PmodJSTK Programmer's Reference Manual

Revision: December 15, 2011

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

Introduction

This document describes the programming interface to the PmodJSTK library that is included as part of the PmodLib library. It describes the capabilities of the PmodJSTK library and all the API functions used to access its features.

The purpose this library is to offer supporting functions for interfacing with the PmodJSTK hardware on either the Cerebot 32MX4 or 32MX7 microcontroller. For additional information on operation and signaling see the Digilent PmodJTSK Reference Manual.

PmodJSTK Basic API Functions

void PmodJSTKInit(SpiChannel chn,uint32_t pbClock,uint32_t bitRate,uint32_t systemClock)

Parameters
SpiChannel chn - spi channel
uint32_t pbClock - peripheral bus clock in Hz
uint32_t bitRate - bit rate desired in Hz

Returns
None

This function opens the desired SPI channel in 8-bit mode as a master, enables the slave select bit, and sets the desired bit rate as a function of pbClock/bitRate. Maximum recommended maximum bit rate is 1Mhz.

void PmodJSTKSendRecv(SpiChannel chn,uint8_t cmdIn,
PmodJSTKAxisButton *jystkAxisButtons)

Parameters
SpiChannel chn - spi channel
uint8_t cmdIn - button LED command
PmodJSTKAxisButton *jystkAxisButtons - pointer to struct containing data returned by the joystick.

Returns: none

This function will operate on the channel specified in parameter chn. Parameter cmdIn will accept the following values and have the following effect on the PmodJSTK:
PMODJSTK_LD1_LD2_OFF: all LEDs off
PMODJSTK_LD1_ON: LD1 on
PMODJSTK_LD2_ON: LD2 on
PMODJSTK_LD1_LD2_ON: LD1 and LD2 on

Parameter *jystkAxisButtons is a pointer to a PmodJSTKAxisButton struct which will be filled as specified below:

xAxis: 16 bit field holding the joystick position on the x axis (between 0 and 1023)
yAxis: 16 bit field holding the joystick position on the y axis (between 0 and 1023)
buttonState: 8 bit field holding the state of the buttons, use the following bit masks to determine button status: PMODJSTK_BTN1, PMODJSTK_BTN2, PMODJSTK_BTN_JSTK, PMODJSTK_BTN_NONE

void PmodJSTKDelay10us()

Parameters
None

Returns
None

This function creates a 10 microsecond delay by blocking for the number of cycles calculated in PmodJSTKInit.

void PmodJSTKDelay15us()

Parameters
None

Returns
None

This function creates a 15 microsecond delay by blocking for the number of cycles calculated in PmodJSTKInit.

PmodJSTK Additional Information

PmodJSTKAxisButton is a structure representing values received from PmodJSTK. Both the Y-axis and X-axis are captured in uint16_t variables. The current state of the button is captured in a uint8_t variable. Shown below is C code implementing the structure:

```
typedef struct
{
	uint16_t xAxis; //X axis position
	uint16_t yAxis; //Y axis position
	uint8_t buttonState; //current state of buttons
}PmodJSTKAxisButton;
```

Table 1: Macro Definitions		
Macro	Value	Description
PMODJSTK_LD1_LD2_OFF	0x80	Turn LD1 and LD2 Off
PMODJSTK_LD1_ON	0x81	Turn LD1 ON only
PMODJSTK_LD2_ON	0x82	Turn LD2 ON only
PMODJSTK_LD1_LD2_ON	0x83	Turn LD1 and LD2 on
PMODJSTK_BTN_NONE	0x0	No Joystick Buttons Pressed
PMODJSTK_BTN_JSTK	0x1	Joystick Stick Button Pressed
PMODJSTK_BTN1	0x2	Joystick BTN1 Pressed
PMODJSTK_BTN2	0x4	Joystick BTN2 Pressed
PMODJSTK_MAX_X_Y_AXIS	1023	Maximum deflection value of X and Y axes
PMODJSTK_MIN_X_Y_AXIS	0	Minimum deflection value of X and Y axes
PMODJSTK_BYTES_PER_XFER	5	Number of bytes read in from PmodJSTK for one poll.

PmodJSTK Example Use:

The following example illuminates the joystick LEDs based on the joystick button pressed, and board LEDs LD1 - LD4 based on the deflection of the joystick, written for the Cerebot32MX4.

```
/* ------------------------------------------------------------ */
/*              PIC32 Configuration Settings                    */
/* ------------------------------------------------------------ */
#pragma config FPLLMUL = MUL_20, FPLLIDIV = DIV_2, FPLLODIV = DIV_1
#pragma config FWDTEN = OFF
#pragma config POSCMOD = HS, FNOSC = PRIPLL
#pragma config FPBDIV = DIV_2

#include <stdint.h>
#include <plib.h>
#include <pmodlib.h>

#define SYSTEM_CLOCK (80000000L)            //System clock speed (8 MHz Crystal/ FPLLIDIV * FPLLMUL / FPLLODIV)
#define PB_CLOCK (SYSTEM_CLOCK/2)           //Peripheral bus clock
#define SPI_BITRATE 625000                  //Bit rate for SPI port

int main(void)
{
        uint8_t ledState = PMODJSTK_LD1_LD2_OFF;

        PmodJSTKAxisButton jstkAxisButtons;
        //Set digital IO for LD1 - LD4 on Cerebot32MX4
        PORTSetPinsDigitalOut(IOPORT_B,BIT_10|BIT_11|BIT_12|BIT_13);
        //Init Joystick on SPI Channel 2
        PmodJSTKInit(SPI_CHANNEL2,PB_CLOCK,SPI_BITRATE,SYSTEM_CLOCK);

        while(1)
        {
                //Poll joystick
                PmodJSTKSendRecv(SPI_CHANNEL2,ledState,&jstkAxisButtons);
                //Set Joystick LED State
                switch(jstkAxisButtons.buttonState)
                {
                        case PMODJSTK_BTN1:
                                ledState = PMODJSTK_LD1_ON;
                                break;
                        case PMODJSTK_BTN2:
                                ledState = PMODJSTK_LD2_ON;
                                break;
                        case PMODJSTK_BTN_JSTK:
                                ledState = PMODJSTK_LD1_LD2_OFF;
                                break;
                }

                //Joystick in X+ turn on LD1
                if(jstkAxisButtons.xAxis > (PMODJSTK_MAX_X_Y_AXIS/2) + 200)
                {
                        PORTClearBits(IOPORT_B,BIT_11);
                        PORTSetBits(IOPORT_B,BIT_10);
                }
                //Joystick in X- turn on LD2
                else if(jstkAxisButtons.xAxis < (PMODJSTK_MAX_X_Y_AXIS/2) - 200)
                {
                        PORTSetBits(IOPORT_B,BIT_11);
                        PORTClearBits(IOPORT_B,BIT_10);
                }
```

```
                //Turn off LD1 and LD2
                else
                {
                        PORTClearBits(IOPORT_B,BIT_10|BIT_11);
                }
                //Joystick in Y+ turn on LD3
                if(jstkAxisButtons.yAxis > (PMODJSTK_MAX_X_Y_AXIS/2) + 200)
                {
                        PORTClearBits(IOPORT_B,BIT_13);
                        PORTSetBits(IOPORT_B,BIT_12);
                }
                //Joystick in Y- turn on LD4
                else if(jstkAxisButtons.yAxis < (PMODJSTK_MAX_X_Y_AXIS/2) - 200)
                {
                        PORTSetBits(IOPORT_B,BIT_13);
                        PORTClearBits(IOPORT_B,BIT_12);
                }
                //Turn off LD3 and LD4
                else
                {
                        PORTClearBits(IOPORT_B,BIT_12|BIT_13);
                }
        }
}
```