

## Breaking a 1000 Series Through Com 2

The ApplicationBuilder and WinBuild programs use a special control sequence to establish communications with a 1000 Series product to by essentially stopping the BASIC program that is running and turning the REMOTE ON. This is equivalent to typing Ctrl-C on an external PC XT keyboard and pressing the REMOT key on the keypad. To establish communications with a 1000 Series product from a remote device (like a PC) from an application program you must follow the following timing sequence exactly. All times are in milliseconds and must be with an accuracy of +/- 1 milliseconds. If your BASIC program in the 1000 Series was not generated by the ApplicationBuilder (or WinBuild), you must remember to turn the REMOTE OFF otherwise, characters printed to the display in the BASIC program will also be echoed out the COM 2 port.

This sequence requires the use of a special break function. This occurs when the 1000 series' RS-232 receive line is pulled to a logic low.

Flush the PC's transmit buffer

Turn the BREAK signal on

Wait 18 milliseconds

Turn the BREAK signal off

Wait 60 milliseconds

Send the following characters (given here in hexadecimal):

0xBA, 0xE5, 0xC1, 0xA5, 0x96

Wait 400 milliseconds

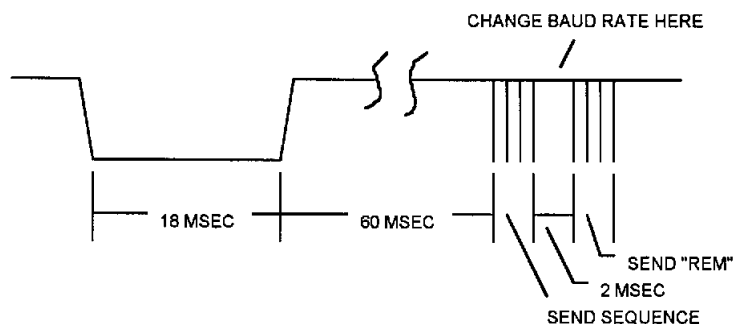
Flush the PC's receive buffer

Send the following characters (given in mixed ASCII and hexadecimal):

'R', 'E', 'M', 0x0D

The PC's receive buffer should read back:

'R', 'e', 'a', 'd', 'y', 0x0D, 0x0A



```

BOOL BreakIt( CHAR *Command, BYTE *Signature )
{
    BOOL fReset;
    CHAR a[32];
    SHORT Status, Reg, GobbleState;

    fReset = !strncmp(Command, "RESET", 6) ? TRUE:FALSE;

    GobbleState = ComGobble;
    ComGobble = FALSE;
    Status = FALSE;

    // set TX low, wait 18ms, set TX back high, wait 60ms
    Reg = inp( PortInfo.uart.u8250.lcr_addr );
    outp( PortInfo.uart.u8250.lcr_addr, Reg|64 );
    delay( 18 );
    outp( PortInfo.uart.u8250.lcr_addr, Reg&~64 );
    ComFlush( 60 );

    // send signature characters, wait 60ms

    do { ComPutChr(*Signature); } while( *Signature++ );
    ComFlush( 60 );

    // force the pc comport to 9600,8,N,1, delay 20ms

    ComStop();
    Config.Com[Config.ComPort].Echo = FALSE;
    PutComPortValues( Config.ComPort, 9600L, DATABITS8, PARITY_NONE, STOPBITS1 );
    ComStart( Config.ComPort );
    ComFlush( 20 );

    // if reset exit early

    if( fReset )
    {
        for( Status = 0 ; Status < 4 ; Status++ )
            ComFlush( 999 );
        Status = TRUE;
        goto ExitOut;
    }

    // verify that the Eason unit is present by echoing a "REM" statment

    if( ComPutStr( "\x7REM\r", 5 ) == FALSE )
        goto ExitOut;

    TimeoutSw = TRUE;
    if( ComGetLine( a, sizeof(a) ) == FALSE )
        goto ExitOut;
    if( strcmp( a, "REM" ) )
        goto ExitOut;

    // verify that the Eason unit is back in BASIC by looking for "Ready"

    if( ComGetLine( a, sizeof(a) ) == FALSE )

```

---

```
        goto ExitOut;
if( strcmp( a, "Ready" ) )
    goto ExitOut;

        // if a command needs to be sent then issue it now

if( *Command && Command != NULL )
    Status = ComPutStr( Command, strlen( Command ) );
else
    Status = TRUE;
```

```
ExitOut:
    TimeoutSw = FALSE;
    ComGobble = GobbleState;
    return( (Status == TRUE) ? TRUE:FALSE );
}
```