Open-Loop BLDC Motor Control

Example Code Introduction for 8-bit NuMicro® Family

Information

Application	This example code based on N76E003 for implementing the open-loop BLDC motor controlled
BSP Version	N76E003_BSP_Keil_C51_V1.0.6
Hardware	NuTiny-N76E003

1 Function Description

1.1 Introduction

In this example code, the NuMicro 8051 N76E003 series is used as an example. Based on PWM and GPIO interrupt of N76E003, it can implement open-loop Brushless DC Motor (BLDC Motor). Pulse Width Modulation (PWM) is a technique to change the output energy by modifying the duty cycle. It can apply to modify the motor speed, control valve … etc.

BLDC Motor rectifies by electric device instead of traditional brush. It can increase the reliability and endurance. The advantage of BLDC Motor is smaller volume, higher reliability, and higher endurance than same rated power brush DC Motor. The disadvantage of BLDC Motor is need the electric device like microcontroller for controlling.

Basic BLDC Motor is equipped with the hall sensor to feedback the signal of rotor state. Microcontroller will output the 6 sets PWM control signal for 3 sets MOS upper/lower-arms to control 3-phase motor. And microcontroller needs 3 GPIO for receiving hall signal for detecting the rotor state.

Open-loop controller is a simple mathematical model of system. Because there is no feedback signal to modify itself to ideal state automatically. The robustness of open-loop system is worse than close-loop system. However, if the system does not need the high accuracy, it can implement on open-loop controller.

This example code is based on open-loop controller to control the BLDC motor and modified the motor speed by variable resistor.

2 Code Description

```
/*---------------------------------------------------------------------------------------*/
/* Pin Functions                                                                         */
/*---------------------------------------  ----------------------------------------------*/
//P07 - HALL Sensor W Phase, HALL_W
//P06 - HALL Sensor V Phase, HALL_V
//P05 - HALL Sensor U Phase, HALL_U
//P12 - Upper-arm MOS control of U phase, uh
//P14 - Lower-arm MOS control of U phase, ul
//P10 - Upper-arm MOS control of V phase, vh
//P00 - Lower-arm MOS control of V phase, vl
//P01 - Upper-arm MOS control of W phase, wh
//P03 - Lower-arm MOS control of W phase, wl
//P15 - Motor on/off switch
//P02 - UART RX
//P16 - UART TX
//P17 - ADC input
```

2.1 Get the target speed by ADC sampling variable resistor

```
unsigned int GetTargetSpeed(void)
{
    unsigned int data s16TargetSpeed;
    /* ADC will sample the variable resistor value on ADCRH. */
    /* Calculate the percentage of Max rotate speed 4500 rpm to target speed. */
    s16TargetSpeed = (((unsigned long int)4500 * (unsigned long int)ADCRH) / 255);

    /* Clear ADN interrupt flag and re-trigger ADC to convert. */
    clr_ADCF;
    set_ADCS;
    return s16TargetSpeed;
}
```

2.2 Sets the the target speed directly

```
main()
{
……
    while (1)
    {
        /* Get Motor realtime speed by ADC */
        if (ADCF == ADC_CONVERT_FINISH)
        {
            s16TargetSpeed = GetTargetSpeed();
        }

        if (MOTOR_ON_OFF_SWITCH == MOTOR_OFF)
        {
            ……
        }
        else if (g_u8TimerIntCount >= 1) /* if the time past x * 10ms( x = 1 ), entering
this if. */
        {
            /* Modified the PWM duty for tracing the target speed */
            temp = ((((signed long int)0x1F3 * (signed long int)s16TargetSpeed)) / (signed
long int)4500);
            u16PWMDutyValue += (( temp > u16PWMDutyValue) ? 1 : (-1));

            /* Set new PWM duty */
            PWM0H = HIBYTE(u16PWMDutyValue);
            PWM0L = LOBYTE(u16PWMDutyValue);
            set_LOAD;
        }
        ……
    }
……
}
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - N76E003_BSP_Keil_C51_V1.0.6
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-N76E003 * 1
  - Brushless DC Motor * 1
  - Variable Resistor * 1
  - On/Off Switch* 1
  - Nu-Bridge * 1

■ Diagram

Following the diagram to connected the components. User can modify the speed of motor by variable resistor and on/off switch to turn on/off the motor.

For printing the debug message, user needs to connect two wire, the UART_RX(P0.2), UART_TX(P1.6) of N76E003 to Nu-Bridge. Then set the COM Port number and Baudrate in terminal tool. User can find the COM Port number in Device Manager of PC and it will display “NuBridge Virtual Com Port (COMX)”. The Baudrate should be 115200.

Enable / disable motor	↔	I/O (P1.5)	I/O (P0.7)	↔	HALL_W
Motor speed control(VR)	↔	ADC (P1.7)	I/O (P0.6)	↔	HALL_V
User Control			I/O (P0.5)	↔	HALL_U
			PWM (P1.2)	↔	UH
			PWM (P1.4)	↔	UL
			PWM (P1.0)	↔	VH
UART RX	↔	TX(P1.6)	PWM (P0.0)	↔	VL
UART TX	↔	RX(P0.2)	PWM (P0.1)	↔	WH
			PWM (P0.3)	↔	WL
Nu-Bridge		N76E003			BLDC Motor

4 Directory Information

- EC_N76E003_Open_Loop_BLDC_Motor_V1.00
  - Common — Common driver files
  - Include — Sample code header, SFR and device driver files
  - SampleCode
    - ExampleCode — Source file of example code
  - Startup — 8051 Startup file

5 How to Execute Example Code

1. This project supports Keil uVersion 5.26 and higher version
2. According to the chapter 4 Directory Information, enter the path EC_N76E003_Open_Loop_BLDC_Motor_V1.00\Sample_Code\ExampleCode\Open_Loop_BLDC_Motor_V1.00 folder and double click the OpenLoopBLDCMotor.uvproj
3. Enter the Keil compile mode window
   a. Build the project
   b. Download
   c. Start / Stop the debug mode
4. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Oct. 16, 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.