PmodHB5 Demo Reference Manual

Revision: February 14, 2012

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Introduction

The Digilent PmodHB5 2A H-Bridge Module (the HB5) is an ideal solution for robotics and other applications where logic signals are used to drive small to medium-sized DC motors, such as the Digilent motor-gearbox. The HB5 works with power supply voltages from 2.5V to 5V, but is commonly operated at 3.3V because that is the supply voltage on most Digilent system boards. This document describes the application that demonstrates the use of HB_MotorLib library functions. It also demonstrates how to use CLPLib library functions.

Note: Project was created using MPLAB v8.83. To use this demo with the HB3 and have the proper RPM calculation, the GND and VCC connections on the J5 header are actually switched. i.e. GND is connected to VCC of the motor and VCC is connected to GND of motor.

Overview

Using the two buttons on Cerebot32Mx4 board, the user controls the duty applied to a motor and the direction. The application reads motor data and outputs the RPM on the CLP. The main loop contains a loop processing button actions and the display. Buttons are read and debounced at a rate of 100us using timer 5. Using buttons, duty is raised and lowered (single button), or direction is changed (double buttons). Pressing BTN2 will raise the duty cycle 1% while pressing BTN 1 will decrease the duty cycle by 1%. If the user releases one button while the other is pressed, known as double buttons action, the direction of the motor will change, but the duty cycle magnitude is maintained.

Note: The demo assumes that PmodCLP J1 is plugged into JA and J2 is plugged into JB, also, the PmodHB5 is to be plugged into JH on the lower set of pins.

Library Operation

Library Interface

HB_MotorLib_Defs.h was edited with the following revisions:

- Define the use of Pmod connector JH 7-12
  - #define _MOTOR1
- Define the fact that the real time clock is incremented from application ISR
  - #define REALTIME_CALLEDBYUSER
- Define the real time factor
  - REAL_TIME_FACTOR 10000 // (= 1s/100 us)

The following library functions from HB_MotorLib_Defs.h were used:

Function	Where (function / file)	Purpose
HB_MotorConfigure1	main/main.c	Configure HB_Motor
HB_MotorCommand1	main/main.c	Command the motor with the desired duty and direction.
HB_MotorReaction1	Display/main.c	Compute and read the motor speed.
mIncrementRealTime	Main/Timer5Handler	Macro used to increment the value of the real time counter

Because PmodCLP port J1 is plugged into JA and J2 is plugged into JB, no changes were required in ClpLib_Config.h. If PmodCLP needs to be connected to another connector, then changes should be made.

The following library functions from ClpLib.h were used:

Function	Where (function / file)	Purpose
ClpPinsConfigure	DeviceInit/main.c	Configure CLP pins.
ClpInit	main/main.c	Initialize CLP.
ClpSetBackLight	main/main.c	Turn backlight ON.
ClpWriteStringAtPos	Display/main.c	Write text to CLP.

For more information on the HB_MotorLib.h library usage, see the PmodHB5 Library Reference Manual. Also, if the user is interested in more information on the PmodCLP library operation, see the PmodCLP Library Reference Manual found under the PmodCLP webpage at www.digilentinc.com .

Demo Documentation

Project Files

File	Containing
main.c	Main application file. Contains main application loop, interface ISR, buttons processing.
util.h	Common Utility Procedures. This is a standard Digilent file, also used in other applications. It was not modified for this application.
util.c	Common Utility Procedures. This is a standard Digilent file, also used in other applications, implementing some led and wait functions. It was not modified for this application.
stdtypes.h	Digilent Standard Type Declarations. This is a standard Digilent file, also used in other applications. It was not modified for this application.

Resources Used

Timer5 configuration is done using Plib macros in AppInit function. SFRs approach is also provided in the commented code.
Plib macro used:
OpenTimer5(T5_ON | T5_SOURCE_INT | T5_PS_1_8, 99);
ConfigIntTimer5(T5_INT_ON | T5_INT_PRIOR_7 | T5_INT_SUB_PRIOR_3);
Meaning:
- T5_ON – timer 5 is ON.
- T5_PS_1_8 (Prescaler 1/8): so the frequency is 1/8 of Peripheral bus freq, which is 1/8 of SYSCLK (65 MHz) = 1 MHz
- Period = 99, so Timer period = (99 + 1) * 1/1MHz = 100 us
- T5_SOURCE_INT : Timer5 triggers interrupt
- T5_INT_ON – T5 interrupt is ON
- T5_INT_PRIOR_7, T5_INT_SUB_PRIOR_3 - timer interrupt priority level 7, subpriority level 3

More information about Timer5 is shown in Timer5Handler.

Functions defined in main.c

void __ISR(_TIMER_5_VECTOR, ipl7) Timer5Handler(void)

Parameters:
none

Interrupt service routine for Timer 5 interrupt. Programmed to execute every 100 us. It is used to:

1. perform software debouncing of the on-board buttons and detect:
   - single button situation: when a button should be processed (being pressed or being kept pressed)
   - double buttons situation: when a button is released while the other is pressed
2. to increment the real time counter
3. as a time base for clearing the action LEDs (LD3 and LD2)
4. as a time base for updating the display

int main(void)

Parameters:
none

Performs basic board initialization and then enters the main program loop.

void Display(double dDuty)

Parameters:
dDuty - the applied duty (between -1 and 1), negative values correspond to the negative direction of motion, while positive numbers correspond to the positive direction of motion.

The function reads the DCMotorReactionContext structure containing motor information. It then displays the applied duty and the RPM speed.

void DeviceInit(void)

Parameters:
none

This routine initializes the on-chip and on-board peripheral devices to their default state. It also calls ClpPinsConfigure to configure the CLP pins.

void AppInit(void)

Parameters:
none

This routine performs application specific initialization. It configures devices and global variables for the application.