Simple Clock & Data Recovery

There are two main objectives in receiving data, one is to extract or sync the RTC to the transmitted data; in turn this will allow us to sample the received data at the correct time. The key to extracting or syncing of the RTC is the use of RFM's Special Start Symbol, which is designed to use the unique characteristics of the Ash Transceiver/Receiver. This Start Symbol breaks the DC balance rules need for proper data reconstruction, but in doing so allows a micro to sync the RTC to the incoming data with very few instructions, and without the need to use a highly stable frequency source (crystal). Please refer to http://www.rfm.com/products/tr_des24.pdf, Section 1.4.3 for details on DC balance data.

The **Special Start Symbol** is three (3) zeros followed by eight (8) ones and a single zero.

000111111110

The eight ones will charge the Base Band Coupling cap high enough that when the first zero is sent and received the Base Band Coupling cap has to discharge until it hits the threshold for a zero in the data slicer. This time is 50% of the bit time if the Base Band Coupling cap is sized according to RFM's recommendations. Please refer to http://www.rfm.com/products/tr_des24.pdf, Section 2.6.1 for details in picking the correct Base Band Coupling Cap.

The micro needs to have the RTC set to interrupt every bit time, for instance 52us for 19.2Kbs. When the RTC interrupts the micro, the firmware needs to count the number of consecutive ones being received. If at any time a zero is received, this count must be reset back to a count of zero. At the point the ones counter reaches a count of seven (7), the firmware must go into a hard wait for the falling edge of the first zero (0). As soon as the micro sees the falling edge of the zero (0), it must reset the RTC timer to ONE bit time. This will sync the RTC to the center of the incoming data bit, which is the optimal sampling point of the data. From this point on, the RTC interrupt will sample the data bit and shift the bit into a holding reregister and return.

This technique has been implemented in low cost micros and DSPs. The smallest used the internal clock without a crystal reference, and was able to extract the clock and data at 19.2Kbs using a 4 MHz RC oscillator.

The following is a pseudo code and flow chart:

RFM
Bob Nelson
02/25/2003

Simple Clock & Data Recovery
Pseudo Code

Init:

Load Bit_Counter with 7
Clear Start_Detect_Flag
Initialize RTC for 1 bit time (52us for 19200)

Main:

Enable Interrupts
.
.
.
Jump Main

ISR:

Start_Detect set goto Get_Data
Receive Data = 1 goto Data_one
Load Bit_Counter with 7
Return from Interrupt

Data_one:

Decrement Bit_Counter
If Bit_Counter = 0 goto Wait
Return from Interrupt

Wait:

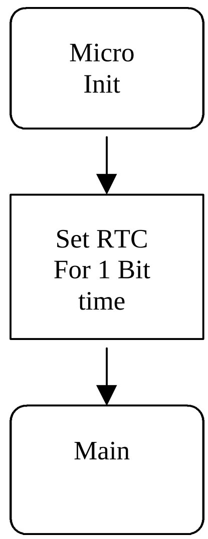
Receive Data = 1 goto Wait
Set RTC for1 Bit time
Set Start_Detect = 1
Return from Interrupt

Get_Data:

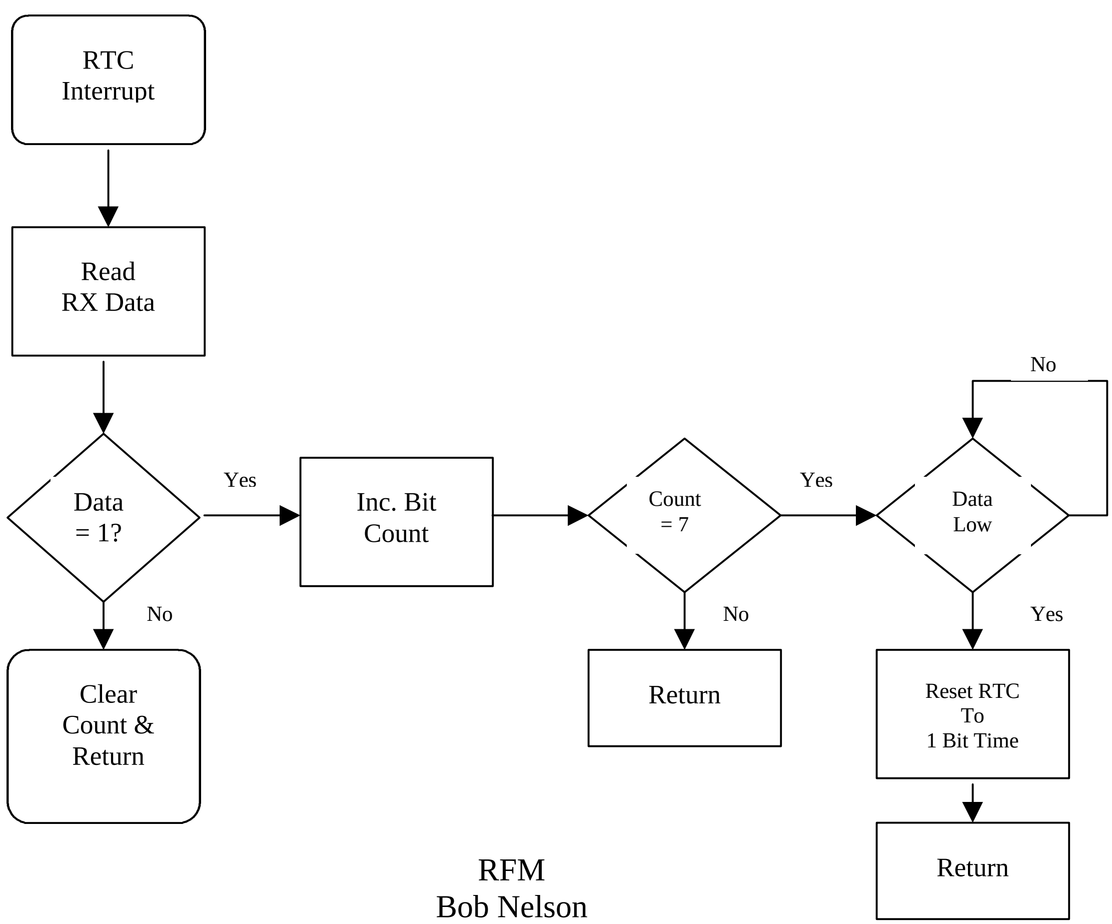
Shift in Data
.
.
.
.
.
Return from Interrupt

RFM
Bob Nelson
02/25/2003

Simple Clock & Data Recovery
Flow Chart



Interrupt Service Routine



RFM
Bob Nelson
02/25/2003