

Solutions to Assignment 9 Serial and Parallel Interfaces

Question 1

- (a) The voltage levels are valid for an RS-232 interface since the positive values are greater than +3 volts and the negative values are less than -3 volts.
- (b) The diagram shows the bit period is about 200 μ s which is a bit rate of about 5000 bps. The nearest standard bit rate is 4800 bps.
- (c) The stop bit must be at a low level. All high-level bits after the start bit must therefore be data or parity bits. Since the eighth bit after the start bit is high it must be a data bit and thus 8 data bits were sent.
- (d) The data bits (high=0) are transmitted from least-significant (LS) to most-significant (MS) bit. Re-writing the transmitted bits in conventional order (from MS to LS) the bits are: 0001 0100 or 0x14 which is a control character (control-T or DC4).

Question 2

Measuring a voltage of 12 volts on the computer's RTS line indicates that the RTS signal is an output. This means the computer is wired up as a DTE. On a 25-pin serial interface pin 2 is transmit data (TxD). A voltage of -4 volts on pin 2 indicates that the TxD pin is an output and thus the tool's interface is also wired up as a DTE. The problem is likely to be that both devices are wired as DTEs, the two outputs are connected to each other, and they will be unable to transfer information. The easiest way to resolve the problem is to connect the two devices using a "null modem" to make one of the devices appear to be a DCE.

Question 3

```
/* APSC 380 Assignment 9 Question 3
   E Casas 97/11/25
*/
#define DATA    0x178
#define STATUS   0x176
#define CONTROL  0x180
#define BUSY     0x80
#define STROBE   0x01

void chprt ( char c )
{
    /* do nothing while busy */
    while ( speak ( STATUS ) & BUSY ) {
    }

    /* write character to data port */
    spoke ( DATA, c ) ;

    /* turn strobe low, then high */
    spoke ( CONTROL, 0 ) ;
    spoke ( CONTROL, STROBE ) ;
}
}
```

Question 4

```
/* APSC 380 Assignment 9 Question 4
   E Casas 97/11/25
*/

void triangle (int n)
{
    int i, j ;

    /* print lines of length [1,n] */
    for ( i=1 ; i<=n ; i++ ) {
        for ( j=0 ; j<i ; j++ ) {
            printf ( "*" ) ;
        }
        printf ( "\n" ) ;
    }

    /* print lines of length [n-1,1] */
    for ( i=n-1 ; i>0 ; i-- ) {
        for ( j=0 ; j<i ; j++ ) {
            printf ( "*" ) ;
        }
        printf ( "\n" ) ;
    }
}
```