2TA4 Lab 5 report

Abdulrahman Derbala

derbalaa

400301521

1. The angular resolution of the given motor which makes 48 steps per revolution is

$$\frac{360°}{48\ steps} = 7.5°/step$$

2. My student number is 400301521, this means the period of one revolution will be 21+33=54
3. The time period between two steps of the stepper motor is defined by

$$\frac{\text{the period of one revolution}}{\text{the number of steps}}$$

   3.1. Full step: $54/48 = 1.125\ seconds/step$
   3.2. Half step: $54/96 = 0.5625\ seconds/step$
4. Prescaler for both full and half step $= (45MHz/10kHz) - 1 = 4{,}499$
   4.1. Full step: OCR $= 54/48 \times 10kHz - 1 = 11{,}249\ count/step$
   4.2. Half step: OCR $= 54/96 \times 48kHz - 1 = 5{,}624\ count/step$
5. **Period is initialized to 11250.**

```
272  void  TIM3_Config(void)
273 {
274
275
276    Tim3_PrescalerValue = (uint32_t) ((SystemCoreClock /2) / 10000) - 1;
277
278    Tim3_Handle.Instance = TIM3; //TIM3 is defined in stm32f429xx.h
279
280    Tim3_Handle.Init.Period = period - 1;
281    Tim3_Handle.Init.Prescaler = Tim3_PrescalerValue;
282    Tim3_Handle.Init.ClockDivision = 0;
283    Tim3_Handle.Init.CounterMode = TIM_COUNTERMODE_UP;
284    HAL_TIM_Base_Init(&Tim3_Handle);
285    HAL_TIM_Base_Start_IT(&Tim3_Handle);
286
287 }
```

```
  if(GPIO_Pin == KEY_BUTTON_PIN)   //GPIO_PIN_0
  {
      if(mode==1){
        mode = 0;
        state=0;
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
      }
      else if(mode==0){
        mode = 1;
        stateh=0;
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
        HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
      }
    //HAL_GPIO_TogglePin(GPIOC,GPIO_PIN_13);
  }

if(GPIO_Pin == GPIO_PIN_1)
{
    if(direction==1){
      direction = 0;
      state=0;stateh=0;
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
    }
    else if(direction==0){
      direction = 1;
      stateh=0;state=0;
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
      HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
    }
    LCD_DisplayInt(1,0,1);
}  //end of PIN_1
```

```
if(GPIO_Pin == GPIO_PIN_2)
{
  period+=1000;
  TIM3_Config();
} //end of if PIN_2

if(GPIO_Pin == GPIO_PIN_3)
{
  period-=1000;
  TIM3_Config();
  //LCD_DisplayInt(1,0,3);

} //end of if PIN_3
```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) //see stm32fxx_hal_tim.c for different callback function names.

//for timer 3 , Timer 3 use update event initerrupt

```
{

    BSP_LED_Toggle(LED4);
    if(mode==0&&direction==0){
            if(state==0){
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,0);
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,1);
                    state = 1;
            }
            else if(state==1){
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,1);
                    state=2;
            }
            else if(state==2){
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,1);
                    state =3;
            }
            else if(state==3){
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
                    state=0;
            }
    }
    else if(mode==1&&direction==0){
            if(stateh==0){
                    HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,1);
```

```
                stateh = 1;
        }
        else if(stateh==1){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,0);
                stateh=2;
        }
        else if(stateh==2){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,1);
                stateh =3;
        }
        else if(stateh==3){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
                stateh=4;
        }
        else if(stateh==4){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,1);
                stateh=5;
        }
        else if(stateh==5){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
                stateh=6;
        }
        else if(stateh==6){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
                stateh=7;
        }
        else if(stateh==7){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
                stateh=0;
        }
}
else if(mode==0&&direction==1){
        if(state==0){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,0);
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,1);
                state = 3;
        }
        else if(state==1){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,1);
                state=0;
        }
        else if(state==2){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
```

```
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,1);
                state =1;
        }
        else if(state==3){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
                state=2;
        }
    }
    else if(mode==1&&direction==1){
        if(stateh==0){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,1);
                stateh = 7;
        }
        else if(stateh==1){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,0);
                stateh=0;
        }
        else if(stateh==2){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,1);
                stateh =1;
        }
        else if(stateh==3){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_14,0);
                stateh=2;
        }
        else if(stateh==4){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,1);
                stateh=3;
        }
        else if(stateh==5){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_4,0);
                stateh=4;
        }
        else if(stateh==6){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,1);
                stateh=5;
        }
        else if(stateh==7){
                HAL_GPIO_WritePin(GPIOC,GPIO_PIN_15,0);
                stateh=6;
        }
    }
}
```