Lab 5: Stepper Motor Control

Instructor:
Dr. Carl Latino
carl.latino@okstate.edu

Graduate Teaching Assistant:
Francisco E. Fernandes Jr.
feferna@okstate.edu

School of Electrical and Computer Engineering
Oklahoma State University
Fall 2019

Lab Assignment

- **Write an Assembly program to turn the stepper motor exactly 360 degrees clockwise by using full-stepping and stop.**

Schedule and Grading

- **You MUST demo a working LAB 5 on October 14, 2019 by the end of your lab section!**
- **Grading for Lab 5:**
  - **Pre-lab quiz: 2 points! Read Chapter 16!**
    - **Due on October 14 at 1:20pm.**
  - **Functionality and Correctness: 8 points.**
    - **No partial credits!** Or it works or it doesn't!
- **Grading penalization:**
  - Students who disrupt the lecture by talking and not paying attention will lose 2 points in their lab's grade!
  - Students who do not follow the lab safety procedures (e.g. coming to lab with shorts and flip flops) will lose 1 points in their lab's grade!

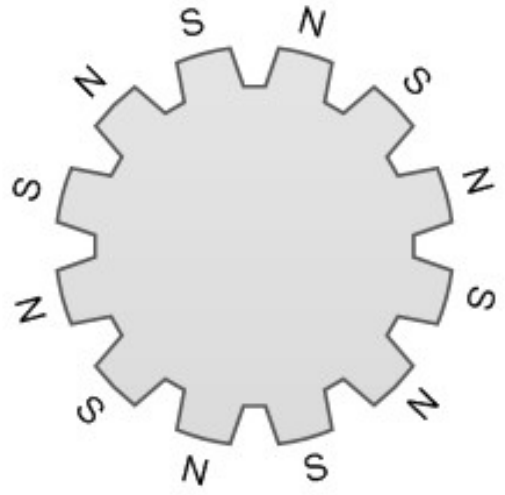
Step Angle

$$Step\ Angle = \frac{360\ degrees}{steps\ per\ revolution}$$

$$steps\ per\ revolution = P \times T$$

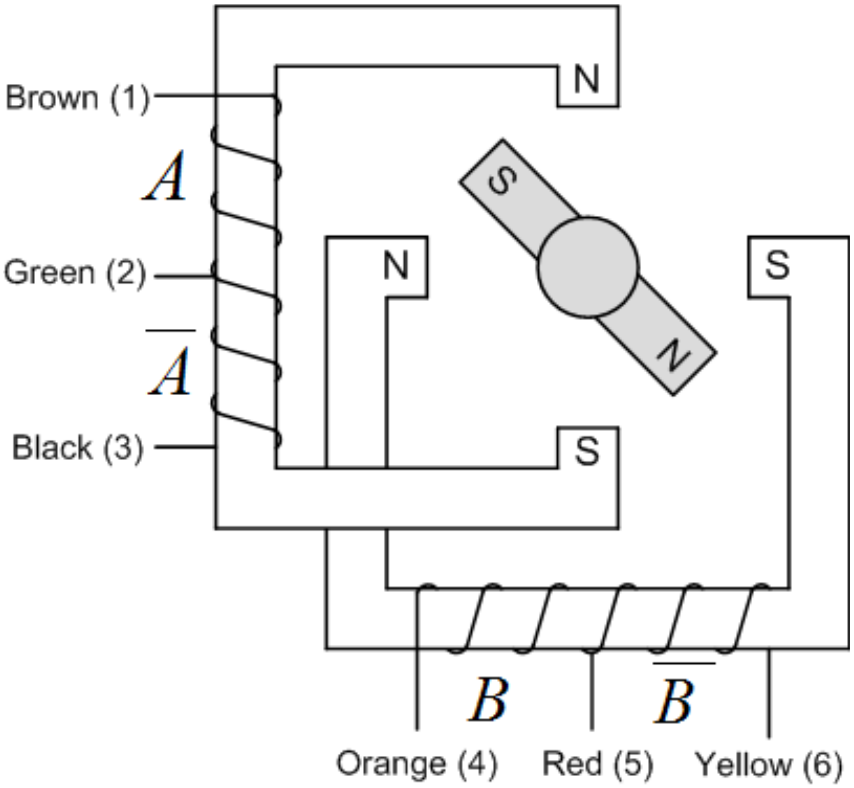
where P is the total number of phases on the stator, and T is the total number permanent-magnetic poles available on the rotor.

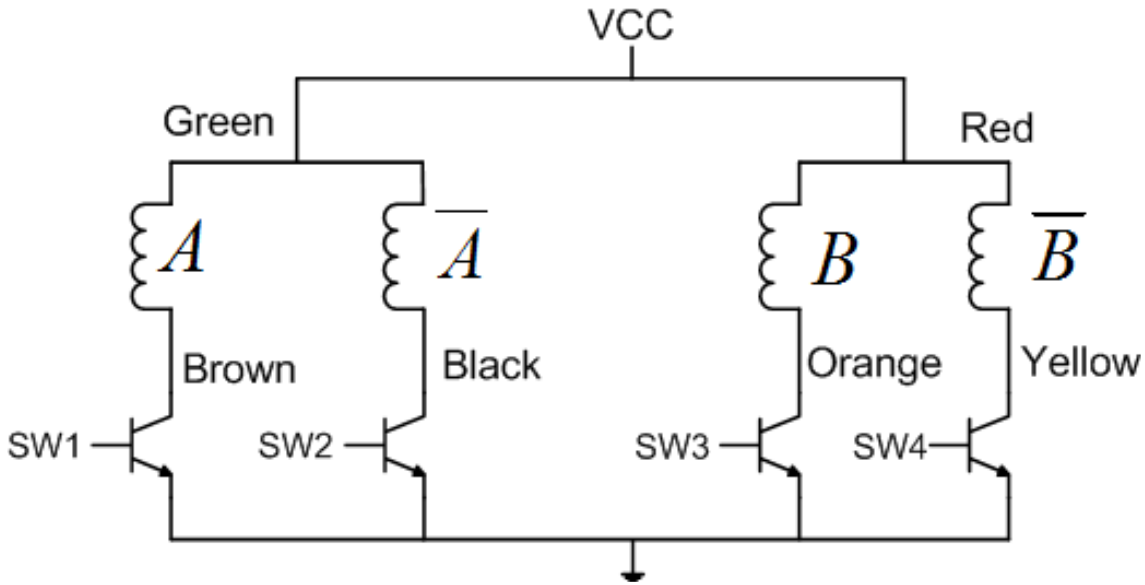
Rotor with only two poles



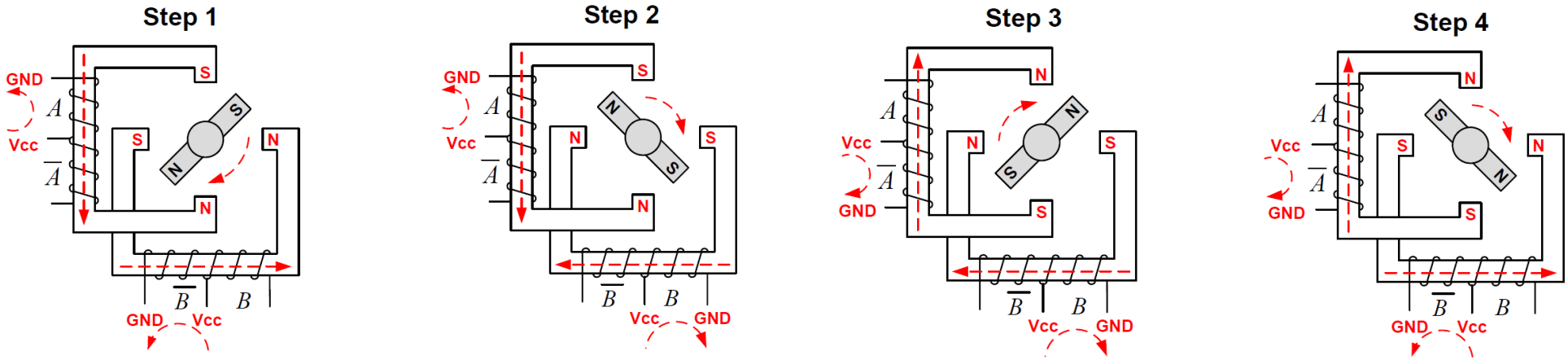
Rotor with 12 poles

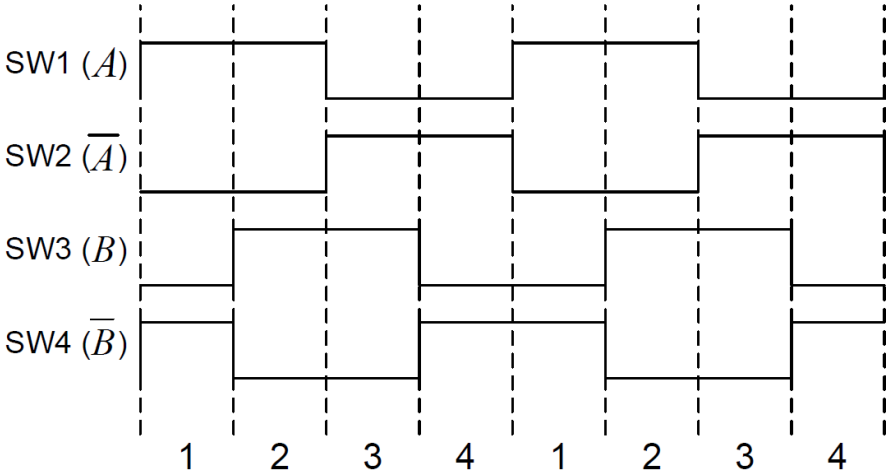
Unipolar Stepper Motor





Full Stepping





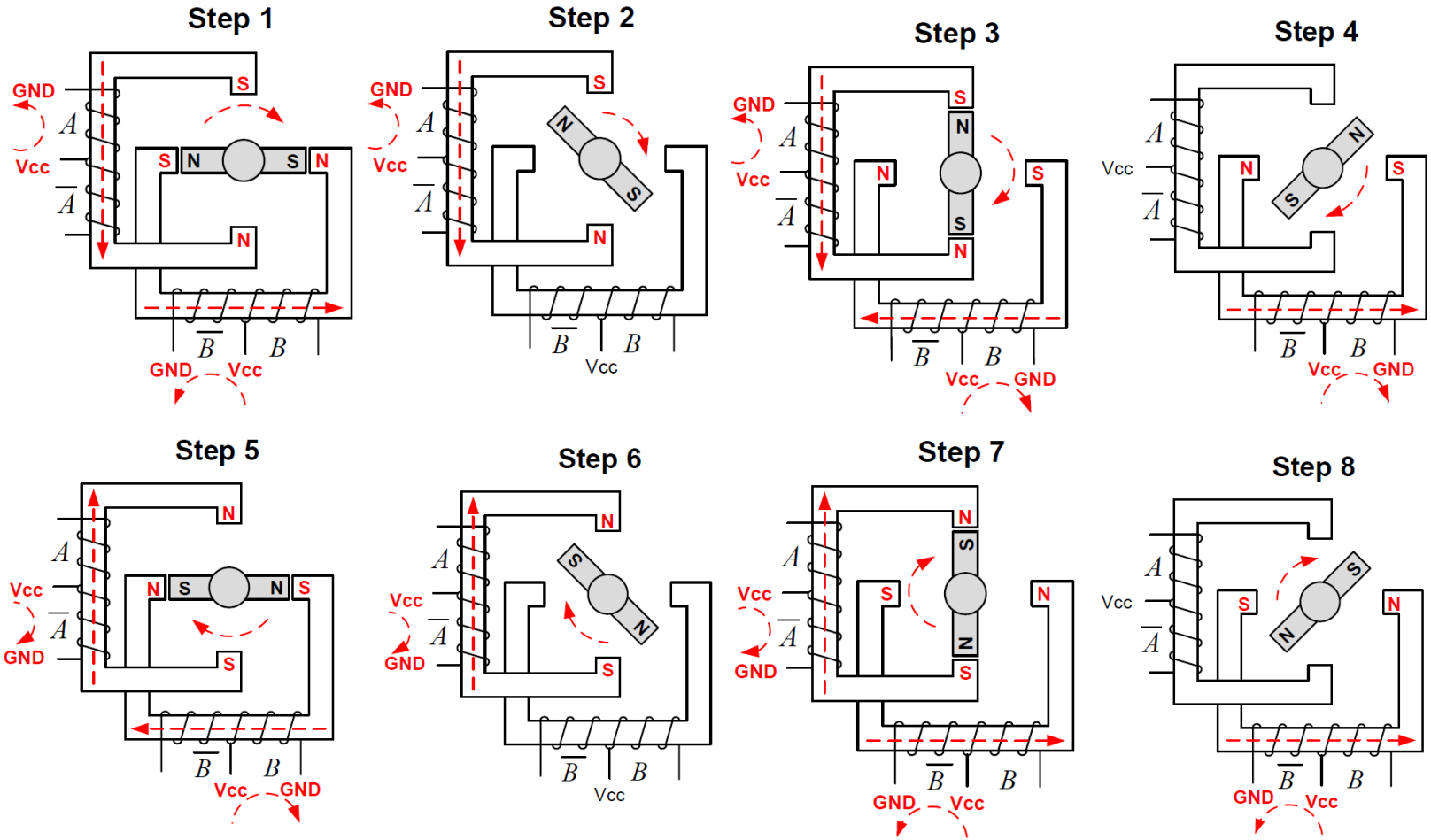
Clockwise sequence:

$$A\,\bar{B}, A\,B, \bar{A}\,B, \bar{A}\,\bar{B}$$

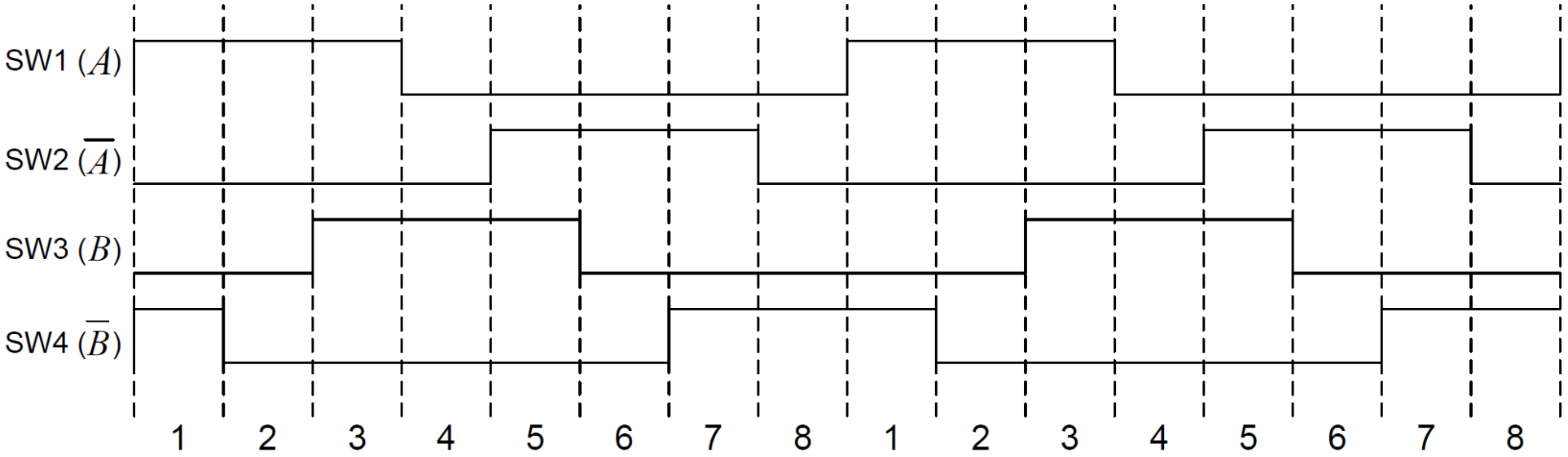
Counter-clockwise sequence:

$$\bar{A}\,\bar{B}, \bar{A}\,B, A\,B, A\,\bar{B}$$

Half Stepping



Half Stepping



Clockwise Sequence:

$$A\,\bar{B}, A, A\,B, B, \bar{A}\,B, \bar{A}, \bar{A}\,\bar{B}, \bar{B}$$

Counter-clockwise Sequence:

$$\bar{B}, \bar{A}\,\bar{B}, \bar{A}, \bar{A}\,B, B, A\,B, A, A\,\bar{B}$$

28BYJ-48 Stepper Motor

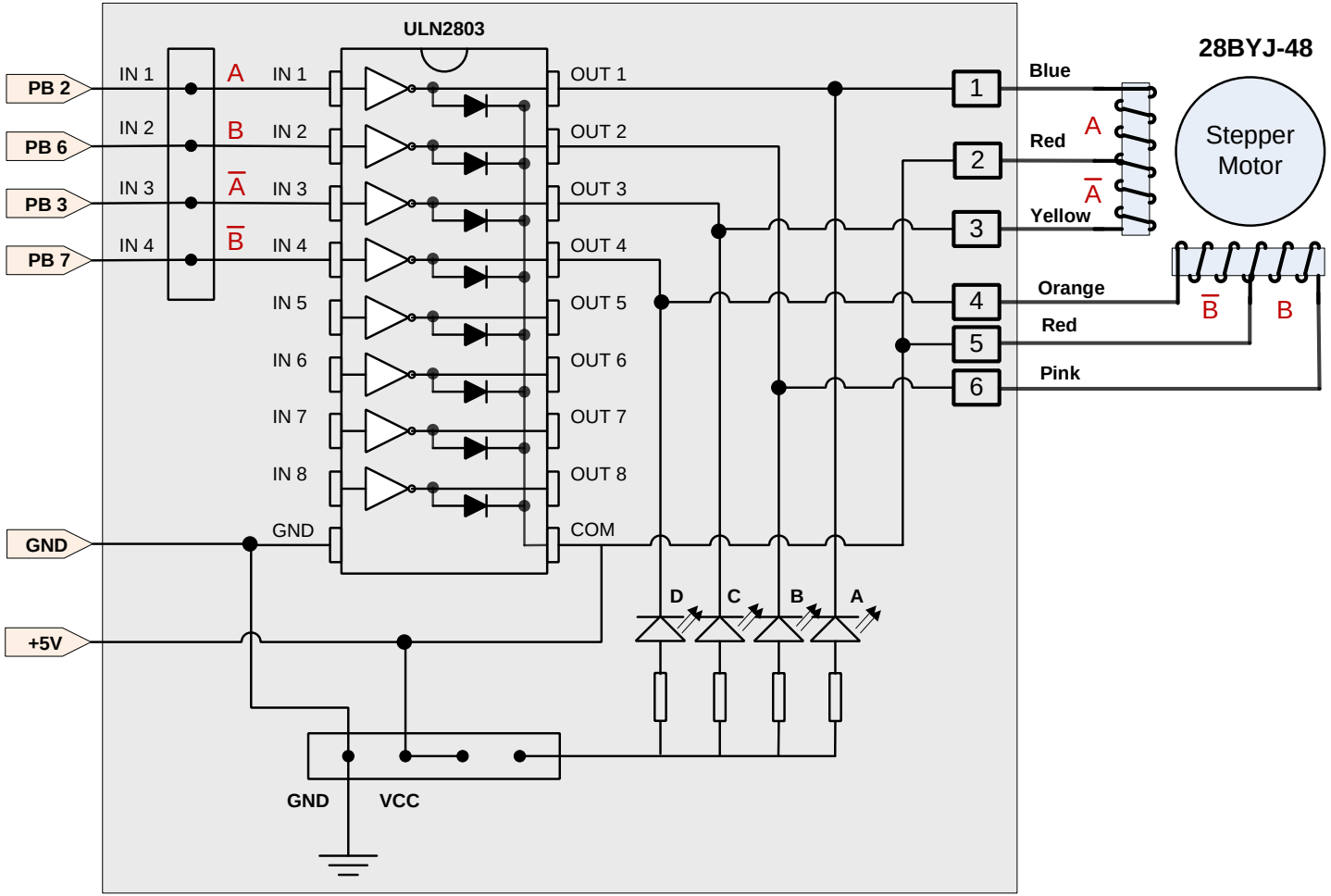
Full-stepping

- *Internal motor: 32 steps per revolution*
- *Great reduction ratio: 1/63.68395, approximately 1/64*
- *Thus, it takes* $32 \times 64 = \mathbf{2048}$ **steps** *per revolution for the output shaft*

Half-stepping

- *Internal motor: 64 steps per revolution*
- *Great reduction ratio: 1/63.68395* $\approx 1/64$
- *Thus, it takes* $64 \times 64 = \mathbf{4096}$ **steps** *per revolution for the output shaft*

28BYJ-48 Stepper Motor

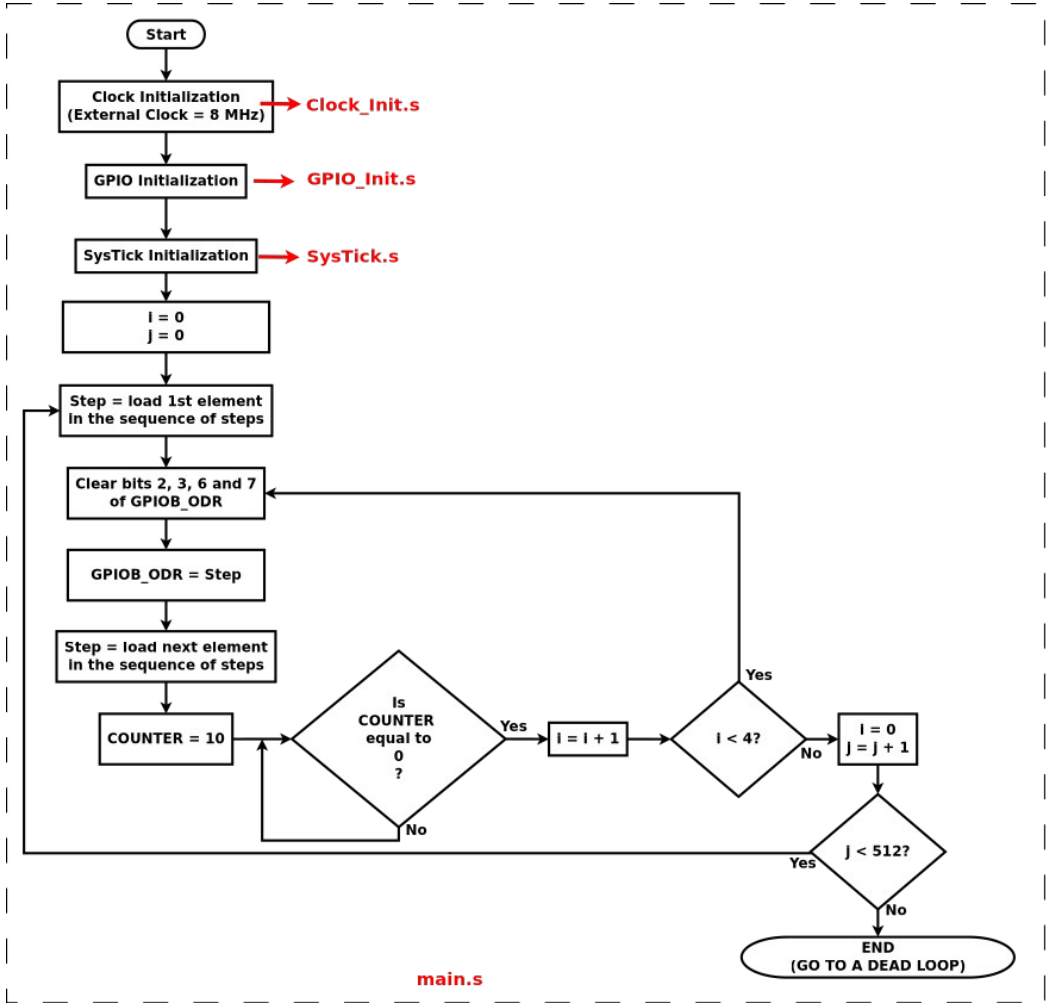


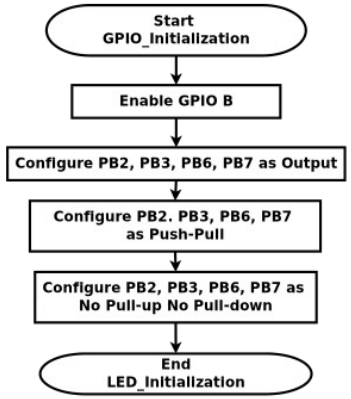
Lab 5: Start-up Code

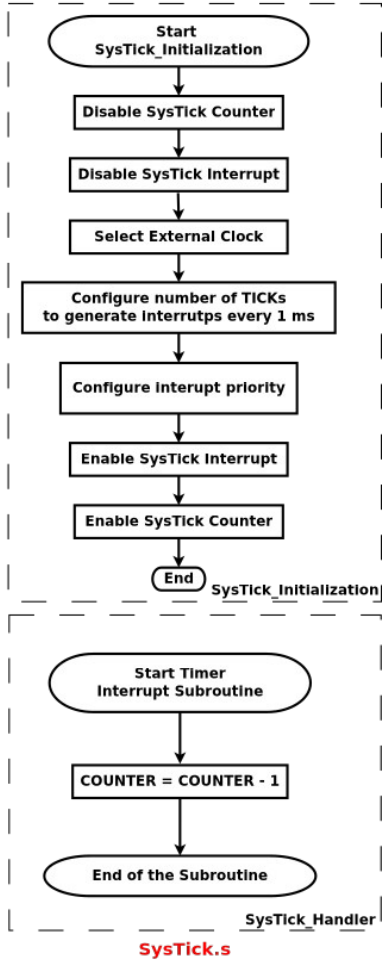
- **No start-up code is given for this lab!**
- **You have to write ALL the necessary CODE for this Lab!**

- **Your experience from the previous labs should help you writing the necessary code for this one!**
- **Note:**
  - You may ask as many question as you need. The T.A. will answer any and every question you may have about the lab!
  - However, if you have mistakes in your code, the T.A. will not fix those mistakes for you.
  - It is your responsibility to write the correct missing code.

Lab 5: Flowchart







Creating Arrays in Assembly (Cortex-M4)

```
.syntax unified

.align

.section .data
steps:   .word 0xFF, 0xFF, 0xFF, 0xFF

.include "stm32l476xx_constants.s"
.include "Clock_Init.s"
.include "SysTick.s"
.include "GPIO_Init.s"

.global main
.global SysTick_Handler

.section .text
```

Array called "steps" with four positions equal to 0xFF.

This array should contain the outputs to make the motor perform a full stepping sequence.

Note: You have to figure out the elements of this array!

Reading an Array in Assembly (Cortex-M4)

```
LDR r4, =#steps// Load the MEMORY ADDRESS of the first element of the array called "steps" into r4

LDR r5, [r4]   // Load into r5 the contents of the memory address pointed by r4.
               // In this case, this will be the first element of the array "steps".

ADD r4, r4, #4 // Update the MEMORY ADDRESS to the NEXT element.
               // Always do r4 = r4 + 4 when updating to the next element in the array.

LDR r5, [r4]   // Now, r5 will be equal to the second element of the array "steps".
```