OWL2pe-Teleport 2400 Baud Land Line Modem

This is a modem that was originally produced for the Apple computers that had the RS485 5 volt desktop bus. I've modified these modems for use with the OWL2pe data logger and the BASIC Stamp microcomputer. It is well suited for this purpose, because of its 5 volt power requirement and the 5 volt levels of the RS485 interface. The modem itself is well made, with proper isolation of the phone line and a full featured AT command set, and supports Bell and CCITT standards. It is built around the Rockwell/Conextant modem chip, RC224ATLV.

Specifications:

• telephone connection: two-wire dial-up operation for communication over public switched telephone network, RJ-11 jack, extra RJ-11 for extension telephone set.
• computer connection: wires stripped and tinned, see below, power 5.3 to 15 volts dc, 100 ma typical, see below.
• modem standards supported: 2400 CCITT v.22bis, 1200 CCITT v.22 or Bell 212a, & 300 CCITT v.21 or Bell 103
• modem features: asynchronous, AT command set, RC224ATLV, originate or answer modes, nonvolatile configuration memory, pulse or tone dialling. auto-answer/auto-dial.
• size, weight: 1.25" x 3.08" x 5.5", 7.9 oz (3.2cm x 7.8cm x 14cm, 224 grams)
• environmental: 0–55 °C operating, -40 to 70 °C storage, 10–95%RH noncondensing

The modifications have to do with the computer interface, not the modem interface to the telephone line. I remove the Apple connectors, so that the interface connections can be made with bare stripped and tinned wires to the OWL2pe terminals. I convert the RS485 interface to straight RS232 at 5 volt levels by biasing the RS485(+) line at 2.5V with a voltage divider. There are extra wires in the cable, and I've soldered those onto pins inside the modem to have direct access to the ring and carrier detect signals.

©2000, 2006 eme systems (All rights reserved)

Typical connection

The modem will connect to the BASIC Stamp system using the bare wires, as follows:

Black wire: common for all signals and power
Red wire: power supply, 5.3 to 15 volts, 200 milliamp maximum, 70 ma typical
Green: power control from OWL2pe to modem, high (5 volts) turns on modem power, low (0 volts) sleep 30 microamps.
Brown: txd(-) serial data input from OWL2pe or Stamp, 5 volt, restig at mark level = 0 volts
Orange: rxd(-) serial data output from modem to OWL2pe or Stamp, 5 volts, resting at mark level = 0 volts
White: ring signal, modem to Stamp. Open collector optoisolator, requires pullup resistor, active when modem power is off.
Yellow: carrier detect and Break signal from modem to Stamp. Rest low, goes to 5 volts to signal event. See text.

Normal operation:

• Modem power is off, Stamp holds green wire at zero volts.
• OWL2pe/Stamp tests the ring detect line periodically. Ring detect white wire can be attached to either a normal Stamp pin, or to one of the counter inputs. It needs a pullup resistor.
• Ring is detected.
• Power is applied to the modem by bringing the green wire high, Then give time ~2 seconds for initialization.
• If modem is configured for auto-answer, it will pick up the line, or the Stamp can command answer.
• Wait for DCD signal on the yellow wire to go high, detects presence of carrier.
• Stamp transmits its menu or its "ready" sequence to the requester (do not transmit until carrier is present)
• Normal data interchange takes place, but the OWL2pe can also monitor the signal on the yellow wire. If that wire goes low again, and stays there, the carrier has been lost. If it goes low and then high again within 0.5 second, then the host is requesting a BREAK.
• When the communication is finished, the OWL2pe can issue the "+++" then pause, and then ATH to hang up.
• Turn off modem power.

The same procedures are followed if the modem needs to originate a call, except the program does not wait for the ring detect signal, and the modem may be configured in originate instead of answer mode.

The following is an example auto-answer program with connections to OWL pins as described in the program header. This program waits for a ring on the phone line, turns on the modem power when that occurs, then goes online in answer mode (you can hear the tone on the line). It waits up to 5 seconds for a carrier to be returned, and if it does detect a carrier, it jumps to the activeCarrier routine. There it sends out the word "Hello", and waits for the user to type the word "owl" (case sensitive). If it receives that within 20 seconds, it reponds with "got it", and returns to hang up the phone. If it does not receive the "owl" in the time allotted, it prints the "timeout" message and then hangs up. After hanging up it goes back to waiting for the next call. This is a demo meant only to demonstrate the essentials of operation with the OWL2pe.

```
{$STAMP BS2pe}
'{$PBASIC 2.5}
' Black wire and cable shield: common
' Red wire:  battery power 6 or 12 volts steady
' Green: OWL2pe output p10, modem OFF when this is input or low, modem ON when high.
' Orange: OWL2pe input p11, rxd(-) serial data in from modem, resting at level = 0 volts.
' Brown: OWL2pe output p12, txd(-) serial data out to modem resting at level = 0 volts.
' Yellow: OWL2pe input p13, carrier detect & Break signal input from modem.
'         Rests low, goes to 5 volts on event.
' White: OWL2pe input p14, ring signal input from modem.
'        Open collector optoisolator, requires pullup resistor.
'        with pullup resistor to p15 (~10kΩ).

idx VAR word  ' index for loop
```

```
initialize:
  LOW 10    ‘ modem power is off.
  HIGH 15   ‘ enable the ring pullup resistor
  PAUSE 1024

main:
  DO
     DEBUG BIN1 IN14    ‘ show the state of the ring signal, high until ring detected.
     NAP 3     ‘ sleep briefly
  LOOP WHILE IN14      ‘ stays in this loop while ring is inactive
  DEBUG “ring detected!”,CR
  HIGH 10       ‘ turn on the power to the modem
  PAUSE 4096     ‘ time to initialize
  SEROUT 12, 16780, [“ATA”,CR]    ‘ go online in answer mode
  FOR idx=0 TO 500
    DEBUG BIN1 IN13   ‘ testing the carrier detect signal
    IF IN13 THEN activeCarrier
    PAUSE 10   ‘ allow 4 seconds for connection
  NEXT
hangup:
  PAUSE 2048
  SEROUT 12,16780, [“+++”]     ‘ back to AT command mode
  PAUSE 2048
  SEROUT 12, 16780, [“ATH0”,CR]   ‘ hang up
  PAUSE 2048
  LOW 10    ‘ turn off modem power
GOTO main

activeCarrier:
  PAUSE 2048
  SEROUT 12,16780, [CR,“Hello...”, CR]     ‘ send prompt
  SERIN 11,16780,20000,timeout, [WAIT(“owl”)]  ‘ await password “owl” up to 20 seconds
  SEROUT 12,16780, [CR,“Got it!”, CR]     ‘ send data
  GOTO hangup

timeout:
  SEROUT 12,16780, [CR,“Sorry, timed out!”,CR]     ‘ send data
  GOTO hangup
```

Here are specific modifications inside the modem in order to make it easier to use with the OWL2pe/BASIC Stamp.

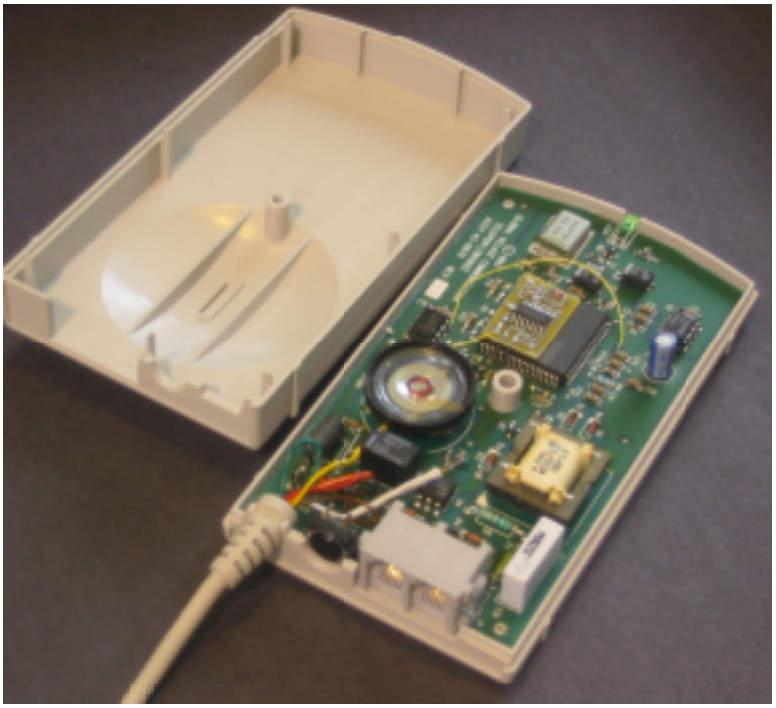
• Power supply, 5.3 to 15 volts with power control (red wire power, green wire control)

• Ring signal, active in power down (white wire)

• Carrier detect and BREAK signal (yellow wire). Logic and timer circuit combines the signals from the carrier detect and txd signals.

• RS485 to TTL serial (orange wire rxd-, brown wire txd-). The (+) input to the modem is internally fixed at 2.5 volts threshold.

• Configuration memory installed (93C46B)



Notes on usage of the modem with the OWL2pe/BASIC Stamp in general

Modems usually are destined for remote sites where you don‘t want any glitches or difficulties. Here are a few tricks that help make the BS2 system with a modem reliable and user-friendly. With a modem, you should never trust that the connection is going to go by the book. If anything can go wrong, it certainly will, sooner or later.

It is a good idea to have a manual or documentation for the modem you are going to use. Modems generally recognize the AT command set, but there are many variations that are there to support enhanced hardware capabilities (such as fax and error correction), and there may be commands that are unsupported in the hardware implementation (such as voice injection). The underlying modem chip defines many of the possibilitiies. The Teleport is built around the Rockwell/Conextant RC224ATLV modem chip, and it is that chip that responds to the AT commands sent from the OWL2pe/BASIC Stamp.

If you want to use the “system” port (pins 1 & 2) on the BS2 for a modem, the trick is to preinitialize the modem (by sending the “E0" command from a regular PC, see below) so that it does not echo commands. Both the BS2 system port and the modem normally echo the characters they receive, so the first command sent from the BS2 to the modem will lock your system up in infinite reverb, unless the echo is turned off. This is not an issue if the data lines to the modem will be connected to one of regular stamp pins, p0 to p15. Those pins do not echo like the p16 port.

You should find a modem that allows you to store a power-up configuration in its eeprom. This is essential if you plan to use the system serial port on BS2 pins 1 and 2. Some cheaper modems do not implement the configuration storage feature. The configuration memory in the modem can relieve the BS2 from having to do the configuration each time at power up. The configuration memory can also store phone numbers for automatic dialling. The modified Teleport modem does have the configuration memory installed.

There are some commands that you will want to store in this non-volatile memory in the modem. That way you can be sure the modem is always initialized in the way that works with your BS2 system. I suggest the following command string:

AT &F E0 Q1 S0=1 &C1 &D3 &W0 &Y0 ^M
where
&F brings in the factory defaults
E0 turns off command echo
Q1 turns off result reporting
S0=1 to autoanswer on the first ring
&C1 enable DCD pin to follow actual carrier status (see below)
&D3 enable hardware reset via DTR drop (see below)
&W0 stores the configuration in eeprom set #0
&Y0 chooses eeprom set #0 as the power-up default.

Some modems don’t support &Yn command and always use set #0 at powerup. The Teleport does support both &Y0 and &Y1.

Result codes are a problem in systems that are set up to auto-answer incoming calls, when using the system port on BS2 pins 1 and 2. On many modems, the echo of the “RING” result code prevents the incoming call from connecting. This not a problem on the regular BS2 pins.

Use the S0=1 or S0=2 command if you want the modem to answer the call on the first or second ring. Use S0=0 if the BS2 will originate the calls (no autoanswer).

If the BS2 will originate calls, you may also want to use the modem‘s capability to store a phone numbers in its memory. The numbers are usually stored with the “Zn” command.

The options for the &C1 and &D3 are discussed in more detail below in connection with the hardware installation.

IMPORTANT: If command echo and result codes are disabled, be sure to attach an advisory note to the modem to that

effect. The note should tell the curious to issue the following command:

AT &F &W0 ^M

which resets the modem to its factory defaults. Otherwise someone at a future date might think the modem is broken.

You may want to turn off the advanced features of data compression and error correction (if it supports them to begin with, which the Teleport does not). The commands to turn off these features are

\N0 turns off error checking
%C0 turns off data compression

Include these in the above string (but not for the Teleport, which does not support them anyway!).

Be sure to turn off these features in the PC as well. This is important. If the error checking and compression is set on the PC end, but the Stamp end does not support those features, the connection can become garbled, or fail, or at least take longer to negotiate. In hyperterminal on the PC, this command can be entered in properties/connect to/configure/advanced-extra settings=\N0%C0
To emphasize again, it is important to turn off the error correction and compression at the PC end, if the PC modem supports it, which most modern PC modems do by default. Of course, if you do have modems that support advanced features on both ends, at the Stamp and the PC, it is fine to go ahead and use them. Test carefully for compatibility.

Here is another command supported by some modems:

S37=6 fixes the first-try baud rate to 2400

This fixes the baud rate at 2400. In hyperterminal, you can enter this in the setup: properties/connect-to/configure/general="2400baud"+"connect only at this speed". The Teleport is by default a 2400 baud modem, so there is no reason for a PC modem to waste time trying to connect at other speeds.

Use the appropriate value for the connection speed you want to establish. By fixing this value, the modems will negotiate the connection more quickly and reliably than if they have to seek out the common ground. Issue the same command both to the modem attached to the OWL2pe/BASIC Stamp and to the modem at the other end of the connection. Quite often older modems are blown away by the bizarre sounds coming from an advanced modem. Do all that you can to match the settings at both ends from the outset.

We usually use the DCD (data carrier detect) line to signal to the OWL2pe that the modem is on line and ready to talk. The DCD line goes high when the modem detects a carrier (when &C1 is is configured as specified above). The DCD line can either go into a regular BS2 pin, to be detected by polling, and/or it can go via a capacitor into the ATN input, to wake up the BS2 for immediate action. I put in both options, so that the program can test the DCD line to help determine the cause of the reset, and branch to the modem routine if appropriate. The RD (ring detect) line from the modem can be added into the mix to give advance warning that the modem is going to be attempting a connection.

It is important not to send anything out through the modem port while the modem is negotiating a connection. Many modems, if they receive characters on their txd line before the connection is established will instantly abort and hang up. Your program must be designed not to send characters to the serial port except when DCD is high..

The DCD line goes from high to low when the modem loses its carrier. The setting in the S10 modem register determines how long after loss of carrier it is before the DCD line goes back low.

Some modems stay on line, receiving noise, even if they lose the carrier. What are the consequences of that? Suppose the BS2 SERIN command will receive one or more bytes. The serin command will probably have a timeout associated with it, so that if it does not receive data within a period of time, it will branch to a special routine to handle the timeout. However, this is far from foolproof with a modem involved. If the modem loses carrier, but still remains connected to the line, then noise from the line can generates random data. It can fool the serin command, so that it never branches to its timeout routine. Any bit of noise is enough to send the serin timer back to the beginning. The upshot is that the system needs a way to reset the BS2 in case anything becomes stuck. Most modems will hang up completely when the carrier is lost, an option in the S register.

One option for hanging up the modem under software control is the +++ ATH0 command. However, that sequence is never a sure thing. The sequence can be corrupted by noise, and sometimes the modem does not "hear" the +++ escape sequence. There is a much more reliable way to reset the modem, via hardware and the DTR line. The DTR line is an input to the modem, and so long as it is kept high, the modem will stay connected. But when the DTR line is brought from high to low, the modem hangs up, resets and executes the initialization commands. That is what happens if the &D3 option is set. There are other options for handling the DTR line, but &D3 option is the most reliable. Hardware reset option is by far the most reliable to guarantee that the modem will be left in a state ready for the next call. A strong alternative to this is the power off and power on. The modem reinitializes when the power cycles. We suggest that all remote modems be connected with means for reset either through DTR or through power cycling.

Another consideration is what will happen with remote commands that cause the the BS2 to go off and do something, such as transmitting back real time data. What if the operator on the other end needs to interrupt the process to return to a main menu? This is not a problem if every command evokes a short quick response. One way to implement a quick response is to add a provision to detect the BREAK condition on the serial line. Normally the RS232 serial line is in a marking (negative voltage) state, and the line pulses positive only briefly to transmit data. A BREAK signal is an intentional positive level for about 0.3 seconds or more. Most PC telecom programs are able to generate a BREAK condition on the serial line. For example, in Hyperterminal, that is done with the CTRL-BREAK key combination. A stamp program can test an input pin on a time scale of 0.2 seconds or less and respond if the BREAK condition is detected.

The OWL2pe-Teleport is wired so that the white wire combines both the DCD and the possibility of a BREAK condition. The white wire normally rests at a low level. It goes high (5 volts) when a carrier is detected and normally stays high so long as the carrier is present. However, if a BREAK condition occurs on the line, the white wire goes back low, and then back high again when the BREAK condition is released. The OWL2pe/Stamp can detect this through polling. Or the white wire can be connected to the ATN input on the OWL2pe/Stamp, which causes a reset on the low to high transition. In the normal operation, that would cause a return to the main menu or command parser.

One more caveat. In some cases where there is heavy danger from lighning, it may be desireable to detach the whole modem circuit from the telephone line via a high-isolation relay, except when a call needs to be made. Or, depend good zap protection and good grounding.