

Quick Start Guide for using the BV4107 Keypad Interface



1. Introduction

The BV4107 is a keypad interface. It will interface to a keypad with up to 16 switches in a 4 x 4 matrix. Other configurations can be used, simply leave out a row or column.

This quick start will go through the steps of interfacing to a 3 x 4 keypad.

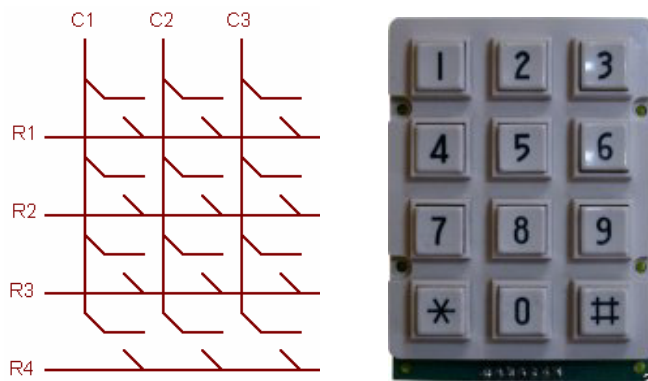


Figure 1 Keypad used for quick start

Above is a schematic of a typical 4 x 3 keypad switch. In this example there are 4 rows by 3 columns. The actual keypad is shown alongside. Using a continuity meter and some patience the pins for the rows and columns can be worked out. They vary from pad to pad.

2. BV4107 Connections

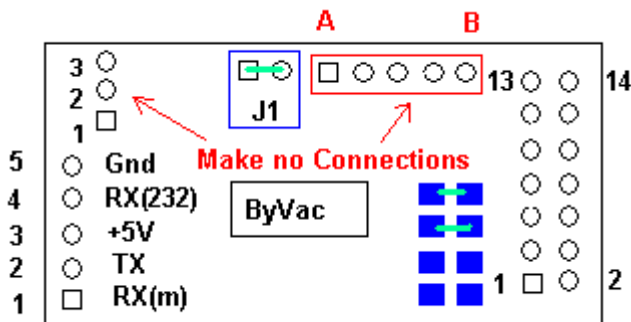


Figure 2 The BV4107 Module connections

All of the row and column connections are on the 14 pin connector. The table shows which pins to use.

Pin	Cmd	Description
1		Ground
2		Reference [1] (output)
3		C1
4		C2
5		Ground
6		C3
7		C4
8		+Ve Power (also pin 3 IASI)
9	KI	Digital input
10		n/c
11		R1
12		R2
13		R3
14		R4

Figure 3 Connection for the 14 way connector

In this application the keypad only has three columns so the forth C4 will not be used. The circuit diagram is shown here. There is no need to connect anything to C4 (pin 7) as this has an internal pull up resistor.

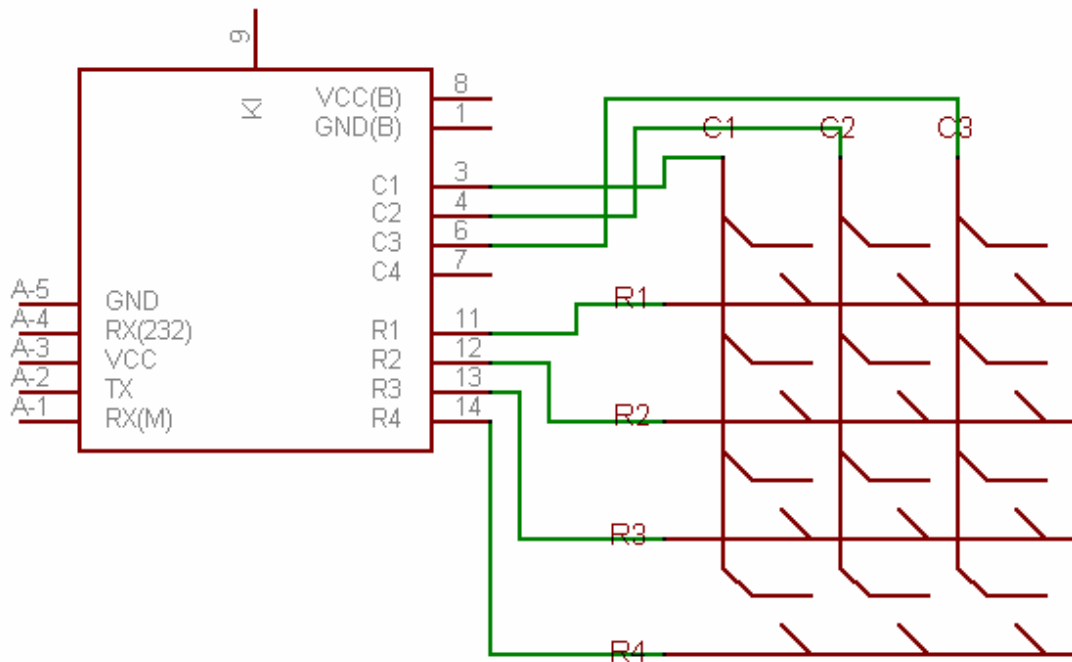


Figure 4 Connection to BV4107

2.1. The IASI Connection

This is the 5 way connection to the left of the board and is the communication to either a PC or a microcontroller.

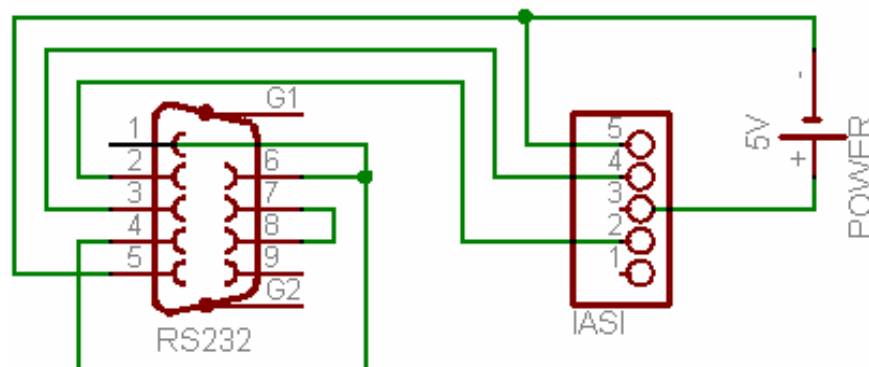


Figure 5 Connecting to PC COM Port

The power to the module requires about 4 to 6V but if logic is to be controlled (quite likely) then a regulated 5V supply will be required between pins 3 and 5. Figure 5 shows how to connect the interface to a standard Com. port. There is no need for a voltage translation device such as a MAX232 if only one device is being used. NOTE Pin 1 is always a square pin.

Once connected the device will be ready to respond to commands issued from the PC. a terminal emulator program should be used such as HyperTerminal, however a free terminal emulator is available and much easier to use called BV terminal. This is included in this pack or it can be downloaded from www.byvac.co.uk

3. Terminal Software

Any terminal software (software that is used with the COM ports) can be used for this display, if you have Microsoft Windows then ideally use BV Terminal. This can be obtained free from www.byvac.co.uk. This is only one file, simply drop the exe file into a suitable directory and launch, there is no installation or set up. The file will write a configuration file to the directory to keep track of the settings, but that's all.

The next section instructions are for BV Terminal, see later for setting up HyperTerminal.

4. Using BV Terminal

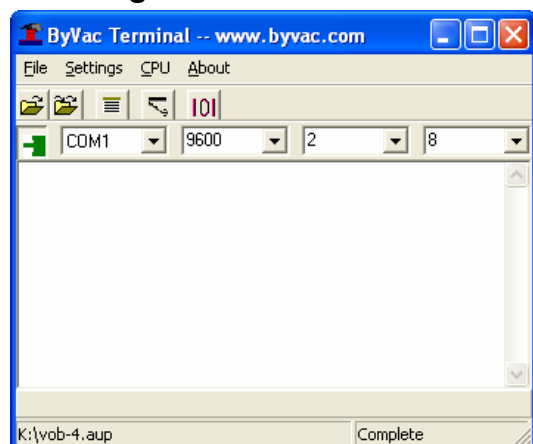


Figure 6 BV Terminal Software

5. Establishing Communication

Select the Com. port that you have the device connected to using the drop down list, only available Com. ports will be shown. Press the small red icon to the left of the BV Terminal window and this should turn green to show that a connection to the selected COM port has been made. If you get an error at this stage then you have selected the wrong COM port or you do not have a COM port on your system.

When the icon is green press <enter> a couple of times until a **L>** prompt is seen, this process establishes the baud rate. In Figure 6 this is shown as 9600 but it can be any you choose up to 57,000. There is no real advantage at this stage for having a higher baud rate, see Figure 7.

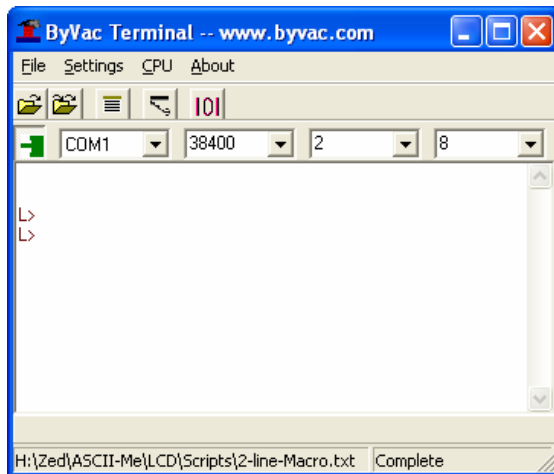


Figure 7 Establishing the Baud Rate

As can be seen from Figure 7 the Baud Rate will be established so setting the actual baud rate on the terminal is not that important.

6. A few Simple Commands

Try some simple commands to become familiar with the interface, for example:

za

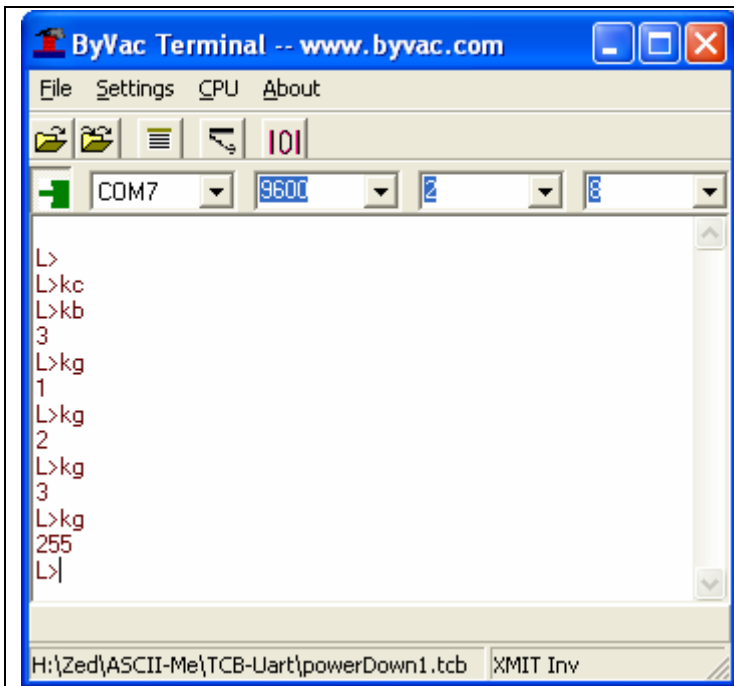
The above will return the current address, this will be 00 by default, try changing it to say D1 by:

zaD1

Try **zv**, this will print the version details. Once communication has been established try some simple commands:

7. Configuring the keypad

If the keypad has been wired according to the diagram in Figure 4 then the key values should match the numbers as illustrated on the keypad in Figure 1. When a 1 is pressed 1 should be returned. Type **KC** and then press keys 1,2 and 3 on the keypad, type **KB** and this should return 3, as this was the number of keys pressed and they are stored in the buffer. Now type **KG** four times and this is the result:



Notice that the result is returned in the order that the keys were pressed. When the buffer is empty (the forth KG command), by default the value returned is 255 but this can be set to any value using the KM command.

8. Setting up the key map

The above may not have produced the result you expect this will be because the key map is not set for how you have wired the keypad. The default keymap values and associated scam codes are as follows:

R1	14	30	46	62
R2	13	29	45	61
R3	11	27	43	59
R4	7	23	39	55
	C1	C2	C3	C4

Scan codes in decimal accessed by the KS command

R1	E	1E	2E	3E
R2	D	1D	2D	3D
R3	B	1B	2B	3B
R4	7	17	27	37
	C1	C2	C3	C4

Scan codes in HEX

R1	0	4	8	12
R2	1	5	9	13
R3	2	6	10	14
R4	3	7	11	15
	C1	C2	C3	C4

EEPROM address map

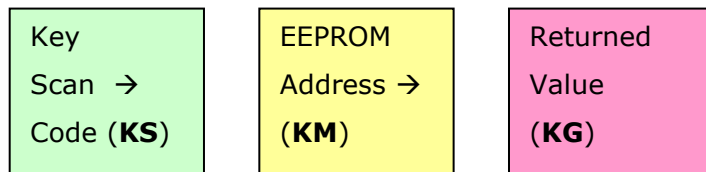
Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Value	1	4	7	15	2	5	8	0	3	6	9	14	10	11	12	13	255

If this is thought in terms of switching rows and columns then, connecting row 1 to column 1 will return a scan code of 14, likewise row 3 column 2 will return a scan code of 27. By using the **KS** command that returns the scan code it can be determined which row and column are connected together.

As an example if KS returned 45 which row and column would be connected? Looking at the first table it can be seen that 45 is on the junction between row 2 and column 3.

The **KG** command translates the scan code into an address that has a value of between 0 and 16. This is shown in yellow on the third table, scan code 45 for example would be translated to address 9. **KG** takes this one step further and uses this address to look up a value in EEPROM shown in pink as the last line of the long table.

For the keypad layout in this example those values correspond to the values printed on the keys, with 15 being assigned to * and 14 to #. The last value on the table at address 16 is the value returned by **KG** if the keyboard buffer is empty.



This illustrates the process and associated commands.

The values in the EEPROM table can be read and set by using the **KM** command: **KM2** will return a value of 7 and **KM16 33** will set the buffer empty return value to 33.