

1. *Angular Resolution* = $\frac{360^\circ}{48^\circ} = 7.5^\circ$
2. Student number is 400179088, last two digits would be 88. Thus the period is determined to be 55 seconds.
3. Half step = 96 steps, Full step = 48 steps

$$\frac{55}{96} = 0.573 \text{ sec/step} \leftarrow \text{Half step}$$

$$\frac{55}{48} = 1.146 \text{ sec/step} \leftarrow \text{Full step}$$

4. Under the assumption that SysClock is 180MHz and Timer Clock Frequency is 10KHz (from previous labs)

$$\text{Prescaler} = \frac{180\text{MHz}}{10\text{KHz}} - 1 = 17999$$

$$\text{OCR}_{\text{Half}} = 0.573 * 10\text{KHz} = 5730$$

$$\text{OCR}_{\text{Full}} = 1.146 * 10\text{KHz} = 11460$$

5. Assume that all the other relevant code needed is pre-defined

```
//Assume system core clock to be 180 MHz, and APB1 to be 45MHz, and timer clock to be 10KHz
// Timer 3 Config
void TIM3_Config(void){
    Tim3_PrescalerValue = (uint16_t) ((SystemCoreClock)/10000)-1;

    Tim3_Handle.Instance = TIM3;

    Tim3_Handle.Init.Period = 55;
    Tim3_Handle.Init.Prescaler = Tim3_PrescalerValue;
    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();
    }
}
```

Figure 1:Tim3_Config

```
//Untested, in theory should work, may need tweaking
void GPIO_Config(GPIO_InitTypeDef pins[4]){
    for (int i = 0; i < 4; i++){
        pins[i].Mode = GPIO_MODE_OUTPUT_PP;
        pins[i].Pull = GPIO_PULLUP;
        pins[i].Speed = GPIO_SPEED_FREQ_VERY_HIGH;
        if(i % 4 == 0){
            pins[i].Pin = GPIO_PIN_2;
            HAL_GPIO_Init(GPIOA, &pins[i]);
        }
        else if(i % 4 == 1){
            pins[i].Pin = GPIO_PIN_3;
            HAL_GPIO_Init(GPIOA, &pins[i]);
        }
        else if(i % 4 == 2){
            pins[i].Pin = GPIO_PIN_6;
            HAL_GPIO_Init(GPIOB, &pins[i]);
        }
        else if(i % 4 == 3){
            pins[i].Pin = GPIO_PIN_7;
            HAL_GPIO_Init(GPIOB, &pins[i]);
        }
    }
}
```

Figure 2: GPIO_Config

```
void fullStep(void){
    if(state == 0){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
    }else if(state == 1){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
    }else if(state == 2){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
    }else if(state == 3){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
        state = 0;
    }
}

void halfStep(void){
    if(state == 0){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
    }else if(state == 1){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
    }else if(state == 2){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
    }else if(state == 3){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
    }else if(state == 4){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
    }else if(state == 5){
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
    }else if(state == 6){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
    }else if(state == 7){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
        state = 0;
    }
}
```

Figure 3: Full-step and Half-step

```
83 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
84 {
85     if(GPIO_Pin == KEY_BUTTON_PIN) //GPIO_PIN_0
86     {
87         | mode = !mode; //Toggle mode
88     }
89 }
90 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
91 {
92     if((*htim).Instance == TIM3){
93         | if(mode == 0){
94             | fullStep();
95         }else{
96             | halfStep();
97         }
98         | state++;
99     }
100 }
```

Figure 4: UB Config and Timer Callback