OS Sensor Clone Design

This design mimics an OS THGR810 sensor.

Getting a sensor to actually work with an OS console was difficult as there is a lot going on that's not obvious. The data timing is quite critical, needing to be within 15 us of the required timing, and the confusing aspect of the receiver in the OS console modifying the data timings.

The Sensor needs to be crystal controlled to get the timings right and has to use the PIC's internal timers to generate the correct RF pulse timings as well as the required transmission periods to suit the OS consoles gated receiver.

Among other things, OS consoles do not run the receiver continuously; it is turned off (actually powered down) except when a transmission is expected, or when in search mode.

There still are some dropouts which I cannot figure out whats causing them, as it seems to be random and infrequent.

Because of the need for 2 crystal oscillators, its not possible to use a smaller PIC with only 8 pins as none of the smaller PICs can support 2 crystal oscillators running simultaneously.

Notes on the graphs.

These are taken from my CRO (oscilloscope). The yellow trace is the data pin of the 433 Mhz transmitter in the clone. The purple trace is the data coming out of the receiver in the OS console.. The CRO is edge triggered on the leading edge of the first data pulse going to the transmitter.

You will notice that the data coming out of the OS receiver is shifted in time compared to whats going into the transmitter and the data pulses themselves are also time shifted, with the shorter pulses being more affected. This effect also happens with the real THGR810 sensor , but the data going into the real sensor has a strange trailing shape of which I have not been able to determine what its for. The graph THGR810.bmp shows the effect.

Graphs 2 & 3, 4 & 5 , 6 & 7 and 8 & 9 are different zooms of the same trace. This was done because my CRO only has 2 delta cursors so I can only show times for 1 trace at a time.