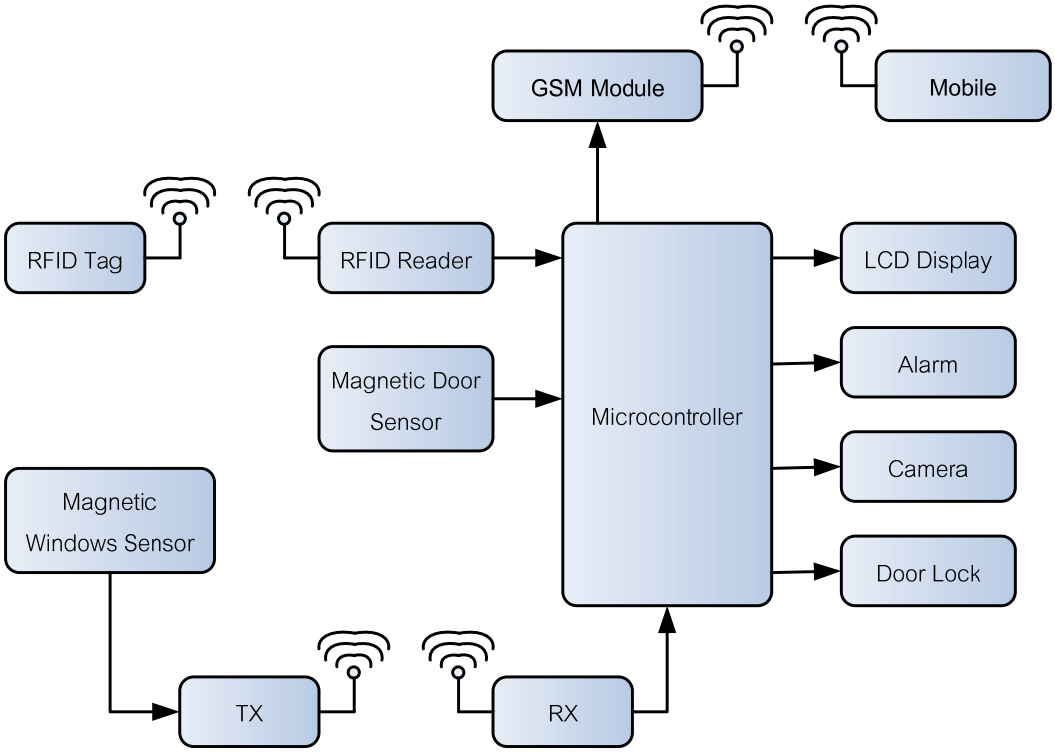
ภาคผนวก

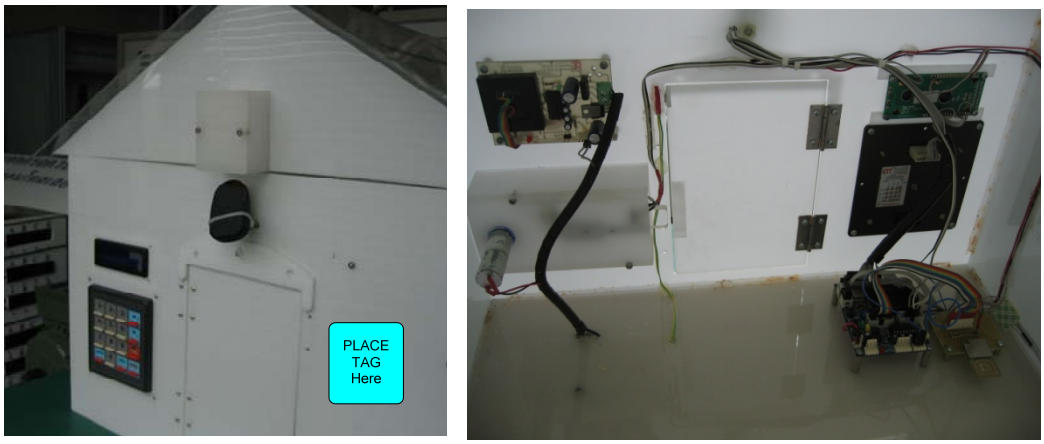
ภาคผนวก ก

ตัวอย่างการประยุกต์ใช้งาน RFID ที่นักศึกษาได้จัดทำขึ้น โดยมีผู้วิจัยเป็นอาจารย์ที่ปรึกษา

1) ระบบรักษาความปลอดภัยและอนุญาตเข้า-ออกบ้านเฉพาะบุคคล (เรวัต เปรมศรี และคณะ)

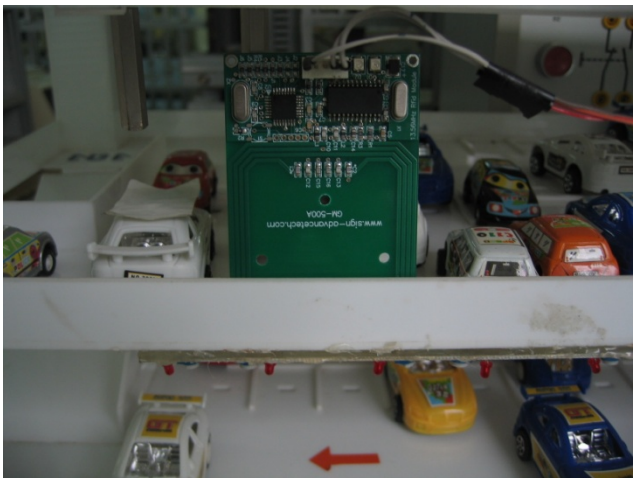


ผังการทำงาน

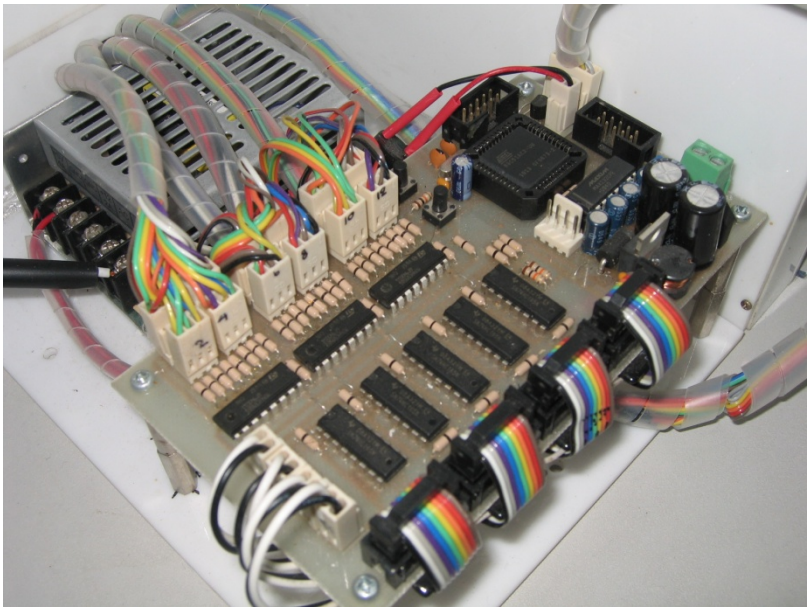


แบบจำลอง

2) ระบบจอดรถควบคุมด้วยไมโครคอนโทรลเลอร์ (มนูญ ธงชัยภูมิ และคณะ)

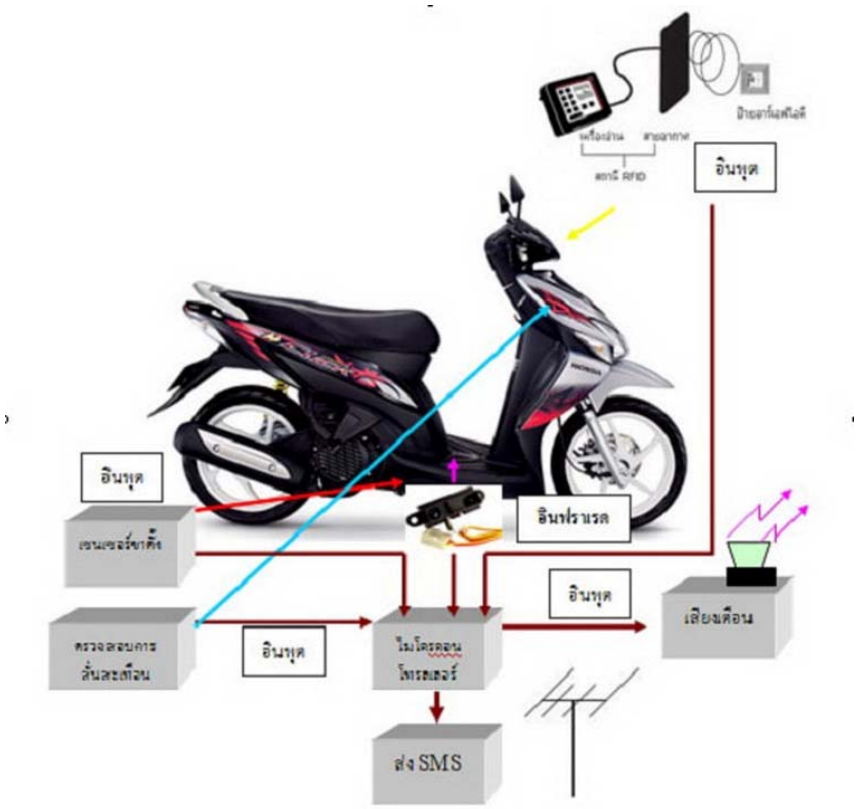


แบบจำลอง

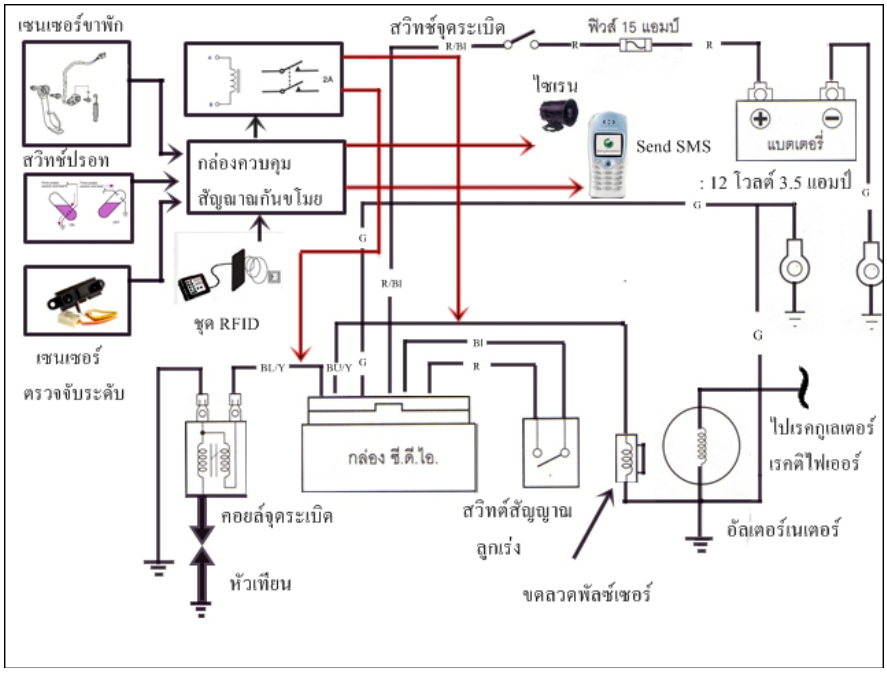


วงจรควบคุม

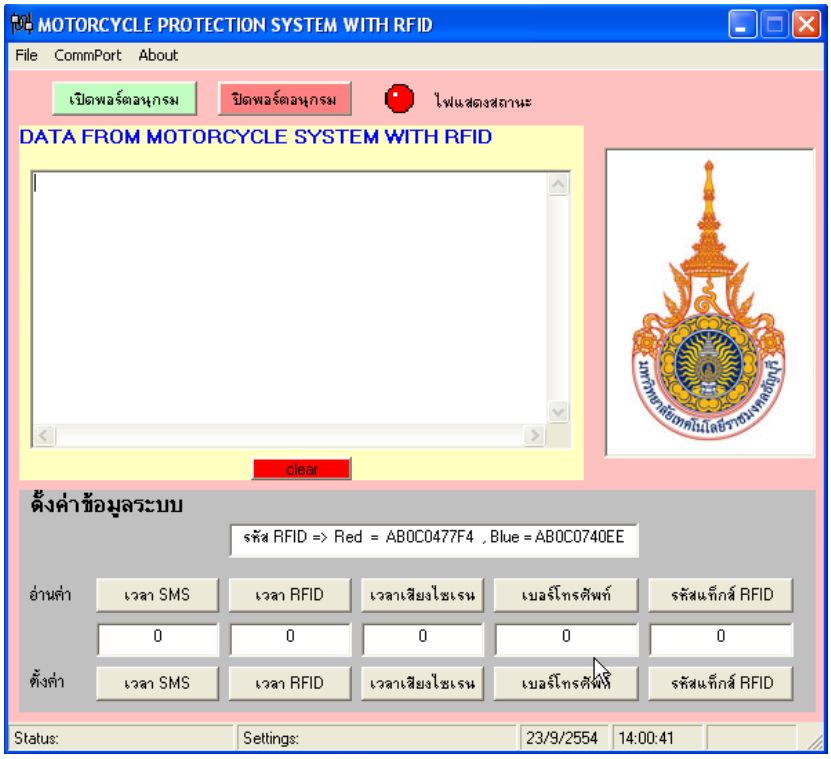
3) ระบบป้องกันการโจรกรรมรถจักรยานยนต์ (เชิดศักดิ์ ทำทอง และคณะ)



แนวความคิด



ผังการควบคุม



การตั้งค่าระบบ

ภาคผนวก ข

1) โปรแกรมตัวอย่างที่ใช้ในการทดสอบชุดทดลองในหัวข้อ 3.3.1

```
//=============================================================================
//  Project         : RFID Door Lock
//=============================================================================
#include <pic.h>
//  configuration
//=============================================================================
__CONFIG ( 0x3F32 );         //config MCU
//  define
//=============================================================================
#define  rs       RA0                 //RS of the LCD
#define  e        RC5                 //E of the LCD
#define  b_light     RC4              //backlight LCD (1 to on backlight)
#define relay     RA1                 //magnetic lock (active high)
#define  lcd_data   PORTB             //LCD data PORT
#define  led       RC3                //led(active high)
#define data0     RC1                 //data0(green wire) of RFID reader
#define data1     RC2                 //data1(white wire) of RFID reader
#define sw1       RA4                  //button to save ID(active low)
#define RX_PIN      RC7               //Rx pin on PIC
#define TX_PIN      RC6                //Tx pin on PIC
//  function prototype                  //=========================================================
void init(void);
void delay(unsigned long data);
void send_config(unsigned char data);
void send_char(unsigned char data);
void send_dec(unsigned long data,unsigned char num_dig);
void e_pulse(void);
void lcd_goto(unsigned char data);
void lcd_clr(void);
void send_string(const char *s);
//  main function                       //=========================================================
void main(void)
{
  unsigned char i, repeat, a, b, database;
  unsigned char data[26];
  unsigned char convert1=0;              //clear convert1
  unsigned int convert2=0;               //clear convert2
  unsigned char mode=0;                  //clear mode
   unsigned char id1[2]={73,0};          //initialize 1st id1 in the array
   unsigned int id2[2]={53490,0};        //initialize 1st id2 in the array
  init();                                //initialization
  lcd_clr();                          //clear the lcd
  id1[1]=0;                           //clear second id1 in the array, first id1 is intialized
  id2[1]=0;                           //clear second id2 in the array, first id2 is intialized
  b_light=1;                           //on the back light
  lcd_goto(0);                         //set the lcd cursor to location 0
  send_string (" RFID Door");                 //display " RFID Door"
  lcd_goto(20);                       //set the lcd cursor to location 20
  send_string (" Security");                //display " Security"
  for(repeat=12; repeat>0; repeat--)           //repeat delay(5000000)for 12 times
  {
    delay(5000000);
  }
```

```
  while(1)
 {
   convert1=0;                          //clear convert1
   convert2=0;                          //clear convert2
    lcd_clr();                         //clear the lcd
   lcd_goto(0);                         //set the lcd cursor to location 0
   send_string ("System Ready");              //display "System Ready"
   lcd_goto(20);                        //set the lcd cursor to location 20
   send_string ("Scan your Tag");             //display "Scan Your Tag"
   while(mode==0)
   {
     if((data0==0)||(data1==0))   mode=1;
     else if(sw1==0)   mode=2;
   }
   while(mode==1)
   {
     if((data0==0)&&(data1==1))
     {
       data[i]=0;
       while((data0==0)&&(data1==1));
     else if ((data0==1)&&(data1==0))                {
       data[i]=1;
       while((data0==1)&&(data1==0));
     }
     i+=1;                          //i+1
     while(i<26)                        //repeat the loop until all 26 bit data are sent (start from 0 to 25)
     {
       while((data0==1)&&(data1==1));           //wait while data0 and data1 remain at high logic level (no
changes at data0 and data1)
       if((data0==0)&&(data1==1))              //if data0 changes (data0 is active low)
       {
         data[i]=0;                     //save that the bit received is 0
         while((data0==0)&&(data1==1));         //wait for data stream finish sending from RFID tag again (data0
or data1 will back to high logic)
       }
       else if ((data0==1)&&(data1==0))         //if data1 received is 0 (data1 is active low)
       {
         data[i]=1;                     //save that the bit received is 1
         while((data0==1)&&(data1==0));         //wait for data stream finish sending from RFID tag again (data0
or data1 will back to high logic)
       }
       i+=1;                          //i+1
     }
     mode=0;                                //after all data are sent, reset as no RFID tag is placed on RFID reader
                                          //no data stream received from RFID tag
     i=0;                                  //clear i
     lcd_clr();                               //clear the lcd

     for(i=0;i<8;i++)                         //loop for data[0]-data[7]
     {
       convert1=(convert1<<1)|data[i+1];              //shift current data and combine with previous data, store
data in convert1       }
     }
     for(i=0;i<16;i++)                        //loop for data[8]-data[25]
     {
       convert2=(convert2<<1)|data[i+9];              //shift current data and combine with previous data, store
data in convert2
     }

     for(b=0;b<2;b++)                         //compare with id[0] and id[1]
     {
```

```
    if((convert1==id1[b])&&(convert2==id2[b])) database=1;     //id doesn't match, set database to 1
  }
  lcd_clr();                                  //clear the lcd
}
//push button is pressed, enter to save mode
while(mode==2)
{
  lcd_clr();                          //clear the lcd
  lcd_goto(0);                        //set the lcd cursor to location 0
  send_string ("    Save Mode   ");           //display " Save Mode "
  lcd_goto(20);                       //set the lcd cursor to location 0
  send_string (" Place Your Tag");            //display " Place Your Tag "

  while(i<26)                         //loop for data less than 26
  {
    while((data0==1)&&(data1==1));            //wait for data0 and data1 to change
    if((data0==0)&&(data1==1))               //if data0 changes (data0 is active low)
    {
      data[i]=0;                      //save that the bit received is 0
      while((data0==0)&&(data1==1));          //wait until data0 and data1 back to high logic
    }
    else if ((data0==1)&&(data1==0))          //if data1 changes (data1 is active low)
    {
      data[i]=1;                      //save that the bit received is 1
      while((data0==1)&&(data1==0));          //wait until data0 and data1 back to high logic
    }
    i+=1;                             //i+1
  }

  mode=0;                             //clear mode
  i=0;                               //clear i
  lcd_clr();                          //clear the lcd

  for(i=0;i<8;i++)                    //loop for data[0]-data[7]
  {
    convert1=(convert1<<1)|data[i+1];         //shift previous data and combine with present data
  }
  for(i=0;i<16;i++)                   //loop for data[8]-data[25]
  {
    convert2=(convert2<<1)|data[i+9];         //shift previous data and combine with present data
  }

  id1[1]=convert1;                    //save convert1 to id1
  id2[1]=convert2;                    //save convert2 to id2

  database=2;                         //display ID no. and "User ID saved" on LCD
  lcd_clr();                         //clear the lcd
}

switch(database)
{
  case 1:                            //id 1 match
    relay=1;                         //door lock released
    led=0;                           //led off
    lcd_goto(0);                     //set lcd cursor to location 0
    send_string("ID No: ");                 //display "ID No: "
    lcd_goto(7);                     //set lcd cursor to location 7
    send_dec(convert1,3);                   //display convert1 in 3 decimal number
    lcd_goto(10);                    //set lcd cursor to location 10
    send_dec(convert2,5);                   //display convert2 in 5 decimal number
    lcd_goto(20);                    //set lcd cursor to location 20
```

```
      send_string("User Identified. ");          //display "User Identified."
      for(repeat=12; repeat>0; repeat--)         //repeat delay(100000) for 12 times
      {
        delay(1000000);
      }
      lcd_clr();                        //clear the LCD
      lcd_goto(0);                      //set lcd cursor to location 0
      send_string("Door lock will");            //display "Door lock will be locked in"
      lcd_goto(20);
      send_string("be locked in ");

      for(a=5;a>0;a--)                  //display number from 5 to 1
      {
        lcd_goto(34);                   //set lcd cursor to location 34
        send_dec(a, 1);                 //display a in 1 decimal number
        for(repeat=10; repeat>0; repeat--)      //repeat delay(100000) for 10 times
        {
          delay(1000000);
        }
      }
      break;

    case 2:                             //save id
      relay=1;                          //door lock released
      led=0;                            //led off
      lcd_clr();                        //clear lcd
      lcd_goto(0);                      //set lcd cursor to location 0
      send_string("ID No: ");                   //display "ID No: "
      lcd_goto(7);                      //set lcd cursor to location 7
      send_dec(convert1,3);                     //display convert1 in 3 decimal number
      lcd_goto(10);                     //set lcd cursor to location 10
      send_dec(convert2,5);                     //display convert2 in 5 decimal number
      lcd_goto(20);                     //set lcd cursor to location 20
      send_string("User ID Saved.");            //display "User ID Saved."
      break;
      default:                          //id doesnt match
      relay=0;                          //door lock locked
      led=1;                            //led light on
      lcd_goto(0);                      //set lcd cursor to location 0
      send_string("ID No: ");                   //display "ID No: "
      lcd_goto(7);                      //set lcd cursor to location 7
      send_dec(convert1,3);                     //display convert in 3 decimal number
      lcd_goto(10);                     //set lcd cursor to location 10
      send_dec(convert2,5);                     //display convert in 5 decimal number
      lcd_goto(20);                     //set lcd cursor to location 20
      send_string("User not found.");           //display "User not found."
      break;
  }

  for(repeat=12; repeat>0; repeat--)            //repeat delay(100000) for 12 times
  {
    delay(1000000);
  }
      i=0;                              //clear i
  led=0;                                //led off
  database=0;                               //no action taken, no RFID tag is placed on RFID reader
  relay=0;                              //off door lock
  convert1=0;                               //clear convert1
  convert2=0;                               //clear convert2
 }
}
```

2) โปรแกรมตัวอย่างที่ใช้ในการทดสอบชุดทดลองในหัวข้อ 3.3.2

```
;################################
;              RFID.             #
;################################
CPU "8051.TBL"
HOF "INT8"
ORG 0
START_POINT_BUFFER:    EQU 040H ; กำหนดหน่วยความจำสำหรับค่า Data RFID
LCD_ADDR:              EQU 030H ; กำหนดหน่วยความสำหรับค่า Address LCD
LCD_DATA:              EQU 031H ; กำหนดหน่วยความสำหรับค่า Data LCD
                       MOV P2,#00H
INITIAL:               MOV SP,#060H
                       MOV TMOD,#20H
                       MOV SCON,#50H
                       MOV TH1,#0FBH
                       SETB TR1
                       MOV P1,#0FFH
                       MOV P0,#00H ; กำหนดค่าเริ่มต้นให้กับพอร์ต 0
MAIN:                  LCALL INIT_LCD ; เรียกโปรแกรมย่อยเพื่อเตรียมการทำงานเริ่มต้น LCD
                       ACALL LCD_CLR ; ล้างจอแสดงผล
                       LCALL INIT_LCD
                       MOV LCD_ADDR,#00H ; กำหนดค่า Address เริ่มต้นเป็น 00
                       LCALL SET_ADDR_LCD
                       MOV DPTR,#TITLE_01 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                       LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 00H-0FH
                       MOV LCD_ADDR,#40H ; กำหนดค่า Address เป็น 40H
                       LCALL SET_ADDR_LCD
                       MOV DPTR,#TITLE_02 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                       LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                ;-------------------------------;
                ;   WAIT RFID DATA   ;
                ;-------------------------------;
WAIT_DATA:      MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                       LCALL SET_ADDR_LCD
                       MOV DPTR,#TITLE_02 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                       LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                       MOV R0,#START_POINT_BUFFER
WAIT_DATA1:     CLR RI
                       JNB RI,$
                       MOV @R0,SBUF
                       INC R0
                       CJNE R0,#START_POINT_BUFFER+14,WAIT_DATA1
                ;-------------------------------------;
                ;      CHECK CARD 1      ;
                ;-------------------------------------;
CHK_CARD1:      MOV R0,#START_POINT_BUFFER
                       CJNE @R0,#02H,START_TX_ERROR          ;STX=02H
                       MOV R0,#START_POINT_BUFFER
                       INC R0
                       CJNE @R0,#32H,CHK_CARD2               ;2
                       INC R0
                       CJNE @R0,#45H,CHK_CARD2               ;E
                       INC R0
                       CJNE @R0,#30H,CHK_CARD2               ;0
                       INC R0
                       CJNE @R0,#30H,CHK_CARD2               ;0
                       INC R0
                       CJNE @R0,#44H,CHK_CARD2               ;D
                       INC R0
                       CJNE @R0,#35H,CHK_CARD2               ;5
                INC R0
                CJNE @R0,#43H,CHK_CARD2                ;C
                INC R0
                CJNE @R0,#33H,CHK_CARD2                ;3
```

```
                INC R0
                CJNE @R0,#46H,CHK_CARD2          ;F
                INC R0
                CJNE @R0,#41H,CHK_CARD2          ;A
                INC R0
                CJNE @R0,#43H,CHK_CARD2          ;C
                INC R0
                CJNE @R0,#32H,CHK_CARD2          ;2
                CPL P2.7
                JNB P2.7,LOOP1
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_1 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
        LOOP1: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_9 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
START_TX_ERROR: LCALL CLR_BUFFER
                        AJMP WAIT_DATA
                ;-------------------------------------;
                ;      CHECK CARD 2      ;
                ;-------------------------------------;
CHK_CARD2:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#32H,CHK_CARD3          ;2
                INC R0
                CJNE @R0,#45H,CHK_CARD3          ;E
                INC R0
                CJNE @R0,#30H,CHK_CARD3          ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD3          ;0
                INC R0
                CJNE @R0,#44H,CHK_CARD3          ;D
                INC R0
                CJNE @R0,#35H,CHK_CARD3          ;5
                INC R0
                CJNE @R0,#45H,CHK_CARD3          ;E
                INC R0
                CJNE @R0,#38H,CHK_CARD3          ;8
                INC R0
                CJNE @R0,#39H,CHK_CARD3          ;9
                INC R0
                CJNE @R0,#37H,CHK_CARD3          ;7
                INC R0
                CJNE @R0,#38H,CHK_CARD3          ;8
                INC R0
                CJNE @R0,#34H,CHK_CARD3          ;4
                CPL P2.6
                JNB P2.6,LOOP2
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_2 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
        LOOP2: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_10 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
                ;---------------------------------;
                ;     CHECK CARD 3     ;
```

```
                ;---------------------------------;
CHK_CARD3:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#32H,CHK_CARD4         ;2
                INC R0
                CJNE @R0,#33H,CHK_CARD4         ;3
                INC R0
                CJNE @R0,#30H,CHK_CARD4         ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD4         ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD4         ;0
                INC R0
                CJNE @R0,#41H,CHK_CARD4         ;A
                INC R0
                CJNE @R0,#34H,CHK_CARD4         ;4
                INC R0
                CJNE @R0,#36H,CHK_CARD4         ;6
                INC R0
                CJNE @R0,#35H,CHK_CARD4         ;5
                INC R0
                CJNE @R0,#32H,CHK_CARD4         ;2
                INC R0
                CJNE @R0,#33H,CHK_CARD4         ;3
                INC R0
                CJNE @R0,#44H,CHK_CARD4         ;D
                CPL P2.5
                JNB P2.5,LOOP3
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_3 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
      LOOP3: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_11 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
                ;---------------------------------;
                ;      CHECK CARD 4    ;
                ;---------------------------------;
CHK_CARD4:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#32H,CHK_CARD5         ;2
                INC R0
                CJNE @R0,#45H,CHK_CARD5         ;E
                INC R0
                CJNE @R0,#30H,CHK_CARD5         ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD5         ;0
                INC R0
                CJNE @R0,#44H,CHK_CARD5         ;D
                INC R0
                CJNE @R0,#35H,CHK_CARD5         ;5
                INC R0
                CJNE @R0,#46H,CHK_CARD5         ;F
                INC R0
                CJNE @R0,#39H,CHK_CARD5         ;9
                INC R0
                CJNE @R0,#43H,CHK_CARD5         ;C
                INC R0
                CJNE @R0,#43H,CHK_CARD5         ;C
                INC R0
                CJNE @R0,#43H,CHK_CARD5         ;C
                INC R0
                CJNE @R0,#45H,CHK_CARD5         ;E
```

```
                CPL P2.4
                JNB P2.4,LOOP4
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_4 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
        LOOP4: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_12 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
                ;---------------------------------;
                ;     CHECK CARD 5     ;
                ;---------------------------------;
CHK_CARD5:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#31H,CHK_CARD6                 ;1
                INC R0
                CJNE @R0,#43H,CHK_CARD6                 ;C
                INC R0
                CJNE @R0,#30H,CHK_CARD6                 ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD6                 ;0
                INC R0
                CJNE @R0,#46H,CHK_CARD6                 ;F
                INC R0
                CJNE @R0,#44H,CHK_CARD6                 ;D
                INC R0
                CJNE @R0,#30H,CHK_CARD6                 ;0
                INC R0
                CJNE @R0,#42H,CHK_CARD6                 ;B
                INC R0
                CJNE @R0,#35H,CHK_CARD6                 ;5
                INC R0
                CJNE @R0,#45H,CHK_CARD6                 ;E
                INC R0
                CJNE @R0,#42H,CHK_CARD6                 ;B
                INC R0
                CJNE @R0,#34H,CHK_CARD6                 ;4
                CPL P2.3
                JNB P2.3,LOOP5
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_5 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
        LOOP5: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_13 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
                ;---------------------------------;
                ;     CHECK CARD 6     ;
                ;---------------------------------;
CHK_CARD6:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#32H,CHK_CARD7                 ;2
                INC R0
                CJNE @R0,#38H,CHK_CARD7                 ;8
                INC R0
                CJNE @R0,#30H,CHK_CARD7                 ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD7                 ;0
```

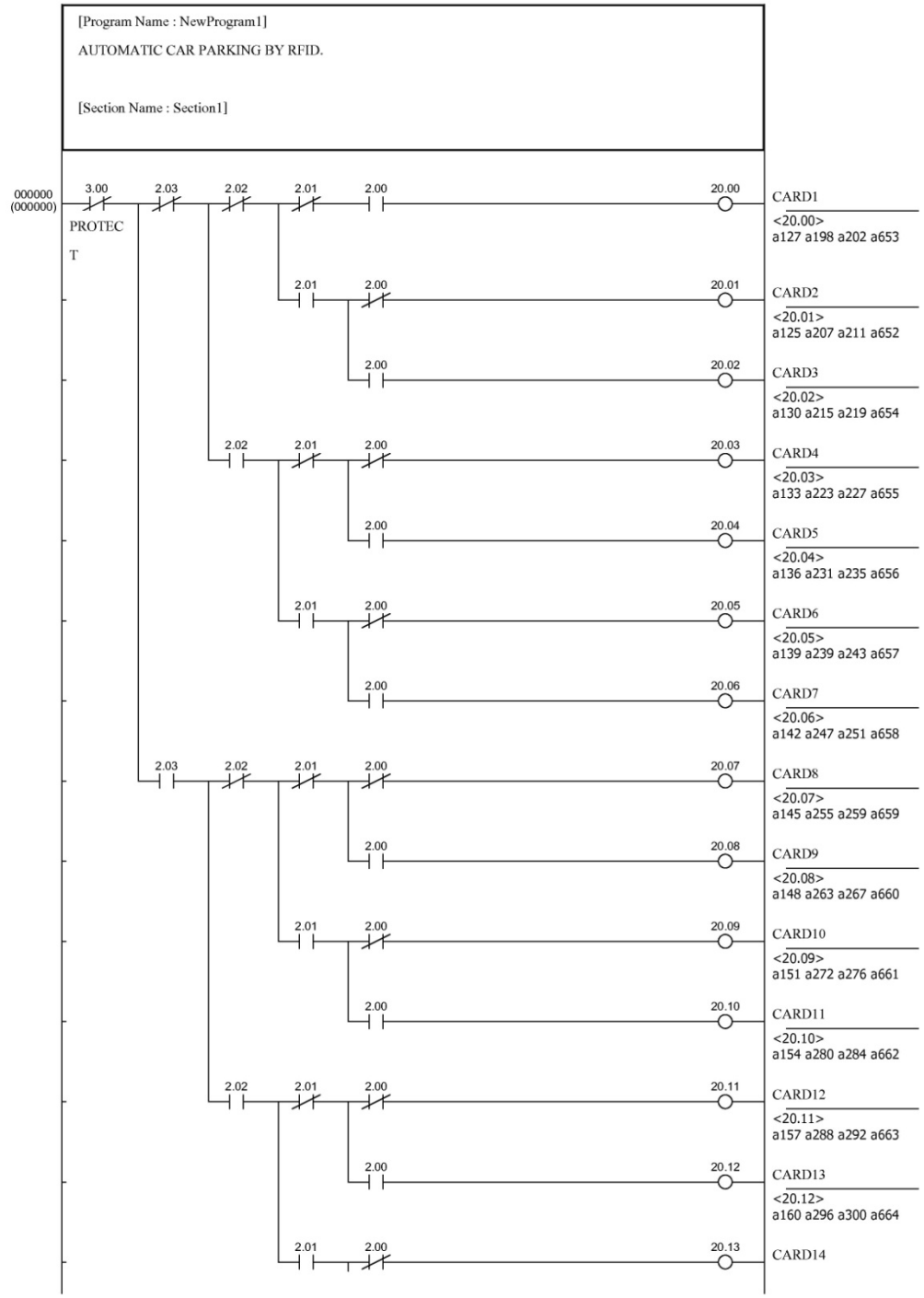
```
                INC R0
                CJNE @R0,#35H,CHK_CARD7          ;5
                INC R0
                CJNE @R0,#45H,CHK_CARD7          ;E
                INC R0
                CJNE @R0,#42H,CHK_CARD7          ;B
                INC R0
                CJNE @R0,#46H,CHK_CARD7          ;F
                INC R0
                CJNE @R0,#36H,CHK_CARD7          ;6
                INC R0
                CJNE @R0,#44H,CHK_CARD7          ;D
                INC R0
                CJNE @R0,#41H,CHK_CARD7          ;A
                INC R0
                CJNE @R0,#34H,CHK_CARD7          ;4
                CPL P2.2
                JNB P2.2,LOOP6
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_6 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
        LOOP6: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_14 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
                ;---------------------------------;
                ;     CHECK CARD 7     ;
                ;---------------------------------;
CHK_CARD7:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#32H,CHK_CARD8          ;2
                INC R0
                CJNE @R0,#38H,CHK_CARD8          ;8
                INC R0
                CJNE @R0,#30H,CHK_CARD8          ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD8          ;0
                INC R0
                CJNE @R0,#34H,CHK_CARD8          ;4
                INC R0
                CJNE @R0,#39H,CHK_CARD8          ;9
                INC R0
                CJNE @R0,#33H,CHK_CARD8          ;3
                INC R0
                CJNE @R0,#41H,CHK_CARD8          ;A
                INC R0
                CJNE @R0,#35H,CHK_CARD8          ;5
                INC R0
                CJNE @R0,#30H,CHK_CARD8          ;0
                INC R0
                CJNE @R0,#30H,CHK_CARD8          ;0
                INC R0
                CJNE @R0,#42H,CHK_CARD8          ;B
                CPL P2.1
                JNB P2.1,LOOP7
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_7 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
        LOOP7: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
```

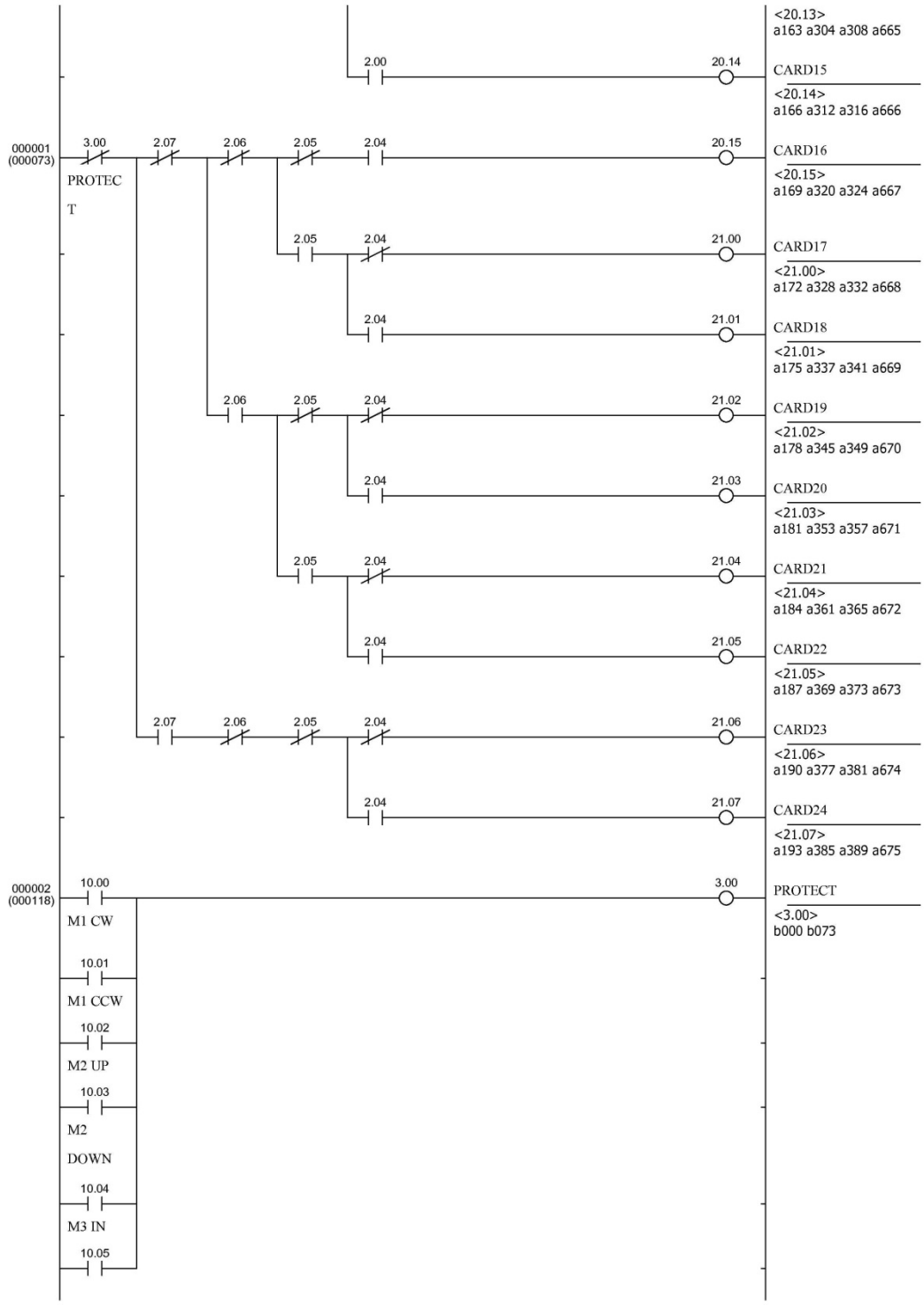
```
                MOV DPTR,#TITLE_15 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
                ;---------------------------------;
                ;     CHECK CARD 8      ;
                ;---------------------------------;
CHK_CARD8:      MOV R0,#START_POINT_BUFFER
                INC R0
                CJNE @R0,#32H,OUT_OF_DATA               ;2
                INC R0
                CJNE @R0,#38H,OUT_OF_DATA               ;8
                INC R0
                CJNE @R0,#30H,OUT_OF_DATA               ;0
                INC R0
                CJNE @R0,#30H,OUT_OF_DATA               ;0
                INC R0
                CJNE @R0,#43H,OUT_OF_DATA               ;C
                INC R0
                CJNE @R0,#31H,OUT_OF_DATA               ;1
                INC R0
                CJNE @R0,#41H,OUT_OF_DATA               ;A
                INC R0
                CJNE @R0,#31H,OUT_OF_DATA               ;1
                INC R0
                CJNE @R0,#39H,OUT_OF_DATA               ;9
                INC R0
                CJNE @R0,#39H,OUT_OF_DATA               ;9
                INC R0
                CJNE @R0,#44H,OUT_OF_DATA               ;D
                INC R0
                CJNE @R0,#31H,OUT_OF_DATA               ;1
                CPL P2.0
                JNB P2.0,LOOP8
                MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_8 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA
      LOOP8: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_16 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY
                AJMP WAIT_DATA

                ;---------------------------------;
                ;      OUT OF DATA      ;
                ;---------------------------------;
OUT_OF_DATA: MOV LCD_ADDR,#040H ; กำหนดค่า Address เป็น 40H
                LCALL SET_ADDR_LCD
                MOV DPTR,#TITLE_03 ; กำหนดตัวชี้ตำแหน่งของตัวอักษรที่ต้องการแสดงผล
                LCALL WRLINE_LCD ; เรียกโปรแกรมย่อยเพื่อเขียนไปยังตำแหน่ง 40H-4FH
                LCALL DELAY_1s
                LCALL LCD_OFF ; เรียกโปรแกรมย่อย ปิดจอแสดงผล
                LCALL DELAY_1s
                LCALL LCD_ON ; เรียกโปรแกรมย่อย เปิดจอแสดงผล
                LCALL DELAY_1s
                LCALL LCD_OFF ; เรียกโปรแกรมย่อย ปิดจอแสดงผล
                LCALL DELAY_1s
                LCALL LCD_ON ; เรียกโปรแกรมย่อย เปิดจอแสดงผล
                LCALL DELAY_1s
                LCALL LCD_OFF ; เรียกโปรแกรมย่อย ปิดจอแสดงผล
                LCALL DELAY_1s
                LCALL LCD_ON ; เรียกโปรแกรมย่อย เปิดจอแสดงผล
                LCALL DELAY
```

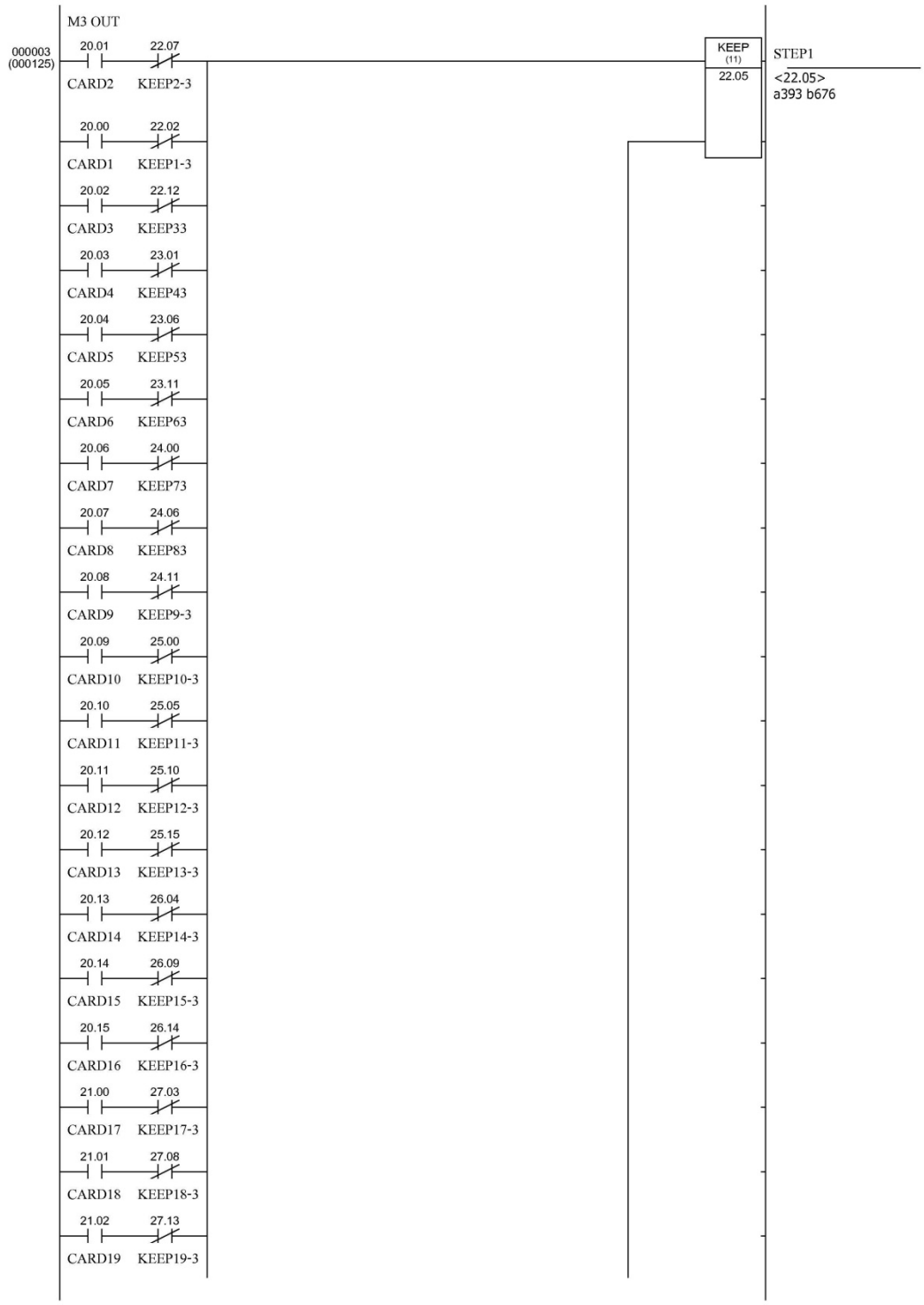
```
                LCALL CLR_BUFFER
                LJMP INITIAL
                ;---------------------------------;
                ;  CLAER RAM BUFFER   ;
                ;---------------------------------;
CLR_BUFFER: MOV R0,#START_POINT_BUFFER
                CLR A
                CLR_BUFFER1: MOV @R0,A
                INC R0
                CJNE R0,#START_POINT_BUFFER+14,CLR_BUFFER1
                RET
;-------------------------------------------------------------------------------
; LCD Initialize โปรแกรมย่อยเพื่อเตรียมการทำงานเริ่มต้น LCD
;-------------------------------------------------------------------------------
INIT_LCD: LCALL DELAY_100ms ; หน่วงเวลา เพื่อให้จอแอลซีดีพร้อมก่อน
                CLR P3.7 ; ขา RS เป็น 0 เพื่อแจ้งให้ทราบว่าเป็นการส่งคำสั่ง
                MOV P0,#00111000B ; กำหนดโหมดเป็นแบบ 8bit
                LCALL LCD_CLK ; ส่งพัลซ์ 1 ลูก ที่ขา Enable
                LCALL DELAY_10ms ; หน่วงเวลาเพื่อให้จอฯพร้อมรับคำสั่งถัดไป
                LCALL LCD_OFF ; เรียกโปรแกรมย่อย ปิดจอแสดงผล
                LCALL LCD_CLR ; เรียกโปรแกรมย่อย ล้างค่าจอแสดงผล
                MOV P0,#00000110B ; กำหนดโหมดการป้อนข้อมูล
                LCALL LCD_CLK ; ส่งพัลซ์ 1 ลูก ที่ขา Enable
                LCALL LCD_HOME ; เรียกโปรแกรมย่อยเพื่อย้ายเคอร์เซอร์ไปยังตำแหน่งแรก
;-------------------------------------------------------------------
; LCD Clear Display โปรแกรมย่อยล้างค่าจอแสดงผล
;-------------------------------------------------------------------
LCD_CLR: CLR P3.7
                MOV P0,#00000001B
                ACALL LCD_CLK
                RET
;-------------------------------------------------------------------------------------
; LCD Return Home โปรแกรมย่อยเพื่อย้ายเคอร์เซอร์ไปยังตำแหน่งแรก
;-------------------------------------------------------------------------------------
LCD_HOME: CLR P3.7
                MOV P0,#00000010B
                ACALL LCD_CLK
                RET
;-------------------------------------------------------------
; LCD Display Off โปรแกรมย่อย ปิดจอแสดงผล
;-------------------------------------------------------------
LCD_OFF: CLR P3.7
                MOV P0,#00001000B
                ACALL LCD_CLK
                RET
;-------------------------------------------------------------------
; LCD Clk โปรแกรมย่อยเพื่อ ส่งพัลซ์ 1 ลูก ที่ขา Enable
;-------------------------------------------------------------------
LCD_CLK: SETB P3.6
                LCALL LCD_DELAY
                CLR P3.6
                LCALL LCD_DELAY
                RET
;-------------------------------------------------------------
; LCD Display On โปรแกรมย่อย เปิดจอแสดงผล
;-------------------------------------------------------------
LCD_ON: CLR P3.7
                MOV P0,#00001100B
                LCALL LCD_CLK
                RET
;-----------------------------------------------------
; LCD Cursor On โปรแกรมย่อย เปิดเคอร์เซอร์
;-----------------------------------------------------
LCD_BLINK: CLR P3.7
                MOV P0,#00001111B
                LCALL LCD_CLK
                RET
```

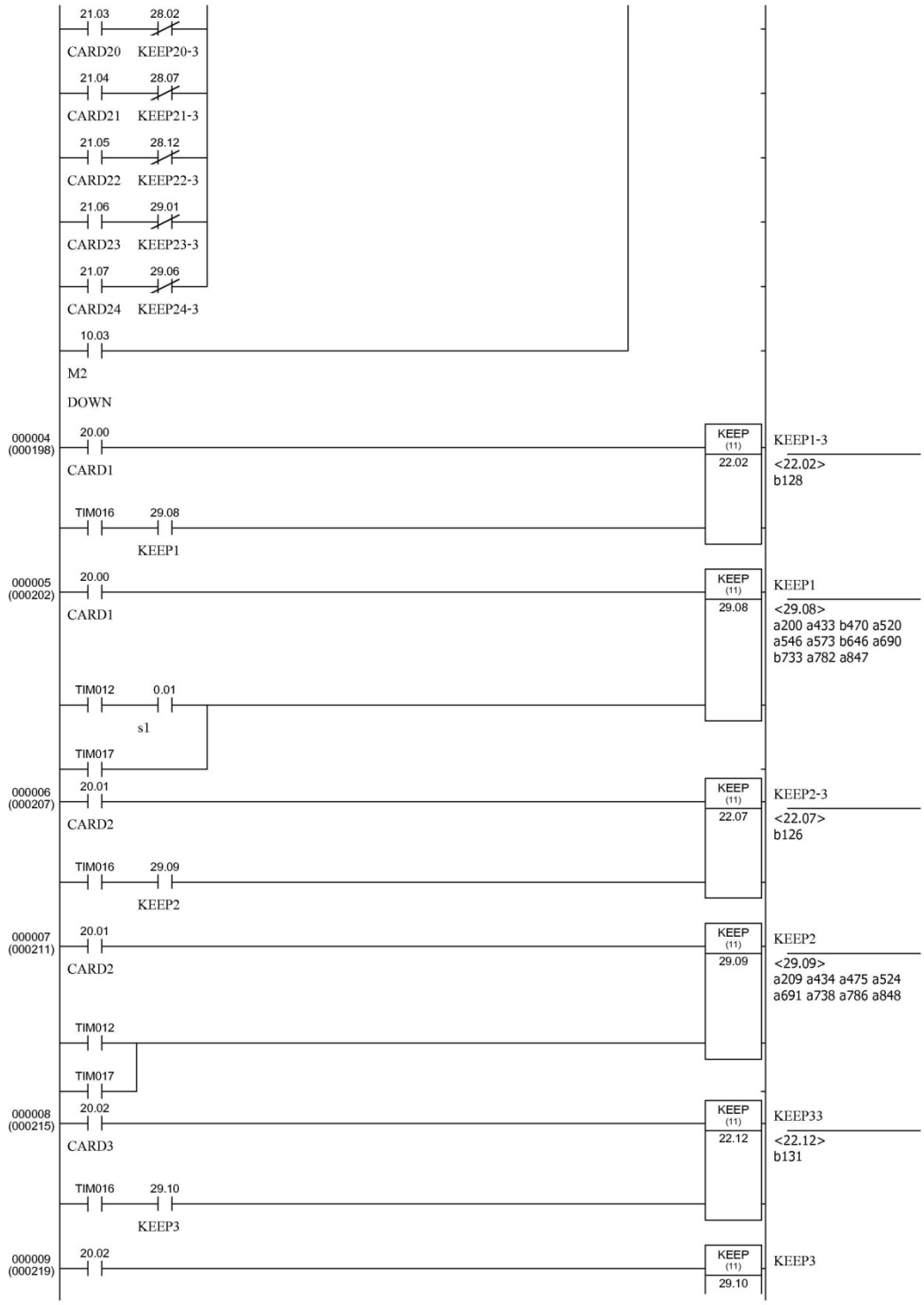
```
;-------------------------------------------------------------------------------
; Set LCD Address โปรแกรมย่อยเพื่อกำหนดตำแหน่งในจอแอลซีดี
;-------------------------------------------------------------------------------
SET_ADDR_LCD: CLR P3.7
              MOV A,LCD_ADDR
              SETB ACC.7
              MOV P0,A
              LCALL LCD_CLK
              RET
;-------------------------------------------------------------------------------
; Write Character to show LCD โปรแกรมย่อยเพื่อแสดงตัวอักษรที่ต้องการ
;-------------------------------------------------------------------------------
WRCHAR_LCD: SETB P3.7
              MOV P0,LCD_DATA
              LCALL LCD_CLK
              LCALL LCD_ON
              RET
;-------------------------------------------------------------------------------
; Write Line of 16 Character from ROM โปรแกรมย่อยเพื่อส่งตัวอักษรจาก ROM ไปจอแอลซีดี
;-------------------------------------------------------------------------------
WRLINE_LCD: MOV R0,#0
              WRLINE_LCD_1: SETB P3.7
              CLR A
              MOVC A,@A+DPTR
              MOV P0,A
              LCALL LCD_CLK
              INC DPTR
              INC R0
              CJNE R0,#16,WRLINE_LCD_1
              LCALL LCD_ON
              RET
;-------------------------------------------------------------------------------
;   DELAY ROUTINE   โปรแกรมย่อยสำหรับหน่วงเวลา
;-------------------------------------------------------------------------------
LCD_DELAY: MOV R7,#002
              LCD_DELAY_1: MOV R6,#0E6H
              LCD_DELAY_2: NOP
              NOP
              DJNZ R6,LCD_DELAY_2
              DJNZ R7,LCD_DELAY_1
              RET
DELAY_10ms: MOV R7,#010
              DELAY_10ms_1: MOV R6,#0E6H
              DELAY_10ms_2: NOP
              NOP
              DJNZ R6,DELAY_10ms_2
              DJNZ R7,DELAY_10ms_1
              RET
DELAY_100ms: MOV R7,#100
              DELAY_100ms_1: MOV R6,#0E6H
              DELAY_100ms_2: NOP
              NOP
              DJNZ R6,DELAY_100ms_2
              DJNZ R7,DELAY_100ms_1
              RET
DELAY_1s: MOV R5,#100
              DELAY_1s_1: LCALL DELAY_10ms
              DJNZ R5,DELAY_1s_1
              RET
DELAY: MOV R2,#150
              DEL1: MOV R3,#80
              DEL2: MOV R4,#80
              DJNZ R4,$
              DJNZ R3,DEL2
              DJNZ R2,DEL1
              RET
;-------------------------------------------------------------------------------
```

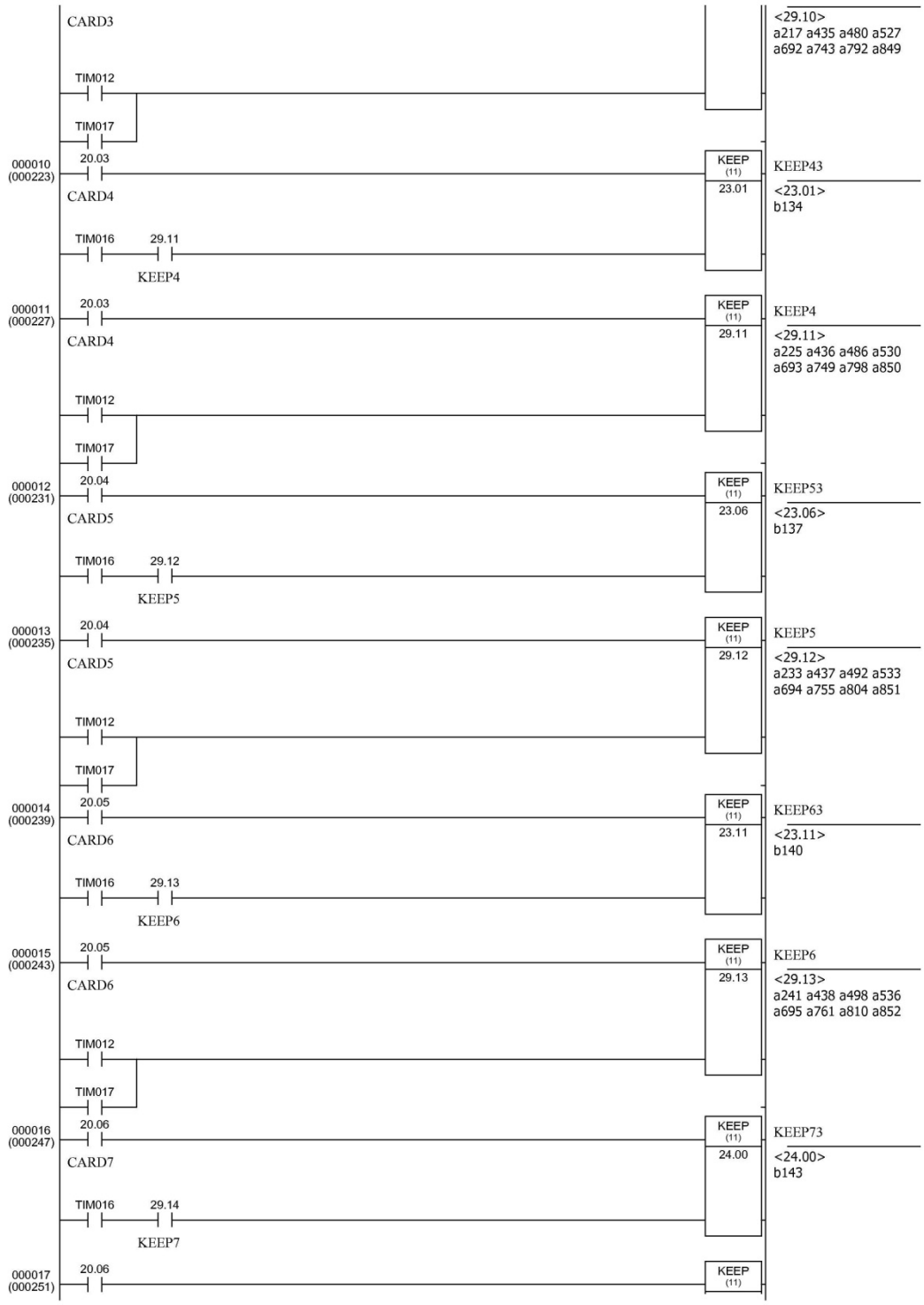
3) โปรแกรมตัวอย่างที่ใช้ในการทดสอบชุดทดลองในหัวข้อ 3.4.2 (เฉพาะ Ladder)

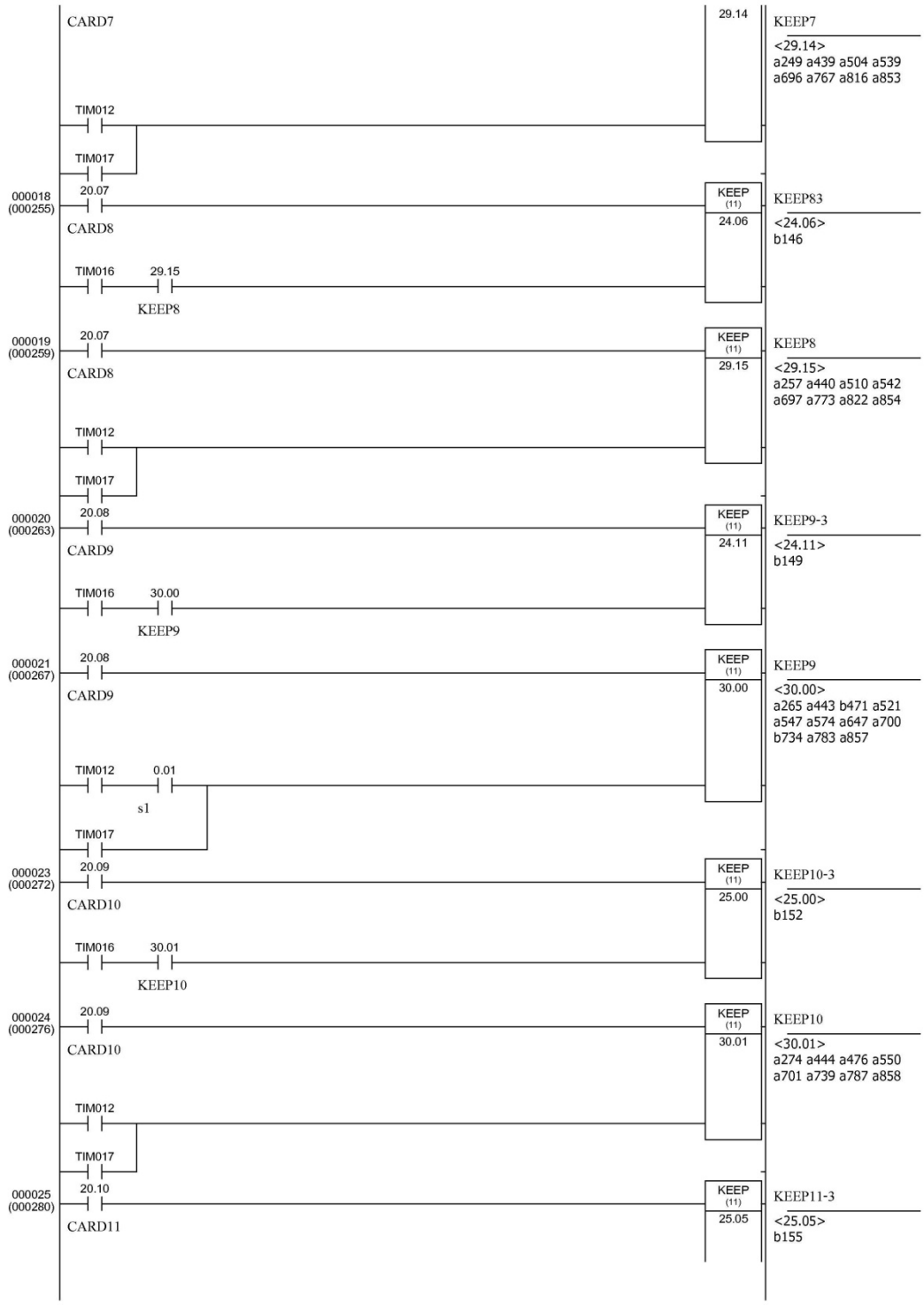


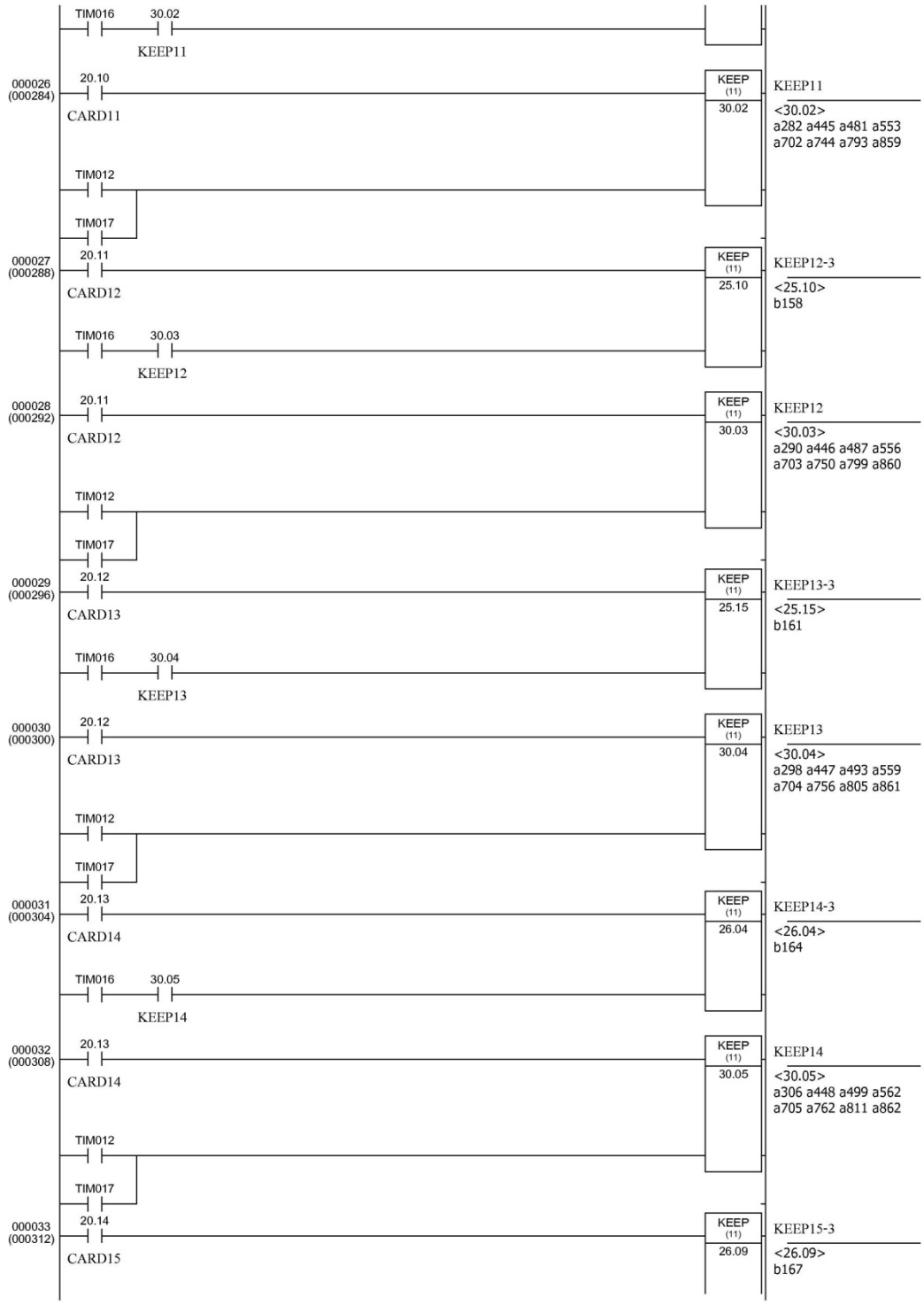

2.00
20.14
<20.13>
a163 a304 a308 a665
CARD15
<20.14>
a166 a312 a316 a666
000001
(000073)
3.00
PROTEC
T
2.07
2.06
2.05
2.04
20.15
CARD16
<20.15>
a169 a320 a324 a667
2.05
2.04
21.00
CARD17
<21.00>
a172 a328 a332 a668
2.04
21.01
CARD18
<21.01>
a175 a337 a341 a669
2.06
2.05
2.04
21.02
CARD19
<21.02>
a178 a345 a349 a670
2.04
21.03
CARD20
<21.03>
a181 a353 a357 a671
2.05
2.04
21.04
CARD21
<21.04>
a184 a361 a365 a672
2.04
21.05
CARD22
<21.05>
a187 a369 a373 a673
2.07
2.06
2.05
2.04
21.06
CARD23
<21.06>
a190 a377 a381 a674
2.04
21.07
CARD24
<21.07>
a193 a385 a389 a675
000002
(000118)
10.00
M1 CW
3.00
PROTECT
<3.00>
b000 b073
10.01
M1 CCW
10.02
M2 UP
10.03
M2
DOWN
10.04
M3 IN
10.05

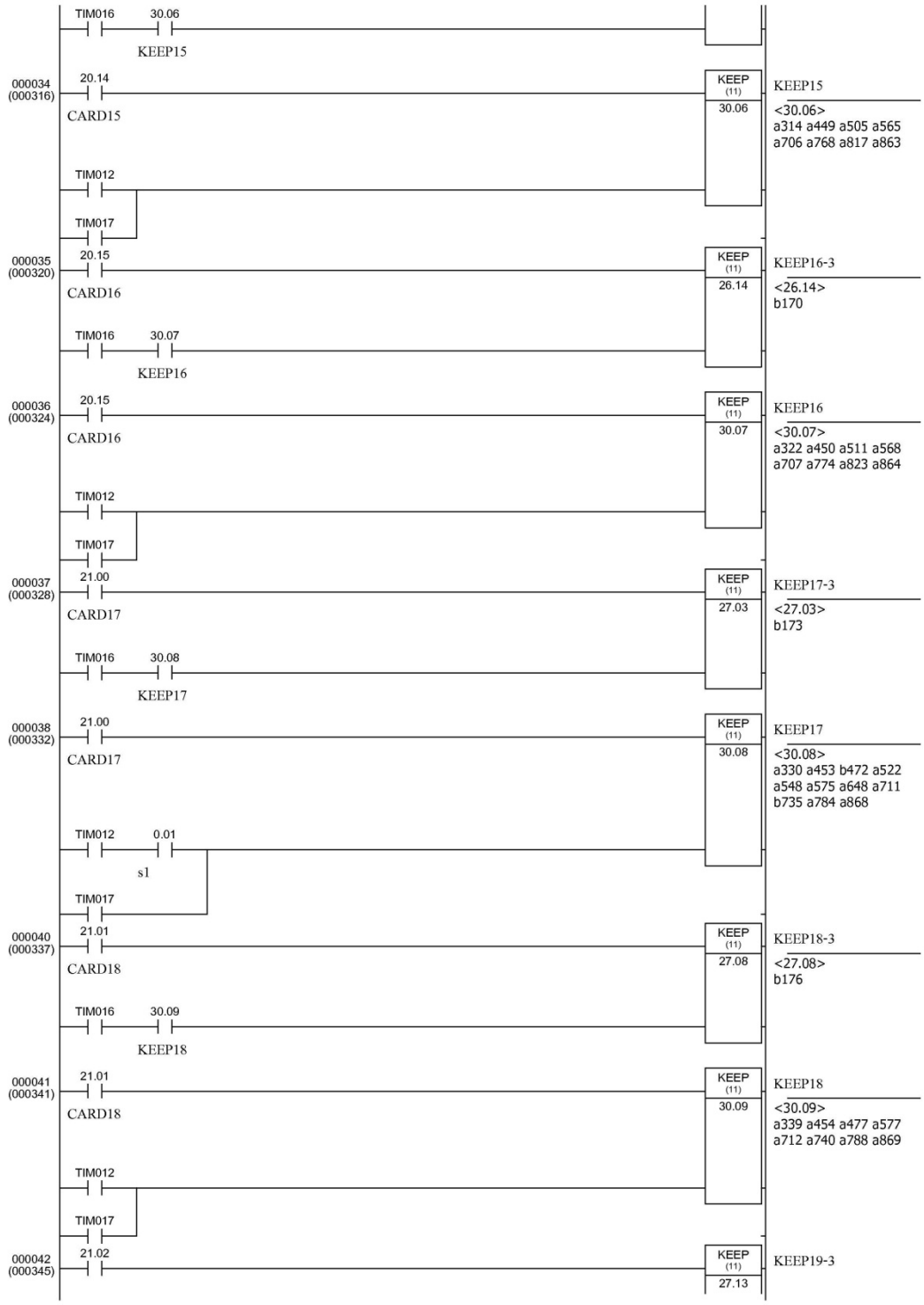

M3 OUT
000003
(000125)
20.01
22.07
CARD2
KEEP2-3
20.00
22.02
CARD1
KEEP1-3
20.02
22.12
CARD3
KEEP33
20.03
23.01
CARD4
KEEP43
20.04
23.06
CARD5
KEEP53
20.05
23.11
CARD6
KEEP63
20.06
24.00
CARD7
KEEP73
20.07
24.06
CARD8
KEEP83
20.08
24.11
CARD9
KEEP9-3
20.09
25.00
CARD10
KEEP10-3
20.10
25.05
CARD11
KEEP11-3
20.11
25.10
CARD12
KEEP12-3
20.12
25.15
CARD13
KEEP13-3
20.13
26.04
CARD14
KEEP14-3
20.14
26.09
CARD15
KEEP15-3
20.15
26.14
CARD16
KEEP16-3
21.00
27.03
CARD17
KEEP17-3
21.01
27.08
CARD18
KEEP18-3
21.02
27.13
CARD19
KEEP19-3
KEEP
(11)
22.05
STEP1
<22.05>
a393 b676

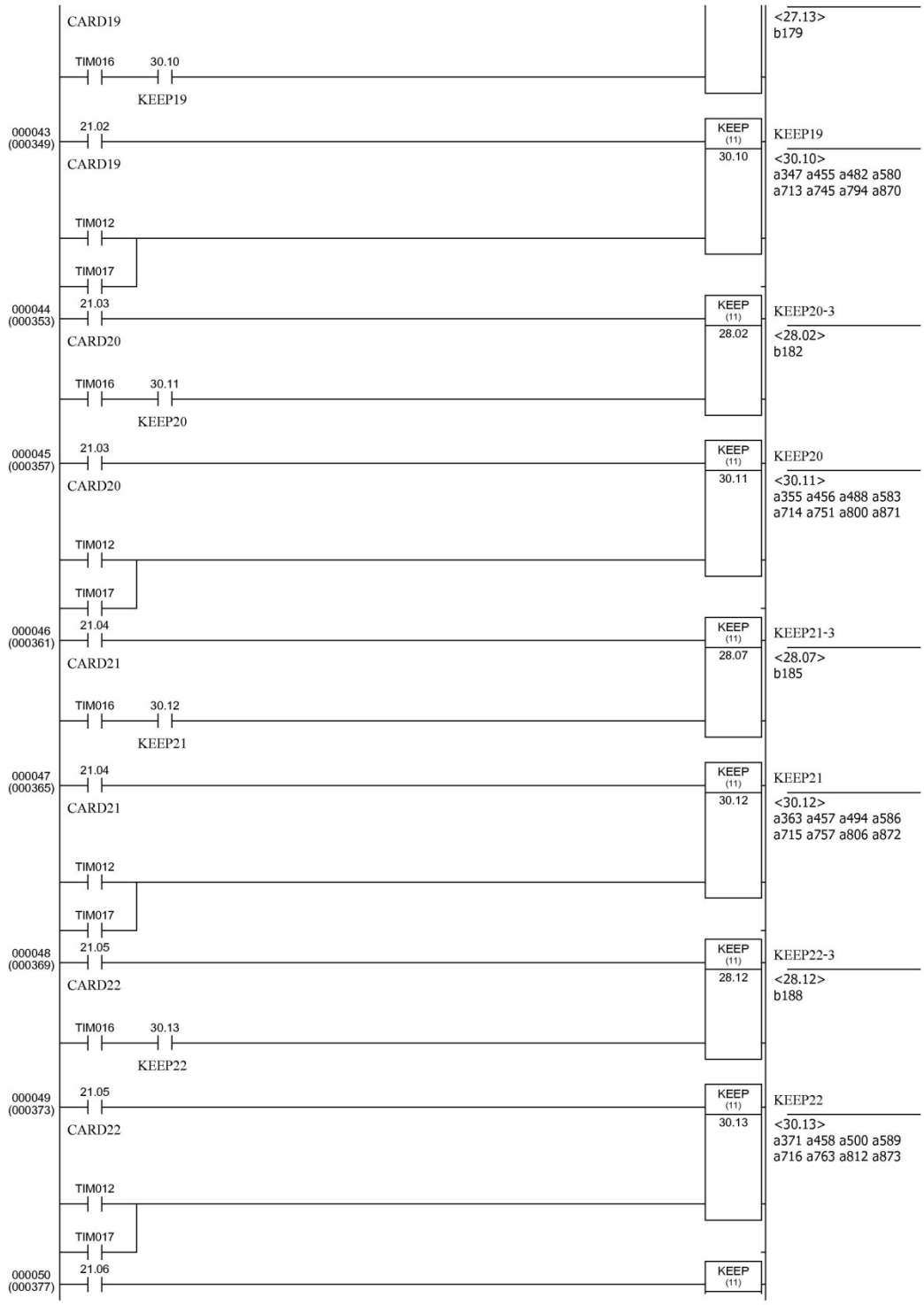

21.03
28.02
CARD20
KEEP20-3
21.04
28.07
CARD21
KEEP21-3
21.05
28.12
CARD22
KEEP22-3
21.06
29.01
CARD23
KEEP23-3
21.07
29.06
CARD24
KEEP24-3
10.03
M2
DOWN
000004
(000198)
20.00
CARD1
KEEP
(11)
22.02
KEEP1-3
<22.02>
b128
TIM016
29.08
KEEP1
000005
(000202)
20.00
CARD1
KEEP
(11)
29.08
KEEP1
<29.08>
a200 a433 b470 a520
a546 a573 b646 a690
b733 a782 a847
TIM012
0.01
s1
TIM017
000006
(000207)
20.01
CARD2
KEEP
(11)
22.07
KEEP2-3
<22.07>
b126
TIM016
29.09
KEEP2
000007
(000211)
20.01
CARD2
KEEP
(11)
29.09
KEEP2
<29.09>
a209 a434 a475 a524
a691 a738 a786 a848
TIM012
TIM017
000008
(000215)
20.02
CARD3
KEEP
(11)
22.12
KEEP33
<22.12>
b131
TIM016
29.10
KEEP3
000009
(000219)
20.02
KEEP
(11)
29.10
KEEP3

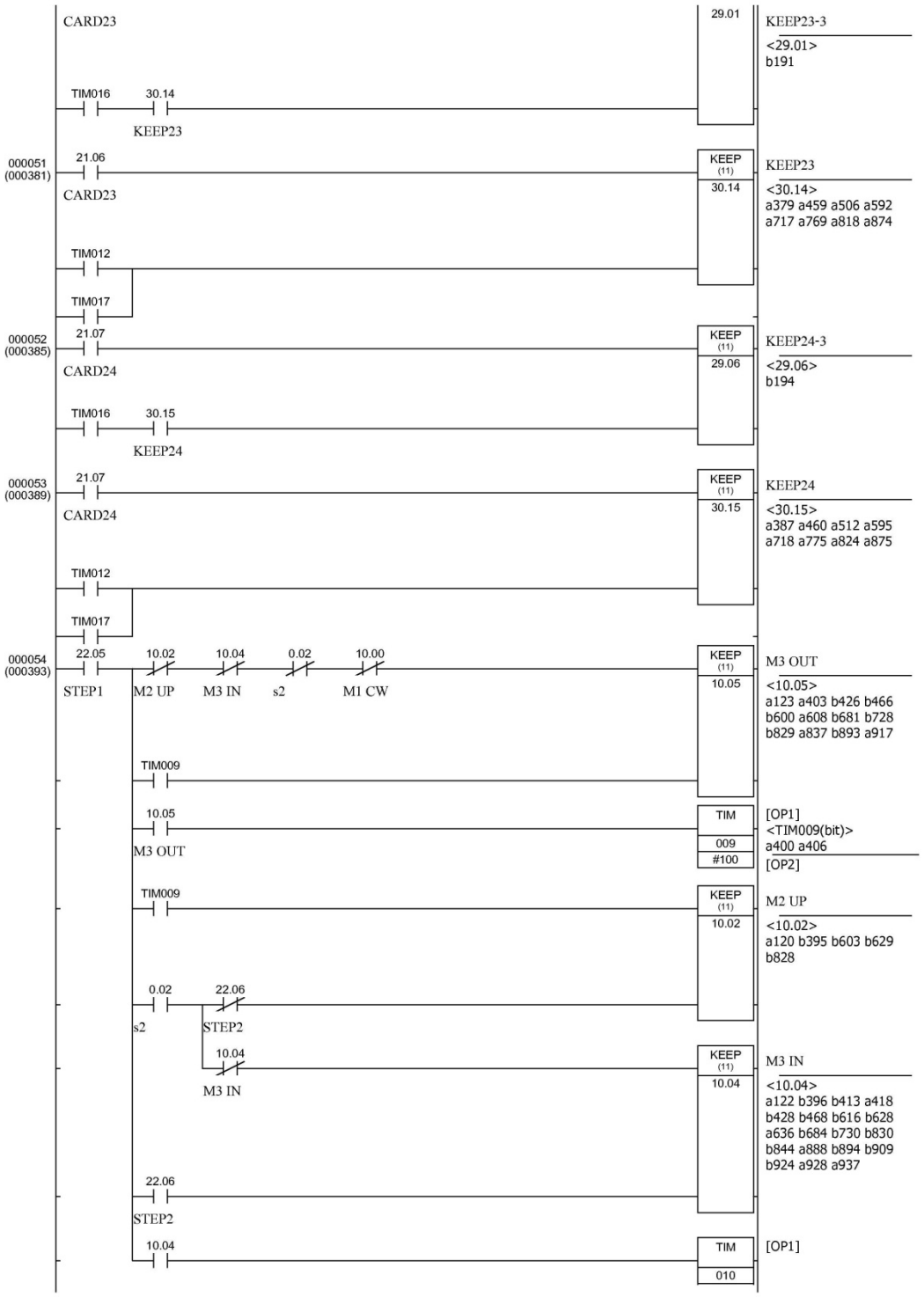

CARD3
<29.10>
a217 a435 a480 a527
a692 a743 a792 a849
TIM012
TIM017
000010
(000223)
20.03
CARD4
KEEP
(11)
23.01
KEEP43
<23.01>
b134
TIM016
29.11
KEEP4
000011
(000227)
20.03
CARD4
KEEP
(11)
29.11
KEEP4
<29.11>
a225 a436 a486 a530
a693 a749 a798 a850
TIM012
TIM017
000012
(000231)
20.04
CARD5
KEEP
(11)
23.06
KEEP53
<23.06>
b137
TIM016
29.12
KEEP5
000013
(000235)
20.04
CARD5
KEEP
(11)
29.12
KEEP5
<29.12>
a233 a437 a492 a533
a694 a755 a804 a851
TIM012
TIM017
000014
(000239)
20.05
CARD6
KEEP
(11)
23.11
KEEP63
<23.11>
b140
TIM016
29.13
KEEP6
000015
(000243)
20.05
CARD6
KEEP
(11)
29.13
KEEP6
<29.13>
a241 a438 a498 a536
a695 a761 a810 a852
TIM012
TIM017
000016
(000247)
20.06
CARD7
KEEP
(11)
24.00
KEEP73
<24.00>
b143
TIM016
29.14
KEEP7
000017
(000251)
20.06
KEEP
(11)

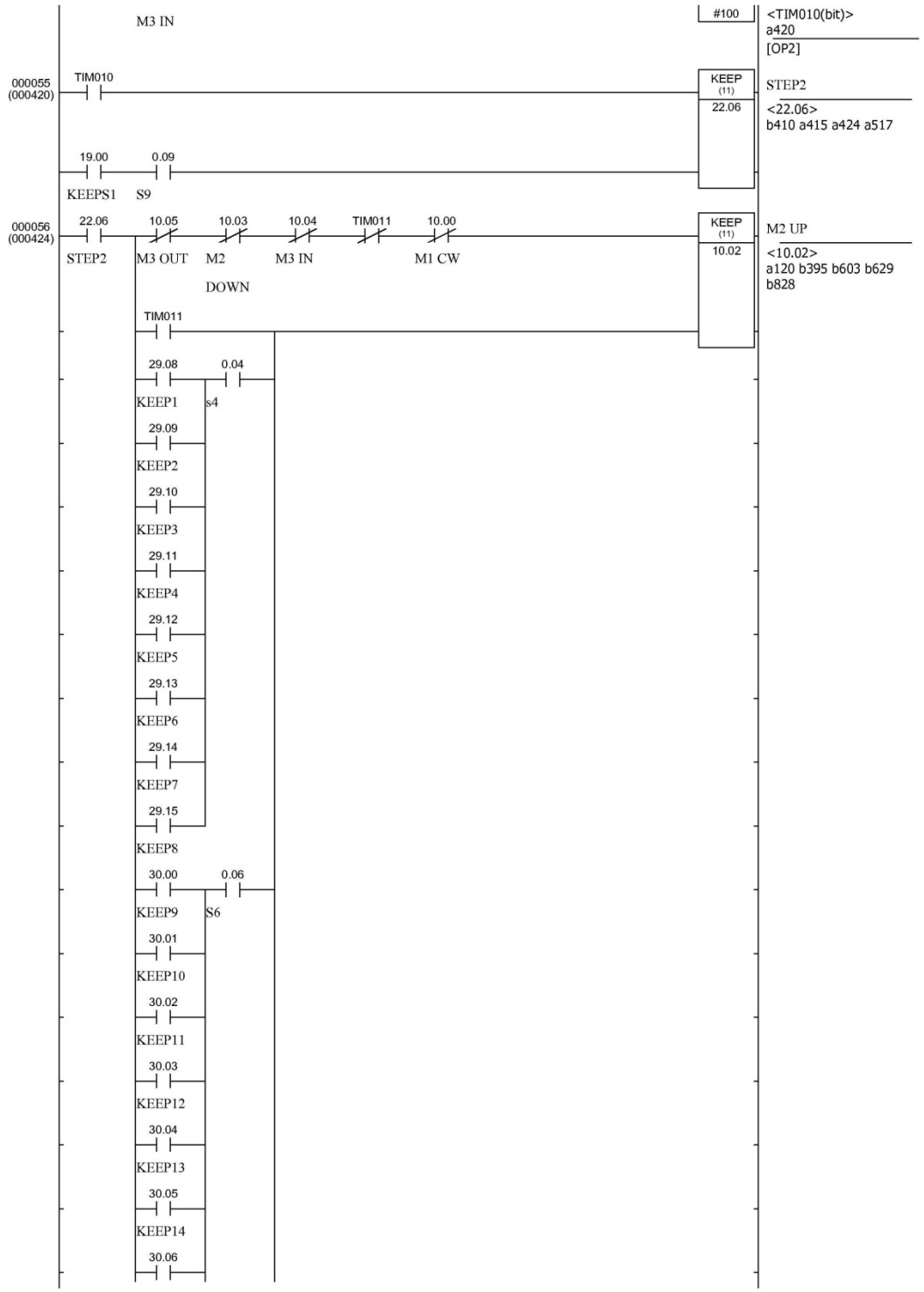

CARD7
29.14
KEEP7
<29.14>
a249 a439 a504 a539
a696 a767 a816 a853
TIM012
TIM017
000018
(000255)
20.07
CARD8
KEEP
(11)
24.06
KEEP83
<24.06>
b146
TIM016
29.15
KEEP8
000019
(000259)
20.07
CARD8
KEEP
(11)
29.15
KEEP8
<29.15>
a257 a440 a510 a542
a697 a773 a822 a854
TIM012
TIM017
000020
(000263)
20.08
CARD9
KEEP
(11)
24.11
KEEP9-3
<24.11>
b149
TIM016
30.00
KEEP9
000021
(000267)
20.08
CARD9
KEEP
(11)
30.00
KEEP9
<30.00>
a265 a443 b471 a521
a547 a574 a647 a700
b734 a783 a857
TIM012
0.01
s1
TIM017
000023
(000272)
20.09
CARD10
KEEP
(11)
25.00
KEEP10-3
<25.00>
b152
TIM016
30.01
KEEP10
000024
(000276)
20.09
CARD10
KEEP
(11)
30.01
KEEP10
<30.01>
a274 a444 a476 a550
a701 a739 a787 a858
TIM012
TIM017
000025
(000280)
20.10
CARD11
KEEP
(11)
25.05
KEEP11-3
<25.05>
b155

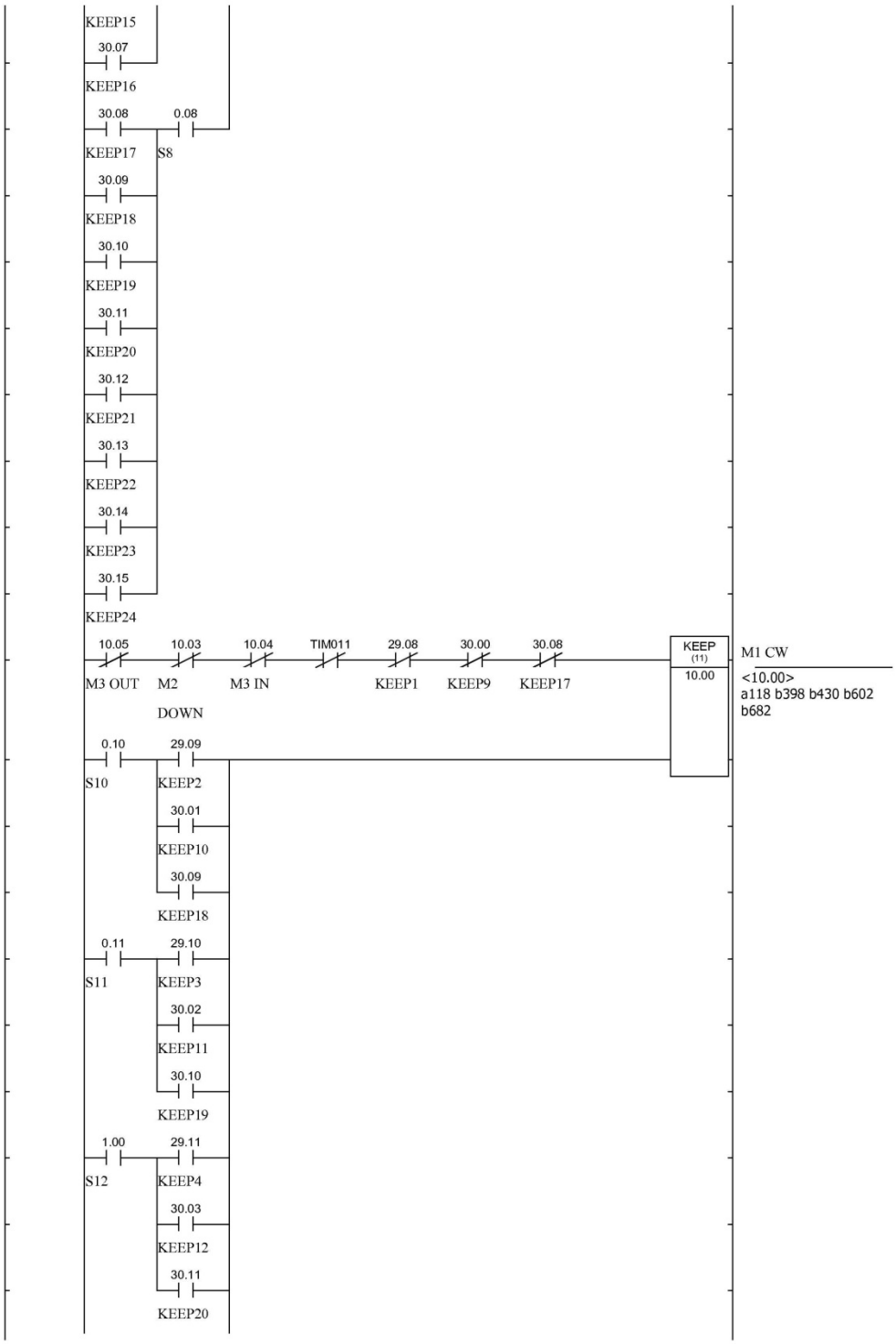

TIM016
30.02
KEEP11
000026
(000284)
20.10
CARD11
KEEP
(11)
30.02
KEEP11
<30.02>
a282 a445 a481 a553
a702 a744 a793 a859
TIM012
TIM017
000027
(000288)
20.11
CARD12
KEEP
(11)
25.10
KEEP12-3
<25.10>
b158
TIM016
30.03
KEEP12
000028
(000292)
20.11
CARD12
KEEP
(11)
30.03
KEEP12
<30.03>
a290 a446 a487 a556
a703 a750 a799 a860
TIM012
TIM017
000029
(000296)
20.12
CARD13
KEEP
(11)
25.15
KEEP13-3
<25.15>
b161
TIM016
30.04
KEEP13
000030
(000300)
20.12
CARD13
KEEP
(11)
30.04
KEEP13
<30.04>
a298 a447 a493 a559
a704 a756 a805 a861
TIM012
TIM017
000031
(000304)
20.13
CARD14
KEEP
(11)
26.04
KEEP14-3
<26.04>
b164
TIM016
30.05
KEEP14
000032
(000308)
20.13
CARD14
KEEP
(11)
30.05
KEEP14
<30.05>
a306 a448 a499 a562
a705 a762 a811 a862
TIM012
TIM017
000033
(000312)
20.14
CARD15
KEEP
(11)
26.09
KEEP15-3
<26.09>
b167

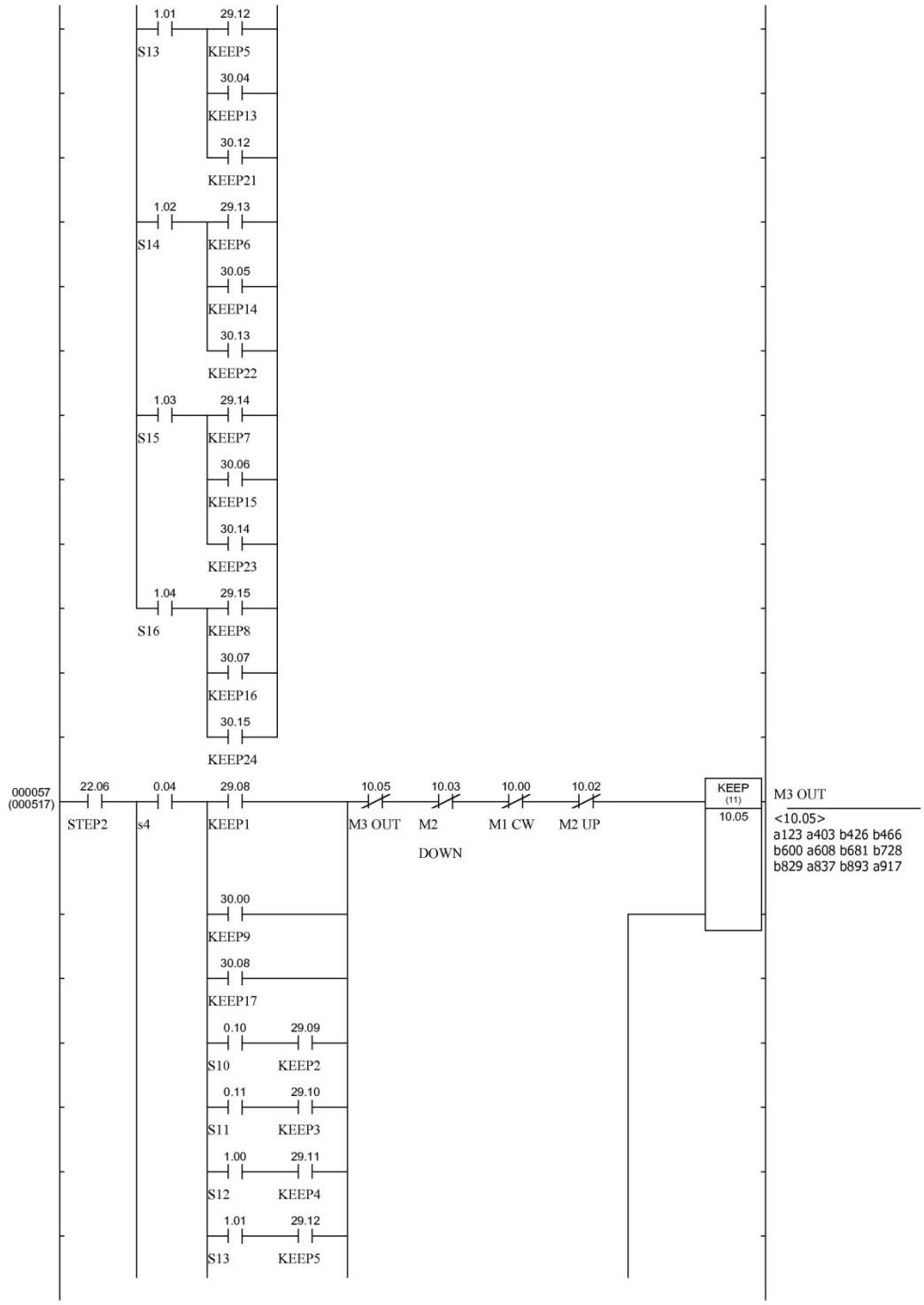

TIM016
30.06
KEEP15
000034
(000316)
20.14
CARD15
KEEP
(11)
30.06
KEEP15
<30.06>
a314 a449 a505 a565
a706 a768 a817 a863
TIM012
TIM017
000035
(000320)
20.15
CARD16
KEEP
(11)
26.14
KEEP16-3
<26.14>
b170
TIM016
30.07
KEEP16
000036
(000324)
20.15
CARD16
KEEP
(11)
30.07
KEEP16
<30.07>
a322 a450 a511 a568
a707 a774 a823 a864
TIM012
TIM017
000037
(000328)
21.00
CARD17
KEEP
(11)
27.03
KEEP17-3
<27.03>
b173
TIM016
30.08
KEEP17
000038
(000332)
21.00
CARD17
KEEP
(11)
30.08
KEEP17
<30.08>
a330 a453 b472 a522
a548 a575 a648 a711
b735 a784 a868
TIM012
0.01
s1
TIM017
000040
(000337)
21.01
CARD18
KEEP
(11)
27.08
KEEP18-3
<27.08>
b176
TIM016
30.09
KEEP18
000041
(000341)
21.01
CARD18
KEEP
(11)
30.09
KEEP18
<30.09>
a339 a454 a477 a577
a712 a740 a788 a869
TIM012
TIM017
000042
(000345)
21.02
KEEP
(11)
27.13
KEEP19-3

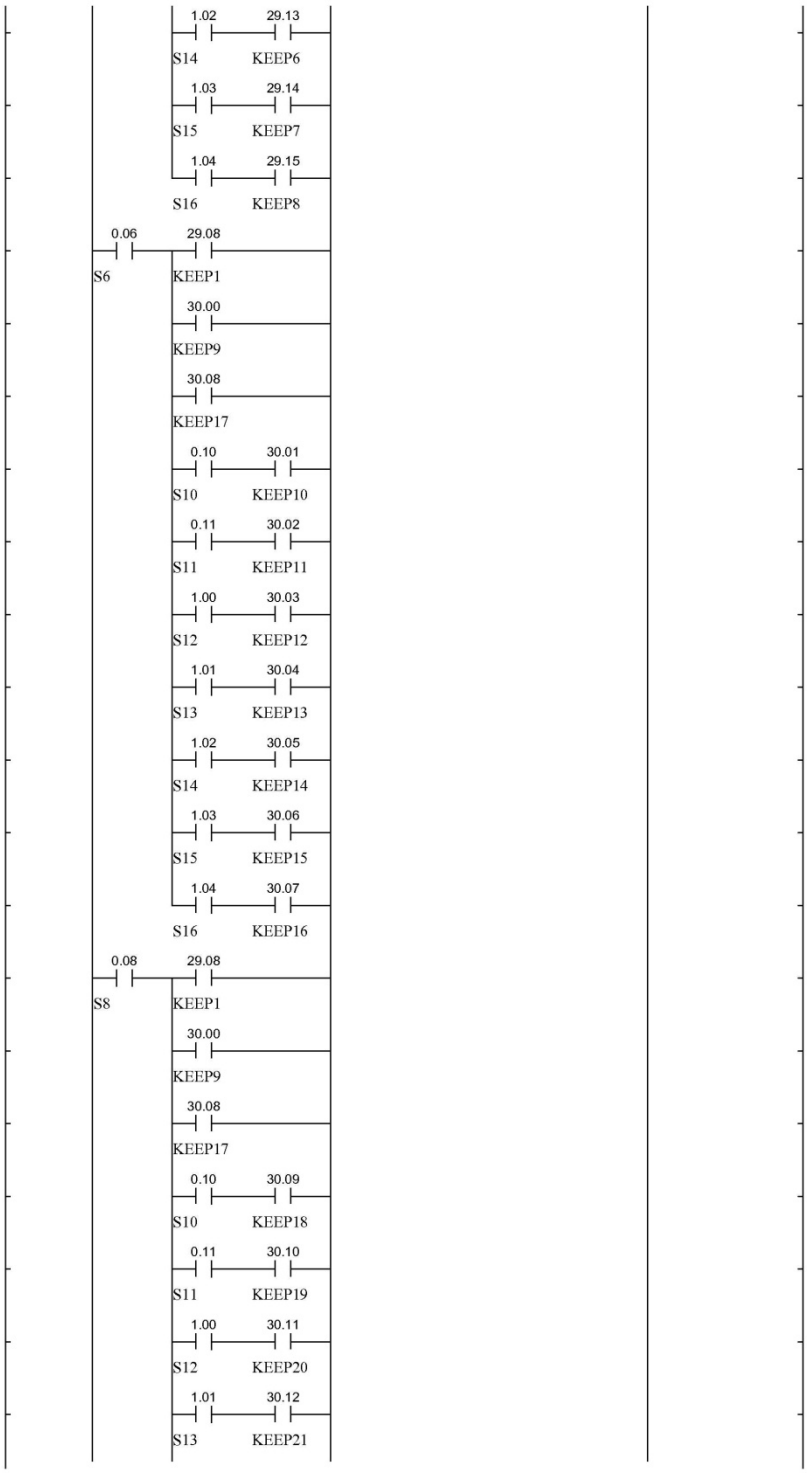

CARD19
TIM016
30.10
KEEP19
000043
(000349)
21.02
CARD19
KEEP
(11)
30.10
KEEP19
<30.10>
a347 a455 a482 a580
a713 a745 a794 a870
TIM012
TIM017
000044
(000353)
21.03
CARD20
KEEP
(11)
28.02
KEEP20-3
<28.02>
b182
TIM016
30.11
KEEP20
000045
(000357)
21.03
CARD20
KEEP
(11)
30.11
KEEP20
<30.11>
a355 a456 a488 a583
a714 a751 a800 a871
TIM012
TIM017
000046
(000361)
21.04
CARD21
KEEP
(11)
28.07
KEEP21-3
<28.07>
b185
TIM016
30.12
KEEP21
000047
(000365)
21.04
CARD21
KEEP
(11)
30.12
KEEP21
<30.12>
a363 a457 a494 a586
a715 a757 a806 a872
TIM012
TIM017
000048
(000369)
21.05
CARD22
KEEP
(11)
28.12
KEEP22-3
<28.12>
b188
TIM016
30.13
KEEP22
000049
(000373)
21.05
CARD22
KEEP
(11)
30.13
KEEP22
<30.13>
a371 a458 a500 a589
a716 a763 a812 a873
TIM012
TIM017
000050
(000377)
21.06
KEEP
(11)

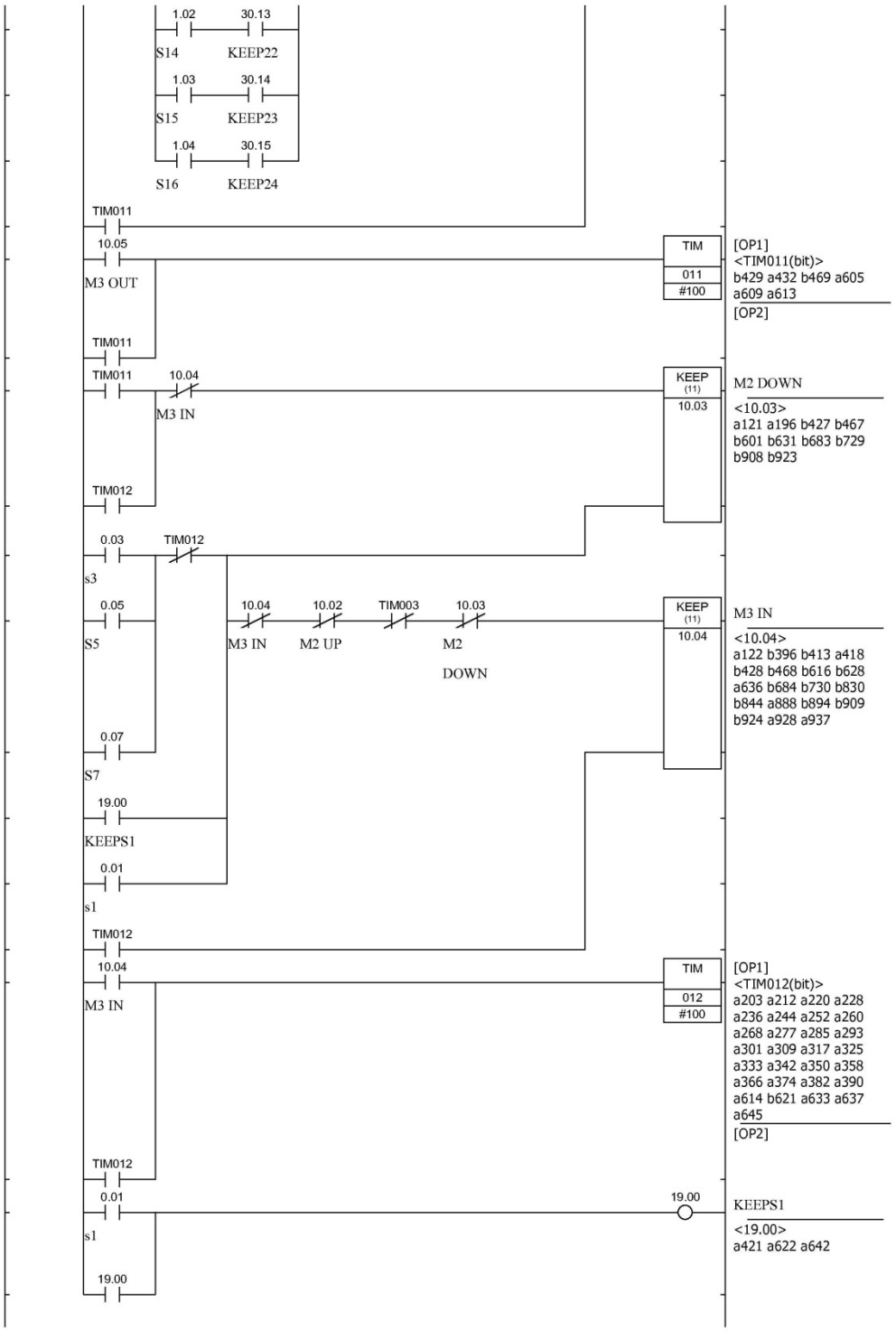

CARD23
29.01
KEEP23-3
<29.01>
b191
TIM016
30.14
KEEP23
000051
(000381)
21.06
CARD23
KEEP
(11)
30.14
KEEP23
<30.14>
a379 a459 a506 a592
a717 a769 a818 a874
TIM012
TIM017
000052
(000385)
21.07
CARD24
KEEP
(11)
29.06
KEEP24-3
<29.06>
b194
TIM016
30.15
KEEP24
000053
(000389)
21.07
CARD24
KEEP
(11)
30.15
KEEP24
<30.15>
a387 a460 a512 a595
a718 a775 a824 a875
TIM012
TIM017
000054
(000393)
22.05
STEP1
10.02
M2 UP
10.04
M3 IN
0.02
s2
10.00
M1 CW
KEEP
(11)
10.05
M3 OUT
<10.05>
a123 a403 b426 b466
b600 a608 b681 b728
b829 a837 b893 a917
TIM009
10.05
M3 OUT
TIM
009
#100
[OP1]
<TIM009(bit)>
a400 a406
[OP2]
TIM009
KEEP
(11)
10.02
M2 UP
<10.02>
a120 b395 b603 b629
b828
0.02
s2
22.06
STEP2
10.04
M3 IN
KEEP
(11)
10.04
M3 IN
<10.04>
a122 b396 b413 a418
b428 b468 b616 b628
a636 b684 b730 b830
b844 a888 b894 b909
b924 a928 a937
22.06
STEP2
10.04
TIM
010
[OP1]

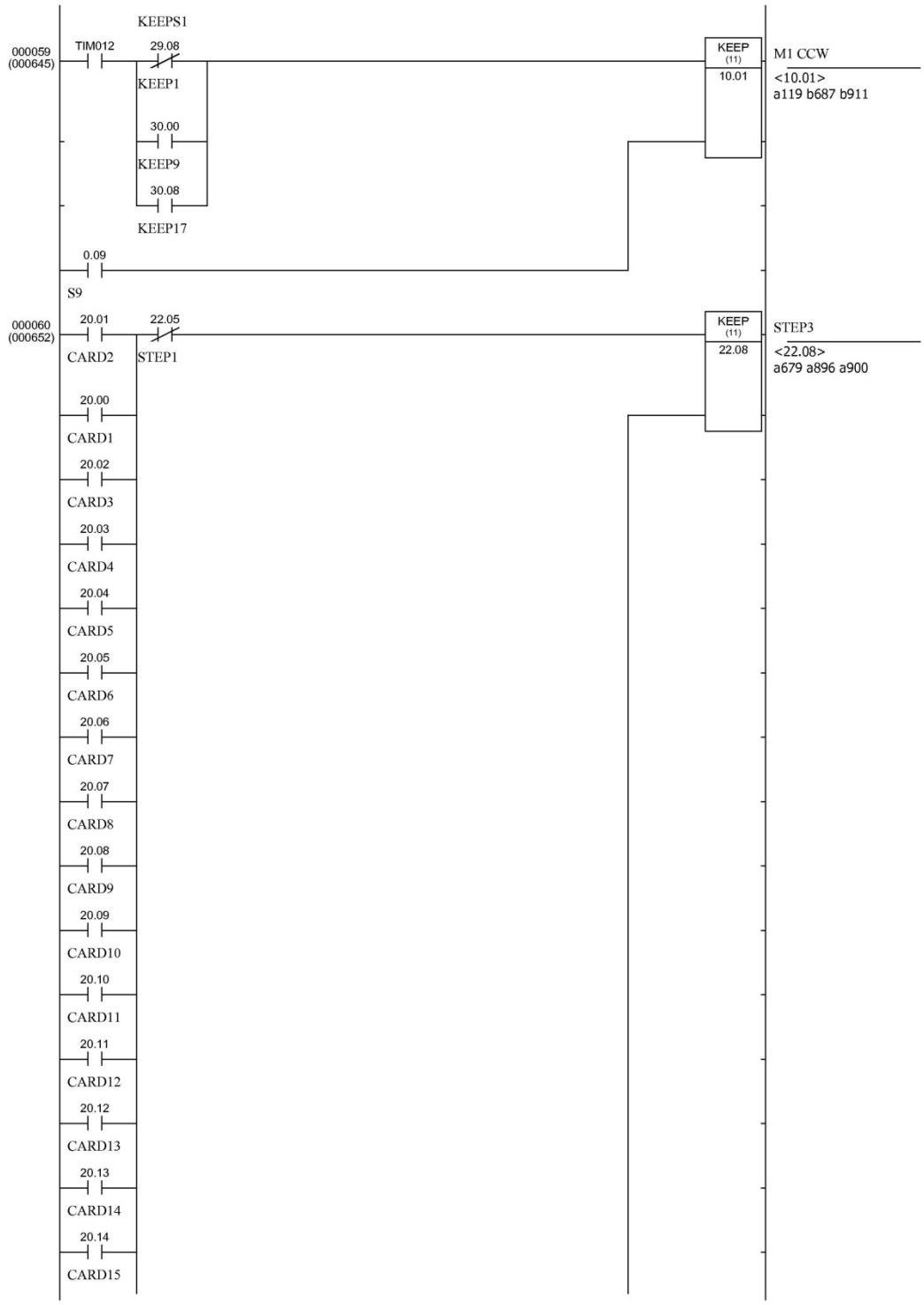

M3 IN
#100
<TIM010(bit)>
a420
[OP2]
000055
(000420)
TIM010
KEEP
(11)
22.06
STEP2
<22.06>
b410 a415 a424 a517
19.00
KEEPS1
0.09
S9
000056
(000424)
22.06
STEP2
10.05
M3 OUT
10.03
M2
DOWN
10.04
M3 IN
TIM011
10.00
M1 CW
KEEP
(11)
10.02
M2 UP
<10.02>
a120 b395 b603 b629
b828
TIM011
29.08
KEEP1
0.04
s4
29.09
KEEP2
29.10
KEEP3
29.11
KEEP4
29.12
KEEP5
29.13
KEEP6
29.14
KEEP7
29.15
KEEP8
30.00
KEEP9
0.06
S6
30.01
KEEP10
30.02
KEEP11
30.03
KEEP12
30.04
KEEP13
30.05
KEEP14
30.06

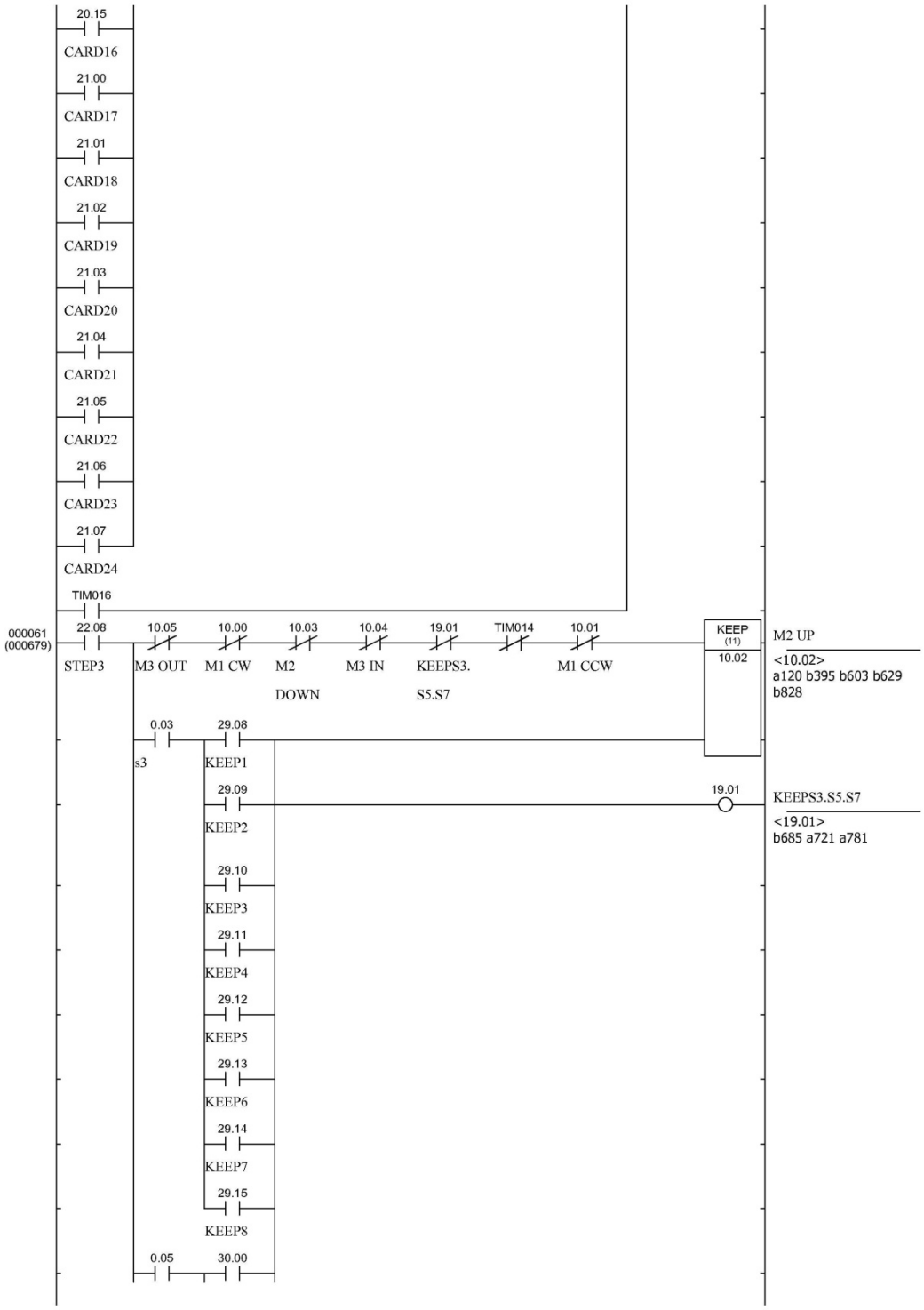

KEEP15
30.07
KEEP16
30.08
KEEP17
0.08
S8
30.09
KEEP18
30.10
KEEP19
30.11
KEEP20
30.12
KEEP21
30.13
KEEP22
30.14
KEEP23
30.15
KEEP24
10.05
M3 OUT
10.03
M2
DOWN
10.04
M3 IN
TIM011
29.08
KEEP1
30.00
KEEP9
30.08
KEEP17
KEEP
(11)
10.00
M1 CW
<10.00>
a118 b398 b430 b602
b682
0.10
S10
29.09
KEEP2
30.01
KEEP10
30.09
KEEP18
0.11
S11
29.10
KEEP3
30.02
KEEP11
30.10
KEEP19
1.00
S12
29.11
KEEP4
30.03
KEEP12
30.11
KEEP20

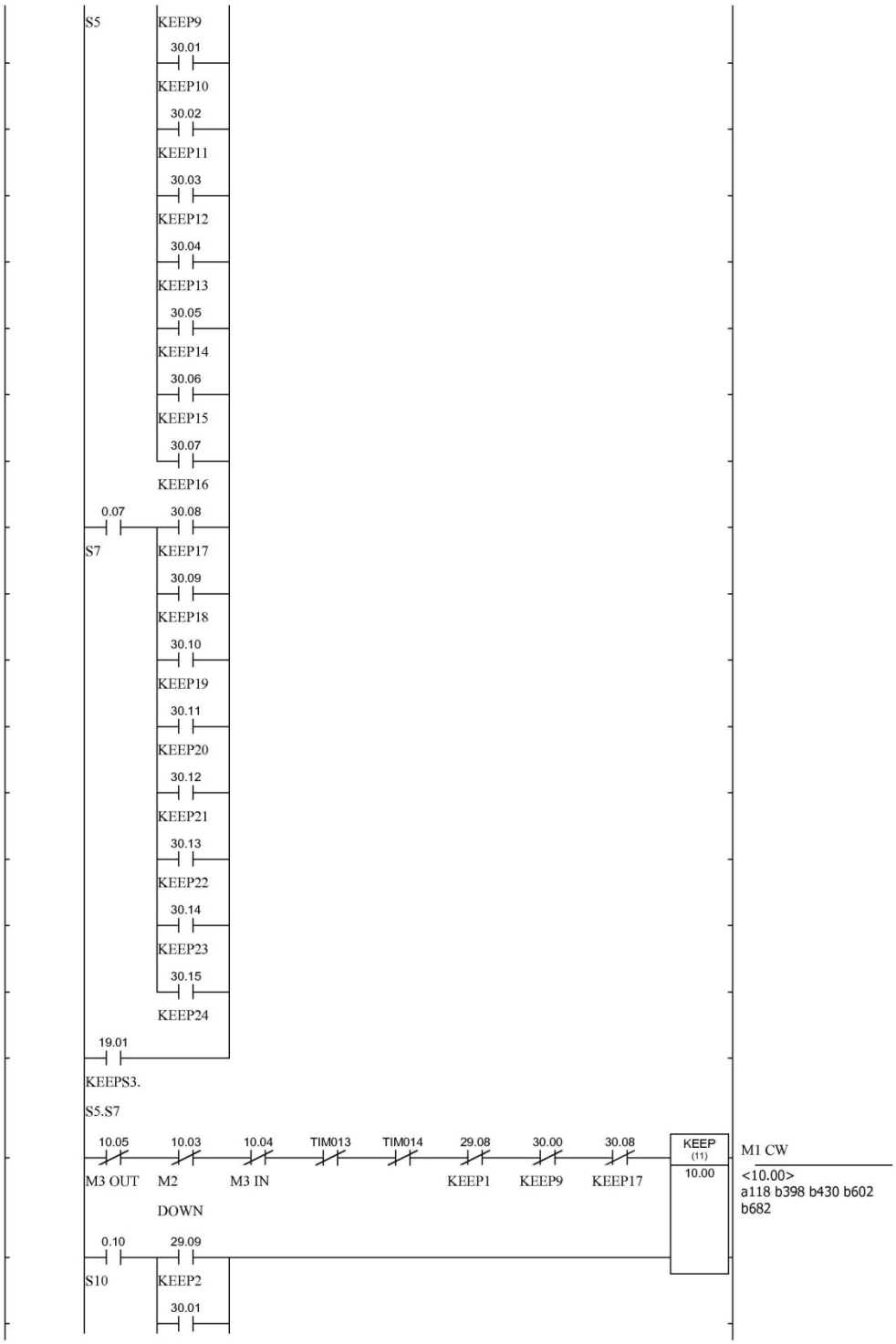

1.01
29.12
S13
KEEP5
30.04
KEEP13
30.12
KEEP21
1.02
29.13
S14
KEEP6
30.05
KEEP14
30.13
KEEP22
1.03
29.14
S15
KEEP7
30.06
KEEP15
30.14
KEEP23
1.04
29.15
S16
KEEP8
30.07
KEEP16
30.15
KEEP24
000057
(000517)
22.06
STEP2
0.04
s4
29.08
KEEP1
10.05
M3 OUT
10.03
M2
DOWN
10.00
M1 CW
10.02
M2 UP
KEEP
(11)
10.05
M3 OUT
<10.05>
a123 a403 b426 b466
b600 a608 b681 b728
b829 a837 b893 a917
30.00
KEEP9
30.08
KEEP17
0.10
29.09
S10
KEEP2
0.11
29.10
S11
KEEP3
1.00
29.11
S12
KEEP4
1.01
29.12
S13
KEEP5

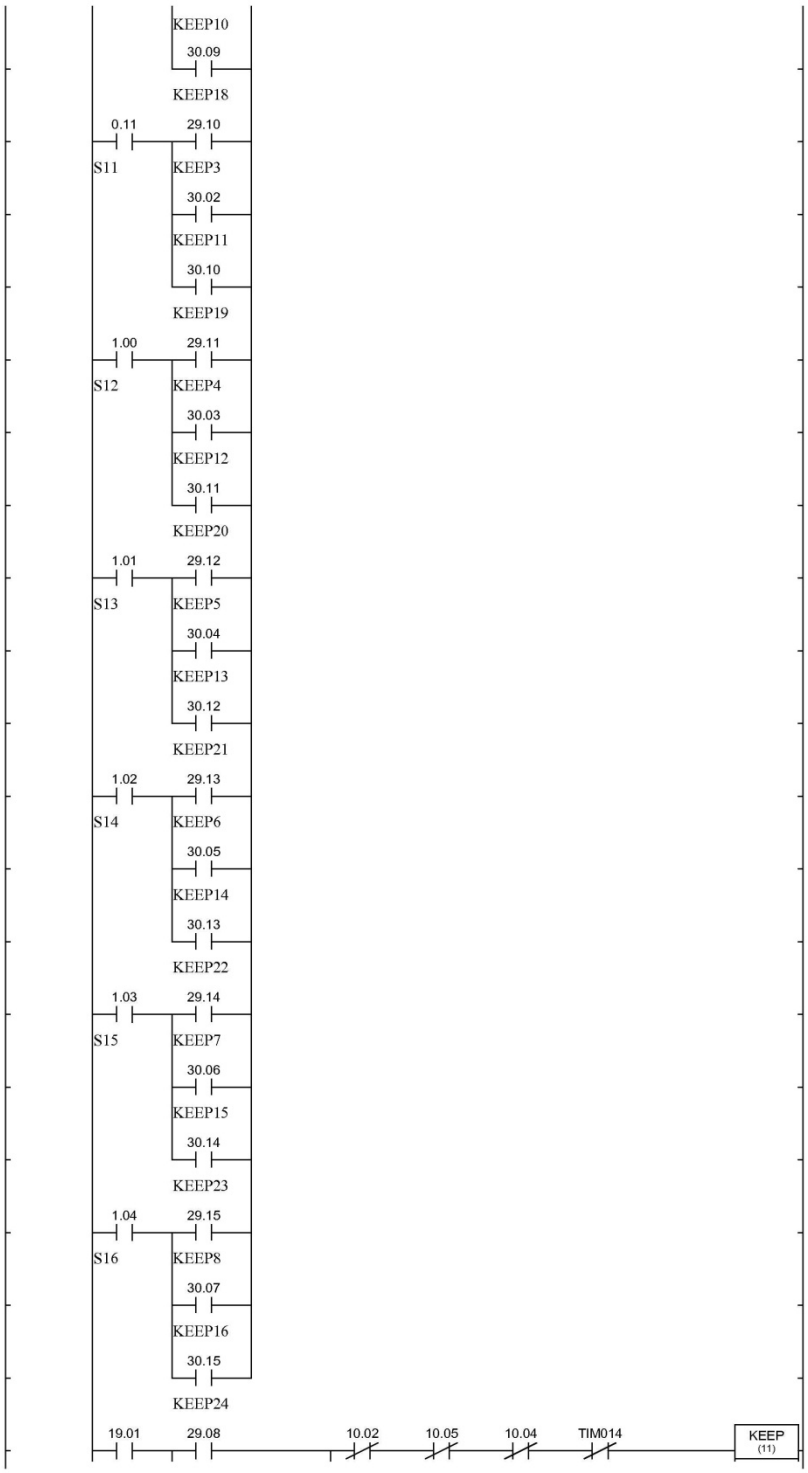

1.02
29.13
S14
KEEP6
1.03
29.14
S15
KEEP7
1.04
29.15
S16
KEEP8
0.06
29.08
S6
KEEP1
30.00
KEEP9
30.08
KEEP17
0.10
30.01
S10
KEEP10
0.11
30.02
S11
KEEP11
1.00
30.03
S12
KEEP12
1.01
30.04
S13
KEEP13
1.02
30.05
S14
KEEP14
1.03
30.06
S15
KEEP15
1.04
30.07
S16
KEEP16
0.08
29.08
S8
KEEP1
30.00
KEEP9
30.08
KEEP17
0.10
30.09
S10
KEEP18
0.11
30.10
S11
KEEP19
1.00
30.11
S12
KEEP20
1.01
30.12
S13
KEEP21

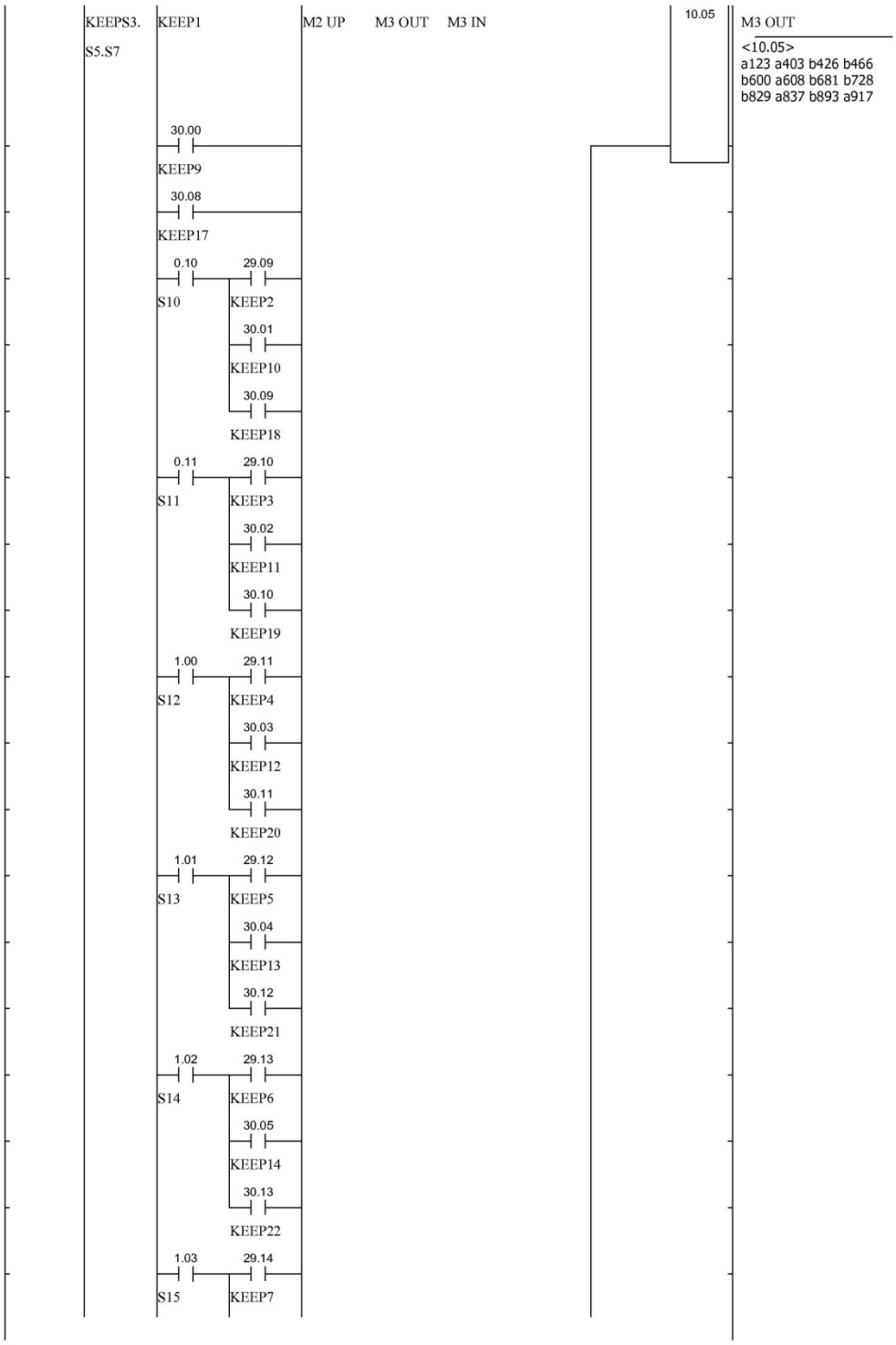

1.02
30.13
S14
KEEP22
1.03
30.14
S15
KEEP23
1.04
30.15
S16
KEEP24
TIM011
10.05
M3 OUT
TIM
011
#100
[OP1]
<TIM011(bit)>
b429 a432 b469 a605
a609 a613
[OP2]
TIM011
TIM011
10.04
M3 IN
KEEP
(11)
10.03
M2 DOWN
<10.03>
a121 a196 b427 b467
b601 b631 b683 b729
b908 b923
TIM012
0.03
TIM012
s3
0.05
S5
10.04
M3 IN
10.02
M2 UP
TIM003
10.03
M2
DOWN
KEEP
(11)
10.04
M3 IN
<10.04>
a122 b396 b413 a418
b428 b468 b616 b628
a636 b684 b730 b830
b844 a888 b894 b909
b924 a928 a937
0.07
S7
19.00
KEEPS1
0.01
s1
TIM012
10.04
M3 IN
TIM
012
#100
[OP1]
<TIM012(bit)>
a203 a212 a220 a228
a236 a244 a252 a260
a268 a277 a285 a293
a301 a309 a317 a325
a333 a342 a350 a358
a366 a374 a382 a390
a614 b621 a633 a637
a645
[OP2]
TIM012
0.01
s1
19.00
19.00
KEEPS1
<19.00>
a421 a622 a642
19.00

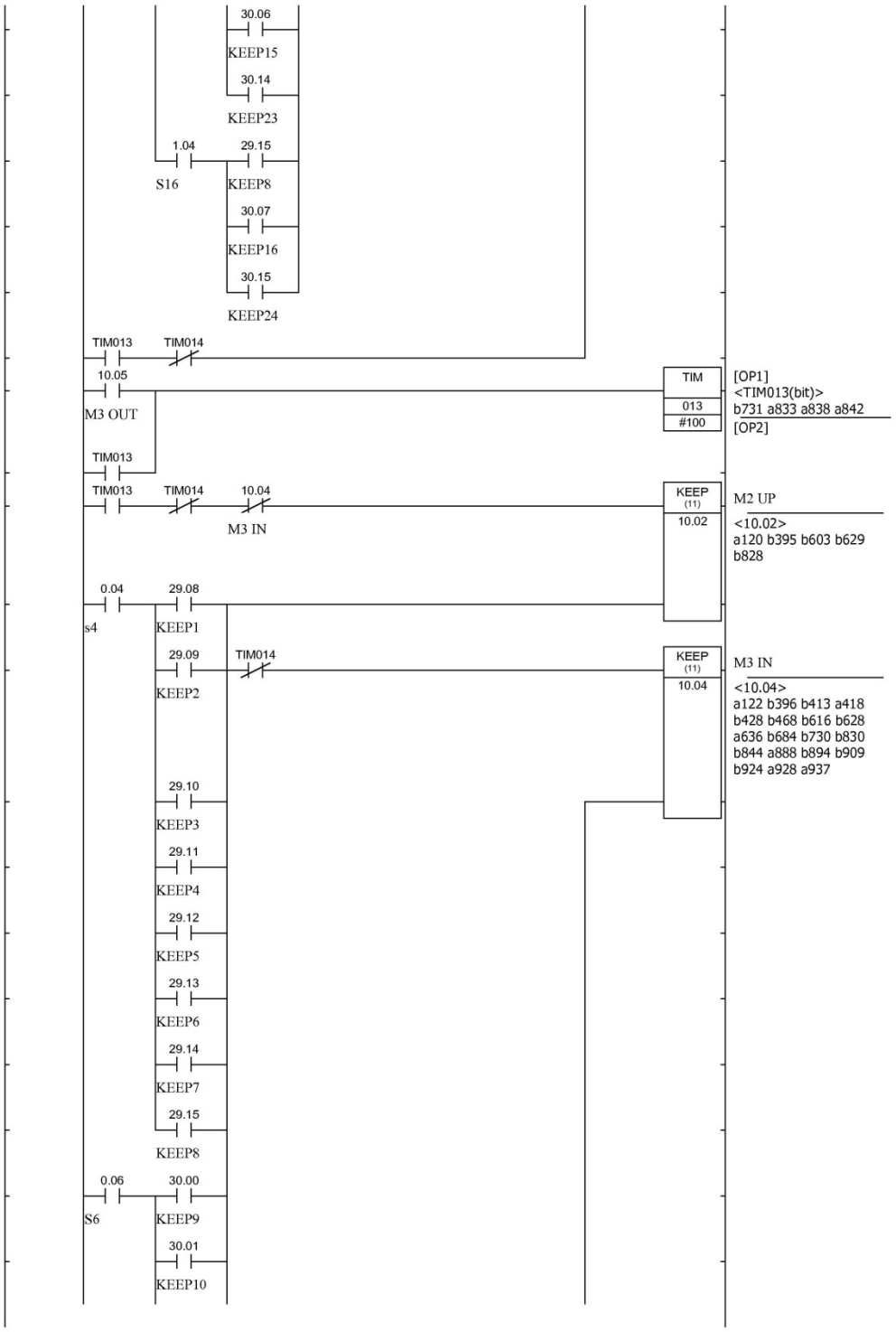

KEEPS1
000059
(000645)
TIM012
29.08
KEEP1
30.00
KEEP9
30.08
KEEP17
KEEP
(11)
10.01
M1 CCW
<10.01>
a119 b687 b911
0.09
S9
000060
(000652)
20.01
CARD2
22.05
STEP1
KEEP
(11)
22.08
STEP3
<22.08>
a679 a896 a900
20.00
CARD1
20.02
CARD3
20.03
CARD4
20.04
CARD5
20.05
CARD6
20.06
CARD7
20.07
CARD8
20.08
CARD9
20.09
CARD10
20.10
CARD11
20.11
CARD12
20.12
CARD13
20.13
CARD14
20.14
CARD15

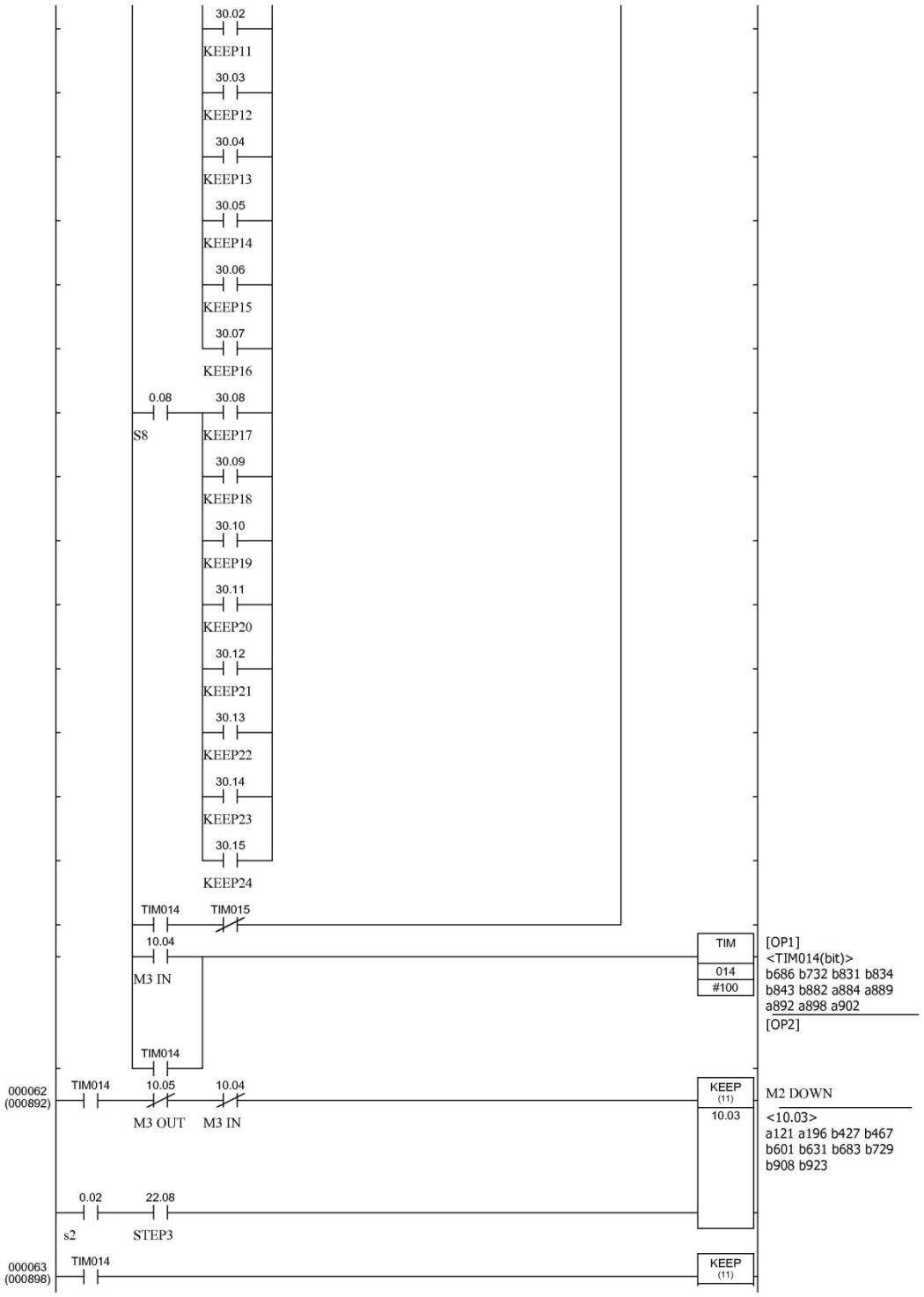

20.15
CARD16
21.00
CARD17
21.01
CARD18
21.02
CARD19
21.03
CARD20
21.04
CARD21
21.05
CARD22
21.06
CARD23
21.07
CARD24
TIM016
000061
(000679)
22.08
STEP3
10.05
M3 OUT
10.00
M1 CW
10.03
M2
DOWN
10.04
M3 IN
19.01
KEEPS3.
S5.S7
TIM014
10.01
M1 CCW
KEEP
(11)
10.02
M2 UP
<10.02>
a120 b395 b603 b629
b828
0.03
s3
29.08
KEEP1
29.09
KEEP2
29.10
KEEP3
29.11
KEEP4
29.12
KEEP5
29.13
KEEP6
29.14
KEEP7
29.15
KEEP8
19.01
KEEPS3.S5.S7
<19.01>
b685 a721 a781
0.05
30.00

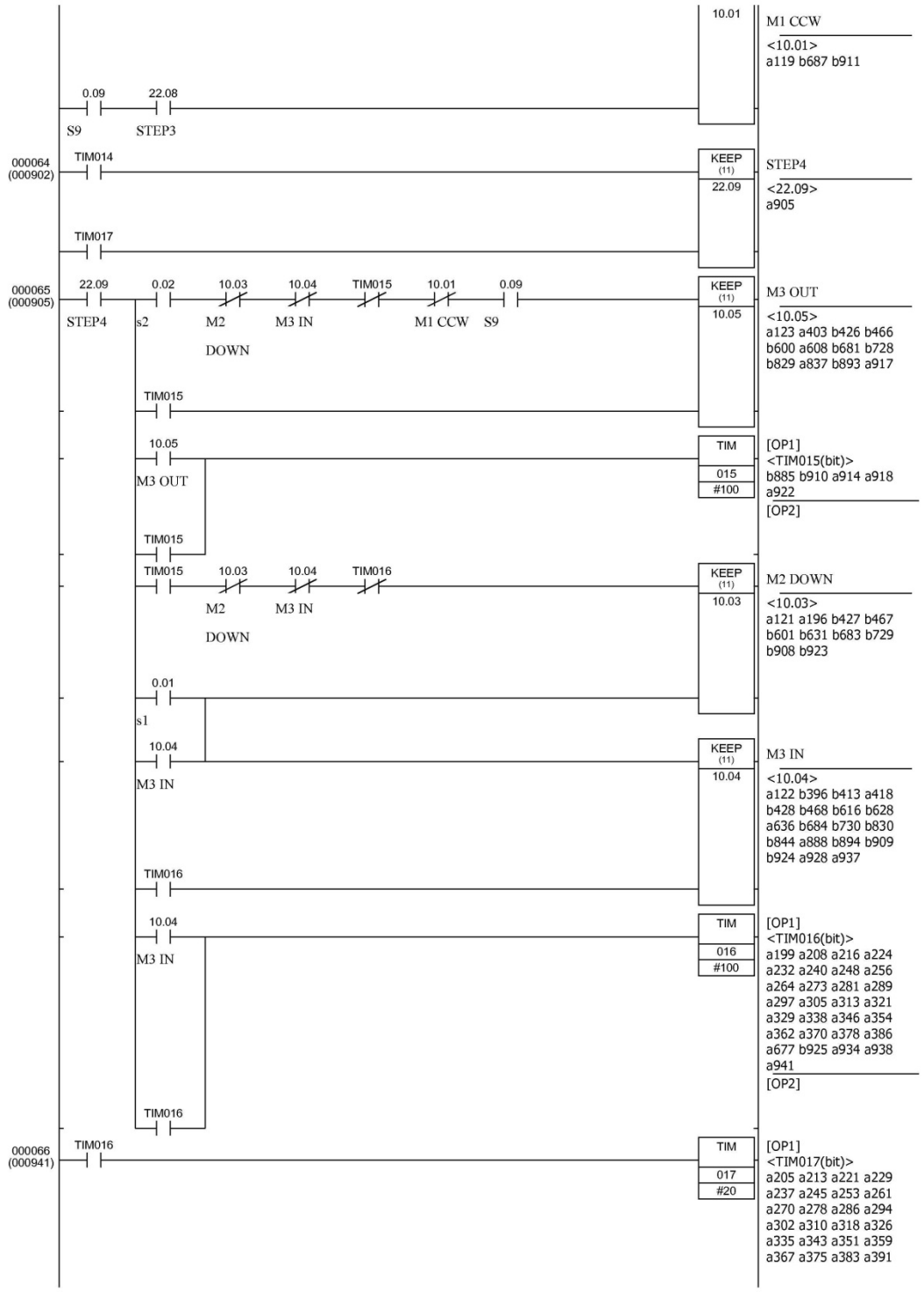

S5
KEEP9
30.01
KEEP10
30.02
KEEP11
30.03
KEEP12
30.04
KEEP13
30.05
KEEP14
30.06
KEEP15
30.07
KEEP16
0.07
30.08
S7
KEEP17
30.09
KEEP18
30.10
KEEP19
30.11
KEEP20
30.12
KEEP21
30.13
KEEP22
30.14
KEEP23
30.15
KEEP24
19.01
KEEPS3.
S5.S7
10.05
10.03
10.04
TIM013
TIM014
29.08
30.00
30.08
KEEP
(11)
10.00
M1 CW
<10.00>
a118 b398 b430 b602
b682
M3 OUT
M2
DOWN
M3 IN
KEEP1
KEEP9
KEEP17
0.10
29.09
S10
KEEP2
30.01


KEEP10
30.09
KEEP18
0.11
29.10
S11
KEEP3
30.02
KEEP11
30.10
KEEP19
1.00
29.11
S12
KEEP4
30.03
KEEP12
30.11
KEEP20
1.01
29.12
S13
KEEP5
30.04
KEEP13
30.12
KEEP21
1.02
29.13
S14
KEEP6
30.05
KEEP14
30.13
KEEP22
1.03
29.14
S15
KEEP7
30.06
KEEP15
30.14
KEEP23
1.04
29.15
S16
KEEP8
30.07
KEEP16
30.15
KEEP24
19.01
29.08
10.02
10.05
10.04
TIM014
KEEP
(11)


KEEPS3.
S5.S7
KEEP1
M2 UP
M3 OUT
M3 IN
30.00
KEEP9
30.08
KEEP17
0.10
29.09
S10
KEEP2
30.01
KEEP10
30.09
KEEP18
0.11
29.10
S11
KEEP3
30.02
KEEP11
30.10
KEEP19
1.00
29.11
S12
KEEP4
30.03
KEEP12
30.11
KEEP20
1.01
29.12
S13
KEEP5
30.04
KEEP13
30.12
KEEP21
1.02
29.13
S14
KEEP6
30.05
KEEP14
30.13
KEEP22
1.03
29.14
S15
KEEP7
10.05
M3 OUT
<10.05>
a123 a403 b426 b466
b600 a608 b681 b728
b829 a837 b893 a917


30.06
KEEP15
30.14
KEEP23
1.04
S16
29.15
KEEP8
30.07
KEEP16
30.15
KEEP24
TIM013
TIM014
10.05
M3 OUT
TIM
013
#100
[OP1]
<TIM013(bit)>
b731 a833 a838 a842
[OP2]
TIM013
TIM013
TIM014
10.04
M3 IN
KEEP
(11)
10.02
M2 UP
<10.02>
a120 b395 b603 b629
b828
0.04
s4
29.08
KEEP1
29.09
KEEP2
TIM014
KEEP
(11)
10.04
M3 IN
<10.04>
a122 b396 b413 a418
b428 b468 b616 b628
a636 b684 b730 b830
b844 a888 b894 b909
b924 a928 a937
29.10
KEEP3
29.11
KEEP4
29.12
KEEP5
29.13
KEEP6
29.14
KEEP7
29.15
KEEP8
0.06
S6
30.00
KEEP9
30.01
KEEP10


30.02
KEEP11
30.03
KEEP12
30.04
KEEP13
30.05
KEEP14
30.06
KEEP15
30.07
KEEP16
0.08
S8
30.08
KEEP17
30.09
KEEP18
30.10
KEEP19
30.11
KEEP20
30.12
KEEP21
30.13
KEEP22
30.14
KEEP23
30.15
KEEP24
TIM014
TIM015
10.04
M3 IN
TIM
014
#100
[OP1]
<TIM014(bit)>
b686 b732 b831 b834
b843 b882 a884 a889
a892 a898 a902
[OP2]
TIM014
000062
(000892)
TIM014
10.05
M3 OUT
10.04
M3 IN
KEEP
(11)
10.03
M2 DOWN
<10.03>
a121 a196 b427 b467
b601 b631 b683 b729
b908 b923
0.02
s2
22.08
STEP3
000063
(000898)
TIM014
KEEP
(11)


10.01
M1 CCW
<10.01>
a119 b687 b911
0.09
S9
22.08
STEP3
000064
(000902)
TIM014
KEEP
(11)
22.09
STEP4
<22.09>
a905
TIM017
000065
(000905)
22.09
STEP4
0.02
s2
10.03
M2
DOWN
10.04
M3 IN
TIM015
10.01
M1 CCW
0.09
S9
KEEP
(11)
10.05
M3 OUT
<10.05>
a123 a403 b426 b466
b600 a608 b681 b728
b829 a837 b893 a917
TIM015
10.05
M3 OUT
TIM
015
#100
[OP1]
<TIM015(bit)>
b885 b910 a914 a918
a922
[OP2]
TIM015
TIM015
10.03
M2
DOWN
10.04
M3 IN
TIM016
KEEP
(11)
10.03
M2 DOWN
<10.03>
a121 a196 b427 b467
b601 b631 b683 b729
b908 b923
0.01
s1
10.04
M3 IN
KEEP
(11)
10.04
M3 IN
<10.04>
a122 b396 b413 a418
b428 b468 b616 b628
a636 b684 b730 b830
b844 a888 b894 b909
b924 a928 a937
TIM016
10.04
M3 IN
TIM
016
#100
[OP1]
<TIM016(bit)>
a199 a208 a216 a224
a232 a240 a248 a256
a264 a273 a281 a289
a297 a305 a313 a321
a329 a338 a346 a354
a362 a370 a378 a386
a677 b925 a934 a938
a941
[OP2]
TIM016
000066
(000941)
TIM016
TIM
017
#20
[OP1]
<TIM017(bit)>
a205 a213 a221 a229
a237 a245 a253 a261
a270 a278 a286 a294
a302 a310 a318 a326
a335 a343 a351 a359
a367 a375 a383 a391