The Design and Development of a Terminal Node Controller and GPS/Telemetry Beacon for Space and Ground Applications

James E. Gutshall, Anuradha Chandrasekaran

Advisor: Dr. Charles M. Swenson

Dept. of Electrical and Computer Engineering, 4120 Old Main Hill,
Utah State University, Logan, UT 84322-4120, USA.
Emails: jeg@cc.usu.edu, slfzl@cc.usu.edu, charles.swenson@ece.usu.edu

Abstract

This document presents the design and development of a new terminal node controller (TNC) and GPS/telemetry beacon for space and ground applications. The techniques developed are unique and innovative in the field of space communication. It outlines the objectives; the system requirements; the system layout; the theory of operation; software; communication protocols, including equipment specifications and other major aspects of the system. An important feature of this system is the extensive use of commercial off the shelf (COTS) components, which has lowered cost and shortened development time.

Key words: terminal node controller, TNC, GPS, telemetry, beacon, space communications, satellite, nanosatellite, transmitter, receiver, transceiver, PCM page, APRS, tracking device, spacecraft, packet assembler, protocol decoder, PIC, amateur radio, Ham radio, microcontroller, digipeater, AX.25, NMEA, Engineering.

INTRODUCTION

The frequency spectrum is fast becoming a scarce resource. As a result, there is a critical need to conserve this asset. One way of achieving this objective is to allocate the same frequency to more subscribers. This can be done through the use of efficient protocols and the use of a TNC [1],[2]. One of these protocols, the AX.25[1], was developed in 1979. In the 1980's, the automatic position reporting system (APRS)[2] was born. APRS (R)[3] allows organizations and individuals to track automobiles, trucks, bicycles, runners, and now, even spacecraft. Since its inception in 1999, the Utah State University (USU) communication team decided to fly a GPS/telemetry beacon and a TNC on the USU satellite (USUSAT). Since pulse code modulation (PCM) pages will be used for downlinking data, it was decided that a TNC that could meet the needs of the uplink should be built. Several TNCs had been designed (as shown in figure 1) and one was chosen for further development as it could withstand harsh environment and temperature constraints. Inexpensive microcontrollers along with a modem [3] integrated circuit (IC) were used in the design. Through the use of software, many options became available to the users. Other microcontrollers had proven costlier and consumed more power. The requirement to keep the cost down, along with the need to lower both mass and power for remote applications, became a factor for choosing the components needed for this design layout. Two separate microcontrollers were used; one for the receive path, and the other for the transmit path. They are interfaced in such a manner that there is no interference between the software programs. This design, thus laid out, can be used as a transmit TNC, a receive TNC, a telemetry encoder and decoder, and a GPS tracking device. This will become a very useful system for the small satellite community. Two other universities along with USU, are flying satellites in the

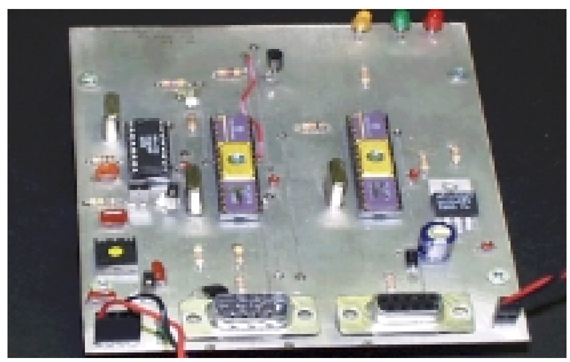

Figure 1: Prototype Development Test Board

[1]http://www.tapr.org/tapr/html/Fax25.html
[2]http://www.tapr.org/tapr/html/Faprswg.html
[3](R) registered to Bob Bruninga

constellation. Provisions have been made to enable the entire ION-F (Ionospheric Observation Nanosatellite Formation) constellation to fly this unique piece of equipment.

SYSTEM REQUIREMENTS

The mission objective of the TNC and GPS/telemetry beacon is to decode and forward correct commands, location, velocity, and telemetry information to all intended recepients with minimal interference and cost. With a view to satisfy the mission objective, the following requirements needed to be met:

• The device would interface with GPS receivers.
• The device would operate at a baud rate of 1200 so that it could be built and tested within the specified time frame. This also allowed the device to be compatible with the standards used by Ham radio operators around the world. Further enhancement could be made to increase the baud rate up to 9600.
• It would demonstrate an autonomous spacecraft tracking system.
• It must withstand a temperature range of -40 degrees to 85 degrees C.
• It must be capable of enduring a vibration of 12 gs along the X, Y, and Z axis.
• It must have a mass of less than 500 gms when combined with the transmitter.
• It must meet survivability and safety requirements.

SYSTEM LAYOUT

A block diagram for the system layout of the TNC and GPS/telemetry beacon is shown in figure 2. A detailed description of each of the blocks is given below:

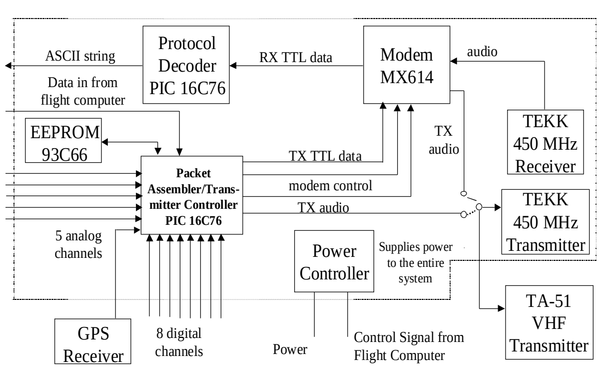

Figure 2: Block Diagram of the TNC/Telemetry Beacon

1. **TEKK KS960-L Receiver**
   The receiver picks up the AX.25 packets in the form of audio tones at 1200 and 2200 Hz and passes them on to the modem for further processing. The TNC and GPS/telemetry beacon module will be contained within the TEKK transceiver enclosure.

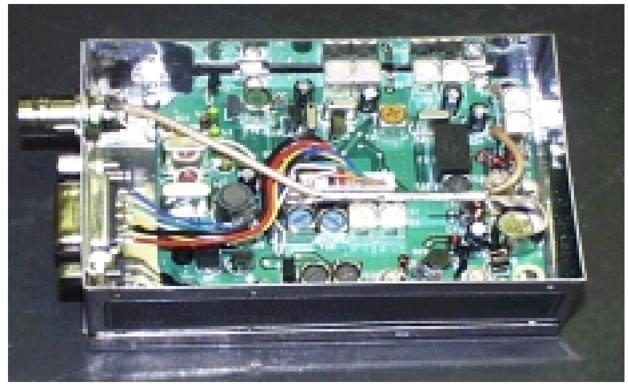

Figure 3: The TEKK KS-960L Transceiver

2. **Modem MX614**
   The modem converts the audio tones into AX.25 transistor transistor logic (TTL) data and passes it to the protocol decoder when it is in the receive mode. In the transmit mode, it receives AX.25 TTL data and converts it into audio tones at 1200 and 2200 Hz and sends it to the transmitter. The M0 and M1 lines are used for switching the modem from receive to transmit mode.

3. **Protocol Decoder**
   The protocol decoder converts the AX.25 TTL data into an ASCII string and sends it to the flight computer or to a ground station computer.

4. **Packet Assembler/Transmitter Controller**
   This device takes in the stored data from the flight computer and converts it into AX.25 packets. These AX.25 TTL packets are sent to the modem where they are converted into audio tones and sent to the transmitter. In the digipeater mode, the packet assembler [4],[5] forwards the AX.25 packets, which were received by the protocol decoder to the modem for retransmission. In the GPS mode, it takes in National Marine Electronic Association (NMEA) sentences at RS232-C levels from a GPS receiver, packetizes them, and sends them to the modem. The modem then converts them into audio tones and passes them on to the transmitter. In the GPS/telemetry beacon mode, the packet assembler combines the NMEA sentences from the GPS receiver with the values from the five analog and eight digital channels. It converts them into AX.25 audio tones and sends them to the transmitter.

5. **The GPS Receiver**
   The GPS receiver acquires signals from the GPS

satellite constellation and calculates its location. It then sends this information to the packet assembler in the form of NMEA sentences where it is packetized.

6. EEPROM 93C66
This is used for storing user selected variables such as station call signs, transmitter delay-time, icons that are used for mapping programs, and the rate at which the telemetry data should be sampled.

7. TEKK KS960L Transmitter and Hamtronics TA-51 VHF Transmitter
These transmitters receive the AX.25 audio tones from the modem or directly from the transmitter controller [6] in the GPS/telemetry beacon mode. The USUSAT is emloying a TA-51 VHF transmitter on another frequency to perform the same fuction.

8. Power Controller
This is used for supplying power to the system.

THEORY OF OPERATION

There are several modes inwhich the TNC and GPS/telemetry beacon can be used. The TNC and GPS/telemetry beacon allows data to be packetized with a bit error rate (BER) of 10^{-12} without consuming any of the on-board flight computer time. This is described below. Refer to the schematic, figure 4, in the appendix.

- TNC receive mode
- TNC transmit mode
- Digipeater mode
- GPS beacon mode
- GPS/telemetry beacon mode

TNC Receive mode

The signal is first picked up by a receiver. For the ION-F program, the receiver is a TEKK KS960-L tuned to a frequency of 450 MHz. The signal is a non return to zero - inverted (NRZ-I), 1200 baud, Bell 202 standard, AX.25 packet frame. The receiver amplifies and demodulates the signal. The audio baseband signal enters at (P3) pin 4 and is then sent through a direct current (DC) blocking capacitor (C5.) It is next passed through a current limiting resistor (R7) and then goes to (U2) pin 5, RXIN on the modem chip. This chip is clocked with a crystal on pins 1 and 2. The demodulated TTL signal comes out of pin 13 of the modem chip and goes to pin 25 (RB4) on the PIC (U1) [7]. This microcontroller is a protocol decoder and is clocked with a crystal connected between pins 9 and 10. The TTL AX.25 signal is converted into an ASCII string by stripping the start flags, decoding the address and relay information, checking the control field and protocol ID, removing the bit stuffing inserted during transmission time, checking the accuracy using frame check sequence (FCS) [8], and discarding the ending flags. After at least two valid flags are received, a signal is sent from pin 21. In the ground base model, a light emitting diode (LED) glows to indicate that the AX.25 packets are being decoded. In the space version, this signal, "Get Ready," tells the flight computer that a packet will be arriving momentarily. This ASCII string is then sent to the flight computer for further processing. The modem modes, transmit and receive, are controlled by the transmitter controller. Another signal on (U2) pin 14 provides a channel busy signal. This signal is fed through a resistor (R6) and a transistor (Q1), which simply inverts the signal and forwards it to pin 6 on the transmitter controller (U3). This acts as a form of carrier detect or channel busy for U3.

TNC Transmit mode

Data enters in (P1) pin 3 at an RS232-C level. It passes through a current limiting resistor (R2.) This causes the voltage to drop to a TTL level. This TTL signal is then fed across JMP3 to (RB6) pin 27 on U3. Within this microcontroller, the software adds information to convert the data into an AX.25 packet. This is done by adding start flags, addressing, and relay information along with a control field, protocol ID, and a frame check sequence. The packet is completed by adding the ending flags. The transmitter controller then checks for the carrier detect on pin 6. Once it detects the channel is clear, RB1 and RB2 (pins 22 and 23 respectively) change status to put the modem (U2) into transmit mode. Once the modem is in transmit mode, the transmitter controller keys the transmit control line by switching RC0, pin 11, to high. This turns Q3 on, which provides the ground to the key line of the transmitter through (P3) pin 2. The PIC waits for the transmitter delay period of about 20 ms (this is user selectable) to allow the transmitter to come to full power. At the same time it is sending flags from RB5 (pin 26) to the transmit data line (pin 11) on the modem. This time delay allows the transmitter to be completely powered up, and the receivers on the channel to lock onto the signal. The data is then sent to the modem (U2) where it is converted into tones and then sent out through pin 7. Next, it goes across JMP1 through R9 and C4, which matches the impedance and filters the signal. It then proceeds through the variable resistor R10. This is where the amplitude and deviation are set. The signal is forwarded through a DC blocking capacitor (C6) into the input circuit of the transmitter by way of (P3) pin 3. After all the data has been sent, the transmitter is keyed off and M0, M1 are set back to receive mode.

Digipeater mode

This is similar to the TNC transmit and TNC receive mode. After decoding the received AX.25 packets, they are passed to the packet assembler where the relay information is altered. Then they are forwarded to the modem for retransmission. A protocol translation can also be made with software changes.

GPS Beacon Mode

A NMEA[4] GPRMC string at 4800 baud enters (P1) pin 3 as an RS-232C signal. It passes through R2, which is a current limiting resistor and causes the voltage to fall off to a TTL level. It crosses JMP3 and enters the packet assembler (U3) at pin 27 (RB6.) The packet assembler is clocked by a crystal on pins 9 and 10. The string is read into file registers and is decoded and reassembled into a compressed position packet. U3 then checks pin 6 to see if the channel is clear. When that has been accomplished, M0 and M1 are switched to transmit mode. It then quickly keys the transmit control line by switching RC0 (pin 11) to high. This controls Q3 and allows it to turn on. It also provides the ground to the key line of the transmitter. The transmitter controller then waits about 20 ms for the delay to allow the transmitter to come to full power while sending flags from RB5 (pin 26) to the transmit data line (pin 11) on the modem. This time delay allows the transmitter to be completely powered up and the receivers on the channel to lock onto the signal. The data is then sent to the modem (U2) where it is converted into tones and sent through pin 7. It goes across JMP1 through R9 and C4, which helps to set impedance and filtering, and then through the variable resistor R10. This is where the amplitude and deviation are set. The signal is forwarded through a DC blocking capacitor (C6) into the input circuit of the transmitter by way of (P3) pin 3. After all the data has been sent, the transmitter is keyed off and M0, M1 are set back to receive mode.

GPS/Telemetry Beacon Mode

The NMEA GPRMC, GPGGA, and GPGSV sentences at RS-232-C levels enter through (P2) pin 2. The signals then pass through resistor R12 to an NPN transistor (Q4), which inverts them. It then enters U3, the packet assembler, at pin 18. This information is then combined with values from five analog and eight digital channels, pins ANALOG-1 through ANALOG-5 and BIT-0 through BIT-7. The microcontroller is clocked by a crystal connected between pins 9 and 10. User selected variables are also stored in U4 - a serial EEPROM 93C66. The software then checks for a carrier detect on U3, pin 6. It also keys the transmit control line by switching RC0 (pin 11) to high. This controls Q3 and allows it to turn on, which provides the ground to the key line of the transmitter. The transmitter controller then waits about 20 ms to allow the transmitter to come to full power while sending flags from RC1 (pin 12) across JMP2. It then passes through resistor R9 and C4, which acts as a shaping and filtering circuit. R10 sets the amplitude and deviation, and the signal is forwarded through capacitor C6 into the input circuit of the transmitter. After 20 ms, the microcontroller sends the data packet using the same path. The key is then released and the cycle is repeated at a user defined rate.

[4] http://www4.coastalnet.com/nmea/0183.htm

SOFTWARE

The software has been written in modular fashion and integrated together. The software for the protocol decoder is independent from that of the packet assembler. The program code has been written in assembly language using Integrated Design Environment (IDE) in Microchips MP-LAB (version 4.1 and version 5.1.) It was then compiled into HEX code. The program code has been loaded into the PIC microcontrollers using the Microchips PICSTART PLUS programmer. The PIC microcontrollers used for the prototype design are the EPROM version of the chips. This is to facilitate reprogramming and code modifications. The flight model will use one time programmable PIC microcontrollers. The entire software is structured in a manner as to enhance ease of operation, maintenance, remote sharing, and code update. Shareware software, such as WinAPRS and DosAPRS, along with UI-View, will be responsible for decoding and linking the information from the GPS/telemetry beacon to the Internet. DosAPRS provides a quantitative interpretation of the received data while WinAPRS shows a graphical representation of the decoded results. Plans are under way to write additional software to make telemetry decoding easier and more user friendly.

COMMUNICATION PROTOCOLS

The TNC and GPS/telemetry beacon uses AX.25 and APRS (R) protocols. The AX.25 Link-Layer Protocol provides a reliable data transport mechanism between two signaling stations. It supports different kinds of frames and functions equally well in either half or full duplex mode. The basic frame structure used by the AX.25 protocol is shown in figure 5. The APRS (R) protocol uses the AX.25 unconnected information (UI) frame format at a link level and facilitates a fast dissemination of data. This frame contains nine fields of

Flags	Destination Address	Source Address	Digipeater Address	Control Field	Protocol ID	Information	FCS	Flags
1	7	7	0-56	1	1	1-256	2	1

Figure 5: AX.25 UI-Frame Construction

T	Sequence No #xxx,	Analog Value 1 aaa,	Analog Value 1 aaa,	Analog Value 1 aaa,	Analog Value 1 aaa,	Analog Value 1 aaa,	Digital Value bbbbbbbb	Comment
1	5	4	4	4	4	4	8	n

Example:

N7VHF-1>BEACON:T#007,126,167,010,134,083,01010101

Figure 6: Telemetry Report Format

data as shown in figure 6. This telemetry report format of the APRS (R) protocol is used for monitoring the satellite health status. The USUSAT sends this telemetry data to the intended recepients around the world. The telemetry report format is contained within the information field of the AX.25. It starts with T# and a three digit frame sequence number 000-999 and then a comma followed by five analog values, each represented by three digits and a comma. It also contains eight digital values, each denoted by one digit. The analog values are expressed as integer numbers ranging from 000-255, and the digital values are represented by 1 or 0. As an example, the USUSAT has the following five analog and eight digital channels:

A0 – base plate temperature
A1 – unregulated bus voltage
A2 – battery temperature
A3 – DC probe
A4 – rate gyro
D0-D5 – flight computer status
D6 – crosslink carrier detect
D7 – UHF carrier data

The interpretation of this telemetry data can be changed at any time. Thousands of stations around the world will be able to interpret this data after they have received four APRS (R) messages indicating how the data is to be interpreted. This interpretation data will be sent over the Internet. The APRS (R) messages and their formats are listed below:

1. A Parameter Name message: This frame specifies the name of each of the 13 parameters. Here 'An' and 'Bn' are the parameter names. The structure of the frame is shown in figure 7.

PARM.	A1 N ...	A2 ,N ...	A3 ,N ...	A4 ,N ...	A5 ,N ...	B1 ,N ...	B2 ,N ...	B3 ,N ...	B4 ,N ...	B5 ,N ...	B6 ,N ...	B7 ,N ...	B8 ,N ...
5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

Example:

KD7KJA>N7VHF-1: PARM.Bplat,Unbus,Btemp,Dcpro,Rgyro,CS0, CS1,CS2,CS3,CS4,CS5,Xcd,Ucd

Figure 7: Telemetry Parameter Name Message Data

2. A Unit/Label message: In this frame, 'U' represents units for the analog channel and 'L' are the labels for the digital channels. The unit format describes what unit is to be displayed and what label is associated with each digital condition. The frame structure is shown in figure 8.

PARM.	A1 N ...	A2 ,N ...	A3 ,N ...	A4 ,N ...	A5 ,N ...	B1 ,N ...	B2 ,N ...	B3 ,N ...	B4 ,N ...	B5 ,N ...	B6 ,N ...	B7 ,N ...	B8 ,N ...
5	1-7	1-7	1-6	1-6	1-5	1-6	1-5	1-4	1-4	1-4	1-3	1-3	1-3

Example:

KD7KJA>N7VHF-1: PARM.Bplat,Unbus,Btemp,Dcpro,Rgyro,CS0, CS1,CS2,CS3,CS4,CS5,Xcd,Ucd

Figure 8: Telemetry Unit/Label Message Data

3. An Equation Coefficients message: The basic frame structure is shown in figure 9. a, b, and c contain three coefficients for each of the five analog channels. The final value of an analog channel is obtained by substituting the coefficients into a second degree equation: $y = a * x^2 + b * x + c$ where x is the raw transmitted value. For example, as shown in figure 9, the channel A2 depicts unregulated bus voltage, up to an accuracy of one hundredth of a volt, and a = 0, b = 6.2, c = 0. If the transmitted raw value 'x' is 167, then the voltage is calculated as:

$$\text{voltage} = 0 * 167^2 + 6.2 * 167 + 0 \quad (1)$$
$$= 1035.4 \text{ hundredths of a volt} \quad (2)$$
$$= 10.354 \text{ volts} \quad (3)$$

EQNS.	A1 a ,b ,c	A2 a ,b ,c	A3 a ,b ,c	A4 a ,b ,c	A5 a ,b ,c
5	1-7	1-7	1-6	1-6	1-5

Example:

KD7KJA>N7VHF-1EQNS:0,0.35,-45,0,6.2,0,0,0.04,0,2,4.56,47,16,3,5

Figure 9: Telemetry Equation Coefficients Message Data

4. A Bit Sense/Project Name message: The telemetry frame is shown in figure 10. The 'x' specifies the bit state that corresponds to the respective bit labels. It indicates whether the bit is active with a one or a zero.

The destination address field of the AX.25 protocol may contain the MIC-E encoded data. This data provides information such as the latitude of the station, a direction indicator, a message code, and an APRS (R) digipeater path in a compressed format. This data along with the data in the information field of

BITS.	B1 x	B2 x	B3 x	B4 x	B5 x	B6 x	B7 x	B8 x	Project Title
5	1	1	1	1	1	1	1	1	0-23

Example:

KD7KJA>N7VHF-1:BITS.10010011,USUSAT

Figure 10: Telemetry Bit Sense/Project Name Message Data

the AX.25 protocol is used to obtain a complete position report and other important information. The three NMEA sentences: GPGGA, GPGSV, GPRMC will be used to communicate between the GPS receiver and the packet assembler.

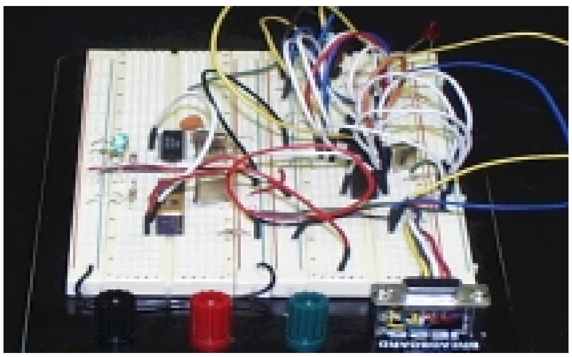

Figure 11: Current Prototype Development Board

SUMMARY

This document has presented the design and development of a TNC and GPS/telemetry beacon for space and ground applications. The need for this subsystem has been discussed. This TNC can easily be integrated with any computer. It will also provide valuable information and feedback to people around the world. This system not only posseses remote sensing and telemetry for space applications, but also has numerous ground applications like monitoring fleet vehicles, monitoring weather and river levels, and monitoring and controlling machinery, etc. There are countless applications for this system. This also allows multiple subcribers to share the same frequency and thus helps to conserve the spectrum. This system will be a low cost, useful tool for autonomous tracking and health monitoring of spacecraft. This technology can be looked upon as a unique strategy for future space missions. This TNC and GPS/telemetry beacon meets all system and safety requirements. The TNC and GPS/telemetry beacon design has been completed and prototype testing is proceeding. The flight unit will be ready by the end of August 2001.

ACKNOWLEDGEMENTS

Appreciation goes to all the members of the USU SatComm Team who have assisted in the testing and documentation of this project. Our sincere gratitude to Dr. Charles M. Swenson for giving us the opportunity to fly the TNC and GPS/telemetry beacon on the USUSAT. Thanks also for his constant support and encouragement in keeping this project alive admist all odds. A big thank you to Air Force Research Laboratory (AFRL) and the National Aeronautics and Space Administration (NASA) for their support. A heartfelt thanks to our family members for their continuous support and patience throughout this project.

References

[1] J. Hansen, "An inexpensive kiss-mode tnc," in QST, November 2000.

[2] P. K. Mike Chepponis, "The kiss tnc: A simple host-to-tnc communications protocol," in ARRL 6th Computer Networking Conference, (Redondo Beach CA), January 1987.

[3] Data Bulletin for the MX614. 4800 Bethania Station Road. Winston-Salem, NC 27105: MX-COM, INC., 1998.

[4] J. Hansen, "Pic-et radio: How to send ax.25 ui frames using inexpensive pic microprocessors,"

[5] J. Hansen, "Using pic microcontrollers in amateur radio projects," in QST, October 1998.

[6] J. D. Robstein, "d/a conversion using pwm and r − 2r ladders to generate sine and dtmf waveforms," in Microchip Application Note 655.

[7] The data sheets on PIC16C7X and PIC16F87X. 2355 W. Chandler Blvd. AZ 85224: Microchip Technology Inc., 1997.

[8] G. Morse, Calculating CRCs by Bits and Bytes. xxxx: BYTE, Sep 1986.

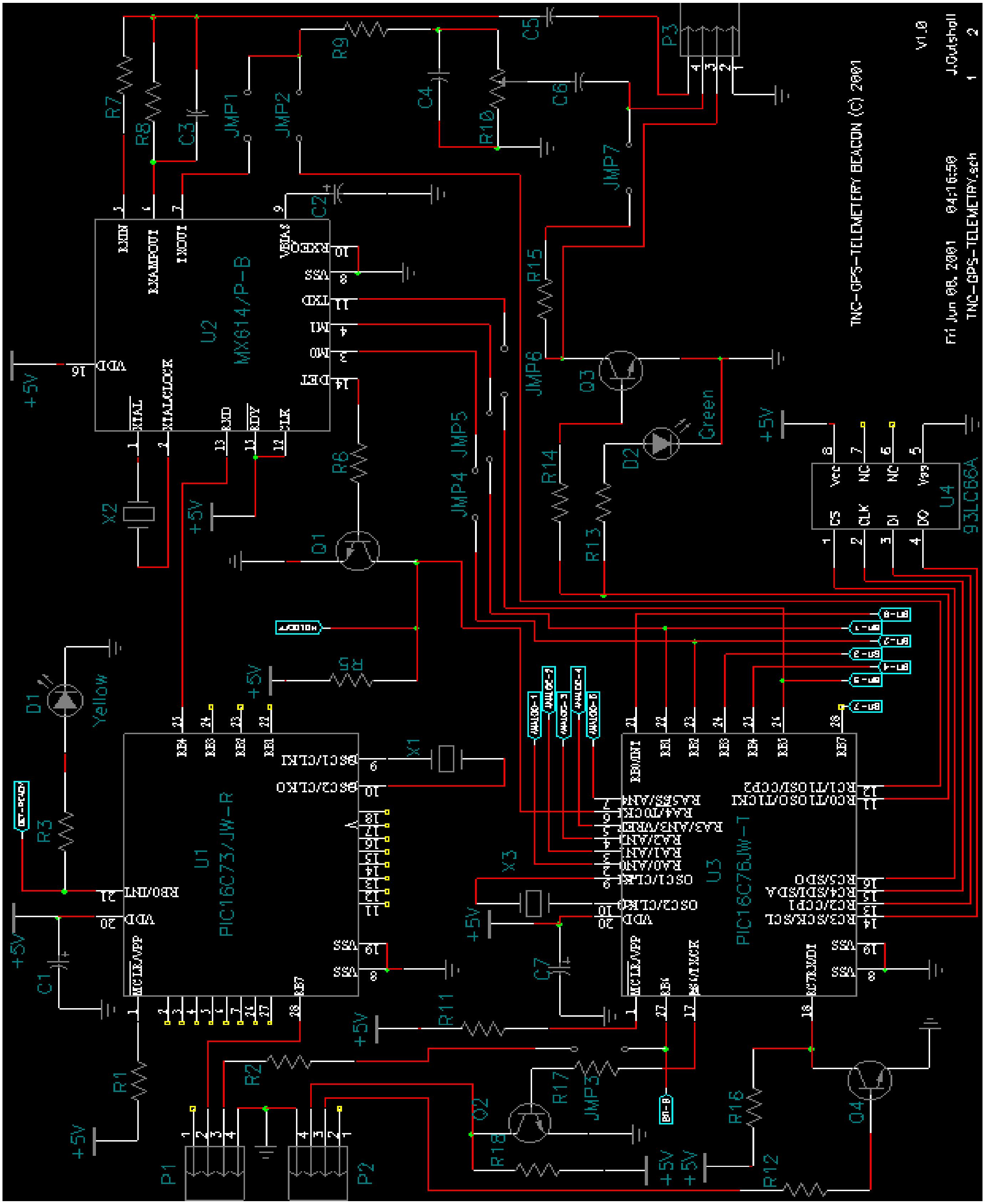

Figure 4: The Schematic for the TNC and GPS/Telemetry Beacon