3TA4 Lab #5 Report

1. Angular Resolution = $\frac{360°}{number\ of\ steps} = \frac{360°}{48} = \mathbf{7.5}°$

2. My student number is 400231297. The last two digit number is 97, larger than 33. So, subtracting 33 from it makes the clock period 97 – 33 = **64 seconds per revolution**.

3. Using the period of 64 seconds per revolution:

   a. Time interval at Half-Stepping = $\frac{Period}{number\ of\ steps} = \frac{64\ seconds}{48\ steps} = \mathbf{1.3333\ sec/step}$

   b. Time interval at Full-Stepping = $\frac{Period}{number\ of\ steps*2} = \frac{64\ seconds}{48\ steps*2} = \mathbf{0.6667\ sec/step}$

   The smallest interval is 0.0001s, so the counter frequency is chosen to be **10 kHz.**

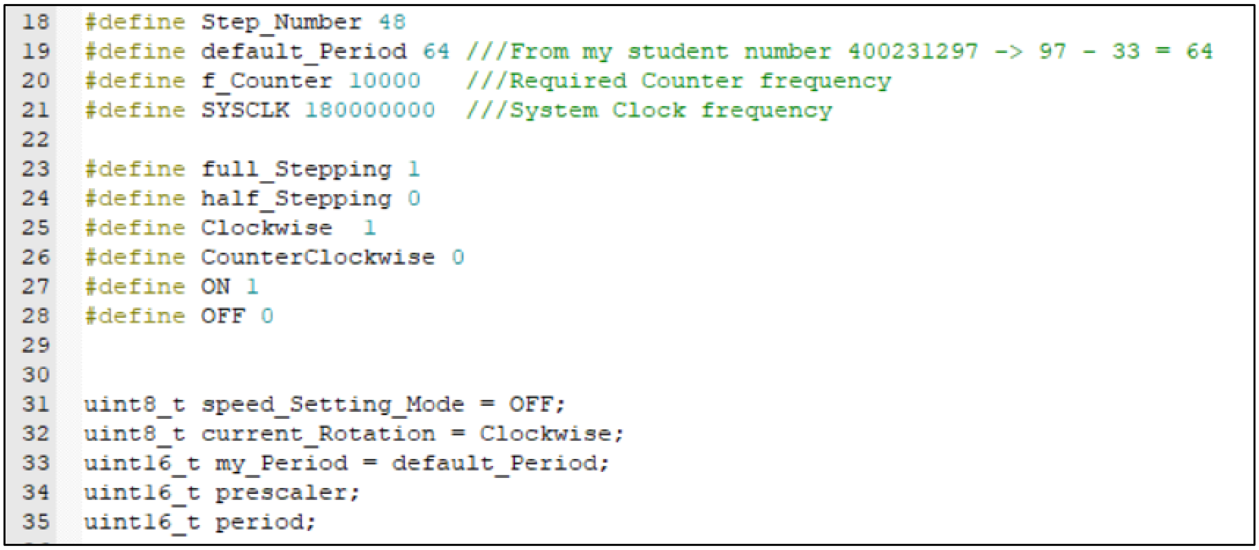
4. SYSCLK Frequency = 180 MHz   Counter Frequency = 10 kHz

   Prescaler = $\frac{f\_SYS}{f\_CNT} - 1 = \frac{180\ MHz}{10*1000\ Hz} - 1 = \mathbf{17999}$

   a. At Full-Stepping: OCR = 1.3333 * 10000 - 1 = **13332**

   b. At Half-Stepping: OCR = 0.6667 * 10000 - 1 = **6666**

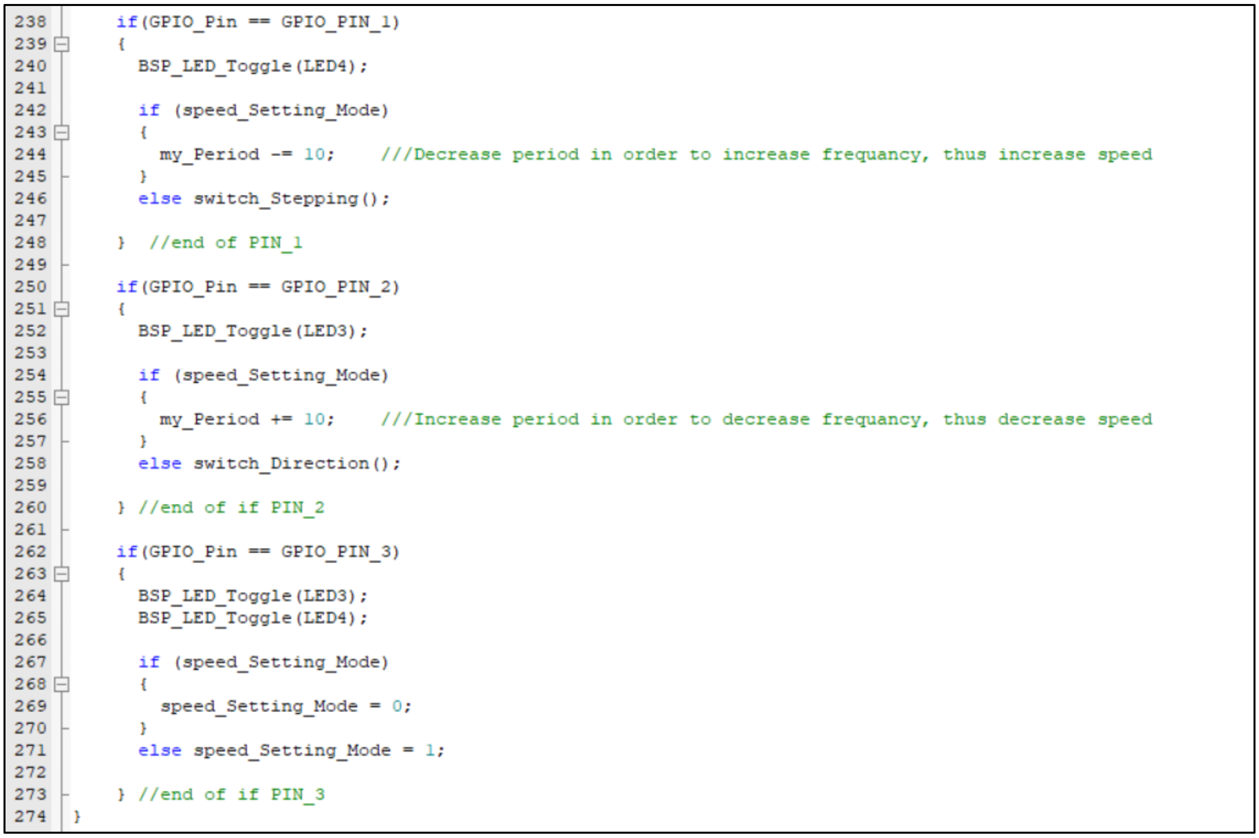
5. The C code for TIM3 and the stepper motor to implement the clock is written below.



```c
#define Step_Number 48
#define default_Period 64 ///From my student number 400231297 -> 97 - 33 = 64
#define f_Counter 10000   ///Required Counter frequency
#define SYSCLK 180000000  ///System Clock frequency

#define full_Stepping 1
#define half_Stepping 0
#define Clockwise  1
#define CounterClockwise 0
#define ON 1
#define OFF 0


uint8_t speed_Setting_Mode = OFF;
uint8_t current_Rotation = Clockwise;
uint16_t my_Period = default_Period;
uint16_t prescaler;
uint16_t period;
```

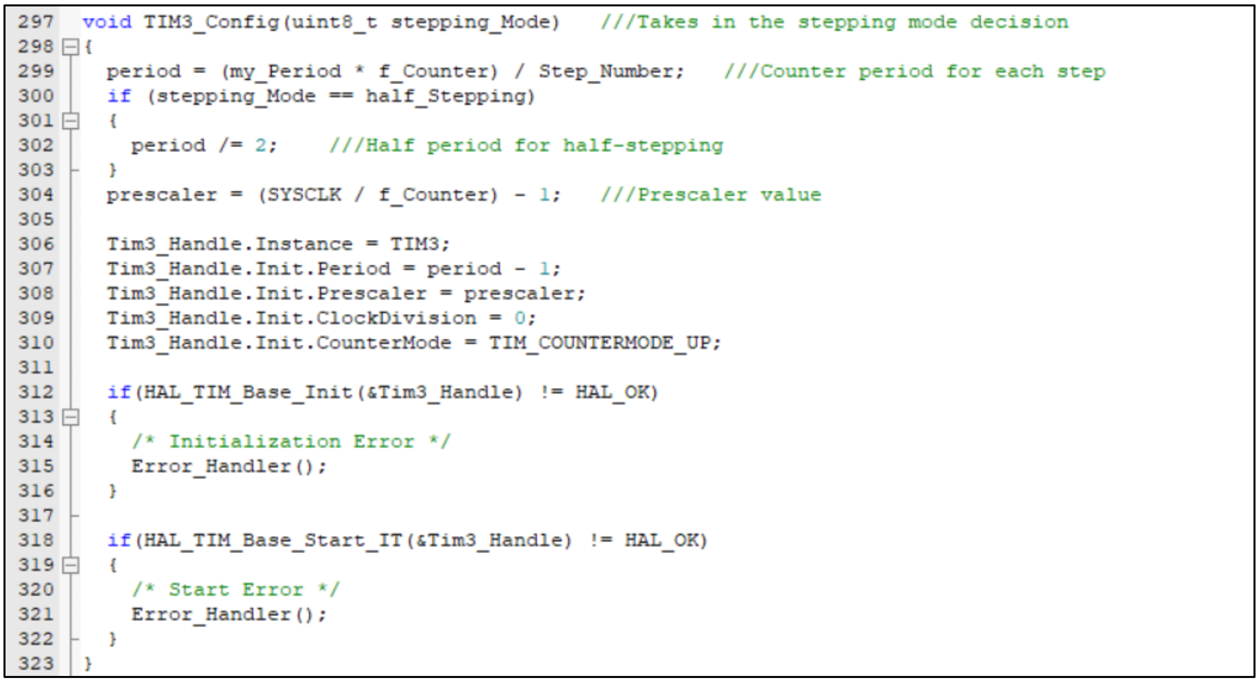
Figure 1: Defines and Constants

3TA4 Lab #5 Report



```
    if(GPIO_Pin == GPIO_PIN_1)
    {
      BSP_LED_Toggle(LED4);

      if (speed_Setting_Mode)
      {
        my_Period -= 10;    ///Decrease period in order to increase frequancy, thus increase speed
      }
      else switch_Stepping();

    }  //end of PIN_1

    if(GPIO_Pin == GPIO_PIN_2)
    {
      BSP_LED_Toggle(LED3);

      if (speed_Setting_Mode)
      {
        my_Period += 10;    ///Increase period in order to decrease frequancy, thus decrease speed
      }
      else switch_Direction();

    } //end of if PIN_2

    if(GPIO_Pin == GPIO_PIN_3)
    {
      BSP_LED_Toggle(LED3);
      BSP_LED_Toggle(LED4);

      if (speed_Setting_Mode)
      {
        speed_Setting_Mode = 0;
      }
      else speed_Setting_Mode = 1;

    } //end of if PIN_3
}
```

Figure 2: Buttons to Change Speed, Switch Stepping Mode and Rotation Direction



```
void TIM3_Config(uint8_t stepping_Mode)    ///Takes in the stepping mode decision
{
  period = (my_Period * f_Counter) / Step_Number;    ///Counter period for each step
  if (stepping_Mode == half_Stepping)
  {
    period /= 2;    ///Half period for half-stepping
  }
  prescaler = (SYSCLK / f_Counter) - 1;    ///Prescaler value

  Tim3_Handle.Instance = TIM3;
  Tim3_Handle.Init.Period = period - 1;
  Tim3_Handle.Init.Prescaler = prescaler;
  Tim3_Handle.Init.ClockDivision = 0;
  Tim3_Handle.Init.CounterMode = TIM_COUNTERMODE_UP;

  if(HAL_TIM_Base_Init(&Tim3_Handle) != HAL_OK)
  {
    /* Initialization Error */
    Error_Handler();
  }

  if(HAL_TIM_Base_Start_IT(&Tim3_Handle) != HAL_OK)
  {
    /* Start Error */
    Error_Handler();
  }
}
```

Figure 3: Configuration for the Counter Timer