

Mechtron 3TA4 Lab 5 Report

Ho (Brian) Chiu

McMaster University

MacID: chiuh1

Student #: 400054774

1. Angular Resolution of a 48 step/revolution stepper motor

$$\frac{360^\circ}{48 \text{ steps}} = 7.5^\circ/\text{step}$$

2. 400054774, $74 > 66$ so my period will be $74 - 33 = 41$

3. a) Half step:

Motor will step $48 \times 2 = 96$ steps,

$$\frac{41s}{96 \text{ steps}} = 0.4217s/\text{step}$$

- b) Full step:

$$\frac{41s}{48 \text{ steps}} = 0.8542s/\text{step}$$

4. For a 4MHz clock, prescaler = 399 for both modes. This makes the clock cycle at 10kHz

- a) Half step: ARR = 4270

- b) Full step: ARR = 8541

5. C code for timer and clock

TIM3 config function generated by cubeMX: Period set to 8541 to start in Full step

```
static void MX_TIM3_Init(void)
{
    /* USER CODE BEGIN TIM3_Init 0 */
    /* USER CODE END TIM3_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM3_Init 1 */
    /* USER CODE END TIM3_Init 1 */
    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 399;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 8541;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim3.Init.AutoReloadPreload =
TIM_AUTORELOAD_PRELOAD_ENABLE;
    if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource =
TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim3,
&sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode =
TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim3,
&sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM3_Init 2 */
    if (HAL_TIM_Base_Start_IT(&htim3) != HAL_OK) {
        //Starting Error
        Error_Handler();
    }
    /* USER CODE END TIM3_Init 2 */
}
```

TIM3 Interrupt function: Increment the phase_counter in the correct direction

Full step has 4 phases, half step has 8

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    if ((*htim).Instance==TIM3)    {
        //tim3 clock

        switch(STATE) {
            case CW:
                phase_count = (phase_count+1)%4;
                                   //count forwards
                cw_or_ccw_state();
                break;

            case CW_HALF:
                phase_count = (phase_count+1)%8;
                cw_half_or_ccw_half_state();
                break;

            case CCW:
                phase_count--; //count backwards
                if(phase_count>3) phase_count = 3;
                cw_or_ccw_state();
                break;

            case CCW_HALF:
                phase_count--; //count backwards
                if(phase_count>7) phase_count = 7;
                cw_half_or_ccw_half_state();
                break;

        }

        sprintf(lcd_buffer,"%u",phase_count+1);
        BSP_LCD_GLASS_Clear();
        BSP_LCD_GLASS_DisplayString((uint8_t*)lcd_buffer);
        //display phase
    }
}
```

Button press interrupt function: Select toggles direction, left toggles full or half step, up and down change speed.

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
    //buffer button pressed
    switch(GPIO_Pin) {

        case GPIO_PIN_0:                //SELECT
            if (STATE==CW)                STATE = CCW;
            else if (STATE==CCW)          STATE = CW;
            else if (STATE==CW_HALF)      STATE = CCW_HALF;
            else if (STATE==CCW_HALF)     STATE = CW_HALF;
            break;

        case GPIO_PIN_1:                //LEFT
            if (STATE==CW)                STATE = CW_HALF;
            else if (STATE==CW_HALF)      STATE = CW;
            else if (STATE==CCW)          STATE = CCW_HALF;
            else if (STATE==CCW_HALF)     STATE = CCW;

            if (STATE == CW || STATE == CCW) htim3.Init.Period
*= 2;    //state changed to CW or CCW, double the period

            else htim3.Init.Period /= 2;
            //state changed to CW_HALF or CCW_HALF, half the period

            if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
Error_Handler();
            break;

        case GPIO_PIN_2:                //RIGHT
            break;

        case GPIO_PIN_3:                //UP
            if (htim3.Init.Period > 1000) htim3.Init.Period -=
1000;    //speed up, decrease period
            if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
Error_Handler();
            break;

        case GPIO_PIN_5:                //DOWN
            htim3.Init.Period += 1000;    //decrease speed,
increase period
            if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
Error_Handler();
            break;
    }
}
```

Full step mode output pin states:

```
void cw_or_ccw_state(void){
    //pin order for CW or CCW
    switch(phase_count){
        case 0: //1
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;

        case 1: //2
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_SET);
            break;

        case 2: //3
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_SET);
            break;

        case 3: //4
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;
    }
}
```

Half step mode output pin states:

```
void cw_half_or_ccw_half_state(void){
    //pin order for CW_HALF or CCW_HALF

    switch(phase_count){
        case 0: //1
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;

        case 1: //2
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;

        case 2: //3
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_SET);
            break;

        case 3: //4
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_SET);
            break;

        case 4: //5
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_SET);
            break;

        case 5: //6
            HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;
```

```
        case 6: //7
HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;

        case 7: //8
HAL_GPIO_WritePin(GPIOE,OUT_A_Pin,GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE,OUT_B_Pin,GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE,OUT_C_Pin,GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE,OUT_D_Pin,GPIO_PIN_RESET);
            break;
    }
}
```