMP  DELAY
   RET
```

Conclusion

The example program has implemented a SDMF and MDMF CID data decoding function which can be used as a reference source for user applications.