genapta

AN001

Interfacing microprocessors with genapta's Quadrature Decoder/ Counter/ Shift Register IC range

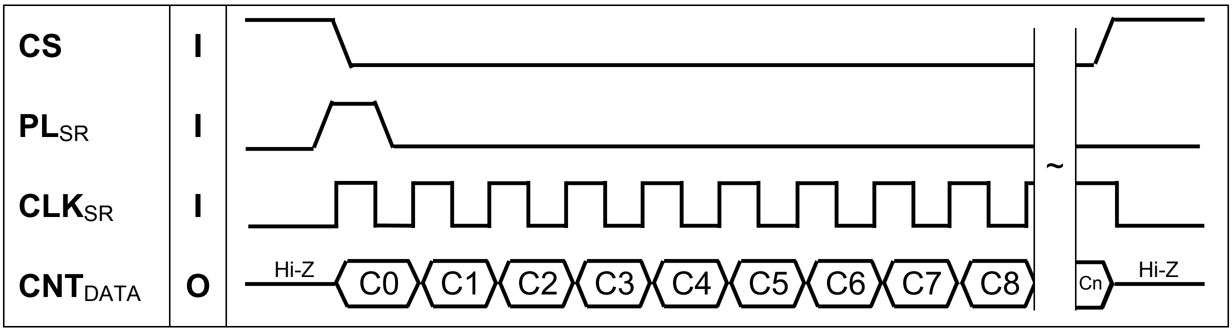
V1.0-081204

This application note describes interfacing with genapta's range of quadrature decoder and counter ICs. A sample circuit and C code for interfacing with Microchip's PIC microcontroller are also given.

In the discussion here, the GEN-2122-5 chip is used for illustration. The GEN-1130-5 differs only in its input connections. The GEN-2212-5 also differs little - it has two encoder inputs, and an extra input to select which channel is read. This can be connected to an additional PIC pin, or to another multiplexer.

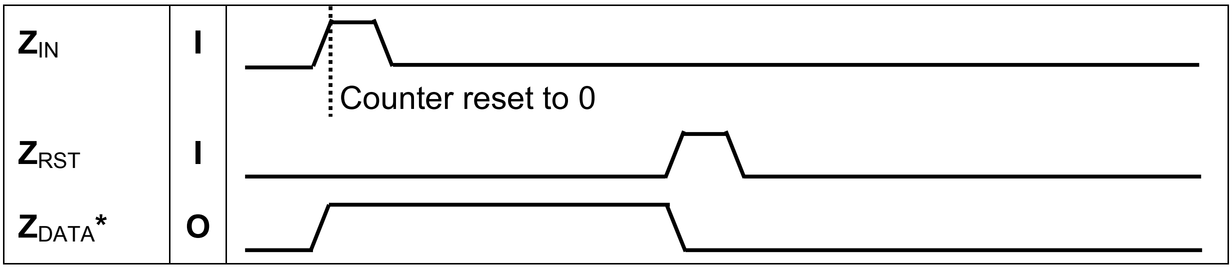
Interfacing with a PIC microcontroller is discussed, however the principles can equally be applied to any other low-cost, low-pin-count microcontrollers.

Operating Sequence : Shift Register



C0 is the LSB of the count value, increasing to Cn (n - number of counter bits).

Operating Sequence : Zero



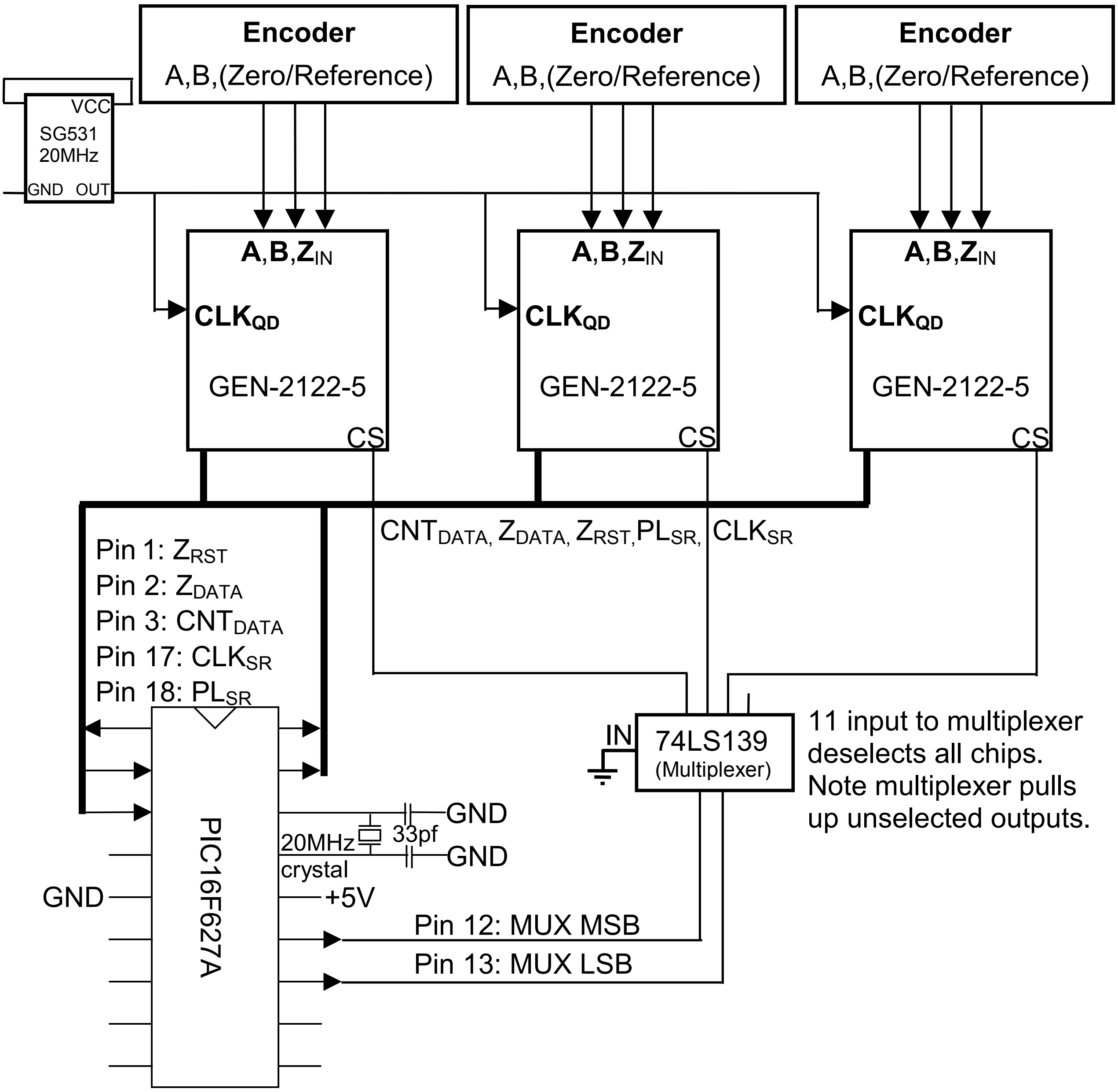
*Note: **Z**$_{\text{DATA}}$ will be Hi-Z (tristate) unless CS is low. Pull CS low to read the data.

Important Notice

genapta decoders are not recommended for use in safety critical applications. eg: Life support systems, critical care medical equipment, ABS braking systems and power steering. Please contact us for clarification if required.

Possible Interfacing Circuit

The circuit depicted here shows a three channel position detection system. Three position encoders are connected, one to each of three GEN-2122-5. More channels can be added as necessary. If only one chip (single or dual channel) is used, then the multiplexer can be removed and the CS line tied to +5V.



11 input to multiplexer deselects all chips. Note multiplexer pulls up unselected outputs.

PIC Code Excerpt

The below C code example can be compiled with HI-TECH Software's free PICC Lite compiler (downloadable from http://www.htsoft.com)

This code is a complete program, although does nothing with the retrieved counts. The pins defined match up with those pins used in the circuit on the preceding page.

Main.h

```
#ifndef _MAIN_H_
#define _MAIN_H_

#define portbit(adr, bit) ((unsigned)(&adr)*8+(bit))
#define bitset(var,bitno) ((var) |= 1 << (bitno))
#define bitclr(var,bitno) ((var) &= ~(1 << (bitno)))
#define bittog(var,bitno) ((var) ^= 1 << (bitno))

//Set multiplexer output
#define SetChannel(channel) (PORTB = (PORTB & 0b00111111) | (channel << 6))

#define NOP() asm("nop")
#define DELAY_IT() { NOP(); NOP(); NOP();}

//Bits in PORTA
#define CLKSR           0     //pin 17
#define PLSR            1     //pin 18
#define ZERORESET       2     //pin 1
#define ZERODATAPIN     3     //pin 2
#define CNTDATAPIN      4     //pin 3

//Bits in PORTB
#define MUX_LSB         6     //pin 12
#define MUX_MSB         7     //pin 13

unsigned int GetLSB(unsigned long num);
unsigned int GetMSB(unsigned long num);
unsigned long GetCount(void);
unsigned short GetZero(void);

#endif
```

Main.c

```
#include <pic.h>
#include <stdio.h>
#include "main.h"

static bit CNTDATA @ ((unsigned)(&PORTA)*8+(CNTDATAPIN));
static bit ZERODATA @ ((unsigned)(&PORTA)*8+(ZERODATAPIN));
unsigned int GetLSB(unsigned long num) { return (unsigned int) num; }
unsigned int GetMSB(unsigned long num) { return (unsigned int)(num>>16); }

void main(void)
{
    unsigned long count;
    unsigned int lsb;
    unsigned int msb;
    unsigned short i;
    STATUS=0;
    TRISA = 0b00011000;  //A3,A4 inputs
    PORTA = 0b00000000;  //Initially Zero
    TRISB = 0b00000000;  //All outputs
    PORTB = 0b00000000;  //Initially Zero

    //........................................
    for (i = 0; i < 3; i++)
    {
        SetChannel(i);
        count = GetCount();

        //printf doesn't support longs, so must convert to two INTs
        lsb = GetLSB(count);
        msb = GetMSB(count);
        printf("Channel %u LSB:%u, MSB:%u\r\n", i, lsb, msb);

        if (GetZero()) printf("Zero passed\r\n");
    }
    //........................................
}

unsigned long GetCount()
{
    int i;
    unsigned long count;
    count = 0;

    bitset(PORTA,PLSR);
    bitset(PORTA,CLKSR);
    DELAY_IT(); //Wait for SR to load
    bitclr(PORTA,CLKSR);
    bitclr(PORTA,PLSR);

    //Now data is static, shift it out.
    for (i = 0; i <= 31; i++)
    {
        bitclr(PORTA,CLKSR);
        count |= ((unsigned long)CNTDATA << (i)); //Read data bit
        bitset(PORTA,CLKSR); //Now toggle clock to shift in next bit
    }
    bitclr(PORTA,CLKSR);
    return count;
}

unsigned short GetZero()
{
    if (ZERODATA) //If Zin was triggered
    {
        //If zero has been passed, reset the latch
        bitset(PORTA,ZERORESET);
        DELAY_IT();
        bitclr(PORTA,ZERORESET);
        return 1;
    } else
    {
        return 0;
    }
}
```