HCTL-2032 Conversion to HCTL-2016 and HCTL-2020

Quadrature Decoder/Counter ICs

AVAGO TECHNOLOGIES

Application Note 5110

Introduction

The application note explained the method of converting the new HCTL-2032 Quadrature Decoder/Counter IC to match the functionality of the HCTL-2016 and the HCTL-2020 Quadrature Decoder/Counter ICs. Table 1 illustrates the features comparison between HCTL-2032 and HCTL-2016/2020.

Conversion of HCTL-2032 to HCTL-2016 / HCTL-2020

Based on table 1, the HCTL-2016/2020 maximum counter size is required to limit to 16 bits only instead of 32 bits. The 32-bit output data is selectable via SEL1 (pin 6) and SEL2 (pin 26). Table 2 illustrates SEL1 and SEL2 configuration for bytes selection.

Table 1. HCTL ICs Features Comparison

Features Description	HCTL-2032	HCTL-2016	HCTL-2020
Operating Voltage	5.0 Volts	5.0 Volts	5.0 Volts
Maximum Operating Frequency	33 MHz	14 MHz	14 MHz
Operating Temperature Range	-40°C - 100°C	-40°C - 85°C	-40°C - 85°C
Maximum Counter Size	32 Bits	16 Bits	16 Bits
Count Modes	1X, 2X, or 4X	4X	4X
Number of Axis Support	2	1	1
Index Channel Support	Yes	No	No
Up/Down Output Indicator	Yes	No	Yes
Quadrature Pulse Indicator	Yes	No	Yes
Cascading Support	Yes	No	Yes
Package Type	PDIP-32 / SOIC-32	PDIP-16 / PLCC-20	PDIP-20 / PLCC-20

Table 2. Bytes Selection Configuration

		BYTE SELECTED	Col 4	Col 5	Col 6
SEL1	SEL2	MSB	2ND	3RD	LSB
0	1	D4			
1	1		D3		
0	0			D2	
1	0				D1

Table 2 can also be found on page 5 of the HCTL-2032 datasheet.

Since the HCTL-2016/2020 is a 16-bit Counter, SEL1 and SEL2 can be configured to select the 3rd byte and LSB only. Based on table 2, in order to configure the HCTL-2032 as 16-bit Counter, the SEL2 can be tied to Ground permanently. The SEL1 can be either 1 or 0, selectable via user control software, to select low byte or high byte.

The second parameter that requires the user attention is the Count Modes selection features available in the HCTL-2032 only. The count modes are selectable, 1X, 2X, or 4X count mode. The count modes are configured via EN1 (pin 2) and EN2 (pin 3) on the HCTL-2032. In order to set the maximum counter size to 16-bit and 4X-encoding mode, the illegal mode is entered with EN1 and EN2 both are set to logic low. When this mode is entered, the user will get a true 16-bit counter with 4X encoding.

Table 3 illustrates the count modes selection.

Table 3. Encoding Mode Selection

EN1	EN2	4X	2X	1X
0	0	Illegal Mode		
1	0	On		
0	1		On	
1	1			On

Figure 1 illustrates the connection for the conversion of HCTL-2032 to HCTL-2016 and HCTL-2020.

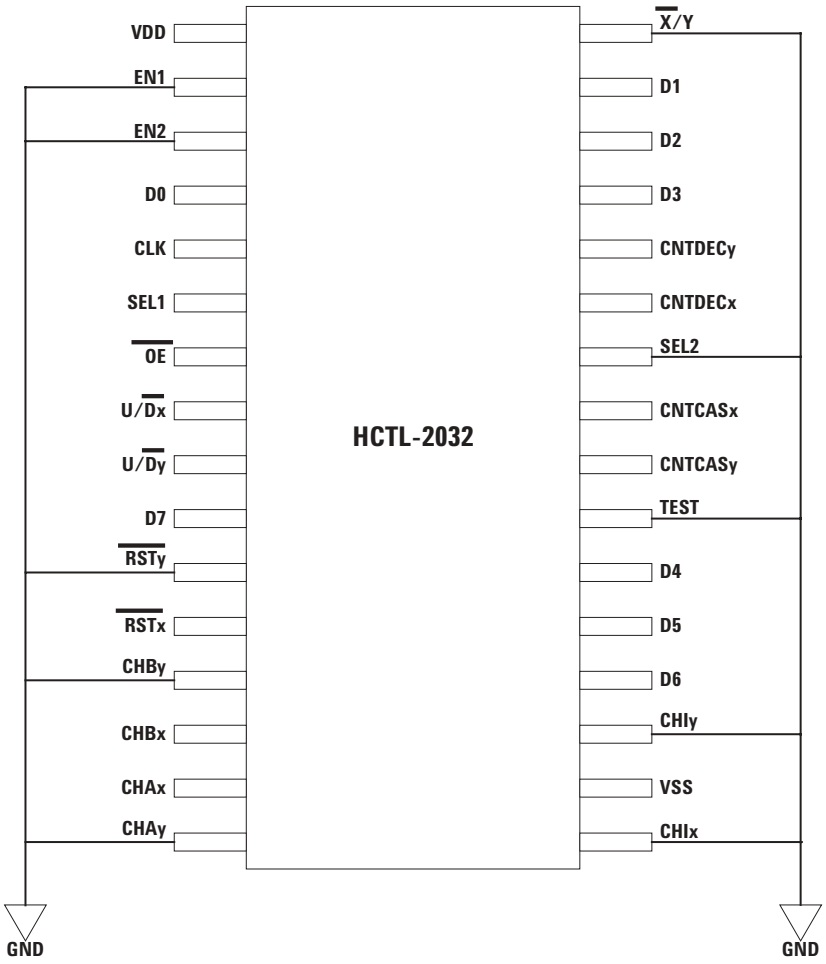

Figure 1. Conversion HCTL-2032 to HCTL-2016/2020

Example software to communicate with the HCTL-2032 to emulate HCTL-2016/2020 funtionality.

Listing

```
//*********************************************************************
//*        Title:     HCTL-2032 Conversion to HCTL-2016/2020           *
//*        Author:    Teng Kong Leong, Senior Application Engineer      *
//*                                                                    *
//*********************************************************************

#include <p18f252.h>                    // Include PIC18F252 definition
#include <delays.h>                     // Include DELAY library
#include <stdlib.h>                     // Include STANDARD library
#include "teng_lcd.h"                   // Include LCD library

//*** Configuration Bits ***
#pragma config OSCS = OFF, OSC = HS
#pragma config PWRT = ON, BOR = ON, BORV = 42
#pragma config WDT = OFF
#pragma config CCP2MUX = OFF
#pragma config STVR = ON, LVP = OFF, DEBUG = OFF
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF
#pragma config CPB = OFF, CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF
#pragma config WRTC = OFF, WRTB = OFF, WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF
#pragma config EBTRB = OFF
//**************************

//*** Function Prototype ***
void fetch_32(void);
//**************************

//*** Port Alias ***
#define POWER       LATAbits.LATA0    // Power LED indicator
#define SEL1        LATAbits.LATA1    // SEL1 pin
#define SEL2        LATAbits.LATA2    // SEL2 pin
#define OE          LATAbits.LATA3    // OE pin
#define RSTx        LATAbits.LATA5    // Reset pin
#define DATA        PORTC             // Output Data
//******************

//*** Variables Declaration ***
unsigned long       BYTE_1;                 // LSB
unsigned long       BYTE_2;
unsigned long       COUNT_OLD;              // Previous Total Count
unsigned long       COUNT_NEW;              // Current Total Count
char                COUNT_ASCII[10];        // Total Count in ASCII
//*****************************

void fetch_32(void)
{
        /*      Fetch 16-bit Data       */
                SEL1    = 0;                                    // Select 3rd byte
                SEL2    = 0;
                OE      = 0;                                    // Enable OE
                BYTE_2  = DATA;                                 // Fetch the 3rd byte

                SEL1    = 1;                                    // Select LSB
                SEL2    = 0;
                BYTE_1  = DATA;                                 // Fetch the LSB
                OE      = 1;                                    // Dis-able OE
}
```

```
void main(void)
{
        /*      Port Initialization     */
        PORTA   = 0x00;                                 // Initialize Port A
        LATA    = 0x00;                                 // Clear Port A latches
        ADCON1  = 0x07;                                 // Set Port A as digital I/O
        TRISA   = 0x00;                                 // All Output
        POWER   = 1;                                    // Turn On LED
        OE      = 1;                                    // Dis-abled OE
        RSTx    = 1;                                    // Dis-abled RESET

        PORTB   = 0x00;                                 // Initialize Port B
        LATB    = 0x00;                                 // Clear Port B latches
        TRISB   = 0x00;                                 // All Output

        PORTC   = 0x00;                                 // Initialize Port C
        LATC    = 0x00;                                 // Clear Port C latches
        TRISC   = 0xFF;                                 // All Input


        /* Initialize LCD */
        OpenLCD( );
        Delay10KTCYx(25);                               // Delay 100 mSec for LCD Initialization

        LCD_Set_Cursor( 0, 0 );                         // Column 0, Line 1
        putrsLCD("HCTL-2032 Count");
        LCD_Set_Cursor( 0, 1 );                         // Column 0, Line 2
        putrsLCD("0");

        /*      Initialize Variables    */
        BYTE_1  = 0;                                    // Clear variables
        BYTE_2  = 0;
        COUNT_NEW       = 0;
        COUNT_OLD       = 0;
        RSTx    = 0;                                    // Reset pulse
        Delay10TCYx(1);                                 // Delay for 10 cycles
        RSTx    = 1;

        /*      Main Loop*/
        while (1)
        {
                fetch_32();                             // Fetch 32-bit data

                /*      Display Data On Display         */
                COUNT_NEW       = (BYTE_2*0x100)+(BYTE_1);// Current Total Count
                if ( COUNT_NEW != COUNT_OLD )
                        {
                                ultoa( COUNT_NEW, COUNT_ASCII );
                                LCD_Set_Cursor( 0, 1 );                         // Column 0, Line 2
                                putrsLCD("       ");
                                LCD_Set_Cursor( 0, 1 );
                                putsLCD( COUNT_ASCII );
                                COUNT_OLD       = COUNT_NEW;    // Current Count became Previous Count
                        }       //*** End If-else Statement ***

        }       //*** End While-Loop Statement ***

}       //*** End Main Program ***
```

NOTE:
Microchip PIC18F252 operating at 10MHz frequency is used to communicate with HCTL-2032 operating at 16MHz.

5989-2699EN - May 29, 2006