

PS/2 Communication

PS/2 Interface between auxiliary device using HMS87C1408B and Host (PC)

OVERVIEW

This document architects a class of auxiliary pointing devices which includes an extension of the PS/2 Mouse, Keyboard as used in the PC system. Typically, this includes such devices as trackballs, scratch pads and joystick. It also includes devices like TrackPoint II which allow the PS/2 auxiliary device port to be used by two or more such devices, either effectively simultaneously or alternately under program control.

The contents for data format, timing, PS/2 communication and operation of auxiliary device such as mouse, trackball, scratch pads and joystick is described in this application.

PS/2 CONNECTOR

The auxiliary device is connected to the system by a 6 pin or 5 pin connector. PS/2 keyboards can have either a 6-pin mini-DIN or a 5-pin DIN connector but mice are only available with 6-pin mini-DIN. You will not find a mouse with a 5-pin DIN connector. If your keyboard has a 6-pin mini-DIN and your computer has a 5-pin DIN the two can be made compatible with the adaptors. Both connectors are completely (electrically) similar, the only practical difference between the two is the arrangement of pins. The pinouts for each connector are shown in the following Figure 1.

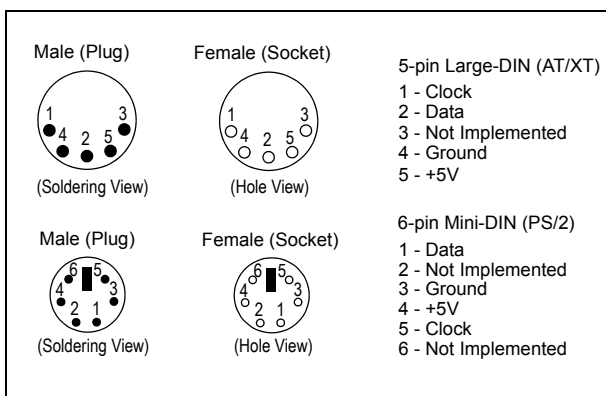


Figure 1. PS/2 Connector

DESCRIPTION

There are four interesting pins on the connectors just described; Ground, +5V, Data and Clock. The +5V is supplied by the host and the keyboard/Mouse's Ground is connected to the host's ground. Data and Clock are both

open collector, which means they are normally held at a high logic level (1) but can easily be pulled down to ground (logic 0). Any device you connect to a keyboard, mouse or host should have large pull-up resistors on the Clock and Data lines. You apply a "0" by pulling the line low and you apply a "1" by letting the line float high. But the ports used in here are C-MOS type bidirectional I/O port because of absence open-drain port.

Refer to Figure 2 as a simple interface to Data and Clock.

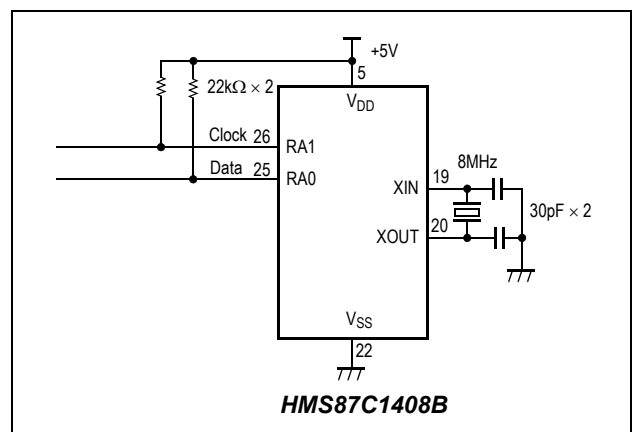


Figure 2. Simple Hardware Connections

DATA TRANSMISSION

The data transmission between auxiliary device and host is composed of 11 bits. This consists of one start bit, eight data bits, one parity bit and one stop bit. The data format is shown in below Figure 3.

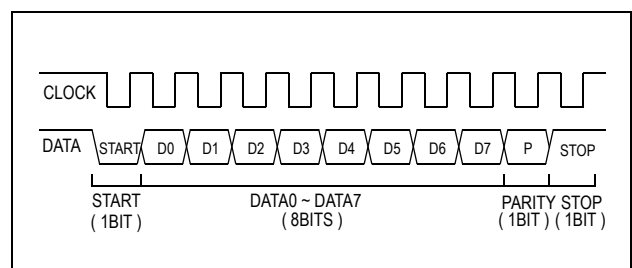


Figure 3. Device-to-host Communication

HOST RECEIVING DATA

Host (PC) samples data while CLK is low.

The following describes the typical sequence of events when the Host (PC) is receiving data from the auxiliary device.

1. The auxiliary device checks the CLK line. If the line is inactive (low), output from the auxiliary device is not allowed.
2. The auxiliary device checks the DATA line. If the line is inactive (low), the auxiliary device controller receives data from the Host.
3. The auxiliary device checks the CLK line periodically during the transmission at intervals not exceeding 100us. If the auxiliary device finds that the Host is holding the CLK line inactive (low), the byte transmission is terminated. The Host (PC) can terminate transmission anytime during the first 10 clock cycles.
4. A final check for terminated transmission is performed at least 5 us after clock ten.
5. The Host can hold the CLK line inactive (low) to inhibit the next transmission.
6. The Host can set the DATA line inactive (low) if it has a byte to transmit to the auxiliary device. The DATA line is set inactive (low) when the start bit (always 0) is placed on the DATA line.
7. The Host raises the CLK line to allow the next transmission.

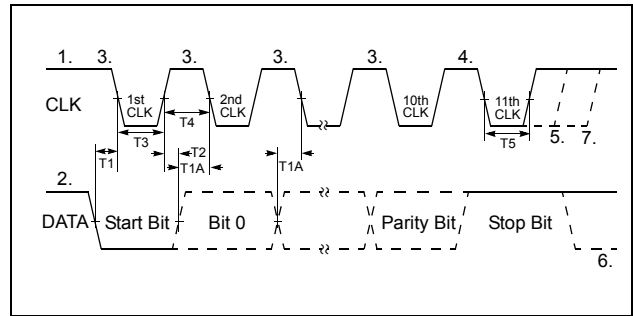


Figure 4. Auxiliary Device Sending Data Timings

Timing	Description	Spec. (Min/Max)
T1	Time from DATA transition to falling edge of CLK1	5/25us
T1A	Time from DATA transition to falling edge of CLK 2-11	5/25us
T2	Time from rising edge of CLK to DATA transition	5/T4-5us
T3	Duration of CLK inactive(low)	30/50us
T4	Duration of CLK active(high)	30/50us
T5	Time to auxiliary device inhibit after clock 11 to ensure the auxiliary device does not start another transmission	>0/50us
TCR	Line contention checking interval	0/100us

Table 1: Auxiliary Device Transmitting Data Timings

HOST SENDING DATA

Auxiliary device samples data while CLK is high.

The following describes the typical sequence of events when the Host(PC) is sending data to the auxiliary device

1. The Host checks for an auxiliary device transmission in process. If a transmission is in process and beyond the 10th clock, the Host must receive the data.
2. The auxiliary device checks the CLK line. If the line is inactive(low), an I/O operation is not allowed.
3. The auxiliary device checks the DATA line. If the line is inactive(low), the Host has data to transmit. The DATA line is set inactive(low) when the start bit(always 0) is placed on the DATA line.
4. The auxiliary device sets the CLK line inactive(low). The Host then places the first bit on the DATA line.
Each time the auxiliary device sets the CLK line inactive (after falling edge), the Host places the next bit on the DATA line until all bits are transmitted.
5. The auxiliary device samples the DATA line for each bit while the CLK line is active(high).
Data must be stable within 1microsecond after the rising edge of the CLK line.
6. The auxiliary device checks for a positive-level stop bit after the 10th clock. If the DATA line is inactive(low), the auxiliary device continues to clock until the DATA line becomes active(high).
Then the auxiliary device clocks the line-control bit and, at the next opportunity, sends a Resend command to the Host.
7. The auxiliary device pulls the DATA line inactive(low), producing the line-control bit.
8. The Host can pull the CLK line inactive(low), inhibiting the auxiliary device.

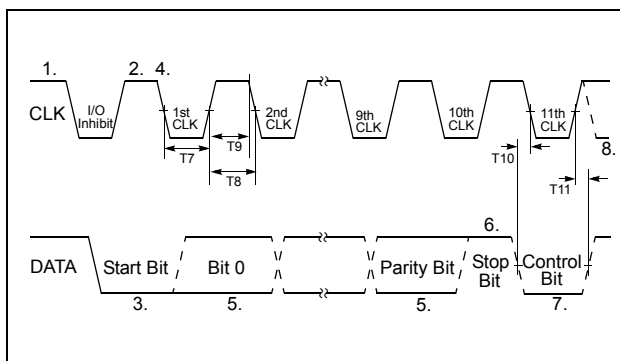


Figure 5. Auxiliary Device Receiving Data Timings

Timing	Description	Spec. (Min/Max)
T6	Auxiliary device response to Host "request to send"	30us/ 10ms
T7	Duration of CLK inactive(low)	30/50us
T8	Duration of CLK active(high)	30/50us
T9	Time from inactive to active CLK transition, used to time when the auxiliary device samples DATA	5/25us
T10	Time from falling edge of line control bit to falling edge of clock 11 CLK	5us/
T11	Time from rising edge of clock 11 to rising edge of line control bit	5/25us
TCX	Line contention checking interval	0/100us

Table 2: Auxiliary Device Receiving Data Timings

Author: Harris Hwang
 MCU application team
 e-mail: sungjae.hwang@magnachip.com

Appendix A:

GMS800 series MICOM ASSEMBLER Thu Jan 20 17:59:03 2005
(PAGE 1)

```

1          PAGE      1000
2          ;=====;
3          ;= Copy Right(c) MagnaChip Semiconductor 2005          =;
4          ;= All Right Reserved.                                =;
5          ;=====;
6          ;= Title   : PS/2 Communication                        =;
7          ;= Subject : 1, Transmit data(MSEADATA) to PC through PS/2 =;
8          ;=         2. Get the mouse command from PC through PS/2 =;
9          ;=         and Store it to Memory(MSECOMMAND)          =;
10         ;= Description:                                       =;
11         ;= Keyboard and mouse are connected to Personal Computer through PS/2 =;
12         ;= Port. The communication through PS/2 port is similar to that of I2C. =;
13         ;= That is, It is use two wires, one is clock and The other is data line.;
14         ;= This program is referred to IBM PS/2 Mouse specification(Ref.II,III) =;
15         ;=====;
16         ;= Device   : HMS87C1408B                              =;
17         ;= OSC      : 8MHz                                     =;
18         ;= Start Date : 2005, 1,                               =;
19         ;= End Date: 2005, 1,                                   =;
20         ;= S/W      : AP/MCU Application Design Team           =;
21         ;= Developer: Harris Hwang                             =;
22         ;=====;
23         ;
24         ;Control Registers
25         RA      EQU      0C0H          ;[R/W] RA Port Data Reg.
26         RAIO    EQU      0C1H          ;[W] RA Port Direction Reg.
27         RAFUNC  EQU      0CAH          ;[W] RA Fuction Selection Reg.
28         IRQLO  EQU      0E5H          ;[R/W] INT Request Reg. Low
29         WDTIF   EQU      6,0E5H
30         BITIF   EQU      5,0E5H
31         BITR    EQU      0ECH          ;[R] Basic Interval Timer Reg.
32         CKCTLR  EQU      0ECH          ;[W] Clock Control Register
33         WDTR    EQU      0EDH          ;[R],[W] Watchdog Timer Register
34
35
36         ;User Memory
37         MSEADATA DS      1             ;The data to transmit to PC
38         MSECOMMAND DS    1             ;The command to get from PC
39         COMMANDBUF DS    1             ;Store command from PC.
40         BITCNT   DS      1             ;To count the transmitted bit
41         PARITY   DS      1             ;To check parity
42         ;Constant
43         STACK    EQU      0BFH
44         OUTPUT_MODE EQU    1111_1111B ;MSESCL=RA.1,MSESDA=RA.0
45         INPUT_MODE EQU    1111_1100B ;MSESCL=RA.1,MSESDA=RA.0
46         SDA_INPUT_MODE EQU  1111_1110B ;MSESCL=RA.1,MSESDA=RA.0
47         ;Port Definition
48         MSESDA   EQU      0,RA         ;Mouse Data Definiton
49         MSESCL   EQU      1,RA         ;Mouse Clk Definition
50         ;Macro Definition
51         ble      macro                 ;A < #imm jump \1(label)
52             bcc  \1                    ;branch less (after CMP)
53         endm
54         ;=====;
55         ;= Vector Area                                          =;
56         ;=====;
57         ORG      OFFFEH
58 FFFE 00E0      DW      Reset          ; Reset

```

```

59      ;=====;
60      ;= Main Routine                                     =;
61      ;=====;
62      ORG      0E000H      ;HMS87C1408B Start Address
63      Reset:
64 E000 60      di
65 E001 40      clrg
66 E002 1EBF    ldx      #STACK      ;Stack Point(00-BFh,OnlyExist 0-page,192byte)
67 E004 8E      txsp
68      ;
69 E005 1E00    ldx      #0
70      Clr0Page:
71 E007 C400    lda      #0
72 E009 FB      sta      {X}+
73 E00A 5EC0    cmpx     #STACK+1      ;0-Page RAM Clear(00-BFh)
74 E00C 70F9    bne      Clr0Page
75      ;
76 E00E E400C0  ldm      RA,#0000_0000B
77 E011 E4FCC1  ldm      RAO,#1111_1100B ;O RA7~2(Open,Output)
78      ;I RA1(PS2 Mouse CLK Input(Initial))
79      ;I RA0(PS2 Mouse Data Input(Initial))
80 E014 E400CA  ldm      RAFUNC,#0      ;Reset Value(RA Port I/O use)
81 E017 E41EEC  ldm      CKCTRL,#0001_1110B ;Initial(-001_0111B),BITCLK=1/(8Mhz/512)=64us
82 E01A E488ED  ldm      WDR,#1000_1000B ;WDT(=64us*256*8=131.072ms)
83      ;No_useei
84      TrnsmitDataToPC:
85 E01D E4AA00  ldm      MSEDATA,#0AAh ;Mouse Diagnostics Success Tx Value
86 E020 3B43E0  call     !XmitMSE1ByteToPC
87 E023 3B16E1  call     !Delay500us
88 E026 E40000  ldm      MSEDATA,#00h ;Mouse ID
89 E029 3B43E0  call     !XmitMSE1ByteToPC
90 E02C 3B16E1  call     !delay500us
91      MSECCommandFromPC:
92 E02F 33C00A  bbc     MSESCL,MSECCommandEnd ;When MouseCLK(RA.1)=1 & MouseDATA(RA.0)=0,
93 E032 03C007  bbs     MSESDA,MSECCommandEnd ;The command from PC is exist.
94 E035 3BB6E0  call     !GetMSE1byteFromPC ;Stored MouseDataFromPC to MSECCommand
95 E038 C501    lda     MSECCommand
96 E03A E502    sta     COMMANDBUF ;Execute as Command
97      MSECCommandEnd:
98 E03C E488ED  ldm     WDR,#1000_1000B ;WDT(=64us*256*8=131.072ms)
99 E03F B1E5    clr1   BITIF
100 E041 2FF9   bra     MSECCommandEnd
101      ;
102      ;=====;
103      ;= Sub Routine                                     =;
104      ;=====;
105      ;-----;
106      ;- Trnsmit Mouse_Data To Personal Computer         -;
107      ;-----;
108      XmitMSE1ByteToPC: ;1machine(cpu) cycle:0.25us@8Mhz
109 E043 C500    lda     MSEDATA
110 E045 33C0FD  bbc     MSESCL,$ ;R0[R/W],R0DD[W,Write Only]
111 E048 13C0FD  bbc     MSESDA,$
112 E04B 21C0    set1   MSESCL
113 E04D 01C0    set1   MSESDA
114 E04F E4FFC1  ldm     RAO,#OUTPUT_MODE ;[w] Switch output mode
115      ;di ;MSESCL=RA.1,MSESDA=RA.0
116 E052 E40803  ldm     BITCNT,#8
117 E055 E40104  ldm     PARITY,#1 ;odd
118 E058 11C0    clr1   MSESDA ;I2C Start Bit("0")
119 E05A 21C0    set1   MSESCL
120      ;*** Time Fix ***

```

```

121 E05C 3B4AE1      call    !Delay19us    ;76cyc(19us,Total=80cycle(20us))
122 E05F 31C0        clr1    MSESCL        ;4cyc(1us)
123 E061 3B22E1      call    !Delay39us    ;156 cycle(39us,Total=160cycle(40us))
124 E064 21C0        set1    MSESCL        ;4cycle(1us,total=160cycle(40us))
125 E066 3B52E1      call    !Delay14us    ;56cycle(14us)
126
127 E069 68          XmitMSE8Bits:
128 E06A EBC010      stc     MSESDA        ;6cyc
129 E06D 5005        bcc     MSEParityAddZero ;2(No),4(Yes)
130 E06F 8904        inc     PARITY        ;4cyc
131 E071 FF          nop
132 E072 2F04        bra     MSEParityAddOne ;4cyc
133
134 E074 FF          MSEParityAddZero:
135 E075 FF          nop          ;2cyc
136 E076 FF          nop          ;2cyc
137 E077 FF          nop          ;2cyc
138
139 E078 21C0        MSEParityAddOne:
140 E07A 3B4AE1      set1    MSESCL        ;4cyc
141 E07D 31C0        call    !Delay19us    ;76cyc(19us,Total=80cycle(20us))
142 E07F 3B22E1      clr1    MSESCL        ;4cyc(1us)
143 E082 21C0        call    !Delay39us    ;156 cycle(39us,Total=160cycle(40us))
144 E084 3B50E1      set1    MSESCL        ;4cycle(1us,total=160cycle(40us))
145 E087 AC03DF      call    !Delay15us    ;61cyc(15.25us)
146
147 E08A 6904        dbne   bitcnt,XmitMSE8bits ;5(No),7(Yes)
148
149 E08C EBC010      ;---
150 E091 3B4AE1      ror     PARITY        ;4cyc,to check odd or even(if data is odd then parity is 0)
151 E094 31C0        stc     MSESDA        ;6cyc,if C=1(data is odd) then set KBDDATALINE
152 E096 3B22E1      set1    MSESCL        ;4cyc,
153 E099 21C0        call    !Delay19us    ;76cyc(19us,Total=80cycle(20us))
154 E09B 3B4CE1      clr1    MSESCL        ;4cyc(1us)
155
156 E09E 01C0        call    !Delay39us    ;156 cycle(39us,Total=160cycle(40us))
157 E0A0 21C0        set1    MSESCL        ;4cyc,
158 E0A2 3B4AE1      set1    MSESDA        ;4cyc, I2C Stop Bit("1")
159 E0A5 31C0        call    !Delay19us    ;76cyc(19us,Total=80cycle(20us))
160 E0A7 3B22E1      clr1    MSESCL        ;4cyc,
161 E0AA 21C0        call    !Delay39us    ;156 cycle(39us,Total=160cycle(40us))
162
163
164 E0AC E4FCC1      set1    MSESCL        ;4cyc
165
166 E0AF 3B22E1      ;*** Time Fix ***
167 E0B2 33C0FD      ;ei
168 E0B5 6F          ldm     RAO,#INPUT_MODE ;[w] Switch input mode
169
170
171
172
173 E0B6 33C0FD      ;MSESCL=RA.1,MSESDA=RA.0
174 E0B9 03C0FD      call    !Delay39us    ;wait Ack(clk line)
175 E0BC 21C0        bbc     MSESCL,$      ;switched to High after 80us(Low)
176 E0BE E4FEC1      ret
177 E0C1 E40803      ;R0[R/W],RODD[W,Write Only]
178 E0C4 E40104      ;-----
179 E0C7 C400        ;- Get PC_Command_For_Mouse(Data) From PC and Store to RCVMSADATA-;
180 E0C9 E40001      ;-----
181
182 E0CC 31C0        GetMSElbyteFromPC:
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

183 E0CE 3B24E1      call    !Delay38us    ;152cyc
184 E0D1 FF          nop                    ;2cyc
185                GetMSE8bits:
186 E0D2 FF          nop                    ;2cyc
187 E0D3 21C0        set1    MSESCL        ;4cyc
188 E0D5 CBC000      ldc     MSESDA        ;4cyc.(Bit0~Bit7)
189 E0D8 68          ror     A              ;2cyc.
190 E0D9 3B26E1      call    !Delay37us    ;148cyc
191 E0DC FF          nop                    ;2cyc.
192 E0DD 31C0        clr1    MSESCL        ;4cyc
193 E0DF 3B26E1      call    !Delay37us    ;147cyc.(148cyc)
194 E0E2 AC03ED      dbne   BITCNT,GetMSE8bits ;5cyc(No)/7cyc(Yes)
195 E0E5 E501        sta     MSECMMAND     ;4cyc.
196 E0E7 21C0        set1    MSESCL        ;4cyc.Parity Bit(10th Clk)
197 E0E9 3B22E1      call    !Delay39us    ;156cyc.9th_bit
198 E0EC 31C0        clr1    MSESCL        ;4cyc.
199 E0EE 3B22E1      call    !Delay39us    ;156cyc.
200 E0F1 21C0        set1    MSESCL        ;4cyc.
201 E0F3 3B4CE1      call    !Delay18us    ;72cyc.,10th_bit
202 E0F6 11C0        clr1    MSESDA        ;4cyc.Low Setting
203 E0F8 E4FCC1      ldm     RAI0,#INPUT_MODE ;5cyc.Change Output(Clk,Data)
204 E0FB 11C0        clr1    MSESDA        ;4cyc.Low Setting
205 E0FD 3B4CE1      call    !Delay18us    ;71cyc.(72cyc)
206 E100 31C0        clr1    MSESCL        ;4cyc.
207 E102 3B22E1      call    !Delay39us    ;156cyc.
208 E105 21C0        set1    MSESCL        ;4cyc.
209 E107 3B4AE1      call    !Delay19us    ;76us
210 E10A 01C0        set1    MSESDA        ;4us
211                ;*** Time Fix
212 E10C E4FCC1      ldm     RAI0,#INPUT_MODE
213 E10F 3B4AE1      call    !Delay19us
214 E112 33C0FD      bbc     MSESCL,$
215 E115 6F          ret
216                ;-----;
217                ;- Time Delay Routine -;
218                ;-----;
219                Delay500us:
220 E116 E41EEC      ldm     CKCTLR,#0001_1110B ;Bit6(WAKEUP),Bit5(PCWDT),Bit4(WDTON)
221                ;Bit3(BTCL),B210(110b=Fxin/512=64us)
222 E119 FF          nop                    ;64us*256=16.384mS at 8MHZ
223 E11A FF          nop
224                Judge500us:
225 E11B C5EC        lda     BTR           ;64us*7=448us
226 E11D 4407        cmp     #07
227                ble Judge500us
228 E11F 50FA        @ bcc     Judge500us
229 E121 6F          ret
230                ;-----;
231                Delay39us:
232 E122 FF          nop                    ;(39us)
233 E123 FF          nop
234                Delay38us:
235 E124 FF          nop                    ;(38us)
236 E125 FF          nop
237                Delay37us:
238 E126 FF          nop                    ;(37us)
239 E127 FF          nop
240 E128 FF          nop                    ;(36us)
241 E129 FF          nop
242 E12A FF          nop                    ;(35us)
243 E12B FF          nop
244 E12C FF          nop                    ;(34us)

```

```

245 E12D FF          nop
246 E12E FF          nop                ; (33us)
247 E12F FF          nop
248 E130 FF          nop                ; (32us)
249 E131 FF          nop
250 E132 FF          nop                ; (31us)
251 E133 FF          nop
252 E134 FF          nop                ; (30us)
253 E135 FF          nop
254 E136 FF          nop                ; (29us)
255 E137 FF          nop
256 E138 FF          nop                ; (28us)
257 E139 FF          nop
258 E13A FF          nop                ; (27us)
259 E13B FF          nop
260 E13C FF          nop                ; (26us)
261 E13D FF          nop
262 E13E FF          nop                ; (25us)
263 E13F FF          nop
264 E140 FF          nop                ; (24us)
265 E141 FF          nop
266 E142 FF          nop                ; (23us)
267 E143 FF          nop
268 E144 FF          nop                ; (22us)
269 E145 FF          nop
270 E146 FF          nop                ; (21us)
271 E147 FF          nop
272 E148 FF          nop                ; (20us)
273 E149 FF          nop
274                Delay19us:
275 E14A FF          nop                ; (19us)
276 E14B FF          nop
277                Delay18us:
278 E14C FF          nop                ; (18us)
279 E14D FF          nop
280 E14E FF          nop                ; (16us)
281 E14F FF          nop
282                Delay15us:
283 E150 FF          nop                ; (15us)
284 E151 FF          nop
285                Delay14us:
286 E152 FF          nop                ; (14us)
287 E153 FF          nop
288 E154 FF          nop                ; (13us, call=8cycle)
289 E155 FF          nop
290 E156 FF          nop                ; (12us)
291 E157 FF          nop
292 E158 FF          nop                ; (11us)
293 E159 FF          nop
294 E15A FF          nop                ; (10us)
295 E15B FF          nop
296 E15C FF          nop                ; (9us)
297 E15D FF          nop
298 E15E FF          nop                ; (8us)
299 E15F FF          nop
300 E160 FF          nop                ; (7us)
301 E161 FF          nop
302 E162 FF          nop                ; (6us)
303 E163 FF          nop
304 E164 FF          nop                ; (5us)
305 E165 FF          nop
306 E166 FF          nop                ; 8cyc (4us, call=8cycle) (add+1)

```

```
307 E167 6F          ret          ; 5cyc  
308          ;-----;  
309          END
```

-- 0 Error(s) --

--- Total Machine Code : 362 Bytes ---

NOTE: