RADIO TELEMETRY RECEIVER

S.R.K.Branavan, H.M.G.Gunasekera and N.H.A.N.L.Nandasiri.
Department of Electronic and Telecommunication Engineering, University of Moratuwa.

RADIO TELEMETRY RECEIVER

S.R.K.Branavan, H.M.G.Gunasekera and N.H.A.N.L.Nandasiri.
Department of Electronic and Telecommunication Engineering, University of Moratuwa.

ABSTRACT

A narrow band tunable radio telemetry receiver capable of operating over the license free bands of 26.9 – 27.3 MHz and 46 – 50 MHz was designed. The receiver is tuneable within the specified frequency bands, and supports a data rate of 9600 baud. It has link control, flow control and backward error correction capability. The receiver has a synthesised local oscillator, allowing digital tuning and remote system control through an RS-232 serial interface. The controller is capable of handling both the receiver and the transmitter and has a 4 kByte buffer each for the transmission and receipt data. The system connects to the external device through an RS-232 serial interface. The controller was designed around the z80 microprocessor. The required hardware was constructed and tested completely. The software was also completed but was only partially tested for the lack of a transmitter to establish a data link.

1. INTRODUCTION

Radio telemetry systems basically provide a wireless data link between two points. They come into their own in situations where a low cost, relatively low data rate data link has to be set up between two fairly close points, but when the two points cannot easily be connected using cable. They are also highly suitable for temporary or ad hoc data links since they can be set up quickly and easily by connecting a transceiver to each of the data terminals.

Telemetry systems are available in a wide range. Basic systems do not process the data in any way but simply treat the data as a signal to be transmitted and received over the radio link. Complex systems on the other hand offer source and destination addressing, link, flow and error control and data buffering along with other facilities. Telemetry systems are available either as a complete transceiver module or as separate independent transmitter and receiver modules.

The aim of this project was to design and build the receiver section of a narrow band tuneable radio telemetry module capable of operating over one or more license free bands, so as to provide a practical solution to short distance, low data rate data transfer problems. The receiver will be tuneable within the specified frequency bands, and support a data rate of 9600 baud. It will also be able to detect errors in the received data and automatically request for a retransmission.

2. THE RADIO TELEMETRY RECEIVER SYSTEM

The objective of the project was to design and build the receiver section of a narrow band, digitally tuneable, stand alone radio telemetry module capable of operating over one or more license free bands. The receiver also had to be able to support a data rate of 9600 baud, and be able to detect and correct errors.

Description of the Radio Telemetry Transceiver System

The radio telemetry transceiver consists of three separate but interrelated sections, these are the transmitter, the receiver, and a controller to manage both the other blocks and the data link itself. The error correction process through an automatic repeat request capability needs a bi-directional data link, which would require two of these transceivers, one at each end. The link would also have to support the transfer of extra error correction information. The radio telemetry receiver supports source and destination addressing, link and flow control, and backward error correction. This is achieved by framing the data into packets with a header containing addressing, link control, flow control and repeat request data, and terminated with a CRC remainder for error checking. The format of this data frame is shown in figure 1. A block diagram of the complete radio telemetry system is shown in figure 2.

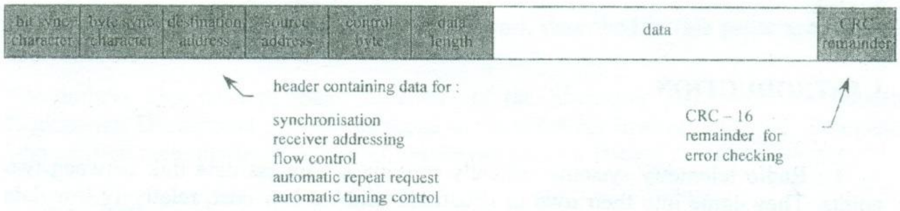
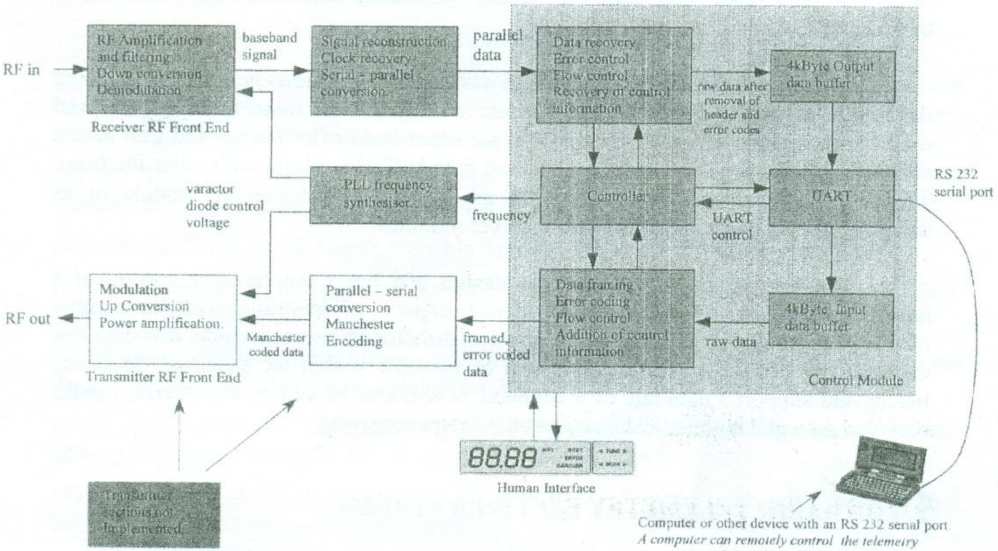


Figure 1 : Format of the data frame supported by the telemetry receiver.



Computer or other device with an RS 232 serial port. *A computer can remotely control the telemetry transceiver, while a dumb device may consider it simply as the serial port of the remote device*

Figure 2 : Block diagram of the Radio Telemetry Transceiver.

The scope of this project covers the receiver and the receiver controller for the radio telemetry system. The receiver is divided mainly into 7 modules for ease of design and construction. These are :

Receiver front end : The input to this module is the RF signal. This RF signal is filtered out, amplified and down converted to the IF. Then IF filtering, amplification and demodulation is done. The baseband signal thus obtained is filtered and amplified and then given to the receiver digital section.

Receiver digital section : This section converts the baseband signal to a parallel data stream. The first stage recovers the clock from the input baseband signal. Then byte synchronisation, serial to parallel conversion and interrupt generation is done. The interrupt service routine reads the parallel data for further processing in the controller.

Controller : The controller handles the overall control of the transceiver. Processing the header information, handling error and flow control, buffering of received data and sending out data through RS232 port to the external device are the basic functions of the controller. Similarly it handles the transmitter side by generating the header, buffering the data to be transmitted, calculating the CRC bytes and sending the frames to the transmitter digital section. It also controls user interface and display.

Programmable Divider and Phase Comparator : This section is essentially the divider and phase comparator for the PLL frequency synthesiser, which produces the local oscillator. The other two parts of the synthesiser – the low pass filter and the voltage controlled oscillator are included in the RF front end.

User Interface : This module allows the user to tune the receiver manually and to scan the operating bands for a transmitter.

Transmitter digital section : This is part of the transmitter. But since it shares certain circuits with the receiver digital section, both of them were designed as one circuit. The transmitter digital section handles the serial to parallel conversion and Manchester encoding of the transmit data, and interrupt generation.

Transmitter RF Front End : This is also part of the transmitter and was as such not implemented. This section converts the baseband signal to be transmitted to a RF signal. Its functions are modulation, up conversion to the final transmission frequency and power amplification of the RF signal to the level required for transmission.

Specifications

The following were our design specifications.

- Stand alone receiver.
- Digitally tunable, synthesized carrier.
- Frequency range : 26.9 - 27.3 MHz and 46.0 - 50.0 MHz
- Binary FM modulation.
- 9,600 baud data rate.
- 150k Channel bandwidth maximum.
- Frequency scanning capability.
- Maximum range of 5km.
- Error detection and automatic repeat request capability.
- Automatic retuning in case of continuous interference.
- Standard RS-232 serial data interface.

3. RECEIVER FRONT END

The receiver front end does the entire signal processing involved in receiving the RF signal and converting it to the baseband signal. This involves RF filtering and amplification, down conversion to the intermediate frequency, IF filtering and amplification, demodulation, baseband filtering and amplification. The RF front end is implemented as a superheterodyne receiver, and consists of the following sections.

Tuned RF Amplifier.

This block is implemented using a wideband RF amplifier IC and external filters. The NE5200D dual RF amplifier with a 1GHz bandwidth and 7dB gain (each) was used (since it was available). Using tunable RF filters will introduce tracking problems between the RF filters and the local oscillator. Also the required frequency bands are each separately smaller than twice the intermediate frequency (20MHz). Therefore it was decided to use two fixed frequency filters -- one for each band – and to switch between them as needed. The filter switching is done using two *two-pole two-throw* relays. The 26.9 – 27.3MHz band is referred to as band 1 and the 46 – 50MHz as band 2. The circuit was tested and gave a pass band gain of 0dB, the amplifier gain compensating for the losses in the filters. Image rejection was 100dB approximately for both the bands. The filters used for both the bands are of the conventional image parameter 3-element band pass structure[4]. Each band 1 filter is made of one full section, whereas each band 2 filter is made of two cascaded full sections. The filters for both bands are made of two cascaded stages. This construction gives a minimum image rejection of 130dB theoretically and about 100dB practically. Figure 3 shows the schematics of a single stage of the filters for the two bands and their simulated responses. For both the filters the practical responses coincided closely with the simulation results, being different by only a few dB in the required frequency bands. All circuit simulations were carried out using *MicroSim DesignLab* versions 5.1 and 7.1 of *MicroSim Corporation*.

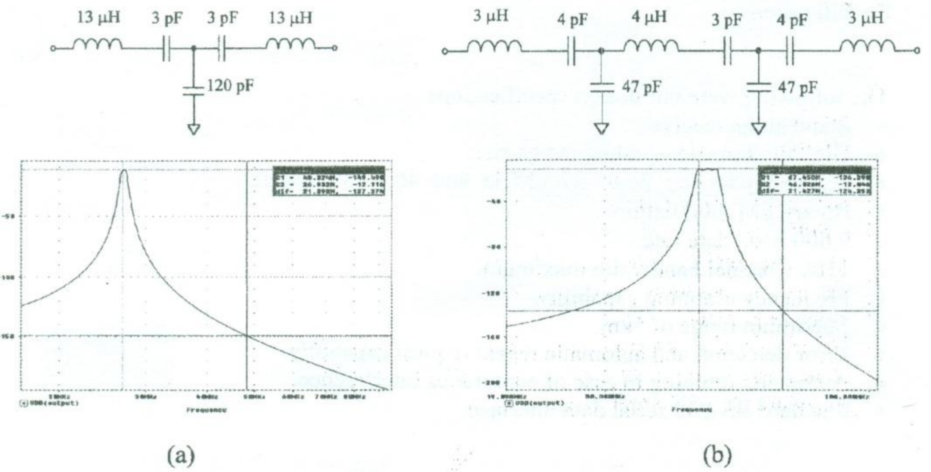


Figure 3 : RF filters and simulated responses; (a) band 1, (b) band 2

Mixer and Local Oscillator.

A varactor diode tuned oscillator is used for the local oscillator to allow digital tuning. The varactor diode itself is controlled by a PLL frequency synthesiser. The LO is implemented using the NE602AN double balanced mixer and oscillator IC. The circuit schematic of the mixer and varactor tuned local oscillator are given in figure 4. Two varactor diodes are used in parallel to achieve the range of capacitance required for tuning over the whole range of operating frequencies. The design shown allows tuning from below 36.9MHz to slightly above 60Mhz for a tuning voltage from 0 to 24V. A dual gate FET (BF960) was used in the cascode configuration as the LO output buffer to stop the PLL frequency synthesis circuits connected to the LO from loading or injecting noise into it. The local oscillator circuit was simulated using its equivalent circuit. But the simulation did not match the practical results sufficiently accurately, and only general trends could be gleaned from the results.

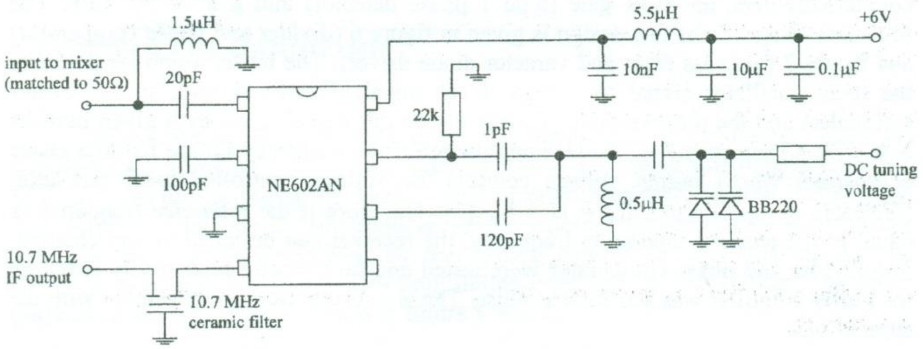


Figure 4 : Schematic diagram of mixer and varactor tuned local oscillator

IF Filter, Amplifier and Detector.

A 10.7 MHz ceramic FM IF filter is used as the IF filter. The IF amplifier and detector are implemented using the TCA420A FM IF amplifier and detector IC. The

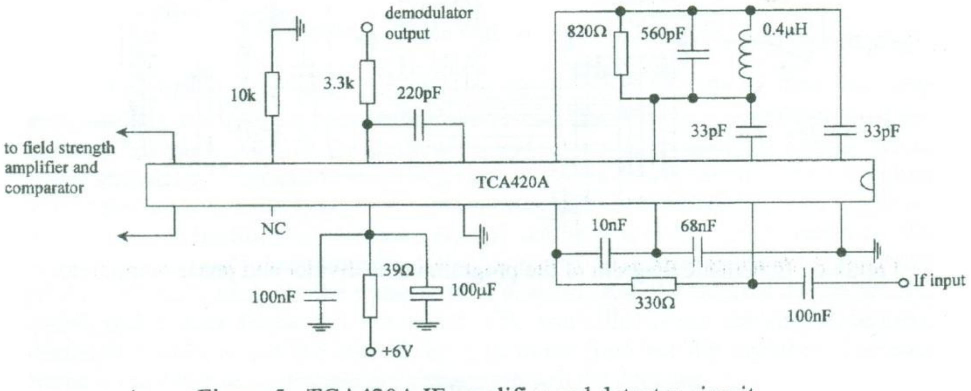


Figure 5 : TCA420A IF amplifier and detector circuit.

schematic diagram of the TCA420A IF and detector circuit is show in figure 5. When tested using a modulated IF signal from a signal generator (an IF signal with no noise added) the circuit produced a noise free demodulated baseband signal for an IF signal strength as low as 1nW (-60dBμ). The demodulator output is amplified to logic level using an amplifier with a first order lowpass response. The signal strength output is also fed through an amplifier and comparator with adjustable gain and threshold.

Synthesised Local Oscillator.

The PLL frequency synthesiser controls the voltage controlled local oscillator to generate an output frequency that is an integer multiple of a reference frequency. The value of the multiplier is used to set the LO frequency and thus tune the receiver. The frequency synthesiser was implemented using digital programmable counters/dividers, an XOR gate (type 1 phase detector) and a 3rd order LPF. The schematic diagram for this design is given in figure 6 (divider and phase comparator) and figure 7 (lowpass filter and varactor diode driver). The buffer amplifier converts the local oscillator signal to a logic level signal. The fixed high speed divider (prescaler) and the programmable divider divide this signal down by a given number N to give a frequency f_{LO} / N. This and the reference frequency f_{ref} are fed to a phase comparator whose output voltage controls the voltage controlled local oscillator. Therefore at equilibrium, $f_{LO} = N \times f_{ref}$,[7]. Therefore if the reference frequency is equal to the channel separation frequency, the receiver can be tuned to any channel. The divider and phase comparator were tested and functioned satisfactorily although the buffer amplifier was sensitive to noise. The test results were in agreement with the simulations.

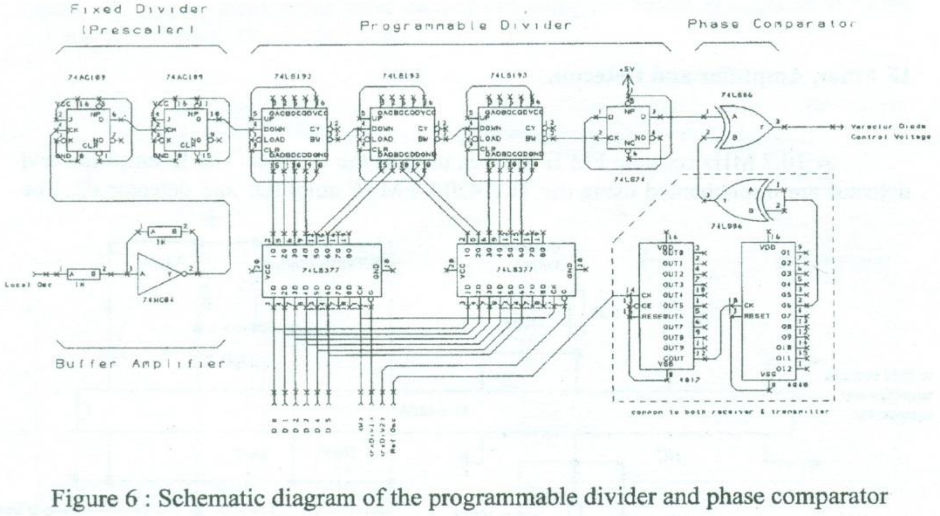


Figure 6 : Schematic diagram of the programmable divider and phase comparator

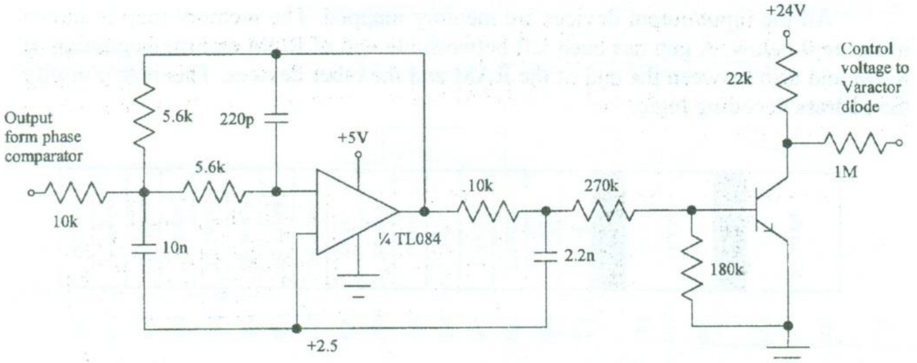


Figure 7 : Phase comparator low pass filter and varactor diode driver.

4. CONTROLLER HARDWARE

In the Radio Telemetry Receiver, the data has to be extracted from the received data frame and sent to the external device for which this data is destined. This involves processing of the control word, error control, flow control, link control, data buffering and serial data transfer to the external device, etc. All of these functions have to be performed by the system control circuit. This circuit was designed to be capable of handling both the receiver and the transmitter. The block diagram of the controller is shown in figure 8 below.

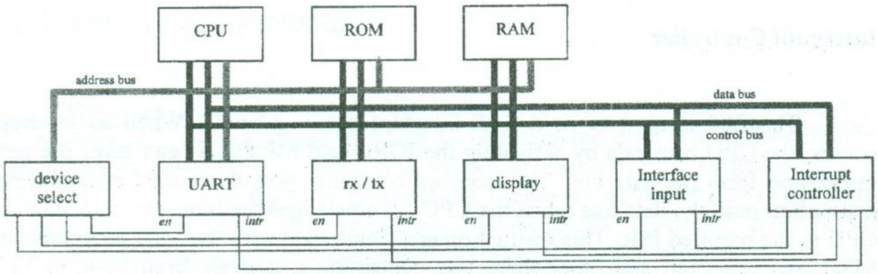


Figure 8 : Block diagram of the controller.

The functions of this controller can be described as follows. The controller gets parallel data from the receiver/transmitter module through the data bus. First the control characters such as the destination and source addresses, the control word, frame size etc. of the data frame are decoded. Then the data is stored in the buffer and CRC calculation is done to check for errors. If an error is detected, the error flag is set in the transmit control word so that a NACK can be sent to the remote receiver. The controller then sends data out through the serial port on request to the external device (if there was no error). On the transmit side, data fed from the external device is CRC coded and a data frame is constructed. The controller reads the source address, destination address and the size of the data frame from the dip switches. The data frame is then sent to the transmitter circuits through the data bus.

All the input/output devices are memory mapped. The memory map is shown in figure 9 below. A gap has been left between the end of ROM and the beginning of RAM and also between the end of the RAM and the other devices. This is to simplify the address decoding logic.

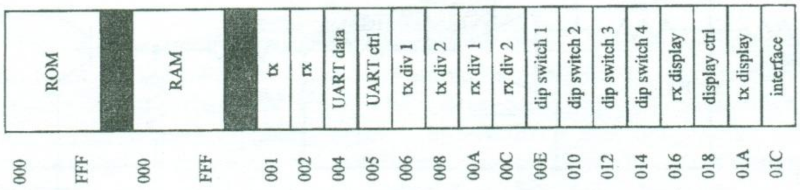


Figure 9 : Memory map of controller

Memory and Device Addressing.

The controller is designed with minimum addressing logic, and the unused memory addresses have to be avoided in the software. Address line 13 is used to disable the ROM and to enable the RAM and address line 14 (A14) is used to disable the RAM when addressing the other devices. These devices are selected using a 4-16 decoder which is activated when A14 goes high and address line 1 to 4 are used to identify the individual devices. Address line 0 is used to select the control/data mode of the UART while it is selected.

Interrupt Controller

The Z80 is used in its default interrupt handling mode. When an interrupt occurs, the CPU responds by activating the IORQ and MI signals and takes the next instruction form the data bus. The interrupt controller puts the ***restart at address P*** instruction onto the data bus when the CPU acknowledges the interrupt, to vector the CPU to the required ISR. This instruction is a single byte with the LSB three bits and MSB two bits high and other three bits containing a number from 000_b to 111_b specifying the location to restart to. The encoder sets these three bits depending on the interrupting device. This is illustrated in the schematic diagram in figure 10. There are seven interrupts and they are connected to the priority encoder according to their priority. All the interrupts are active low. The 7 interrupts are given below

rx intr - activated when a byte is received.
tx intr - activated when the tx buffer is empty.
rx ready - activated by UART when a byte received and available.
rx ready and DSR - indicates a presence of a command.
tx ready - activated by UART when tx buffer is empty.
disp intr - activated periodically to update the display. Since the *disp intr* is a symmetrical square wave, to avoid the interrupt

being active throughout the half cycle, dispCtrl select is used to disable the interrupt as it is serviced.

tx tune, rx tune, scan - these three together form the interface interrupt.

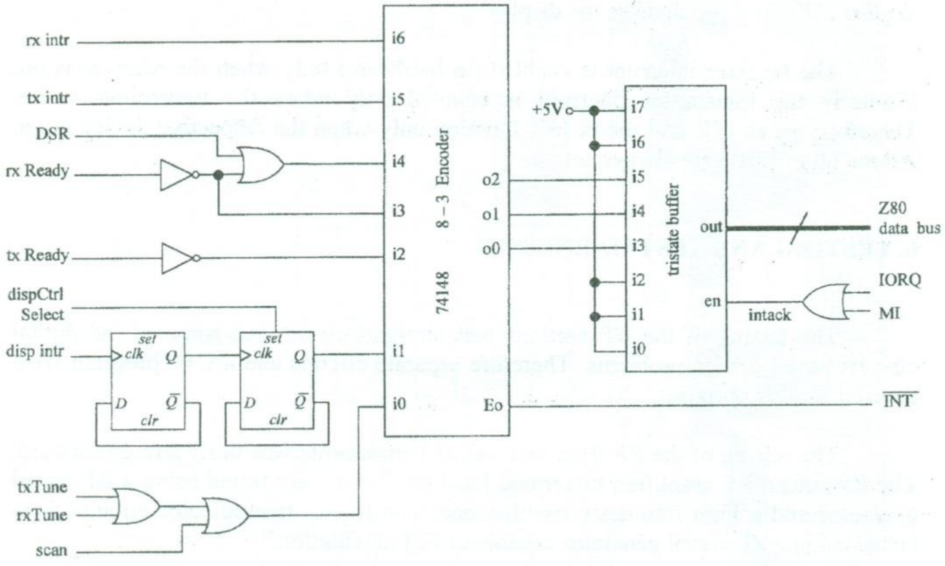


Figure 10 : Schematic diagram of the interrupt controller.

5. CONTROLLER SOFTWARE

The controller for this system was designed to cope with both the transmitter and receiver. All the processing required to maintain the data link is done by the controller's software. The software was written in Z80 assembly language, and is divided into 8 sections – the main program and 7 interrupt service routines. The main program does most of the data processing required of the controller, does overall system control and contains the data transfer loop (which cycles between transmitting, receiving and data processing). The data processing involves decoding and using he received control word, CRC decoding, processing errors, CRC encoding of the transmit data, and preparing the transmit header and data frame. The CRC encoding / decoding and transceiver tuning are done by two separate procedure. The interrupt service routines do the following distinct tasks.

rx ISR – checks for correct addresses, decodes control word and saves it to memory, and writes received data to the receive buffer.

tx ISR – writes a byte to the transmitter from the header or the transmit buffer as required.

data out ISR – writes the current system status to UART if the status has currently changed. Otherwise writes a data byte to the UART from the receive buffer.

data in ISR – reads a data byte from the UART to transmit buffer.
command in ISR – reads a command from the UART and set the relevant flags.
interface ISR – handles input from the user interface and sets the relevant variables.
display ISR – updates the display

The receiver interrupt is enabled (in hardware) only when the receiver is on. Similarly the transmitter interrupt is enabled only when the transmitter is on. Therefore the rx ISR and the tx ISR function only when the respective device is on. All the other ISR's are always active.

6. TESTING AND TEST HARNESSES

The testing of the RF sections was straightforward, but some of the digital circuits posed certain problems. Therefore separate circuits and a C++ program were devised for this purpose.

The testing of the RF front end and its components was fairly straightforward. The RF filters, RF amplifier, mixer and local oscillator were tested using a RF signal generator and a high frequency oscilloscope. The IF and demodulator circuits were tested using a RF signal generator capable of FM modulation.

For the proper testing of the clock recovery circuits, a reasonably random, Manchester coded data stream was required. Therefore a pseudo random bit stream was generated using a 7-bit maximal length feedback shift register pseudo random bit sequence (PRBS) generator. This bit stream was then Manchester coded and used to test the clock recovery circuits.

The hardware of the controller was tested section by section using short pieces of assembly code. Code was written to test the interrupt controller and device select logic, and the UART and the circuits handling serial communications.

The software for the control module was simulated, fully tested and debugged using a Z80 simulator. The software consists of the main code and 7 interrupt service routines. Four of these ISR's and part of the main code handle the data processing required for the operation of the data link with the remote transceiver. The testing of the other 3 ISR's and the rest of the main code was straightforward since they could be tested on a single standalone receiver. But the testing of the control module's data handling procedures is complicated by the requirement of an actual data frame containing valid sync character, destination address, control word, and CRC encoded data (with and without errors). This problem was overcome by designing a limited "transceiver emulator" using a C program for DOS and an interfacing circuit to link the controller to the parallel port of a PC.

This test harness connects directly to the data bus, the relevant interrupt lines and device select lines of the controller at the point where the transmitter/receiver digital sections are supposed to connect. It therefore tests only the controller and not the complete system. The transmitter/receiver digital sections have to be tested separately. The program, through the parallel port sends/receives data, monitors the transmitter/receiver select lines and activates the required interrupt lines. It also connects to the controller through the serial port and functions as the external device. Then the program can test the controller by sending serial data to it and monitoring the transmitted data. Similarly it can send out data through the parallel port simulating received data and check the data that the controller sends it through the serial port. The schematic diagram of the interfacing circuit is shown in figure 11.

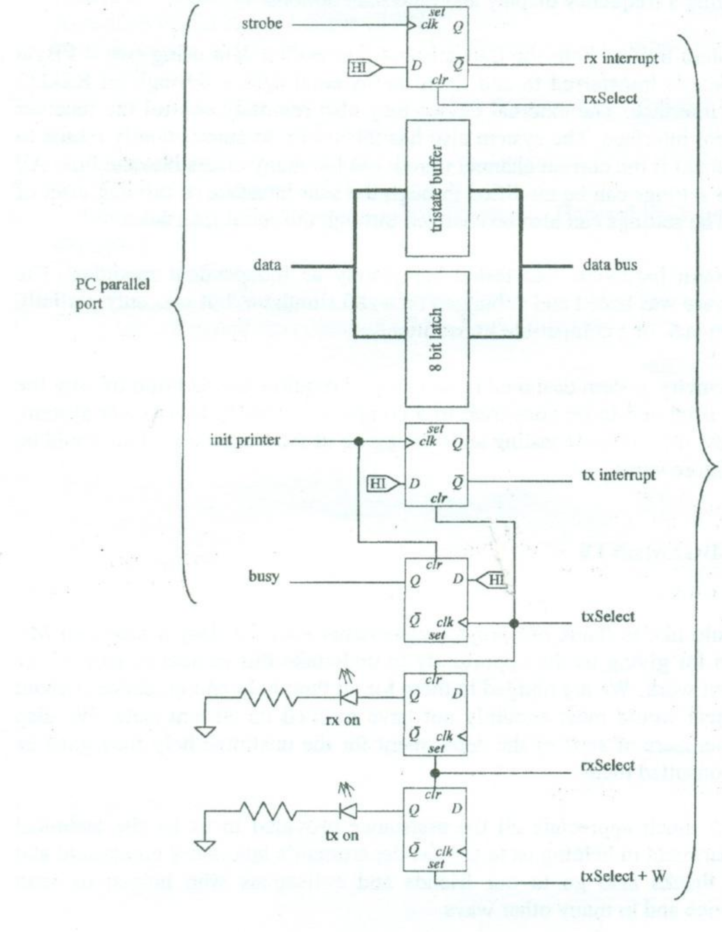


Figure 11 : Parallel port interfacing circuit

7. CONCLUSION

The radio telemetry receiver designed and constructed in this project meets all the required specifications and also includes some extra features. It is a narrow band tuneable radio telemetry receiver capable of operating over the license free bands of 26.9 – 27.3 MHz and 46.0 – 50.0 MHz. It supports a data rate of 9,600 baud, and has link control, flow control and backward error correction capability. The receiver has a PLL frequency synthesised local oscillator based on a varactor diode tuned VCO, which allows accurate and stable digital tuning. Tuning is facilitated through a user interface including a frequency display and tune/scan buttons.

The system buffers both the transmit and the receive data using two 4 kByte buffers. This data is transferred to and from the external device through an RS-232 standard serial interface. The external device may also remotely control the receiver through the serial interface. The system also has the ability to automatically retune to a noise free channel if the current channel introduces too many errors into the link. All relevant system settings can be modified through the user interface or through a set of DIP switches. The settings can also be changed through the serial interface.

The system hardware was tested completely as independent modules. The controller software was tested and debugged on a z80 simulator, but was only partially tested due to the lack of a compatible transmitter.

The telemetry system designed in this project requires the addition of only the transmitter RF front end to be converted to a complete telemetry transceiver system. This would allow the complete testing and debugging of the transceiver. This could be the subject of future work.

ACKNOWLEDGEMENTS

We would like to thank our project supervisors Prof. I.J. Dayawanse and Mr. B.S. Samarasiri for giving us the opportunity to undertake this project as part of our final year project work. We are obliged to them for all their help and guidance without which this project would most certainly not have reached its current state. We also thank all the members of staff of the department for the unstinted help they gave us whenever we consulted them.

We very much appreciate all the assistance provided to us by the technical staff of the department in helping us to use the department's laboratory equipment and facilities. Our thanks also go to our friends and colleagues who helped us with references, advice and in many other ways.

BACKGROUND READING

1. Dennis Roddy and John Coolen, *Electronic Communications,* Prentice Hall, 4th edition, 1995
2. Nicholas Maslin, *HF Communications, A Systems Approach,* Pitman Publishing, 1987.
3. A Bruce Carlson, *Communication Systems,* McGraw-Hill international editions, 3rd edition, 1986.
4. Anatol I Zverev, *Handbook of Filter Synthesis*, John Wiley and Sons Inc. 1967.
5. American Radio Relay League, *The Radio Amateur's Handbook,* American Radio Relay League , 1983.
6. Paul Horowitz, Winfield Hill, *The Art of Electronics,* Cambridge University Press, 2nd edition, 1995.
7. National Semiconductor Corp, *Digital PLL Synthesis,* National Semiconductor Corp, Application note 335, April 1983.
8. Kathe Spracklen, *Z80 and 8080 Assembly Language Programming,* Hayden Book Company, 1991.
9. William Barden, *Z80 Microcomputer Handbook,* Harvard W. Sams and Company,
   1st edition, 1987.
10. E.A. Parr, *Logic Designer's Handbook,* Granada, 1984.
11. Zilog Inc., *Product Specifications Data Book,* Zilog Inc, 1995.