# UNIQUE CONSIDERATIONS FOR DATA RADIO UARTS

By John Anthes, Murata Electronics, North America, Inc., Dallas, Texas

A receiver system, used for data recovery, involves the sensing of a signal in the microvolt range with an antenna. The information on the signal is then detected and amplified to a logic level. This paper will address special considerations required in order to recover data after it has been processed through a receiver system.

PULSE RESPONSE THROUGH A RECEIVER

All receivers utilize filters. A filter performs the function of rejecting signals at undesired frequencies and passing signals at desired frequencies. Most receiver systems, such as the RX and TR series built by Murata, use both bandpass filters and lowpass filters. Bandpass filters will pass signals over a band of frequencies and reject signals lower or higher in frequency. Low pass filters will pass signals from d.c. to some higher corner frequency and then reject signals that are higher than the corner frequency.

fo
Bandpass filter
Response

d.c.   fc   frequency
Lowpass filter
Response
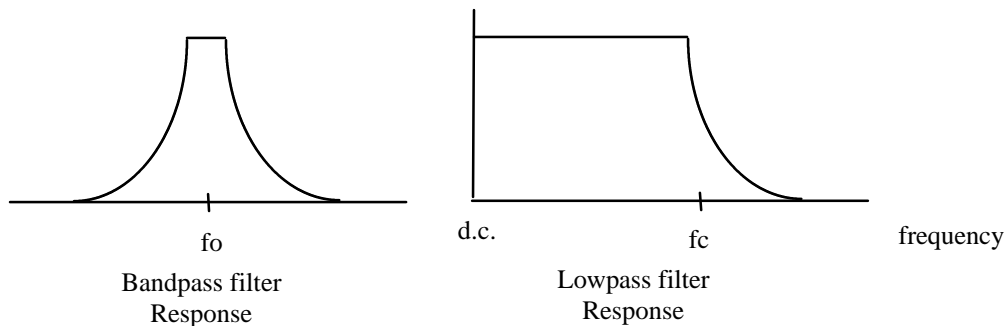
RF filters are usually in the early stages of a receiver. Their primary function is to offer selectivity and intermodulation protection to the rest of the receiver. The lowpass filters are usually after the detector section and before the bit slicer of the receiver. This filter, if much narrower than all the previous filters in the receiver, sets the noise bandwidth of the receiver and thus the sensitivity. The noise power can be determined by the following equation:

Noise Power Out in dBm $= -174 + NF + Gain + 10\log(BW)$

NF is the noise figure of the receiver in dB
Gain is the signal gain of the receiver in dB
BW is the Noise Bandwidth in Hz.

The above equation implies that a narrower filter bandwidth will lower the noise power of a receiver, and thus allow for greater sensitivity.

In order to fully appreciate the trade-off between filter bandwidth and data rate, we need to analyze the effects of the filter on the data. The plots below show simulation of data, for different data rates, after passing through a lowpass filter with a bandwidth of 20 kHz. The filter, used in this simulation, is similar to the filter used in the RX and TR products manufactured by Murata.

10000 bits/sec

20000 bits/sec

40000 bits/sec
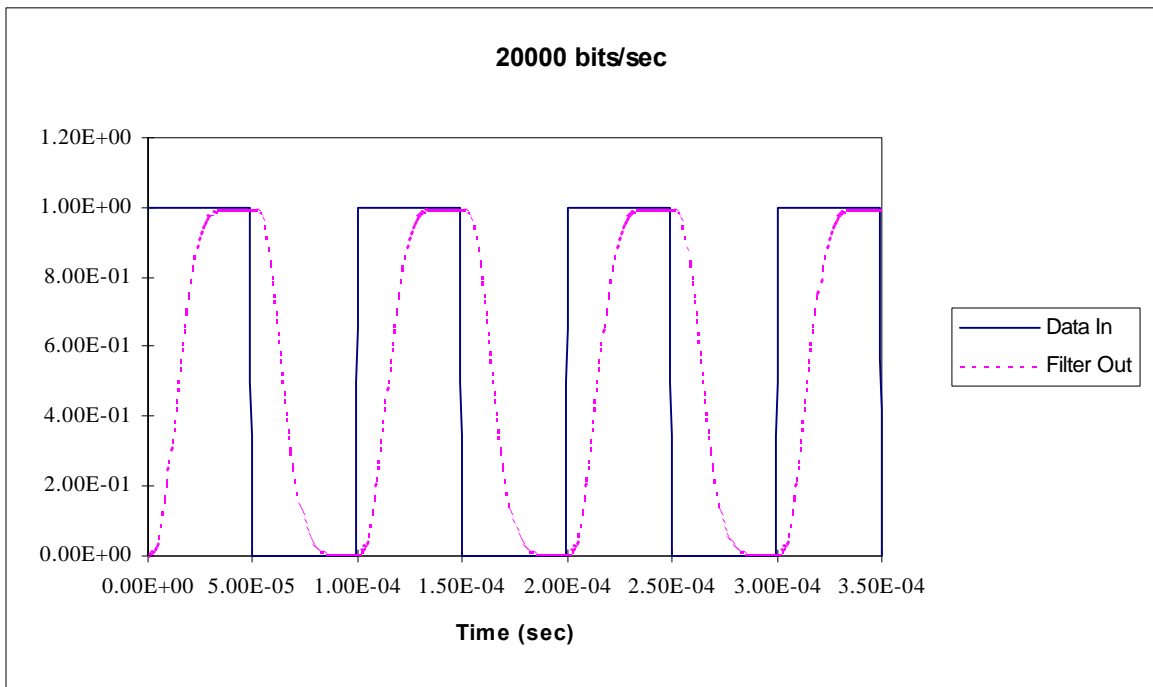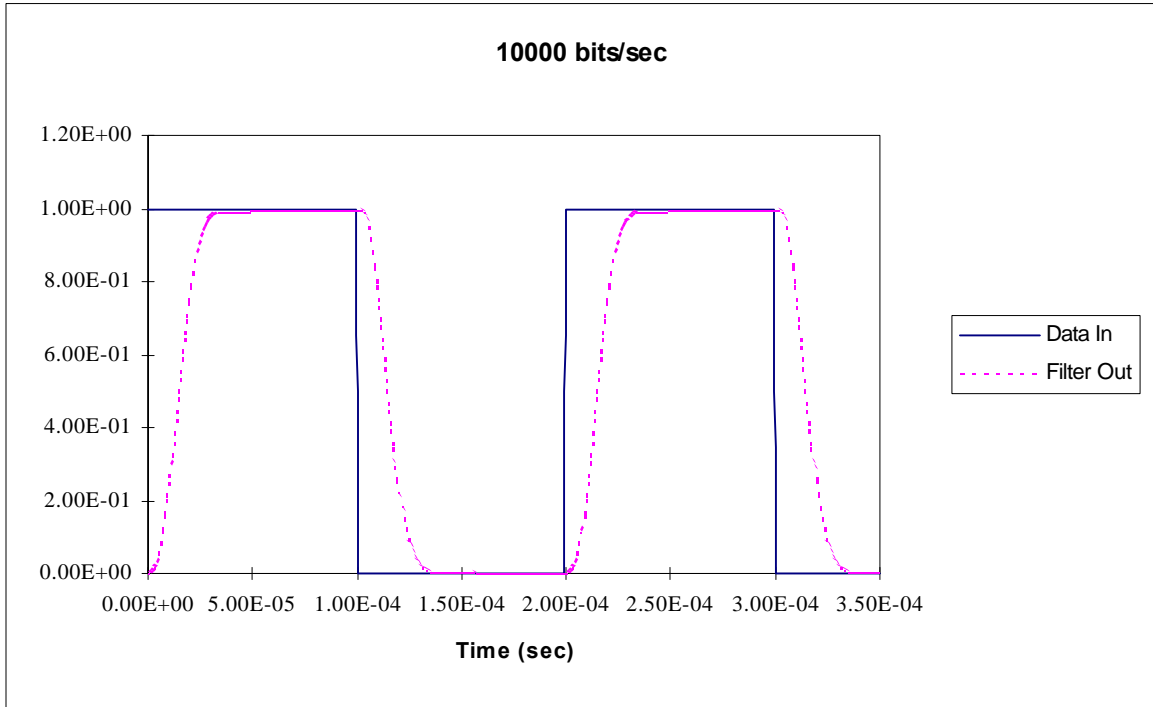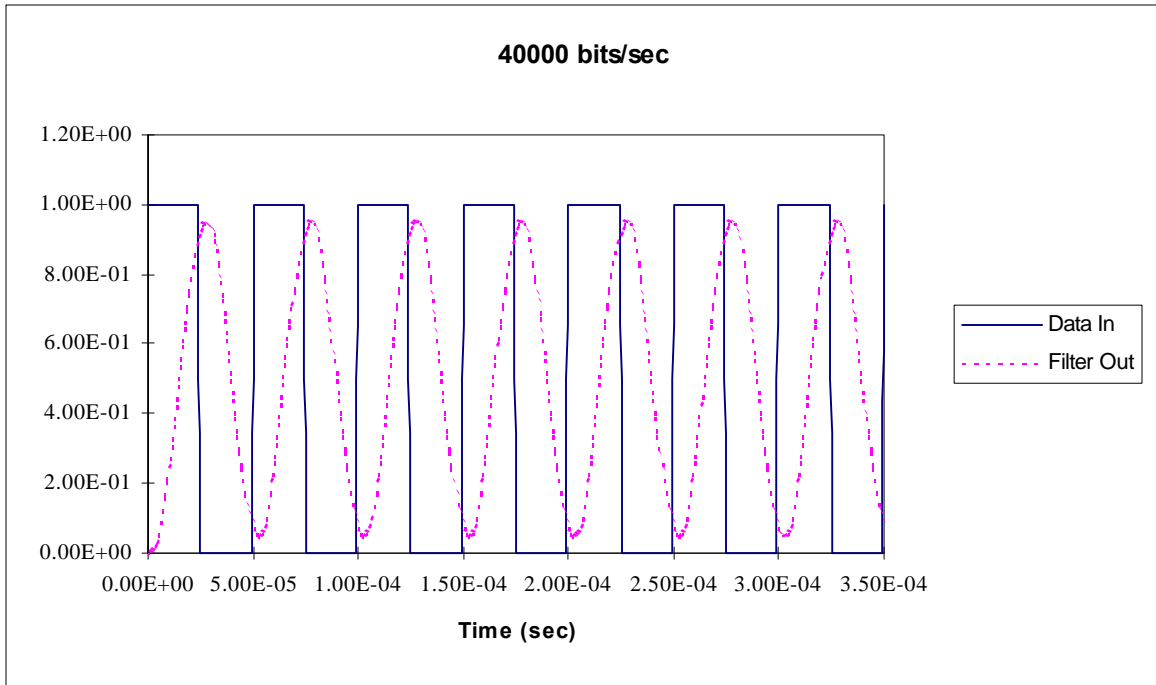
The above plots show the effects of the filter on the data as the data rate is increased. It is clear that as the data rate increases the available pulse at the output of the filter shrinks.

THE EFFECTS OF A BIT SLICER

Most receivers, such as the RX series built by Murata, implement a bit slicer circuit following the lowpass filter stage. The bit slicer is a threshold and comparator circuit that keeps the output squelched or quiet until a signal crosses the threshold. Using the above 20000 bit/sec example with a threshold of .2, let's analyze the effects of the output when the input signal level is at .25 and 1.

Low Signal Level of .25
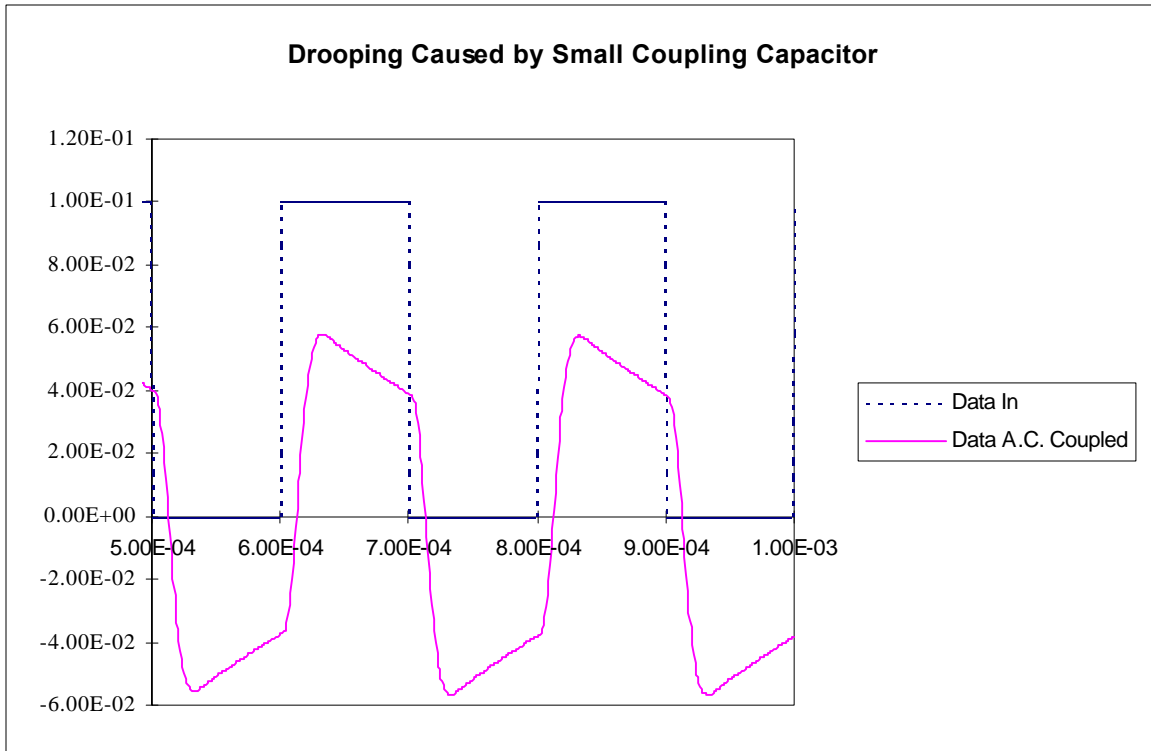
High Level Signal of 1
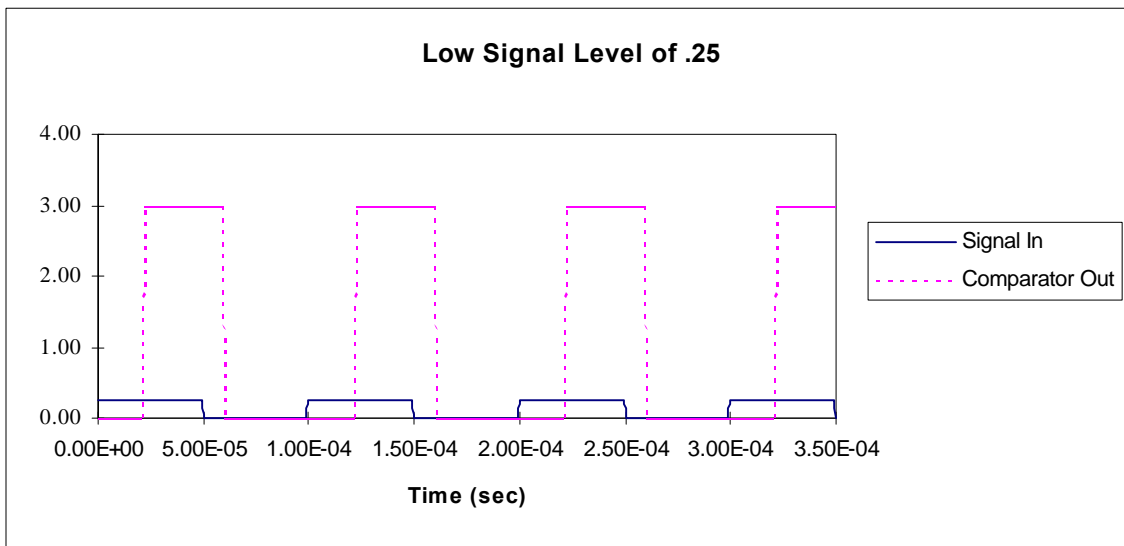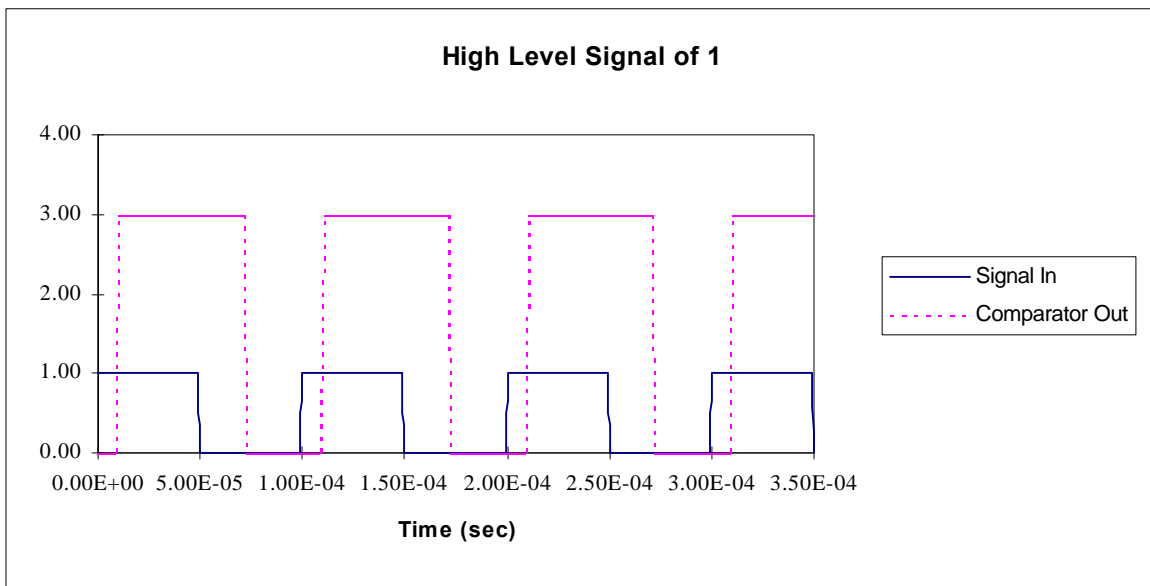
Comparison of the simulation results shows that for a low level signal, the data out of the comparator, for the 1's, is approximately 30% narrower than the data in. In the case of a high level signal, the data out of the comparator, for the 1's, is approximately 25% wider than the data in. This is important to understand if maintaining the data width or pulse width is critical. When maintaining data width or pulse width is important to the performance of the receiver system, it will be necessary to either lower the data rate or increase the low pass filter bandwidth.

The ASH transceiver, recently introduced by Murata, offers a solution to the above problem by introducing a second threshold that is maintained at the 50% point of the pulse. This is done by tracking the peak of the log detected input pulse and offsetting the threshold down by 6 dB. This assures that the threshold is located at the most optimum place to maintain pulse or data width.

BALANCED VS UNBALANCED CODE

Many receivers, such as the ASH receiver and transceiver, A.C. couple the data by use of a series capacitor. This is done to remove the D.C. offsets in the amplifiers and comparators. The maximum length of continuous 1's or 0's, in the data being sent, is critical in sizing the series capacitor. If the capacitor is too small, the data will begin to decay or droop. This can cause detection errors in the bit slicer. The plot below shows a simulation when the series coupling capacitor is too small for the data pulse width. Note the drooping present on the data at the output of the series coupling capacitor. The disadvantage of a large series coupling capacitor is that it requires a longer time to charge the capacitor to the steady state condition. The receiver, therefore, requires more time before it is able to decode the data.

Drooping Caused by Small Coupling Capacitor

Data In

Data A.C. Coupled

1.20E-01
1.00E-01
8.00E-02
6.00E-02
4.00E-02
2.00E-02
0.00E+00
-2.00E-02
-4.00E-02
-6.00E-02

5.00E-04   6.00E-04   7.00E-04   8.00E-04   9.00E-04   1.00E-03

One

Zero

For this reason, it is best to use data that is D.C. balanced. Simply stated, D.C. balanced is when the number of 1's and the number of 0's in a period of time are equal. It is best when the time period being averaged is on the same order of magnitude as a symbol or bit period. One possible approach is to use Manchester coding as displayed below:
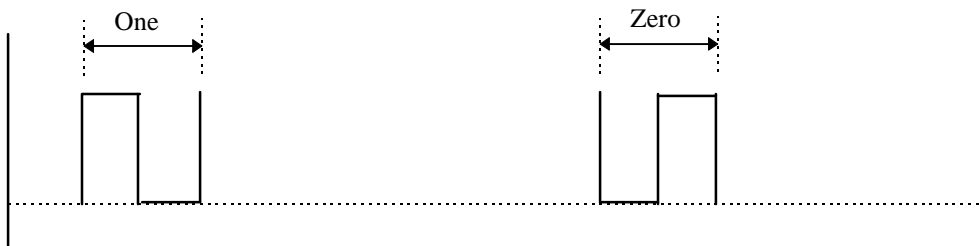
Manchester coding ensures that each bit of the data is D.C. balanced. Another advantage of Manchester coding is it provides an edge within each bit period that can be used to align the receiver's clock. A long string of 1's or 0's, using Non-Return to Zero (NRZ) coding, provides no information for clock extraction. The disadvantage of Manchester coding is that it requires twice the bandwidth as compared to simple NRZ codes. Another possible solution is to create a symbol table. A symbol table with 16 different symbols can be generated using 6 bits which guarantees that no more than 4 consecutive bits are the same. This scheme requires only 1.5 times the bandwidth when compared with NRZ coding. The table below presents a symbol table that has been used at Murata:

|	
Nibble = 0	010101
Nibble = 1	110001
Nibble = 2	110010
Nibble = 3	100011
Nibble = 4	110100
Nibble = 5	100110
Nibble = 6	100110
Nibble = 7	010110
Nibble = 8	011010
Nibble = 9	101001
Nibble = 10	101010
Nibble = 11	001011
Nibble = 12	101100
Nibble = 13	001101
Nibble = 14	001110
Nibble = 15	011100

CLOCK RECOVERY

The plot below illustrates a typical low level signal in the presence of noise at the bit slicer's input.
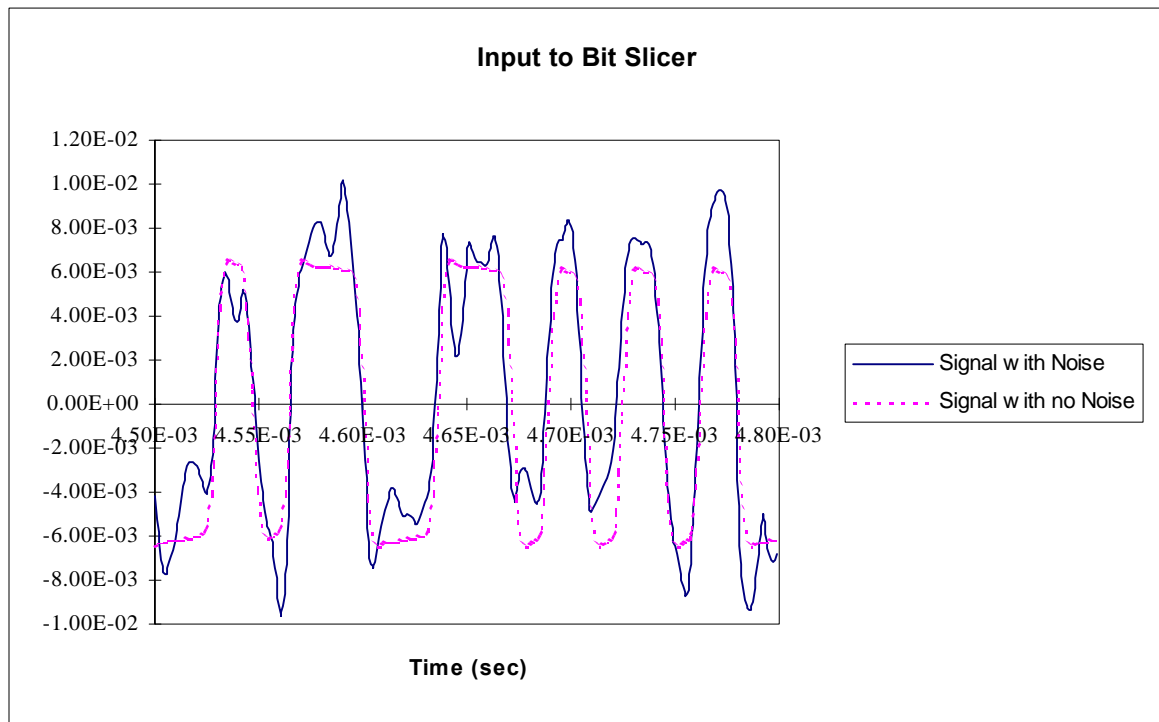
The dash line is data without noise and the solid line is data with noise present. As the signal approaches the noise level of the receiver, it becomes more difficult for the bit

slicer to accurately define the zero crossing points. This causes what is known as jitter on the data edges. Using the data in the above plot, the plot below compares the bit slicer output for data with and without noise. The jitter at the output of the bit slicer can be seen by comparing the leading and falling edges of each pulse.

Jitter at the Output
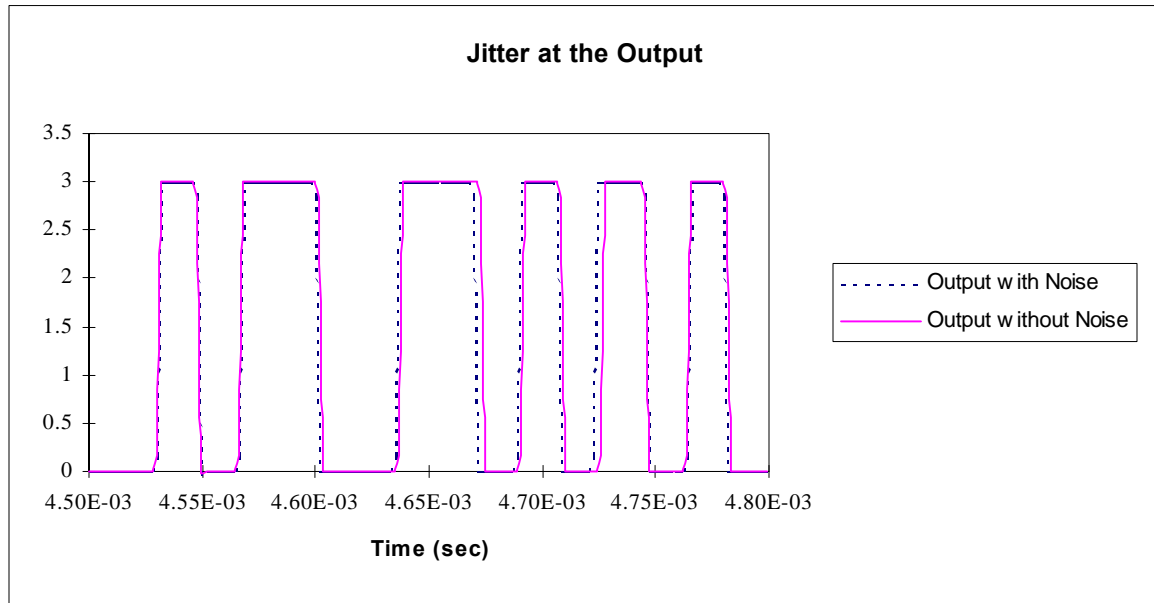
The data edges are important in defining where the bit intervals occur, so the data can be properly sampled. Since the cause of the jitter is due to random noise, it is possible to average enough edges to estimate the true bit interval. The BASIC program, listed below, is an example of a technique to accomplish this. The DSO variable in the program is the input variable from the bit slicer that indicates a zero crossing has occurred.

```
' CLOCK_R2.BAS
' 98.12.13 @ 16:40 CST
' Copyright Frank H. Perkins, Jr., 1998


' Constants:

RampTop = 10000              ' PLL ramp top value
RampMid = 5000               ' PLL ramp mid value
RampInc = 1000               ' PLL ramp increment
RampBmp = 250                ' 2.5% PLL advance/retard value


' Variables:

DSO = 0                      ' data slicer output
LDS = 0                      ' last data slicer output
PLL = 4500                   ' PLL ramp value
```

```
CLK = 0                          ' PLL clock value
LCK = 0                          ' last PLL clock value
DRO = 0                          ' data recovery output


ClkData:                         ' clock recovery and data estimation

  IF (DSO XOR LDS) = 1 THEN      ' if at an edge then
    IF PLL >= RampMid THEN       ' if ramp at or past midpoint
      PLL = PLL + RampBmp        ' bump PLL to advance
    ELSE
      PLL = PLL - RampBmp        ' else retard
    END IF
  END IF

  PLL = PLL + RampInc            ' increment ramp
    IF (PLL >= RampTop) THEN
      PLL = PLL - RampTop        ' wrap on overflow
    END IF

  LCK = CLK                      ' store last clock value
    IF (PLL - RampMid) < 0 THEN
      CLK = 0                    ' clock is 0 before midpoint
    ELSE
      CLK = 1                    ' clock is 1 at or beyond midpoint
    END IF

  IF CLK = 1 THEN                ' if clock 1
    IF (CLK XOR LCK) = 1 THEN    ' and it was 0 last sample
      DRO = DSO                  ' estimate data value
      FLG = 1                    ' and set data flag
    ELSE
      FLG = 0                    ' else reset flag if not 0 to 1 transition
    END IF
  ELSE
    FLG = 0                      ' and/or reset flag if clock 0
  END IF
```

This program samples the data 10 times during a bit interval. When a zero crossing occurs in the data, and the sample counter is beyond the midpoint of the count, an offset is added to the sample counter. Likewise, when a zero crossing occurs before the midpoint of the count, an offset is subtracted from the sample counter. This continues until the count of the sample counter fits within the average zero crossings of the data.

This program declares a 1 or 0 based on a single sample taken at the center of the counter. This scheme, to extract clock information, can be programmed into a DSP.

There is one other possible cause for jitter when using the ASH transceiver product manufactured by Murata. The proper selection of the resistor on pin 14 (PRATE) is extremely important. The ASH technology ensures stability and low current by sequencing the receiver amplifiers on and off. In a sense, the receiver is sampling the input signal at the rate set by the resistor on pin 14 (PRATE). If the sampling or sequencing rate is not fast enough, a good representation of the data will not occur and the output data will appear jittery. Murata recommends that the sequencing rate be set to at least 10 times the data rate expected.

CONCLUSION

Due to the effects of the filters and noise as outlined in this paper, it is highly recommended that a hardware UART not be considered for data radio applications. Such UARTs were intended for high signal to noise ratio applications, with little to no jitter or pulse width degradation. Therefore, for data radio applications, which must work with low signal to noise ratios, a software UART design is required in order to extract clock and data information with large variances in timing.