

Embedded system experiment report #2: Motor Control

Hong Chenhui^a

^a*DrRedTheRed*

Professor/You Wang

Abstract—Unfathomably rudimentary motor control with LCD1602 display. Powered by STM32F103C8.

keywords—*L^AT_EX* No.

Contents

1	Task description	1
2	Code structure	1
3	Implementation	1
3.1	Hardware	1
	Items • Wiring	
3.2	Manual	1
3.3	Code	1

1. Task description

Develop a stepper motor control system via a temperature sensor and a LCD display. Set a reference temperature on which the motor's speed and direction based on, displaying current values.

2. Code structure

Aside from the built-in STM32 library functions, the code is organized as follows.

1. **main.c** contains the main program, where
 - The keys undergo a scanning process to procure the input slated for processing.
 - The temperature sensor data is engaged to initiate the activation of the motor.
 - The digital displays are configured.
2. **led.c** contains the led display function.
3. **StepMotor.c** contains the stepper motor driver.
4. **DS18B20.c** contains the temperature sensor reader.
5. **1602.c** contains the LCD1602 display driver.
6. **key.c** contains the key input reader.

3. Implementation

3.1. Hardware

3.1.1. Items

1. STM32F103C8 chip
2. Keypad
3. LCD1602
4. DS18B20 temperature sensor
5. Stepper motor

3.1.2. Wiring

- **STM32F103C8** P1.3-P1.6 → **Stepper Motor** IA-ID
- **STM32F103C8** → **LCD1602**
- **STM32F103C8** P3.7 → **DS18B20** J1
- **STM32F103C8** P2.0-P2.1 → **Keypad** K2-K1

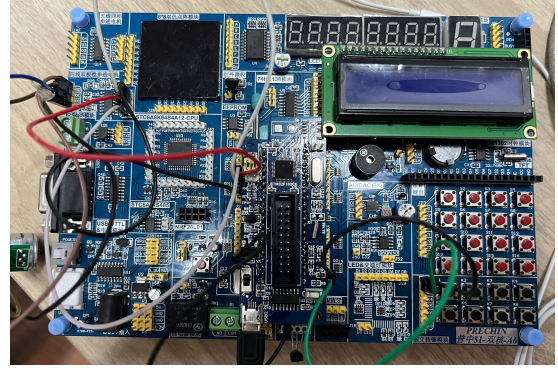


Figure 1. Board layout.

3.2. Manual

- **K1**: Forward/Backward adjustment selection.
- **K2**: Each press increases/decreases the set temperature number incrementally.

3.3. Code

Only few parts of the code are omitted for brevity.

```
1  /*****
2  main function
3  *****/
4
5  #include "system.h"
6  #include "SysTick.h"
7  #include "led.h"
8  #include "usart.h"
9  #include "ds18b20.h"
10 #include "StepMotor.h"
11 #include "1602.h"
12 #include "key.h"
13
14
15
16 int main()
17 {
18     u8 i=0;
19     float temper;
20     int Temper;
21     int Set_Temper;
22     int delta;
23     unsigned int KEY;
24     unsigned int flag;
25     flag=1;
26     Set_Temper=30;
27     Motor_Init( );
28     LCD_INIT( );
29     LCD_Init( );
30     KEY_Init( );
31     SysTick_Init(72);
32     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
33     LED_Init( );
34     USART1_Init(9600);
35     while(DS18B20_Init())
36     {
37         LCD_ShowString(1,1,"Status Abnormal");
38         delay_ms(500);
39     }
40
41
42     while(1)
43     {
44         i++;
45
46         KEY=KEY_Scan(0);
47         if(KEY==1)
48         {
49             Set_Temper=Set_Temper+flag;
50         }
```

```

51 }
52 else if(KEY==2)
53 {
54     flag=-flag;
55 }
56 if(i%50==0)
57 {
58     temper=DS18B20_GetTemperture();
59     Temper=(int)temper;
60
61     if(Set_Temper>=0){
62         LCD_ShowString(1,1,"PV: ");
63         LCD_ShowNum(1,5,Temper,2);
64         LCD_ShowString(1,9,"SP: ");
65         LCD_ShowNum(1,13,Set_Temper,2);
66     }else {
67         LCD_ShowString(1,1,"PV: ");
68         LCD_ShowNum(1,5,Temper,2);
69         LCD_ShowString(1,9,"SP:-");
70         LCD_ShowNum(1,13,-Set_Temper,2);
71     }
72
73     if(Temper-Set_Temper>0)
74     {
75         int delay;
76         float speed;
77         int speed_int;
78         int speed_dec;
79         if (Temper-Set_Temper<=20)
80         {
81             delay=(20-(Temper-Set_Temper))+1;
82         }
83         else if (Temper-Set_Temper>20)
84         {
85             delay=(20-(20))+1;
86         }
87
88         speed=20000/(4*delay);
89         speed_int=speed/100;
90         speed_dec=speed-speed_int*100;
91
92         FWDMotor_two(delay);
93         LCD_ShowString(2,1,"SPEED:-");
94         LCD_ShowNum(2,8,speed_int,2);
95         LCD_ShowString(2,10,".");
96         LCD_ShowNum(2,11,speed_dec,2);
97         LCD_ShowString(2,13,"r/s");
98         //show stepMotor's speed on LCD1602
99     }
100     else if (Set_Temper-Temper>0)
101     {
102         int delay;
103         float speed;
104         int speed_int;
105         int speed_dec;
106         if (-Temper+Set_Temper<=20)
107         {
108             delay=(20-(-Temper+Set_Temper))+1;
109         }
110         else if (-Temper+Set_Temper>20)
111         {
112             delay=(20-(20))+1;
113         }
114
115         speed=20000/(4*delay);
116         speed_int=speed/100;
117         speed_dec=speed-speed_int*100;
118
119         REVMotor_two(delay);
120         LCD_ShowString(2,1,"SPEED: ");
121         LCD_ShowNum(2,8,speed_int,2);
122         LCD_ShowString(2,10,".");
123         LCD_ShowNum(2,11,speed_dec,2);
124         LCD_ShowString(2,13,"r/s");
125         //show stepMotor's speed on LCD1602
126     }
127     else if (Set_Temper-Temper==0)
128     {
129         LCD_ShowString(2,1,"SPEED: 00.00r/s");
130         //show stepMotor's speed on LCD1602
131     }
132 }

```

```

142 }
143 }
144 }
145 }

```

Code 1. main.c

```

1 #include "StepMotor.h"
2 #include "SysTick.h"
3
4 GPIO_InitTypeDef GPIO_InitStructure;
5
6 void Motor_Init(void)
7 {
8     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
9
10    //GPIO_InitTypeDef GPIO_InitStructure;
11    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
12    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3 | GPIO_Pin_4
13    | GPIO_Pin_5 | GPIO_Pin_6;
14    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
15    GPIO_Init(GPIOA, &GPIO_InitStructure);
16 }
17
18 //A+ -> B+ ->A- ->B-
19
20 void FWDMotor_two(unsigned int speed)
21 {
22
23     GPIO_SetBits(GPIOA,GPIO_Pin_3);
24     GPIO_ResetBits(GPIOA,GPIO_Pin_5 | GPIO_Pin_6|
25         GPIO_Pin_4);
26     delay_ms(speed);
27
28     GPIO_SetBits(GPIOA,GPIO_Pin_5);
29     GPIO_ResetBits(GPIOA,GPIO_Pin_3 | GPIO_Pin_6|
30         GPIO_Pin_4);
31     delay_ms(speed);
32
33     GPIO_SetBits(GPIOA,GPIO_Pin_4);
34     GPIO_ResetBits(GPIOA,GPIO_Pin_3 | GPIO_Pin_5|
35         GPIO_Pin_6);
36     delay_ms(speed);
37
38     GPIO_SetBits(GPIOA,GPIO_Pin_6);
39     GPIO_ResetBits(GPIOA,GPIO_Pin_4 | GPIO_Pin_5|
40         GPIO_Pin_3);
41     delay_ms(speed);
42 }
43
44 //A+ -> B- ->A- ->B+
45
46 void REVMotor_two(unsigned int speed)
47 {
48     GPIO_SetBits(GPIOA,GPIO_Pin_3);
49     GPIO_ResetBits(GPIOA,GPIO_Pin_5 | GPIO_Pin_6|
50         GPIO_Pin_4);
51     delay_ms(speed);
52
53     GPIO_SetBits(GPIOA,GPIO_Pin_6);
54     GPIO_ResetBits(GPIOA,GPIO_Pin_4 | GPIO_Pin_5|
55         GPIO_Pin_3);
56     delay_ms(speed);
57
58     GPIO_SetBits(GPIOA,GPIO_Pin_4);
59     GPIO_ResetBits(GPIOA,GPIO_Pin_6 | GPIO_Pin_3|
60         GPIO_Pin_5);
61     delay_ms(speed);
62
63     GPIO_SetBits(GPIOA,GPIO_Pin_5);
64     GPIO_ResetBits(GPIOA,GPIO_Pin_6 | GPIO_Pin_3|
65         GPIO_Pin_4);
66     delay_ms(speed);
67 }

```

Code 2. StepMotor.c

```

1 #include "key.h"
2 #include "SysTick.h"
3
4 void KEY_Init(void)

```

```

5 {
6     GPIO_InitTypeDef GPIO_InitStructure;
7     RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
8
9     GPIO_InitStructure.GPIO_Pin=KEY1_Pin|KEY2_Pin;
10    GPIO_InitStructure.GPIO_Mode=GPIO_Mode_IPU;
11    GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;
12    GPIO_Init(KEY_Port, &GPIO_InitStructure);
13 }
14
15 u8 KEY_Scan(u8 mode)
16 {
17     static u8 key=1;
18     if(key==1&&(KEY1==0||KEY2==0))
19     {
20         delay_ms(1);
21         key=0;
22         if(KEY1==0)
23         {
24             return KEY1_VALUE;
25         }
26         else if(KEY2==0)
27         {
28             return KEY2_VALUE;
29         }
30     }
31     else if(KEY1==1&&KEY2==1)
32     {
33         key=1;
34     }
35     if(mode==1)
36     {
37         key=1;
38     }
39     return 0;
40 }

```

Code 3. key.c

```

1 #include "stm32f10x.h"
2 #include "1602.h"
3
4 void LCD_INIT(){
5     GPIO_InitTypeDef GPIO_InitStructure;
6
7     RCC_APB2PeriphClockCmd(LCD_PORT_RCC, ENABLE);
8
9     GPIO_InitStructure.GPIO_Mode=GPIO_Mode_Out_PP;
10    GPIO_InitStructure.GPIO_Pin=LCD_PIN;
11    GPIO_InitStructure.GPIO_Speed=GPIO_Speed_50MHz;
12    GPIO_Init(LCD_PORT, &GPIO_InitStructure);
13 }
14
15
16
17 #define LCD_DataPort GPIOB
18 void WriteData(u8 data){
19     GPIO_WriteBit(LCD_PORT, GPIO_Pin_15, (BitAction)((data
20     & 0x80) >> 7)); //D7
21     GPIO_WriteBit(LCD_PORT, GPIO_Pin_14, (BitAction)((data
22     & 0x40) >> 6));
23     GPIO_WriteBit(LCD_PORT, GPIO_Pin_13, (BitAction)((data
24     & 0x20) >> 5));
25     GPIO_WriteBit(LCD_PORT, GPIO_Pin_12, (BitAction)((data
26     & 0x10) >> 4));
27     GPIO_WriteBit(LCD_PORT, GPIO_Pin_11, (BitAction)((data
28     & 0x08) >> 3));
29     GPIO_WriteBit(LCD_PORT, GPIO_Pin_10, (BitAction)((data
30     & 0x04) >> 2));
31     GPIO_WriteBit(LCD_PORT, GPIO_Pin_9, (BitAction)((data
32     & 0x02) >> 1));
33     GPIO_WriteBit(LCD_PORT, GPIO_Pin_8, (BitAction)(data &
34     0x01)); //D0
35 }
36
37 void LCD_Delay()
38 {
39     unsigned char i, j;
40
41     i = 2;
42     j = 239;
43     do
44     {
45         while (--j);
46     } while (--i);
47 }

```

```

42 }
43
44 void LCD_WriteCommand(unsigned char Command)
45 {
46     LCD_RS=0;
47     LCD_RW=0;
48     WriteData(Command);
49     LCD_EN=1;
50     LCD_Delay();
51     LCD_EN=0;
52     LCD_Delay();
53 }
54
55 void LCD_WriteData(unsigned char Data)
56 {
57     LCD_RS=1;
58     LCD_RW=0;
59     WriteData(Data);
60     LCD_EN=1;
61     LCD_Delay();
62     LCD_EN=0;
63     LCD_Delay();
64 }
65
66 void LCD_SetCursor(unsigned char Line, unsigned char
67     Column)
68 {
69     if(Line==1)
70     {
71         LCD_WriteCommand(0x80|(Column-1));
72     }
73     else if(Line==2)
74     {
75         LCD_WriteCommand(0x80|(Column-1+0x40));
76     }
77 }
78
79 void LCD_Init()
80 {
81     LCD_WriteCommand(0x38); //?????,????,5*7??
82     LCD_WriteCommand(0x0c); //???,???,???
83     LCD_WriteCommand(0x06); //???????,??????,????
84     LCD_WriteCommand(0x01); //????,??
85 }
86
87 void LCD_ShowChar(unsigned char Line, unsigned char
88     Column, char Char)
89 {
90     LCD_SetCursor(Line, Column);
91     LCD_WriteData(Char);
92 }
93
94 void LCD_ShowString(unsigned char Line, unsigned char
95     Column, char *String)
96 {
97     unsigned char i;
98     LCD_SetCursor(Line, Column);
99     for(i=0; String[i]!='\0'; i++)
100     {
101         LCD_WriteData(String[i]);
102     }
103 }
104
105 int LCD_Pow(int X, int Y)
106 {
107     unsigned char i;
108     int Result=1;
109     for(i=0; i<Y; i++)
110     {
111         Result*=X;
112     }
113     return Result;
114 }
115
116 void LCD_ShowNum(unsigned char Line, unsigned char
117     Column, unsigned int Number, unsigned char Length)
118 {
119     unsigned char i;
120     LCD_SetCursor(Line, Column);
121     for(i=Length; i>0; i--)
122     {
123         LCD_WriteData(Number/LCD_Pow(10, i-1)%10+'0');
124     }
125 }

```

Code 4. 1602.c