M480 Stepper Motor

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The example code is M480 GPIO control 4-Phase, 8-Beat stepper motor.
BSP Version	M480 Series BSP CMSIS V3.03.001
Hardware	NuMaker-PFM-M487 V3.0

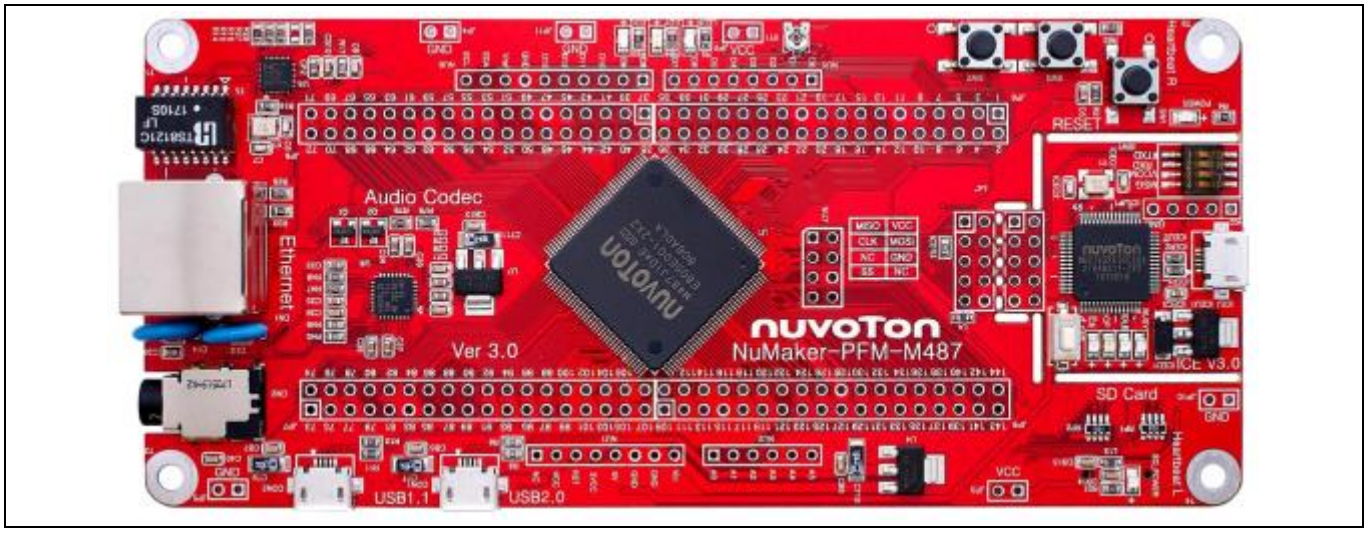
1 Function Description

1.1 Introduction

This code uses M480 timer to control I/O that can drive stepper motor by Darlington IC(ULN2803). This document includes everything you need to know about controlling a 28BYJ-48 stepper motor with the ULN2003 driver and M480 timer.
The 28BYJ-48 is one of the cheapest stepper motors you can find. Although it is not super accurate or powerful, it is a great motor to use for smaller projects or if you just want to learn about stepper motors.

For more information about the NuMaker-PFM-M487 board, please visit Nuvoton website to download the UM_NuMaker-PFM_M487_User_Manual:

https://www.nuvoton.com/hg/products/microcontrollers/arm-cortex-m4-mcus/User-Manual/?__locale=en&resourcePage=Y&category=&pageIndex=3



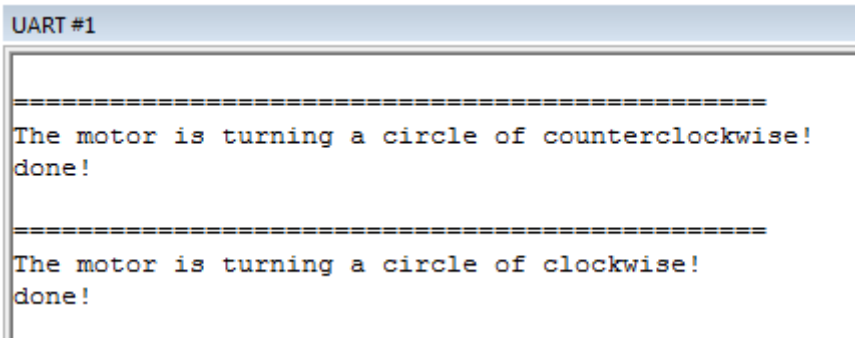
NuMaker-PFM-M487 Board

1.2 Principle

The most commonly used stepper motor is the 28-BYJ48 stepper motors. The motor has a 4 coil unipolar arrangement and each coil is rated for +5V hence it is relatively easy to control with any basic microcontrollers. These motors have a stride angle of 5.625°/64, this means that the motor will have to make 64 steps to complete one rotation and for every step it will cover a 5.625° hence the level of control is also high.
The 28BYJ-48 stepper motor consumes high current and hence, we will need to use a driver IC like the ULN2003 in order to control the motor with M480. Known for its high current and high voltage capacity, the ULN2003 gives a higher current gain than a single transistor and enables the low voltage and low current output of a microcontroller to drive a higher current stepper motor

1.3 Demo Result

After executed this example code, user will be seeing that M480 control stepper motor(28-BYJ48) rotates one turn and reverses one turn with GPIO.



2 Code Description

This main() function is used to initialize platform for timer and GPIO. After that, the motor is turning a circle of clockwise and counterclockwise.

```
void main(void)
{
    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Init UART0 for printf */
    UART_Open(UART0, 115200);

    /* Open Timer0 in periodic mode, set timer frequency to 200HZ */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 200);

    /* Enable timer interrupt */
    TIMER_EnableInt(TIMER0);

    /* Enable Timer0 NVIC */
    NVIC_EnableIRQ(TMR0_IRQn);

    /* Configure PA.0 PA.1 PA.2 PA.3 set open drain mode, motor control IO */
    GPIO_SetMode(PA, (BIT0 | BIT1 | BIT2 | BIT3), GPIO_MODE_OPEN_DRAIN);
    printf("\n================================================\n");
    printf("The motor is turning a circle of counterclockwise!\n");
    Rotation_MOTOR(d360, counterclockwise);
    printf("done!\n");
    printf("\n================================================\n");
    printf("The motor is turning a circle of clockwise!\n");
    Rotation_MOTOR(d360, clockwise);
    printf("done!\n");
    while(1);
}
```

This TMR0_IRQHandler () function is used to set GPIO status to control Stepper Motor.

```
unsigned char CW[8] = {0x08,0x0C,0x04,0x06,0x02,0x03,0x01,0x09};
unsigned char CCW[8] = {0x09,0x01,0x03,0x02,0x06,0x04,0x0c,0x08};
unsigned char Dir_flag=0;
volatile unsigned int  Motor_Count=0;
void TMR0_IRQHandler(void)
{
    if(TIMER_GetIntFlag(TIMER0) == 1) {
        /* The motor is turning a circle of clockwise! */
        if(Dir_flag)
        {
            /* calculation the step of motor */
            if(Motor_Count%Motor_Step==0)
            PA->DOUT=CW[0];
            else
            PA->DOUT=CW[Motor_Step-(Motor_Count%Motor_Step)];
        }
        else
        /* The motor is turning a circle of counterclockwise! */
        {
            /* calculation the step of motor */
            if(Motor_Count%Motor_Step==0)
            PA->DOUT=CCW[0];
            else
            PA->DOUT=CCW[Motor_Step-(Motor_Count%Motor_Step)];
        }
        Motor_Count--;
        if(Motor_Count==0)
        {
            TIMER_Stop(TIMER0);
        }
        /* Clear Timer0 time-out interrupt flag */
        TIMER_ClearIntFlag(TIMER0);
    }
}
```

3 Software and Hardware Environment

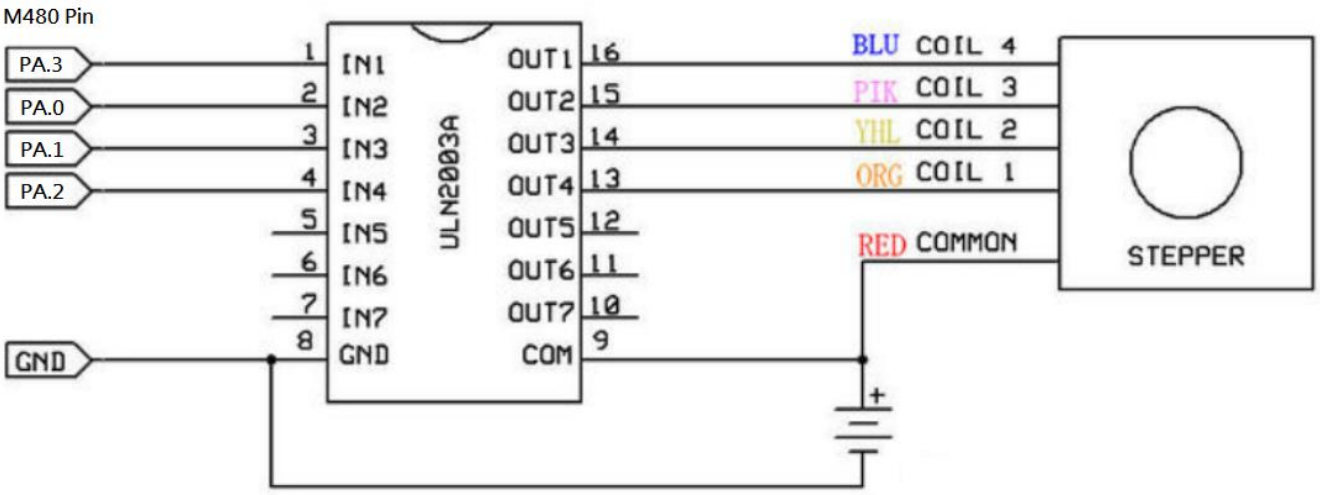
Software Environment

- BSP version
  - M480 Series BSP CMSIS V3.03.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuMaker-PFM-M487 V3.0 board
- Diagram

The below picture is to show that is the diagram of example code.



4 Directory Information

- EC_M480_StepperMotor_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code?

1. Browsing into sample code folder by Directory Information (section 4) and double click Step_Motor.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Jul. 18, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.