

```

1  typedef volatile union {
2      struct {
3          unsigned char L;
4          unsigned char H;
5      };
6      unsigned int      I;
7  }IM;
8
9  typedef volatile union {
10     struct {
11         unsigned char m1;
12         unsigned char m2;
13         unsigned char m3;
14         unsigned char m4;
15     };
16     unsigned long      L;
17 }LM;
18
19 typedef union
20 {
21     struct {
22         unsigned      S1:1;
23         unsigned      S0:1;
24         unsigned      H1:1;
25         unsigned      H0:1;
26         unsigned      PIR:1;
27     }c;
28     unsigned char      event;
29 }ev;
30
31 typedef union
32 {
33     struct {
34         unsigned      S:1;
35         unsigned      H:1;
36         unsigned      CO:1;
37         unsigned      PIR:1;
38     }c;
39     unsigned char      event;
40 }ev1;
41
42 typedef union {
43     struct {
44         unsigned char      st1;
45         unsigned char      st[15];
46         unsigned long      m1;
47     }c;
48     unsigned char      data[20];
49 }rxxx;
50
51 typedef union
52 {
53     struct {
54         unsigned int      g_uart0_tx_count;
55         unsigned char      TX_DATA[20];
56     }c;
57     unsigned char      data[22];
58 }txxx;
59
60 typedef union
61 {
62     struct {
63         unsigned int      H_ADCYZL,S_ADCYZL,S_ADCYZ;
64         unsigned int      H_BDAT1,H_BDAT2,H_BDAT3,H_BDAT4,H_PGZ;
65         unsigned int      wsb_time;
66         unsigned int      RESULT1,RESULT2,RESULT3,RESULT4,S_PGZ;
67         unsigned int      SMOKE_VALUE,SMOKE_NO_ALARM_VALUE,SMOKE_NEWZERO_XDD;
68         unsigned int      SMOKE_NEWZERO_VALUE,SMOKE_ALARM_VALUE;
69         unsigned int      MCNT1,MCNT2;
70         unsigned char      SGZJSQ;
71         unsigned char      HGZJSQ;

```

```

72     }c;
73     unsigned char data[45];
74 }dttt;
75 #define LED      P1_bit.no0
76 #define RX       P1_bit.no1
77 #define TX       P1_bit.no2
78 #define INT      P1_bit.no3
79 #define PIRDY    P1_bit.no4
80 #define YWCY     P2_bit.no0
81 #define WDCY     P2_bit.no1
82 #define WDY      P2_bit.no2
83 #define DYC      P2_bit.no3
84 #define FLED     P4_bit.no1
85 #define OPK      P4_bit.no2
86 #define PIRDY_b  P6_bit.no0
87 #define LED_B    P6_bit.no1
88 #define PIRC     P13_bit.no7
89 struct {
90     unsigned    JKJC:1;
91     unsigned    Y_YWB:1;
92     unsigned    Y_HWB:1;
93     unsigned    S_STH0:1;
94     unsigned    H_HTH0:1;
95     unsigned    TXWC:1;
96     unsigned    ROMTX:1;
97     unsigned    PIR_30S_E:1;
98     unsigned    TIME_W:1;
99     unsigned    HEAT_GZ : 1;
100    unsigned    YBTXQL:1;
101    unsigned    int_200ms:1;
102 }flag;
103 union {
104     struct {
105         unsigned    SDIRB:1;
106         unsigned    SMOKE_GZ : 1;
107         unsigned    EER_SDGZ:1;
108         unsigned    EEW_SDGZ:1;
109     }c;
110     unsigned char IG;
111 }gz;
112 unsigned char stop_t;
113 unsigned char TIME_t;
114 unsigned char rx_bz;
115 unsigned char txint_count;
116 unsigned char PIR_30S,PIR_CONT;
117 IM      SENU_Y;
118 unsigned char SENU_B;
119 IM      SENU_S;
120 unsigned char STH1_Z;
121 unsigned char STH0_Z;
122 unsigned char HTH1_Z;
123 unsigned char HTH0_Z;
124 unsigned char SVAL_Z;
125 unsigned char HVAL_Z;
126 unsigned char DECL_Z;
127 unsigned char PIRS_Z;
128 unsigned char PIRE_Z;
129 unsigned char ILED_Z;
130 unsigned char DETY_Z;
131 unsigned char STAT_Z;
132 unsigned char STAT_Z1;
133 unsigned char OCS1_Z;
134 unsigned char OPEC_Z;
135 LM      IDNU_Z;
136 unsigned char DRSO_Z;
137 unsigned char CXTO_Z;
138 unsigned char ED01_Z;
139 unsigned char ED02_Z;
140 unsigned char ED04_Z;
141 unsigned char DBID_Z;
142 unsigned char DBID_LSZ;

```

```

143 volatile unsigned char * gp_uart0_tx_address;
144 volatile const unsigned char * gp_uart0_tx_address_r;
145 unsigned char rx_data;
146 unsigned char RW_DataFla_buff[22];
147 unsigned char TX_TIME;
148 unsigned char err_type;
149 unsigned char js_data;
150 IM rxd_jcend;
151 IM sz_mos;
152 LM js_data1;
153 ev ev_int;
154 ev ev_bj;
155 ev ev_ls;
156 rxxx rx;
157 txxx tx;
158 dttt dd;
159 unsigned int gzd05,gzd1,gzd2,gzd3,gzd4,gzd5,gzd6;
160 unsigned char h_4s,h_20s,h_40s,h_60s,ws_time,ws_ext;
161 #define H_ADCYZL dd.c.H_ADCYZL
162 #define S_ADCYZL dd.c.S_ADCYZL
163 #define S_ADCYZ dd.c.S_ADCYZ
164 #define H_BDAT1 dd.c.H_BDAT1
165 #define H_BDAT2 dd.c.H_BDAT2
166 #define H_BDAT3 dd.c.H_BDAT3
167 #define H_BDAT4 dd.c.H_BDAT4
168 #define H_PGZ dd.c.H_PGZ
169 #define wsb_time dd.c.wsb_time
170 #define RESULT1 dd.c.RESULT1
171 #define RESULT2 dd.c.RESULT2
172 #define RESULT3 dd.c.RESULT3
173 #define RESULT4 dd.c.RESULT4
174 #define S_PGZ dd.c.S_PGZ
175 #define SMOKE_VALUE dd.c.SMOKE_VALUE
176 #define SMOKE_NO_ALARM_VALUE dd.c.SMOKE_NO_ALARM_VALUE
177 #define SMOKE_NEWZERO_XDD dd.c.SMOKE_NEWZERO_XDD
178 #define SMOKE_NEWZERO_VALUE dd.c.SMOKE_NEWZERO_VALUE
179 #define SMOKE_ALARM_VALUE dd.c.SMOKE_ALARM_VALUE
180 #define MCNT1 dd.c.MCNT1
181 #define MCNT2 dd.c.MCNT2
182 #define SGZJSQ dd.c.SGZJSQ
183 #define HGZJSQ dd.c.HGZJSQ
184
185 const unsigned char I9SI[] = {"01.00.00,15/10/19"};
186 const unsigned char ERROR[] = {"ERROR"};
187 const unsigned char OK[] = {"OK"};
188 #pragma interrupt r_uart0_interrupt_send(vect=INTST0)
189 #pragma interrupt r_uart0_interrupt_receive(vect=INTSR0)
190 #pragma interrupt r_it_interrupt(vect=INTIT)
191 #pragma interrupt r_intc2_interrupt(vect=INTP2)
192 #pragma interrupt r_tau0_channel0_interrupt(vect=INTTM00)
193
194 void main(void);
195 void R_Systeminit(void);
196 void R_PORT_Create(void);
197 void R_ADC_Create(void);
198 void R_CGC_Create(void);
199 void R_IT_Create(void);
200 void R_WDT_Create(void);
201 void R_INTC_Create(void);
202 void R_SAU0_Create(void);
203 void R_UART0_Create(void);
204 void R_TAU0_Create(void);
205 void R_IT_Start(void);
206 void R_INTC2_Start(void);
207 void R_UART0tx_Start(void);
208 void R_TAU0_Channel0_Start(void);
209 void R_INTC2_Stop(void);
210 void R_FLASH(void);
211 void STOP_TIME(void);
212 void HEAT_CY(void);
213 void HEAT_V_GX(void);

```

```

214 unsigned int HEAT_PROCESS(void);
215 unsigned char HEAT_MN(unsigned int wdz);
216 void H_ALARM_PD(void);
217 void SMOKE_CY(void);
218 void SMOKE_V_GX(void);
219 unsigned int DATA_PROCESS(void);
220 void SMOKE_ZERO(void);
221 unsigned int SMOKE_MN(void);
222 void CHECK_ALARM(void);
223 void PIR_CY(void);
224 void SD_DATACS(void);
225 void SD_SAMPLE_PJ(void);
226 void STAS_DATA(void);
227 void DELAY2MS(void);
228 void YBF_TX20MS(void);
229 void SOTPYAS(unsigned char ss_d1);
230 void RX_EEROR(void);
231 void HEAT_WS(void);
232 unsigned char RXDATAJS1(void);
233 unsigned char RXDATAJS2(void);
234 unsigned char RXDATAJS3(void);
235 unsigned char RXDATAJS4(void);
236 void RX_DATAACX(void);
237 void DBID_ER(void);
238 void R_UART0_Send(void);
239
240 void main(void)
241 {
242     __DI();
243     R_Systeminit();
244     R_IT_Start();
245     __EI();
246     SOTPYAS(10);
247     R_FLASH();
248     SD_SAMPLE_PJ();
249     SD_DATACS();
250     R_INTC2_Start();
251     while (1)
252     {
253         STOP_TIME();
254         if(OPEC_Z==1)
255         {
256             if((sz_mos.L&0X01)&&(ED01_Z==1))
257             {
258                 if(flag.TIME_W==1 || flag.Y_YWB==1)
259                 {
260                     SMOKE_CY();
261                     if(SGZJSQ==0)
262                     {
263                         SMOKE_V_GX();
264                         S_PGZ=DATA_PROCESS();
265                         if(flag.Y_YWB==0)
266                             SMOKE_ZERO();
267                     }
268                     SVAL_Z=SMOKE_MN();
269                     CHECK_ALARM();
270                 }
271             }
272             if((sz_mos.L&0X02)&&(ED02_Z==1))
273             {
274                 HEAT_CY();
275                 if(HGZJSQ==0)
276                 {
277                     HEAT_V_GX();
278                     H_PGZ=HEAT_PROCESS();
279                     if(sz_mos.L&0X04)
280                         HEAT_WS();
281                     else
282                         HVAL_Z=HEAT_MN(H_PGZ);
283                 }
284                 H_ALARM_PD();

```

```

285     }
286     STAS_DATA();
287     YBF_TX20MS();
288     }
289 }
290 }
291 void SOTPYAS(unsigned char ss_d1)
292 {
293     while(ss_d1--)
294     {
295         __stop();
296         WDTE=0XAC;
297     }
298 }
299 void R_FLASH(void)
300 {
301     IM d1;
302     R_pdl(&RW_DataFla_buff[0]);
303     d1.H=RW_DataFla_buff[0];
304     d1.L=RW_DataFla_buff[1];
305     SMOKE_NEWZERO_VALUE=d1.I;
306     d1.H=RW_DataFla_buff[2];
307     d1.L=RW_DataFla_buff[3];
308     SMOKE_ALARM_VALUE=d1.I;
309     d1.H=RW_DataFla_buff[4];
310     d1.L=RW_DataFla_buff[5];
311     sz_mos.I=d1.I;
312     SENU_S.L=RW_DataFla_buff[6];
313     SENU_S.H=RW_DataFla_buff[7];
314     SENU_B=RW_DataFla_buff[8];
315     d1.L=RW_DataFla_buff[9];
316     d1.H=d1.L&0X0F;
317     d1.L=d1.L&0XF0;
318     d1.L>>=4;
319     SENU_Y.H=d1.L+17;
320     SENU_Y.L=d1.H;
321     DBID_LSZ=RW_DataFla_buff[10];
322     DBID_Z=DBID_LSZ;
323     if(sz_mos.L>32)
324     {gz.c.EEW_SDGZ=1;
325     sz_mos.I=0x000b;
326     }
327 }
328 void SD_DATACS(void)
329 {
330     unsigned int sd1;
331     DRSO_Z=1;
332     if(RESF&0X10)
333     DRSO_Z=2;
334     if(RESF&0X01)
335     DRSO_Z=3;
336     if(sz_mos.L&0X01)
337     {
338         if((500<SMOKE_NEWZERO_VALUE)|| (SMOKE_NEWZERO_VALUE<20))
339         {
340             gz.c.EER_SDGZ=1;
341             SMOKE_NEWZERO_VALUE=60;
342         }
343         if((600<SMOKE_ALARM_VALUE)|| (SMOKE_ALARM_VALUE<20))
344         {
345             gz.c.EER_SDGZ=1;
346             SMOKE_ALARM_VALUE=30;
347         }
348         SMOKE_NO_ALARM_VALUE=SMOKE_ALARM_VALUE>>2;
349         SMOKE_NEWZERO_XDD=800;
350         gzd05=SMOKE_NEWZERO_VALUE;
351         gzd1=gzd05;
352         gzd2=gzd05;
353         gzd3=gzd05;
354         gzd4=gzd05;
355         gzd5=gzd05;

```

```

356     gzd6=gzd05;
357 }
358     STH1_Z=45;
359     STH0_Z=0;
360     HTH1_Z=65;
361     HTH0_Z=0;
362     SVAL_Z=23;
363     HVAL_Z=23;
364     OPEC_Z=1;
365     PIRE_Z=0;
366     ED01_Z=1;
367     ED02_Z=1;
368     ED04_Z=1;
369     sdl=sz_mos.L;
370     sdl&=~0x04;
371     DETY_Z=sdl;
372     DECL_Z=1;
373     if(sz_mos.L&0x02)
374     {
375         if(sz_mos.L&0x04)
376         {
377             HTH1_Z=45;
378             DECL_Z=1;
379         }
380         else
381             DECL_Z=2;
382     }
383 }
384 void SD_SAMPLE_PJ(void)
385 {
386     unsigned char dl;
387     for(dl=6;dl!=0;dl--)
388     {
389         SOTPYAS(8);
390         if(sz_mos.L&0x01)
391         {
392             SMOKE_CY();
393             if(SGZJSQ==0)
394             {
395                 SMOKE_V_GX();
396                 S_PGZ=DATA_PROCESS();
397             }
398         }
399         if(sz_mos.L&0x02)
400         {
401             HEAT_CY();
402             if(HGZJSQ==0)
403             {
404                 HEAT_V_GX();
405                 H_PGZ=HEAT_PROCESS();
406             }
407         }
408     }
409     h_4s=H_PGZ>>2;
410     h_20s=h_4s;
411     h_40s=h_4s;
412     h_60s=h_4s;
413 }
414 void STOP_TIME(void)
415 {
416
417     flag.TIME_W=0;
418     if(flag.JKJC==1)
419     {
420         TIME_t+=10;
421         stop_t=10;
422     }
423     else
424     {
425         stop_t=20;
426         TIME_t+=20;

```

```

427     }
428     if(TIME_t>=20)
429     {
430         TIME_t=TIME_t-20;
431         flag.TIME_W=1;
432     }
433     while(stop_t!=0)
434     {
435         if(rx_bz==0&&flag.TXWC==0 )
436         {
437             DBID_ER();
438             __stop();
439             __nop();
440         }
441         PIR_CY();
442         WDTE=0XAC;
443     }
444 }
445 void DBID_ER(void)
446 {
447     if(DBID_Z==DBID_LSZ)
448     return;
449     W_pdlDB(&DBID_LSZ);
450     DBID_Z=DBID_LSZ;
451 }
452 void SMOKE_CY(void)
453 {
454     unsigned char D1,gbz=0;
455     OPK=1;
456     DELAY2MS();
457     ADS = 0X00;
458     ADCE=1;
459     ADIF=0;
460     for(D1=10;D1!=0;D1--);
461     ADCS = 1U;
462     while(ADIF==0);
463     S_ADCYZL=ADCR>>6;
464     ADIF=0;
465     if(S_ADCYZL>300)
466     gbz=1;
467     __DI();
468     FLED=1;
469     for(D1=9;D1!=0;D1--);
470     ADCS = 1U;
471     for(D1=4;D1!=0;D1--);
472     FLED=0;
473     __EI();
474     while(ADIF==0);
475     S_ADCYZ=ADCR>>6;
476     ADCE=0;
477     OPK=0;
478     if(S_ADCYZ>=S_ADCYZL)
479     {SMOKE_VALUE=S_ADCYZ-S_ADCYZL;
480     if(SMOKE_VALUE<5)
481     gbz=2;
482     }
483     else
484     gbz=3;
485     if(gbz!=0)
486     {SMOKE_VALUE=gbz;
487     SGZJSQ++;
488     if(SGZJSQ>9)
489     gz.c.SMOKE_GZ=1;
490     }else
491     {
492     gz.c.SMOKE_GZ=0;
493     SGZJSQ=0;
494     }
495 }
496 void HEAT_CY(void)
497 {

```

```

498     unsigned char D1,gdz=0;
499     WDY=1;
500     ADS = 0X01;
501     ADCE=1;
502     ADIF=0;
503     for(D1=30;D1!=0;D1--);
504     ADCS = 1U;
505     while(ADIF==0);
506     H_ADCYZL=ADCR>>6;
507     ADIF=0;
508     ADCE=0;
509     WDY=0;
510     if(H_ADCYZL<5)
511         gdz=1;
512     if(H_ADCYZL>1000)
513         gdz=2;
514     if(gdz!=0)
515     {
516         HGZJSQ++;
517         if(HGZJSQ>9)
518             flag.HEAT_GZ=1;
519         }else
520     {
521         flag.HEAT_GZ=0;
522         HGZJSQ=0;
523     }
524 }
525 void SMOKE_V_GX(void)
526 {
527     RESULT4 =RESULT3;
528     RESULT3 =RESULT2;
529     RESULT2 =RESULT1;
530     RESULT1 =SMOKE_VALUE;
531 }
532 void HEAT_V_GX(void)
533 {
534     H_BDAT4 =H_BDAT3;
535     H_BDAT3 =H_BDAT2;
536     H_BDAT2 =H_BDAT1;
537     H_BDAT1 =H_ADCYZL;
538 }
539 unsigned int DATA_PROCESS(void)
540 {
541     unsigned int pgzd;
542     pgzd=RESULT1;
543     pgzd+=RESULT2;
544     pgzd+=RESULT3;
545     pgzd+=RESULT4;
546     return pgzd/=4;
547 }
548 unsigned int HEAT_PROCESS(void)
549 {
550     unsigned int pgzd;
551     pgzd=H_BDAT1;
552     pgzd+=H_BDAT2;
553     pgzd+=H_BDAT3;
554     pgzd+=H_BDAT4;
555     return pgzd/=4;
556 }
557 //-----
558 //Response to slowly developing fires
559 void SMOKE_ZERO(void)
560 {
561     static unsigned int ze_d1,ze_d2,ze_d3,ze_dd;
562     MCNT1++;
563     if(MCNT1<450)
564         return;
565     MCNT1=0;
566     MCNT2++;
567     ze_d1=SMOKE_ALARM_VALUE>>1;
568     ze_d2=ze_d1>>1;

```



```

569     ze_d3=ze_d2>>1;
570
571     if(S_PGZ>gzd05)
572     ze_dd=S_PGZ-gzd05;
573     else
574     ze_dd=gzd05-S_PGZ;
575     if(ze_d1>ze_dd)
576     goto SMOKE_ZERO_3;
577     if(S_PGZ>gzd1)
578     ze_dd=S_PGZ-gzd1;
579     else
580     ze_dd=gzd1-S_PGZ;
581     if(ze_d2<ze_dd)
582     goto SMOKE_ZERO_5;
583 SMOKE_ZERO_3:
584     if(S_PGZ>gzd6)
585     ze_dd=S_PGZ-gzd6;
586     else
587     ze_dd=gzd6-S_PGZ;
588     if(ze_d3<ze_dd)
589     goto SMOKE_ZERO_5;
590     SMOKE_NEWZERO_VALUE=S_PGZ;
591     if(SMOKE_NEWZERO_VALUE>SMOKE_NEWZERO_XDD)
592     gz.c.SDIRB=1;
593     else
594     gz.c.SDIRB=0;
595 SMOKE_ZERO_5:
596     if(!(MCNT2&0x01))
597     {
598     gzd6=gzd5;
599     gzd5=gzd4;
600     gzd4=gzd3;
601     gzd3=gzd2;
602     gzd2=gzd1;
603     }
604     gzd1=gzd05;
605     gzd05=S_PGZ;
606 }
607 unsigned int SMOKE_MN(void)
608 {
609     unsigned int sm_d1,sm_d2;
610     sm_d1=S_PGZ<<5;
611     sm_d2=SMOKE_ALARM_VALUE*23;
612     sm_d1+=sm_d2;
613     sm_d2=SMOKE_NEWZERO_VALUE<<5;
614     if(sm_d1>sm_d2)
615     {
616     sm_d1-=sm_d2;
617     sm_d1/=SMOKE_ALARM_VALUE;
618     if(sm_d1<=9)
619     sm_d1=9;
620     if(sm_d1>=99)
621     sm_d1=99;
622     }
623     else
624     sm_d1=9;
625     if(gz.IG!=0&&sm_d1<54)
626     {
627     sm_d2=0;
628     if(gz.c.SMOKE_GZ==1)
629     sm_d2|=1;
630     if(gz.c.SDIRB==1)
631     sm_d2|=2;
632     if(gz.c.EER_SDGZ==1)
633     sm_d2=4;
634     if(gz.c.EEW_SDGZ==1)
635     sm_d2=8;
636     return sm_d2;
637     }
638     return sm_d1;
639 }

```

```

640 void HEAT_WS(void)
641 {
642     unsigned char wd1, wd2;
643     if(flag.TIME_W==0)
644         return;
645     ws_time++;
646     wd2=23;
647     if(ws_time>=5)
648     {
649         ws_time=0;
650         h_60s=h_40s;
651         h_40s=h_20s;
652         h_20s=h_4s;
653     }
654     h_4s=H_ADCYZL>>2;
655     if(h_60s>h_4s)
656     {
657         wd1=h_60s-h_4s;
658         if(wd1>=2)
659         {
660             ws_ext=0;
661             wsb_time+=wd1;
662             if(wd1>=8)
663                 wsb_time+=wd1;
664             if(wd1>=16)
665                 wsb_time+=wd1;
666         }
667         else
668             ws_ext++;
669     }
670     else
671         ws_ext++;
672     if(ws_ext>=3)
673     {
674         wsb_time=0;
675         ws_ext=10;
676     }
677     if(wsb_time>375)
678         wd2=45;
679     if(wsb_time>450)
680         wd2=55;
681     if(H_PGZ<=382)
682         wd2=56;
683     if(flag.HEAT_GZ==1)
684         wd2=1;
685     HVAL_Z=wd2;
686 }
687 unsigned char HEAT_MN(unsigned int wdz)
688 {
689     unsigned int hm_d1,hm_d2;
690     hm_d1=0;
691     if(flag.HEAT_GZ==1)
692         return hm_d1=1;
693     hm_d2=wdz>>2;
694     if(hm_d2>=208)
695         return 9;
696
697     if(hm_d2>=206)
698     {
699         return 10;}
700
701     if(hm_d2>=187)
702     {
703         hm_d1=hm_d2-187;
704         hm_d1<=<3;
705         hm_d1=hm_d1/18;
706         hm_d1=19-hm_d1;
707         return hm_d1;}
708
709     if(hm_d2>=165)
710     {

```

```
711     hm_d1=hm_d2-165;
712     hm_d1<=<=3;
713     hm_d1=hm_d1/21;
714     hm_d1=28-hm_d1;
715     return hm_d1;}
716
717     if(hm_d2>=142)
718     {
719         hm_d1=hm_d2-142;
720         hm_d1<=<=3;
721         hm_d1=hm_d1/22;
722         hm_d1=37-hm_d1;
723         return hm_d1;}
724
725     if(hm_d2>=119)
726     {
727         hm_d1=hm_d2-119;
728         hm_d1<=<=3;
729         hm_d1=hm_d1/22;
730         hm_d1=46-hm_d1;
731         return hm_d1;}
732
733     if(hm_d2>=98)
734     {
735         hm_d1=hm_d2-98;
736         hm_d1<=<=3;
737         hm_d1=hm_d1/20;
738         hm_d1=55-hm_d1;
739         return hm_d1;}
740
741     if(hm_d2>=79)
742     {
743         hm_d1=hm_d2-79;
744         hm_d1<=<=3;
745         hm_d1=hm_d1/18;
746         hm_d1=64-hm_d1;
747         return hm_d1;}
748
749     if(hm_d2>=62)
750     {
751         hm_d1=hm_d2-62;
752         hm_d1<=<=3;
753         hm_d1=hm_d1/16;
754         hm_d1=73-hm_d1;
755         return hm_d1;}
756
757     if(hm_d2>=51)
758     {
759         hm_d1=hm_d2-51;
760         hm_d1<=<=3;
761         hm_d1=hm_d1/10;
762         hm_d1=81-hm_d1;
763         return hm_d1;}
764
765     if(hm_d2>=41)
766     {
767         hm_d1=hm_d2-41;
768         hm_d1<=<=3;
769         hm_d1=hm_d1/9;
770         hm_d1=90-hm_d1;
771         return hm_d1;
772     }
773
774     if(hm_d2>=32)
775     {
776         hm_d1=hm_d2-32;
777         hm_d1<=<=3;
778         hm_d1=hm_d1/9;
779         hm_d1=99-hm_d1;
780         return hm_d1;
781     }
```

```

782
783     if(hm_d2<=31)
784         hm_d1=99;
785     if(wdz<=10)
786         hm_d1=1;
787     return hm_d1;
788 }
789 void CHECK_ALARM(void)
790 {
791     if(SVAL_Z>=STH1_Z)
792         flag.Y_YWB=1;
793     else
794         flag.Y_YWB=0;
795     if(SVAL_Z<=STH0_Z)
796         flag.S_STH0=1;
797     else
798         flag.S_STH0=0;
799 }
800 void H_ALARM_PD(void)
801 {
802     if(flag.HEAT_GZ==1)
803         HVAL_Z=1;
804     if(HVAL_Z>=HTH1_Z)
805         flag.Y_HWB=1;
806     else
807         flag.Y_HWB=0;
808     if(HVAL_Z<=HTH0_Z)
809         flag.H_HTH0=1;
810     else
811         flag.H_HTH0=0;
812 }
813 void PIR_CY(void)
814 {
815     if((sz_mos.L&0X08)&&(flag.int_200ms==1))
816     {
817         flag.int_200ms=0;
818         if(PIRE_Z==0||ED04_Z==0)
819         {
820             PIRS_Z=0;
821             flag.PIR_30S_E=1;
822             PIR_30S=0;
823             return;
824         }
825         if(flag.PIR_30S_E==1)
826         {
827             PIR_30S++;
828             if(PIR_30S>100)
829             {
830                 flag.PIR_30S_E=0;
831                 PIR_30S=0;}
832             PIRS_Z=0;
833             PIR_CONT=0;
834             return;
835         }
836         if(PIRC==1)
837         {
838             PIR_CONT++;
839             if(PIR_CONT>=3)
840                 PIRS_Z=1;
841         }
842         else
843             PIR_CONT=0;
844     }
845 }
846 void YBF_TX20MS(void)
847 {
848     if(STAT_Z==1&&flag.YBTXQL==0)
849     {
850         txint_count=5;
851         do{
852             WDTE=0XAC;

```

```

853     if((flag.TXWC==0)&&(rx_bz==0))
854     {
855         R_INTC2_Stop();
856         if(txint_count!=0)
857             txint_count--;
858         R_TAU0_Channel0_Start();
859         TX_TIME=2;
860         flag.TXWC=1;
861         TX=0;
862     }
863     if(flag.YBTXQL==1)
864         txint_count=0;
865     }while(txint_count);
866 }
867 }
868 //
869 void STAS_DATA(void)
870 {
871     evl dl;
872     if(flag.Y_YWB==1 || flag.Y_HWB==1 || wsb_time!=0)
873         flag.JKJC=1;
874     else
875         flag.JKJC=0;
876     ev_ls.event=0;
877     if(flag.Y_YWB==1)
878         ev_ls.c.S1=1;
879     else
880         ev_int.c.S1=0;
881     if(flag.S_STH0==1)
882         ev_ls.c.S0=1;
883     else
884         ev_int.c.S0=0;
885     if(flag.Y_HWB==1)
886         ev_ls.c.H1=1;
887     else
888         ev_int.c.H1=0;
889     if(flag.H_HTH0==1)
890         ev_ls.c.H0=1;
891     else
892         ev_int.c.H0=0;
893     if(PIRS_Z==1)
894         ev_ls.c.PIR=1;
895     else
896         ev_int.c.PIR=0;
897     ev_bj.event=ev_ls.event;
898     if(ev_bj.c.S1==1&&ev_int.c.S1==1)
899         ev_ls.c.S1=0;
900     if(ev_bj.c.S0==1&&ev_int.c.S0==1)
901         ev_ls.c.S0=0;
902     if(ev_bj.c.H1==1&&ev_int.c.H1==1)
903         ev_ls.c.H1=0;
904     if(ev_bj.c.H0==1&&ev_int.c.H0==1)
905         ev_ls.c.H0=0;
906     if(ev_bj.c.PIR==1&&ev_int.c.PIR==1)
907         ev_ls.c.PIR=0;
908     if(ev_ls.event!=0)
909         flag.YBTXQL=0;
910     dl.event=0;
911     if(ev_bj.c.S1==1 || ev_bj.c.S0==1)
912         dl.c.S=1;
913     if(ev_bj.c.H1==1 || ev_bj.c.H0==1)
914         dl.c.H=1;
915     if(ev_bj.c.PIR==1)
916         dl.c.PIR=1;
917     STAT_Z1=dl.event;
918     if(ev_bj.event!=0)
919         STAT_Z=1;
920     else
921         STAT_Z=0;
922     if(gz.c.SMOKE_GZ==1)
923         {OCS1_Z=3;}

```

```

924     else
925     {
926         if(gz.c.SDIRB==1)
927             OCS1_Z=2;
928         else
929             OCS1_Z=0;
930     }
931 }
932
933 void R_Systeminit(void)
934 {
935     PIOR = 0x00U;
936     R_CGC_Create();
937     R_PORT_Create();
938     R_SAU0_Create();
939     R_ADC_Create();
940     R_WDT_Create();
941     R_IT_Create();
942     R_INTC_Create();
943     R_TAU0_Create();
944     IAWCTL = 0x00U;
945 }
946
947 void R_PORT_Create(void)
948 {
949     PIOR = 0x00;
950     PM1 = 0X0A;
951     PM2 = 0X03;
952     PM4 = 0X01;
953     PM6 = 0X00;
954     P1 = 0X00;
955     P2 = 0X00;
956     P4 = 0X00;
957     P6 = 0X00;
958     PIM1 = 0XA;
959     PMC1 = 0X00;
960     PMC4 = 0X00;
961     ADPC = 0X03;
962     PM1 = 0X0A;
963     PM2 = 0X03;
964     PM4 = 0X01;
965     PM6 = 0X00;
966     P6 = 0X00;
967 }
968 void R_ADC_Create(void)
969 {
970     ADCEN = 1U;
971     ADM0 = 0X00;
972     ADMK = 1U;
973     ADIF = 0U;
974     ADPR1 = 1U;
975     ADPR0 = 1U;
976     PM2 |= 0x01U;
977     ADM0 = 0X3C;
978     ADM1 = 0X20;
979     ADM2 = 0X00;
980     ADUL = 0XFF;
981     ADLL = 0X00;
982     ADS = 0X00;
983 }
984 void R_CGC_Create(void)
985 {
986     CMC = 0X00;
987     MSTOP = 1U;
988     MCM0 = 0U;
989     OSMC = 0X10;
990     HIOSTOP = 0U;
991 }
992 void R_INTC_Create(void)
993 {
994     PMK0 = 1U;

```

```

995     PIF0 = 0U;
996     PMK1 = 1U;
997     PIF1 = 0U;
998     PMK2 = 1U;
999     PIF2 = 0U;
1000    PMK3 = 1U;
1001    PIF3 = 0U;
1002    PPR12 = 1U;
1003    PPR02 = 1U;
1004    EGN0 = 0X04;
1005    PMC1 &= 0xF7U;
1006    PM1 |= 0x08U;
1007 }
1008 void R_INTC2_Start(void)
1009 {
1010     PIF2 = 0U;
1011     PMK2 = 0U;
1012 }
1013 void R_INTC2_Stop(void)
1014 {
1015     PMK2 = 1U;
1016     EGP0=0X00;
1017     PIF2 = 0U;
1018 }
1019 void R_IT_Create(void)
1020 {
1021     TMKAEN = 1U;
1022     ITMC = 0X0000;
1023     TMKAMK = 1U;
1024     TMKAIF = 0U;
1025     TMKAPR1 = 1U;
1026     TMKAPR0 = 1U;
1027     ITMC = 0X0BB7;
1028 }
1029 void R_IT_Start(void)
1030 {
1031     TMKAIF = 0U;
1032     TMKAMK = 0U;
1033     ITMC |= 0x8000;
1034 }
1035 void R_TAU0_Create(void)
1036 {
1037     TAU0EN = 1U;
1038     TPS0 = 0X0000;
1039     TT0 = 0X0A0F;
1040     TMMK00 = 1U;
1041     TMIF00 = 0U;
1042     TMMK01 = 1U;
1043     TMIF01 = 0U;
1044     TMMK01H = 1U;
1045     TMIF01H = 0U;
1046     TMMK02 = 1U;
1047     TMIF02 = 0U;
1048     TMMK03 = 1U;
1049     TMIF03 = 0U;
1050     TMMK03H = 1U;
1051     TMIF03H = 0U;
1052     TMPR100 = 1U;
1053     TMPR000 = 1U;
1054     TMR00 = 0X0000;
1055     TDR00 = 0X270F;
1056     TO0 &= ~0X0001;
1057     TOE0 &= ~0X0001;
1058 }
1059 void R_TAU0_Channel0_Start(void)
1060 {
1061     TMIF00 = 0U;
1062     TMMK00 = 0U;
1063     TS0 |= 0X001;
1064 }
1065 void R_SAU0_Create(void)

```

```

1066 {
1067     SAU0EN = 1U;
1068     __nop();
1069     __nop();
1070     __nop();
1071     __nop();
1072     SPS0 = 0X00;
1073     R_UART0_Create();
1074 }
1075 void R_UART0_Create(void)
1076 {
1077     ST0 |= 0X03;
1078     STMK0 = 1U;
1079     STIF0 = 0U;
1080     SRMK0 = 1U;
1081     SRIF0 = 0U;
1082     SREMK0 = 1U;
1083     SREIF0 = 0U;
1084     STPR10 = 1U;
1085     STPR00 = 1U;
1086     SRPR10 = 1U;
1087     SRPR00 = 1U;
1088     SMR00 = 0X0023;
1089     SCR00 = 0X8097;
1090     SDR00 = 0x3200;
1091     NFEN0 |= _01_SAU_RXD0_FILTER_ON;
1092
1093     SIR01 = _0004_SAU_SIRMN_FECTMN | _0002_SAU_SIRMN_PECTMN | _0001_SAU_SIRMN_OVCTMN;
1094
1095     SMR01 = _0020_SAU_SMRMN_INITIALVALUE | _0000_SAU_CLOCK_SELECT_CK00 |
1096     _0100_SAU_TRIGGER_RXD | _0000_SAU_EDGE_FALL |
1097     _0002_SAU_MODE_UART | _0000_SAU_TRANSFER_END;
1098     SCR01 = 0X4097;
1099     SDR01 = 0x3200;
1100     SO0 |= _0001_SAU_CH0_DATA_OUTPUT_1;
1101     SOE0 |= _0000_SAU_CHANNEL0_NORMAL;
1102     SOE0 |= _0001_SAU_CH0_OUTPUT_ENABLE;
1103     PMC1 &= 0xFDU;
1104     PM1 |= 0x02U;
1105     PMC1 &= 0xFBU;
1106     P1 |= 0x04U;
1107     PM1 &= 0xFBU;
1108 }
1109 void R_UART0tx_Start(void)
1110 {
1111     SO0 |= _0001_SAU_CH0_DATA_OUTPUT_1;
1112     SOE0 |= _0001_SAU_CH0_OUTPUT_ENABLE;
1113     SS0 |= _0001_SAU_CH0_START_TRG_ON;
1114     STIF0 = 0U;
1115     STMK0 = 0U;
1116 }
1117 void R_UART0_Send(void)
1118 {
1119     SMR00 |= _0001_SAU_BUFFER_EMPTY;
1120     STMK0 = 1U;
1121     if(flag.ROMTX==1)
1122     {
1123         TXD0 = *gp_uart0_tx_address_r;
1124         gp_uart0_tx_address_r++;
1125     }
1126     else
1127     {
1128         TXD0 = *gp_uart0_tx_address;
1129         gp_uart0_tx_address++;
1130     }
1131     tx.c.g_uart0_tx_count--;
1132     STMK0 = 0U;
1133 }
1134 void R_WDT_Create(void)
1135 {
1136     WDTIMK = 1U;
1137     WDTIIF = 0U;

```



```

1135 }
1136
1137 void DELAY2MS(void)
1138 {unsigned char D2;
1139     WDTE=0XAC;
1140     D2=0Xf0;
1141     while(--D2);
1142 }
1143
1144 static void __near r_it_interrupt(void)
1145 {
1146
1147     if(stop_t!=0)
1148         stop_t--;
1149     flag.int_200ms=1;
1150
1151 }
1152 static void __near r_tau0_channel0_interrupt(void)
1153 {
1154     unsigned char t0_d1;
1155     if(flag.TXWC==0)
1156     {
1157         rx_bz++;
1158         if(rx_bz>=5)
1159             RX_EEROR();
1160     }
1161     else
1162     {
1163         if(TX_TIME!=0)
1164             TX_TIME--;
1165         if(TX_TIME==0)
1166         {
1167             TX=1;
1168             t0_d1=0X5;
1169             while(--t0_d1);
1170             flag.TXWC=0;
1171             rx_bz=3;
1172             TS0 |=0X001;
1173             PMK2 = 1U;
1174             SS0 |= _0002_SAU_CH1_START_TRG_ON;
1175             SRIF0 = 0U;
1176             SRMK0 = 0U;
1177         }
1178     }
1179 }
1180 void RX_EEROR(void)
1181 {
1182     rx_bz=0;
1183     rx.data[15]=0;
1184     SRMK0 = 1U;
1185     ST0 |= _0002_SAU_CH1_STOP_TRG_ON ;
1186     SRIF0 = 0U;
1187     TT0 |= 0X001;
1188     TMMK00 = 1U;
1189     TMIF00 = 0U;
1190
1191     EGP0=0X00;
1192     PIF2 = 0U;
1193     PMK2 = 0U;
1194 }
1195 static void __near r_intc2_interrupt(void)
1196 {
1197     switch(rx_bz){
1198         case 0:
1199             if(INT==0){
1200                 EGP0=0X04;
1201                 TMIF00 = 0U;
1202                 TMMK00 = 0U;
1203                 TS0 |=0X001;
1204                 rx_bz=1;
1205             }break;

```

```

1206     case 1:
1207     RX_ERROR();
1208     break;
1209     case 2:
1210     case 3:
1211     case 4:
1212     rx_bz=3;
1213     TS0 |=0x001;
1214     PMK2 = 1U;
1215     SS0 |= _0002_SAU_CH1_START_TRG_ON;
1216     SRIF0 = 0U;
1217     SRMK0 = 0U;
1218     break;
1219     }
1220 }
1221 static void __near r_uart0_interrupt_receive(void)
1222 {
1223     rx_data = RXD0;
1224     rx_bz=3;
1225     TS0 |=0x001;
1226     err_type = (unsigned char)(SSR01 & 0x0007U);
1227     SIR01 = (unsigned int)err_type;
1228     if(rx.data[15]>=16)
1229     err_type=0x81;
1230     else
1231     {
1232         rx.data[rx.data[15]]=rx_data;
1233         if(rx.data[15]==0)
1234         {
1235             rxd_jcend.I=0;
1236             if(rx_data!=';')
1237             err_type=0x82;
1238         }
1239         if(rx.data[15]>=6)
1240         {
1241             rxd_jcend.H=rxd_jcend.L;
1242             rxd_jcend.L=rx_data;
1243             if(rxd_jcend.I==0x0D0A)
1244             {
1245                 RX_DATACX();
1246                 if(flag.ROMTX==0)
1247                 gp_uart0_tx_address=tx.c.TX_DATA;
1248                 if(tx.c.g_uart0_tx_count==0)
1249                 {
1250                     flag.ROMTX=1;
1251                     tx.c.g_uart0_tx_count=7;
1252                     gp_uart0_tx_address_r=ERROR;
1253                 }
1254                 R_UART0tx_Start();
1255                 R_UART0_Send();
1256             }
1257         }
1258     }
1259     rx.data[15]++;
1260     if(tx.c.g_uart0_tx_count>0)
1261     rx.data[15]=0;
1262     if (err_type != 0)
1263     RX_ERROR();
1264 }
1265 static void __near r_uart0_interrupt_send(void)
1266 {
1267     rx_bz=3;
1268     TS0 |=0x001;
1269     if (tx.c.g_uart0_tx_count > 0U)
1270     {
1271         if(tx.c.g_uart0_tx_count==1)
1272             TXD0 = 0x0A;
1273         if(tx.c.g_uart0_tx_count==2)
1274             TXD0 = 0x0D;
1275         if(tx.c.g_uart0_tx_count>=3)
1276             if(flag.ROMTX==1)

```

```

1277     {
1278         TXD0 = *gp_uart0_tx_address_r;
1279         gp_uart0_tx_address_r++;}
1280     else
1281     {
1282         TXD0 = *gp_uart0_tx_address;
1283         gp_uart0_tx_address++;}
1284
1285     tx.c.g_uart0_tx_count--;
1286 }
1287 else
1288 {
1289     SMR00 &= ~_0001_SAU_BUFFER_EMPTY;
1290     if ((SSR00 & _0040_SAU_UNDER_EXECUTE) == 0U)
1291     {
1292         if(CXTO_Z==0xff)
1293             RX_EROR();
1294     }
1295 }
1296 }
1297 void RX_DATA CX(void)
1298 {
1299     unsigned char rx_d1;
1300     unsigned int rx_d2;
1301     tx.c.g_uart0_tx_count=0;
1302     flag.ROMTX=0;
1303     rx.data[19]=rx.data[1];
1304     rx.data[18]=rx.data[2];
1305     rx.data[17]=rx.data[3];
1306     rx.data[16]=rx.data[4];
1307     CXTO_Z=0;
1308     if(rx.data[5]=='?')
1309     {
1310         switch(rx.c.ml){
1311             case 0x53454E55 :
1312             {
1313                 tx.c.TX_DATA[0]=SENU_Y.H/10+0X30;
1314                 tx.c.TX_DATA[1]=SENU_Y.H%10+0X30;
1315                 tx.c.TX_DATA[2]=SENU_Y.L/10+0X30;
1316                 tx.c.TX_DATA[3]=SENU_Y.L%10+0X30;
1317                 tx.c.TX_DATA[4]='-';
1318                 tx.c.TX_DATA[5]=SENU_B/10+0X30;
1319                 tx.c.TX_DATA[6]=SENU_B%10+0X30;
1320                 tx.c.TX_DATA[7]='-';
1321                 rx_d2=SENU_S.I;
1322                 tx.c.TX_DATA[8]=rx_d2/10000+0X30;
1323                 rx_d2=rx_d2%10000;
1324                 tx.c.TX_DATA[9]=rx_d2/1000+0X30;
1325                 rx_d2=rx_d2%1000;
1326                 tx.c.TX_DATA[10]=rx_d2/100+0x30;
1327                 rx_d2=rx_d2%100;
1328                 tx.c.TX_DATA[11]=rx_d2/10+0x30;
1329                 rx_d2=rx_d2%10;
1330                 tx.c.TX_DATA[12]=rx_d2+0x30;
1331                 tx.c.g_uart0_tx_count=15;
1332                 return;}
1333             case 0x49395349 :
1334             {
1335                 flag.ROMTX=1;
1336                 tx.c.g_uart0_tx_count=19;
1337                 gp_uart0_tx_address_r=I9SI;
1338                 return;}
1339             case 0x44455459 :
1340             {
1341                 tx.c.TX_DATA[0]=DETY_Z/10+0X30;
1342                 tx.c.TX_DATA[1]=DETY_Z%10+0X30;
1343                 tx.c.g_uart0_tx_count=4;
1344                 return;}
1345             case 0x4445434C :
1346             {
1347                 tx.c.TX_DATA[0]=DECL_Z+0X30;

```

```

1348 tx.c.g_uart0_tx_count=3;
1349 return;}
1350 case 0x44544143 :
1351 {
1352 tx.c.TX_DATA[0]=DETY_Z/10+0X30;
1353 tx.c.TX_DATA[1]=DETY_Z%10+0X30;
1354 tx.c.TX_DATA[2]=' ','';
1355 tx.c.TX_DATA[3]=DECL_Z+0X30;
1356 tx.c.g_uart0_tx_count=6;
1357 return;}
1358 case 0X45443031:
1359 {
1360 if(sz_mos.L&0X01)
1361 {
1362 tx.c.TX_DATA[0]=ED01_Z+0X30;
1363 tx.c.g_uart0_tx_count=3;
1364 }
1365 return;}
1366 case 0X45443032:
1367 {
1368 if(sz_mos.L&0X02)
1369 {
1370 tx.c.TX_DATA[0]=ED02_Z+0X30;
1371 tx.c.g_uart0_tx_count=3;
1372 }
1373 return;}
1374 case 0X45443034:
1375 {
1376 if(sz_mos.L&0X08)
1377 {
1378 tx.c.TX_DATA[0]=ED04_Z+0X30;
1379 tx.c.g_uart0_tx_count=3;
1380 }
1381 return;}
1382 case 0x53544831 :
1383 {
1384 tx.c.TX_DATA[0]=STH1_Z/10+0X30;
1385 tx.c.TX_DATA[1]=STH1_Z%10+0X30;
1386 tx.c.g_uart0_tx_count=4;
1387 return;}
1388 case 0X53544830:
1389 {
1390 tx.c.TX_DATA[0]=STH0_Z/10+0X30;
1391 tx.c.TX_DATA[1]=STH0_Z%10+0X30;
1392 tx.c.g_uart0_tx_count=4;
1393 return;
1394 }
1395 case 0x48544831 :
1396 {
1397 tx.c.TX_DATA[0]=HTH1_Z/10+0X30;
1398 tx.c.TX_DATA[1]=HTH1_Z%10+0X30;
1399 tx.c.g_uart0_tx_count=4;
1400 return;}
1401 case 0x48544830:
1402 {
1403 tx.c.TX_DATA[0]=HTH0_Z/10+0X30;
1404 tx.c.TX_DATA[1]=HTH0_Z%10+0X30;
1405 tx.c.g_uart0_tx_count=4;
1406 return;
1407 }
1408 case 0x5356414C :
1409 {
1410 tx.c.TX_DATA[0]=SVAL_Z/10+0X30;
1411 tx.c.TX_DATA[1]=SVAL_Z%10+0X30;
1412 tx.c.g_uart0_tx_count=4;
1413 return;}
1414 case 0x4856414C :
1415 {
1416 tx.c.TX_DATA[0]=HVAL_Z/10+0X30;
1417 tx.c.TX_DATA[1]=HVAL_Z%10+0X30;
1418 tx.c.g_uart0_tx_count=4;

```

```

1419     return;}
1420     case      0x50495253   :
1421     {
1422     tx.c.TX_DATA[0]=PIRS_Z+0x30;
1423     tx.c.g_uart0_tx_count=3;
1424     return;}
1425     case      0x50495245   :
1426     {
1427     tx.c.TX_DATA[0]=PIRE_Z+0x30;
1428     tx.c.g_uart0_tx_count=3;
1429     return;}
1430     case      0x494C4544   :
1431     {
1432     tx.c.TX_DATA[0]=ILED_Z+0x30;
1433     tx.c.g_uart0_tx_count=3;
1434     return;}
1435     case      0x53544154   :
1436     {
1437     tx.c.TX_DATA[0]=STAT_Z+0x30;
1438     tx.c.TX_DATA[1]=' ';
1439     tx.c.TX_DATA[2]=STAT_Z1/10+0x30;
1440     tx.c.TX_DATA[3]=STAT_Z1%10+0x30;
1441     if(STAT_Z==1)
1442     flag.YBTXQL=1;
1443     tx.c.g_uart0_tx_count=6;
1444     if(ev_int.c.S1==0&&ev_bj.c.S1==1)
1445     {ev_int.c.S1=1;return;}
1446     if(ev_int.c.S0==0&&ev_bj.c.S0==1)
1447     {ev_int.c.S0=1;return;}
1448     if(ev_int.c.H1==0&&ev_bj.c.H1==1)
1449     {ev_int.c.H1=1;return;}
1450     if(ev_int.c.H0==0&&ev_bj.c.H0==1)
1451     {ev_int.c.H0=1;return;}
1452     if(ev_int.c.PIR==0&&ev_bj.c.PIR==1)
1453     {ev_int.c.PIR=1;return;}
1454     return;}
1455     case      0x4F435331   :
1456     {
1457     tx.c.TX_DATA[0]=OCS1_Z+0x30;
1458     tx.c.g_uart0_tx_count=3;
1459     return;}
1460     case      0x4F504543   :
1461     {
1462     tx.c.TX_DATA[0]=OPEC_Z+0x30;
1463     tx.c.g_uart0_tx_count=3;
1464     return;}
1465     case      0x494444E55   :
1466     {
1467         rx_d1=IDNU_Z.m4;
1468         if(rx_d1>=0x0a) rx_d1+=0x37;
1469         else rx_d1+=0x30;
1470     tx.c.TX_DATA[0]=rx_d1;
1471
1472         rx_d1=IDNU_Z.m3;
1473         if(rx_d1>=0x0a) rx_d1+=0x37;
1474         else rx_d1+=0x30;
1475     tx.c.TX_DATA[1]=rx_d1;
1476
1477         rx_d1=IDNU_Z.m2;
1478         if(rx_d1>=0x0a) rx_d1+=0x37;
1479         else rx_d1+=0x30;
1480     tx.c.TX_DATA[2]=rx_d1;
1481
1482         rx_d1=IDNU_Z.m1;
1483         if(rx_d1>=0x0a) rx_d1+=0x37;
1484         else rx_d1+=0x30;
1485     tx.c.TX_DATA[3]=rx_d1;
1486
1487     tx.c.g_uart0_tx_count=6;
1488     return;}
1489     case      0x4452534F   :

```

```

1490 {
1491 tx.c.TX_DATA[0]=DRSO_Z+0x30;
1492 tx.c.g_uart0_tx_count=3;
1493 return;
1494 }
1495 case 0x44424944 :
1496 {
1497 rx_d1=DBID_Z;
1498 tx.c.TX_DATA[0]=rx_d1/100+0x30;
1499 rx_d2=rx_d1%100;
1500 tx.c.TX_DATA[1]=rx_d2/10+0x30;
1501 tx.c.TX_DATA[2]=rx_d2%10+0x30;
1502 tx.c.g_uart0_tx_count=5;
1503 return;}
1504 }
1505 return;}
1506 if(rx.data[5]==' ')
1507 {
1508 switch(rx.c.ml){
1509 case 0x45443031:
1510 {
1511 if(sz_mos.L&0x01){
1512 rx_d1=RXDATAJS1();
1513 if((rx_d1==0)&&(js_data<=1))
1514 {
1515 ED01_Z=js_data;
1516 flag.ROMTX=1;
1517 tx.c.g_uart0_tx_count=4;
1518 gp_uart0_tx_address_r=OK;
1519 }}
1520 return;}
1521 case 0x45443032:
1522 {
1523 if(sz_mos.L&0x02){
1524 rx_d1=RXDATAJS1();
1525 if((rx_d1==0)&&(js_data<=1))
1526 {
1527 ED02_Z=js_data;
1528 flag.ROMTX=1;
1529 tx.c.g_uart0_tx_count=4;
1530 gp_uart0_tx_address_r=OK;
1531 }}
1532 return;}
1533 case 0x45443034:
1534 {
1535 if(sz_mos.L&0x08) {
1536 rx_d1=RXDATAJS1();
1537 if((rx_d1==0)&&(js_data<=1))
1538 {
1539 ED04_Z=js_data;
1540 flag.ROMTX=1;
1541 tx.c.g_uart0_tx_count=4;
1542 gp_uart0_tx_address_r=OK;
1543 }}
1544 return;}
1545 case 0x53544831 :
1546 {
1547 if(RXDATAJS2()==0)
1548 {STH1_Z=js_data;
1549 flag.ROMTX=1;
1550 tx.c.g_uart0_tx_count=4;
1551 gp_uart0_tx_address_r=OK;
1552 ev_int.c.S1=0;
1553 }
1554 return;}
1555 case 0x53544830 :
1556 {
1557 if(RXDATAJS2()==0)
1558 {STH0_Z=js_data;
1559 flag.ROMTX=1;
1560 tx.c.g_uart0_tx_count=4;

```

```
1561     gp_uart0_tx_address_r=OK;
1562     ev_int.c.S0=0;
1563 }
1564 return;}
1565 case      0x48544831 :
1566 {
1567     if(RXDATAJS2()==0)
1568     {
1569         HTH1_Z=js_data;
1570         flag.ROMTX=1;
1571         tx.c.g_uart0_tx_count=4;
1572         gp_uart0_tx_address_r=OK;
1573         ev_int.c.H1=0;
1574     }
1575     return;}
1576 case      0x48544830 :
1577 {
1578     if(RXDATAJS2()==0)
1579     {
1580         HTH0_Z=js_data;
1581         flag.ROMTX=1;
1582         tx.c.g_uart0_tx_count=4;
1583         gp_uart0_tx_address_r=OK;
1584         ev_int.c.H0=0;
1585     }
1586     return;}
1587 case      0x50495245 :
1588 {
1589     rx_d1=RXDATAJS1();
1590     if((rx_d1==0)&&(js_data<=1))
1591     {
1592         PIRE_Z=js_data;
1593         if(sz_mos.L&0X08)
1594         {
1595             if(PIRE_Z==1)
1596                 PIRDY=1;
1597             else
1598                 PIRDY=0;
1599         }
1600         flag.ROMTX=1;
1601         tx.c.g_uart0_tx_count=4;
1602         gp_uart0_tx_address_r=OK;
1603     }
1604     return;}
1605 case      0x494C4544 :
1606 {
1607     rx_d1=RXDATAJS1();
1608     if((rx_d1==0)&&(js_data<=1))
1609     {
1610         ILED_Z=js_data;
1611         if(ILED_Z==1)
1612             LED=1;
1613         else
1614             LED=0;
1615         flag.ROMTX=1;
1616         tx.c.g_uart0_tx_count=4;
1617         gp_uart0_tx_address_r=OK;
1618     }
1619     return;}
1620 case      0x4F504543 :
1621 {
1622     rx_d1=RXDATAJS1();
1623     if((rx_d1==0)&&(js_data<=1))
1624     {
1625         OPEC_Z=js_data;
1626         flag.ROMTX=1;
1627         tx.c.g_uart0_tx_count=4;
1628         gp_uart0_tx_address_r=OK;
1629     }
1630     return;}
1631 case      0x49444E55 :
```

```

1632     {
1633     if(RXDATAJS4()==0)
1634     {
1635     IDNU_Z.L=js_data1.L;
1636     flag.ROMTX=1;
1637     tx.c.g_uart0_tx_count=4;
1638     gp_uart0_tx_address_r=OK;
1639     }
1640     return;}
1641     case    0x4452534F    :
1642     {
1643     if(RXDATAJS1()==0)
1644     {
1645     DRISO_Z=js_data;
1646     flag.ROMTX=1;
1647     tx.c.g_uart0_tx_count=4;
1648     gp_uart0_tx_address_r=OK;
1649     }
1650     return;}
1651     case    0x44424944    :
1652     {
1653     if(RXDATAJS3()==0)
1654     {
1655     DBID_LSZ=js_data;
1656     flag.ROMTX=1;
1657     tx.c.g_uart0_tx_count=4;
1658     gp_uart0_tx_address_r=OK;
1659     }
1660     return;}
1661     }
1662     return;}
1663     if(rx.data[5]=='+')
1664     {
1665     switch(rx.c.ml){
1666     case    0x50495243    :
1667     {
1668     if(PIRS_Z==1)
1669     {
1670     ev_int.c.PIR=0;
1671     ev_bj.c.PIR=0;
1672     PIRS_Z=0;
1673     }
1674     PIR_CONT=0;
1675     flag.ROMTX=1;
1676     tx.c.g_uart0_tx_count=4;
1677     gp_uart0_tx_address_r=OK;
1678     return;}
1679     case    0x4358544F    :
1680     {
1681     flag.ROMTX=1;
1682     tx.c.g_uart0_tx_count=4;
1683     gp_uart0_tx_address_r=OK;
1684     CXTO_Z=0xFF;
1685     return;}
1686     }
1687     return;}
1688 }
1689 unsigned char RXDATAJS1(void)
1690 {
1691     unsigned char js_d1;
1692     unsigned char js_gz=0;
1693     if(rx.data[15]!=8)
1694     return js_gz=0x80;
1695     js_d1=rx.data[6];
1696     if(js_d1<=0x2f)
1697     js_gz|=1;
1698     if(js_d1>=0x3a)
1699     js_gz|=2;
1700     if(js_gz!=0)
1701     return js_gz;
1702     js_data=js_d1-0x30;

```



```

1703     return 0;
1704 }
1705 unsigned char RXDATAJS2(void)
1706 {
1707     unsigned char js_d1;
1708     unsigned char js_gz=0;
1709     if(rx.data[15]!=9)
1710         return js_gz=0x80;
1711     js_d1=rx.data[6];
1712     if(js_d1<=0x2f)
1713         js_gz|=1;
1714     if(js_d1>=0x3a)
1715         js_gz|=2;
1716     if(js_gz!=0)
1717         return js_gz;
1718     js_data=(js_d1-0x30)*10;
1719     js_d1=rx.data[7];
1720     if(js_d1<=0x2f)
1721         js_gz|=4;
1722     if(js_d1>=0x3a)
1723         js_gz|=8;
1724     if(js_gz!=0)
1725         return js_gz;
1726     js_d1=js_d1-0x30;
1727     js_data+=js_d1;
1728     return 0;
1729 }
1730 unsigned char RXDATAJS3(void)
1731 {
1732     unsigned char js_d1;
1733     unsigned char js_gz=0;
1734     if(rx.data[15]!=10)
1735         return js_gz=0x80;
1736     js_d1=rx.data[6];
1737     if(js_d1<=0x2f)
1738         js_gz|=1;
1739     if(js_d1>=0x3a)
1740         js_gz|=2;
1741     if(js_gz!=0)
1742         return js_gz;
1743     js_data=(js_d1-0x30)*100;
1744     js_d1=rx.data[7];
1745     if(js_d1<=0x2f)
1746         js_gz|=4;
1747     if(js_d1>=0x3a)
1748         js_gz|=8;;
1749     if(js_gz!=0)
1750         return js_gz;
1751     js_d1=(js_d1-0x30)*10;
1752     js_data+=js_d1;
1753     js_d1=rx.data[8];
1754     if(js_d1<=0x2f)
1755         js_gz|=4;
1756     if(js_d1>=0x3a)
1757         js_gz|=8;
1758     if(js_gz!=0)
1759         return js_gz;
1760     js_d1=js_d1-0x30;
1761     js_data+=js_d1;
1762
1763     return 0;
1764 }
1765 unsigned char RXDATAJS4(void)
1766 {
1767     unsigned char js_d1;
1768     unsigned char js_gz=0;
1769     if(rx.data[15]!=11)
1770         return js_gz=0x80;
1771
1772     js_d1=rx.data[6];
1773     if(js_d1<=0x2f)

```

```
1774     js_gz|=1;
1775     if(js_d1>=0x47)
1776         js_gz|=2;;
1777     if(js_gz!=0)
1778         return js_gz;
1779     if(js_d1>=0x41)
1780         js_d1-=0x37;
1781     else
1782         js_d1-=0x30;
1783     js_datal.m4=js_d1;
1784
1785     js_d1=rx.data[7];
1786     if(js_d1<=0x2f)
1787         js_gz|=1;
1788     if(js_d1>=0x47)
1789         js_gz|=2;
1790     if(js_gz!=0)
1791         return js_gz;
1792     if(js_d1>=0x41)
1793         js_d1-=0x37;
1794     else
1795         js_d1-=0x30;
1796     js_datal.m3=js_d1;
1797
1798     js_d1=rx.data[8];
1799     if(js_d1<=0x2f)
1800         js_gz|=1;
1801     if(js_d1>=0x47)
1802         js_gz|=2;
1803     if(js_gz!=0)
1804         return js_gz;
1805     if(js_d1>=0x41)
1806         js_d1-=0x37;
1807     else
1808         js_d1-=0x30;
1809     js_datal.m2=js_d1;
1810
1811     js_d1=rx.data[9];
1812     if(js_d1<=0x2f)
1813         js_gz|=1;
1814     if(js_d1>=0x47)
1815         js_gz|=2;
1816     if(js_gz!=0)
1817         return js_gz;
1818     if(js_d1>=0x41)
1819         js_d1-=0x37;
1820     else
1821         js_d1-=0x30;
1822     js_datal.m1=js_d1;
1823     return 0;
1824 }
1825
```