



I/O-485 User's Guide Table of Contents

Chapter 1: Introduction	
I/O-485 Features	2
I/O-485 Boards & Networks	3
General Conventions	3
RS-485	3
EIA RS-485 Standard	3
Chapter 2: Getting Started	
Power Requirements	4
RS-485 Link & Optocoupling Block	6
Address Selection	6
Set Up Example	7
Chapter 3: Functions	
Resets	8
Hardware Resets	8
Software Reset Command	8
Ports & Pins on U1 and U2	8
Parallel Ports	8
Serial Peripheral Interface	9
Pulse Width Modulation	9
Interrupts	9
Multiple Stepping Motors	9
A/D Converter	9
A/D Converter Software	10
Reading an 8 Bit ADC	11
Reading a 10 or 12 Bit ADC	12
Differences Between 10 & 12 Bit ADCs	13
Chapter 4: Software	
Broadcasting Commands	14
Initializing the Network	15
Selecting a Station & U1, U2, or Both	16
Resets	16
Software Reset	16
Hardware Resets	17
Programming Example	17
Appendix A: Calculating Power Requirements	
Appendix B: Starting a Session	



I/O-485 User's Guide

32 Station Multidrop RS485 System with A/D Converter

DISCLAIMER:

RMV ELECTRONICS INC. does not assume any liability arising from the application and/or use of the product/s described herein, nor does it convey any license. RMV ELECTRONICS INC. products are not authorized for use as components in medical, life support or military devices without written permission from RMV ELECTRONICS INC.

The material enclosed in this package may not be copied, reproduced or imitated in any way, shape or form without the written consent of RMV ELECTRONICS INC. This limitation also applies to the firmware that the Integrated Circuits in this package might contain.

WARRANTY: *RMV ELECTRONICS INC. will replace, free of charge, faulty components in this package with the exception of those parts that might be damaged by improper use of the unit; for a period of 6 months after the date of purchase.*

Chapter 1 INTRODUCTION

The I/O-485 line of boards is a data acquisition and control system which allows the user to connect a master computer to up to 32 remote stations using only one cable in a multidrop configuration. The full duplex link is also widely known as a "4 wire link" which in fact requires 5 wires since all the signals are referenced to ground. An optional line is used by our system to carry power to the remote stations.

The I/O-485 boards can be connected to the computer using our specifically designed RS232/RS485 converter boards or a third party RS232/RS485 converter.

The major advantages of using the I/O-485 are:

1. Easy set of commands emerging from the use of two ITC232-A IC's per station.
2. No need to do any programming on-board.
3. Low cost per station.

I/O-485 Features:

- 32 stations, RS485 multidrop link via either 5 or 6 wires.
- Two independent ITC232-A chips per station. The ITC232-A is a multifunctional IC for real time data acquisition and control. Please refer to the ITC232-A User's Guide for a detailed description of this IC [some limitations apply to the use of port A]. The use of two ITC232-A's allows for:
 - 32 TTL compatible I/O lines per station (42 if only 1 station is present in the network). These lines are independently configurable as inputs or outputs.
 - Four additional lines which can be used as parallel inputs (PD) or as a Serial Peripheral Interface (SPI) for serial synchronic data transfers between the ITC232-A's and other IC's such as ADC's or DAC's. (The other SPI is used for the included ADC).
 - Two independent 10-10000 Hz, 0-100% duty cycle Pulse Width Modulation channels.
 - High and Low multiple interrupt lines (available only when the station is active).
 - Baud rate configurable from 300 Bauds up to 115200 Bauds.
 - Eight channels for reading relative resistance (or a liquid's conductance) or capacitance.
 - Four stepping motor logic ports capable of managing monophasic, biphasic and half-step modes. Emphasis has been placed in making the stepper interface easy to use.
 - A 1 keystroke "Again" command allows to repeat the previous command by sending only one character (thus allowing a faster communication rate).
 - On-screen help (a summary of all the ITC232-A commands is sent to the computer).
 - A configuration report command which sends to the computer the active configuration of any parallel port, the PWM or the stepper motor ports. The value written on a port can also be read, thus allowing the verification of written values, i.e. that the one the station selected is actually the one responding.
- Eleven channel Analog/Digital Converter with 8, 10 or 12 bits of resolution.
- Five addressing modes, three of which are broadcasting modes:
 - i. All ITC232-A's in all the stations receive the commands from the PC.
 - ii. Only the "even" (U1) ITC232-A in all stations receive commands from the PC.
 - iii. Only the "odd" (U2) ITC232-A in all stations receive commands from the PC.
 - iv. Only one of the ITC232-As in only one station is enabled.
 - v. Both ITC232-A's in one station are enabled
- 2500V optoisolators between the serial link and the station circuit eliminates differential ground problems.
- Links up to 4000 feet long (1200 m) are possible. This can be extended further by inserting one or more RS485 repeaters in the data link.
- Remote power capability, where the I/O-485 board is entirely powered via a 6th wire in the link (useful in the field where no power is available at the station site).

- Reverse polarity diode protection on all power inputs including the power carried by the link.
- Low cost due to:
 - Intrinsic low cost of the units.
 - No need to program anything "on-board" thus allowing for very low costs associated with software development.
 - The set of functions embedded in the ITC232-A IC drastically reduces associated hardware. As an example, an excellent precision liquid level measurement system based on conductivity can be implemented with the addition of only two 4066 IC's, an inverter (4069 or transistor) and a capacitor.

I/O-485 Boards and Networks:

The I/O-485 boards offer an inexpensive solution for multidrop, long haul data acquisition and control applications. The simplest system involving these boards includes a computer or terminal connected via an RS485 link to one I/O-485 board. The terminal may be a PC (DOS or MAC) with an RS232/RS485 adapter (also supplied by RMV Electronics Inc.) connected to one of the computer serial ports. From the adapter, a 5 or 6 wire line (depending on whether or not power is sent to the stations over the line) carries the data to and from the remote station. Two wires send data, 2 wires receive data, one additional line is common ground (usually the computer's) and finally an optional wire carries power in the form of any DC voltage compatible with the 7805 regulator on each station. A system might require only one station. In this mode, the I/O-485 board user has 10 more I/O lines (for a total of 42) at his or her disposal (the station addressing pins are released to become general I/O lines).

General Conventions

RS485:

This standard defines a multidrop, balanced link for data transfer over distances of up to 4000 feet.

EIA RS-485 Standard

EIA standard RS-485, introduced in 1983, is an upgraded version of EIA RS422-A. Increasing use of balanced data transmission lines in distributing data to several system components and peripherals over relatively long lines brought about the need for multiple driver/receiver combinations on a single twisted pair line. EIA RS-485 takes into account RS-422-A requirements for balanced-line transmission plus additional features allowing for multiple drivers and receivers. Standard RS-485 differs from the RS-422-A standard primarily in the features that allow reliable multi-point communications.

For the drivers these features are:

- One driver can drive as many as 32 unit loads and a total line termination resistance of 60Ω or more (one unit load is typically one passive driver and one receiver).
- The driver output, off-state, leakage current shall be 100 μA or less with any line voltage from -7V to 7V.
- The driver shall be capable of providing a differential output voltage of 1.5V to 5V with common-mode line voltages from -7V to 12V.
- Driver must have self protection against contention (multiple drivers contending for the transmission line at the same time). That is, no driver damage shall occur when its outputs are connected to a voltage source of -7V to 12V whether its output state is a binary 1, binary 0 or passive.

For receivers these features are:

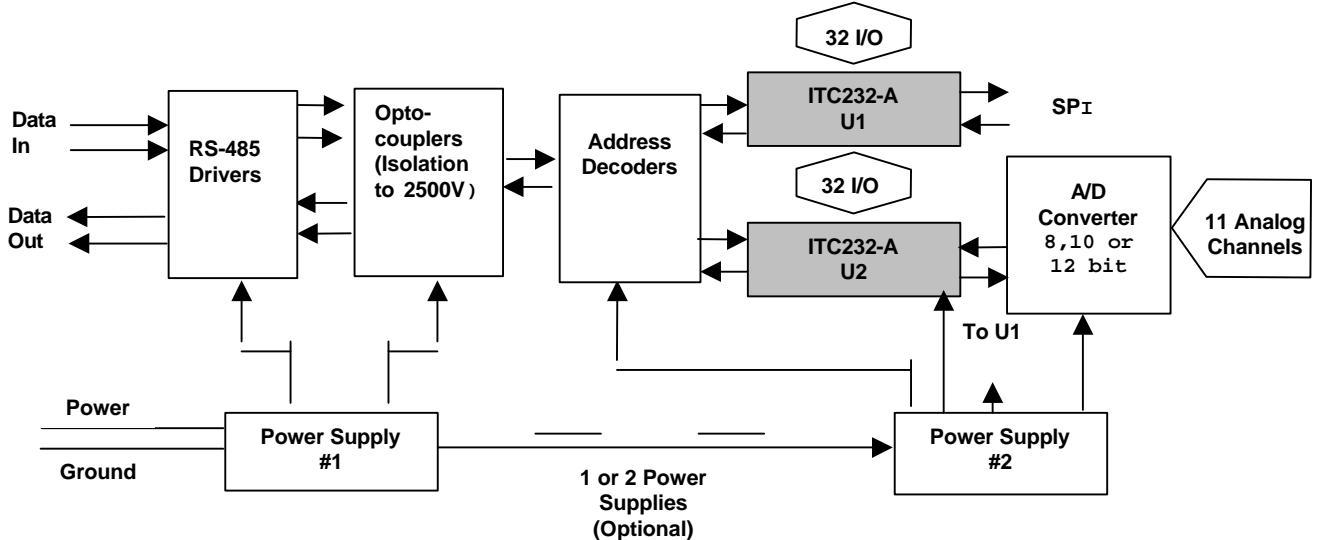
- High receiver input resistance, 12 kΩ minimum.
- A receiver input common-mode range of -7V to 12V.

- Differential input sensitivity of $\pm 200\text{mV}$ over a common-mode range of -7V to 12V .

Chapter 2 GETTING STARTED

It is assumed that you are familiar with the ITC232-A IC that is the heart of the I/O 485 Data Acquisition and Control Board, and the ITC232-A set of commands. If you are not, please refer to the **ITC232-A User Guide** that describes the IC in full detail. Instead of the User Guide you can also familiarize yourself with the ITC232-A through the ITC232-A simulator/tutorial program for DOS that is included with the I/O 485 board.

The following is a block diagram of the I/O 485 Data Acquisition and Control Board.



The two major tasks to be performed when setting up your I/O485 Data Acquisition and Control system are: 1.) getting power to all the stations in the system, and 2.) establishing the RS485 network for all serial I/O.

Power Requirements

SAFETY WARNING: TWO DIFFERENT BUILDINGS CAN HAVE GROUNDS AT DIFFERENT POTENTIAL, IN SOME CASES UP TO 100 VOLTS OR MORE. THUS, WHEN HOOKING UP A LINK, BE AWARE THAT A LARGE DIFFERENTIAL VOLTAGE BETWEEN THE LOCAL GROUND AND THE LINK GROUND MIGHT BE PRESENT AND THEREFORE YOU MIGHT GET ELECTROCUTED IF YOU TOUCH BOTH GROUNDS SIMULTANEOUSLY.

The I/O-485 board contains 2 electrically isolated circuit sections: the section that carries out the commands sent by the computer and the section that handles the RS485 link. The former is connected to the "local" ground where the I/O-485 board is placed. The other section of the circuit is connected to the ground line in the link that is normally connected to the computer ground. The two circuits need to be isolated because the potential difference between the two grounds can be very large (over 100 Volts), for example between 2 different buildings. Connecting these 2 grounds together might result in destruction of the drivers, damage to the cable and of course, closing the circuit with your body might result in electric shock.

Data travels from one section to the other via 3 optocouplers with isolation rated at 2500 Volts.

Since the two circuit sections are electrically isolated, two power sources are required. However, since optoisolation is not always needed, the I/O-485 boards may also be powered by only one power supply. The following power supply combinations are possible:

- i. Two independent power supplies on each station.
- ii. One power supply feeding the functional section of the I/O-485 board, the link is powered from the 6th line in the link cable.
- iii. There is no need for optoisolation and power for all the circuits is drawn from the 6th line in the link cable or from one local power supply.

For the I/O-485 boards to operate properly, both circuits must be powered one way or another. Thus, both LED's D7 (for REG1) and D9 (for REG2) must be lit.

The following table shows the use of the different jumpers and power connectors to address all different possibilities:

MODE	PWR1	PWR2	JP3	JP4	JP5
Two local independent power supplies	Power in	Power in	1 - 2	1 - 2	1 - 2
One local power supply + power via link *	Power in	---	1 - 2	1 - 2	2 - 3
Power via link only **	---	---	2 - 3	2 - 3	2 - 3

* The PWR3 connector on Page 3 of 3 of the schematics (NETWORK PWR) allows the use of unregulated power IN or OUT to any of the boards. This means that power can be brought into the link power line from any station and used to power a distant board that does not have access to a local power supply. In a situation such as this, the board into which the network power supply is being fed must have jumper JP5 set to 1-2 and JP3 and JP4 set according the table with respect to that board's power supply configuration. The board that is receiving the network power has jumpers JP3, JP4, and JP5 all set to position 2-3. It is recommended to use only ONE power supply for the power line in the entire network and to place it at the computer end of the network (see Appendix #1 for instructions related to calculating the power requirements of a network).

** Make sure when powering the entire board from the link power line that the local ground and the link ground are connected together (JP3, jumper on pins 2 - 3).

It is difficult to determine beforehand the parameters involved in powering one or more remote stations via the link power line. The following characteristics need to be considered:

- a. The voltage applied to the line.
- b. The resistance of the wire (power + ground lines).
- c. The load at each station and their relative position in the network. Appendix A explains how to calculate the power requirements for any given network.

All power inputs, including those to and from the link, are reverse polarity protected by a diode. An inverted diode across each 7805 protects the latter from reverse over-voltage.

RS485 Link and Optocoupling Block

The electronics involved in the transmission and reception of data are shown on page 3 of 3 of the schematics. PJ1 and PJ2, in the schematic correspond to the two phone jack type connectors on the I/O485 board. TS1 and TS2 are the two screw type terminals provided on the I/O485 board. These two types of connectors are used for serial data I/O that is sent by the master (usually the PC).

Serial data is received into the I/O-485 board via the YELLOW and BLUE wires and is transmitted out via the BLACK and WHITE wires. A resistor network (RP2) is placed across the 2 pairs of data wires thus, different resistance values or none at all can be used. 120 Ohms is recommended, but in fact, RP2 should match the impedance of the data transmission line.

IMPORTANT: For optimal operation, the RS485 link requires two resistor networks, one at each end (the RS232/RS485 converter RMV Electronics Inc. supplies has a computer end resistor network). Intermediate I/O-485 stations work better with the resistor network removed. Consequently, for intermediate I/O485 stations you must remove the RP2 IC from its socket. However, if it is the terminating board, or the only I/O 485 board do not remove the resistor network IC (RP2).

The data lines are protected by 4 diodes and 2 Zeners that prevent voltages over 6 Volts from reaching the drivers.

There are three optocouplers per board. One (U9) receives data from the computer, a second optocoupler (U11) sends data out and finally a third (U10) enables the transmission of data only when the station is selected.

Address Selection

All I/O 485 boards within a network must be uniquely addressed in order to identify each in the link. This is done by placing jumpers (bridge) on the terminal strip H1. A jumper must be placed the pins which correspond to a board's station number (address). Placing the jumper on H1 positions 1-2 disables the address selector. This is done in a system that uses only one 485 board.

All of the U3's (ITC232-A) in the network are listening to and paralleling the commands sent to either U1 (ITC232-A) and/or U2 (ITC232-A). This is of no consequence since the only portion of U3 used is PA (Port A). As a result, PA on U1 and U2 should not be used for any purpose unless the system is not addressable (i.e. there is only 1 station and H1, 1-2 are bridged together). Under very careful control, PA on U1 and U2 can be used but it is not recommended.

IMPORTANT: PA is normally used for address selection. Thus, avoid using PA for any other task in an addressable system. If only one I/O-485 board is present in the network then place a jumper on H1 1-2. This releases 5 pins on PA of U1 and U2 for general use.

All PA pins on U3 have pull down resistors. Thus, upon a reset, PA on U3 is in a high impedance state and the decoded address, as determined by the resistors corresponds to Station #0 (S#0). CSU1 and CSU2 however, are not enabled and therefore U1 and U2 on that station are deaf and mute (see the truth table on sheet 2/3 of the schematics). This means that if S#0 is accidentally reset while another station is being addressed by the computer, S#0 will also be selected by the pull down resistors, but since U1 and U2 are not enabled, S#0 will not start paralleling the commands sent to another station and therefore the user does not need to worry about a station doing what it is not supposed to do.

A bridge must be placed on H1 or H2 (Sheet 2 of 3 of the schematics) in order for the corresponding station to work. As was noted previously, placing the bridge on H1 positions 1-2, disables the address

selector.

Look at the truth table in Sheet 2 of 3 of the schematics to understand how PA, U3 and the logic gates control the flow of data. They work as follows: If only CSU1 is High, then the response reaching the computer originated in U1. If only CSU2 is High then the computer receives the response from U2. If both CSU1 and CSU2 are High then both U1 and U2 in the selected station execute the commands but the response reaching the computer is generated by U1. If BROAD (for broadcasting) is High then all U1's and U2's or U1's and U2's will respond (depending on the levels of CCU1 and CSU2). However, the only station to respond is the selected station. U2 if only CSU2 is High, U1 if only CSU1 is High, and U1 if both CSU1 and CSU2 are High.

To find any required address and mode proceed as follows:

1. Determine the station number (from 0 to 31). S.
2. Determine I which represents the IC combination. For U1, I = 32, for U2, I = 64.
3. Determine B. For broadcasting B = 128, otherwise B = 0
4. The number to be written to PA = S + I + B.

Three LED's signal the status on the board; D2 is on whenever the station is selected, D4 is always on and flickers as data come into the station, D3 lights up when the station is selected and it flickers when data is sent to the computer.

Set Up Example

The simplest configuration consists of a PC at one end of the network and the I/O485 board on the other end. The set up for this basic configuration is described as follows:

- Step 1: Power is applied to the I/O485 board through either the power jack or screw terminal at PWR1 (BOARD PWR).
- Step 2: The board is connected to the network. In this single board configuration, this involves connecting the RS485 board to a PC. Unless the PC comes equipped with an RS485 port, a converter such as the RMV RS232-485 Converter is required to connect the I/O485 board to the PC. A 5 or 6 wire line is connected to PJ1 with the other end inserted into the converter. From the converter a serial cable completes the link to the computer.
- Step 3: The address is set for the board. As this is a single board system, the jumper on terminal strip (H1) that selects the board's address is set on pins 1-2.
- Step 4: Connections for analog and digital channels (data acquisition and control) are made either through the 34 pin ribbon header (CON2) which can receive both analog and digital signals under certain conditions, or the 40 pin header strip (CON1) which is digital I/O only.

Chapter 3 FUNCTIONS

Resets

1. Hardware Resets

The boards can be reset in several different ways:

- a. By pressing the reset switch (SW1). This should only be done under well controlled conditions since otherwise the computer might not be aware that the reset took place and therefore it might not issue the necessary initialization commands for that station. The result is that the station incoming might ignore

commands.

b. By pulling High PB0 on U2 (with a jumper between pins 1-2 on JP1)

c. By selecting address 31 **if and only if** JP1 pins 2 and 3 are bridged together. In this case, obviously the network can only have 31 stations addressed from 0 to 30. Removing all the jumpers from JP1 disables all hardware resets available from the computer.

The table below summarizes the possibilities:

HARDWARE RESETS	Jumpers on JP1
No hardware reset	None
Hardware reset by bringing PB0_2 High.	1 - 2
Hardware reset by selecting station address 31	2 - 3

2. Software Reset Command

The ITC232-A chips may also be reset using the RESET[CR] command but this is not advisable since the analog/digital converter is not reset this way. Furthermore, if only one station and one IC in that station is selected, that IC and all of U3's throughout the network will be reset but no other ITC232-A will. Thus, it is advisable to set the network in full Broadcast mode before issuing a RESET command.

Ports & Pins on U1 and U2

1. Parallel Ports:

PB & PC: PB and PC on both U1 and U2 are always available to the user. PB0 and PB1 on U1 may be used for reset purposes according to the jumper position on JP1 and JP2. This means that there are 30 to 32 available general I/O lines organized in 4 ports 8 bit wide. Please refer to the **ITC232-A User Guide** for a detailed description on the use of these ports and their associated functions.

PD/SPI: This port, which can serve as a 4 bit input parallel port or an SPI is only available on U1. PD/SPI on U2 is used by the A/D converter.

PA: Abstain from using this port unless the jumper on H1 is placed between pins 1-2 thus making the system non-addressable. In such a case, the lower 5 pins on PA can be used as the ones in PB or PC. (In this case, be aware that writing to PA writes the entire port. Thus, whenever PA is being used as outputs, make sure the value written to PA5 and PA6 coincides with the desired value for addressing U1 or U2).

2. Serial Peripheral Interfaces (SPI)

The SPI on U1 is brought to CON1 (PD0_1..PD3_1). The SPI on U2 is used for the A/D converter. To use the SPI on U1 in an external circuit, tie PD3_1 to VCC and place a 1K resistor between PD2_1 and either ground or VCC, according to the polarity required by the external IC clock. See the **ITC232-A User Guide** for more details about using the SPI.

3. Pulse Width Modulation (PWM) pins

The PWM pin on both U1 and U2 is available on CON2 as PWM1 and PWM2. These pins can be used independently and they can be functional while the corresponding ITC232-A is doing something else. Please refer to the **ITC232-A User Guide** for a detailed explanation on how to use these pins.

4. Interrupts

Both U1 and U2 IRQL pins are connected together. The same happens with IRQH. This is because an interrupt should be signaled no matter which IC receives it. A capacitor in parallel with a bleeding resistor can be used to bring multiple interrupts to these pins since they are edge sensitive only.

IMPORTANT: In an addressable network, only interrupts coming from the selected station will reach the computer, unless the IRQL or IRQH lines from different stations are hooked up together.

5. Multiple stepping motors

Having two ITC232-A's on board the I/O485 allow the use of more than 1 stepping motor. However, the ITC232-A will not allow simultaneous control of two or more steppers, because it will recognize a CR as a stepper emergency stop and a CR must be sent in order to switch between U1 and U2. However, if simultaneous stepping is not required, the I/O485 can be used to control more than one stepper motor in rapid sequence.

Analog/Digital Converter (ADC)

The I/O-485 boards are shipped with an 11 channel 8, 10 or 12 bits Analog/Digital converter (ADC). Before going into the details about how to use the ADC, a short note on the pros and cons of using different resolutions: Eight bits has the advantage that the value is obtained in only 1 reading. This results in twice the speed attainable with the other two options since the latter require 2 readings to assemble the 10 or 12 bit value. Higher resolutions allow for higher precision and also reduce the error on lower values. This happens because the error is 1 LSB. Thus, a value of 100 units will have a 1% error which might be acceptable, but a value of 10 will bear an error of 10% which in most cases is unacceptable. Thus, one can read 12 bits and then shift the result to the right by 4 bits (divided by 16). The result is now expressed as an 8 bit value, with a resolution of ~0.4% [1/256] while the error has been very significantly reduced (in the above-mentioned example, it is reduced from 10% to ~0.6%).

The software required to control the ADC is specific for each resolution and therefore there is an entire section devoted to running them.

The reference voltage for all channels is set by the 10 K Ω multi turn pot (TRIM) on the boards and it appears on a single pin header next to the trim pot (TP1) so that you can easily measure it. The reference voltage determines the upper measurable range for all channels. Voltages above this threshold will yield the full-scale value for the ADC and values below GND will yield 0. The maximal voltage that can be directly read is close to VCC (~ +5 V). It is recommended that the impedance of the circuit being read be < 10 K Ω .

WARNING: Negative voltages might destroy the ADC chip.

Analog/Digital Converter Software

The ADC interfaces with the ITC232-A via a 3 line synchronic serial interface (SPI, PD/PS pins on U2). One of these lines (PD1/PS_TX) carries into the ADC the data used to select the analog channel for the conversion. In the case of the 8 and 10 bit ADC's, only the 4 most significant bits of the value received by the ADC are used to select the channel. The 12 bit ADC requires that the lower nibble be \$C instead of \$0. Since the ITC232-A sends out 8 bit words, the following table shows the values required for any specific

channel.

ADC TABLE

CHANNEL	8 & 10 bit ADC (HEX)	12 bit ADC (HEX)
AN0	\$00	\$0C
AN1	\$10	\$1C
AN2	\$20	\$2C
AN3	\$30	\$3C
AN4	\$40	\$4C
AN5	\$50	\$5C
AN6	\$60	\$6C
AN7	\$70	\$7C
AN8	\$80	\$8C
AN9	\$90	\$9C
AN10	\$A0	\$AC
HALF SCALE	\$B0	\$BC
ZERO *	\$C0	\$CC
FULL SCALE*	\$D0	\$DC

* Unavailable on the 8 bit ADC.

Another line, PD0/PS_RX, is used to read the result of the last conversion into the ITC232-A. Finally, PD2/PS_CK carries the clock signal necessary to move data between the ICs.

The ITC232-A's Serial Peripheral Interface must be configured to the ADC requirements prior to use. To achieve this, send only once, at the beginning of a session regardless of the ADC resolution used, the command **PCSA128** (see under **COMMANDS** in the **ITC232-A User Guide**).

Even though the 8, 10 and 12 bit ADC chips are pin compatible and they work essentially in the same way, the way of reading data into the ITC232-A and from there to the converter varies. Thus, the different IC's will be discussed separately.

Reading an 8 Bit ADC

To configure the SPI connected to the ADC use the command: **PCSA128[CR]**

The MC145041 currently used operates as follows: The analog channel is selected by the value sent from the ITC232-A (see ADC Table). Once a channel is selected, the conversion takes place immediately and the result is stored in an internal shift register.

IMPORTANT: This value needs to be clocked back into the ITC232-A and therefore you should read the result of the conversion **immediately** after selecting the analog channel. Otherwise, the obtained result is historical.

In order to write and read to the MC145041 you will use the **PWSn** and **PRS** commands (n represents the number used to select the analog channel as per the ADC Table. If you are always reading the same channel, you do not need to select the channel every time since whenever a PRS command is issued, the last value written to the SPI is automatically and simultaneously clocked out the PD1/PS_TX pin. This value is either the last value written to the SPI (with the last PWSn command) or 0 if none has yet been written following a reset. Thus, when you issue PWSn followed by PRS this is what actually happens:

1. The analog channel is selected and a conversion takes place,
2. n is now stored in the ITC232-A as the last value written to the SPI.
3. The result of the conversion is waiting in the MC145041 serial register.
4. The PRS command clocks that result into the ITC232-A while n is automatically sent again to the MC145041 thus starting a new conversion, the result of which is now in turn waiting to be clocked out the MC145041 shift register. Should you issue a new PRS without sending PWSn first, the last result for that analog channel would be clocked into the ITC232-A while the same channel number is clocked out thus generating a new conversion. Please note that an initial PWSn **MUST** be sent in order to initially select the channel.

A fast rate of conversion can be attained by using the @ command after a PRS (see **ITC232-A User Guide** and below).

Examples:

We assume the PCSA128 command has already been issued in order to configure the SPI, that U2 is selected and that an 8 bit ADC is being used.

- (a) **To read channels AN0, AN1 and AN2 successively:**
PWS0 (for AN0), PRS (reads AN0), PWS16 (selects AN1), PRS (reads AN1), PWS32 (selects AN2), PRS (reads AN2).
- (b) **Reading channel AN5 multiple times:**
PWS80 (selects AN5 and makes the first conversion), PRS... commands not related to the SPI, if any..., PRS and so on. Should you wish to read AN5 after an interval following the last conversion, send PRS twice and discard the 1st result because it corresponds to the last conversion before the interval.
- (c) **Reading channel AN8 in a continuous loop:**
This is the fastest way of reading successive values from the SPI: PWS128 (selects AN8), PRS (yields the first result), @, @, @ ... Note that no other command can be issued between @'s. Otherwise, that command would be the one repeated using @. On the computer side you will receive, along with the conversion results, the @PRS legend. For more details about the @ command refer to **LIST OF COMMANDS**, Again command in the ITC-232-A Users Guide.

Reading a 10 or 12 Bit ADC

Our I/O-485-10 boards use either an MC154051 or a TLC1543 instead of the MC145041 used in the I/O-485-8 series. The I/O-485-12 boards use a TLC2543, 12 bit ADC. All these ADC's have 11 channels and 8, 10 or 12 bits of resolution. The 8 bit IC and the 10 or 12 bit IC's are very similar in the way they operate. The only difference is that the 10 and 12 bit ADC's return the result in two bytes instead of 1. Thus, the SPI has to be read twice to pull out the result. These IC's are built in such a way that there is a shift in the result, and therefore once the two bytes have been assembled, the result must be divided by 64 or 16 to obtain the actual value.

The 8, 10 and the 12 bit ADC are rather similar in their way of operation. However, some important differences exist in the way the data flow. Thus, we will first address the 10 bit ADC and then the 12 bit one.

To operate the 10 bit ADC proceed as follows: (Remember, send only the characters within <> and an [Enter] at the end of each command).

1. Configure the SPI by sending from the computer <P>ort <C>onfigure <S>erial <A>ll <128>.
2. Select the analog channel with <P>ort <W>rite <S>erial n where n is selected from the ADC ADDRESS SELECT TABLE above.
3. Send <P>ort <R>ead <S>erial. The Least Significant Byte of the previous conversion is returned. Send PRS again (or @) and the Most Significant Byte of the previous conversion is returned. Keep track of the number of PRS commands sent in order to know whether you are reading the LSByte (always first) or the MSByte. Should you lose track, you can reset the ADC by using PB2 on U2.
4. The result returned by the 10 bit ADC is shifted 6 bits to the left. Thus, the result must be divided by 64. Should you wish to assemble the result in one variable V, use $V = (\text{LSByte} + \text{MSByte} * 256) / 64$. Note: If you are typing commands from a terminal you will not be able to calculate the results. You might think that the values fluctuate "at random". This is not so. What is happening is that the 6 LSB's are fluctuating at random, but they must be discarded anyway.

Example:

For practical reasons, let us use test channel AN11 (Half Scale) which always returns the half scale value (512 or \$200 for our 10 bit ADC).

- a. Working from a terminal program, configure the results in ASCII Hexadecimal with CRAH [Enter].
- b. Configure the SPI with PCSA128 [Enter].
- c. Select channel AN11 by typing PWS\$B0 [Enter].
- d. Get a reading with PRS [Enter] followed by @. You should get \$00 and \$80 which corresponds to \$8000 (32768 decimal). Divide by \$40 or 64 decimal to obtain 512, the value of half the scale.
- e. Configure results in decimals with CRAD [Enter]. Then type PRS [Enter] followed by @. You get 000 and 128. To compute the result as shown in 4, do: $R = (000 + 128 * 256) / 64 = 512$.

The description above works best when only one channel is been used. However, if channels are changed you should insert a third "PRS" command. This is because when you change channels with "PWSxx" the last conversion is sent out. This of course corresponds to the last channel and should be discarded. The example above shows how to perform the 3 tests (Zero, Full scale and half scale) the MC145041 provides in QBasic[®]. Other channels are read identically:

Differences Between a 10 Bit and 12 Bit ADC:

Even though in principle the 10 and 12 bit ADC's are very similar, for example the channel table above is the same for both, there are some important differences too. Some of the significant differences are as follows:

10 Bit ADC

1. The lower 4 bits (nibble) in the channel select value is irrelevant (set to 0).
2. The result returned is 10 bits long.
3. The result needs to be shifted 6 places (divided by 64)

12 Bit ADC

1. The lower 4 bits (nibble) in the channel select value **are** relevant. They work as follows:

- Bit.0: 0 for Unipolar (unsigned), 1 for Bipolar (in this case the MSB is the sign and $0 = (\text{Ref} + \text{Ref-})/2$.
 - Bit.1: 0 for MSB first, 1 for LSB first.
 - Bit.2 and 3: 0,0 for 16 bit transfers, 01 for 12 bit transfers and 10 for 8 bit transfers (we use 16)
2. The result returned is 12 bits long.
 3. The result needs to be shifted 4 places (divided by 16)

Chapter 4 SOFTWARE

The following description will only deal with the commands that are specifically required for using the I/O-485 boards. Please refer to the **ITC232-A User Guide** or the ITC232-A simulator/tutorial program for DOS for a detailed description of the commands.

The commands described below can be sent as ASCII strings using any programming language or they can be typed from within a terminal program. The latter possibility is recommended when setting up and debugging a system, before writing the actual code.

If the terminal mode is selected, the CRAD, CRAB or CRAH configurations are best. If the boards are controlled from within a program, then the CRAP configuration is recommended.

Remember that throughout this description all of the U3's in the network are listening simultaneously to what comes from the computer. The vertical lines in | Command | describe an optional command. [CR] means a Carriage Return sent after each command.

Addressing modes:

The ITC232-A's (U1 and U2) in the network can be addressed as follows:

- Only U1 in only 1 station
- Only U2 in only 1 station
- Both U1 and U2 in only 1 station
- U1's only in all stations
- U2's only in all stations
- U1's and U2's in all stations

This offers great flexibility for configuring systems in which all the stations are identical since once in broadcasting mode, only one command is required to configure all the stations. This is particularly useful for changing the Baud rate of the entire network, or configuring the SPI's connected to the ADC. PA is used for selecting the address (PA0..PA4) and the addressing mode (PA5..PA7). The table below shows the different addressing modes:

Actions resulting from the value on PA

ACTION	Addr val(dec)*	PA7 (BROAD)	PA6 (CSU2)	PA5 (CSU1)
No ITC232-A is selected	0	0	0	0
Enable only U1 in selected station only	1	0	0	1

ACTION	Addr val(dec)*	PA7 (BROAD)	PA6 (CSU2)	PA5 (CSU1)
Enable only U2 in selected station only	2	0	1	0
Enable both U1 and U2 in selected station only	3	0	1	1
No broadcasting effect. U1 and U2 are disabled	4	1	0	0
Broadcast only to all U1's in the network. Only U1 in selected station answers	5	1	0	1
Broadcast only to all U2's in the network. Only U2 in selected station answers	6	1	1	0
Broadcast to all U1's and U2's in all the stations. Only U1 in selected address answers	7	1	1	1

* This value should be added to the station address required (0-31) and then written to PA.

Broadcasting Commands

There are 3 different broadcasting modes. The computer can broadcast to all the U1's in the network, to all the U2's in the network or to all the U1's **and** U2's in the network.

U3-PA7 = High, enables broadcasting. This however does not result in any action if CSU1 and/or CSU2 are not enabled. If CSU1 is selected, (High) all U1's will execute commands until the BROAD line (PA7 on U3) or the CSU1 line (PA5 on U3) goes Low. The only IC that answers back is the U1 in the selected station. The same applies to U2. When both CSU1 and CSU2 are High, all U1's and U2's in the network follow the commands but only U1 in the selected station answers back.

Examples:

a. Configuring PC0 as an output in all the U1's in the network and getting all U2's to generate a 50 Hz PWM signal:

It is assumed that all PA pins were configured as outputs at initialization time.

```
PWA160[CR] { 160 = 61010 0000 which enables BROAD and CSU1 on station #0 }
[CR]*      { A change in station or broadcast mode might return no prompt. To get one back, send an
            additional CR }
PCC1[CR]   { Port Configure C 1 }
```

More commands for all U1's:

```
PWA192[CR] { 192 = 61100 0000 keeps BROAD High, makes CSU1 Low thus disabling all U1's and
            brings CSU2 High thus enabling all the U2's in the network. Only the one in station #0 will
            answer back }
[CR]       { A change in station or broadcast mode might return no prompt. To get one back, send an
            additional CR }
W50[CR]    { Generates a 50 Hz signal }
```

More commands for all U2's:

PWA32[CR] { 32 =60010 000 brings BROAD Low and CSU1 High thus leaving U1 in station #0 enabled }
 [CR] { A change in station or broadcast mode might return no prompt. To get one back, send an additional CR }

b. Getting all the ITC232-A's in the network to operate at 115200 Bauds:

It is assumed that all PA pins were configured as outputs at initialization time.

PWA224[CR] { 224 = 61110 0000 brings BROAD, CSU1 and CSU2 High. All ITC232-A's in the network are listening, only U1, station #0 will answer back }
 [CR]* { A change in station or broadcast mode might return no prompt. To get one back, send an additional CR }
 B115200[CR] { Set new Baud rate. The computer serial port Baud rate should be changed accordingly }
 PWA32[CR] { 32 =60010 0000 brings BROAD Low and CSU1 High thus leaving U1 in station #0 enabled }
 [CR]* { A change in station or broadcast mode might return no prompt. To get one back, send an additional CR }

* If the station selected changes or the IC is changed within a station, U3 might switch the formerly enabled ITC232-A before it acknowledges the command. This results in no prompt returned. Thus, a 2nd CR must be sent. This is not always necessary, but the complexity of the rules that apply makes it easier to just send an additional CR after changing addresses, broadcasting mode or switching between U1 and U2.

Initializing the Network

PA on U3 is used to select the station address, which IC (U1 and/or U2) is connected to the computer and to set the broadcasting line BROAD. At power up or following a reset, PA is all inputs pulled Low by resistors. Thus, PA must be configured as all outputs and then a suitable address must be written to it.

Example:

PCA255[CR] { Configures PA as all outputs. The value after a reset or power up condition is 0 on all the pins }

Expect no feedback here; no IC is enabled in any station because PA0..PA7 = 0 and therefore so are CSU1 and CSU2. Thus, introduce a suitable delay. For 9600 Bauds it should be > 5 ms.

PWA224[CR] { Writes bin 1110 0000 to PA. PA7 = 1 gets all the stations in broadcast mode, PA6 = 1 q enables U2 and PA5 = 1 enables U1. PA0..PA4 select station #0. Note that all ITC232-A's in the network will now be listening to the computer but only U1 in station #0 will answer back }

Expect no feedback here either; the acknowledgment of the ITC232-A comes out before the command takes place.

[CR] { Request a prompt }

|PCA255[CR] | { This command is optional. It is necessary if feedback about the address selected is required because it sets up all PA'S as outputs }
 |PWA244[CR] | { Same }
 |PRA[CR] | { Returns the address selected on the ITC232-A thus providing feedback about the address that was selected. Must return 244 ! }

 |CRAP[CR] | { Set up the program configuration for maximal speed }
 |B115200[CR] | { Change the Baud rate. Make sure you change Baud rates on the computers side }

... More broadcast commands such as Baud rate changes, etc...

Select new address for next commands

Selecting a Station and U1, U2, or Both in that Station

PA0..PA4 on U3 select the address (0-31). PA5 enables U1. PA6 enables U2. If both U1 and U2 are selected, they both respond to the commands but the only one to answer back to the computer is U1.

Resets

There are several ways to reset different parts of the network:

Software Reset

The RESET[CR] command resets the enabled ITC232-A's in the network. Thus, if all U1's were enabled but none of the U2's were, then the latter would not be reset. This allows to reset one U1 or one U2 at a time, all U1's simultaneously, all U2's simultaneously and all U1's and U2's simultaneously. Resetting an individual ITC232-A is strongly discouraged. Besides making the network unstable, the computer has to go throughout the entire initialization process for that station anyway.

NOTE: The RESET[CR] command does NOT result in a hardware reset and therefore, other components, such as the ADC are NOT reset.

Hardware Resets:

We call this type "Hardware resets" because there is a track on the board that actually changes levels the same way as if the reset switch were pressed.

Using address #31: By placing a jumper on JP1 2-3, a hardware reset is generated whenever station #31 is selected. There is no need to specify U1, U2 or BROADCAST since the event is generated by U3 and the associated address decoder. Placing the JP1, 2-3 jumper in all boards allows the resetting of the entire network with an address selecting command.

Using PB0, U2: This reset mode requires pins 1 and 2 on JP1 to be connected together. It is assumed that BROAD and CSU2 are selected (CSU1 is irrelevant). The following commands produce the reset:

|PCB1[CR] | { Configures PB0 as an output, necessary only if PB0 was an input }
 |PWB1 | { Brings PB0 High. Necessary only if the previous value on phantom bit was 0, otherwise the act of configuring PB0 as an output generates the reset. To understand the way phantom bits work, read **Using Parallel Ports** in the **ITC232-A User Guide** or in the simulator/tutorial. }

Using PB0, U1: A general network hardware reset using PB0, U1 can be generated as follows:
 (It is assumed that PA is configured as all outputs, PB0 is an input and its phantom value is 0)

```
PWA224[CR]    { 224 = bin 1110 0000 brings BROAD, CSU1 and CSU2 High. All ITC232-A's in
               the network are listening, only U1, station #0 will answer back }
[CR]*         { A change in station or broadcast mode might return no prompt. To get one back,
               send an additional CR }
PCB1[CR]      { Configures PB0 as an output. }
PWB1          { Brings PB0 High thus producing the reset }
```

Expect no response from any station here.

Resetting the ADC: The A/D converter (U5) might need to be reset independently of the rest of the circuit (this is particularly the case for the 10 and 12 bit versions, to regain synchronization between the computer and the byte the ADC is sending out). To this effect, PB1, U2 can be used, provided a jumper is placed between terminals 1 and 2 on JP2. In this case, the ADC will be reset whenever a hardware general reset is implemented (as usual) or when PB1 is brought high. Change | PWB1| for | PWB2| in the command sequence above.

Programming Example

The following is an example in QBasic showing how to handle the serial port, the initialization of the network and the analog to digital converters:

```
DECLARE FUNCTION Read8bits& (ch$)
DECLARE FUNCTION Read12bits& (ch$)
DECLARE FUNCTION Read10bits& (s$)
DECLARE SUB WriteSerial (s$)
DIM ValReturned

False = 0: True = 1
display = False ' Make it true to display what is happening
CLS : CLEAR : CLOSE
PRINT SPACE$(20);
COLOR 0, 7
PRINT "I/O-485-10 demo by RMV Electronics Inc."
COLOR 7, 0
PRINT
PRINT "Make sure board(s) are connected and press a key"
DO: LOOP UNTIL INKEY$ <> ""

' Use COM1 or COM2 below
OPEN "COM2: 9600,N,8,1,CD0,CS0,DS0,OP0,RS,TB2048,RB2048" FOR RANDOM AS #1

PRINT : PRINT "Expect no response here if board is just reset"
WriteSerial ("PCA255") ' Configure PA as all outputs in al U3's

PRINT : PRINT "Expect no response here if board is just reset"
WriteSerial ("PWA224") ' Get the network in broadcast mode (all U1's + U2's)

' Configure all PA's (U3 + U1 + U2) as outputs, necessary only in a network
' with more than 1 station and only in order to be able to ask any chip
' on the network if the station number written corresponds with the one
' received (to verify that the correct address was selected). If only
' one station is in use and therefore the jumper on H1 is on 1 -2,
' then the next command could be used for configuring PA on U1 or U2 as
' required.
```

```

WriteSerial ("PCA255")

WriteSerial ("PWA64") ' Select U2 station #0
WriteSerial ("PCSA128") ' Configure the SPI for the ADC

' You might want here to reset the ADC with PB2 and the corresponding
' jumper (see RESETS in the I/O-485 manual)

DO
CLS
PRINT TAB(25); " Select which ADC to test"
PRINT
PRINT TAB(30); "0. 0 to Exit"
PRINT TAB(30); "1. 8 bits"
PRINT TAB(30); "2. 10 bits"
PRINT TAB(30); "3. 12 bits"
INPUT menu
LOOP UNTIL menu >= 0 AND menu < 4

SELECT CASE menu

CASE 0
END

CASE 1
PRINT
FOR number = 1 TO 3
PRINT number
x = Read8bits("$B0")
COLOR 0, 7
PRINT "Half scale: ", x
COLOR 7, 0
NEXT number

CASE 2

' Carry out 3 reading from 3 different channels
PRINT
FOR number = 1 TO 3
PRINT number
x = Read10bits("$B0")
COLOR 0, 7
PRINT "Half scale: ", x
COLOR 7, 0

x = Read10bits("$D0")
COLOR 0, 7
PRINT "Full scale: ", x
COLOR 7, 0

x = Read10bits("$C0")
COLOR 0, 7
PRINT "Zero: ", x
COLOR 7, 0
NEXT number

CASE 3
' Carry out 3 reading from 3 different channels
' Mind that the least significant digit must be $C for configuration
' purposes required by the 12 bit ADC used.
PRINT
FOR number = 1 TO 3
PRINT number
x = Read12bits("$BC")
COLOR 0, 7
PRINT "Half scale: ", x
COLOR 7, 0

```

```

x = Read12bits("$DC")
COLOR 0, 7
PRINT "Full scale: ", x
COLOR 7, 0

x = Read12bits("$CC")
COLOR 0, 7
PRINT "Zero: ", x
COLOR 7, 0
NEXT number

END SELECT

END

FUNCTION Read10bits& (ch$)
' Where ch$ is the channel on the ADC
DIM LSB, MSB, r
SHARED ValReturned

ch$ = "PWS" + ch$
WriteSerial (ch$)
WriteSerial ("PRS") 'dummy reading

WriteSerial ("PRS") ' Get MSByte
MSB = ValReturned

WriteSerial ("PRS") ' Get LSByte
LSB = ValReturned
r = LSB + 256 * MSB
r = r / 64
Read10bits& = r
END FUNCTION

FUNCTION Read12bits& (ch$)
' Where ch$ is the channel on the ADC. Make sure you make the last digit
' on the address = $C ($xC) when you call the function
DIM LSB, MSB, r
SHARED ValReturned

ch$ = "PWS" + ch$
WriteSerial (ch$)
WriteSerial ("PRS") 'dummy reading

WriteSerial ("PRS") ' Get MSByte
MSB = ValReturned

WriteSerial ("PRS") ' Get LSByte
LSB = ValReturned
r = LSB + 256 * MSB
r = r / 16
Read12bits& = r

END FUNCTION

FUNCTION Read8bits& (ch$)
' Where ch$ is the channel on the ADC
SHARED ValReturned
ch$ = "PWS" + ch$
WriteSerial (ch$)

WriteSerial ("PRS") ' GetResult
Read8bits = ValReturned

END FUNCTION

SUB WriteSerial (s$)
DIM t, r$, C$, n

```

SHARED ValReturned

```
r$ = ""
PRINT #1, s$
IF display THEN PRINT s$
t = TIMER + .2
DO
  IF LOC(1) <> 0 THEN
    C$ = INPUT$(1, #1)
    r$ = r$ + C$
  END IF
LOOP UNTIL C$ = ">" OR TIMER > t

IF C$ <> ">" THEN
  PRINT "No response from IO485"
ELSE
  IF display THEN PRINT r$
  'Strip CR, LF and IRQ's
  s$ = ""
  FOR n = 1 TO LEN(r$)
    C$ = MID$(r$, n, 1)
    IF C$ = "L" OR C$ = "H" THEN
      IF display THEN PRINT C$
    ELSEIF C$ = CHR$(13) OR C$ = CHR$(10) OR C$ = " " THEN C$ = C$
    ELSE
      s$ = s$ + C$
    END IF
  NEXT n

  ' s$ now has the received string without CR or LF; Detect if OK received
  IF LEFT$(LTRIM$(s$), 2) = "OK" THEN
    s$ = RIGHT$(s$, LEN(s$) - 2) ' Strip the OK
    s$ = LEFT$(s$, LEN(s$) - 1) ' Strip the >

    ' Detect an error
    ELSEIF LEFT$(s$, 1) = "?" THEN
      PRINT "Error "; MID$(s$, 2, 1)
  ' Finally, the number of steps left when stepping is interrupted
  ELSE IF LEN(s$) >= 3 THEN s$ = LEFT$(s$, LEN(s$) - 3): 'Strip the "OK>"
  END IF

  ValReturned = VAL(LTRIM$(s$))
END IF

IF display THEN
  PRINT SPACE$(30);
  COLOR 0, 7: PRINT "Press a key ";
  COLOR 7, 0
  PRINT
  DO: LOOP UNTIL INKEY$ <> ""
END IF
END SUB
```

APPENDIX A

Calculating the Power Requirements of an I/O 485 Network

Figure 1a shows the network equivalent circuit when power is brought to the stations via the link. RL=Resistance of the line, the number is the station number and P=power and G=ground. RS=Equivalent resistance of a station.

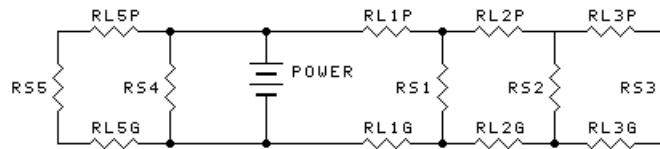


Figure 1a

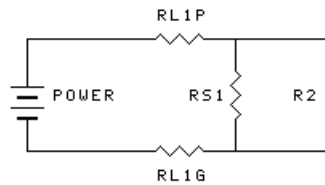


Figure 1b

The dotted line represents what happens when the link is not powered at one end, but from an intermediate station. Should power be applied on one end, disregard the dotted line and the other side of the circuit. Otherwise, do the calculations for each side of the link separately.

To solve this circuit proceed as follows:

1. We first need to know the equivalent resistance of each station. This CANNOT be measured directly with an Ohm meter. Instead remove the station from the network, apply a voltage, 8.5 volts or higher and measure the current used by the board. By Ohms law, $R = V/I$ where V is the voltage applied and I is the current measured in Amps.
2. Let us now imagine that only station #1 is present on the network (ignore everything to the right of A and B and everything to the left of the dotted line). The voltage between A and B can be extrapolated from the expression:

$$RT1 \text{ is to PV as}$$

$$RS1 \text{ is to X}$$

 where $RT1 = RL1P + RS1 + RL1G$ and PV is the Power Voltage. X= $RS1 \text{ PV} / RT1$. X is the voltage drop introduced by RS1. Thus, the actual voltage between A and B is $VAB = VP - X$. This voltage, and this applies to all stations, MUST be 8.5 Volts or higher. Otherwise the voltage regulators in series with a polarity protection diode will not work properly.
3. Let us now add station #2 to the network. $RL2p + RL2g + RS2$ can be represented as R2 in Fig. 1b. RS1 and

R2 can be reduced to one resistance (R) by solving $1/R = 1/R_{S1} + 1/R_2$.

The voltage drop produced by this new value of R can be extrapolated as above by: R_{T1} is to PV as R is to X. $X = R * PV/R_{T1}$ and $V_{AB} = PV - X$.

This new voltage between A and B is then used as PV in the analysis of station #2.

In a real network, you might have more than 2 stations. In this case, R2 in Fig. 1b represents the equivalent of all the equivalent resistances of all the stations.

Mind that in your calculations each new total equivalent resistance must be placed in parallel with the former **station** resistance ($1/R = 1/R_2 + 1/R_{S1}$) and **NOT** in parallel with the former **total** resistance.

4. The easiest way to do the entire network equivalent resistance calculations is to start from the station farthest from the power supply and get all the equivalent resistance values for each station. Then calculate the voltage drops for each station and use that as PV for the next one getting closer and closer to the power supply.
5. If you have stations on both sides of the power supply, calculate each end separately. Once you have an equivalent circuit with only 2 resistances in parallel, (one for each end), solve them into one using $1/R = 1/R_1 + 1/R_2$. R represents the entire network equivalent resistance.

APPENDIX B

Starting a Session

Send from the computer (expect no answer back):

```
PCA255[CR]          {PA=outputs on all U3's}
PWA224[CR]          {Broadcast mode}
PCA255[CR]          {PA=outputs in all other ITC232-A's in the network}
```

Select an address or broadcast other commands.

Selecting an Address:

If you wish to verify the address you will select, do:

```
Address = 96 + Station # (0 to 31)      {Select both U1 and U2 in the station}
PWA address [CR]                        {and tell U3}
```

```
NewAddress = 32 (for U1) or 96 (for U2) + Station # (0 to 31)
PWA Address [CR]                        {The value is written again, this time to either U1 or U2
                                        only. Not necessary if you wish both to be selected}
PRA[CR]                                  {The value returned MUST be the same as the value in
                                        NewAddress, thus confirming the selected address}
```

Important Tip:

An ITC232-A will NOT be able to understand any new commands until it has sent out a prompt (>). Thus, you should not send out any new characters until a prompt is received. However, when an address is changed, or when the network is started, no ITC232-A is enabled and therefore none sends any prompt back. There are two ways of addressing this situation:

1. By inserting a suitable delay after an address change or when starting a session. The delay can be calculated using 96/Bauds. At the beginning of the session, the boards start at 9600 Bauds. The delay should then be 10 ms or longer. In QBasic, delays that small are difficult to produce. To create a 0.1 second delay in QBasic do the following:

T=Timer + 0.1

DO: LOOP UNTIL t<TIMER

2. When a very short delay is required and the language does not lend itself to short delays, you can send CR on after another until a prompt (>) is returned. This is the way to do it in QBasic assuming the COM port is #1:

DO

DO

PRINT #1, CHR\$(13)

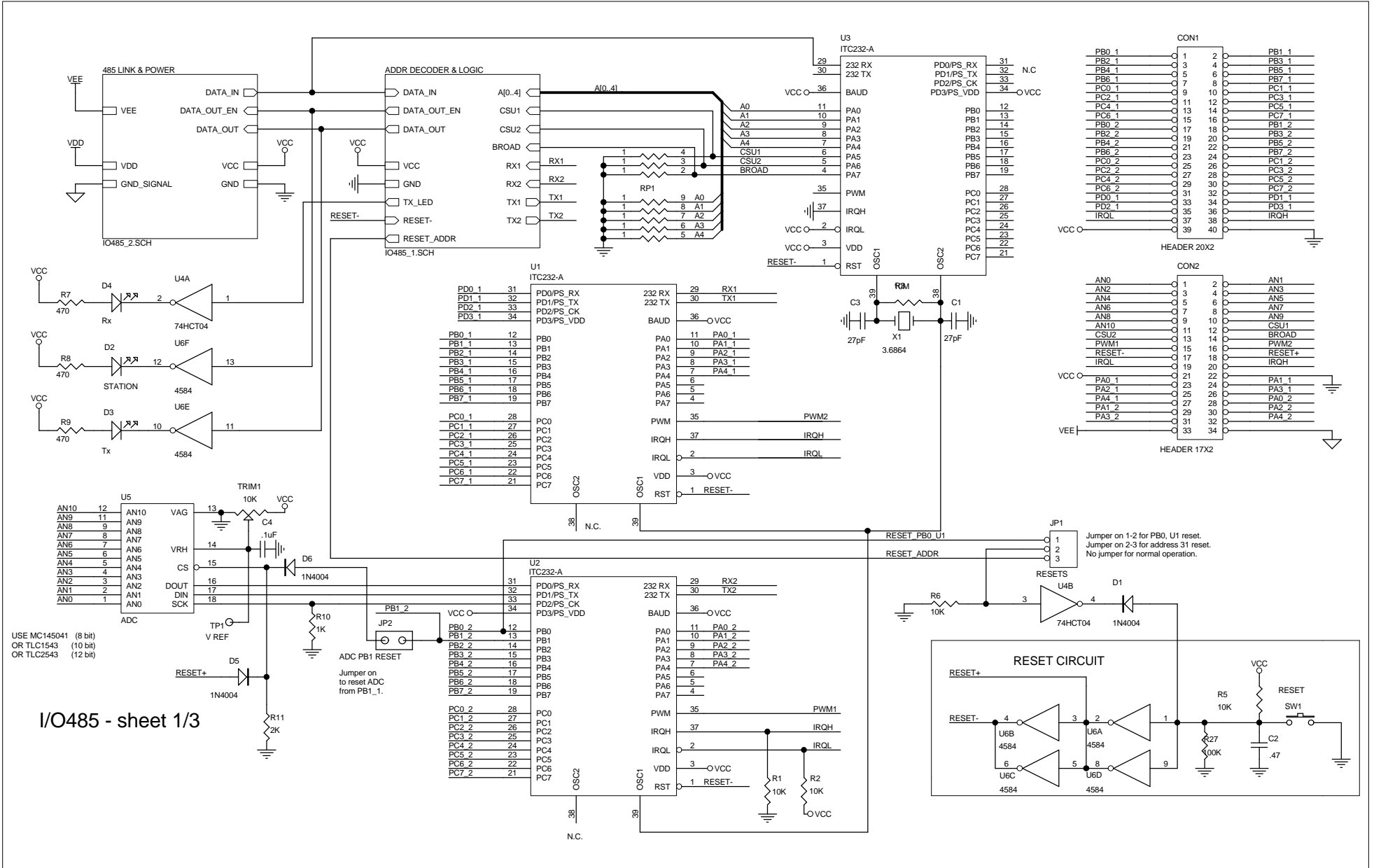
LOOP UNTIL LOC(1)<>0

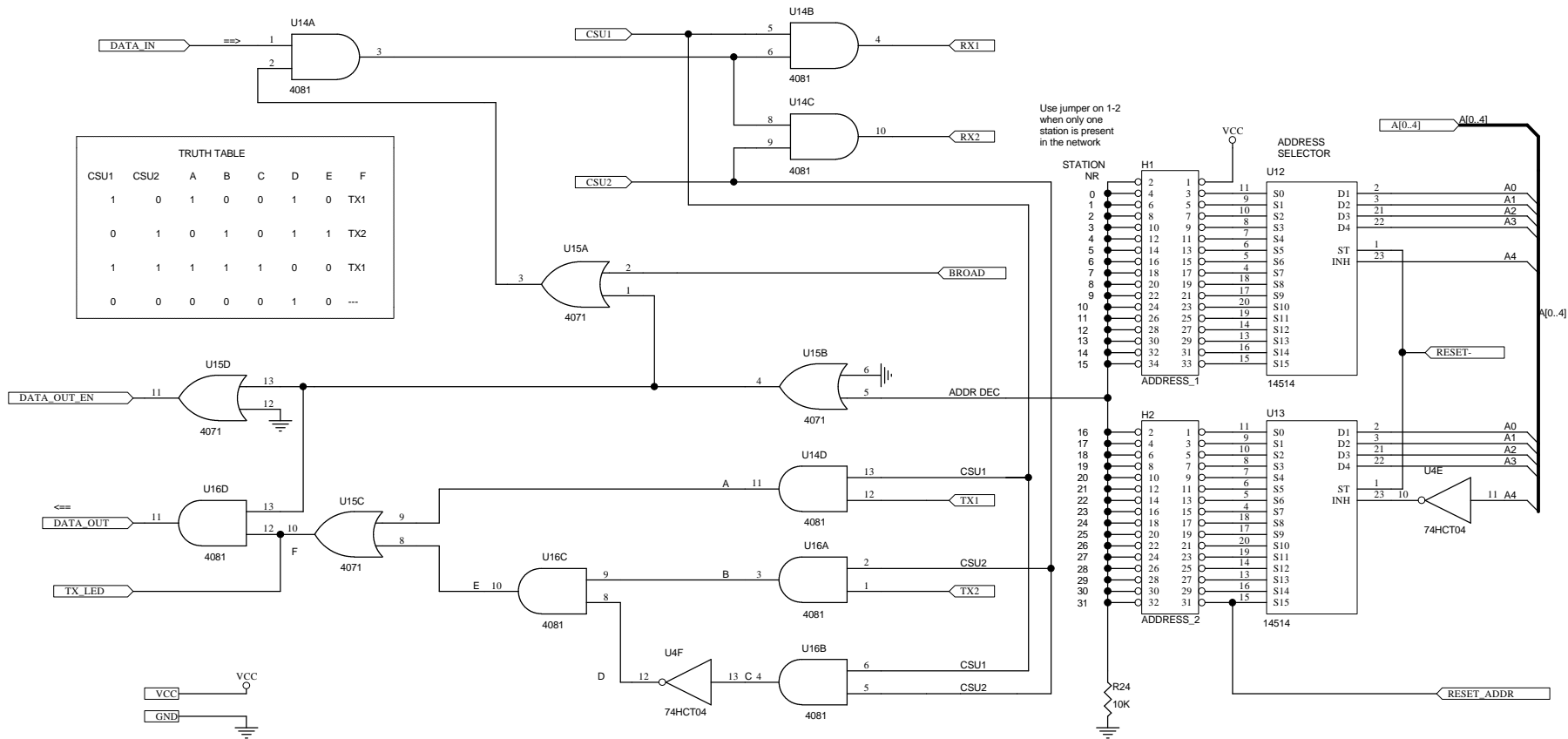
S\$=INPUT\$(1,#1)

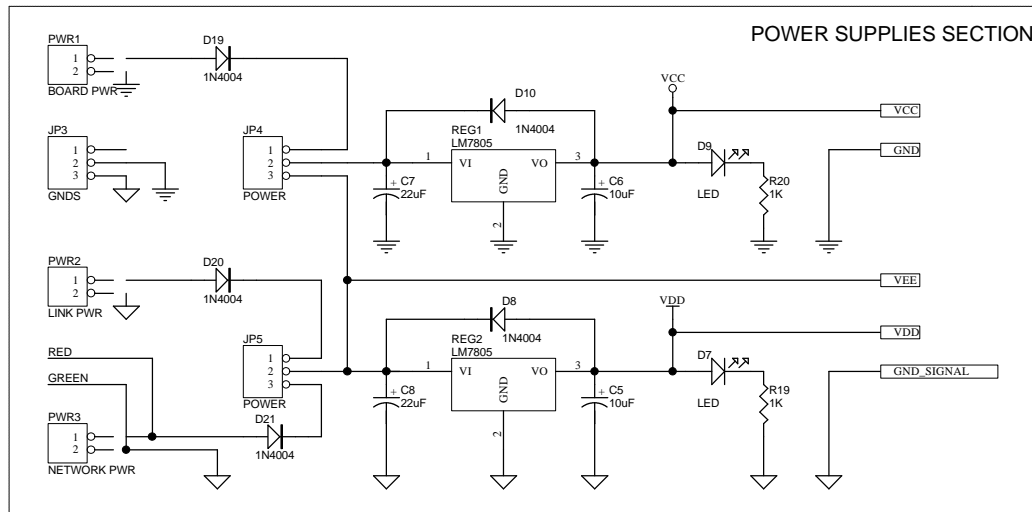
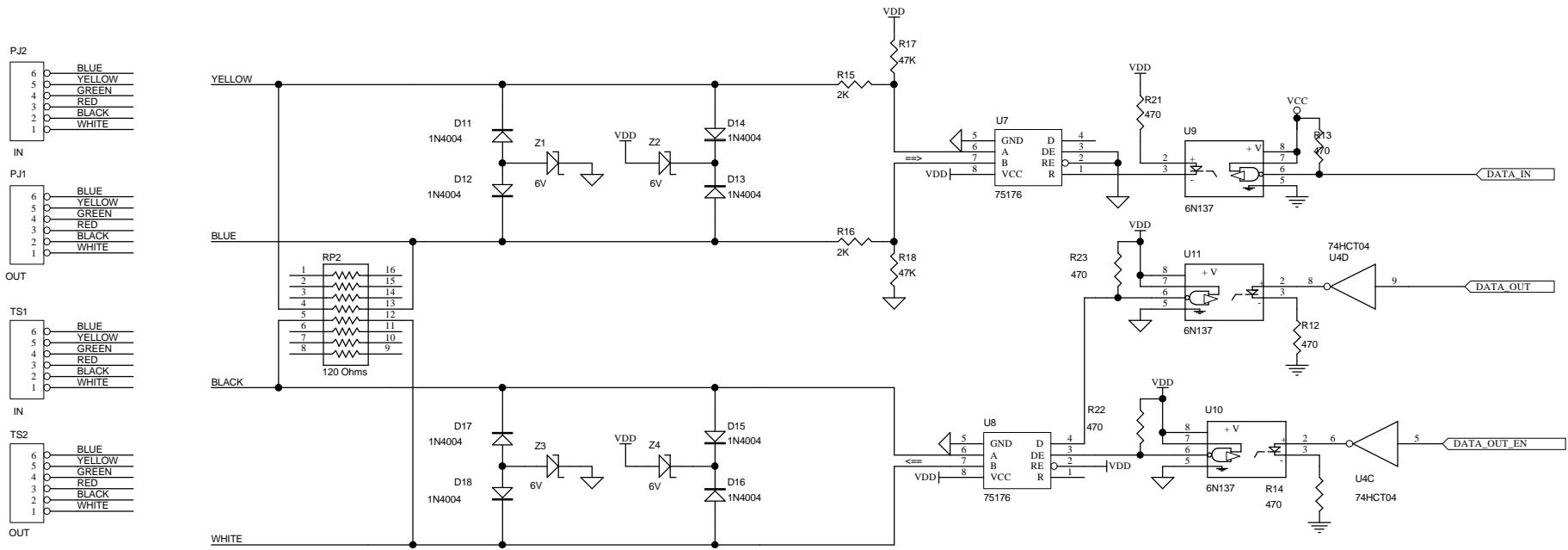
LOOP UNTIL RIGHT\$(RTRIM\$(S\$),1) = ">"

W\$ = " ": GOSUB WriteSerial {Send out an additional CR using a routine that waits for the ">" prompt (such as WriteSerial in the QBasic listing in this manual) to clear the buffer}

SCHEMATICS SHEETS

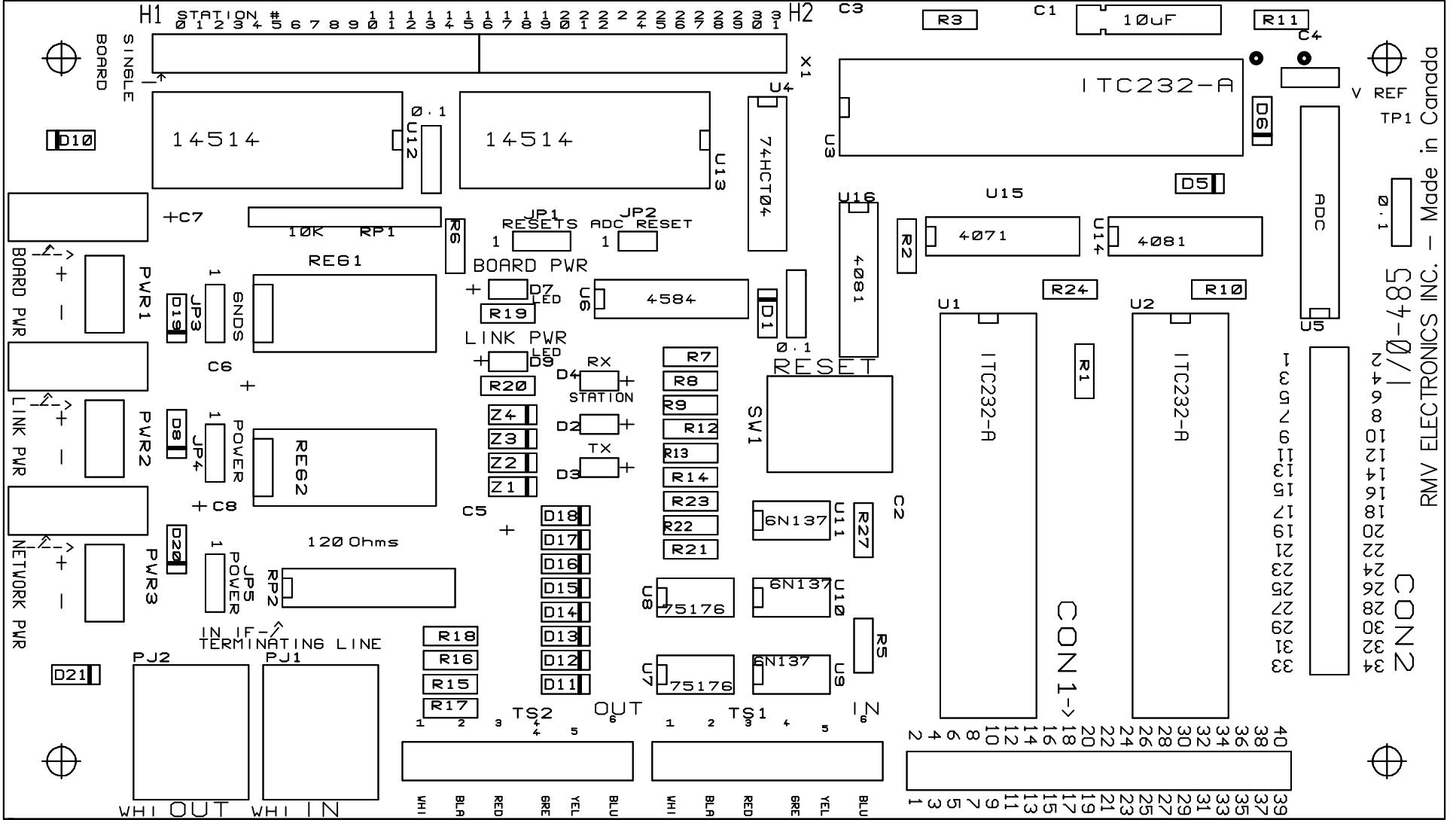






PRINTED CIRCUIT BOARD VIEWS

TOP SILKSCREEN - Obverse view



RMV ELECTRONICS INC. - Made in Canada
 CON2
 0-485
 REF TP1
 0.1
 1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34

TOP LAYER – Obverse view

