Building the N8VEM Single-Board Computer for the Total Beginner

This is a guide to help with building the N8VEM Single-Board Computer (or SBC). It is aimed at the total beginner, and I will do my best to not assume much of any previous electronics experience or knowledge.

I built my 1st circuit board just a month or two before my N8VEM, and I had no electronics experience prior to that. I had a solid foundation in programming, but did not know much about hardware side, however, I was motivated to learn. I found the construction process to be a bit confusing at times, but it is my hope that this guide will help those who have limited experience with these things as I did.

The basic outline is as follows:

1) order the N8VEM SBC printed circuit board (pcb)
2) obtain the parts list
3) order the parts
4) construct the board and other required elements
5) place the ICs (“integrated circuits” or chips)
6) configure the board
7) test the board
8) next steps

OK, so let’s start!

1) Order the N8VEM SBC pcb
   a. OK, so this is the easy part. Andrew Lynch is the originator of this project and he manages the pcbs, so you need to contact him at lynchaj@yahoo.com and tell him you want to purchase an N8VEM SBC pcb.

2) Obtain the parts list
   a. I presume that since you found this guide, you must have found the wiki site for the N8VEM project. Anyway, it’s here:
      http://n8vem-sbc.pbworks.com/w/page/4200908/FrontPage

      On this site, you can find folders that contain a bunch of documents for each separate pcb that is offered for the N8VEM system. They provide different functions from keyboard and mouse control to VGA output to color graphics and sound to mass storage. Here is where you can get the most up-to-date information about the parts you will need.

      As of April 2011, the N8VEM SBC is at version 2, and the parts I used to build mine are listed in Appendix A in the back of this document.

3) Order the parts
   a. My main suppliers are Jameco (www.jameco.com) and DigiKey (www.digikey.com). I also use eBay for various items that are not available at either Jameco or DigiKey.
   b. In general, I find Jameco to have better prices and their website easier to use. With DigiKey, their search engine which is really powerful once you know how to use it (and you know exactly what you want) and they have superfast service.
c. Tips:
   i. Make sure you check that you’re ordering the right part! Check the part against the pcb (either the physical pcb sitting in front of you, or a detailed picture of it). This is to make sure the part will fit properly.
   ii. Each part will have a DATASHEET on it (usually an Adobe Acrobat (.pdf) file which is available as a link on the website you’re ordering from) which will give you the details about the item, along with detailed measurements.
   iii. If you’re buying resistors, make sure you’ve got the right OHMS.
   iv. For capacitors, make sure you’ve got the right FARADS.
   v. For resistor packs (aka resistor arrays or resistor networks, abbreviated RR) make sure you get the right OHMS and the right type (BUSSED or ISOLATED) and the right number of pins. Every resistor pack I’ve used on the N8VEM project is of the BUSSED type.
   vi. For Integrated Circuits (or ICs, or chips), as long as number of the chip is very similar, you’re usually OK. For example, a 74LS245 chip may be available as 74LS245N or SN74LS245N, which will be very similar, but have very subtle differences. No solid rules here, but if you’re in doubt, ask a question on the discussion forum at http://groups.google.com/group/n8vem/topics.
   vii. IC sockets are recommended for ALL ICs, because they allow you to remove and switch out chips at will, without having to desolder and resolder the ICs on the pcb. Basically, if you’ve got a 14-pin IC you need a 14-pin socket, a 20-pin IC needs a 20-pin socket, and so on. You solder the socket to the pcb, and you plug the IC into the socket.
   viii. Look at one of the N8VEM pcbs or a picture of one. The solder holes (the tiny holes in the pcb with a thin silvery edge) that are the closest together are 0.1 inches (which is 2.54mm) and the larger distances are usually factors of this (0.2 inches, 0.3 inches, etc). REMEMBER THIS. Use this info to make calculate if the parts you order will fit in your pcb. Or you could just measure your spacing to be sure.

4) Construct the board!
   a. OK, this is the really fun part. I usually start with the chip sockets (which are recommended for ALL chips). First, take the pcb and lay it on a static-free surface (I usually use a medium to large anti-static bag for this) with the “silkscreen” side upwards (the side with the while lettering on it and outlines of the parts). Place the chip sockets in, making sure the notch in the socket matches the notch on the silkscreen outline (usually pointing to the left). Since the pcb is lying on the table, the pins will not penetrate all the way through, but that’s okay. You’re just getting them in place for now.

   ![N8VEM PCB](image)

When they’re all in, place a stiff, flat surface that covers the TOP of the all the sockets (a firm piece of card works well, or an old CD works for a smaller pcb). You’re going to use this to flip the pcb over.
without all the sockets flying off. So grip the pcb and the firm card together (with the sockets
sandwiched in-between), and flip the thing over and lie it down on the table again. The pcb should be
flipped over now, with the pins from the IC sockets should be pointing up. The pcb is now in soldering
position.

Soldering probably needs its own whole section, but I’m just going to do a flyby. I learned to solder by
watching videos on YouTube. There’s an excellent one here:
http://www.youtube.com/watch?v=I_NU2ruzyc4 (“How and WHY to Solder Correctly” by
CuriousInventor on YouTube). I used a 15-watt electronics soldering iron with a narrow tip from
RadioShack along with 60/40 Rosin-Core Solder (0.032” diameter). I also bought a few small cheap
circuit boards (with holes and foil rings around the holes), stuck bare wires in them, and practiced
soldering on this setup first to get the timing right before taking the heat to my N8VEM pcbs. I also used
a Donegan head-magnifier with 2.5x lenses to do almost all of my soldering. Since you wear it on your
head, it frees up your hands. Magnification helps, because you can see the solder liquefy and be sucked
into each joint, and you can be sure your joint is good. It’s also binocular, so you have good depth
perception. I’ve never had a cold solder joint using one to help me solder and I’m just a beginner, so I
recommend using this magnifier. It’s not a necessity, though. There are a thousand ways to do
everything and you will find your own way of doing things, too.

OK. Once the sockets are all in and soldered on, I then put in the other parts. Many of the other parts
(resistors, capacitors, etc) have long wires. What I do is bend the wires with a pair of small pliers
(because I like nice 90-degree bends) before inserting them into the holes (from the silk-screen side),
and when they’re in place, bend the protruding portion of the wires so they’re at about 45 degrees to
the surface of the pcb. This bend keeps the part in place while the pcb is upside-down so you can solder
it. Here’s an example:

ONE MORE IMPORTANT NOTE: make sure the part you are putting in is the correct part and that it is
oriented correctly. This is VERY IMPORTANT, because if you use the wrong part in and solder it in, it is a
major pain to remove it (by desoldering) and replace it. I tend to check, double check, and triple check
everything before it goes into a solder hole.

After you’ve soldered the part in, take some wire clippers and very gently clip off the excess wire close
to the solder joint. Make sure you do this gently, because if you twist the wire as you cut it, you may
disrupt or break the solder joint. Some people recommend clipping the wire BEFORE soldering.

Certain parts (such as diodes, LEDs, resistor arrays, large cylindrical/barrel capacitors, etc.) have a
certain direction (or polarity) that they have to go in. Bussed resistor arrays have the 1st pin marked with a dot printed on the part itself. This pin goes in pin 1 of the pcb. Pin 1 for every part is always marked with a square solder hole on the pcb. Individual resistors and non-barrel capacitors can go in any direction. If you’re not sure, remember to check the Datasheet for the part from the website you bought it from (Jameco, DigiKey, or other).

Another note: do this in batches. I tend to solder in batches of parts at a time. That is, I put 20-30 parts in, solder them all, clip the wires, and then do another batch. I’ve found that if I put too many wires in at one time, the protruding wires get in the way of the soldering iron, and soldering becomes a pain.

Work your way through all the parts, making sure you’re putting the right parts in the right places, and with the correct orientation. Some parts (like headers – basically rows of pins used for plugging in cables, etc.) may need you to tape them into place so they don’t fall out (on the non-solder side of the pcb) as you solder them.

5) Place the ICs (chips)
   a. So now the soldering is complete! Now is the time to place the chips into the sockets. Make sure you follow a picture of the bare pcb (so you can read the silkscreen which has the chip IDs on them) to make sure you’re putting the right chips in the right sockets. I do this because usually the IC sockets will cover up most of the important identifiers printed on the pcb so this is why it’s important to see an image of the bare pcb as a guide.

   You will likely need to bend the pins of the ICs inwards a bit to make them fit into the chip sockets. To do this, hold the chip firmly by the ends without pins, place the row of pins you want to bend onto a hard flat surface, and apply steady pressure until they bend in the desired amount. This way, all the pins are bent nice and evenly. Go slowly, though, because it’s much easier to bend pins “in” then it is to bend them “out”. 
6) Configure the board  
   a. Now you’ve got to place the option jumpers on the board so it functions correctly. The N8VEM SBC V2 pcb has twelve 3-pin headers (labeled K1-K12). The standard jumper positions are in the 1-2 position. In other words, you place a 2-pin jumper over pins 1 and 2, leaving pin 3 open. Remember, pin 1 has the square solder hole. If you can’t tell looking at your pcb, load up a photo of the bare pcb (either yours or one on the N8VEM wiki). The jumper positions and their functions are provided in Appendix B.

This is the completed N8VEM SBC v2 board with chips and jumpers inserted:

7) Test the board  
   a. The N8VEM SBC is a single board computer without any video capabilities in its solitary form. Therefore, in order to test it, you will need to connect it to a terminal (which will provide the keyboard input and monitor output). The easiest way to do this is to construct a cable which will provide a serial port (with a D-sub connector or DE-9 plug) which will plug into a “host” computer running a terminal emulation program. This program will allow you to interface with the N8VEM using the host computer’s keyboard and monitor.

First, you must build a cable with an IDC-10 plug on one end (plastic rectangular connector with 2 rows of 5 pins) and a female DE-9 plug (ie. a serial port plug) on the other end. Here’s a picture of an IDC-10:
This female DE-9 plug is what you will plug into the serial port of your host computer. Serial ports (on the back of the computer) are male ports (ie. they have pins), so the plug at the end of this cable must be a female plug (they have holes). There are some builders that use combinations of male/female plugs connected to various cables, null-modem cables, etc. This way can definitely work, but I think this method can be very complicated, is dependent on too many other factors (what the specific cable wiring pattern is), and can be very hit-or-miss. I prefer not to do it this way. I prefer to build the cable from the beginning knowing exactly what is going where and why.

The following cable layout shows what is being connected where. The pins must be connected as follows:

<table>
<thead>
<tr>
<th>IDC-10 side</th>
<th>DE-9 side</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

The #1 pin on the IDC plug is marked with an embossed triangle on the plug, and this pin corresponds to pin #1 on the pcb which is marked with a square solder hole (at the lower-right most position of the plug on the N8VEM SBC V2 pcb). The pins on the DE-9 plug are usually marked right on the plug itself in tiny numbers. Strip some wires and solder away. You can also use ribbon cable and plugs that connect directly to the ribbon cable without soldering. Because of the non-standard pin connections, however, you will likely need to do some soldering or “custom connecting” depending on the parts you order.

Always double and triple check where you’re soldering something before you solder it. When you’ve built your cable, use your multimeter to check connectivity between each pin on the IDC-10 side and the DE-9 side according to the arrangement above to make sure you got it right.

In order to test connections in a plug you cannot stick the multimeter lead into the hole (because it will not fit). Instead, take a spare piece of wire, stick it into the hole for the pin you want to test, and then touch the multimeter lead to that wire, as follows:
Once the cable tests as expected, you’re ready to move on!

b. Program the EPROM

i. To do this, you will need an **EPROM programmer**. An **EPROM** is a chip that you can load information onto that will remain there (unlike RAM which gets erased when you disconnect the power). It basically becomes a little read-only “hard-drive” that holds the system files. The only difference is that it is Read-Only Memory (ROM) once it’s plugged into your N8VEM, so it cannot be used to store data like a real hard-drive. EPROM stands for **Eraseable Programmable Read-Only Memory**, which means it can be erased (with an ultraviolet light EPROM eraser) and then reprogrammed (with an EPROM programmer).

A **UV light EPROM eraser** costs about $20 from many sellers on eBay. It erases the EPROM chip by shining a light into the window on top of the chip for about 15-20 minutes and erases the chip.

An **EPROM programmer** is a device that plugs into a computer (via a serial/parallel/USB cable), and has an IC socket in it where you plug your EPROM into. You then load a binary image file into your PC, and it sends this image file to your chip, thus “programming” it. You can get these on eBay for around $40. There is some debate about which EPROM programmer is best. I own the **Willem EPROM Programmer 5.0e**, and while it works, it is somewhat cumbersome to use and I believe there are probably better choices out there. That being said, it is a very popular programmer, so there’s lots of help out there if you get stuck. Each EPROM programmer is capable of programming numerous chip types and to do this, the programmer has numerous different settings (numbers of pins, timings, etc) which have to be matched to the EPROM chip type.

For the **Willem EPROM programmer**, you select the chip type from a menu in the software, and it tells you how to set the dip-switches on the **programmer device**. It shows you a picture of the dip-switch settings, so this is fairly straight-forward. Where it gets tricky is that some chips also require you to change **jumpers** on the board as well, and this is not documented very clearly. Not setting these parameters correctly can result in damaging the EPROM chip. Also, the Willem programmer requires a **serial** interface to communicate to the device, AND either a **USB** cable or a **9v power adapter** to provide power to the board.

Some of the newer EPROM programmers will set the programmer parameters electronically (not requiring the user to set any dip switches or jumpers), and only require a **USB** to
communicate to the programmer AND provide power.

ii. Find a **system build/BIOS** image to load onto your EPROM chip (mine is a 27C801 chip for the N8VEM SBC V2).

You can do this from the N8VEM wiki page in the **“Software information” folder -> “Software and ROM Images” folder**. Unfortunately, finding a suitable BIOS can be tricky (because each is built to support different configurations of hardware), but I would recommend the following image for 1st time builders:

**Romimage_Feb28_2009_TOOLS_Zork.bin** (in the “Feb 28 2009 Build” sub-folder)

I like this image because it contains basic **tools** and **utilities** for transferring files to your N8VEM through the terminal program on your host PC, testing your hardware, has a **RAM Drive** (A:), and has a bunch of cool stuff on the ROM drive (F:) like Zork, Wordstar, Microsoft BASIC, etc.

iii. Start up your EPROM programmer software, plug in your EPROM programmer, and configure it for your EPROM. Details on configuration and setup are in **Appendix C** for my setup (using a Willem EPROM programmer to program a 27C801 EPROM using a windows system)

iv. Plug in your **erased EPROM** into the IC socket in the programmer.

v. Check that the EPROM is empty (select the menu item “**Action**”->“**Blank check**”). It should then display a message after a few minutes verifying your EPROM is empty. If not, put the EPROM back into the EPROM Eraser and erase it for another 5-10 minutes and recheck it by repeating this step.

vi. Load your **ROM image** into the EPROM programmer software (“**File**” -> “**Load**”), and “program” your EPROM chip (“**Action**” -> “**Program/test RAM**”). This can take up to 10 minutes or so with the Willem programmer.

   When the programming is complete it will automatically check the EPROM against the image it just loaded onto it to make sure it’s correct. If it is not, you will get an error message. This verify step should match perfectly, confirming you programmed the EPROM successfully. Once your EPROM chip is successfully programmed, cover the erase window on it with a **small opaque sticker**. This is to prevent your data from being corrupted or erased by ambient UV light.

   If your EPROM did not program successfully, check all your jumper settings and cables, erase the EPROM, and try to burn the image to the EPROM again.

   Failed EPROM programming can be due to failing to fully erase the EPROM using the UV light eraser. I have found that erasing it for 15-20 minutes works for my setup and chip, but your mileage may vary. Be careful not to “over-erase” which can damage the chip. If you have trouble with this, ask on the Discussion Forum.
vii. **Congratulations!** You’ve programmed your EPROM which now contains the system files you will need to boot your N8VEM. Plug this chip into your N8VEM (making sure you plug it in the right way by using the notch on the top of the chip as a reference and matching this to the socket).

c. **Turn it on!!!!!**

i. OK, time to check to see if all the work has paid off.

ii. Set up your host computer
   1. Load your terminal emulation program and check the settings (these are the correct settings for the ROM image I indicated earlier):
      
      | Setting     | Value     |
      |-------------|-----------|
      | Baud:       | 38400     |
      | Data bits:  | 8         |
      | Parity:     | None      |
      | Stop Bits:  | 1         |
      | Flow Control: | None    |

iii. Plug one end of the **N8VEM serial cable** you built earlier into the N8VEM SBC (P3 header) and the other end into your host computer serial port.

iv. Plug in the **power source** to the N8VEM. When you do this, I recommend watching the board very closely for the first minute, making sure it doesn’t start smoking anywhere and checking the chips and power supply to make sure they don’t get too hot too quickly. If anything alarming happens, **disconnect** the power source immediately and ask for help on the Discussion Forums.

v. Switch on the N8VEM! Wait a few seconds, check again that nothing is smoking, and then press the “reset” pushbutton (SW1).

vi. If everything is perfect, the terminal screen on the host computer should display a welcome screen which should look like the following (for the ROM image selected earlier):

   ![Welcome Screen](image)

If so, **congratulations!** You’ve successfully built a single-board computer, loaded the operating system onto it, and interfaced to it using a cable you built as well! Great job!
If you see a screen of garbled characters like this:

you still may be okay. The most common reason for garbled characters is an incorrect BAUD setting, so double-check that first, and compare to the BAUD setting that your BIOS is built to work at (usually listed in the description file for the BIOS image you burned to the EPROM). There are other reasons that could account for this, but checking the BAUD setting on the terminal program is the quickest and easiest place to start. I ran into this problem with my board, and correcting the baud speed fixed it. If you still have problems, post a question on the discussion boards. There’s bound to be someone who has seen that problem and knows how to fix it!

Spend some time finding your way about the N8VEM CP/M operating system. With the system build I selected, Drive A: is a RAM drive, which means it can store files (up to 445KB) which will stay there until you power off. Drive B: contains some useful utilities like FORMAT.COM, which formats the A: drive initially, and XMODEM.COM which is used to transfer files to the N8VEM via the terminal program on your host computer. Drive F: contains some fun programs like Zork 1, 2, and 3. It also has Wordstar (one of the first successful word processors), Microsoft BASIC, and a bunch of other utilities as well (like PIP.COM, which is used to copy files).

So hopefully your machine is working! If it is, you can start thinking about what other peripheral boards you would like to add to it to really make it interesting. My favorites are the PropIO board (a board that allows you to use a keyboard, VGA monitor, and a MicroSD card for mass storage), the Bus Monitor (which shows you the data traveling back and forth on your bus as you use your N8VEM, which is really cool to look at), and the Sprite-Color-Graphics board (which provides color graphics, stereo sound, and joystick support so you can play some games). There are also boards that allow you to add a hard drive/CF card, add a SD card, add a floppy drive, add a composite monitor, etc. And when all that’s done, you can work on building a computer case to house your masterpiece!

I hope this guide has helped. Enjoy!

Lok.
### Appendix A – N8VEM SBC v2 Parts List

<table>
<thead>
<tr>
<th>Build</th>
<th>Board Ref</th>
<th>Item Description</th>
<th>Jameco Part #</th>
<th>Digikey Part #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 N8VEM</td>
<td>BT1</td>
<td>BATTERY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C1</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C2</td>
<td>10 uF</td>
<td>C1V5</td>
<td>P13478-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C3</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C4</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C5</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C6</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C7</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C8</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C9</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C10</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C11</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C12</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C13</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C14</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C15</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C16</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C17</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C18</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C19</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C20</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C21</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C22</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C23</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C24</td>
<td>1.0 uF</td>
<td>C1V5</td>
<td>545609</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C25</td>
<td>1.0 uF</td>
<td>C1V5</td>
<td>545609</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C26</td>
<td>1.0 uF</td>
<td>C1V5</td>
<td>545609</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C27</td>
<td>1.0 uF</td>
<td>C1V5</td>
<td>545609</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C28</td>
<td>1.0 uF</td>
<td>C1V5</td>
<td>545609</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C29</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C30</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C31</td>
<td>22 uF</td>
<td>C1V5</td>
<td>1946308</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>C32</td>
<td>0.1 uF</td>
<td>C2</td>
<td>25523</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>D1</td>
<td>LED</td>
<td>LEDV</td>
<td>253833</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>D1</td>
<td>JUMPER (LED)</td>
<td></td>
<td>3M9447-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>D2</td>
<td>DIODE</td>
<td></td>
<td>1N4148FS-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>JP1</td>
<td>JUMPER</td>
<td></td>
<td>3M9447-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>JP2</td>
<td>JUMPER</td>
<td></td>
<td>3M9447-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K1</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K2</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K3</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K4</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K5</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K6</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K7</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K8</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K9</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K10</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K11</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>K12</td>
<td>CONN_3</td>
<td></td>
<td>3M9448-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>P1</td>
<td>PARALLEL</td>
<td>PIN_array_13x2</td>
<td>609-3240-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>P2</td>
<td>POWER</td>
<td></td>
<td>109146</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>P3</td>
<td>SERIAL</td>
<td>PIN_ARRAY_5x2</td>
<td>67821</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>P4</td>
<td>CLK_CPU</td>
<td>8dip300 - 4 MHz</td>
<td>XC259-ND</td>
</tr>
<tr>
<td>Build</td>
<td>Board Ref</td>
<td>Item Description</td>
<td>Jameco Part #</td>
<td>Digikey Part #</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>------------------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>P5</td>
<td>CLK_UART</td>
<td>8dip300 - 1.8432 MHz</td>
<td>XC231-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>P6</td>
<td>RESET</td>
<td>SIL-2</td>
<td>108338</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>R1</td>
<td>10</td>
<td>R3</td>
<td>690380</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>R2</td>
<td>10K</td>
<td>R3</td>
<td>691104</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>R9</td>
<td>100</td>
<td>R3 (x10)</td>
<td>100QBK-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>R11</td>
<td>10K</td>
<td>R3</td>
<td>691104</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>R12</td>
<td>4700</td>
<td>R3</td>
<td>691024</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>RR1</td>
<td>1000</td>
<td>SIL-6</td>
<td>4306R-1-102LF-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>RR2</td>
<td>4700</td>
<td>SIL-6</td>
<td>4306R-1-472LF-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>SW1</td>
<td>SW_PUSH</td>
<td>SW_PUSH_SMALL</td>
<td>1586074</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U1</td>
<td>16550</td>
<td>40dip600</td>
<td>288809</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U2</td>
<td>27C080</td>
<td>32dip600</td>
<td>156275</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U3</td>
<td>74LS08</td>
<td>14dip300</td>
<td>46375</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U4</td>
<td>74LS10</td>
<td>14dip300</td>
<td>46404</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U5</td>
<td>74LS125</td>
<td>14dip300</td>
<td>46501</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U6</td>
<td>74LS139</td>
<td>16dip300</td>
<td>46623</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U7</td>
<td>74LS14</td>
<td>14dip300</td>
<td>46640</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U8</td>
<td>74LS245</td>
<td>20dip300</td>
<td>47212</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U9</td>
<td>74LS245</td>
<td>20dip300</td>
<td>47212</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U10</td>
<td>74LS243</td>
<td>14dip300</td>
<td>eBay</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U11</td>
<td>74LS245</td>
<td>20dip300</td>
<td>47212</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U12</td>
<td>74LS174</td>
<td>16dip300</td>
<td>296-3664-5-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U13</td>
<td>74LS174</td>
<td>16dip300</td>
<td>296-3664-5-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U14</td>
<td>74LS32</td>
<td>14dip300</td>
<td>47466</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U15</td>
<td>74LS32</td>
<td>14dip300</td>
<td>47466</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U16</td>
<td>74LS32</td>
<td>14dip300</td>
<td>47466</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U17</td>
<td>74LS244</td>
<td>20dip300</td>
<td>296-1653-5-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U18</td>
<td>74LS174</td>
<td>16dip300</td>
<td>296-3664-5-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U19</td>
<td>8255</td>
<td>40dip600</td>
<td>52417</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U20</td>
<td>C96ABC</td>
<td>C96ABC MC</td>
<td>859561</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U21</td>
<td>DS1302</td>
<td>8dip300</td>
<td>176778</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U22</td>
<td>MAX232</td>
<td>16dip300</td>
<td>318959</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U23</td>
<td>SRAM_512KO</td>
<td>32dip600</td>
<td>1927617</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U24</td>
<td>Z80-CPU</td>
<td>40dip600</td>
<td>35705</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U25</td>
<td>DS1210</td>
<td>8dip300</td>
<td>DS1210+-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>U26</td>
<td>74LS06N</td>
<td>14dip300</td>
<td>296-15069-5-ND</td>
</tr>
<tr>
<td>1 N8VEM</td>
<td>X1</td>
<td>CRystal</td>
<td>SIL-2 - 32.768 KHz</td>
<td>325244</td>
</tr>
</tbody>
</table>
## Appendix B – N8VEM SBC v2 Jumper Settings

<table>
<thead>
<tr>
<th>Board Reference</th>
<th>Jumper Description</th>
<th>Jumper Setting</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1-2</td>
<td>2-3</td>
</tr>
<tr>
<td>JP1</td>
<td>battery backup</td>
<td>x</td>
<td>n/a</td>
</tr>
<tr>
<td>JP2</td>
<td>One bit input port</td>
<td>x</td>
<td>n/a</td>
</tr>
<tr>
<td>K1</td>
<td>U2 EPROM pins (32-pin or 28-pin)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td>UART side hardware handshaking (DSR or CTS)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td>UART side hardware handshaking (DTR or RTS)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K4</td>
<td>serial side hardware handshaking (DSR or CTS)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K5</td>
<td>serial side hardware handshaking (DTR or RTS)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K6</td>
<td>U2 chip type (27C080 EPROM or 29C040 flash)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K7</td>
<td>U23 SRAM (512K or 128K)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K8</td>
<td>U2 chip type (27C080 EPROM or 29C040 flash)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K9</td>
<td>Parallel Port power status (pin 25)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K10</td>
<td>MCPL (Memory Page Config Latched)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K11</td>
<td>MCPL (Memory Page Config Latched)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K12</td>
<td>Bus Interrupt (pin A23)</td>
<td>x *</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* = default setting
Appendix C – Willem EPROM Programmer Configuration

1. Use **Willem Software version 0.97ja** (version 3) – this is because there have been some internet documented cases of later software versions causing some unusual problems with certain chips (including 27C080/27C801 which is used on the N8VEM). It should be on the software CD your programmer came with. If not, there is a copy on the N8VEM wiki page. Load up the software on your host PC and it should look like this:

![Willem Eprom (0.97ja) - Rommage_Feb28_2009TOOLS_Zork.bin](image)

2. Plug in your Willem EPROM programmer to your **parallel cable** and plug your parallel cable into your PC. Also plug in your **power source** (either a USB cable or an AC Adapter). Some of the LEDs on the Willem EPROM programmer should light up.
3. Set up the jumpers on your Willem EPROM programmer board as follows:

- Move jumpers in J4 to the right if using version 3 of software (which is v0.97ja)
- Move jumper normally in the lower position of J3 here...
- ...over to this jumper block making sure to jumper P1 and A19 together.
I set this jumper (J10) to "VDC" because I was using an AC adapter instead of the USB.
Check the Willem software to tell you how to set these DIP switches:

4. In the Willem software, select “Help” → “Test hardware”. At the bottom of the window, there should be a message “Hardware present” which means it detected your hardware properly and you are ready to program an EPROM!

If the message says “Hardware Error”, check all your cables and settings and repeat.