MICROCHIP

TB3136

PID Control on PIC16F161X by using a PID Peripheral

Author: Daniel Hou
Microchip Technology Inc.

INTRODUCTION

Several products in the PIC® device family have many features that can be applied to motor control or power supplies, including the Proportional Integral Derivative (PID) control module. A PID control mode is implemented in this module to do a discrete-time PID or PI, PD, etc. control. This technical brief describes briefly how the module works, how to use this module in users' applications, and what should be observed.

BASIC PRINCIPLE

In control theory, a PID controller is an algorithm that uses the present error (proportional), the sum of the present and all previous errors (integral), and the difference between the present and previous change (derivative) to correct errors and provide stability in a system. It provides feedback to a system through a series of iterations, using the present error as well as previous errors to calculate a new input to the controller.

In a continuous-time system, the PID controller can be transferred from time domain to s-domain by using a Laplace Transform. The PID output equation is shown below.

$$\frac{C(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s$$

Next, a Finite Difference Method Z-transform is implemented on the continuous-time output equation.

The method is simply replacing 's' with $(1-z)^{-1}/T$, and T is the sampling period of the discrete-time system. The discrete-time PID output equation is shown below.

$$\frac{C_{o(z)}}{E(z)} = \frac{\left(K_p + K_i T + \frac{K_d}{T}\right) + \left(-K_p{}^{o} - \frac{2K_d}{T}\right)z^{-1} + \left(\frac{K_d}{T}\right)z^{-2}}{1 - z^{-1}}$$

Where:

$$K1 = K_p + K_i T + \frac{K_d}{T}$$

$$K2 = -K_p - \frac{2K_d}{T}$$

$$K3 = \frac{K_d}{T}$$

Then the output equation can be written as:

$$\frac{C_{o(z)}}{E(z)} = \frac{K1 + K2 \cdot z^{-1} + K3 \cdot z^{-2}}{1 - z^{-1}}$$

At last, as z^{-1} represents one delay in time, the output equation is transferred into a difference equation in time domain by replacing z^0 with [k], z^{-1} with [k-1], and z^{-2} with [k-2]:

$$C_0[k] = C_0[k-1] + {}^0K_1 e[k] + {}^0K_2 e[k-1] + {}^0K_{3(2)} e[k-2]$$

Where $c_0[k]$ is the current PID output, $c_0[k-1]$ is the previous PID output, and K1, K2 and K3 are the parameters of the current error, the previous error and the one before the previous error.

Based on the difference equation, the module is designed as shown in Figure 1.

IMPLEMENTATION

Inputs

To implement this module, three major inputs need to be calculated or determined, which are the set point value, the PID input value (feedback value), and the K1, K2 and K3 values.

The set point value is designed by users based on their own application, and it should be a 15-bit signed or a 16-bit unsigned integer. Users are free to design their set point, such as a CCP value of a PWM waveform, or an ADC value of the desired voltage. The set point value is written into the PIDxSET register.

The PID input value, also known as the feedback, should be measured by users on the process they want to control (motor, power supply, heater, etc.). The measured value is stored in the PIDxIN register and should be the exact same type as the set point value. For instance, if the set point value is a 16-bit ADC value, the PID input should also be a 16-bit ADC value that users measure from their application.

To calculate K1, K2 and K3, two things need to be designed by the user. One is the K_p, K_i and K_d in the users control systems. Another is the sampling period T of the system. Once these are designed, the K1, K2 and K3 can be calculated. Since the sampling period is taken into account, the calculated results of K1, K2 and K3 are usually not integers. The user needs to scale them up to three 15-bit signed integers, and then store them in the PIDxK1, PIDxK2 and PIDxK3 registers, respectively. Please note that the scale factors for K1, K2 and K3 should be the same, and the PID output value should be scaled back down by the same factor in order to be used as the real output of the PID controller. All the scaling is done in software by the user.

Overall, in order to set up the module, the user needs to set the PIDxK1, PIDxK2, PIDxK3 and PIDxSET registers as inputs. The module is able to do the PID calculation by itself.

Operating

After setting up the set point value and the K1, K2 and K3 values, the PID module is ready to operate. Please note that the calculation is started when PIDxINL (the lower byte of PIDxIN) is written, so any changes to any register has to be made before the low value of the PIDxIN register is updated.

A BUSY flag of the PIDxCON register is set to '1' when the module is busy calculating, and the flag will return to '0' when the module is done calculating. A PIDxDIF interrupt flag is also triggered after one successful calculation. The user can read the value in PIDxOUT after the BUSY flag returns to '0' or the PIDxDIF interrupt is triggered, and scale the value down by the same factor used when scaling up the K values to get the real PID output of this iteration.

Sampling Period

Since it's a discrete-time system, a sampling rate should be determined. The sampling period of the system should be the time from when one PID input is loaded until the next PID input is loaded. Please note that the sampling period is not the time of one PID calculation, since users can implement their own algorithms or programs in one iteration and the PID module calculation could only take up a small amount of time. Users can choose the sampling period according to their needs, but within the limitation of the module and the system (accumulator size, K1, K2, K3 registers' size, etc.).

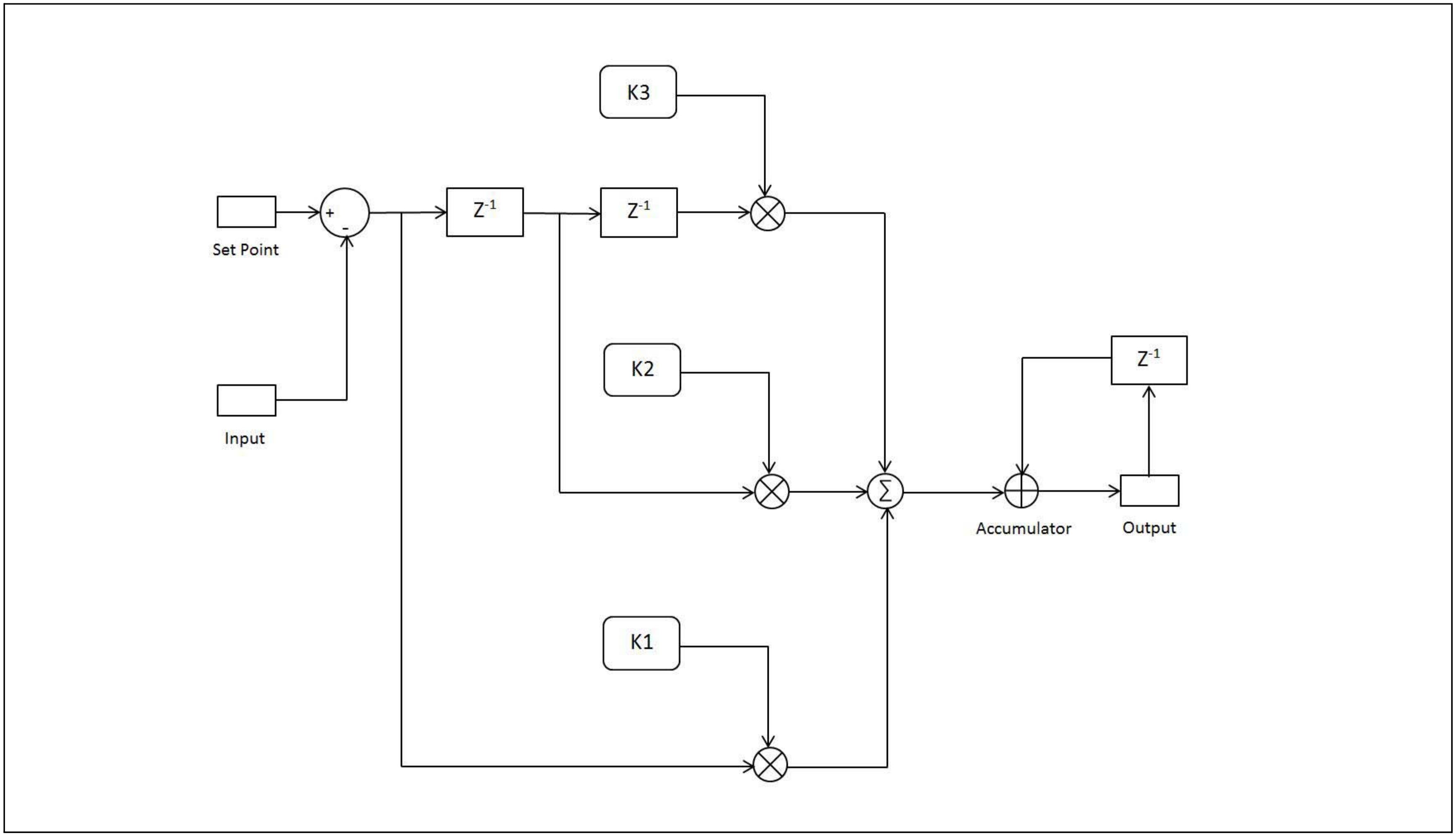
Accumulator

A 35-bit accumulator is used to accumulate the PID output. Please note that the accumulator may overflow when the error of the PID controller (set point, PID input) is too big or accumulates too much. There are certain circumstances that the accumulator may overflow easily. One is choosing a rather fast sampling rate, which makes the error add up quickly in a short time. One is integral windup, where there is a rapid big change of the set point, which results in a large error. A system with a long response time could also cause the accumulator to overflow since the error does not reduce fast with a long response time.

CONCLUSION

The PID module can be used in many control systems applications. Since the module does the multiplication and accumulation for the user, it saves users a lot of time and allows for implementation of more operations based on their own applications. It takes only nine instruction cycles for the module to do a PID calculation, while it takes around 1000 instruction cycles by using software to do the same calculation. Therefore, there is a huge efficiency improvement using the PID module.

FIGURE 1: PID MODULE BASIC DATA FLOW BLOCK DIAGRAM, PID MODULE

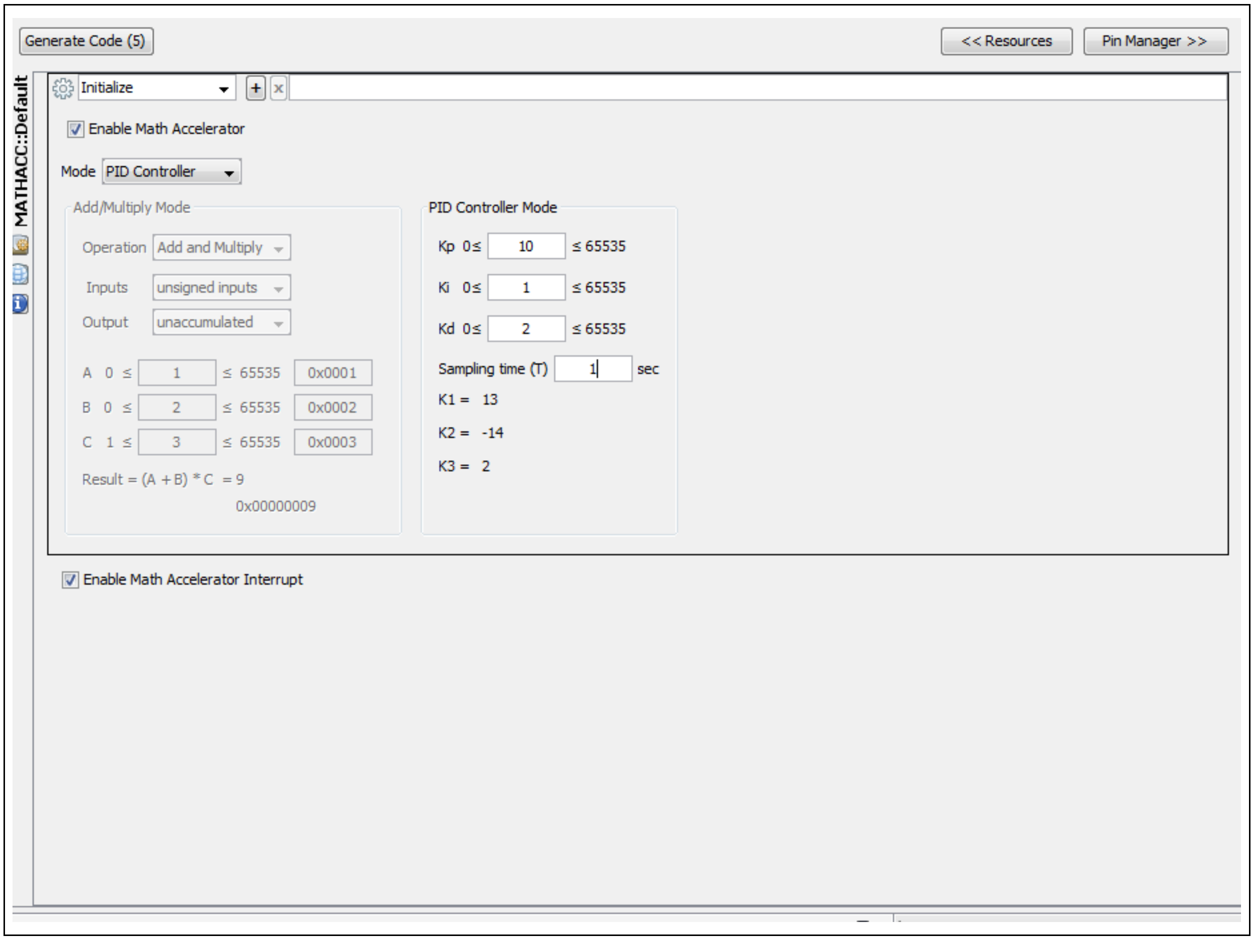


APPENDIX A: CODE FOR IMPLEMENTING THE MATH ACCELERATOR


```
/**
 * Setup the PID mosdule to do a PID calculation.
 */
//////////////////////////////////////////////////////////////////////
PID1ACCU = 0;                   // Clear the Accumulator (PID1ACCU)
PID1ACCHH = 0;                  // Clear the Accumulator (PID1ACCHH)
PID1ACCHL = 0;                  // Clear the Accumulator (PID1ACCHL)
PID1ACCLH = 0;                  // Clear the Accumulator (PID1ACCLH)
PID1ACCLL = 0;                  // Clear the Accumulator (PID1ACCLL)
PID1CON1bits.MODE = 0b101;      // PID mode
PID1CON1bits.EN = 1;            // PID module is enabled (PID1CON1)
//////////////////////////////////////////////////////////////////////
/**
 * Calculate K1, K2, K3 based on Kp, Ki, Kd and the sampling rate
 */
//////////////////////////////////////////////////////////////////////
kp = 912;
ki = 9000;
kd = 41;
// Sampled at 256 Hz. Sampling period is 3.9 ms
K1 = kp + (ki >> 8) + (kd << 8);// K1 = Kp + Ki*T + Kd/T
K2 = -kp - (2 * (kd << 8));// K2 = -Kp – 2*Kd/T
K3 = (kd << 8);// K3 = Kd/T
//////////////////////////////////////////////////////////////////////
/**
* Load in the values
*/
//////////////////////////////////////////////////////////////////////
PID1K1 = K1;
PID1K2 = K2;
PID1K3 = K3;

PID1SET = 100;
PID1INH = 0; //
PID1INL = 0; // This starts the math execution (PIDxINL)
//////////////////////////////////////////////////////////////////////
while(PID1CONbits.BUSY); // Wait for the module to complete
result = PID1OUT; // Load the result
```

APPENDIX B: SCREEN SHOT OF HOW TO INITIALIZE THE PID MODULE BY USING MPLAB CODE CONFIGURATOR



NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-63277-497-2

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

01/27/15