

Globaltec Electronics Ltd

Hyundai GMS34/36/37K Application note

Version : 2.0

Created by : Alex YEE
Technical Marketing Manager
Globaltec Electronics Ltd

Date; 18th Jan 2000

Acknowledgement

I would take this opportunity to acknowledge to Hyundai Application Team Leader Mr Jacky Lee , and his colleagues, Mr MJ Kang, Mr Ryu, whom shared a lot of their valuable experience , useful information to us in using these series of MCU.

Preface

This third version of the GMS34k/36k/37k MCU application note is based on the second version and has more information added.

This note is to summarize the necessary information in using this MCU either in remote application or non-remote application.

More subroutines are added to enrich the content and help engineers to get familiar with this MCU more easily.

Caution

- Globaltec Electronics Ltd. has the copyright of the stated software.
- The contents of this application note are subject to change for the reasons of improvement with notice, please contact Globaltec if you have any enquiry.
- The information, diagrams and other data in this application note are for reference only. Either Hyundai Microelectronics or Globaltec Electronics are in no way responsible for any violations of patents, other rights of the third party generated by the use of this application note or subroutine and any cause or damage by using the subroutine in this application note.

Table of Contents

Acknowledge

Preface

Section A Software

Section B Hardware

Section C Application Software

Section D Hyundai Standard Code Selection Guide

Section E Globaltec Standard Code Selection Guide

Section A Software

1. Compiler

In the given development tools, you should find the following software compiler

- 1.1. GMS30K.LIB
- 1.2. GR80.EXE
- 1.3. GS.BAT
- 1.4. GA80.EXE
- 1.5. MIDS_GS0.HLP
- 1.6. GMSCRF.EXE
- 1.7. GMSHEX.EXE
- 1.8. GMSLST.EXE
- 1.9. GMSROM.EXE
- 1.10.SIM.BIN
- 1.11.GMSTST.EXE
- 1.12.LOAD.EXE

Please check you got all this files in the zip file.

2 Running the compiler

2.1. SRC files

2.1.1. All program source code filenames must use SRC as the extension, e.g. test.src.

2.1.2. Syntax to run the compiler

By typing the GS <filename>, then your program source code will be generated. Four files with different extension will be created

It is better to keep the program file within the compiler directory.

Caution: do not type in the extension of the filename when using the GS command, otherwise your original <filename>.src will be erased. Please to make a backup to be filename.bak before doing compiling.

2.2. Compiled files

2.2.1. List file

2.2.1.1. <filename>.LST is the ROM instruction map and error code list

2.2.1.2. <filename>.RHX is the hex file for programming EPROM or software simulator or hardware emulator

2.2.1.3. ROM DUMP FILE

<filename>.dmp is storing the data for masking, to release masking , please submit the rhx files, dmp files and also the masking option and verification sheet (in the back of the GMS34K/36K/37K manuals)

3. Number

3.1. Only accepted decimal number, hex A-F are expressed as 10 –15. See the program example in appendix I

4. Program

4.1. Include Library

4.1.1. Inside the program, the “<TAB> INCLUDE GMS30K.LIB” must be placed in the top of your program file. You can create your program file by using ASCII code text editor, for an example, DOS edit command, or word processor or Qeditor, etc.

4.2. Page Header

4.2.1.1. In the beginning of the software, you must state which page of ROM memory you are using by “PG 00””PG 15”.

4.3. Each Page is only 64 byte long, please have to count for the extra in each page. Extra byte will cause compile error.

4.4. Initial Instructions

The first three instructions in the appendix are necessary in order for testing the MCU after IC fabrication. They are listed below:

PG 00

LXI 0

LYI 15

SO

4.4.1. A good practice to clear the accumulator when start the mcu, so the fourth instruction would be LAZ (load zero to accumulator).

4.5. Don't use watchdog timer reset (WDTR) in the call subroutine. This would further avoid the MCU to be hanged up in some unpredicated case happen especially in poor power supply environment.

4.6. Use BR \$ in the top and end of each page except PG 00. The usage if the MCU lost stack due to some power fluctuation or noise immunity, when the program reach this instruction, then the watch dog timer will action if the program is running not according to your original program flow. Of course, you must use BR instruction to jump to some useful location for your program. This two byte should not be your include in your program flow. They are the idle instruction.

4.7. Label

4.7.1. Label is placed at the beginning of each line. Duplicate labels will be warning in the compiler with "D" error code and fatal error message will be displayed.

4.8 Symbol definition (compiler directives)

4.8.1. EQU
Symbol EQU expression

e.g.

COUNT EQU 2

Usually the symbol defined with EQU served as symbolic constant.

4.8.2. REMARKS
Remarks begin with " ;".

4.8.3. END statement
At the end of the files, place the END to denote EOF.

4.8.4. ERROR MESSAGE
"U" will be marked in front of the undefined command.

4.9. Compiler Macro Instruction

4.9.1. BR & BL

4.9.1.1 BR (short branch)

Branch within the page

4.9.1.2. BL (long branch)

Branch across the page, two byte usage.

4.9.2 CAL & CALL

4.9.2.1. CAL (short call)

Call the subroutine within the page of memory.

4.9.2.2. CALL (Long Call)

Call the subroutine across the page of memory , two byte will be used.

5.0 Unused Instruction

Please fill in those unused instructions in the non-used program area. This is a must request by Hyundai in which the instructions are used for testing in mass production.

6.0 Unused area

Please fill NOP for unused ROM area.

7.0 Do not use WDT reset mask option when REMOUT, this would improve the noise immunity of the MCU.

8.0 Choose mask option a “state of L forcibly” when D-port is using as key matrix application.

9.0 Software simulator

9.1. Window and dos type software simulator are ready to borrow out for customer to develop their own software, please contact Globaltec for more details.

Section B Hardware

10.0. Resonator and crystal

When using 320KHz resonator, it is necessary to add two 100pf capacitors work with the 320KHz resonator though you choose the “INCLUSION OF CONDENSER FOR OSCILLATOR” to “YES” in the mask option list.

When using high speed clock, please use fosc /48 mask option, all the masked option for inclusion of condenser should choose “No” option.

Please note that either you use high speed crystal or low speed resonator, the internal cpu clock speed will not increase.

For low speed resonator, the clock speed will be fosc/6. For high speed crystal say 4MHz, the clock speed will be fosc/48.

11. Please consult the manual for the suitable brand name for the crystal and resonator, some of brand may not match the MCU oscillator and will have unstable oscillation.

12. Mask Option

12.1. State of “L” forcibly

This option will override the software setting of D port after STOP mode. Even you set D port to be high before stop mode, the state of D port will change to L level. Be ware of large current drainage especially in the key matrix application.

12.2. No state change in the STOP mode

It is recommended to use this option and let your software control the hardware.

12.3. D8, D9 of GMS34140

Please note that D8, D9 are must with mask option a, state of “L” forcibly, when designing hardware, please carefully with this mask option. Only for GMS34140.

13. EPROM EMULATOR

For work with the EVA board, it is advise to use eprom emulator to avoid continue programming eprom and erase eprom.

14. Emulator

Globaltec has developed its own Emulator for this series, please contact for details.

15. Advantage of 36K / 37K series

15.1 LVD

This is a very useful reset device in which will improve the reset timing when the remote is changing battery. When the low voltage is detected, the MCU will tight to reset state.

15.2 Inclusion the IR remout transistor. This can save one more external component. Please take care if customer are using two IR diode instead of one , please calculate the possible current sink into the MCU to avoid overcurrent of the REMOUT port.

Section C Software subroutine

16. Binary Increment routine

Purpose:

To increase the three digit binary counter, this counter is very useful when use in timer application.

Improvement

With a little change, it can be converted to binary /decimal up/down counter

```

        INCLUDE    GMS30K.LIB

;BASIC THREE DIGIT BINARY INCREMENT ROUNTINE
;      RAM ASSIGNMENT
;      DIGIT
;      15      14      13
;BR 0  MSB          LSB

ORG     0
LXI     0           ;point to ram location [0:13] – [0:15]
LYI     13

LMIY   14          ;STORE 14 INTO [0:13]
LMIY   15          ;STORE 15 INTO [0:14]
LMIY   15          ;STORE 15 INTO [0:15]

LAZ                    ;CLEAR ACCUMULATOR
LYI     13           ;POINT BACK TO [0:13]

INCR1  IM          ;A <- {0:13} +1
        BR        INCR2 ;BRANCH TO INCR2 IF DIGIT OVERFLOW
        XMA                    ; SAVE TO MEMORY [0:13]
        BR        QUIT

INCR2  LMAIY       ;[0:13] <- A, Y <- Y+1
        LAZ                    ; A <- 0
        BR INCR1   ;increase 10th digit

QUIT   END
```

17. Binary decrement routine

```
INCLUDE GMS30K.LIB
```

```
;BASIC THREE DIGIT BINARY DECREMENT ROUTINE
```

```
; RAM ASSIGNMENT
```

```
; DIGIT
```

```
; 15 14 13
```

```
;BR 0 MSB LSB
```

```
ORG 0
```

```
LXI 0 ;point to ram location [0:13] – [0:15]
```

```
LYI 13
```

```
LMIY 14 ;STORE 14 INTO [0:13]
```

```
LMIY 15 ;STORE 15 INTO [0:14]
```

```
LMIY 15 ;STORE 15 INTO [0:15]
```

```
LAZ ;CLEAR ACCUMULATOR
```

```
LYI 13 ;POINT BACK TO [0:13]
```

```
DECR1 DM ;A <- [0:13] – 1
```

```
BR DECR2 ;BRANCH IF [M] >= 1
```

```
BR DEC3 ;BR IF [M] = 0
```

```
NOP
```

```
DECR3 LMAIY
```

```
BR DECR1
```

```
DECR2 XMA
```

```
BR QUIT
```

```
QUIT END
```

18. CLEAR MEMORY

```
        INCLUDE    GMS30K.LIB
        PG 00
        ;Clear memory program
        ;RAM  register
        ;X    Y
        ;0    0 - 15
        ;1    0 - 15

        LXI    0        ;point to memory X = 0, Y =0

START  LAZ                ;clear accumulator
        CAL    CLSY      ;call clear X = 0 , Y=0 – 15 memory

L3     LXI    1        ;clear X=1, Y =0 - 15
        CAL    CLSY
        BR    START    ;end of the software

CLSY  LMA                ; A <- M(X, Y)
        IY                ;Y <- Y + 1
        BR    RT        ; if Y >15, branch to RTN
        BR    CLSY      ;if Y <= 15, branch to CLSY

RT     RTN

        END
```

19. Using the D port

;D-port is a open drain port, sink current is better than source output

;external pull-high resistor is needed

;if need a logic one output

```
        INCLUDE    GMS30K.LIB
        PG 00

        LXI    0        ; INIT.
        LYI    15
        SO
        LXI    0
        LYI    15
        RO

LP      LAZ
        LXI    2        ;turn off D8
        LYI    0        ;D8 = 0
        RO

        LXI    2
        LYI    1        ;turn on D9
        SO            ;D9 = 1

        WDTR            ; reset the watchdog timer

BR     LP            ; loop back for testing

        END
```

20. 0 to 99 decimal counter

```
        INCLUDE  GMS30K.LIB
        PG 00

MAIN    LXI 0
        LYI 15
        SO

        LAZ
CLEAR  LMA
        DY
        BR CLEAR

IDFT    LXI 0
        LYI 14
        LAZ

INCR1   IM
        ALEI 9
        BR IDFT1
        BR INCRDD

IDFT1   XMA
        BR IDFT

INCRDD  LAZ
        LMAIY
        IM
        ALEI 9
        BR IDFT1

        LYI 15
        LAZ
        LMA
        BR IDFT
        END
```

21 ;Decreasing Counter

INCLUDE GMS30K.LIB

PG 00

MAIN LXI 0

LYI 15

SO

LXI 0

LYI 14

LMIIY 15

LAZ

L1 LYI 14

DECR1 DM

XMA

MNEZ

BR DECR1

LYI 15

DM

XMA

MNEZ

BR L1

END