DSP PID Controller

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The document shows the performance difference between CPU calling software PID function and DSP PID library.
BSP Version	M451 Series BSP CMSIS v3.01.002
Hardware	NuTiny-EVB-M451-LQFP100 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 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 Function Description

1.1 Introduction

Proportional--integral--derivative (PID) controllers have been the backbone of motion and process control for decades. The document shows the performance difference between CPU calling software PID function and DSP PID library.

The main system configuration of the M451 MCU is as follows:

- System clock frequency is 72 MHz
- Clock source is from PLL, and PLL source is from HXT
- Using Timer0's counter to compare the difference of performance

1.2 DSP PID Description

A Proportional Integral Derivative (PID) controller is a generic feedback control loop mechanism widely used in industrial control systems. A PID controller is the most commonly used type of feedback controller.

Algorithm:

y[n] = y[n-1] + A0 * x[n] + A1 * x[n-1] + A2 * x[n-2]

A0 = Kp + Ki + Kd

A1 = (-Kp) - (2 * Kd)

A2 = Kd

where Kp is proportional constant, Ki is Integral constant and Kd is Derivative constant

1.3 Demo Result

The application execution result as shown in Figure 1-1 Demo result. It took 508 and 1044 HXT clocks by calling DSP PID library and software PID function, respectively. One HXT time will be 83.33 ns.

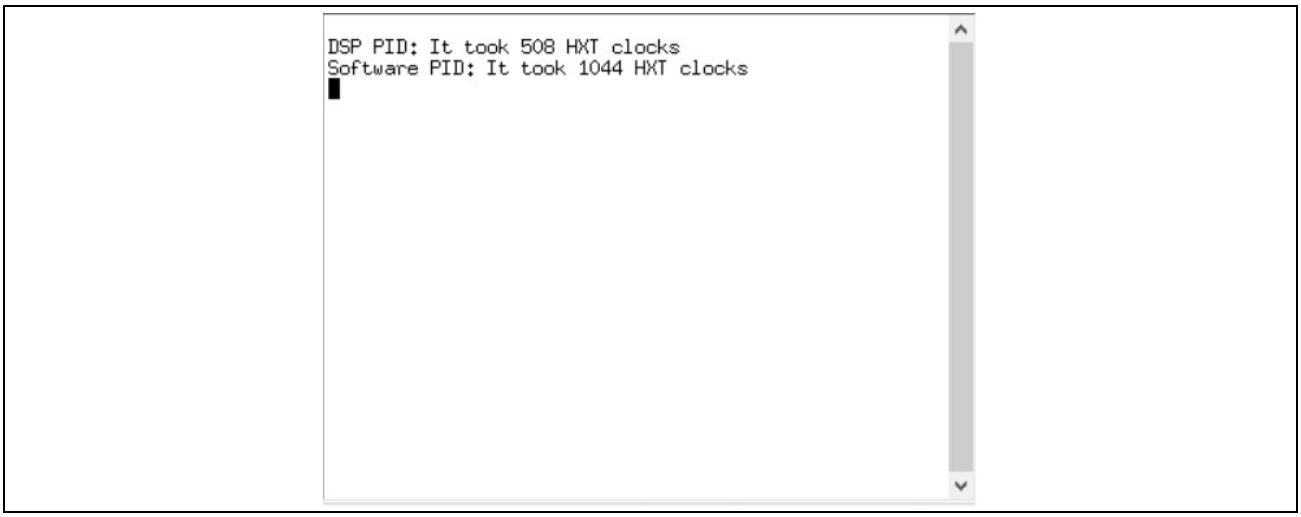

Figure 1-1 Demo result

2 Demo Code Description

Demo code project file located under folder \\SampleCode\StdDriver\DSP_PID_Controller\Keil\

Initialize system, UART and Timer Module in main.c.

```
PIDS.Kp = 0.4;
PIDS.Ki = 0.4;
PIDS.Kd = 0;
/* Target value*/
target = 500;
/* Inital value */
ival = 0;
/* Initial value and target value error */
ee = target - ival;

/* Unlock protected registers */
SYS_UnlockReg();

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

/* Lock protected registers */
SYS_LockReg();

/* Init UART for printf */
UART_Init();

/* Init TIMER0 for performance comparing */
TIMER_Open(TIMER0, TIMER_CONTINUOUS_MODE, 1);

TIMER_Start(TIMER0);
```

Following code initialize and calculate PID controller function 100 times by calling DSP library

```
/* Initial DSP PID controller function*/
arm_pid_init_f32(&PIDS,0);

/* Calculate PID controller function 100 times*/
```

```
    for(i = 1; i < 100; i++)
    {
        output[i] = arm_pid_f32(&PIDS,ee);
        ee = target-output[i-1];
    }
```

Following code initialize and calculate PID controller function 100 times by calling software PID function.

```
float PID(float in)
{
    float out;

    A0 = Kp + Ki;
    A1 = -Kp - (2 * Kd);
    A2 = Kd;

    /* y[n] = y[n-1] + A0 * x[n] + A1 * x[n-1] + A2 * x[n-2]  */
    out = (A0 * in) + (A1 * state[0]) + (A2 * state[1]) + (state[2]);

    /* Update state */
    state[1] = state[0];
    state[0] = in;
    state[2] = out;

    /* return to application */
    return (out);
}

int32_t main(void)
{
    …
    /* Calculate PID controller function 100 times*/
    for(i = 1; i < 100; i++)
    {
        output[i] = PID(ee);
        ee = target-output[i-1];
    }
    …
}
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - M451 Series BSP CMSIS v3.01.002
- IDE version
  - Keil uVersion 5.24

Hardware Environment

- Circuit component
  - NuTiny-EVB-M451-LQFP100 V1.3
- Diagram
  - Connect Nu-Link ICE to JP2 for downloading code and debugging

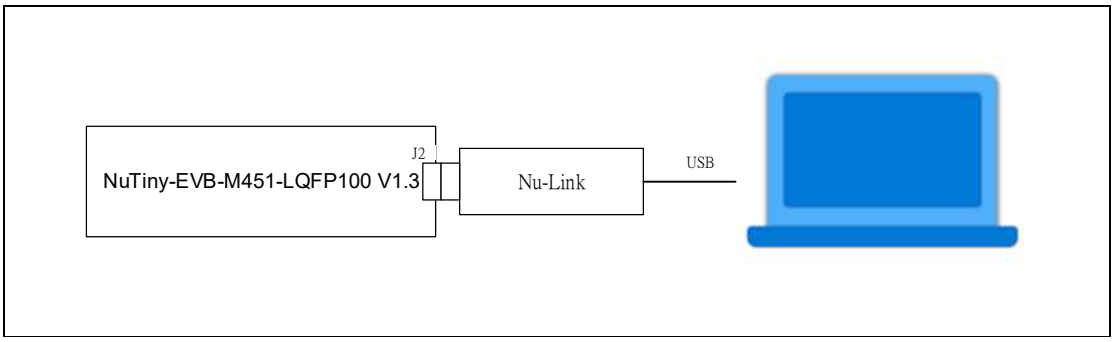

Figure 1-2 Hardware environment

4 Directory Information

EC_M451_DSP_PID_Controller_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	
StdDriver	Source file of example code
DSP_PID_Controller	
KEIL	Example files of KEIL project.

5 How to Execute Example Code

1. Opening the path \\SampleCode\StdDriver\DSP_PID_Controller\Keil\ folder by Directory Information (section 4) and double click DSP_PID_Controller.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
   d. Open Hyper-Terminal tool
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Oct 25, 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.

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.