Digital Lab 4:

Experiment 3:

Stepper Motor Control

Date: 2024/04/02

Class: 電機三全英班

Group: Group 11

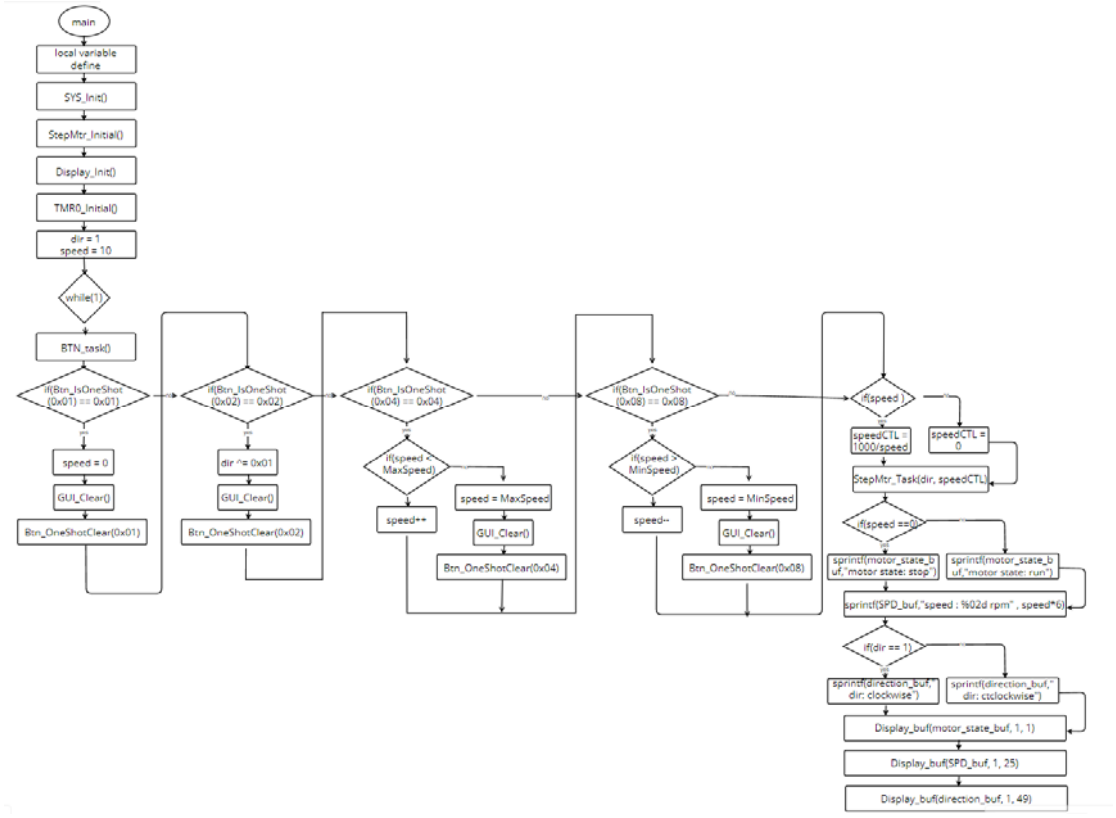
Name: B103105006 胡庭翊

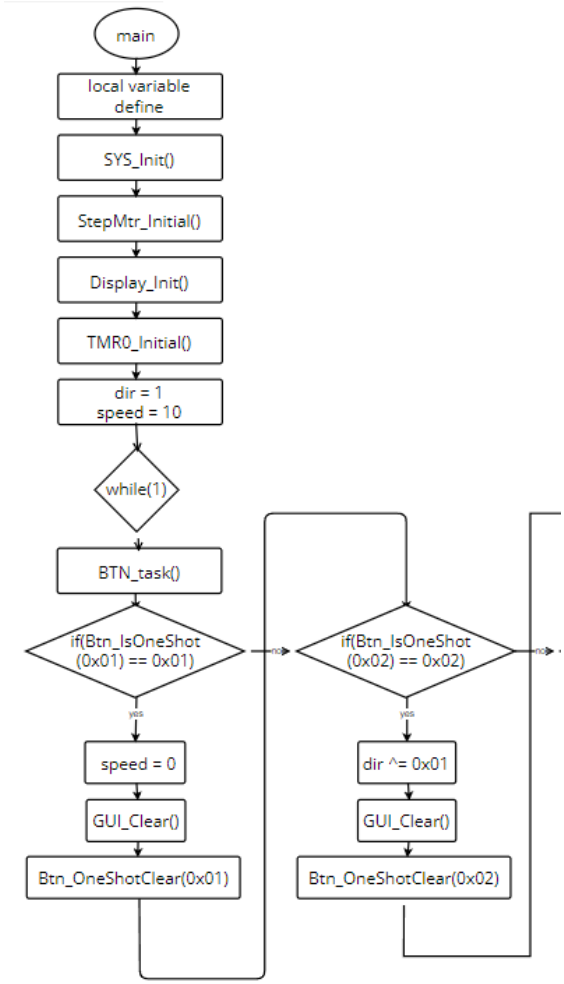
I. Annotated Code

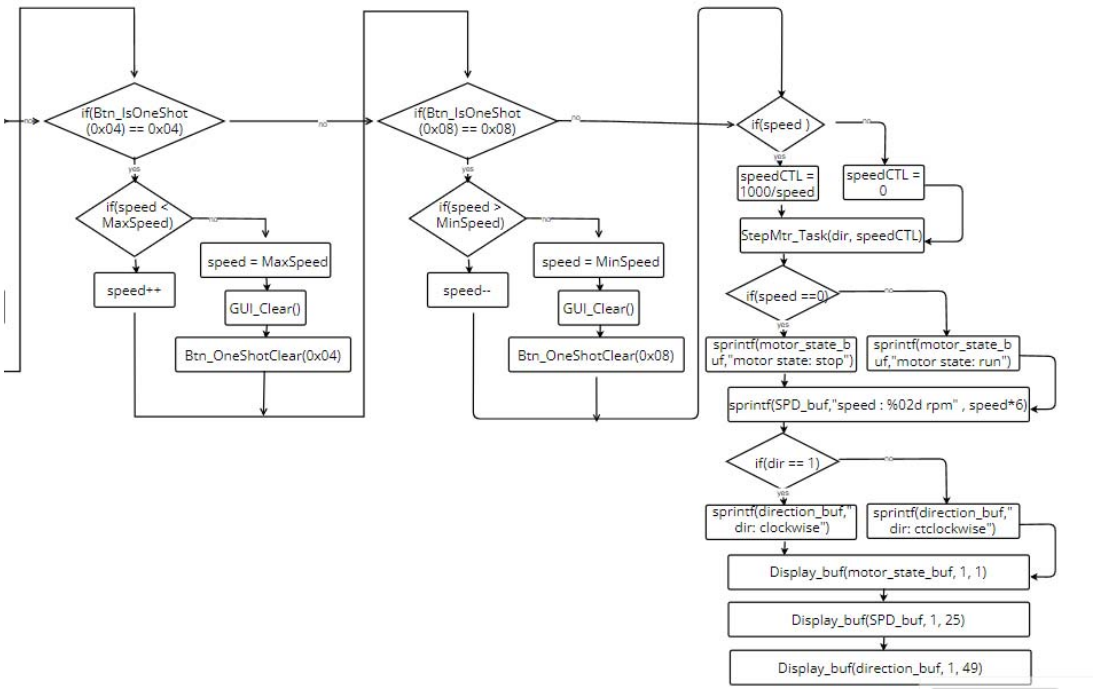
```c
#include "stdio.h"
#include "NuMicro.h"
#include "tmr.h"
#include "system_init.h"
#include "GUI.h"
#include "display.h"
#include "BNCTL.h"
#include "StepMotorAgent.h"

/* define max and mini speed */
#define MaxSpeed    17
#define MinSpeed    1

/* global variable define */
uint32_t    timecount;
uint8_t     dir;
uint32_t    speed;

int main(void)
{
        /* local variable define */
        char motor_state_buf[30];
        char SPD_buf[30];
        char direction_buf[30];
        uint32_t  speedCTL;

        /* Init System */
        SYS_Init();

        /*button initialize*/
        BTN_init();

        /*Step Motor initialize*/
        StepMtr_Initial();

        /* GUI display initialize */
        Display_Init();

        /* Init TMR0 for timecount */
        TMR0_Initial();

        /* Set initial value for speed direction */
        dir = 1;
        speed = 10;//10rounds in a sec

    while(1) //always conduct
        {
            /* Scan button */
            BTN_task();
            if(Btn_IsOneShot(0x01) == 0x01){
            //stop
                speed = 0;
                GUI_Clear();//clear the GUI previous output
                Btn_OneShotClear(0x01);//clear the flag
            }
            if(Btn_IsOneShot(0x02) == 0x02){
                dir ^= 0x01;//change the direction
                GUI_Clear();
                Btn_OneShotClear(0x02);
            }
            if(Btn_IsOneShot(0x04) == 0x04){
            //speed up
                if(speed < MaxSpeed)
                    speed++;
                else
                    speed = MaxSpeed;
                GUI_Clear();
                Btn_OneShotClear(0x04);
            }
```

```
70                  if(Btn_IsOneShot(0x08) == 0x08){
71                  //speed down
72                      if(speed > MinSpeed)
73                          speed--;
74                      else
75                          speed = MinSpeed;
76                          GUI_Clear();
77                          Btn_OneShotClear(0x08);
78                  }
79
80                  /* Step motor output */
81                  if(speed)
82                      speedCTL = 1000/speed; //true speed convertion
83                  else
84                      speedCTL = 0; //stop
85
86                  StepMtr_Task(dir, speedCTL);
87
88                  //write motor state buffer
89                  if (speed ==0)
90                      sprintf(motor_state_buf,"motor state: stop");
91                  else
92                      sprintf(motor_state_buf,"motor state: run");
93                  sprintf(SPD_buf,"speed : %02d rpm" , speed*6);//6~102
94
95                  //write direction buffer
96                  if (dir == 1)
97                      sprintf(direction_buf,"dir: clockwise");
98                  else
99                      sprintf(direction_buf,"dir: ctclockwise");
100
101
102                 Display_buf(motor_state_buf, 1, 1);//motor state
103                 Display_buf(SPD_buf, 1, 25);
104                 Display_buf(direction_buf, 1, 49);//direction
105         }
106 }
107
```

II. Program Flow




main
local variable define
SYS_Init()
StepMtr_Initial()
Display_Init()
TMR0_Initial()
dir = 1
speed = 10
while(1)
BTN_task()
if(Btn_IsOneShot (0x01) == 0x01)
yes
speed = 0
GUI_Clear()
Btn_OneShotClear(0x01)
no
if(Btn_IsOneShot (0x02) == 0x02)
yes
dir ^= 0x01
GUI_Clear()
Btn_OneShotClear(0x02)
no


if(Btn_IsOneShot (0x04) == 0x04)
yes
if(speed < MaxSpeed)
speed++
no
speed = MaxSpeed
GUI_Clear()
Btn_OneShotClear(0x04)
no
if(Btn_IsOneShot (0x08) == 0x08)
yes
if(speed > MinSpeed)
speed--
no
speed = MinSpeed
GUI_Clear()
Btn_OneShotClear(0x08)
no
if(speed)
yes
speedCTL = 1000/speed
no
speedCTL = 0
StepMtr_Task(dir, speedCTL)
if(speed ==0)
yes
sprintf(motor_state_buf,"motor state: stop")
no
sprintf(motor_state_buf,"motor state: run")
sprintf(SPD_buf,"speed : %02d rpm" , speed*6)
if(dir == 1)
yes
sprintf(direction_buf," dir: clockwise")
no
sprintf(direction_buf," dir: ctclockwise")
Display_buf(motor_state_buf, 1, 1)
Display_buf(SPD_buf, 1, 25)
Display_buf(direction_buf, 1, 49)

III. Thoughts

This electrical engineering experiment provided me with the opportunity to further understand the control principles of stepper motors. We used C language to control the stepper motor, building upon our familiarity with stepper motor principles from the previous semester where we worked with Verilog. This time, we utilized C language along with the one-shot function, coupled with ULN2003A and M487 boards, to control the speed and direction of the motor and display its current status on the board's screen.

During the experiment, we encountered some challenges, particularly in understanding the interaction between C language and hardware. However, through careful reading of documentation and hands-on experience, we gradually learned how to use the one-shot function to control the stepper motor and successfully achieved speed and direction control. The most exciting part was successfully displaying the current status of the stepper motor on the board's screen, which left us feeling fulfilled and proud.

Through this experiment, we not only deepened our understanding of the control principles of stepper motors but also improved our ability to interact with hardware in the C language environment. Additionally, the challenges we faced during the experiment helped us become more familiar with troubleshooting and problem-solving techniques. Overall, this experiment was a valuable learning experience that laid a solid foundation for our future research and applications in related fields.