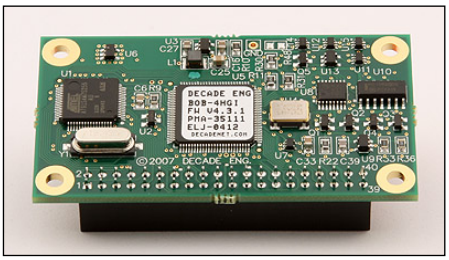
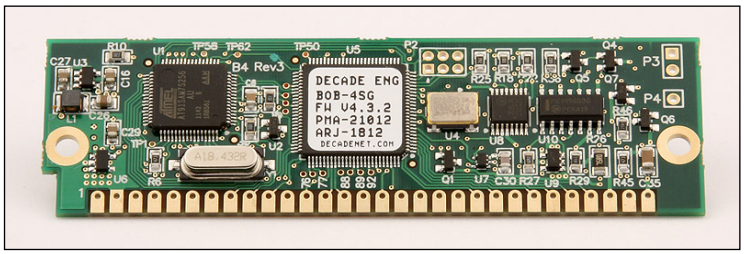
BOB-4 Quick-Start Guide

07 May 2015

*See **www.decadenet.com** for the latest revision of this document.*



BOB-4H (2x20 header format)



BOB-4S (30-pin SIMM 'legacy' format)

Introduction

BOB-4 lets your microcontroller or PC display text and vector graphics on standard TV monitors. The approach taken here is to quickly establish BOB-4 communication with a PC, so the user can type text onto the TV screen and begin experimenting with commands described in the BOB-4 Application Guide. Skills acquired in this way should transfer readily to the development of embedded applications, where most BOB-4 modules are eventually used.

Requirements

To follow these instructions closely, you will need a Windows PC with available serial COM port or USB/Serial adapter, and Bray's Terminal software. I/O Gear USB adapters such as model GUC232A are recommended, but most adapter brands are acceptable for this exercise. Bray's Terminal is a free download and provides a useful macro capability.

In addition to the BOB-4H or BOB-4S module, you will need a breadboard or proto board capable of mounting a compatible connector along with the RS-232 interface circuit in one of the drawings below. Decade Engineering sells 9-pin male/female serial data cables, RS-232 interface ICs, and 30-pin SIMM sockets (for BOB-4S). All other necessary components must be procured elsewhere. For BOB-4H customers, the SIS developer board is a convenient alternative to breadboard construction.

A video signal source is not required, because BOB-4 can generate video on-board, but you should have one if you wish to investigate video overlay performance. Cameras and DVD players with composite video output (yellow RCA jack) generally work well. A TV with video input or TV monitor is required in any case. It's best to have access to more than one TV monitor, because some of the newer flat-screen models can be quirky.

Finally, you will need a quality 5VDC power supply rated at 150mA or greater, with over-current protection. Don't skimp on this requirement. Modern high-density ICs, as used in BOB-4, have nanometer dimensions and exhibit precious little tolerance for electrical abuse. These are NOT your father's 5V logic chips. Power supply problems have damaged most of the BOB-4 modules returned to Decade Engineering for repair. Please read the **Caution** section of this document before proceeding.

Caution

Be certain that you have a stable 5VDC power supply, regulated to ±5% or better, BEFORE proceeding with BOB-4 operation. Applied voltage must never exceed +5.500V under any circumstances, however briefly. Excessive voltage and poor regulation can damage the BOB-4 module, and such damage is not covered by the warranty. Glitches must not occur at turn-on or turn-off time, or when load conditions change. Application circuits should include an electrolytic bypass capacitor of 100~470uF near the +5V input pin. Switch-mode power supplies should include at least one extra L/C filter stage beyond the basic components required for operation. Residual switching spikes should be suppressed to no more than 10 millivolts at BOB-4's power input. See the BOB-4 Power Supply Application Note for additional information.

Header Connectors for BOB-4H

Here are a few example part numbers for 'BergStik' style male pin headers to mate with BOB-4H. These are all conventional THT (thru-hole technology) 40x2 pin headers with .025" square pins and .100" spacing that divide into a pair of 20x2 headers:

NSH-80DB-S1-TG (Robinson-Nugent)
2380-6121TG (3M)
517-6121TG (Mouser)
PH2-80-SGA (Adam Technologies)

SIMM Sockets for BOB-4S

Jameco catalog number 100302 has been tested with good results. 30-pin SIMM sockets are bundled with BOB-4S modules sold by Decade Engineering when the purchase quantity is 1~9 (only). Additional sockets may be ordered under Decade Engineering part number *SIMM-30*.

Hookup Notes

The drawings below show typical BOB-4 hookups to a PC using the asynchronous serial 'TTL-232' main port at 9600 bps. Many additional BOB-4 features are available via hookup options explained in the BOB-4 Application Guide. One option that can be especially useful during application development is a hookup to BOB-4's debug port, which is easily implemented with unused portions of the RS-232 interface IC in these drawings.

ACR0~ACR3 and CTSI are configuration shunt options. Install ACR3 and CTSI shunt connectors or wire as shown.

BOB-4's primary ground pins must be linked through a PCB ground plane or other very-low-impedance connection. Use 22 AWG or larger wire (or multiple wires) in hand-wired installations.

PC COM port pin assignment is for a standard male/female 9-pin modem cable with all pins wired straight through. This hookup will not work with null-modem cables. The PC's data flow control signals are looped back in these examples, so that PC application programs may ignore handshake setup issues. The only PC COM port connections actually *required* are serial data in/out and ground. BOB-4 does support hardware and software flow control schemes, however.

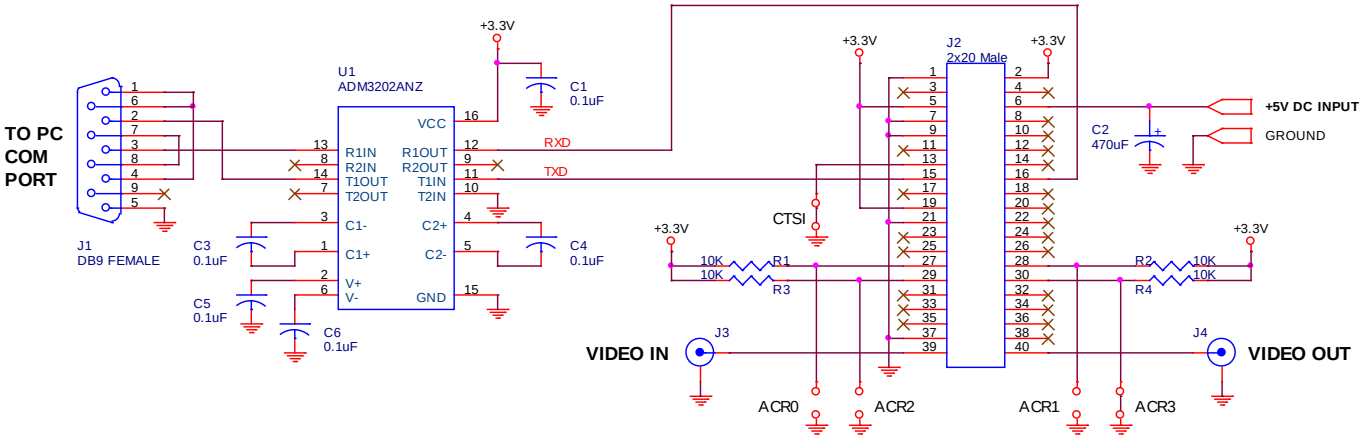
Analog Devices' ADM3202ANZ is supplied in a conventional DIP-16 package (available from Decade Engineering). Substitutions are available from other manufacturers, but older RS-232 interface chips that are not specified for use on 3.3V power will not work correctly.

0.1uF (100nF) capacitor polarity generally isn't marked and doesn't matter, but if you wish to use some that *are* marked, please refer to Analog Devices' ADM3202 data sheet for guidance.

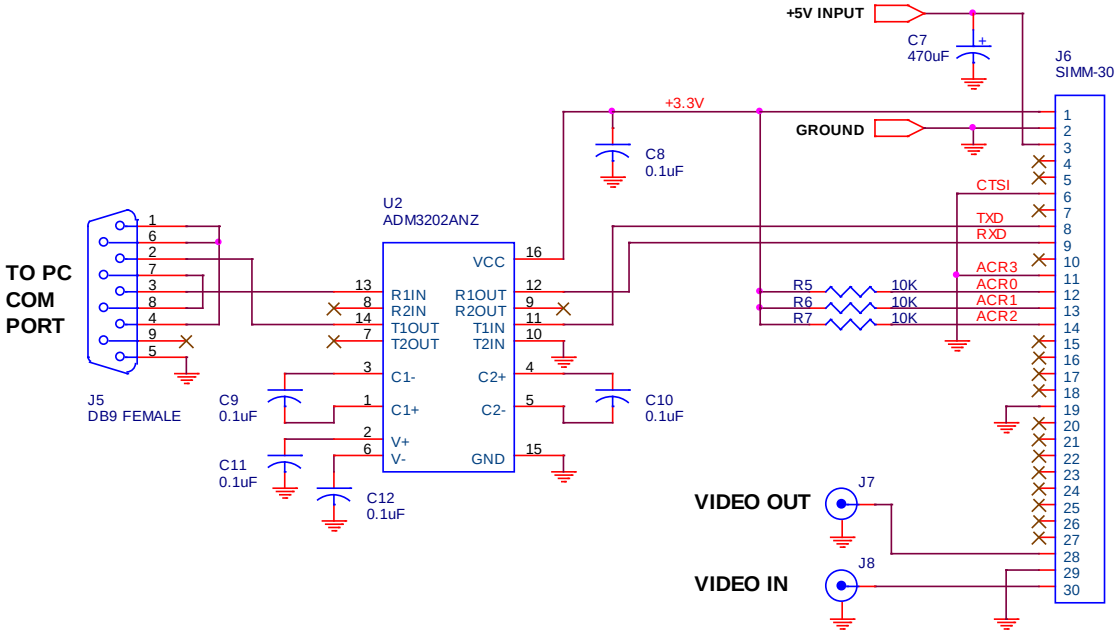
All of the +3.3V connection bubbles must be linked together. BOB-4S (the 30-pin SIMM version) does not output +3.3V power on pin-1 by default. Add a solder blob on the pads marked "J1" (bottom side, near pin number 1) of the BOB-4S module to enable this output.

When circuit construction is complete, remove the BOB-4 module from its static-shield bag and mount it in the application circuit, taking care to follow ESD safety procedures. Connect a high-quality +5VDC power source and TV monitor. Double-check power supply polarity! Video input is optional; connect a composite video signal source if you wish to evaluate genlock-overlay performance. Power up the TV monitor and select the correct video input if necessary. Make sure that BOB-4 is connected to a PC serial COM port, via USB adapter if necessary, and power it up. A big "BOB-4" logo will appear on the TV monitor unless the factory default bootscript has been cleared.

BOB-4H Hookup

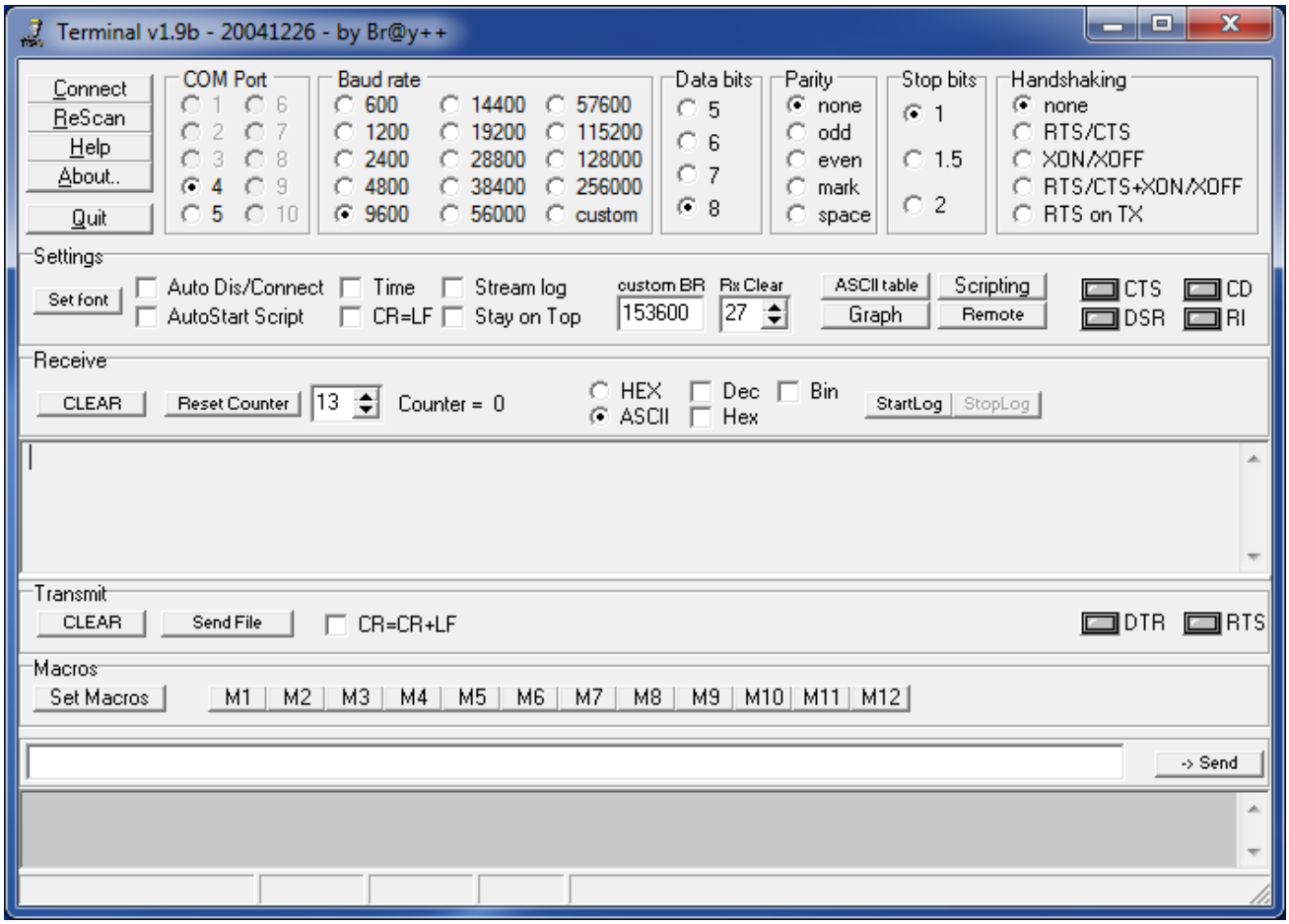


BOB-4S Hookup



Communicating with BOB-4

We're going to use what is arguably the simplest and most direct method to make BOB-4 print text on the TV monitor: a terminal emulation program running on your PC. Terminal programs make the PC behave like a serial terminal — once the standard user interface for central 'mainframe' computers. These exercises were developed with Bray's Terminal V1.9B running on a Win7 Pro 64-bit machine. Terminal doesn't require 'installation.' Just place the downloaded .exe file in a convenient location and launch it. Depending on version, the Terminal window looks something like this:



Terminal should find available COM Ports automatically. If the COM Port number buttons are all gray, confirm that another program isn't using the desired port and click the ReScan button. As shown above, select an available COM Port, set Baud rate to 9600, Data bits to 8, Parity to none, Stop bits to 1, and Handshaking to none. Click **Connect** – the same button should now say **Disconnect**.

In case your PC has assigned a COM port number beyond Terminal's range, open Device Manager > Ports (COM & LPT), and double-click the desired port. A window titled "USB Serial Port (COMx) Properties" should appear. Click the "Port Settings" tab and then click the "Advanced…" button. In the COM Port Number drop-down menu, select an available port assignment within Terminal's range and then click "OK." Click the "ReScan" button in Terminal, which should now recognize the modified port.

To confirm Terminal communication with BOB-4, type some text into the **Send** edit box and click the **Send** button. If your hookup is working, the big BOB-4 logo clears and your text immediately prints using the default 12x13 font in the upper left corner of the TV screen. Garbage printing usually indicates a baud rate mismatch.

Now let's learn to transmit commands to BOB-4. One of the most common application requirements is to clear the screen. BOB-4 obediently does this upon receipt of the **J** command. Here's the **J** command syntax statement as it appears in the BOB-4 Application Guide:

<CSI>nJ

<CSI> stands for Control Sequence Introducer, which must be entered as "**$1b[**" or "**$1B[**" in Terminal. 1B is the ASCII Escape code as a hexadecimal number, while the dollar sign commands translation into ASCII. Other terminal emulators may handle this issue differently. In Bray's Terminal's **Send** edit box, a complete **J** command looks like:

$1b[2J

The **n** parameter in the syntax statement was given a value of 2 in order to clear the entire screen instead of a portion. Click **Send** to transmit the command. If you send another string of printable text after clearing the screen, you will see that print position was also restored to the upper left corner by this version of the **J** command. See the **J** command entry in the BOB-4 Commands section of the App Guide for details.

Carriage-Return <CR> and Line-Feed <LF> codes may be embedded in printable text sent via Terminal in the same way that Escape <ESC> was entered, that is, <CR> is entered as **$0D**, and <LF> is entered as **$0A**. Try them out!

It's often useful to review BOB-4's current configuration. For that purpose, send

$1b[5}

Yup, the right-brace is another command specifier. This command also provides a convenient way to test outbound communication. The lengthy configuration report scrolls beyond the receive window in Terminal, but you can easily scroll back to the beginning if desired. Here's an example report:

```
BOB4 (software v4.3.5, boot v4.3.1, logic rev.22, board v4.2)

Running parameters:
 video mode: local NTSC non-interlaced
 pixel rate: 9.375 MHz
 first pixel position: 123
 last pixel position: 539
 pixels per line: 416
 first line: 39
 last line: 246
 lines per frame: 208
 SPI memory device 0: none (0 bytes)
 SPI memory device 1: none (0 bytes)
 SPI memory device 2: none (0 bytes)
 SPI memory device 3: none (0 bytes)
 slave SPI mode: 0
 comms ACR pins: 0x9 (SPI master, rate = conf)

Configuration:
 NTSC (16=0)
 interlaced: no (17=0)
 external video mode: automatic (18=3)
 external video mode follow: yes (19=1)
 high pixel rate: yes (20=1)
 underscan restriction: yes (21=1)
 frame buffer size: 3 (34560 bytes) (22=3)
 pixel rate: 0 MHz (23=0)
 horizontal position: 0 (24=0)
 horizontal size: 0 (25=0)
 vertical position: 0 (26=0)
```

```
early lines: 0 (36=0)
vertical size: 0 (27=0)
overlay enabled: yes (28=1)
local horizontal adjustment: 54 (29=54)
NTSC chroma rate: 100.000 % (30=100000)
PAL chroma rate: 100.000 % (31=100000)
blinking enabled: no (32=0)
blink period: 10 (33=10)
blink duty: 50 (34=50)
screen flip: normal (35=0)
BOB-4 only mode: yes (9=1)
STX allowed for string start: no (10=0)
CR/LF mode: not set (11=0)
config baud rate: 9600 (40=9600)
flow control: XON/OFF (41=1)
8 bits, 1 stop bit, no parity, echo mode off (42=4)
slave SPI signaling mode: (pin configured) (44=4)
boot script size: 13 bytes (8=13)
l0: low (48=0)
l1: low (49=0)
l2: low (50=0)
l3: low (51=0)
l4: low (52=0)
SPI memory device 0: none (rate=8) (56=0)
SPI memory device 1: none (rate=8) (57=0)
SPI memory device 2: none (rate=8) (58=0)
SPI memory device 3: none (rate=8) (59=0)

Fonts:
   0: bob3          304 chars  12 x 13   2 bpp  internal
   1: 8x13           96 chars  8 x 13    2 bpp  internal
   2: target         14 chars  13 x 13   2 bpp  internal
   3: misc           96 chars  8 x 14    1 bpp  internal
   4: 6x10           96 chars  6 x 10    2 bpp  internal
   5: 13x34          96 chars  13 x 34   2 bpp  internal
   6: 20x40          96 chars  20 x 40   2 bpp  internal
   7: bob4            1 chars  136 x 33  2 bpp  internal
```

Configuration statements such as “16=0” mean that configuration item 16 is set to zero, making BOB-4 compatible with NTSC video rather than PAL video. All of the corresponding **v** commands are listed and described in the App Guide by parameter **n** value: 16 in this case. The example setting could have resulted from a user command entered as follows, where parameter **m** is zero:

```
$1b[16;0v
```

Many of the **v** commands don’t take effect until they’re saved into flash memory and BOB-4 is rebooted. The configuration save command may be issued just once after a salvo of configuration setting commands. Here it is:

```
$1b[1v
```

Have fun with BOB-4, but take care to avoid typing anything other than “**[**“ after the Escape code until you achieve a good understanding of command structure. That little error can be confounding.

This document, although written for XBOB-4, is also highly relevant to BOB-4 applications that involve interfacing with existing data source equipment: XBOB-4 Quick-Start Guide & POS Tutorial

Ω