Project 3: Stepper Motor and SysTick Timer

Goals

1. Understand the basic concept of system timer (SysTick)
2. Learn how to use SysTick to create a time delay function
3. Understand the basic procedure of interrupt handling
4. Understand the limitation of GPIO output current, and learn to use Darlington transistor arrays to perform high-current driving with low input current
5. Understand the usage of full stepping and half stepping to control the speed and position of a stepper motor

Project Submission.

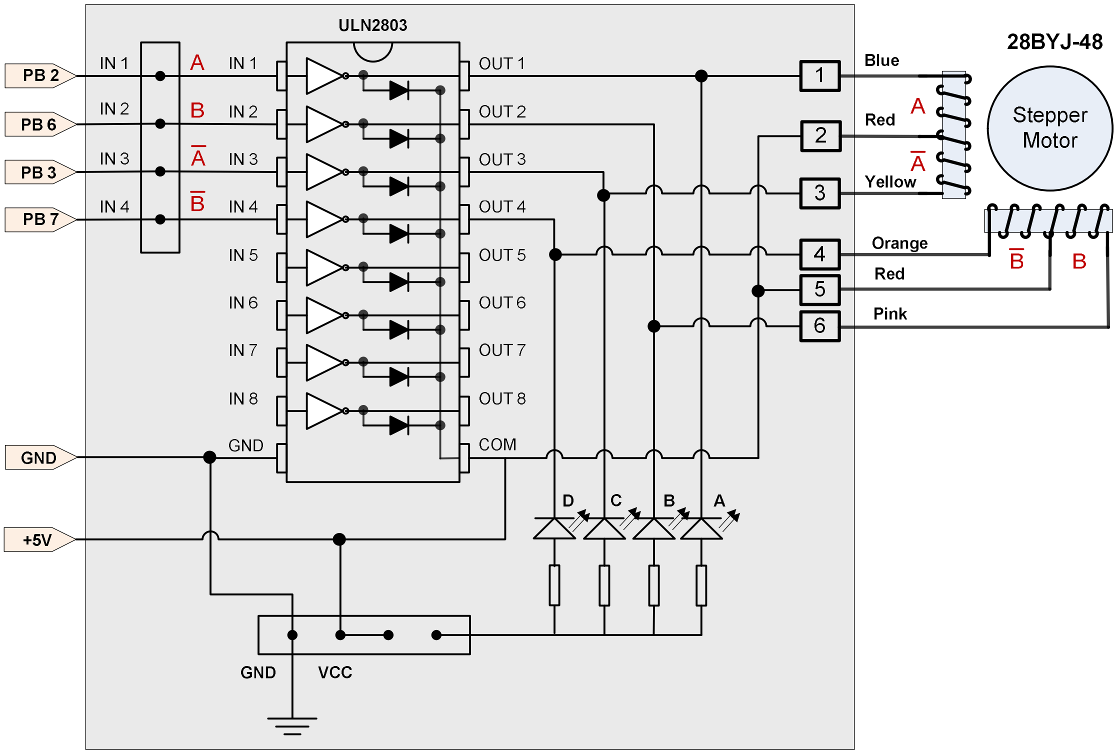
1. Zip your whole project folder and submit it on Blackboard.
   If your submission does not contain the whole project, 50% of the total points will be deducted.
   If your code cannot be compiled or built successfully, 50% of the total points will be deducted.
2. Write a README document (in PDF format) to briefly
   a. describe the basic structure of your project,
   b. explain how to test your project to demonstrate that it works correctly,
   c. describe your test procedure and results and use some pictures to show your test. (optionally, you can make a video to demonstrate how you have tested your project.)

Project Assignment

Write a Keil uVision 5 project with C code to generate delay by SysTick timer interrupt which will control the rotation speed of a stepper motor. Your code needs to use two different buttons of the joystick to give two different speeds to the stepper motor.

Press the up button of the joystick to turn the stepper motor exactly 360 degrees clockwise by using half-stepping, and press the down button to turn the motor exactly 180 degrees counter-clockwise by using half-stepping.

Interfacing the stepper motor requires four pins. You should select the following four pins to control the stepper motor: **PB 2**, **PB 6**, **PB 3**, and **PB 7**. The textbook provides a connection diagram for stepper motor *Mabuchi #PF35T*, which is very similar to the diagram below.



Note that the four output pins should be configured as open-drain. In your project, you can output the following bit sequence for either full stepping or half stepping. Full stepping bit sequence has 4 output bit combinations continuously looping, and half stepping bit sequence has 8 output bit combinations continuously looping.

Between any two bit combinations in a bit sequence, your code needs to generate a certain amount of time delay by using SysTick's delay function (recall it is in an example project). For example, to output the first bit combination in the full stepping bit sequence, i.e., $A = 1, \bar{A}$=0, $B = 1, \bar{B}$=0, your code needs to output PB2=1, PB3=0, PB6=1, and PB7=0.

Full Stepping Bit Sequence (this sequence includes 4 pulses, i.e., 4 step angles, then 512=2048/4 such sequences will make outside shaft rotate 360 degrees).

A	1	1	0	0
$\bar{A}$	0	0	1	1
B	1	0	0	1
$\bar{B}$	0	1	1	0

Half Stepping Bit Sequence (this sequence includes 8 pulses, still has 4 step angles due to half stepping, so 512=2048/4 such sequences will make outside shaft rotate 360 degrees).

A	1	1	1	0	0	0	0	0
$\bar{A}$	0	0	0	0	1	1	1	0
B	1	0	0	0	0	0	1	1
$\bar{B}$	0	0	1	1	1	0	0	0

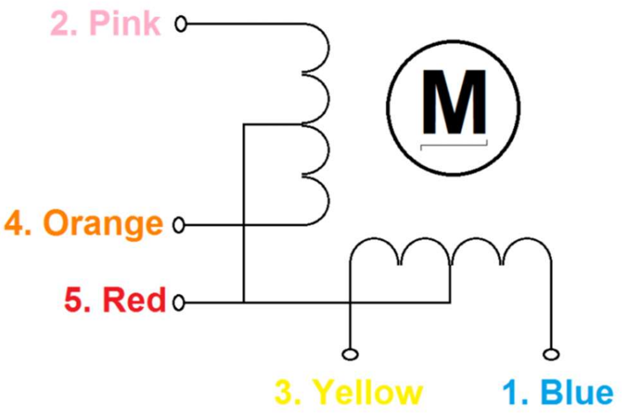
Grading Rubric

1. (10 pts) for README document.
2. (40 pts) for your project code to implement the above assigned task.

Preparation

1. Read the textbook Chapter 16 Stepper Motor
2. Watch video tutorial: How the Stepper motors are made and how they operate (Credit goes to *pcbheaven*)
   a. Part 1 (5 minutes): http://www.youtube.com/watch?v=MHdz3c6KLrg
   b. Part 2 (8 minutes): http://www.youtube.com/watch?v=t-3VnLadIbc
3. Read Chapter 11.7 System Timer
4. Processor Clock. You may use the example project's system clock code, which gives 80MHz clock signal. For your reference, four different clock sources can be used to drive the system clock (SYSCLK):
   a) 16MHz HSI (high-speed internal) oscillator clock
   b) 4-48MHz HSE (high-speed external) oscillator clock
   c) PLL (phase-locked loop) clock that is clocked by HSI, HSE, and MSI.
   d) MSI (multispeed internal) oscillator clock

Stepper Motor



The motor set in your lab kit has a ULN2003 Darlington Array.

Motor model	**28BYJ-48**	Number of phases	4
Rated voltage	5V DC	Geared reduction ratio	1/64
DC resistance per phase	50Ω±7%(25℃)	Pull in torque	>300gf.cm / 5VDC 100pp

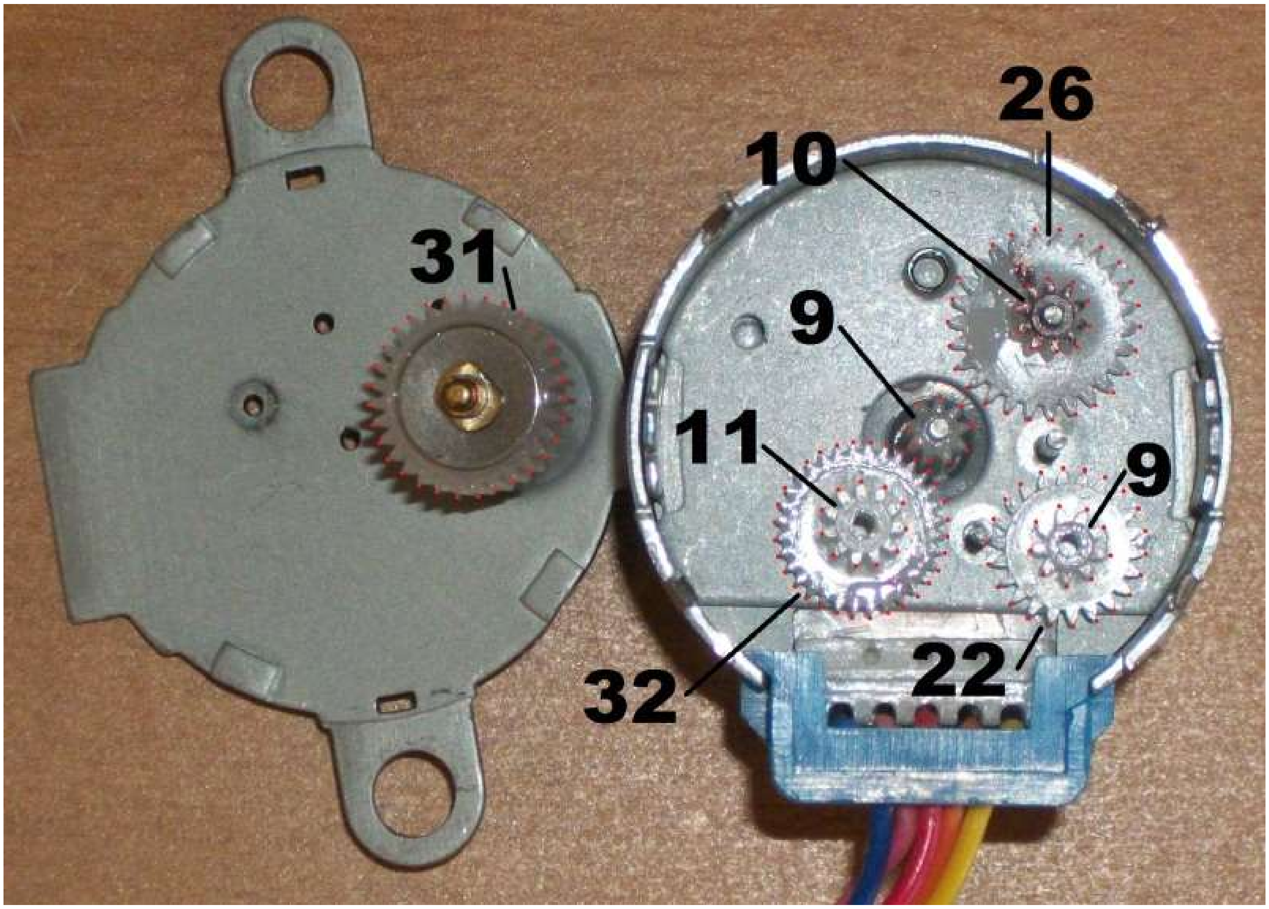


image from *forum.arduino.cc*

The gear ratio is:

$$\frac{31 \times 32 \times 26 \times 22}{11 \times 10 \times 9 \times 9} = 63.68395$$

If the output shaft rotates 1 resolution (gear with 31 teeth in the figure), the internal shaft (gear with 9 teeth in the middle) must rotate approximately 64 resolutions.

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

Warning: Motor Overheating

The motor constantly draws electrical currents. The motor will be overheated if you leave the power on for an extended period. **Make sure to disconnect the power (Vcc) to the Darlington array if you are not debugging/testing it.**