PID Control Operation in Fixed-point Format Using M031

Example Code Introduction for 32-bit NuMicro® Family

Document Information

Application	This example code uses the M031 series microcontroller (MCU) to realize PID control operation in fixed-point format.
BSP Version	M031_Series_BSP_CMSIS_V3.06.000
Hardware	NuMaker-M032SE V1.3

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. Overview

This PID example code does not require additional external hardware modules and can be executed by other NuMicro M0/M23 microcontroller (MCU).

The PID control is a common arithmetic operation used in feedback control system. The fundamental PID controller is consisted of proportional, integral and derivative terms. The 32-bit Cortex-M0/M23 processor, with built-in 32-bit multiplier and register shift function, is proper to do discrete-time PID control algorithm.

This example code implements the fixe-point arithmetic calculation in Q15.0 format, that is, the value range is in (-32768, 32767). User may refer to it to modify the code to meet different value requirement.

1.1 Principle

The block diagram of discrete-time PID controller is shown below, where n means the current state and T is the sampling time.

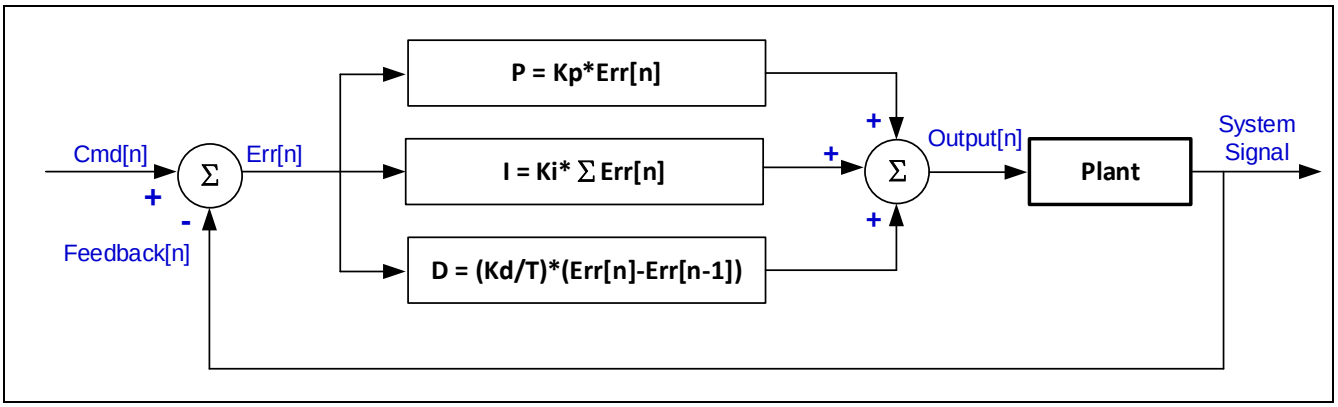

Figure 1-1 Block Diagram of PID Control

Where Kp, Ki and Kd are the 3 key parameters in PID control. Usually the sampling time is a constant, the term of Kd/T could be utilized as a new Kd. Therefore, the mathematical formula of discrete-time PID control is denoted as follows:

Output[n] = Kp*Err[n] + Ki*Σ(Err[n]) + Kd*(Err[n] – Err[n-1])

Since the Cortex-M0/M23 provides a 32-bit multiplier and register shifter, the MCU can execute the mathematical operation in 16-bit Q15.0 format with good efficiency. The example code demonstrates the PID control program in fixed-point Q15.0 format.

1.2 Demo Result

This example code is developed on the NuMaker-M032SE board. User may use the Keil IDE to open the example project, connect the board to PC through USB port and start a terminal AP with a COM port in the baud rate 115200 bsp. When the MCU runs the example code, the terminal AP will receive the upload data from MCU. The demo result is shown in Figure 1-2. The value of Output_PID will be increased and approach to target command at final, which is 50 in the example code.

```
CNT= (  16);   Output_PID= (49);  PID_Var.i32_I= (1616000);
CNT= (  17);   Output_PID= (49);  PID_Var.i32_I= (1624000);
CNT= (  18);   Output_PID= (49);  PID_Var.i32_I= (1632000);
CNT= (  19);   Output_PID= (50);  PID_Var.i32_I= (1640000);
CNT= (  20);   Output_PID= (50);  PID_Var.i32_I= (1640000);
CNT= (  21);   Output_PID= (50);  PID_Var.i32_I= (1640000);
```

Figure 1-2 Part of Upload Data to PC

2. Code Description

The PID arithmetic function code, which is located in *PID_Example.c*.

```
#include "PID_Example.h"

/*
 *    @brief        PID(Proportional-integral-derivative) control function
 *
 *    @param[in]    C              The variables of PID controller, inculding Kp, Ki, Kd,
                                   integral buffer and output limit
 *    @param[in]    i32_Cmd        The command input of PID controller.
 *    @param[in]    i32_Feedback   The feedback of PID controller.
 *
 *    @return       Output of PID operation
 *
 *    @details      Output = (command-feedback)*[PID operation].
 *                    The arithmetic operations is fixed-point in Q15.0 format
 *                    The value range of PID output in Q15.0 is from -32768 to 32767,
 *                    however, the value range of integral operation is in Q31.0 format.
 */
int32_t PID(PIDC_T* C, int32_t i32_Cmd, int32_t i32_Feedback)
{
    int32_t output;

    /* Calculate error */
    C->i32_Err = i32_Cmd - i32_Feedback;

    /* Integral calculation */
    C->i32_I += C->i32_Ki * C->i32_Err;

    /* PID calculation */
    output = C->i32_Kp * C->i32_Err + C->i32_I + C->i32_Kd * (C->i32_Err -
             C->i32_Last_Err);

    /* Set output in Q15.0 format */
    output = output >> 15;

    if(output > C->i32_Out_Limit)
        output = C->i32_Out_Limit;
    else if(output < -C->i32_Out_Limit)
        output = -C->i32_Out_Limit;

    /* Update last error */
    C->i32_Last_Err = C->i32_Err;

    /* Return the output of PID Controller */
    return output;
}
```

The definition of variables used in the function is defined in *PID_Example.h*.

```
#include "NuMicro.h"

/* The variable structure of PID controller */
typedef struct {
    int32_t i32_Err;                        /* Err = command-input */
    int32_t i32_I;                          /* I = Integral value */
    int32_t i32_Last_Err;                   /* Last_Err = Last error */
    int32_t i32_Kp, i32_Ki, i32_Kd;         /* Parameters of PID: Kp, Ki, Kd */
    int32_t i32_Out_Limit;                  /* Positive Limit of Output in Q15.0 format */
}PIDC_T;
```

The PID arithmetic function is executed in TIMER0 interrupt. The output of PID will approach to the target command after PID function is executed several times.

In general, the value of the variables is in Q15.0 format with the range from -32768 to 32767. User may modify the code to change the Q format.

3. Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - Example by M031_Series_BSP_CMSIS_V3.06.000
- IDE version
  - Keil uVersion 5.36

3.2 Hardware Requirements

- NuMaker-M032SE V1.3.
  - **Note:** The switches of TXD and RXD in Nu-Link2-Me must be set to ON.

4. Directory Information

The directory structure is shown below.

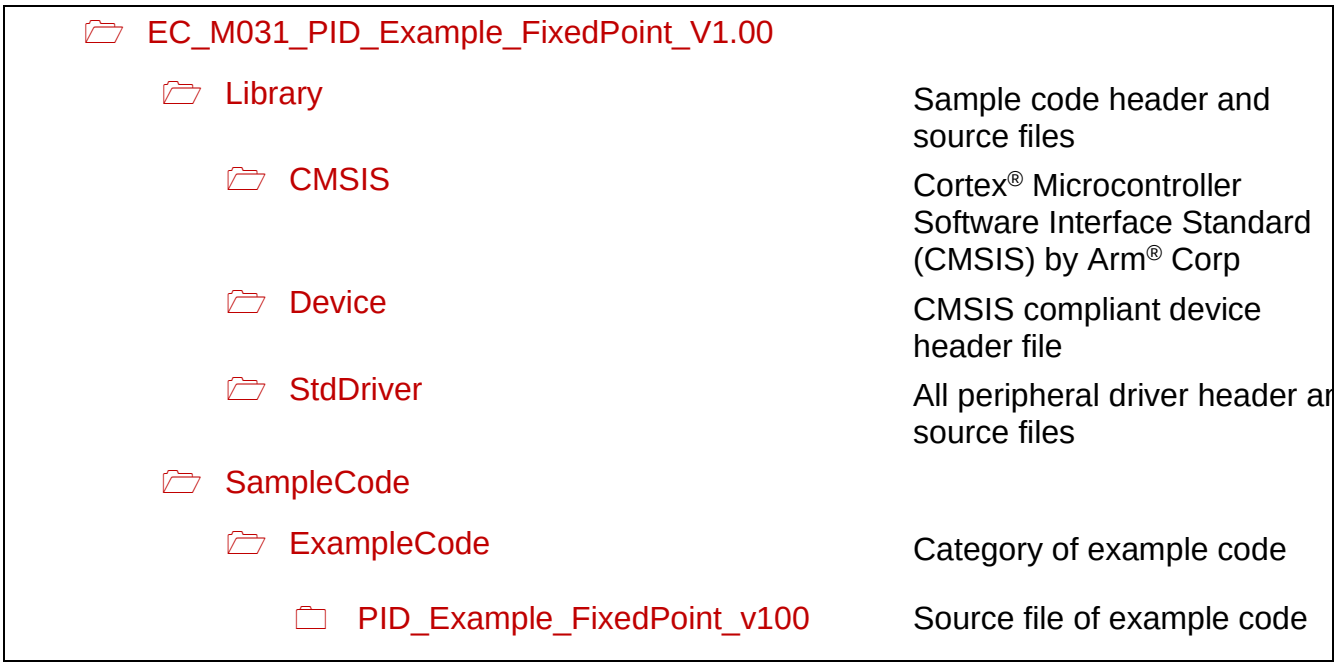

Figure 4-1 Directory Structure

5. Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *PID_Example_V1.uvprojx*.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Start the Terminal AP and set the COM port with the baud rate in 115200bps.
4. Enter debug mode.
   - Run

6. Revision History

Date	Revision	Description
2023.12.7	1.00	Initial version.

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.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.