

**DR1200/1201DK**

**Virtual Wire  
Development Kit  
Manual**

## **1 Virtual Wire® Development Kit Introduction**

- 1.1 Purpose of the Virtual Wire® Development Kit
- 1.2 Intended User
- 1.3 General Description
- 1.4 Key Features
- 1.5 Development Kit Contents

## **2 Low-Power Wireless Data Communications**

- 2.1 Operational Considerations
- 2.2 Regulations
- 2.3 Example Applications
- 2.4 FAQs

## **3 Developing a Virtual Wire® Application**

- 3.1 Simulating the Application
- 3.2 I/O and Power Considerations
- 3.3 Communications Protocol
- 3.4 Antenna Considerations
- 3.5 Internal Noise Management
- 3.6 Final Product Testing
- 3.7 Regulatory Certification
  - Regulatory Authority
  - Product Certification
  - Certification Testing

## **4 Installation and Operation**

- 4.1 Development Kit Assembly Instructions
- 4.2 Data Radio Board
  - Antenna Options

- 4.3 Protocol Board
  - Node Programming
  - RS232 Interface
  - LED Functions
- 4.4 Terminal Program
  - Installation
  - Configuration
  - Operation

## **5 Theory of Operation**

- 5.1 Data Radio Boards
  - I/O Interface
  - TR1000/TR1001 ASH Transceiver
  - Specifications
- 5.2 Protocol Board
  - I/O Interface
  - RS232 Interface
  - Protocol Microcontroller
  - CMOS/RS232 Level Converter
  - Specifications
- 5.3 Protocol Firmware
  - Description
  - Message Format
- 5.4 Terminal Program
  - Description
  - Source Code Listing

## **A Drawings**

ASH Receiver Block Diagram & Timing Cycle  
ASH Transceiver Block Diagram  
Antenna Mounting  
Node Programming  
DR1200 and DR1201 Data Radio Schematic  
DR1200 and DR1201 Data Radio Bill of Materials  
DR1200 and DR1201 Data Radio Component Placement  
916.5 MHz Test Antenna Drawing  
868.35 MHz Test Antenna Drawing  
PB1001 Protocol Board Schematic  
PB1001 Protocol Board Component Placement  
PB1001 Protocol Board Bill of Materials

## **Virtual Wire® Development Kit Hardware Warranty**

**Limited Hardware Warranty.** Murata Electronics, N.A., Inc. warrants solely to the purchaser that the hardware components of the Virtual Wire® Development Kit (the “Kit”) will be free from defects in materials and workmanship under normal use for a period of 90 days from the date of shipment by Murata. This limited warranty does not extend to any components which have been subjected to misuse, neglect, accident, or improper installation or application. RFM's entire liability and the purchaser's sole and exclusive remedy for the breach of this Limited Hardware Warranty shall be, at RFM's option, when accompanied by a valid receipt, either (i) repair or replacement of the defective components or (ii) upon return of the defective Kit, refund of the purchase price paid for the Kit. **EXCEPT FOR THE LIMITED HARDWARE WARRANTY SET FORTH ABOVE, MURATA AND ITS LICENSORS PROVIDE THE HARDWARE ON AN AS IS BASIS, AND WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.** Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

**Limitation of Liability.** **IN NO EVENT SHALL MURATA OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES (WHETHER SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHERWISE) IN EXCESS OF THE PRICE ACTUALLY PAID BY YOU TO MURATA FOR THE KIT, REGARDLESS OF UNDER WHAT LEGAL THEORY, TORT, OR CONTRACT SUCH DAMAGES MAY BE ALLEGED (INCLUDING, WITHOUT LIMITATION, ANY CLAIMS, DAMAGES, OR LIABILITIES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR FOR INJURY TO PERSON OR PROPERTY) ARISING OUT OF THE USE OR INABILITY TO USE THE KIT, EVEN IF MURATA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

### **Special notice on restricted use of Virtual Wire® Development Kits**

Virtual Wire® Development Kits are intended for use solely by professional engineers for the purpose of evaluating the feasibility of low-power wireless data communications applications. The user's evaluation must be limited to use of an assembled Kit within a laboratory setting which provides for adequate shielding of RF emission which might be caused by operation of the Kit following assembly. In field testing, the assembled device must not be operated in a residential area or any area where radio devices might be subject to harmful electrical interference. This Kit has not been certified for use by the FCC in accord with Part 15, or to ETSI I-ETS 300 220 or I-ETS 300 220-1 regulations, or other known standards of operation governing radio emissions. Distribution and sale of the Kit is intended solely for use in future development

of devices which may be subject to FCC regulation, or other authorities governing radio emission. This Kit may not be resold by users for any purpose. Accordingly, operation of the Kit in the development of future devices is deemed within the discretion of the user and the user shall have all responsibility for any compliance with any FCC regulation or other authority governing radio emission of such development or use, including without limitation reducing electrical interference to legally acceptable levels. All products developed by user must be approved by the FCC or other authority governing radio emission prior to marketing or sale of such products and user bears all responsibility for obtaining the FCC's prior approval, or approval as needed from any other authority governing radio emission.

If user has obtained the Kit for any purpose not identified above, including all conditions of assembly and use, user should return Kit to RF Monolithics, Inc. immediately.

The Kit is an experimental device, and Murata makes no representation with respect to the adequacy of the Kit in developing low-power wireless data communications applications or systems, nor for the adequacy of such design or result. Murata does not and cannot warrant that the functioning of the Kit will be uninterrupted or error-free.

The Kit and products based on the technology in the Kit operate on shared radio channels. Radio interference can occur in any place at any time, and thus the communications link may not be absolutely reliable. Products using Virtual Wire® technology must be designed so that a loss of communications due to radio interference or otherwise will not endanger either people or property, and will not cause the loss of valuable data. Murata assumes no liability for the performance of products which are designed or created using the Kit. **Murata products are not suitable for use in life-support applications, biological hazard applications, nuclear control applications, or radioactive areas.**

Murata and Virtual Wire are registered trademarks of Murata Electronics N.A., Inc. MS-DOS, QuickBASIC, and Windows are registered trademarks of Microsoft Corporation.

# **1 Virtual Wire® Development Kit Introduction**

## **1.1 Purpose of the Virtual Wire® Development Kit**

The Virtual Wire® Development Kit is a tool for evaluating the feasibility of a low-power wireless data communications application. The kit also facilitates the development of the actual system. In addition, the modules in the kit are available from Murata for use in system manufacturing.

## **1.2 Intended Kit User**

The Virtual Wire® Development Kit is intended for use by a professional engineer with a working knowledge of data communications. The kit itself is not intended as an end product, or for use by individuals that do not have a professional background in data communications. Please refer to the Special Notices section in the front of this manual.

## **1.3 General Description**

The Virtual Wire® Development Kit allows the user to implement low-power wireless communications based on half-duplex packet transmissions. The kit contains the hardware and software needed to establish a wireless link between two DOS-based computers with RS232C serial ports. The kit includes two communications nodes, with each node consisting of a data radio board and host protocol board, plus accessories. The DR1200-DK kit operates at 916.5 MHz, and the DR1201-DK operates at 868.35 MHz.

## **1.4 Key Features**

The Virtual Wire® Development Kit includes a number of key features:

- “Out of the box” operation between two DOS-based PC's
- 3 Vdc low-current UHF data radio transceivers (916.5 MHz)
- Excellent receiver off-channel interference rejection
- Wide dynamic range receiver log detection and AGC for resistance to on-channel interference
- Reference antennas
- 4.5 Vdc low-current protocol boards based on an ATMEL AT892051 microcontroller
- On-board CMOS logic to RS232C level conversion with bypass provisions for direct CMOS logic interface
- Packet link-layer protocol with ISO 3309 error detection and automatic packet retransmission; up to 32 message bytes per packet transmission (ASCII or binary)
- DC-balanced data coding for robust RF transmission performance
- Simple packet protocol to application layer interface & example application software
- Diagnostic LEDs for system performance evaluation
- Up to 15 different node addresses supported; jumper programmable

## **1.5 Development Kit Contents**

- Two data radio transceiver boards
- Two host protocol boards
- Two reference antennas
- 3.5" floppy disk with example application software
- Manual

## **2 Low-Power Wireless Applications**

### **2.1 Operational Considerations**

Low-power wireless (RF) systems typically transmit less than 1 mW of power, and operate over distances of 3 to 60 meters. Once certified to comply with local communications regulations, they do not require a license or "air-time fee" for operation. There are more than 60 million systems manufactured each year that utilize low-power wireless for security, control and data transmission. Many new applications for low-power wireless are emerging, and sales are expected to top 100 million systems per year by the end of the decade.

The classical uses for low-power wireless systems are one-way remote control and alarm links, including garage door openers, automotive "keyless entry" transmitters, and home security systems. Recently, a strong interest has developed in two-way data communications applications. These low-power wireless systems are used to eliminate nuisance cables on all types of digital products, much as cordless phones have eliminated cumbersome phone wires. RFM's Virtual Wire® Development Kits are intended to support the design of these types of low-power wireless applications. Most low-power wireless systems operate with few interference problems. However, these systems operate on shared radio channels, so interference can occur at any place and at any time.

Products that incorporate low-power wireless communications must be designed so that a loss of communication due to radio interference or any other reason does not create a dangerous situation, damage equipment or property, or cause loss of valuable data.

## **2.2 Regulations**

While low-power wireless products do not have to be individually licensed, they are subject to regulation. Before low-power wireless systems can be marketed in most countries, they must be certified to comply with specific technical regulations. In the US, the FCC issues this certification. In most of Europe and Scandinavia, certification is based on ETSI standards, and administered by the PTTs.

While technical regulations vary from country to country, they follow the same general philosophy of assuring that low-power wireless systems will not significantly interfere with licensed radio systems. Regulations specify limitations on fundamental power, harmonic and spurious emission levels, transmitter frequency stability, and transmission bandwidth.

## **2.3 Example Applications**

Applications for low-power wireless data communications are growing very rapidly. The following list of example applications demonstrates the diversity of uses for low-power wireless technology:

- Wireless bar-code readers and bar-code label printers
- Smart ID tags for inventory tracking and identification
- Wireless automatic utility meter reading systems
- Wireless credit card readers and receipt printers for car rentals, restaurants, etc.
- Communications links for hand-held terminals, HPCs, PDAs, and peripherals
- Portable and field data logging
- Location tracking (follow-me phone extensions, etc.)
- Sports telemetry
- Surveying system data links
- Engine diagnostic links
- Polled wireless security alarm sensors



- Authentication and access control tags
- Arcade games

## 2.4 FAQs

### 1. *Why does the Virtual Wire® Development Kit include a packet protocol microcontroller? Why not connect the data radio board directly to a computer serial port?*

You can hook a data radio board directly to a computer serial port (using an RS232 to 3 V CMOS level converter). However, the results are not likely to be satisfactory. First, error detection is limited to byte parity checking, which will let many errors go undetected. Also, the DC balance in the data can be very poor, which will greatly reduce the data radio's range.

Packet protocol is used extensively in two-way data communications. For example, the Internet and digital cellular phones use packet transmissions. While there are many packet protocols in use, they all provide a basic set of features, including an effective means for transmission error detection, and routing support (such as a "to" and "from" address). This allows error free data communications to be performed in a highly automatic way. The protocol microcontroller used in the Virtual Wire® Development Kit provides error detection and automatic message retransmission, message routing, link failure alarms and DC-balanced packet coding.

### 2. *What is the operating range of my low-power wireless systems?*

In our tests in an electrically quiet outdoor location, we easily communicate 60 meters with the DR1200-DK and DR1201-DK. However, operating range in a given situation is influenced by building construction materials and contents when indoors, and by other radio systems operating in the vicinity, and noise generated by nearby equipment. The Virtual Wire® Development Kit can be taken into a target

environment and used to help gain a sense of operating range for the proposed system. See the Appendix in the *ASH Transceiver Designer's Guide* for additional information.

*3. Can I communicate between more than two nodes in the same location with a low-power communications link?*

Yes. One of the benefits of packet transmissions in channel sharing. In the case of the Virtual Wire® Development Kit, each protocol board can be programmed to have one of fifteen addresses, with address “0” reserved for messages that are broadcast to all nodes. For example, node 1 can be transmitting bar-code readings to node 2 while node 4 is transmitting bar-codes to node 7 in the same location. So long as the average channel usage is less than about 12%, randomly transmitted messages will get through without excessive transmission “collisions” and transmission retries.

### **3 Developing a Virtual Wire® Application**

#### **3.1 Simulating the Application**

There are hundreds of potential applications for short-range wireless communications links. Because there can be so many different variables in a potential application, simulating the application is often the best way to gain insight into its feasibility. Virtual Wire® Development Kits can be very helpful in simulating potential applications. The following simulation check list covers issues common to most low-power wireless applications. The user should also consider what other specific issues apply to the application being simulated:

- Maximum operating range required
- Type of operating environment (outdoor, indoor, indoor building construction, etc.)
- Number of nodes (transceivers) required in the application

- Node interaction (communications between pairs of nodes only, one master node and several slave nodes, communications between any two nodes, etc.)
- Possible on-channel interference/noise sources (ISM equipment, electrical equipment, nearby spread-spectrum systems, etc.)
- Channel usage (average and peak number of messages expected each minute, average message transmission/acknowledgment duration, average percentage of time the channel is in use, etc.)
- Message characteristics (average and maximum length; message type such as data, telemetry, control codes, etc.)
- Antenna logistics (omnidirectional, directional, hidden, etc.)
- Environmental considerations

Indoor radio propagation is an issue for special consideration. In most indoor locations, “dead spots” can be found where reception is very difficult. These can occur even if there appears to be a line of sight relationship between two nodes. These “dead spots”, or nulls, are due to multiple transmission paths existing between two locations because of reflections off metal objects such as steel beams, concrete rebar, metal door, window and ceiling tile frames, etc. Nulls occur when the path lengths effectively differ by an odd half-wavelength. Deep nulls are usually very localized, and can be avoided by moving either node slightly.

Diversity reception techniques are very helpful in reducing indoor null problems. Many low-power wireless systems involve communications between a master and multiple slave units. In this case, the master transmission can be sent twice; first from one master and then again from a second master in a slightly different location. The nulls for each master will tend to be in different locations, so a slave is very likely to hear the transmission from one or the other master. Likewise, a transmission from a slave is likely to be heard by at least one of the masters. Hand-held applications usually involve some movement, so automatic packet retransmission often succeeds in completing the transmission as hand motion moves the node through the null and back into a good transmission point.

## 3.2 I/O and Power Considerations

The DR1200 and DR1201 Data Radio boards require a DC power supply in the range of 2.7 to 3.3 Vdc with less than 10 mV of ripple, and a peak current capability of up to 15 mA. Quiescent current in the receive mode is approximately 5.5 mA with a 3 volt power supply. The average current with an RF signal being transmitted is approximately 6 mA and the peak current in the RF transmit mode is approximately 12 mA. Care must be taken to avoid reversing the polarity of the power supply since diode protection is not provided. Another concern is ESD as static-sensitive, devices are used on the Data Radio board. Note the Protocol Board operates from 4.5 Vdc.

## 3.3 Communications Protocol

Almost all two-way wireless data communications use some form of packet protocol to automatically assure information is received correctly at the correct destination. The protocol provided with the Virtual Wire® Development Kits is a *link-layer* protocol, and includes the following features:

- 16-bit ISO 3309 error detection calculation to test message integrity
- 4-bit TO/FROM address routing with 15 different node addresses available
- ASCII or binary message support, up to 32 bytes per packet
- Automatic packet retransmission until acknowledgment is received; 8 retries with semi-random back-off delays plus “acknowledge” and “link failure” alarm messages.

Also included with the Kits is a simple terminal program with source code to provide an example of interfacing host (application) software to the Virtual Wire® link layer protocol. Most users will develop specific host software to match the needs of their application. The protocol software does not require or support hardware flow control, so the host software will have to do some timekeeping to interface the protocol software. Study the source code listing and comments for the details of this interface. Users familiar with hardwired packet networks may consider the 32 message bytes per packet

limit quite small. Packets sent by low-power wireless systems are kept deliberately short to improve performance where on-channel burst interference and low signal-to-noise conditions are often encountered.

### **3.4 Antenna Considerations**

Suitable antennas are crucial to the success of a Virtual Wire® application. Here are several key points to consider in designing antennas for your application:

- Where possible, the antenna should be placed on the outside of the product. Also, try to place the antenna on the top of the product. If the product is “body worn”, try to get the antenna away from the body as far as practical.
- Regulatory agencies prefer antennas that are permanently fixed to the product. Antennas can be supplied with a cable, provided a non-standard connector is used to discourage antenna substitution (these connectors are often referred to as “Part 15” connectors).
- An antenna can not be placed inside a metal case, as the case will shield it. Also, some plastics (and coatings) significantly attenuate RF signals and these materials should not be used for product cases, if the antenna is going to be inside the case.
- The antenna designs used in the kit are included in the Drawings section of the manual. Many other antenna designs are possible, but efficient antenna development requires access to antenna test equipment such as a vector network analyzer, calibrated test antenna, antenna range, etc. Unless you have access to this type of equipment, the use of an antenna consultant is recommended.
- A patch or slot antenna can be used in some applications where an external antenna would be subject to damage. These types of antennas usually have to be designed on a case-by-case basis.

### 3.5 Internal Noise Management

RF transceivers operating under FCC "Part 15" rules are sensitive to noise in the passband of the receiver, because the desired transmitted signals are at very low power levels. Commonly encountered internal noise sources are microprocessors, both for control functions and computer functions; brush-type motors and high-speed logic circuits. If the rise time and fall time of the clock for a microprocessor are fast enough to produce harmonics in the frequency range of the receiver and the harmonics fall within the passband of the receiver, then special care must be taken to reduce the level of the harmonic at the antenna port of the receiver. If the engineer has the option, he should choose a microprocessor that has the slowest rise and fall time he can use for the application to avoid the troublesome harmonics in the UHF band. If possible, brush-type motors should be avoided, since arcing of the brushes on the commutator makes a very effective spark gap transmitter. If it is necessary to use a brush-type motor, spark suppression techniques should be used. Such motors can be purchased with spark suppression built-in. If the motor does not have built-in spark suppression, bypass capacitors, series resistors and shielding may have to be employed. High-speed logic circuits produce noise similar to microprocessors. Once again, the engineer should use logic with the slowest rise and fall times that will work for his application.

The items listed below should be considered for an application that has one or more of the above noise sources included. It may not be possible to follow all of these guidelines in a particular application.

- Locate the RF transceiver and its antenna as far from the noise source as possible.
- If the transceiver must be enclosed with the noise source, remotely locate the antenna using a coaxial cable.
- Terminate high speed logic circuits with their characteristic impedance and use microstrip interconnect lines designed for that impedance.

- Keep PCB traces and wires that carry high-speed logic signals or supply brush-type motors as short as possible. Such lines act as antennas that radiate the unwanted noise.
- If possible, enclose the noise source in a grounded metal box and use RF decoupling on the input/output lines.
- Avoid using the same power lines for the RF transceiver and the noise source or at least thoroughly filter (RF decouple) such power lines. It is advisable to use separate voltage regulators, if possible.
- If the antenna cannot be remotely located, place it as far from the noise source as possible (on the opposite end of the pc board). Orient the antenna such that its axis is in the same plane with the pc board containing the noise source. Do not run wires that supply the noise source in close proximity to the antenna.

### **3.6 Final Product Testing**

Any wireless data communications system must be thoroughly tested due to the “anything can happen in any sequence” nature of wireless communications. It is highly recommended that beta sites be chosen for operational system testing which represent the “limit” situations the system can be expected to operate in.

Testing for regulatory certification is discussed in Section 3.7. It is recommended that the user either establish an RF test range or a working relationship with a recognized test lab early in the system development phase, to allow for periodic evaluation of the system’s emissions during development. Many labs are experienced in solving radiation problems that cause certification test failures and/or jamming of the low-power wireless link. Identifying these types of problems early can save a lot of development time.

### 3.7 Regulatory Certification

*Regulatory Authority* - Worldwide, man-made electromagnetic emissions are controlled by international treaty and the ITU (International Telecommunications Union) committee recommendations. These treaties require countries within a geographical region to use comparable tables for channel allocations and emission limits, to assure that all users can operate with reasonable levels of interference.

Recognizing a need to protect their limited frequency resources, many countries have additional local laws, regulations and government decrees for acceptable emission levels from various electronic equipment, both military and commercial. By requiring that each model of equipment be tested and an authorization permit issued after payment of a tax (called a grant fee), the government attempts to control the sale of poor quality equipment and also has record of the known manufacturers.

Enforcement and expectation of the local law varies, of course. USA, Canada, and most European countries have adopted ITU tables for their respective radio regions. Australia, Hong Kong and Japan also have extensive rules and regulations for low power transmitters and receivers, but with significant differences in the tables for that radio region. Most other countries have less formal regulations, often modeled on either USA or EU regulations.

In any country, it is important to contact the Ministry of Telecommunications or Postal Services to determine any local limitations, allocations, or certifications PRIOR to assembling or testing your first product. The mildest penalty is often total loss of your import, export and foreign exchange privileges.

These laws and requirements are applicable to the finished product, in the configuration that it will be sold the general public or the end user. OEM components often can not be certified, since they require additional non-standard attachments before they have any functional purpose.



Unless otherwise marked, Murata Virtual Wire® Development Kit modules have not been certified to any particular set of regulations. Each module has suggested countries for use, depending on current allocations and technical limits. Emissions from receivers can be an unexpected problem, and the Murata modules have special features to help with this part of the emission testing.

*Product Certification* - General requirements for emissions and ingressions ( called susceptibility, if errors occur) are controlled by engineering standards of performance, regulations, and the customer's expectations.

In USA and Canada, for example, you must formally measure the emissions, file for a certification or authorization, and affix a permanent marking label to every device, prior to offering your system for sale. Regulations allow you to build only a small number (usually 5 pieces) for testing and in-company use, before certification and marketing. Trade shows and product announcements can be a problem for marketing, when the products are advertised without proper disclaimers. With Internet access, go to "www.fcc.org" for USA information or "www.ic.gc.ca" for Canada. The Canada rules are RCC-210, Revision 2. FCC CFR 47, Parts 2 and 15, contains the needed information for USA sales.

European Union (EU) requirements allow self-certification of some systems and require formal measurement reports for other systems. In all cases, however, the directives demand the "CE mark" be added to all compliant devices before any device is freely shipped in commerce. In the EU, the EMC Directive also adds various tests and expectations for levels of signal that will permit acceptable operation.

The EU directives introduce the concepts of a "cognizant body", a "notification body" and a "construction file". Cognizant bodies are simply technical experts recognized by the EU committees to review technical regulations and compliance. Any acceptable test lab will have a preferred cognizant body for their certifications. Each regulatory body will have at least one engineer designated as the notification body for that country, and he

receives any communication about certification or changes to a certified system. While this may seem confusing, it does avoid the legal problems of engineering titles and varied bureaucratic ministry names.

Construction files (CF) are a common format for presenting pictures, schematics and all other information on the parts and processes used to actually build a certified system, The report of antenna range measurements will be included in the CF. Your cognizant body will review the construction file before granting the authorizations for CE mark and EU identification label on your system.

The first problem in the EU is usually Border Customs, who have been trained to look for the CE logo marking for all products. You may need special forms or permits to simply ship pre-production models to your test lab. The Internet web site “[www.etsi.co.fr](http://www.etsi.co.fr)” has information for ordering the full EU marketing regulations.

*Certification Testing* - The emissions are measured in a calibrated environment defined by the regulations. USA and Canada use an “open field” range with 3 meters between the device under test (DUT) and the antenna. The range is calibrated by measurement of known signal sources to generate range attenuation (correction) curves in accordance with ANSI C63.4-1992.

EU measurement rules are based on a similar arrangement, but a “standard dipole” antenna is substituted for the DUT to calibrate the range attenuation. Since the EU measurements are comparison or substitution rules, they are often easier to follow for informal pre-testing by the designer. ETSI-300-220 has drawings to completely describe a typical test configuration.

The United States and Canadian requirements are contained in ANSI C63.4-1992, including a step-by-step test calibration and measurement procedure. Since these rules include range attenuation factors, one must make twice the measurements of the EU

test method. Other countries follow one of these two techniques, with exception for a 10 meter range (separation) measurement or a different range of test frequencies.

Each of the listed contacts will have resources to provide (for a fee) current regulations and certification forms. They also can suggest sources for your formal tests, either commercial labs or the government testing office. Unless you want to invest in a qualified radiated signals test range, the commercial labs can help you with preliminary measurements and some expertise in correcting any difficulties that are noted.

Contacts for further information and current test facilities listings:

**ANSI**

Institute of Electrical & Electronics Engineers,  
345 East 47th Street, New York, NY 10017 USA

**ETSI**

European Telecommunications Standard Institute  
F-06921 Sophia Antipolis Cedex FRANCE

**FCC**

Federal Communications Commission  
Washington DC 20554 USA

**Canada DOC**

Industrie Canada  
Attn: Certification, Engineering and Operations Section,  
1241 Clyde Avenue, Ottawa K1A 0C8 CANADA

**UNITED KINGDOM**

LPRA (manufacturing association information)  
Low Power Radio Association  
The Old Vicarage, Haley Hill, Halifax HX3 6DR UK  
or  
Radiocommunications Agency (official)  
Waterloo Bridge House, Waterloo Road  
London SE1 8UA

**JATE**

Japan Approvals Institute (JATE)  
Isomura Bldg, 1-1-3 Toranomom  
Minato-ku Tokyo JAPAN

## **4 Virtual Wire® Development Kit Installation and Operation**

### **4.1 Development Kit Assembly Instructions**

Kit assembly includes the following steps:

- Install the antennas (or antenna cables) on the data radio boards
- Set the node address on the protocol boards
- Obtain and install AAA batteries in the protocol boards (power switch OFF)
- Plug the data radio boards onto the protocol boards
- Connect 9-Pin PC cables between the protocol boards and the host computers
- Install the terminal program in the host computers
- Configure the terminal programs and test the Virtual Wire® communications link

Take care in plugging a transceiver board into a protocol board. The transceiver board must be oriented so that THE BOARD RESTS ON THE NYLON SCREW SUPPORTS AND NOT OVER THE BATTERIES. Be sure that the transceiver board pins are correctly plugged into the protocol board connector. It is possible to plug the transceiver board in so that a pin is hanging out on the left or right. BE SURE TO INSPECT THE CONNECTOR ALIGNMENT BEFORE APPLYING POWER. Options and adjustments are discussed below:

### **4.2 Data Radio Boards**

The DR1200 Data Radio board is configured to operate at a data rate of 22.5 kbps on a frequency of 916.50 MHz, and the DR1201 is configured to operate at 868.35 MHz. The kits are shipped with a pair of data radio boards and matching antennas. Data Radio boards with antennas can be purchased separately for development of applications.

*Antenna Options-* Antennas are supplied with the data radio boards that can simply be soldered to the pad provided for the 50 ohm RF input (see the Drawings section of the

manual). These antennas are simple base-loaded monopoles. Alternatively, a 50 ohm coaxial cable (RG-178, etc.) can be soldered to the RF input pad and the adjacent ground pad(s), if a remotely located antenna is used. The remote antenna should have an impedance of approximately 50 ohms, preferably with a VSWR of less than 2:1. A remote antenna is necessary if the transceiver is housed inside a metal box, which is very desirable if a noise source such as a high-speed microprocessor, high-speed logic or a brush-type motor is mounted in close proximity to the transceiver board itself.

### 4.3 Protocol Board

*Node Programming* - The node address on each protocol board can be set from 1 to 15 by placing jumpers on the double row of pins located between the two IC's. With no jumpers, the node address is 1. Placing one jumper across the pins nearest to the RS232 connector programs the board to address 2. The rule is that the node number is the binary "jumper" value +1, with jumper pins closest to the RS232 connector being the LSB position. The exception is jumpers on all pins, which is interpreted as node 1. Node 0 is reserved for packets broadcasted to all other nodes. There is a node address table in the Drawings section of the manual.

*Power Supply* - Each node can be operated from three 1.5 V AAA batteries.

*RS232 Interface* - Level conversion from CMOS to RS232 levels is provided by the MAX 218 IC. It is possible to remove this IC and jumper socket Pin 7 to 14 and Pin 9 to 12 for direct CMOS operation.

*LED Functions* - Three LED indicators are provided on the protocol board. The LED labeled **RXI** indicates RF signals are being received. The LED **RF RCV** indicates that a valid RF packet has been received. The LED **PC RCV** indicates that a message has been received from the PC.

## 4.4 Terminal Program

*Installation* - The terminal program is designed to operate under MS-DOS, version 5.0 or higher, on a PC equipped with an RS232 port on COM1 or COM2. To install the terminal program, create a hard drive directory for the Virtual Wire® application and copy the files on the enclosed disk to the directory. After connecting the Virtual Wire® node to the computer and turning the node on, start the terminal program from within its directory by typing VWT97V02.

*Configuration* - The configuration file, VWT97.CFG is provided with default values as follows:

“COM1:””19200””2” (“com port number:””baud rate””TO node address”)

The configuration file can be edited from the terminal program. To edit from the terminal program, use ALT-S. Note that one protocol board is addressed as node 1 and the other is programmed as node 2 when received. As a minimum, the VWT97.CFG file will have to be edited on at least one of the host computers before the Virtual Wire® link can be tested.

*Operation* - The following files are included on the terminal program disk:

VWT97V02.BAS	QuickBASIC (DOS) Source File
VWT97V02.EXE	Terminal Program Executable File
VWT97.CFG	Terminal Program Configuration File (ASCII)
VWT97V02.TXT	Terminal Program Notes (ASCII)

The program uses the following control keys:

Esc	End task (transmit)
ALT-A	Read Node Number

ALT-B	Broadcast Mode
ALT-C	Clear Screen
ALT-D	Decrement TO Node Address
ALT-H	Help Screen
ALT-I	Increment TO Node Address
ALT-R	Protocol Board Reset
ALT-S	Configuration Set Up
ALT-T	Protocol Board Self-Test
ALT-V	Protocol Board Low Voltage Check
ALT-X	Exit Program
CTL-N	Software Reset of the Node Address
CTL-T	Telemetry Mode (no ACK/NAK; both nodes must enable)
Function Key F1	Sends multi-packet test message

When VWT97V02.EXE begins running, the program looks for VWT97.CFG in the current directory and obtains the COM port, baud rate and the TO node address for the communications link. If VWT97.CFG is not present, the program requests configuration information and builds VWT97.CFG. Pressing ALT-S will display configuration information and allow real-time configuration editing. After reading or building the configuration file, the program gets the FROM node address from the protocol board and enters the terminal mode (...UNIT NOT RESPONDING... message displayed if protocol board does not respond). The terminal mode screen includes three windows. The top window is the MESSAGES RECEIVED window, and displays packets sent to the local node. The bottom window is the ENTER MESSAGE TO SEND window, where messages to be sent are input. A blinking cursor of the form “\_” is provided. VWT97V02.EXE supports “plain text” ASCII messages.

The middle window is the MESSAGES SENT window, which shows message packets as sent (SOT [02h] and EOT [03h] symbols displayed as “☺” and “♥”), the packet number, and the packet status. This window gives a real time depiction of how

messages are packetized and sent. Message PACKET# is 1-7 (recycling). Packet STATUS is RX OK ON n (n is the retry number 1-8) for acknowledged packets.

From the keyboard, the user may enter text up to 79 characters per message. The Backspace Key allows for character erasures and corrections, and the Delete Key erases all characters entered but not transmitted. Pressing the Enter Key terminates the message input and starts packetizing and transferring the message to the protocol board via the serial port. Also, when 79 characters are input, message is automatically sent.

If the Enter Key is pressed as the first keyboard action after starting the terminal program, the default message is:

**\*\*Virtual Wire RF Link Test\*\***

A received message is displayed on a single line in the MESSAGES RECEIVED window, whether it is made up of one or more packets. When the whole message is received, it is displayed. If a multi-packet message is partially received and then the first packet of new message is received, the partial message is discarded and reception of the new message is begun.

When text reaches the bottom of the MESSAGES RECEIVED or the MESSAGES SENT window, the text will scroll until ALT-C is invoked to clear all windows. When 79 characters are input or the Enter Key is pressed in the ENTER MESSAGE TO SEND window, this window is cleared and the cursor moves back to the left side of the window.

In the event that the link between the PC and the protocol board is lost (low batteries, ON/OFF switch off, no cable, etc.) a TIME OUT - VW UNIT NOT RESPONDING alarm message will be displayed. If a packet is unacknowledged after eight tries, a LINK



FAULT message will be displayed in the STATUS column of the MESSAGES SENT window.

There are many other possible ways to interact with the Virtual Wire® Link Layer Protocol. The main purposes of the VWT97V02 terminal program is to demonstrate software handshaking with the protocol and to support initial Development Kit testing. Check RFM's Internet web site, [www.wireless.murata.com](http://www.wireless.murata.com), for additional Development Kit software and software application notes.

## **5 Theory of Operation**

### **5.1 Data Radio Boards**

*I/O Interface-* Referring to the Data Radio Board schematic diagram, connector P1 is the interface connector to the protocol board. Pin 1 is the transmitter data input and can be driven directly by a CMOS gate. The transmitter is pulse ON-OFF modulated by a signal on this line changing from 0 to 3 volts. A high level turns the transmitter oscillator on and a low level turns it off. The input impedance to this line is approximately 27 kilohms. Pin 2 is a Vcc line for the TR1000/TR1001 ASH transceiver.

Pin 3 is the PTT line that enables the transmit mode. This line puts the TR1000 (or TR1001) in the transmit mode when it is high (2.5 volts minimum at 2.0 mA maximum). Pin 4 is a Vcc line connected in parallel with Pin 2. Pin 5 is ground. Pin 6 is unused. Pin 7 is a Vcc line, connected in parallel with Pin 2 and Pin 4. Pin 8 the data output pin from the transceiver. This data output is CMOS compatible and is capable of driving a single CMOS gate or a bipolar transistor with a 51 K base resistor. The last connection to the data radio board is the 50 ohm antenna input. The antenna can either be connected directly to the board or connected remotely by using a 50 ohm coaxial cable.

*TR1000/TR1001 ASH Transceiver* - The heart of the DR1200 Data Radio board is the TR1000 ASH transceiver (DR1201 uses the TR1001). This miniature ceramic-metal

hybrid provides a bilateral digital signal to RF signal communication capability. The following section provides an introduction to the ASH transceiver's features, capabilities, theory of operation and configurability:

### ASH Transceiver Features

- Designed for short-range wireless packet data communications
- Supports RF data rates up to 115.2 kbps
- 3 V, low current operation with integrated power down function
- Stable, easy to use, with all critical RF functions contained in the hybrid
- Robust receiver performance with full sensitivity up to 1 GHz
- Highly configurable with a minimum of external parts
- Choice of OOK or ASK transmitter modulation
- Rugged, miniature ceramic-metal package
- Low implementation cost
- Easy certification to FCC, ETSI and similar low-power radio regulations

RFM's amplifier-sequenced hybrid (ASH) transceivers are ideal for short-range wireless data communications where small size, low power consumption and low cost are required. All critical RF functions are contained in the hybrid, simplifying and speeding design-in. The receiver section is sensitive and stable. Two stages of SAW filtering provide excellent receiver out-of-band rejection. The transmitter includes provisions for on-off keyed (OOK) or amplitude-shift keyed (ASK) modulation. The transmitter employs SAW filtering to suppress output harmonics, facilitating compliance with FCC 15.249, ETSI I-ETS 300 220-1 and similar low-power radio regulations.

ASH transceiver technology offers a rich set of enabling features in short-range wireless applications. Key features include:

- Small size - the ASH transceiver package footprint is nominally 0.28 x 0.40 inch, with a package volume of only 0.009 in<sup>3</sup> (146 mm<sup>3</sup>). Transceiver operating is

configured using a dozen miniature passive components, making it practical to add short-range wireless data connectivity to a watch, pen, PDA or shirt pocket PCS phone.

- Low power - the ASH transceiver operates from 3 Vdc, drawing only 6 mA average in transmit and 1.8 to 7.5 mA in receive (set-up dependent). In addition, the transceiver has an integrated power-down mode to support long duration operation from small batteries.
- Robust operation - the ASH transceiver is a single-channel data radio, employing amplitude-shift keyed modulation. But unlike simple AM systems, extensive consideration has been given to operating robustness in the transceiver architecture. The receiver chain includes a narrow-band SAW filter and a SAW delay line, which together provide excellent out-of-band rejection. The logarithmic receiver detector features more than 70 dB of dynamic range. This is combined with 30 dB of AGC, providing 100 dB of overall receiver dynamic range. Data is recovered from the detected base-band signal using a compound data slicer which provides both excellent threshold sensitivity for low-level signals and good rejection of interference 8-10 dB below the peak level of stronger desired signals. Operating robustness is inherent in the signal processing of the radio itself, providing considerable flexibility in the choice of data protocols that can be used with the transceiver.

## ASH Transceiver Operation

The ASH transceiver's unique feature set is made possible by its system architecture. The heart of the transceiver is the amplifier-sequenced receiver section, which provides over 90 dB of stable RF and detector gain without any special shielding or decoupling provisions. Stability is achieved by distributing the total RF gain over *time*. This is in contrast to a superheterodyne receiver, which achieves stability by distributing total RF gain over multiple frequencies.

Refer to the Block Diagram and Timing Cycle drawing in Section A of the manual for the following discussion. This drawing shows the basic amplifier-sequenced receiver architecture. Note that the bias to RF amplifiers RFA1 and RFA2 are independently controlled by a pulse generator, and that the two amplifiers are coupled by a surface acoustic wave (SAW) delay line, which has a typical delay of 0.5  $\mu$ s.

An incoming RF signal is first filtered by a narrow-band SAW filter, and is then applied to RFA1. The pulse generator turns RFA1 ON for 0.5  $\mu$ s. The amplified signal from RFA1 emerges from the SAW delay line at the input to RFA2. RFA1 is now switched OFF and RFA2 is switched ON for 0.55  $\mu$ s, amplifying the RF signal further. The ON time for RFA2 is usually set at 1.1 times the ON time for RFA1, as the filtering effect of the SAW delay line stretches the signal pulse from RFA1 somewhat. As shown in the timing diagram, RFA1 and RFA2 are never on at the same time, assuring excellent receiver stability. Note that the SAW filter and delay line act together to provide very high receiver ultimate rejection.

Amplifier-sequenced receiver operation has several interesting characteristics that can be exploited in system design. The RF amplifiers in an amplifier-sequenced receiver can be turned on and off almost instantly, allowing for very quick power-down (sleep) and wake-up times. Also, both RF amplifiers can be off between ON sequences to trade-off receiver noise figure for lower average current consumption. The effect on noise figure can be modeled as if RFA1 is on continuously, with an attenuator placed in front of it with a loss equivalent to  $10 \cdot \log_{10}(\text{RFA1 duty factor})$ , where the duty factor is the average amount of time RFA1 is ON (up to 50%).

Please refer to the ASH Transceiver Block Diagram in Section A for the following discussion:

Antenna port - the only external RF components needed for the ASH transceiver are the antenna, antenna matching coil and electrostatic discharge (ESD) protection choke.

Receiver chain - the narrow-band SAW filters provides high receiver RF selectivity. The output of the SAW filter drives amplifier RFA1. This amplifier includes provisions for detecting the onset of saturation (AGC Set), and for switching between 35 dB of gain and 5 dB of gain (Gain Select). AGC Set is an input to the AGC Control function, and Gain Select is the AGC Control function output. ON/OFF control to RFA1 (and RFA2) is generated by the Pulse Generator & RF Amp Bias function. The output of RFA1 drives the low-loss SAW delay line, which has a nominal delay of 0.5  $\mu$ s. Note that the SAW RF filter and SAW delay line both contribute to the excellent out-of-band rejection of the receiver.

The second amplifier, RFA2, provides 51 dB of gain below saturation. The output of RFA2 drives a square-law detector with 19 dB of threshold gain. The onset of saturation in each section of RFA2 is detected and summed to provide a logarithmic response. This is added to the output of the square-law detector to produce an overall detector response that is square law for low signal levels, and transitions into a log response for high signal levels. This combination provides excellent threshold sensitivity and more than 70 dB of detector dynamic range. In combination with the 30 dB of AGC range in RFA1, more than 100 dB of receiver dynamic range is achieved.

The detector output drives a three-pole, 0.05 degree equiripple low-pass filter response with excellent group delay flatness and minimal pulse ringing. The 3 dB bandwidth of the filter is adjusted with a single external resistor to match the data rate and data encoding of the transmitted signal.

The filter is followed by a base-band amplifier which boosts the detected signal to the BBOUT pin, which is coupled to the CMPIN pin or to an external data recovery process (DSP, etc.) by a series capacitor.

When the transceiver is placed in power-down or in a transmit mode, the output impedance of BBOUT becomes very high. This feature helps preserve the charge on

the coupling capacitor to minimize data slicer stabilization time when the transceiver switches back to the receive mode.

**Data Slicers** - The CMPIN pin drives two data slicers, which convert the analog signal from BBOUT back into a data stream. The best data slicer choice depends on the system operating parameters. Data slicer DS1 is a capacitor-coupled comparator with provisions for an adjustable threshold. DS1 provides the best performance at low signal-to-noise conditions. The threshold, or squelch, offsets the comparator's slicing level, and is set with a resistor between the RREF and THLD1 pins. This threshold allows a trade-off between receiver sensitivity and output noise density in the no-signal condition. S2 is a "dB-below-peak" slicer. The peak detector charges rapidly to the peak value of each data pulse, and decays slowly in between data pulses (1:1000 ratio). The slicer trip point can be set from 0 to 12 dB below this peak value with a resistor between RREF and THLD2. DS2 is best for ASK modulation where the transmitted signal has been shaped to minimize signal bandwidth.

**AGC Control** - The output of the Peak Detector also provides an AGC Reset signal to the AGC Control function through the AGC comparator. The purpose of the AGC function is to extend the dynamic range of the receiver, so that two transceivers can operate close together when running ASK and/or high data rate modulation. The AGC also prevents receiver saturation by a strong in-band interfering signal, allowing operation to continue at short range in the presence of the interference. The onset of saturation in the output stage of RFA1 is detected and generates the AGC Set signal to the AGC Control function. The AGC Control function then selects the 5 dB gain mode for RFA1. The AGC comparator will send a reset signal when the Peak Detector output (multiplied by 0.8) falls below the fixed reference voltage for DS1. A capacitor at the AGCCAP pin avoids AGC "chattering" during the time the signal propagates through the log detector, low-pass filter and charges the peak detector. The AGC capacitor also allows the AGC hold-in time to be set longer than the peak detector decay time to avoid AGC chattering during runs of "0" bits in the received data stream. Note that AGC operation requires the peak detector to be functioning, even if DS2 is not used. AGC

operation can be defeated by connecting the AGCCAP pin to VCC, or latched ON connecting a resistor between the AGCCAP pin and ground.

Receiver pulse generator and RF amplifier bias - The receiver amplifier-sequence operation is controlled by the Pulse Generator & RF Amplifier Bias module, which in turn is controlled by the PRATE and PWIDTH input pins, and the Power Down Control Signal from the Modulation & Bias Control function.

Transmitter chain - the transmitter chain consists of a SAW delay line oscillator TXA1, followed by a modulated buffer amplifier TXA2. The SAW filter suppresses transmitter harmonics to the antenna. Note that the same SAW devices used in the amplifier-sequenced receiver are reused in the transmit modes. A resistor is used to provide decoupling between the RF VCC and the power amplifier Vcc. (See R18 on the DR1200 schematic diagram.)

Transmitter operation supports two modulation formats, on-off keyed (OOK) modulation, and amplitude-shift keyed (ASK) modulation. When OOK modulation is chosen, the transmitter output turns completely off between “1” data pulses. When ASK modulation is chosen, a “1” pulse is represented by a higher transmitted power level, and a “0” is represented by a lower transmitted power level. OOK modulation provides compatibility with first-generation ASH technology, and provides for power conservation. ASK modulation must be used for high data rates (data pulses less than 30  $\mu$ s). ASK modulation also reduces the effects of some types of interference and allows the transmitted pulses to be shaped to control modulation bandwidth. The transmitter RF output voltage is proportional to the input current to the TXMOD pin, which modulates TXA2. A resistor in series with TXMOD adjusts the peak transmitter output power.

The four transceiver operating modes - receive, transmit ASK, transmit OOK and power-down (“sleep”), are controlled by the Modulation & Bias Control function, and are selected with the CNTRL1 and CNTRL0 control pins. CNTRL1 and CNTRL0 are CMOS compatible inputs.

## ASH Transceiver Configurability

ASH transceivers are highly configurable, offering the user great flexibility in optimizing for specific applications and protocol formats. The operating configuration is set using low-cost resistors and capacitors. Key points of configurability include:

- Adjustable receiver sensitivity versus current consumption
- Adjustable receiver low-pass filter to support various data rates/encoding techniques
- Adjustable peak transmitter output power
- Conventional or “dB below peak” data slicer select
- Adjustable thresholds (squellch settings) for each data slicer
- Adjustable AGC hold-in time and AGC latch/defeat function
- OOK or ASK modulation with adjustable ASK modulation depth
- Continuous or duty-cycled operation (integrated power down function)
- 2.7 to 3.5 Vdc power supply range (down to 2.2 Vdc over limited temperature range)

## *Data Radio Board Specifications*

### Operating Frequency

DR1200	916.5 MHz
DR1201	868.35 MHz

Modulation                      On-Off Keyed

Antenna                         50 ohm

Operating Data Rate            22.5 kbps (44.4  $\mu$ s min. pulse width @ TX input)

TX Frequency Tolerance        less than  $\pm 200$  kHz, including set-on, temperature and aging drift (5 year)



TX Output Power	-6 dBm nominal
TX Harmonics	less than -32 dBc
Receiver Performance	BER less than $10E-4$ for a -85 dBm input (22.5 kbps)
RX Pulse Distortion	less than $\pm 25\%$ for a 44.4 $\mu$ s TX pulse
RX Dynamic Range	-85 to -10 dBm
Data DC Balance	receiver performance shall be maintained for data with an average “1” density from 45 to 55%
Data Run Length	receiver performance shall be maintained for “1” or “0” run lengths of at least 6 bits
RX Off-Channel Rejection	
DR1200-DK	greater than 60 dB, 0.25 to 890 MHz and 945 to 2500 MHz
DR1200-DK	greater than 60 dB, 0.25 to 842 MHz and 897 to 2500 MHz
RX On-Channel Rejection	less than 30% BER degradation for an interfering signal at least 15 dB below the desired signal after 16 bits (50% duty cycle) of the desired signal received
RX No-Signal Output	less than one noise “spike” average in any 10 ms interval under “white thermal noise” reception conditions

Transceiver Mode Change	
RX to TX	10 $\mu$ s
TX to RX	78 $\mu$ s
DC Power Supply	2.7 to 3.5 Vdc, 10 mV max peak-to-peak ripple
Supply Current, RX Mode	less than 4.0 mA ave @ 3 Vdc supply
Supply Current, TX Mode	less than 12 mA peak @ 3 Vdc supply
I/O Data Interface	3 V CMOS logic level for serial TX input; serial RX output capable of driving one 3 V CMOS gate
TX/RX Control Input	low for RX, high for TX (source 2 mA @ 2.5 V min.)
Operating Temperature Range	-40 to +85 deg C

## 5.2 Protocol Board

*I/O Interface* - Connector J1 (see Protocol Board schematic) is the I/O interface between the protocol board and the data radio board. J1-Pin 1 carries the transmit data signal from U2-Pin 7 to the transmitter input on the Data Radio board. J1-Pin 2 provides Vcc to the Data Radio board. J1-Pin 3 provides the transmit enable signal (PTT) from PNP transistor Q2. The Data Radio board requires 2 mA at 2.5 V on the PTT input to enable the transmit mode. J1-Pin 7 is another Vcc input to the Data Radio board. J1-Pin 5 is ground. J1-Pin 4 is a third Vcc input to the Data Radio board. J1-Pin 8 carries the receiver digital output from the Data Radio board. Q1 provides a high input impedance buffer between this signal and the input to U2. J1-Pin 6 is unused in the DR1200-DK and DR1201-DK implementation.

*RS232 Interface* - Connector J2 is the RS232 interface on the protocol board. This 9-Pin female connector is configured to appear as a DCE (modem). The protocol board implements software flow control, so only J2-Pin 2 and J2-Pin 3 carry active signals. J2-Pin 2 (RD) sends data to the host computer, and J2-Pin 3 receives data from the host computer (TD). J2 Pins 4 and 6 are connected (DTR & DSR), and J2 Pins 1,7 and 8 are connected (CD, RQS, CTS) J2-Pin 5 is ground.

*Protocol Microcontroller* - The link-layer protocol is implemented in an ATMEL AT89C2051 microcontroller U2. The 8-bit microcontroller operates from an 22.118 MHz quartz crystal. The microcontroller includes 2 Kbytes of flash EPROM memory and 128 bytes of RAM. The microcontroller also includes two 16-bit timers and one hardware serial port, making it especially suitable as a link-layer packet controller. The timers, serial port and input interrupts remain active while the processor is in the power-saving idle mode, allowing the link-layer protocol to be implemented on a low average current budget.

Inputs to the microcontroller include the node programming pins ID0 - ID3, on Pins 14, 15, 16 and 17, the buffered receive data (RRX) on Pin 6, the CMOS-level input from the host computer on Pin 2. Outputs from the microcontroller include the transmit data on Pin 7, the data output to the host computer on Pin 3, the transmit enable signal Pin 19, the RS232-transceiver control on Pin 18, and the LED outputs on Pins 8 (RXI), 9 (RF RCV), and 11(PC RCV). Diode D2 and capacitor C7 form the power-up reset circuit for the microcontroller.

*CMOS/RS232 Level Converter* - Conversion to and from RS232 and 4.5 V CMOS logic levels is done by U1, a Maxim MAX218 Dual RS232 Transceiver. The operation of the MAX218 is controlled by microcontroller U2, to minimize average current consumption. L1, D1 and C5 operate in conjunction with the IC's switch-mode power supply to generate  $\pm 6.5$  V for the transmitter and receiver conversions. Pin 3 on the MAX218 controls the switched-mode supply via U2 Pin 18. The RS232 serial input signal from J2-Pin 3 is input on U1-Pin 12 and is converted to a 4.5 V CMOS level (note inversion)

and output on U1-Pin 9. The CMOS serial output signal from U2-Pin 2 is input on U1-Pin 7 and converted to an RS232 output (note inversion) on U1-Pin 14. This signal is found on J2-Pin 3.

The RS232 conversion can be bypassed for direct CMOS operation by removing U1 from its socket and placing one jumper in socket Pins 7 and 14 and a second jumper in socket Pins 9 and 12.

### *Protocol Board Specifications*

Host Interface	RS232 DCE compatible 9-Pin female (modem) connector, 19.2 kbps, byte asynchronous, 1 start bit, 8 data bits, no parity, 1 stop bit
Radio Interface	Murata Data Radio Type-1 interface, 8-Pin SIP connector, 22.5 kbps, 12 DC-balanced symbol bits/byte, with integrated PTT control
Power Supply	4.5 Vdc nominal from 3 AAA batteries
Operating Temperature Range	0 to 70 deg C
Storage Temperature Range	-40 to +85 deg C

## **5.3 Protocol Firmware**

*Description* - The purpose of this data-link protocol is to provide automatic, verified, error-free transmission of messages between Virtual Wire® Radio Nodes via RS232 serial connections to the host processors. Operation on both the RS232 side and the radio side is half-duplex.

Operation of the RS232 serial connection is 19.2 kbps, with eight data bits (byte), one stop bit, and no parity bit. The transmission rate on the radio side is approximately 22.5 kbps, using 12-bit DC-balanced symbols for each data byte. The radio receiver is slightly “squelched” when not receiving data, and will output occasional random positive noise spikes.

The following I/O lines are implemented on the protocol microcontroller:

- radio receive line (RRX)
- radio transmit line (RTX)
- radio transmit/receive control line, high on transmit (PTT)
- RS232 receive line (PRX)
- RS232 transmit line (PTX)
- Maxim 218 ON/OFF control line
- node ID input lines (ID0 through ID3)
- three LED control Lines (RXI, RF RCV and PC RCV)

The link-layer protocol is capable of transmitting/receiving binary data bytes of any bit pattern. Messages are sent and received from the RS232 interface in standard asynchronous format via PTX and PRX. The first byte of the RS232-side messages contain a “TO/FROM” address, with the high nibble the “TO” node ID, and the low nibble the “FROM” node ID. The second byte is the message sequence number (1-7 recycling or 8 used for telemetry packets), the third byte is the number of data bytes in the message (up to 20 hex), followed by the data bytes. A single message can contain up to 32 data bytes, with 16 to 24 data bytes typical.

The protocol software continually tests the RRX line and the PRX line, searching for a start bit. When a start bit is detected on one of the input lines, the software will attempt to receive a message on that input line. If an error is detected in the message, it will be discarded and the software will resume testing the input lines.

If a valid message is received on the PRX input line, the software will format a radio packet from the message and queue the packet for transmission. The packet format shall include a start symbol (22 hex), the TO/FROM byte, the message number, the number of data bytes in the message, the data bytes, and a 16 bit error detection “frame check sequence” (FCS). The FCS is calculated based on all bits in the message *following* the start symbol. The FCS as defined by ISO 3309 is used.

Each byte transmitted by the radio is converted into a 12 bit, dc-balanced symbol. DC balance “trains” the receiver data slicer for best noise immunity by setting the threshold half way between a “1” and “0” value. The dc-balanced symbols used have no more than 3 bits of the same value in a row. This limited “run length” allows the receiver data slicer to be tuned to recover quickly from a heavy noise burst or strong interfering signal.

The queued packet is transmitted (RTX line with PTT invoked), and the software then looks for a “packet received” acknowledgment (RRX line). An acknowledgment packet includes the start byte, the TO/FROM byte, the packet number being acknowledged, hex En (n is the retry number correctly received) and the FCS. When an acknowledgment is received for the queued packet, an acknowledgment message (packet less start symbol and FCS) is sent on the PTX line, the packet is discarded, and the software resumes testing the input lines. If a packet acknowledgment is not received in 120 ms, the packet is resent after a randomly selected delay of (approximately) 0 , 120, 240 or 360 ms. If the packet is not acknowledged after a total of eight tries, the software will send a “NAK” message on the PTX line (TO/FROM address, packet number and hex DD), discard the packet, and resume testing the input lines.

When a start symbol is detected on the RRX line, the software will attempt to receive and verify a message by checking for a correct TO/FROM address, a valid packet sequence number, a valid number of data bytes (or “ACK” character), and a correct FCS calculation. If the packet is verified and the “TO” nibble matches, the TO/FROM address, packet sequence number, number of data bytes and the data bytes of the

message are sent out on the PTX line and a packet acknowledgment is transmitted back on the RTX line. Otherwise, the message is discarded and testing of the input lines is resumed. The software will accept message packets and acknowledgment packets in any sequence. If an acknowledged packet is *received a second time* (based on the current value of the message sequence counter) it is reacknowledged on RTX but not retransmitted on PTX.

The TO/FROM address of 00h is treated as a “broadcast” packet. In this case, a received packet is sent out on the PTX line if the number of data bytes are in a valid range and the FCS calculation matches. In the broadcast mode, the packet is transmitted eight times to enhance probability of reception. A broadcast packet is not acknowledged by the receiving node(s). A packet with a packet number of 8 is treated as a telemetry packet. If the TO address matches the local node number and the number of bytes and FCS is valid, it is placed on the PTX, but is not acknowledged.

If a start bit is detected on either RRX or PRX, the software receives and acts on the information on that input line, and does not service the other input line until it has received and acted on the data from the first input line. Host software must implement a simple transmit message software flow control to accommodate this characteristic. When the host software is ready to send a message to the protocol software, it tests the availability of the PRX interrupt by sending just the TO/FROM address character to the protocol board. If this TO/FROM address is echoed back within 50 ms, it has control of the PRX interrupt process and can send the rest of the message in the following 200 ms. If a character is input with the high nibble equal to the local node address (or the byte equal to 00h for a broadcast packet) within the 50 ms window, it could be an inbound message for the local node, and the rest of the message should be received and tested to see if it is a valid message. If no character is echoed in the 50 ms following the TO/FROM character transmission, the protocol board can be assumed busy on an RRX interrupt either receiving a packet or tripped by receiver output noise. The host program should retry. An inbound packet can occur at any time, so any character with the high nibble equal to the local node address or any 00h byte should

be processed to test for a valid message. In addition to packet messages, special messages are provided, such as “Run Self Test”. Special message formats are given below.

### *Protocol-Host Message Formats*

#### *General Message Format:*

| TO/FROM Byte | Packet #, 01h to 08h | Size/Status Byte | Message (up to 32 bytes) |

Note: the Size/Status byte indicates the number of bytes in the message (up to 20h), or a status message (e.g., ACK = 0Enh, NAK = 0DDh, etc.) Packet # 08h indicates a telemetry packet. A TO address of 0h indicates a broadcast packet.

#### *Link Status Messages:*

ACK | TO/FROM | Packet # | En | (n = 1 to 8)

NAK | TO/FROM | Packet # | DD |

#### *Special Messages from the Host to the Protocol:*

Reset: | FROM/FROM | 00h | 01h | 30h |

Send Node Address: | 00h | 00h | 01h | 31h |

Set Node Address: | 00h | 00h | 01h | 34h |

Run Self Test: | FROM/FROM | 00h | 01h | 33h |



### *Special Messages from the Protocol to the Host:*

Message from Host too Long: | FROM/FROM | 00h | 01h | 30h |

Failed Self Test: | FROM/FROM | 00h | 01h | 33h |

Passed Self Test: | FROM/FROM | 00h | 01h | 34h |

Local Node Address: | FROM/FROM | 00h | 01 | 35h |  
where FROM/FROM is the local address

## **5.4 Terminal Program**

*Description* - VWT97V02.EXE is a compiled QuickBASIC (4.5) program that demonstrates interaction with the Virtual Wire® Protocol. The commented source code listing is provided below for reference. Be sure to check RFM's Internet web site - [www.wireless.murata.com](http://www.wireless.murata.com) for new software updates. *Note: This terminal program does not include provisions for automatic compliance with the duty-cycle requirements of the new ETSI I-ETS 300 220-1 regulations for the 868.00 - 868.60 MHz band (DR1201-DK). Limiting the duty-cycle in the final application software will be required for ETSI certification. At the time this manual is was written, the proposed duty cycle limit was seconds per hour, but this may change in the final release of the specification. Be sure to check the final version of the regulation before completing product development.*

```
'VWT97V02.BAS  VWDK TERMINAL PROGRAM, Copyright RF Monolithics, Inc., 1996, 1997
'REV 04-15-97  ADDED MULTIPLE BAUD RATE SUPPORT
'REV 04-25-97  MISC SCREEN CLEAN UPS
'REV 05-06-97  ADDED "TO ADDRESS" ALERT MESSAGE
'REV 05-10-97  GENERAL CLEAN UPS
'REV 05-15-97  VWT97V01 FIRST RELEASE
'REV 11-14-98  VWT97V02 TUNED FOR DR12xx-DK ASH TRANSCEIVER DEVELOPMENT KITS
```

```
DEFINT A-Z: OPTION BASE 1
```

'THE SPECIAL MESSAGE FORMAT IS THE SAME TO AND FROM THE VIRTUAL WIRE UNIT.

'NOTE T/F = F/F SINCE MESSAGE IS NOT FOR ANY OTHER UNITS.

```
MSG.FROM.VW$ = CHR$(&H0) 'NULL = UNKNOWN MESSAGE WAS RECEIVED
MSG0.FROM.VW$ = "0" '(&H30) MESSAGE PACKET i.e. NUMBER BYTES TOO LONG
MSG1.FROM.VW$ = "1" '(&H31) LOW BATTERY
MSG2.FROM.VW$ = "2" 'BATTERY VOLTAGE OK
MSG3.FROM.VW$ = "3" 'FAILED SELF TEST
MSG4.FROM.VW$ = "4" 'PASSED SELF TEST
MSG5.FROM.VW$ = "5" 'TO/FROM (T/F) = F/F = VIRTUAL WIRE UNIT ADDRESS
```

```
MSG0.TO.VW$ = "0"      'RESET VW UNIT
MSG1.TO.VW$ = "1"      'SEND ADDRESS
MSG2.TO.VW$ = "2"      'TEST BATTERY AND RETURN RESULTS
MSG3.TO.VW$ = "3"      'RUN SELF TEST
MSG4.TO.VW$ = "4"      'CHANGE NODE ADDR.
```

FULL.MSG\$ = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789ABCD "

```
F1$ = CHR$(0) + CHR$(59)      'FUNCTION KEY F1
F2$ = CHR$(0) + CHR$(60)
F3$ = CHR$(0) + CHR$(61)
F4$ = CHR$(0) + CHR$(62)
F5$ = CHR$(0) + CHR$(63)
F6$ = CHR$(0) + CHR$(64)
F7$ = CHR$(0) + CHR$(65)
F8$ = CHR$(0) + CHR$(66)
F9$ = CHR$(0) + CHR$(67)
F10$ = CHR$(0) + CHR$(68)
```

TAB\$ = CHR\$(9)

```

SHIFT.TAB$ = CHR$(0) + CHR$(&HF)

ALT.C$ = CHR$(0) + CHR$(46)
ALT.S$ = CHR$(0) + CHR$(31)
ALT.X$ = CHR$(0) + CHR$(45)
ALT.A$ = CHR$(0) + CHR$(30)
ALT.H$ = CHR$(0) + CHR$(35)
ALT.I$ = CHR$(0) + CHR$(23)
ALT.V$ = CHR$(0) + CHR$(47)
ALT.T$ = CHR$(0) + CHR$(20)
ALT.D$ = CHR$(0) + CHR$(32)
ALT.B$ = CHR$(0) + CHR$(48)
ALT.R$ = CHR$(0) + CHR$(19)

CTRL.T$ = CHR$(20)
CTRL.N$ = CHR$(14)

'COMM. CONTROL CHARACTERS
SOH$ = CHR$(1)      'START OF HEADER
STX$ = CHR$(2)      'START OF TEXT - START OF XMISSION
ETX$ = CHR$(3)      'END OF TEXT - END OF XMISSION
EOT$ = CHR$(4)      'END OF TRANSMISSION
ETB$ = CHR$(&H17)   'END OF BLOCK - LAST DATA BYTE 0 - 16
NAK$ = CHR$(&H15)   'NEGATIVE ACKNOWLEDGE - CHECK SUM NO COMPUTE
ACK$ = CHR$(6)      'ACKNOWLEDGE - CHECK SUM OK

RCV.PTRX = 1        'POINTER FOR RECEIVE WINDOW
RCV.PTRY = 5

LAST.MSG$ = "***Virtual Wire RF Link Test**" 'tweaked 97.05.10

Packet = 1
TO.ADDR = 2          'DEFAULT
TO.ADDR$ = "2"       'DEFAULT added 97.05.10
COM.PORT$ = "COM2:"  'DEFAULT
BAUD.RATE$ = "4800"  'DEFAULT

ON ERROR GOTO PRTERRO 'set up error handler (files, com port, etc.)
GOSUB SETUP.DSK      'read/build/save configuration files to disk 97.05.10

OPEN COM.PORT$ + BAUD.RATE$ + ",N,8,1,RS,CD0,DS0,CS0" FOR RANDOM AS #1 LEN = 2048

CLS
GOSUB GET.ADDRESS
GOSUB TOADDR.MSG     'added 97.05.10

MAIN:

COLOR 7, 0           'Ensure screen state 97.05.10

RCV.PTRX = 1        'POINTER FOR RECEIVE WINDOW
RCV.PTRY = 5

GOSUB SCREEN1

IF LOC(1) > 0 THEN
    DATAB$ = INPUT$(LOC(1), #1) 'CLEAR COMM. BUFFER
ELSE
    END IF

DATAB$ = ""

'***** EDIT MESSAGE *****

MINI.WORD:

ERASE MINI.MSG$      'MESSAGE FOR SENDING TO VW UNIT
LOCATE 23, 1, 1, 5, 7

MINI:

IF BUSY = 0 AND NUMBER.TO.SEND > 0 THEN

```

```

    GOSUB SEND.PACKET
ELSE
END IF

'-----CHECK COMM. BUFFER -----

IF LOC(1) > 0 THEN      'KEEP LOOKING FOR COMMUNICATIONS
    GOSUB READ.BUFFER
    IF BUSY > 0 THEN      'SEE IF IT IS RESPONSE TO LAST PACKET
        IF LEN(DATAB$) = 3 THEN      'LENGTH CORRECT FOR VW.ACK OR VW.NAK
            IF ASC(MID$(DATAB$, 2, 1)) = Packet THEN 'CORRECT RESPONSE
                Packet = Packet + 1
                IF Packet = 8 THEN
                    Packet = 1
                ELSE
                    END IF
            ELSE
                END IF

            '----- VW.ACK$  &HEn   n = 1 to 9

            IF RIGHT$(DATAB$, 1) >= CHR$(&HE1) AND RIGHT$(DATAB$, 1) <= CHR$(&HE9) THEN
                RMSG$ = "RX OK ON " + CHR$((ASC(RIGHT$(DATAB$, 1)) - &HE0) OR &H30)
            ELSEIF RIGHT$(DATAB$, 1) = CHR$(&HEE) THEN      'For compatibility with VWT.BAS (4-25-
97)
                RMSG$ = "RX OK"
            ELSEIF RIGHT$(DATAB$, 1) = VW.NAK$ THEN
                IF BROADCAST = 1 THEN
                    RMSG$ = "BROADCAST FINISHED"
                ELSE
                    IF (ASC(LEFT$(DATAB$, 1)) AND &HF) = TO.ADDR THEN
                        '-ATMEL DOES NOT SEND "NAK" BUT COULD IN FUTURE
                        RMSG$ = "RF LINK FAULT (REMOTE)"      'tweaked 97.05.10
                    ELSEIF (ASC(LEFT$(DATAB$, 1)) AND &HF0) / 16 = TO.ADDR THEN
                        RMSG$ = "RF LINK FAULT" 'NO RESPONSE TIME OUT - tweaked 97.05.10
                    ELSE
                        END IF
                    END IF
                END IF

                BEEP
                BUSY = 0      'IF LINK FAULT THEN STOP SENDING
                NUMBER.TO.SEND = 0
            ELSE
                RMSG$ = ""
            END IF

            DATAB$ = ""
            '--- PUT MESSAGE ON DISPLAY

            IF RMSG$ <> "" THEN
                SELECT CASE PACKET.PTR
                    CASE 1
                        LOCATE 16, 62, 0
                        PRINT RMSG$;
                    CASE 2
                        LOCATE 17, 62, 0
                        PRINT RMSG$;
                    CASE 3
                        LOCATE 18, 62, 0
                        PRINT RMSG$;
                    CASE ELSE
                        END SELECT
                BUSY = 0
                PACKET.PTR = PACKET.PTR + 1
                IF PACKET.PTR > NUMBER.TO.SEND THEN      'LAST PACKET SENT
                    PACKET.PTR = 1
                    NUMBER.TO.SEND = 0
                    GOSUB CLEAR.W3
                    COLOR 0, 7
                    LOCATE 23, 1, 1, 5, 7
                ELSE
                    END IF
            ELSE
                END IF
            ELSE
                'WRONG PACKET# ERROR
            END IF
        END IF
    END IF

```

```

        ELSE
            GOSUB RECEIVE.DATA 'LONGER THEN THREE BYTES IS INCOMING PACKET.
        END IF
    ELSE
        GOSUB RECEIVE.DATA
    END IF
ELSE 'Nothing in buffer
END IF

'----- END OF CHECK COMM. BUFFER -----

IF TELEMETRY = 1 AND RETURN.FROM.SEND = 1 THEN
    RETURN.FROM.SEND = 0
    BUSY = 0
    NUMBER.TO.SEND = 0
    COLOR 7, 0
    GOSUB CLEAR.W2
    GOSUB CLEAR.W3
    COLOR 0, 7
    START.OF.PACKET = 0
    PACKET.PTR = 1
    GOTO MINI.WORD
ELSE
END IF

KY$ = INKEY$
IF KY$ = "" THEN
    GOTO MINI
END IF

IF KY$ = ESC$ THEN
    BUSY = 0
    NUMBER.TO.SEND = 0
    COLOR 7, 0
    GOSUB CLEAR.W2
    GOSUB CLEAR.W3
    COLOR 0, 7
    START.OF.PACKET = 0
    PACKET.PTR = 1
    GOTO MINI.WORD
ELSE
END IF

'----- CHANGE UNIT ADDRESS -----

IF KY$ = CTRL.N$ THEN
    GOSUB CLEAR.W3
    LOCATE 23, 1, 1, 5, 7
    COLOR 0, 7
    INPUT ; "ENTER NEW ADDRESS 1 - 15: ", NODE$
    COLOR 7, 0
    NODE = VAL(NODE$) AND &HF
    Packet$ = CHR$(0) 'PACKET 0
    N.BYTES$ = CHR$(2) '2 BYTE MESSAGE
    MSG$ = MSG4.TO.VW$ 'SPECIAL MESSAGE FOR VW UNIT - CHANGE NODE ADDR.
    TAD$ = CHR$(0)
    MSG.FORMAT$ = TAD$ + Packet$ + N.BYTES$ + MSG$ + CHR$(NODE)
    PRINT #1, MSG.FORMAT$;
    DELAY.TIME! = TIMER
    DO WHILE ABS(TIMER - DELAY.TIME!) < 1! 'DELAY FOR BATTERY TEST
        LOOP
    IF LOC(1) > 0 THEN 'LOOK FOR RESPONSE
        GOSUB READ.BUFFER
    ELSE
    END IF
    DATAB$ = ""

    KY$ = ALT.A$ 'SET UP NEXT CODE TO READ NEW ADDRESS
ELSE
END IF

'----- GO READ VW ADDRESS -----

```

```

IF KY$ = ALT.A$ THEN
    GOSUB CLEAR.W1
    COLOR 0, 7
    GOSUB GET.ADDRESS
    LOCATE 1, 29
    PRINT " "; FROM; " ";
    KY$ = ""
    GOSUB CLEAR.W1
    GOSUB CLEAR.W2
    GOSUB CLEAR.W3
    COLOR 0, 7
    LOCATE 23, 1, 1, 5, 7
    GOTO MAIN:
ELSE
END IF

'----- TOGGLE BROADCAST FLAG -----

'BROADCAST SENDS MESSAGE TO ALL VW UNITS.

IF KY$ = ALT.B$ THEN
    SAVE.X = POS(0)
    SAVE.Y = CSRLIN

    IF BROADCAST = 1 THEN
        BROADCAST = 0
        COLOR 7, 0
        LOCATE 3, 30, 0
        PRINT " ";
        COLOR 0, 7
    ELSE
        BROADCAST = 1
        LOCATE 3, 30, 0
        PRINT " BROADCAST MODE ENABLED ";
    END IF

    LOCATE SAVE.Y, SAVE.X, 1, 5, 7
    KY$ = ""
ELSE
END IF

'----- TOGGLE TELEMETRY FLAG -----

'TELEMETRY DOES NOT RECEIVE AN "ACK"

IF KY$ = CTRL.T$ THEN
    SAVE.X = POS(0)
    SAVE.Y = CSRLIN

    IF TELEMETRY = 1 THEN
        TELEMETRY = 0
        Packet = 1
        COLOR 7, 0
        LOCATE 3, 30, 0
        PRINT " ";
        COLOR 0, 7
    ELSE
        TELEMETRY = 1
        LOCATE 3, 30, 0
        PRINT " TELEMETRY ENABLED ";
    END IF
    LOCATE SAVE.Y, SAVE.X, 1, 5, 7
    KY$ = ""
ELSE
END IF

'----- VW BATTERY TEST COMMAND -----

IF KY$ = ALT.V$ THEN
    Packet$ = CHR$(0)
    N.BYTES$ = CHR$(1)
    MSG$ = MSG2.TO.VW$
    'BATTERY TEST
    'PACKET 0
    '1 BYTE MESSAGE
    'SPECIAL MESSAGE FOR VW UNIT - BATTERY TEST

```

```

TAD$ = CHR$((FROM * 16 + FROM) AND &HFF)
MSG.FORMAT$ = TAD$ + Packet$ + N.BYTES$ + MSG$
PRINT #1, MSG.FORMAT$;
DELAY.TIME! = TIMER
SAVE.X = POS(0)
SAVE.Y = CSRLIN
GOSUB CLEAR.W2
COLOR 7, 0
LOCATE 17, 30, 0
PRINT " TESTING BATTERY ";
COLOR 0, 7
DO WHILE ABS(TIMER - DELAY.TIME!) < 1!      'DELAY FOR BATTERY TEST
LOOP
IF LOC(1) > 0 THEN      'LOOK FOR RESPONSE
GOSUB READ.BUFFER
IF LEN(DATAB$) = 5 THEN      'ECHO + RETURNED STRING
IF LEFT$(DATAB$, 1) = TAD$ AND MID$(DATAB$, 2, 1) = TAD$ THEN
IF RIGHT$(DATAB$, 1) = MSG2.FROM.VW$ THEN 'BATTERY VOLTAGE OK
GOSUB CLEAR.W2
COLOR 7, 0
LOCATE 17, 30
PRINT " BATTERY OK ";
COLOR 0, 7
GOSUB ShowIt      'added 97.05.10
GOSUB CLEAR.W2      'added 97.05.10
ELSEIF RIGHT$(DATAB$, 1) = MSG1.FROM.VW$ THEN
GOSUB CLEAR.W2
COLOR 7, 0
LOCATE 17, 30
PRINT " LOW BATTERY ";
COLOR 0, 7
BEEP
GOSUB ShowIt      'added 97.05.10
GOSUB CLEAR.W2      'added 97.05.10
ELSE
COLOR 7, 0      'added 97.05.10
LOCATE 17, 30
PRINT " INVALID TEST - RETRY ALT-V "
COLOR 0, 7
BEEP
GOSUB ShowIt
GOSUB CLEAR.W2
END IF
ELSE
COLOR 7, 0      'added 97.05.10
LOCATE 17, 30
PRINT " INVALID TEST - RETRY ALT-V "
COLOR 0, 7
BEEP
GOSUB ShowIt
GOSUB CLEAR.W2
END IF
ELSE
COLOR 7, 0      'added 97.05.10
LOCATE 17, 30
PRINT " INVALID TEST - RETRY ALT-V "
COLOR 0, 7
BEEP
GOSUB ShowIt
GOSUB CLEAR.W2
END IF
ELSE
GOSUB CLEAR.W2
COLOR 7, 0
LOCATE 17, 30
PRINT " VW UNIT NOT RESPONDING ";      'tweaked 97.05.10
COLOR 0, 7
BEEP
GOSUB ShowIt      'added 97.05.10
GOSUB CLEAR.W2      'added 97.05.10
END IF
LOCATE SAVE.Y, SAVE.X, 1, 5, 7
COLOR 0, 7      'added 97.05.10
KY$ = ""
DATAB$ = ""
ELSE

```

```

'not ALT-V
END IF

'----- VW SELF TEST COMMAND -----

IF KY$ = ALT.T$ THEN      'SELF TEST
  Packet$ = CHR$(0)      'PACKET 0
  N.BYTES$ = CHR$(1)     '1 BYTE MESSAGE
  MSG$ = MSG3.TO.VW$     'SPECIAL MESSAGE FOR VW UNIT - SELF TEST
  TAD$ = CHR$((FROM * 16 + FROM) AND &HFF)
  MSG.FORMAT$ = TAD$ + Packet$ + N.BYTES$ + MSG$
  PRINT #1, MSG.FORMAT$;
  DELAY.TIME! = TIMER
  SAVE.X = POS(0)
  SAVE.Y = CSRLIN
  GOSUB CLEAR.W2
  COLOR 7, 0
  LOCATE 17, 30, 0
  PRINT " TESTING VW UNIT ";
  LOCATE 18, 30
  PRINT " RESET VW AFTER SELF TEST ";      'tweaked 97.05.10
  COLOR 0, 7
  DO WHILE ABS(TIMER - DELAY.TIME!) < 2!  'DELAY FOR TEST
  LOOP
  IF LOC(1) > 0 THEN      'LOOK FOR RESPONSE
    GOSUB READ.BUFFER
    IF LEN(DATAB$) = 5 THEN      'ECHO + RETURNED STRING
      IF LEFT$(DATAB$, 1) = TAD$ AND MID$(DATAB$, 2, 1) = CHR$(0) THEN
        IF RIGHT$(DATAB$, 1) = MSG4.FROM.VW$ THEN 'TEST GOOD
          GOSUB CLEAR.W2
          COLOR 7, 0
          LOCATE 17, 30
          PRINT " VW UNIT TEST OK ";      'tweaked 97.05.10
          COLOR 0, 7
          GOSUB ShowIt      'added 97.05.10
          GOSUB CLEAR.W2    'added 97.05.10
        ELSEIF RIGHT$(DATAB$, 1) = MSG3.FROM.VW$ THEN
          GOSUB CLEAR.W2
          COLOR 7, 0
          LOCATE 17, 30
          PRINT " VW FAILED SELF TEST ";    'tweaked 97.05.10
          COLOR 0, 7
          BEEP
          GOSUB ShowIt      'added 97.05.10
          GOSUB CLEAR.W2    'added 97.05.10
        ELSE
          GOSUB CLEAR.W2      'added 97.05.10
          COLOR 7, 0
          LOCATE 18, 30
          PRINT " INVALID TEST - RETRY ALT-T "
          COLOR 0, 7
          BEEP
          GOSUB ShowIt
          GOSUB CLEAR.W2
        END IF
      ELSE
        GOSUB CLEAR.W2      'added 97.05.10
        COLOR 7, 0
        LOCATE 18, 30
        PRINT " INVALID TEST - RETRY ALT-T "
        COLOR 0, 7
        BEEP
        GOSUB ShowIt
        GOSUB CLEAR.W2
      END IF
    ELSE
      GOSUB CLEAR.W2      'added 97.05.10
      COLOR 7, 0
      LOCATE 18, 30
      PRINT " INVALID TEST - RETRY ALT-T "
      COLOR 0, 7
      BEEP
      GOSUB ShowIt
      GOSUB CLEAR.W2
    END IF
  END IF

```



```

ELSE
    GOSUB CLEAR.W2
    COLOR 7, 0
    LOCATE 17, 30
    PRINT " VW UNIT NOT RESPONDING ";
    COLOR 0, 7
    BEEP
    GOSUB ShowIt
    GOSUB CLEAR.W2
END IF

LOCATE SAVE.Y, SAVE.X, 1, 5, 7
COLOR 0, 7
KY$ = ""
DATAB$ = ""
MSG$ = MSG0.TO.VW$
MSG.FORMAT$ = TAD$ + Packet$ + N.BYTES$ + MSG$
'SPECIAL MESSAGE FOR VW UNIT - RESET

ELSE
    'Not self test
END IF

'----- RESET UNIT -----

IF KY$ = ALT.R$ THEN
    Packet$ = CHR$(0)
    N.BYTES$ = CHR$(1)
    MSG$ = MSG0.TO.VW$
    TAD$ = CHR$((FROM * 16 + FROM) AND &HFF)
    MSG.FORMAT$ = TAD$ + Packet$ + N.BYTES$ + MSG$
    PRINT #1, MSG.FORMAT$;
    SAVE.X = POS(0)
    SAVE.Y = CSRLIN
    GOSUB CLEAR.W2
    COLOR 7, 0
    LOCATE 17, 30, 0
    PRINT " VW RESET COMMAND ";
    GOSUB ShowIt
    GOSUB CLEAR.W2
    LOCATE SAVE.Y, SAVE.X, 1, 5, 7
    COLOR 0, 7
    KY$ = ""
    DATAB$ = ""
ELSE
    END IF

IF KY$ = ALT.X$ THEN
    SAVE.X = POS(0)
    SAVE.Y = CSRLIN
    GOSUB CLEAR.W2
    COLOR 7, 0
    LOCATE 17, 30, 0
    BEEP
    PRINT " EXIT PROGRAM Yes/No ";
    COLOR 0, 7
    KY$ = ""
    DO WHILE KY$ = ""
        KY$ = INKEY$
    LOOP
    IF KY$ = "Y" OR KY$ = "y" THEN
        COLOR 7, 0
        CLS
        END
        ELSE
            GOSUB CLEAR.W2
        END IF
        LOCATE SAVE.Y, SAVE.X, 1, 5, 7
        COLOR 0, 7
        KY$ = ""
    ELSE
        END IF

IF KY$ = ALT.C$ THEN
    START.OF.PACKET = 0
    PACKET.PTR = 1
    BUSY = 0

```

```

COLOR 7, 0
CLS
GOTO MAIN
ELSE
END IF

IF KY$ = ALT.H$ THEN
COLOR 7, 0
CLS
LOCATE 1, 1, 0
PRINT "<KEY> "
PRINT
PRINT "<DEL> Deletes line."
PRINT "<BACKSPACE> Deletes character to left."
PRINT "<ENTER> At start of line sends last message."
PRINT "<ALT>+<C> Clears screen."
PRINT "<ESC> Resets "; CHR$(&H22); "SEND PACKET"; CHR$(&H22); " Mode."
PRINT
PRINT "<ALT>+<S> Invokes set up program."
PRINT "<ALT>+<X> Returns to system."
PRINT "<ALT>+<I> Increments "; CHR$(&H22); "TO"; CHR$(&H22); " address."
PRINT "<ALT>+<D> Decrements "; CHR$(&H22); "TO"; CHR$(&H22); " address."
PRINT
PRINT "<ALT>+<B> Broadcast i.e. send message to all VW units."
PRINT "<ALT>+<T> Test - VW unit to perform selftest - cycle power after test."
PRINT "<ALT>+<R> Reset - VW unit to start over (warm boot)."
PRINT "<ALT>+<V> VW unit battery voltage test."
PRINT
PRINT "<ALT>+<A> Read VW address."
PRINT "<F1> To send a full buffer."
PRINT "<CTRL>+<T> For Telemetry."
PRINT "<CTRL>+<N> To change Node address."

LOCATE 23, 30
PRINT "<ANY KEY> to continue ..."

KY$ = ""
DO WHILE KY$ = ""
    KY$ = INKEY$
LOOP

CLS
GOTO MAIN

ELSE
    'Not ALT-H
END IF

IF KY$ = ALT.I$ OR KY$ = ALT.D$ THEN
    IF KY$ = ALT.I$ THEN
        TO.ADDR = TO.ADDR + 1
        IF TO.ADDR > 15 THEN
            TO.ADDR = 1
        ELSE
            END IF
    ELSE
        TO.ADDR = TO.ADDR - 1
        IF TO.ADDR = 0 THEN
            TO.ADDR = 15
        ELSE
            END IF
    END IF

    TO.ADDR$ = CHR$(TO.ADDR + 48)      'tweaked 97.05.10
    SAVE.X = POS(0)
    SAVE.Y = CSRLIN

    LOCATE 1, 4
    COLOR 0, 7
    PRINT " ";
    PRINT TO.ADDR;
    PRINT " ";
    LOCATE SAVE.Y, SAVE.X
    KY$ = ""
ELSE

```

```

END IF

IF KY$ = ALT.S$ THEN
    GOSUB SET.CONFIG
    GOTO MAIN
ELSE
END IF

IF BUSY > 0 THEN
    GOTO MINI      'WAIT FOR COMMUNICATIONS TO COMPLETE
ELSE
END IF

IF KY$ = DEL$ THEN
    KY$ = ""
    LOCATE 23, 1, 1, 5, 7
    PRINT STRING$(80, " ")
    LOCATE 23, 1, 1, 5, 7
    MESSAGE$ = ""
ELSE
END IF

IF KY$ = BACK$ THEN 'DELETE LAST CHARACTER
    KY$ = ""
    LT = LEN(MESSAGE$)
    IF LT > 0 THEN
        MESSAGE$ = LEFT$(MESSAGE$, LT - 1)
        PS = POS(0)
        IF PS > 1 THEN
            PS = PS - 1
            LOCATE CSRLIN, PS: PRINT " ";
            LOCATE CSRLIN, PS
        ELSE
            IF CSRLIN > 20 THEN
                LN1 = CSRLIN - 1
                LOCATE LN1, 80
                PRINT " ";
                LOCATE LN1, 80
            ELSE
            END IF
        END IF
    END IF
ELSE
END IF

IF KY$ = CR$ THEN
    IF MESSAGE$ = "" THEN
        MESSAGE$ = LAST.MSG$
    ELSE
    END IF
    IF TELEMETRY = 1 THEN
        IF LEN(MESSAGE$) > 30 THEN
            MESSAGE$ = LEFT$(MESSAGE$, 30)
        ELSE
        END IF
    END IF
    GOSUB BUILD.MSG
    PACKET.PTR = 1          'POINTS TO ARRAY TO SEND
    GOSUB SEND.PACKET
    KY$ = ""
ELSE
END IF

IF KY$ = F1$ THEN
    MESSAGE$ = FULL.MSG$
    LRDM$ = LEFT$(FULL.MSG$, 1)
    FULL.MSG$ = RIGHT$(FULL.MSG$, LEN(FULL.MSG$) - 1) + LRDM$
    IF TELEMETRY = 1 THEN
        MESSAGE$ = LEFT$(FULL.MSG$, 30)
    END IF
    GOSUB BUILD.MSG
    PACKET.PTR = 1          'POINTS TO ARRAY TO SEND
    GOSUB SEND.PACKET
    KY$ = ""
ELSE

```

```

END IF

IF KY$ >= BLANK$ THEN
    MESSAGE$ = MESSAGE$ + KY$      'ADD CHARACTER TO MESSAGE
    PRINT KY$;                     'PUT IT ON THE SCREEN

    IF LEN(MESSAGE$) = 79 OR ((BROADCAST = 1 OR TELEMETRY = 1) AND LEN(MESSAGE$) = 30) THEN
        BEEP
        GOSUB BUILD.MSG
        PACKET.PTR = 1              'POINTS TO ARRAY TO SEND
        GOSUB SEND.PACKET
        DO UNTIL KY$ = ""
            KY$ = INKEY$            'EMPTY BUFFER
        LOOP
    ELSE
        END IF
        KY$ = ""
    ELSE
        END IF

    GOTO MINI

'***** SET UP DISK DRIVE FOR SAVING DATA FILES *****

SETUP.DSK:

'READ SETUP FILE FROM DISK (edited 97.05.10)

FAULT = 0      'LOOK FOR ERROR CODE

OPEN "VWT97.CFG" FOR INPUT AS #2

IF FAULT = 0 THEN 'FILE IS OPEN
    INPUT #2, COM.PORT$, BAUD.RATE$, TO.ADDR$      'READ CONFIG. VALUES
    TO.ADDR = ASC(TO.ADDR$) - 48
    CLOSE #2
ELSE
    CLOSE #2
    GOSUB SET.CONFIG
END IF

RETURN

'----- SET CONFIGURATION -----

SET.CONFIG:

COLOR 7, 0
CLS
LOCATE 10, 1
PRINT "ADDRESS OF VW UNIT YOU WANT TO TALK TO: "; TO.ADDR$; 'tweaked 97.05.10

LOCATE CSRLIN, 50
PRINT "<1> TO CHANGE."
PRINT
PRINT "YOUR CURRENT COM PORT - "; COM.PORT$; 'tweaked 97.05.10

LOCATE CSRLIN, 50
PRINT "<2> TO CHANGE."
PRINT
PRINT "YOUR CURRENT BAUD RATE IS (MATCH TO VW): "; BAUD.RATE$;

LOCATE CSRLIN, 50
PRINT "<3> TO CHANGE."
PRINT
PRINT "TO EXIT";

LOCATE CSRLIN, 50
PRINT "<ESC>"

LOCATE 20, 1

SETLP0:
KY$ = ""

```

```

DO WHILE KY$ = ""
  KY$ = INKEY$
LOOP
IF KY$ = "1" THEN
  INPUT "ENTER ADDRESS YOU WANT TO TALK TO (1 - 15): ", TO.ADDR$ 'tweaked 97.05.10
  TO.ADDR = VAL(TO.ADDR$) AND &HF
  GOTO SET.CONFIG
ELSE
  END IF

IF KY$ = "2" THEN
  IF COM.PORT$ = "COM1:" THEN
    COM.PORT$ = "COM2:"
  ELSE
    COM.PORT$ = "COM1:"
  END IF
  GOTO SET.CONFIG
ELSE
  END IF

IF KY$ = "3" THEN
  IF BAUD.RATE$ = "4800" THEN
    BAUD.RATE$ = "9600"
  ELSEIF BAUD.RATE$ = "9600" THEN
    BAUD.RATE$ = "19200"
  ELSEIF BAUD.RATE$ = "19200" THEN
    BAUD.RATE$ = "4800"
  END IF
  GOTO SET.CONFIG
ELSE
  END IF

IF KY$ = ESC$ THEN
  CLOSE #1

  OPEN COM.PORT$ + BAUD.RATE$ + ",N,8,1,RS,CD0,DS0,CS0" FOR RANDOM AS #1
  CLS
  LOCATE 10, 10
  PRINT "SAVE CONFIGURATION VALUES TO DISK      Y/N      ";

LP01:
  KY$ = ""
  DO WHILE KY$ = ""
    KY$ = INKEY$
  LOOP

  IF KY$ = "Y" OR KY$ = "y" THEN
    GOSUB SAVE.CONFIG
    RETURN
  ELSE
    END IF

  IF KY$ = "N" OR KY$ = "n" THEN
    RETURN
  ELSE
    END IF

ELSE
  END IF

GOTO SETLP0

'-----SAVE CONFIGURATION FILE -----

SAVE.CONFIG:

  FAULT = 0
  OPEN "VWT97.CFG" FOR OUTPUT AS #2
  IF FAULT = 0 THEN 'FILE IS OPEN
    PRINT #2, CHR$(34); COM.PORT$; CHR$(34); CHR$(34); BAUD.RATE$; CHR$(34); CHR$(34); TO.ADDR$;
    CHR$(34)
  ELSE
    END IF
  CLOSE #2

```

RETURN

'----- READ COMMUNICATIONS BUFFER HERE AND BUILD STRING -----'

'NOTE: A STRING BUILT AS FOLLOWS - IF THE FIRST CHARACTER IS A NULL  
'WILL BE DISPLAYED AS ALL NULLS BY THE QB4.5 DEBUGGER. THE LENGTH IS  
'CORRECT AND IS THE ONLY CLUE THAT THE DATA IS REALLY IN THE STRING.  
'A LESSON FROM THE SCHOOL OF HARD KNOCKS.

READ.BUFFER:

```
'---- CHANGE TO HANDLE NULL CHARACTER
DO WHILE LOC(1) > 0
  R.DATAB$ = R.DATAB$ + INPUT$(LOC(1), #1)
  COM.TIME! = TIMER
  DO
  LOOP UNTIL ABS(TIMER - COM.TIME!) > .01  'MAKE SURE ALL RECEIVED
LOOP

IF LEN(R.DATAB$) > 35 THEN
  DATAB$ = LEFT$(R.DATAB$, 35)
  R.DATAB$ = RIGHT$(R.DATAB$, LEN(R.DATAB$) - 35)
ELSE
  DATAB$ = R.DATAB$
  R.DATAB$ = ""
END IF
```

RETURN

'----- LINE EDITOR SCREEN -----'

SCREEN1:

```
CLS
PRINT "TO:          FROM (My address):";
COLOR 0, 7
PRINT " ";
PRINT FROM;
PRINT " ";
LOCATE 1, 4
PRINT " ";
PRINT TO.ADDR;
PRINT " ";
COLOR 7, 0

LOCATE 4, 1
PRINT "MESSAGES RECEIVED";
GOSUB CLEAR.W1
LOCATE 15, 1
PRINT "MESSAGES SENT          PACKET#          STATUS";
GOSUB CLEAR.W2
LOCATE 21, 1
PRINT "ENTER MESSAGE TO SEND";
GOSUB CLEAR.W3
COLOR 0, 7
LOCATE 25, 1
PRINT " ALT+H FOR HELP ";
LOCATE 25, 18
PRINT " ALT+I or +D "; CHR$(&H22); "TO"; CHR$(&H22); " addr. ";
LOCATE 25, 44
PRINT " ALT+X EXIT PGM. ";
LOCATE 25, 62
PRINT " ALT+B - BROADCAST ";

IF BROADCAST = 0 THEN
  COLOR 7, 0
  LOCATE 3, 30
  PRINT "          ";
  COLOR 0, 7
ELSE
  LOCATE 3, 30
  PRINT " BROADCAST MODE ENABLED ";
END IF
```

```

IF TELEMETRY = 0 THEN
    COLOR 7, 0
    LOCATE 3, 30, 0
    PRINT "                                ";
    COLOR 0, 7
ELSE
    LOCATE 3, 30, 0
    PRINT "    TELEMETRY ENABLED    ";
END IF

LOCATE 23, 1, 1, 5, 7

RETURN

'----- CLEAR RECEIVE WINDOW -----

CLEAR.W1:

COLOR 0, 7
LOCATE 5, 1
FOR A = 1 TO 10
    PRINT STRING$(80, " ")
NEXT A
COLOR 7, 0
RETURN

'----- CLEAR SEND WINDOW -----

CLEAR.W2:

COLOR 0, 7
LOCATE 16, 1
FOR A = 1 TO 4
    PRINT STRING$(80, " ")
NEXT A
COLOR 7, 0
RETURN

'----- CLEAR EDIT WINDOW -----

CLEAR.W3:

COLOR 0, 7
LOCATE 22, 1
FOR A = 1 TO 2
    PRINT STRING$(80, " ")
NEXT A
COLOR 7, 0
RETURN

'----- BUILD MESSAGE ARRAY TO SEND TO VW UNIT -----

'EACH ARRAY ELEMENT WILL BE <= TO 32 CHARACTERS. THIS IS THE LENGTH OF
'THE MESSAGE BUFFER IN THE VW UNIT.
'THE TOTAL MESSAGE TO BE SENT IS BUFFERED BY ASCII CONTROL CHARACTERS "STX"
'AND "ETX".
'EXAMPLE: A 64 BYTE MESSAGE IS TO BE FORMATED. IT WILL BE DIVIDED INTO 3
'STRINGS AS FOLLOWS:
'1. STX + FIRST 31 CHARACTERS.
'2. NEXT 32 CHARACTERS.
'3. LAST CHARACTER + ETX.

BUILD.MSG:

NUMBER.TO.SEND = 1    'NUMBER OF PACKETS
IF LEN(MESSAGE$) > 30 THEN 'LONGER THEN 1 PACKET
    MINI.MSG$(1) = STX$ + LEFT$(MESSAGE$, 31)
    MESSAGE$ = RIGHT$(MESSAGE$, (LEN(MESSAGE$) - 31))

    L = LEN(MESSAGE$) \ 32
    B = (LEN(MESSAGE$) MOD 32)

```

```

        IF L = 0 THEN
            NUMBER.TO.SEND = 2
        ELSE
            NUMBER.TO.SEND = L + 2
        END IF

    IF L > 0 THEN
        FOR A = 1 TO L
            MINI.MSG$(A + 1) = LEFT$(MESSAGE$, 32)
            MESSAGE$ = RIGHT$(MESSAGE$, (LEN(MESSAGE$) - 32))
        NEXT A
    ELSE
        END IF
    IF B > 0 THEN
        MINI.MSG$(L + 2) = RIGHT$(MESSAGE$, B) + ETX$
    ELSE
        MINI.MSG$(L + 2) = ETX$
    END IF
ELSE
    MINI.MSG$(1) = STX$ + MESSAGE$ + ETX$
END IF

GOSUB CLEAR.W2
LOCATE 16, 1
COLOR 0, 7
FOR A = 1 TO NUMBER.TO.SEND
    PRINT MINI.MSG$(A);
    LOCATE CSRLIN, 42
    PRT = Packet + A - 1
    IF TELEMETRY = 1 THEN
        PRINT 8
    ELSE
        IF PRT < 8 THEN
            PRINT PRT
        ELSE
            PRINT PRT - 7
        END IF
    END IF
NEXT A

GOSUB CLEAR.W3
COLOR 0, 7
LOCATE 23, 1, 1, 5, 7
IF TELEMETRY = 0 THEN
    PRINT "BUSY SENDING DATA";
ELSE
    END IF

LAST.MSG$ = ""
FOR A = 1 TO NUMBER.TO.SEND
    LAST.MSG$ = LAST.MSG$ + MINI.MSG$(A)      'SAVE FOR RETRANSMIT
NEXT A
IF RIGHT$(LAST.MSG$, 1) = ETX$ THEN
    LAST.MSG$ = LEFT$(LAST.MSG$, LEN(LAST.MSG$) - 1)  'REMOVE ETX
ELSE
    END IF
IF LEFT$(LAST.MSG$, 1) = STX$ THEN
    LAST.MSG$ = RIGHT$(LAST.MSG$, LEN(LAST.MSG$) - 1)  'REMOVE STX
ELSE
    END IF
MESSAGE$ = ""
RETURN

'----- TRANSMIT PACKET -----

SEND.PACKET:

IF PACKET.PTR <= NUMBER.TO.SEND THEN
    IF TELEMETRY = 1 THEN
        RETURN.FROM.SEND = 1
    ELSE
        RETURN.FROM.SEND = 0
    END IF

    TO.FROM$ = CHR$((TO.ADDR * 16 + FROM) AND &HFF)

```



```

IF BROADCAST = 1 THEN      'SEND TO ALL VW UNITS.
    TO.NAME$ = CHR$(0)     'T/F = 00
ELSE
    TO.NAME$ = TO.FROM$    'SEND TO ADDRESSED UNIT
END IF
IF TELEMETRY = 1 THEN
    Packet = 8
ELSE
END IF

MSG.FORMAT$ = TO.NAME$ + CHR$(Packet AND &HF)      'ADD PACKET#
MSG.FORMAT$ = MSG.FORMAT$ + CHR$(LEN(MINI.MSG$(PACKET.PTR)) AND &HFF)'#BYTES
MSG.FORMAT$ = MSG.FORMAT$ + MINI.MSG$(PACKET.PTR)  'ADD MESSAGE
TRY$ = 0

SEND.AGAIN:
TRY$ = TRY$ + 1
PRINT #1, LEFT$(MSG.FORMAT$, 1); 'SEND T/F TO GET ATTENTION OF VW UNIT
OK = 2
DELAY.TIME! = TIMER

DO WHILE ABS(TIMER - DELAY.TIME!) < .5    'DELAY FOR RESPONSE
    IF LOC(1) > 0 THEN
        DATAB$ = INPUT$(LOC(1), #1)
        'GOSUB READ.BUFFER ***OLD CODE; NOT FAST ENOUGH FOR DR12xx-DK KITS
        '-- A VW UNIT ALWAYS ECHOS FIRST CHARACTER IF NOT BUSY
        L = LEN(DATAB$)
        IF LEN(DATAB$) = 1 AND DATAB$ = TO.NAME$ THEN 'GOT VW'S ATTENTION
            DATAB$ = ""
            '--- SEND REMAINDER OF PACKET
            PRINT #1, RIGHT$(MSG.FORMAT$, LEN(MSG.FORMAT$) - 1); 'SEND IT
            BUSY = 1
            OK = 0
            EXIT DO
        ELSE
            OK = 1
        END IF
    ELSE
        OK = 1
    END IF
END IF
LOOP

IF OK = 1 THEN 'T/F DIDN'T GET THROUGH
    IF LEN(DATAB$) = 1 THEN 'VW WAS BUSY AT TIME AND PICKED UP
        DATAB$ = ""
        DELAY.TIME! = TIMER
        '--- DELAY WHILE VW UNIT RECOVERS
        DO WHILE ABS(TIMER - DELAY.TIME!) < .5    'DELAY FOR RESPONSE
            LOOP
        GOTO SEND.AGAIN      'PARTIAL DATA.
    ELSE
        GOSUB RECEIVE.DATA  'LONGER THEN ONE BYTE IS INCOMING PACKET.
    END IF
ELSE
END IF
IF OK = 2 THEN 'NO RESPONSE TIME OUT - TRY AGAIN
    IF TRY$ < 10 THEN
        DELAY.TIME! = TIMER
        '--- DELAY WHILE VW UNIT RECOVERS
        DO WHILE ABS(TIMER - DELAY.TIME!) < .5    'DELAY FOR RESPONSE
            LOOP
        GOTO SEND.AGAIN
    ELSE
        BUSY = 0
        NUMBER.TO.SEND = 0
        COLOR 7, 1
        GOSUB CLEAR.W2
        BEEP
        LOCATE 17, 5
        COLOR 0, 7
        PRINT "TIME OUT - VW UNIT NOT RESPONDING."
        GOSUB CLEAR.W3
        COLOR 0, 7
        LOCATE 23, 1, 1, 5, 7
    END IF
ELSE

```

```

        END IF
    ELSE
        BUSY = 0
    END IF

RETURN

'----- RECEIVE PACKETS AND DISPLAY -----

RECEIVE.DATA:

IF LEN(DATAB$) >= 4 THEN 'LENGTH CORRECT FOR RECEIVED PACKET

    TT = ((ASC(LEFT$(DATAB$, 1)) AND &HF0) / 16)
    '---THIS GETS AROUND NULL CHAR. PROBLEM WITH DEBUGGER

    IF TT = FROM OR LEFT$(DATAB$, 1) = CHR$(0) THEN 'MY ADDR.

        '--- SEE IF MESSAGE LENGTH = NUMBER OF BYTES

        IF ASC(MID$(DATAB$, 3, 1)) = LEN(RIGHT$(DATAB$, LEN(DATAB$) - 3)) THEN

            SAVE.Y = CSRLIN
            SAVE.X = POS(0)

            '---SCROLL RECEIVE WINDOW IF NEEDED

            GOSUB SCROLL.WINDOW1

            LOCATE RCV.PTRY, RCV.PTRX, 0

            PRINT DATAB$

            IF MID$(DATAB$, 4, 1) = STX$ THEN
                IF START.OF.PACKET = 1 OR LEN(PRTMSG$) > 0 THEN 'A PACKET WAS LOST
                    COLOR 7, 0
                    PRINT " PACKET LOST @ 1 "
                    COLOR 0, 7
                    BEEP

                    START.OF.PACKET = 0 'link overload com reboot 97.05.15
                    PRTMSG$ = ""
                    GOSUB BootCom
                    GOTO MAIN

                    RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS
                    RCV.PTRX = POS(0)
                    GOSUB SCROLL.WINDOW1
                ELSE
                    END IF

                START.OF.PACKET = 1 'USE TO DETECT LOST PACKET
                PRTMSG$ = RIGHT$(DATAB$, LEN(DATAB$) - 4) 'REMOVE STX
                IF RCV.PTRX > 1 THEN
                    'PRTMSG$ = CR$ + PRTMSG$ 'STX IN MIDDLE OF LINE IS A NEW START
                    PRINT CR$;
                    RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS
                    RCV.PTRX = POS(0)
                    GOSUB SCROLL.WINDOW1
                ELSE
                    END IF
            ELSEIF START.OF.PACKET = 1 THEN
                PRTMSG$ = PRTMSG$ + RIGHT$(DATAB$, LEN(DATAB$) - 3) 'USE AS IS
            ELSE
                COLOR 7, 0
                PRINT " PACKET LOST @ 2 "
                COLOR 0, 7
                BEEP

                START.OF.PACKET = 0 'link overload com reboot 97.05.15
                PRTMSG$ = ""
                GOSUB BootCom
                GOTO MAIN

                RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS

```

```

        RCV.PTRX = POS(0)
        GOSUB SCROLL.WINDOW1
        START.OF.PACKET = 0
        PRMSG$ = ""
    END IF

    IF RIGHT$(PRMSG$, 1) = ETX$ THEN 'REMOVE IT AND ADD RETURN.
        PRMSG$ = LEFT$(PRMSG$, LEN(PRMSG$) - 1) 'REMOVE ETX
        IF LEN(PRMSG$) > 79 THEN
            COLOR 7, 0
            PRINT " PACKET LOST @ 3 "
            COLOR 0, 7
            BEEP
            RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS
            RCV.PTRX = POS(0)
            GOSUB SCROLL.WINDOW1
            START.OF.PACKET = 0
            PRINT RIGHT$(PRMSG$, 79)
            RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS
            RCV.PTRX = POS(0)
            PRMSG$ = ""
        ELSE
            PRINT PRMSG$ 'PUT MESSAGE ON SCREEN
            RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS
            RCV.PTRX = POS(0)
            START.OF.PACKET = 0
            PRMSG$ = ""
        END IF
    ELSE
    END IF

    RCV.PTRY = CSRLIN 'UPDATE RECEIVE WINDOW POINTERS
    RCV.PTRX = POS(0)

    LOCATE SAVE.Y, SAVE.X, 1, 5, 7 'RESTORE SCREEN LOCATION

ELSE
END IF
ELSE
END IF
ELSE
END IF
END IF

DATAB$ = ""

RETURN

'----- SCROLL RECEIVE WINDOW -----

SCROLL.WINDOW1:

    IF RCV.PTRY >= 15 THEN
        RCV.PTRY = 14 'LIMIT TO LAST LINE
        RCV.PTRX = 1
        FOR A = 5 TO 14
            LOCATE A, 1, 0
            PRINT STRING$(80, " ");
            LOCATE A, 1
            FOR B = 1 TO 80
                PRINT CHR$(SCREEN(A + 1, B));
            NEXT B
        NEXT A
        LOCATE 14, 1
        PRINT STRING$(80, " ");
        LOCATE RCV.PTRY, RCV.PTRX
    ELSE
    END IF

RETURN

'----- GET ADDRESS OF VIRTUAL WIRE UNIT -----

GET.ADDRESS:

Try:

```

```

TryIt = 0

Retry:

TryIt = TryIt + 1

LOCATE 10, 20, 0
PRINT "POLLING FOR VIRTUAL WIRE ADDRESS, TRY #"; TryIt;

IF LOC(1) > 0 THEN
    DATAB$ = INPUT$(LOC(1), #1) 'CLEAR COMM. BUFFER
ELSE
    END IF

DATAB$ = ""

Packet$ = CHR$(0) 'PACKET 0
N.BYTES$ = CHR$(1) '1 BYTE MESSAGE
MSG$ = MSG1.TO.VW$ 'SPECIAL MESSAGE FOR VW UNIT - SEND ADDR.

FROM = 0

TO.FROM$ = CHR$(0) 'GET ADDR. USES T/F =0 AND PACKET = 0
MSG.FORMAT$ = TO.FROM$ + Packet$ + N.BYTES$ + MSG$
PRINT #1, MSG.FORMAT$;

DELAY.TIME! = TIMER
DO WHILE ABS(TIMER - DELAY.TIME!) < .5 'DELAY FOR RESPONSE
LOOP

IF LOC(1) > 0 THEN

    GOSUB READ.BUFFER

    '-- VW UNIT ALWAYS ECHOS FIRST CHARACTER IF NOT BUSY

    P = INSTR(1, DATAB$, MSG5.FROM.VW$)
    IF P >= 5 THEN 'VW address message came back
        FROM$ = MID$(DATAB$, P - 3, 1)'BACK UP TO T/F
        FROM = ASC(FROM$) AND &HF 'got the FROM address from the VW unit
    ELSE 'did not get address
        END IF

ELSE 'nothing in input buffer
    END IF

DATAB$ = ""

IF FROM = 0 THEN 'did not get FROM address, so

    IF TryIt < 8 THEN
        GOTO Retry 'retry several times automatically
    END IF

    CLS 'auto retry did not help, let user know
    LOCATE 10, 5
    PRINT "VW unit not responding - check power, cables, com port, heavy RF noise"
    LOCATE 11, 5
    PRINT "<R> for retry, <ESC> to main program (then <ALT-S> for configuration set up)"

    KY$ = ""
    DO WHILE KY$ = ""
        KY$ = INKEY$
    LOOP

    IF KY$ = "r" OR KY$ = "R" THEN
        CLS 'added 97.05.10
        GOTO Try 'manual retry
    ELSE 'drop out to main program for configuration set up
        END IF

    END IF

RETURN

```

```

TOADDR.MSG:      'added 97.05.10

COLOR 15, 1
CLS

LOCATE 10, 20, 0
PRINT "Who do you want to talk to today??"
LOCATE 12, 20, 0
PRINT " Double check your TO Address!!"
BEEP
DELAY.TIME! = TIMER
DO WHILE ABS(TIMER - DELAY.TIME!) < 2   'DELAY FOR RESPONSE
LOOP

COLOR 7, 0
CLS

RETURN

ShowIt:          'added 97.05.10

DELAY.TIME! = TIMER
DO WHILE ABS(TIMER - DELAY.TIME!) < 1   'DELAY FOR RESPONSE
LOOP

RETURN

BootCom:
  OPEN COM.PORT$ + BAUD.RATE$ + ",N,8,1,RS,CD0,DS0,CS0" FOR RANDOM AS #1 LEN = 2048
RETURN

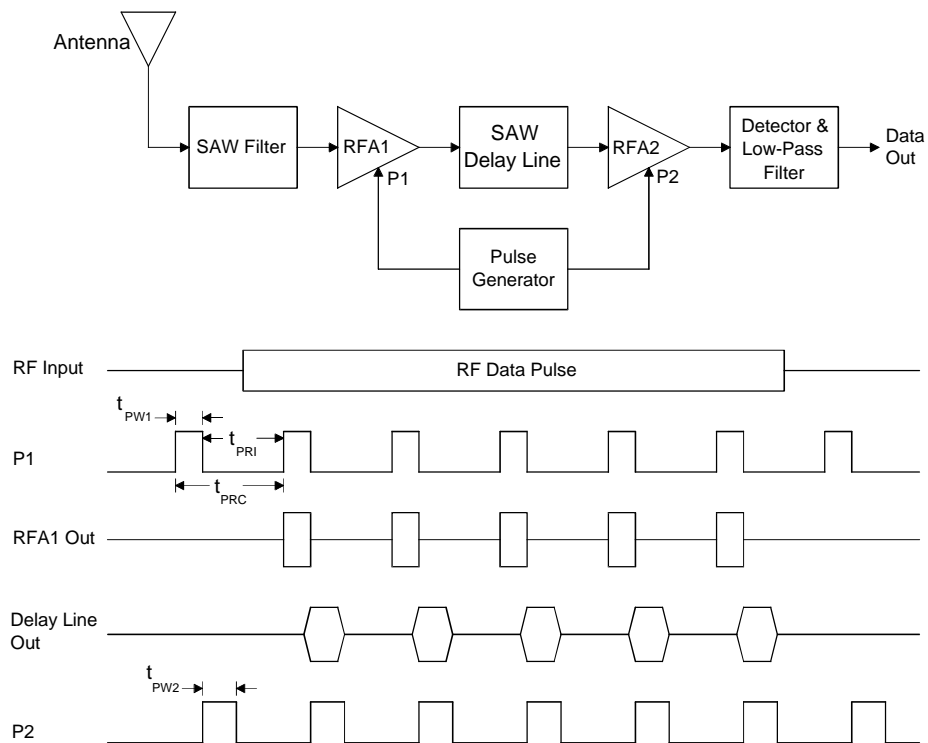
' ***** ERROR RECOVERY *****

PRTERRO:
FAULT = ERR
RESUME NEXT

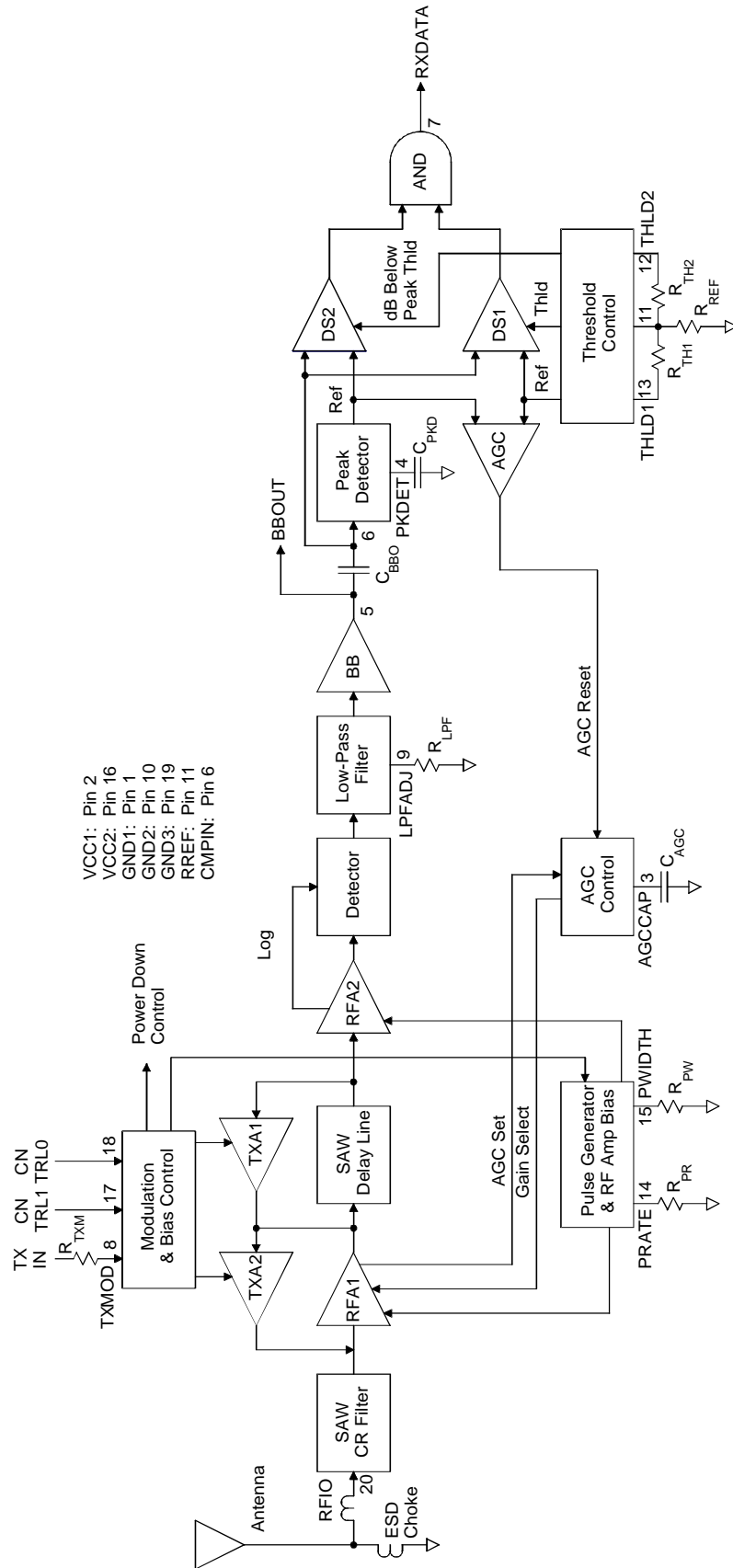
END

```

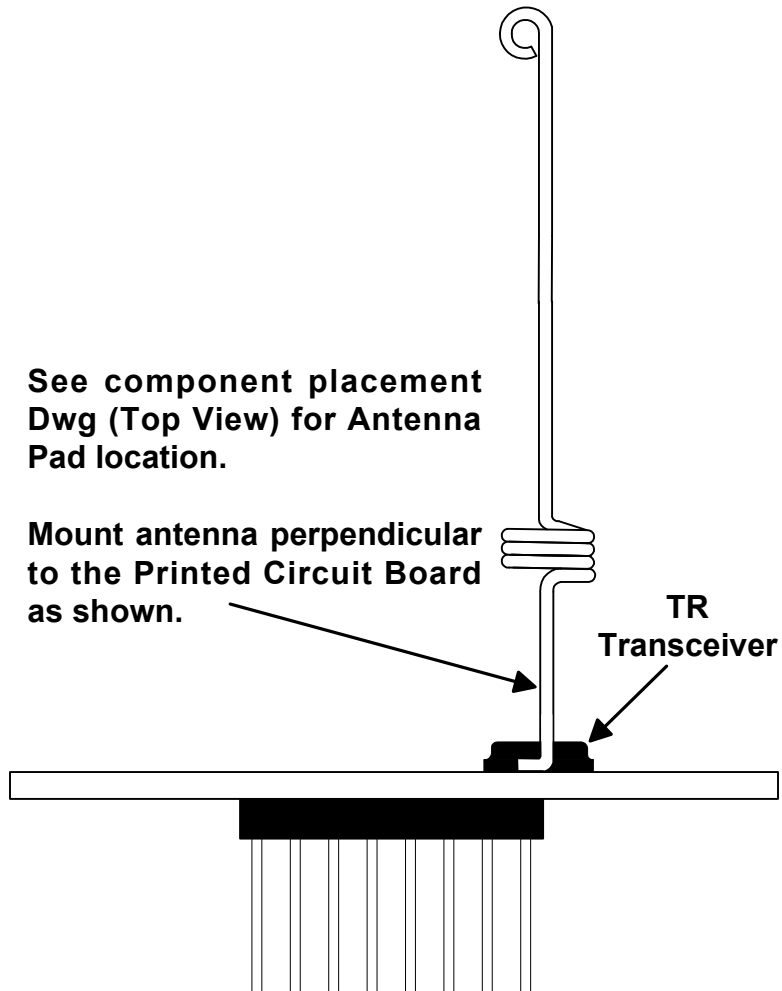
## ASH Receiver Block Diagram & Timing Cycle



## ASH Transceiver Block Diagram



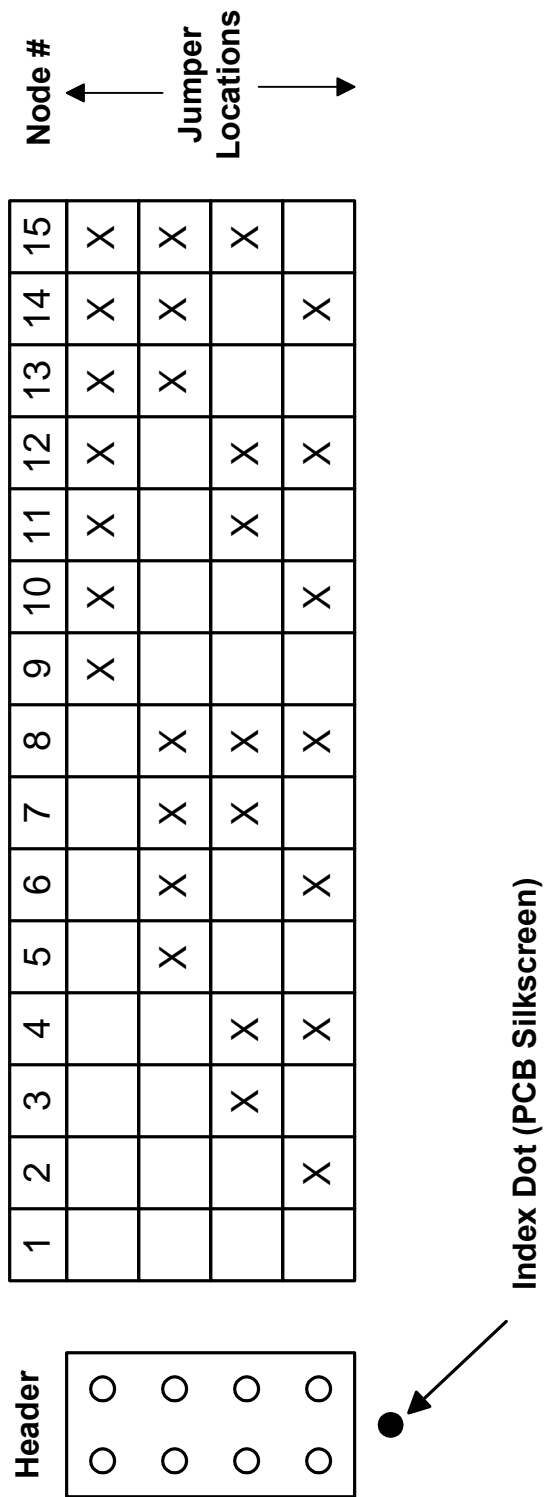
## Antenna Mounting Detail

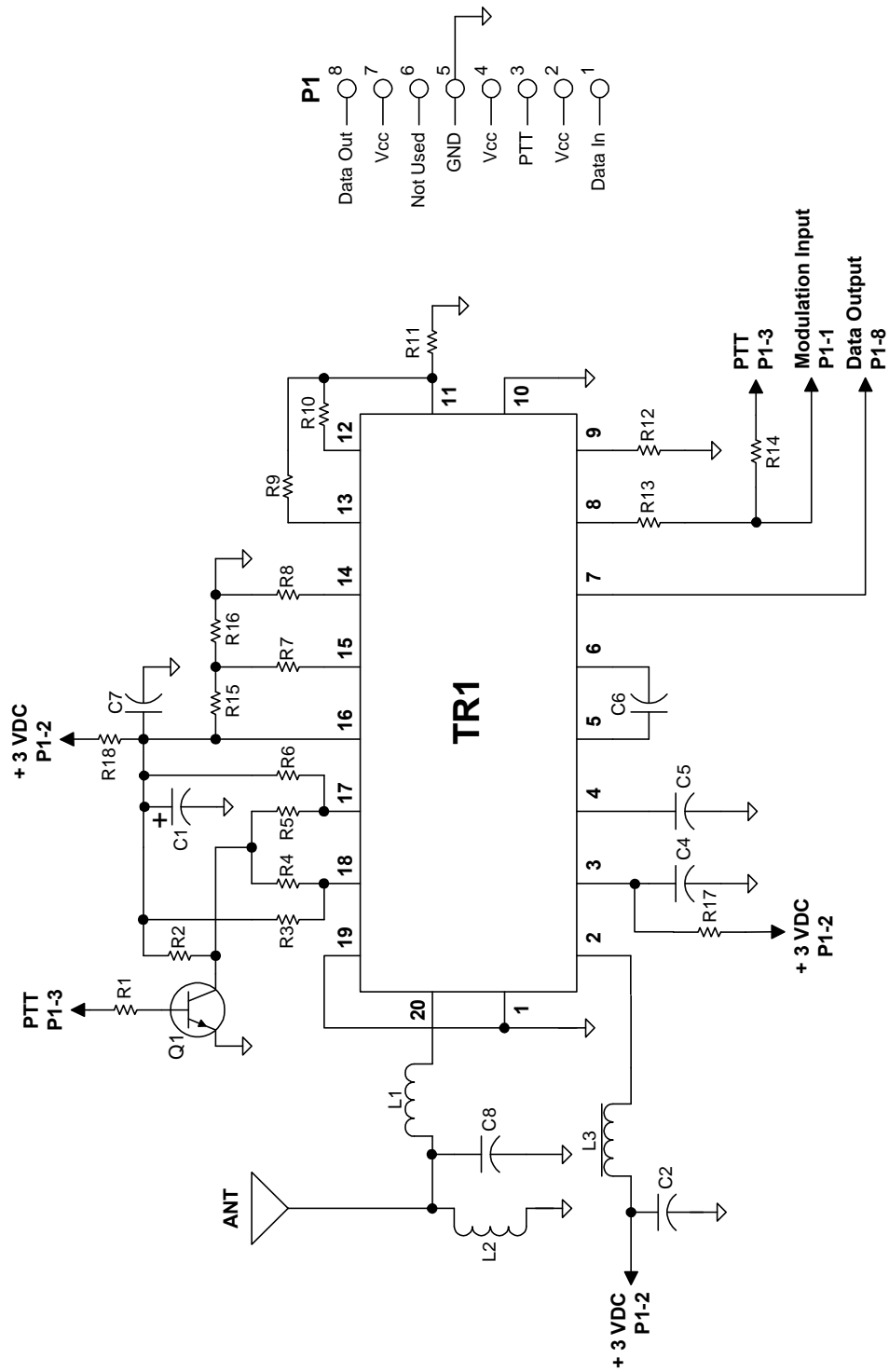


**View From Antenna Port of PCB**



# **Node Programming Jumper Locations** **Node # = binary + 1**



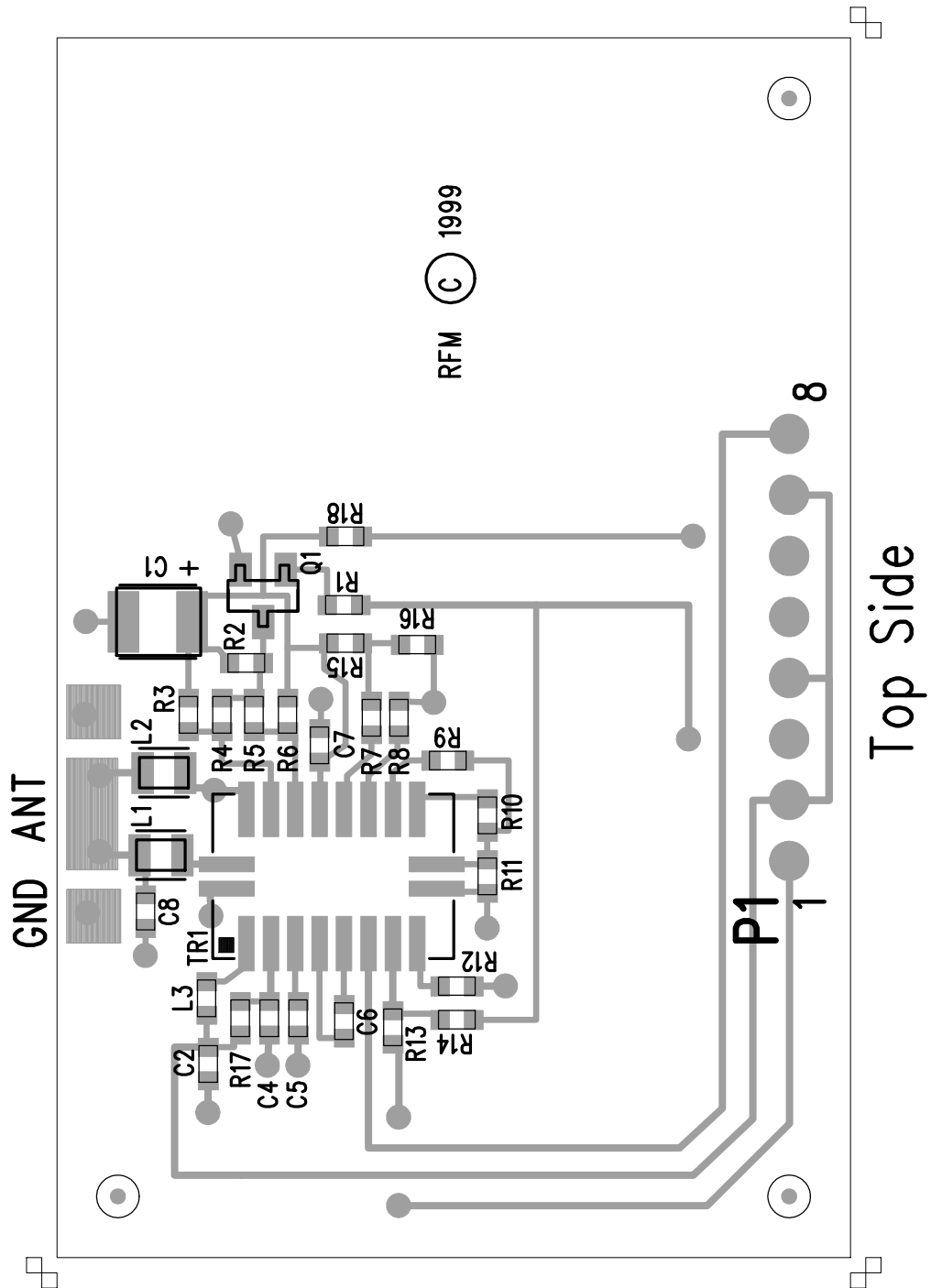


Schematic, Data Radio  
Date: 03/02/1999, LM

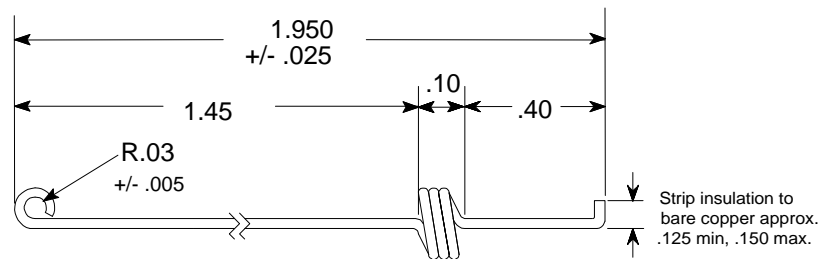
## DR1200/DR1201 Bill of Materials

Ref Des	Qty	MURATA P/N	Description
PCB1	1	400-1427-001X5	Printed Circuit Board
TR1	1	TR1000/TR1001	Xcvr, TR1000 for DR1200, TR1001 for DR1201
L1	2	500-0583-100	Inductor, SMT, 10 nH, (Coilcraft 0805HT-10NTJ)
L2	1	500-0583-101	Inductor, SMT, 100 nh, (Coilcraft 0805CS-101TK)
L3	1	500-0834-001	Ferrite, Chip Bead (Fair-Rite 2506033017YO)
Q1	1	500-0183-001	Xstr, SOT, MMBT222L
C1	1	500-0675-106	Capacitor, SMT, 10uf, Kemet T491B106K006AS
C2	1	500-0621-270	Capacitor, SMT, 27 pF, 5%, 0603
C6	1	500-0621-153	Capacitor, SMT, 0.015 uF, %10, 0603
C7	1	500-0621-101	Capacitor, SMT, 100 pF, 5%, 0603
R1	1	500-0620-273	Res, Chip, 27K, .1W, 5%, 0603
R2	1	500-0620-473	Res, Chip, 47K, .1W, 5%, 0603
R3, R5, R16, R17	1	500-0620-001	Res, Chip, 0.0, .1W, 5%, 0603
C3, C4, C5, C8, R4, R6, R10, R14, R15,	0	000-0000-00	Not Used on DR1200
R7, R8	2	500-0620-274	Res, Chip, 270K, .1W, 5%, 0603
R9, R11	2		Res, Chip, 100K, .1W, 1%, 0603
R12	1	500-0620-303	Res, Chip, 30K, .1W, 5%, 0603
R13	1	500-0620-472	Res, Chip, 4.7K, .1W, 5%, 0603
R18	1	500-0620-101	Res, Chip, 100, .1w, 5%, 0603
P1	1	500-0644-001	HDR, 8Pin

# Data Radio Component Placement



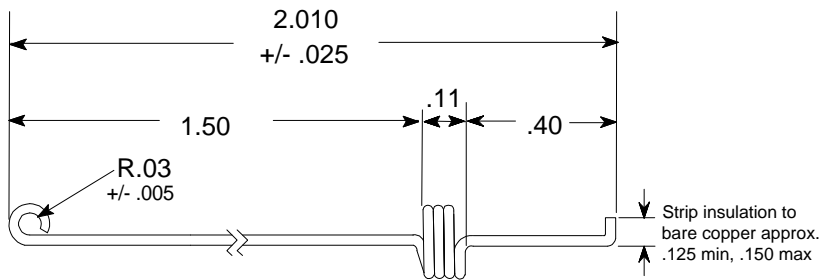
## 916.5 MHz Test Antenna Drawing



Not drawn to scale. Units in inches.  
22 AWG insulated solderable magnet wire.  
3 turns close wound on .130 in. dia.  
Finished ID = .130, +/- .003

916.5 MHZ ANT  
7/07/98 LAM  
(c) 1998 RFM  
400-1309-001

## 868.35 Mhz Test Antenna Drawing

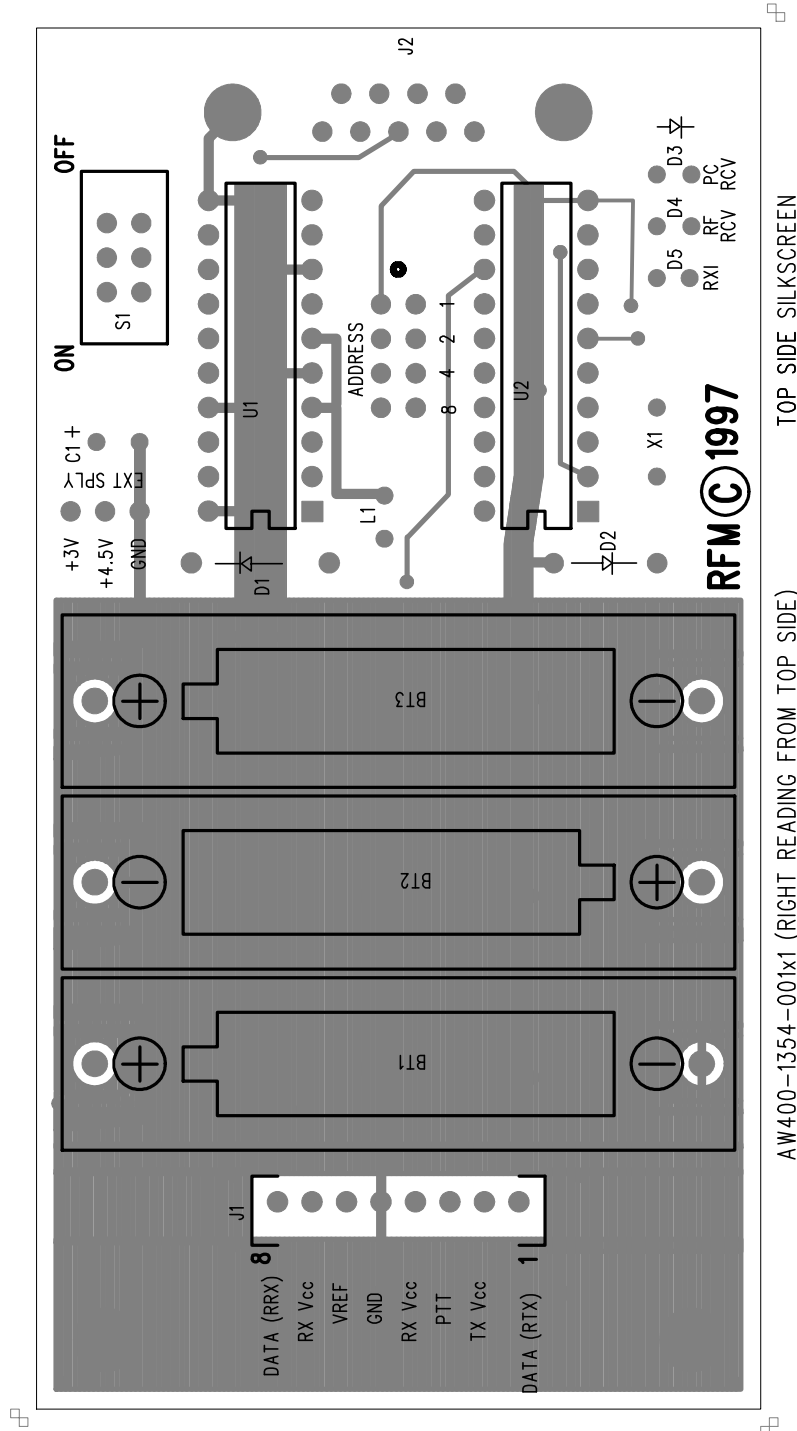


Not drawn to scale. Units in inches.  
22 AWG insulated solderable magnet wire.  
3.5 turns close wound on .125 in. dia.  
Finished ID = .125, +/- .003

868.35 MHZ ANT  
7/07/98 LAM  
(c) 1998 RFM  
400-1406-001



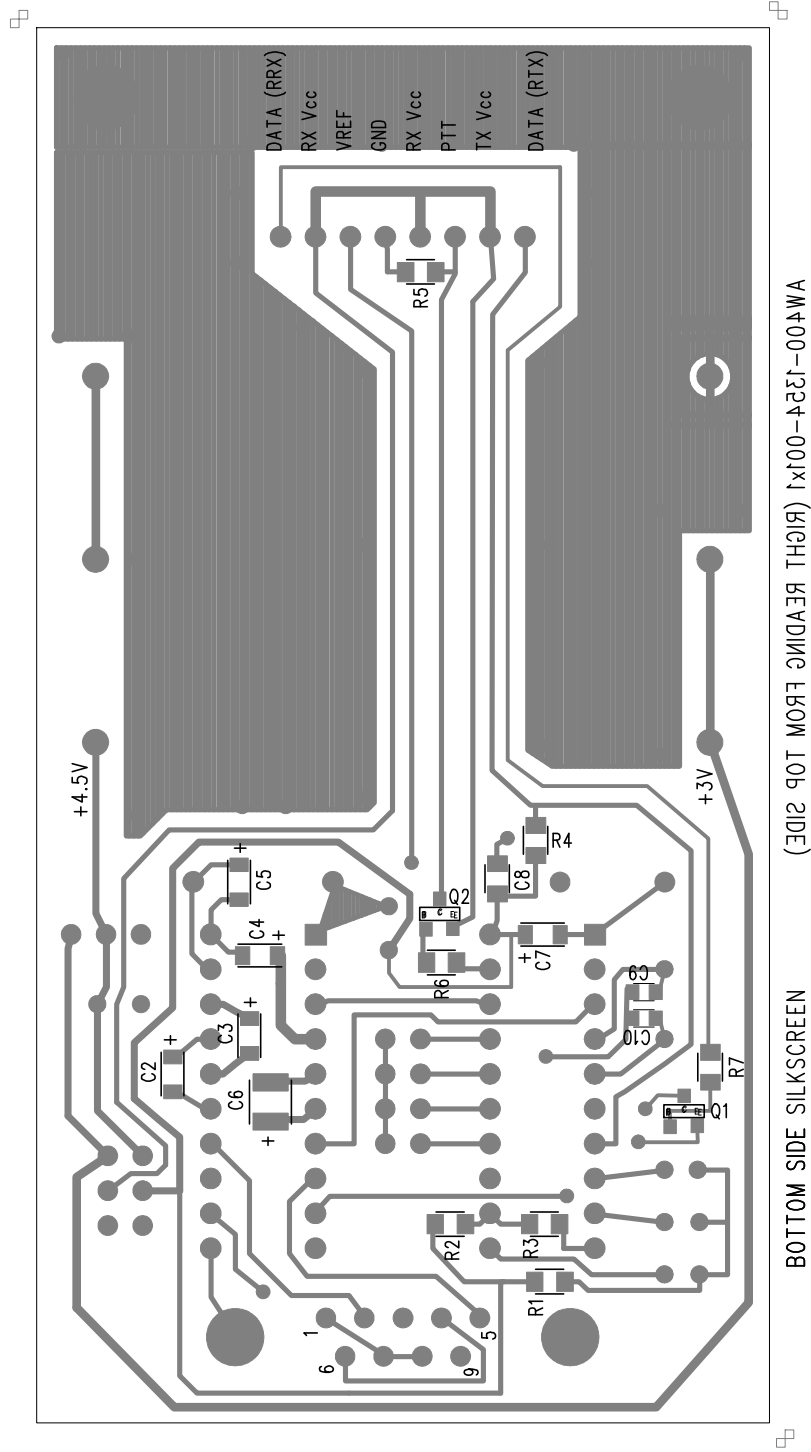
# PB1001-1 Protocol Board Top Side Component Placement





# PB1001-1 Protocol Board

## Bottom Side Component Placement



## PB1001-1 Protocol Board Bill of Materials

Qty	MURATA P/N	Vendor	Vendor P/N	Description
1	400-1354-001x1			Printed Circuit Board
1	500-0669-001	Newark	51F2912	Cap, electrolytic, 100uf 25V
5	500-0243-105	Newark	89F5035	Cap. SMT, Kemet T491A105K016AS
1	500-0244-106	Newark	92F5768	Cap, SMT, Kemet T491B106K006AS
1	500-0623-104			Cap. chip, 0805, 0.1uf 25V
1	500-0646-001	Digi-Key	1N5819CT-ND	Diode, Schottky, 1N5819
1	500-0051-001	Digi-Key	1N4148CT-ND	Diode, High speed switching, JANTX1N4148
3	500-0647-001	Digi-Key	LT1034-ND	T-1 Ultrabright LED
1	500-0648-001	Digi-Key	WM3206-ND	PCB connector, Molex 22-02-2085
1	500-0649-001	Newark	89N1583	PCB socket, 9 pin, SPC Technology DE9S-FRS
1	500-0650-001	Newark	44F4268	Inductor, 15uh
1	500-0651-002	Force Electronics	10-89-6084	8 pin dual row header, Molex 10-89-6084
1	500-0183-001	Motorolla	MMBT2222AL	Xstr, SOT, MMBT2222AL
1	500-0653-001	Newark	MMBT2907AL	Xstr, SOT, MMBT2907AL
1	500-0022-182			Resistor, chip, 1.8K(J), .2w, 0805
1	500-0732-001			Resistor, chip, 154K, .2w, 1%, 0805
1	500-0673-104			Resistor, chip, 100K, .2w, 1%, 0805
1	500-0022-204			Resistor, chip, 200K(J), .2w, 0805
2	500-0022-513			Resistor, chip, 51K(J), .2w, 0805
1	500-0022-103			Resistor, chip, 10K(J), .2w, 0805
1	500-0724-001	Augat	SSTS220PC	Switch, DPDT
1	500-0655-002	Digi-Key	CTX063-ND	22.1184 MHz Xtal, Series Resonant
2	500-0656-001	Digi-Key	ED3320-ND	20 pin IC socket
1	500-0657-001	Digi-Key	MAX218CIPP-ND	RS232 Transceiver, MAX218CIPP
1	500-0658-002	Arrow Electronics	AT89C2051-24PC	24MHz, PDIP, com temp
1	500-0659-002	Keystone	2446	AAA battery holder, single cell
2	500-0660-001	Digi-Key	H560-ND	Screw, 6-32, 1/2 inch, nylon
2	500-0661-001	Digi-Key	H620-ND	Nut, 6-32, nylon
4	500-0665-001	McMaster Carr	9723K22	Bumper feet, .375 square