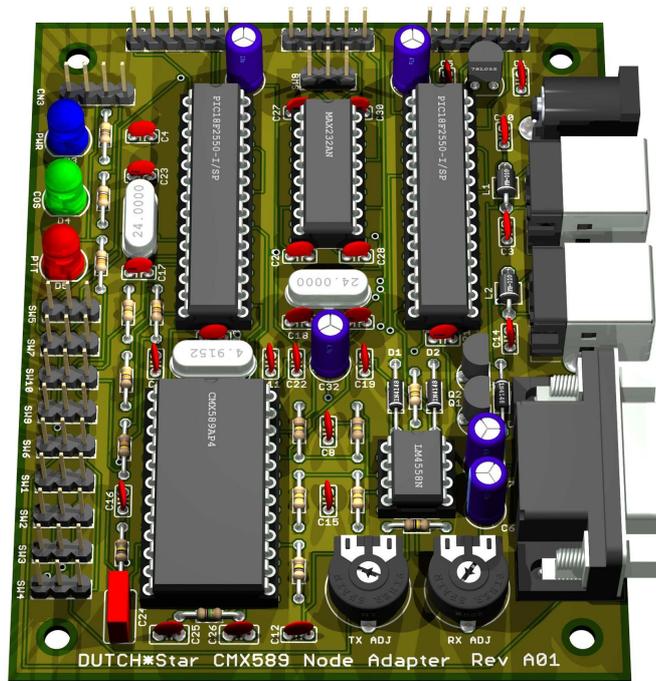


# *DUTCH\*Star Node Adapter*



## **Construction and Reference Manual**

Version 1.10 for PCB revisions A and A01

Fred N. van Kempen PA4YBR  
The DUTCH\*Star Project, <info@dutch-star.nl>

Website: <http://www.dutch-star.nl/>



MicroWalt Corporation  
PO BOX 8, 1400 AA BUSSUM  
THE NETHERLANDS

PH : +31 35 7503090  
FAX: +31 35 7503091

Internet: <http://www.microwalt.nl/>  
E-Mail: [info@microwalt.nl](mailto:info@microwalt.nl)

Copyright 2009-2010 MicroWalt Corporation DUTCH\*Star.

All Rights Reserved.

Last Revision: 02/21/2010

# TABLE OF CONTENTS

TABLE OF CONTENTS .....	3
1. Introduction.....	5
2. Construction Notes.....	7
2.1 GENERAL.....	7
2.2 BOARD CORRECTIONS.....	7
2.3 ASSEMBLY.....	7
3. Hardware Setup .....	9
3.1 Connectors and Cabling .....	9
3.2 Jumpers.....	10
4. Software Setup and Configuration.....	13
4.1 DRIVERS .....	13
4.2 TOOLS - Configuration .....	15
4.3 TOOLS - Testing .....	18
4.4 Setting Up DVAR HotSpot .....	20
5. Updating the Firmware .....	21
Appendix A: PCB Parts Placement.....	23
Appendix B: Parts List .....	25
B.1 Parts for Rev.A .....	25
Appendix C: Corrections.....	27
C.1 Corrections for Rev. A boards .....	27



# 1. Introduction

Welcome to the exciting world of Digital Communications for Amateur Radio!

**D-STAR** (Digital Smart Technologies for Amateur Radio) is a communications system for Amateur Radio operators where the voice audio is transmitted as a digital signal rather than as analog audio.

Operators use standard VHF/UHF/SHF radio transceivers (handheld units, mobile radios, and so on), and their voice audio is converted to a digital data stream by those transceivers. The data is then transmitted on-air with the use of a GMSK-based modem that is (of course) built in to them. Most of these transceivers can also handle regular, analog (FM) based audio.

Although communications between these transceivers (in Simplex mode) to provide noise-free digital communications between the operators is of course possible, the real power of this mode of operation is unleashed by using a repeater system. The repeater extends the range of these transceivers considerably, but, being a digital (data-based) system, it is extremely easy to connect multiple repeaters into a network of repeaters, using RF links, the Internet, or both.

One of the most criticised aspects of the D-STAR system, however, is that there is only one global manufacturer of radio equipment for it, that being Icom. So, until recently, getting to know D-STAR meant buying some equipment (transceivers, repeater equipment, and so forth) from Icom, set things up, and play- all at considerable cost.

We are now seeing a series of software projects and hardware developments that bring the fun (and power) of D-STAR to the home brewer, however. Software projects allow you to connect your PC to the D-STAR network, and listen to the conversations that take place there. Various hardware projects have resulted in products that you can build (or assemble) yourself, and with them you can convert your PC into an almost-real D-STAR repeater, for example.

The DUTCH\*Star Node Adapter is such a product. Once you finish building it (either using your own parts, or by assembling the 'parts kit' you purchased from us) you will be able to connect it to a PC, configure it, and then set it up as a full-featured repeater for D-STAR users in your area, costing a *lot* less than a comparable setup using commercial products, such as those offered by Icom!

Now... get your tools out, and have fun building!



## 2. Construction Notes

### 2.1 GENERAL

Construction of the DUTCH\*Star Node Adapter PCB Kit is easy, but, like all electronics assembly tasks, it does require some attention. We won't bother you with the usual warnings about soldering irons being **hot** and **lead** being **poisonous if swallowed**, as we all know that, right?

The PCB has a clear-white printing showing where the various parts go, what their names are, and, where applicable, what their polarity or orientation is. An enlarged copy of that can be found at the end of this document in Appendix A: PCB Parts Placement, by the way.

In general, all components should be placed in a logical order. It looks better, and, in some cases, gives better results if the leads of the components are cut 'flush' at the bottom side of the PCB, meaning, cut them off as short as possible. This avoids ugly bumps sticking out. This can be done easily with any good flat-head cutting tool.

### 2.2 BOARD CORRECTIONS

If errors were found after production of the PCB, we list them in this manual, including tips on how to fix things on your board. See APPENDIX B: Corrections for notes and instructions specific to the revision of the board you have.

Please take note of these issues **BEFORE** doing any soldering; if you have to "fix" these errors after completing assembly of the board, you may damage the board beyond repairs!

You can also check the website if any new updates were issued since the writing of this revision of the manual.

### 2.3 ASSEMBLY

Actual assembly of the kit is best done in a series of steps, the first step being inspection of all the parts in the kit. Spread out the parts on a flat, clean surface (preferably a white-coloured kitchen table, although family members do not always appreciate such use of house tables) and sort them into the various groups of components (resistors, capacitors and so on.)

Where applicable, sort them again based on value.

- Work from small (and flat) components to big and tall ones: first all the resistors, then the diodes, ferrite bead and then the small 2.5mm capacitors.

Inspect the PCB for soldering errors, and clean up where needed.

- Now place the crystals (be careful not to drop them, many crystals can be damaged if they are subjected to such a shock, it will cause their frequencies to be changed!), the transistors and the 78L05 voltage regulator - this one looks like one of the transistors, but isn't one of them.

Re-check the PCB, and measure the resistance between the GND and VCC lines, this should be high impedance.

- Place the IC sockets, the elco's and the LED's.

If available, you could now connect a 5V DC source to the board, and the blue (PWR) LED should light up. You could also measure for correct voltages on the various IC pins to make sure "all is well".

- Place the various (DB9, USB etc) connectors.

You can now connect the PCB to a USB port on a PC, and... nothing should happen! However, you can now measure a voltage between 4.6 and 5.2V on pin1 of SW7, the DC source selector; this is the USB bus power.

- Lastly, cut the supplied pin-header strip to the sizes needed (see the list in **Appendix A: Parts List**) and sort them based on the number of pins. Solder them onto the PCB, and beware that many people find it hard to make them stand up straight. It can be done, really, but it does take some concentration to get them lined up nicely.
- Place the supplied jumpers on them in the 1-2 positions. The **RED jumper** is intended for the **SW7** (Power Select) header. Also, use a small screwdriver to tune all trimpots in their middle positions. We will re-tune them later.

After all components have been soldered onto the PCB, perform a careful inspection of the PCB, especially its soldering side. It often is a good idea to clean the PCB with a flux-removing substance such as Deflux, or simply using alcohol. Flux can, in some cases, cause really-hard-to-find problems, so the more you can remove, the better the results will be.

Connect the PCB to the USB port, and, again, measure voltages on the IC socket pins to make sure you don't have any surprises there.

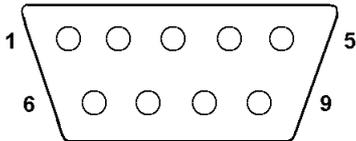
When all seems good, insert the IC's into their sockets and hope for the best while powering up the board again...

## 3. Hardware Setup

### 3.1 Connectors and Cabling

To make use of the board, you must have available one (or two, see below) radio's capable of doing high-speed data. This is usually the case with modern transceivers- just check whether it supports 'data' (also known as 'packet') modes at 9600bps. Quite often, these radios have a special connector (port) for this mode (standard seems to be a 6-pin Mini-DIN plug, and also 13-pin DIN plugs are seen.) The radio usually can be configured to enable these ports for use through their configuration software or menu system.

Depending on the radio used, you should now acquire or make an interconnecting cable between the board and the radio(s). The pin-out of the D-SUB connector on the board is as follows:

1	Radio MIC (audio input)	
2	Radio COS (squelch)	
3	Radio SPK (audio output)	
4	Radio PTT	
5	GND	
6		
7		
8		
9	Power Input (optional, 20V max)	

This is the connector when seen from the outside, so, when looking at the connector's pins. It is also the soldering side of the female connector that mates with it, the one which you will use to create a cable for your radio.

For a **single radio**, this is a straightforward task. In case of doubt, leave Pin9 (Power Input) unconnected and only connect it if you are sure the radio can supply the required power and its voltage will not exceed our limit.

If your configuration is set up for **duplex mode**, you will either have to connect it to a single radio which is capable of handling full-duplex communications (special repeater-mode radio's) or you must connect two separate radio's, one for RX (receive), and the other for TX (transmit.) Dual-radio operation will often require some sort of 'splitter' cable, where the single (female) DB9 end connects to our board, and the other end of the splitter has two (male) DB9's with each carrying only half the signals; the **RX connector has (2, 3, 5 and optionally 9)**, and the **TX connector then has (1, 4 and 5.)**

It is recommended to **only "draw" power (if any) from the RX radio** in a dual-radio configuration.

## 3.2 Jumpers

Assuming you got the cable(s) all set to go, we must now check out the various switches ("jumpers") on the board. There are many, because the board can operate in "Version 4" or "Version 5" mode, and it can also operate in "USB/HotSpot" or "Standalone" mode.

The switches are:

Switch	1-2 function	2-3 function
<b>SW1</b>	COS signal fed into transistor for positive COS signals.	COS signal fed directly to PIC (negative COS signal.)
<b>SW2</b>	COS signal fed directly to PIC (negative COS signal.)	COS signal taken from transistor for positive COS signals.
<b>SW3</b>	Select MAIN CPU for PTT control (USB/HotSpot mode)	Select OPTIONAL CPU for PTT control (Standalone mode)
<b>SW4</b>	Select MAIN CPU for TXDATA (USB/HotSpot mode)	Select OPTIONAL CPU for TXDATA (Standalone mode)
<b>SW5</b>	Select USB/HotSpot mode.	Select Standalone mode.
<b>SW6</b>	Set for Firmware V.4	Set for Firmware V.5
<b>SW7</b>	Operating power is taken from the USB bus.	Operating power is taken from the radio connector or DC connector, and the USB bus power is disconnected.
<b>SW8</b>	Select TTL for SERIAL	Select RS232 for SERIAL
<b>SW9</b>	Set for Firmware V.5	Set for Firmware V.4
<b>SW10</b>	Set for Firmware V.5	Set for Firmware V.4
<b>JP1</b>	Select PROG/RESET mode.	

The **COS signal** of the radio can be negative (0V indicates a valid signal) or positive (a positive voltage indicates a valid signal), and the SW1 and SW2 switches must be set accordingly. Most radios provide a positive signal, which means **SW1** in the 1-2 position, and **SW2** in the 2-3 position.

The board can run in **single-CPU** ("USB/HotSpot mode") or it can run **completely standalone**, using a second CPU. **SW5** tells the MAIN CPU which mode to set up, and then **SW3 (PTT)** and **SW4 (TXDATA)** select which of the two CPU's handles the TX (sending) part of the system.

The **PROG/RESET** jumper is only needed when using DUTCH\*Star HSA firmware. It forces a RESET of the board, and when still inserted when the board is reset (or powered up), it will make it enter the PROGRAM mode of the USB Boot Loader, which allows you to (forcibly) update the firmware of the board.

**Operating power** can be taken from either the USB bus, or from the radio itself. In USB/HotSpot mode, we will normally use the USB bus power for the board.

When running in Standalone mode, it may be useful to select power to be drawn from the DC source, which then is either the DC jack (standard 2.1mm) or power is drawn from the connected radio.

**WARNING: PIN IS NEGATIVE !**

### Which Version?

The remaining switches are used to set up the board for use with either *Version 4* or *Version 5* releases of the original firmware written by Satoshi Yasuda JK1ZRW. The "Version 5" release was designed for a new PCB, and this new PCB used a slightly different allocation of I/O signals.

**Although the DUTCH\*Star HSA firmware supports both modes of operation, it defaults to "Version 4" mode, so assume "Version 4" for this firmware until you feel comfortable switching it to another mode !**

If you wish to run **Version 4 firmware**, set **SW9** and **SW10** to 2-3, and **SW6** to 1-2. For **Version 5 firmware**, set **SW9** and **SW10** to 1-2, and **SW6** to 2-3.

An interesting feature of the "Version 5" firmware is the **support of the RSSI signal** generated by the GMSK modem chip. This signal is tempered a little by feeding it past a big(-ish) capacitor, and then the resulting (tempered) voltage is presented to an analog input of the CPU, where it can be processed and passed on to the application code.

The **SERIAL port** (used for processing D-STAR's "slow data" traffic) can be operated using either TTL (**SW8** set to 1-2) or RS232 (**SW8** set to 2-3) levels. **Please note that this port is only supported in the "Version 5" firmware !**

You can check the schematics to see what these switches actually do, although for most people, this will be irrelevant to know.



## 4. Software Setup and Configuration

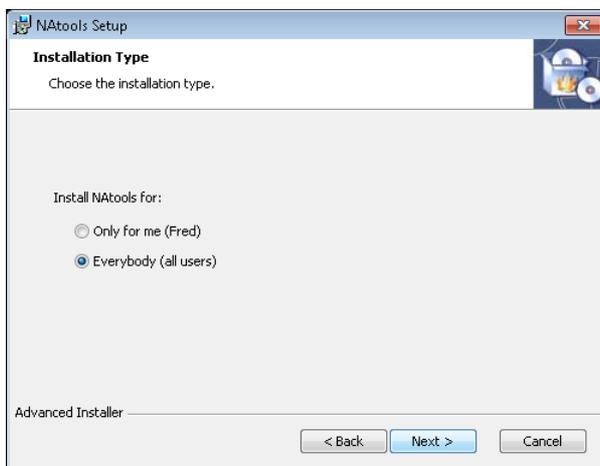
The Node Adapter board can be used in Standalone mode, as a self-contained mini-repeater, or it can be set up for use with a PC running the 'DVAR HotSpot' application written by Mark McGregor KB9KHM (see our website under Links.) In this latter mode, the board acts exactly the same as the Mini-HotSpot PCB.

### 4.1 DRIVERS

The first step is to **install the drivers** for your operating system. The drivers have now been merged with a new set of test-, play- and configuration tools into a package called the NAtools (Node Aadapter tools.) This package can be installed on any of the recent versions of Windows, and a version for Linux-based systems is also available. The most recent version of these packages can be downloaded from our website, under the **Software** menu option.

As an example, we'll show the installation of the package for a **64-bit version of Windows 7**. It is assumed that you already downloaded the installation package from the website, and placed it somewhere on the PC where it can be executed, which will result in a screen similar to the one on the right →

Click on **Next**, and select whether this package is only for the current user, or for all users of that PC.



Normally, we will log on as Administrator (or as a user with Administrator rights) so we can install the drivers, and then allow all other users of the PC to actually use the software, so the defaults as selected are good to go.

Remember that most Windows-based PCs are set up such that you cannot (properly) install software packages and/or drivers if you do not have Administrator-level rights on your account!

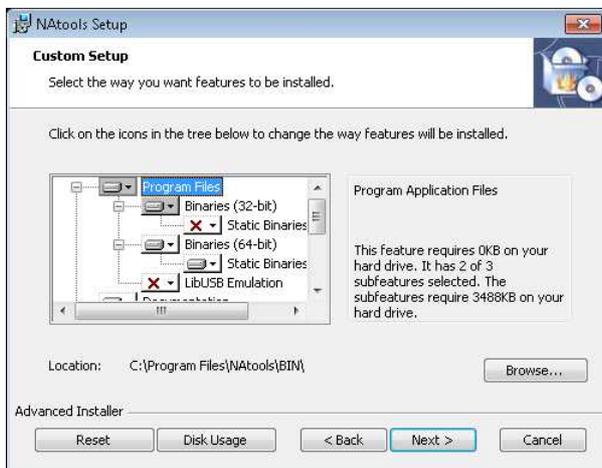
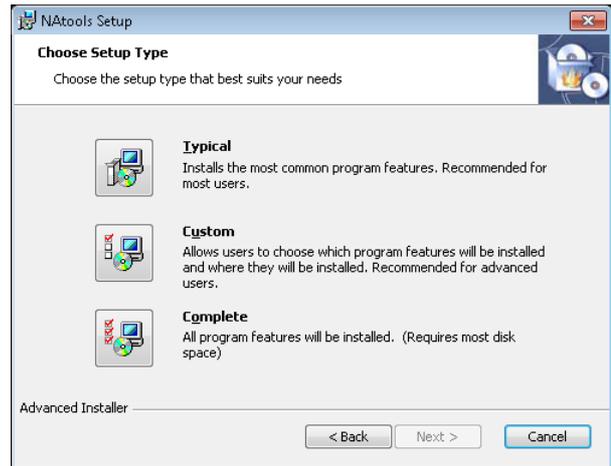
Click on **Next** to continue (and don't forget to also go to the next page..)

Ah, you're back!

Since this is a manual, and we're expected to show you what you can (and cannot) do, we will do a **Custom** type of installation, which shows all possible options.

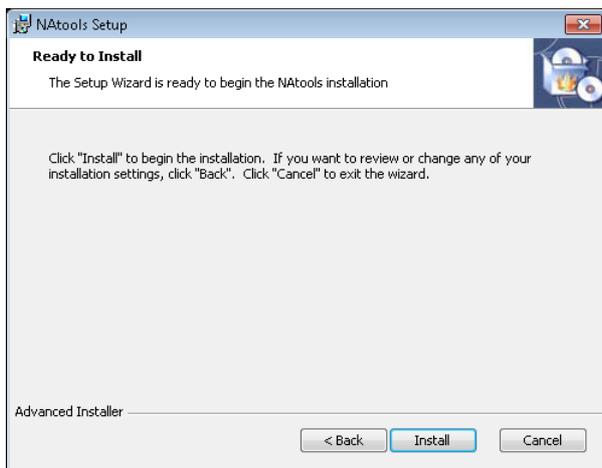
Most of the time, you will be doing a **Typical** or **Complete** installation, however, as these are created to accommodate most users.

The NAtools package consists of a number of modules, most of which are optional, and only a few are required. Since we are installing on a 64-bit Windows system, we also get the option to install 32-bit versions of the various programs. This can be useful at times, so we will leave that option to you.



For normal operation with the DVAR HotSpot application, we just need the drivers and the test/configuration tools, meaning the **"Binaries (64-bit)", "Documentation", "Audio Files" and "USB drivers"** modules. You could also select to install the Static versions of those programs. Static programs do not require any additional DLL files; these are compiled into those programs. This is sometimes easier to handle when managing remote PCs, for example.

If you intend to also do some D-STAR programming, you may wish to select the **"Development Files"** module in the package; these can be used to write programs for the D-STAR system.



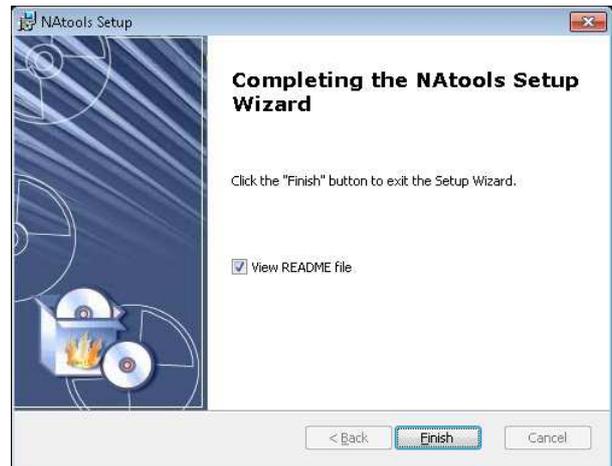
The **"LibUSB Emulation Support"** module should be installed if you want to run older programs with your board which are still based on that driver. These are being phased out, but until they are completely gone, we offer a compatible interface that runs on the new drivers, but which makes the programs believe they are (still) talking to a LibUSB driver.

Examples of such programs include older versions of the DVAR HotSpot application, and the various tools written by Satoshi Yasuda.

After you have finished selecting the desired modules, click on **Next** to continue. When ready, click on **Install** to begin installing the selected modules onto your system!

You may receive warnings about drivers not being signed- these can safely be ignored, as the drivers are indeed safe. If all goes well, you should end up with a screen saying that the installation was successful.

If you wish, you can have it display the **README** file for you, which contain a brief summary of what has been installed, and a summary of changes since the previous version(s) of the package.



After clicking on **Finish**, you may also receive a message stating that you can now plug in any devices you may wish to install. That's actually a good idea.. connect your Node Adapter board to any of the available USB sockets now.

If you are using the newer Windows systems, installation of the drivers will be automatic. The Windows XP and Windows 2003 systems will display the well-known screens asking to select automatic mode or manual install - just select **Automatic**, as the drivers have already been installed now, so the installation procedure will find them.

After successful installation, you should be able to see the board listed as a device under the 'GMSK Devices' group in the 'Device Manager' screen of the System control panel.

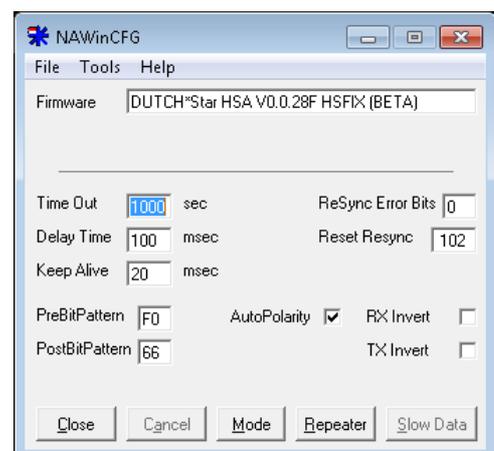
## 4.2 TOOLS - Configuration

When the drivers have been installed, it is time to fire up the **NAWinCFG** program, and if all is well, you should get a display such as:

The program can also tell you that it was unable to locate a board. This can be caused by many factors, but most often they are one of:

- driver installation was not (really) successful;
- firmware was not yet installed in the board;
- bad USB ID was programmed in the board

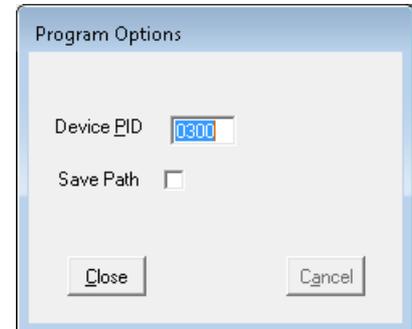
All of these are usually fixable, so don't despair. If you get any error messages when starting the program, it is most likely an installation problem, maybe because you tried to install the package as a regular user without Administrative rights on the system.



Log off as that user, log in as Administrator, and repeat the installation, then try again.

If you are certain that the driver installation is fine, it may be that the PIC microprocessor chip was not programmed with firmware yet. If you suspect this is the case, you **must obtain firmware**, and program that into the board using the **Update Firmware** procedure from the **Tools** menu. Detailed instructions for can be found in a separate chapter in this manual.

One other issue that might cause the problem is that the board may have been programmed with a **non-standard USB ID** number. Each USB device has its own device ID, so the drivers can tell them apart. Guess what happens if the driver (and our software) is looking for one ID, but the board is actually using another one? Good guess. You can see (and change) the ID used by the software in the Device PID setting in the Options item in the Tools menu.

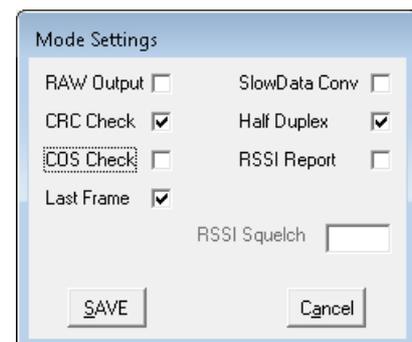


The **default setting** (expected by most applications and tools) is **0300** (which is a hexadecimal number, the decimal representation is 768), but we have also been using an alternate setting of 0302, for example.

Going back to those main settings: most can (and should) be ignored, and left as they are.

We may have to fiddle with the **AutoPolarity** and/or the **Invert** flags, however, as their settings depend on the radio(s) we are using. A good starting point is to enable the AutoPolarity flag, and clear the Invert flags. If it later turns out that the signal cannot be received, just test with various settings of these flags- usually, one works! Once you have a working configuration, it is probably a good idea to turn off the AutoPolarity flag, to avoid errors later.

The Mode menu screen also has some flags that can be useful, most notably the "COS Check" (default ON) and "Half Duplex" (default OFF) settings.



The DUTCH\*Star HSA firmware supports many methods for detecting when a valid signal is present on the receiver. Depending on which version of the board is being used, we can separate these into the "**Hardware COS**" and "**Software COS**" groups, really. By default, "Hardware COS" is enabled, indicated by the "COS Check" flag. In this mode, the processor constantly monitors the state of the internal COS line, and, if that becomes active, starts up the receiver. If the COS signal drops, the processor will terminate the receiver.

The actual source of this COS signal line will be determined by the various jumper settings on the board (SW1 and SW3), but the source is always the hardware, hence the name.

If this flag is cleared, the processor will simply always enable the receiver, and scan the incoming data until it decides that it has found the beginning of a valid transmission.

This "**COS in software**" mode is easily recognized by the fact that when this mode is active, you'll see the COS LED (green) be lit dimly. It is turned ON and OFF at the receiver clock rate, so, at about 4800Hz, hence the LED showing as dimly lit.

**In many, many cases, this Soft-COS mode provides the best results, and we recommend using it if the situation permits.**

For systems that operate in **Simplex** (half-duplex) mode it may be required to enable the Half-Duplex flag, which tells the board to disable the receiver when it is sending data to the radio. It turns out that some radios leak a portion of the audio being transmitted back to their receiver side, which in our case would then be picked up by the receiver side of the board... and we'll hear our own modulation, or at least parts of it!

If your board can handle **RSSI reporting** (meaning, you have a board of at least revision B, or a board made by Satoshi Yasuda with version greater than 2.05, and so on) you can enable this feature by checking the flag. RSSI reporting simply means, that the GMSK modem chip will report the current quality of the signal back to the processor, which in turn can report this back to the PC. The DVAR HotSpot application supports this feature.

The remaining menu's only contain settings useful when the board is used together with tools other than the HotSpot application, and also when configured on a Node Adapter board which has more features, including a standalone mode of operation, where it can act as a repeater without a PC having to be attached. For now, we will ignore those screens.

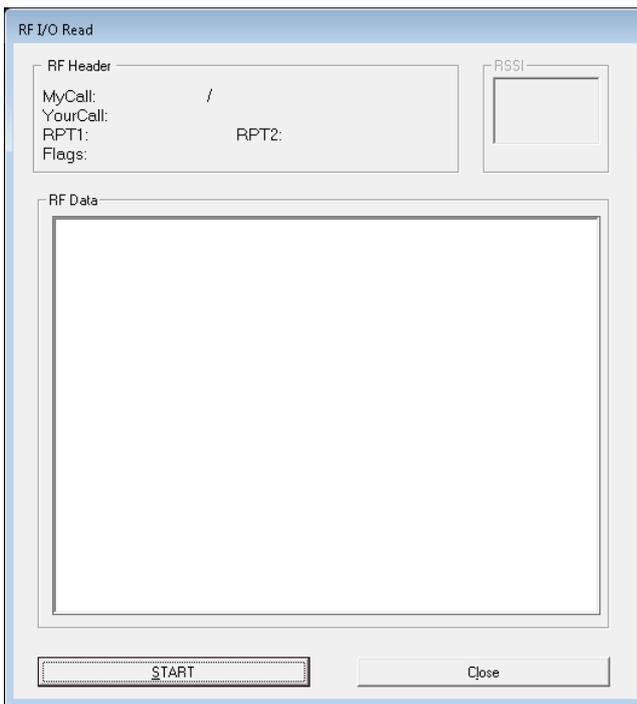
### 4.3 TOOLS - Testing

After configuring the board it is time to test it, and, if needed, adjust its audio levels.

Start up the **NAWinTEST** program, which should just show you what version of firmware it detects on the board, and then waits for a transmission to arrive:



The first thing we must do is to make sure the receiver works properly. Click on the **RF Read** button, and you'll see a screen similar to this:



Click on the **START** button, and key up a D-STAR compatible system and you should see the relevant information being displayed by the program.

If it does not, make sure that the board's COS (GREEN) LED lights up when you are keying up; it should. If the LED does not light up, something is wrong with the COS setup between the radio and the board - refer to the "**Hardware Setup**" chapter for more information on setting the COS jumpers for your radio.

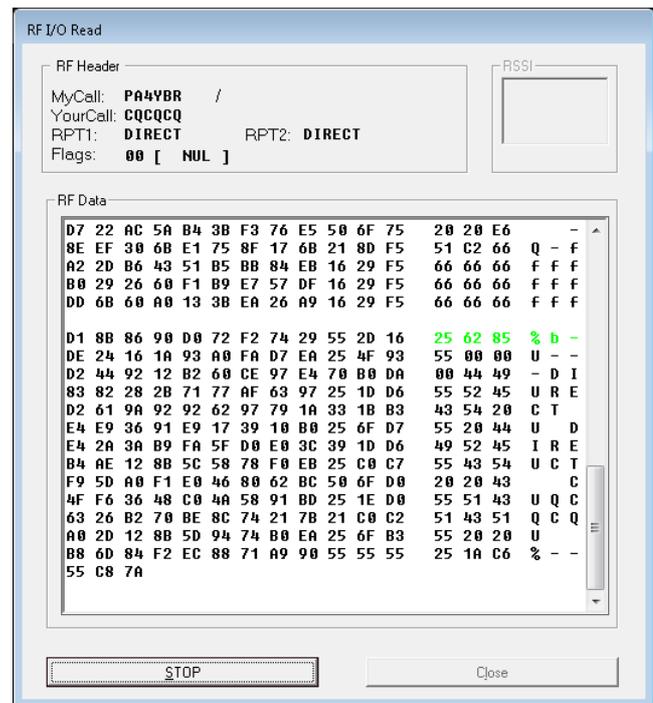
If the LED lights up, but no information is displayed, it may be needed to adjust the trimpots a little bit.

Start out with VR2 set to midlevel, and key up repeatedly until the information shows up, and repeat this while gently adjusting the setting of VR2. If you cannot find a setting that works, you may have to go back to the "**Configuration**" section of this chapter to play with the various Polarity settings, as described there.

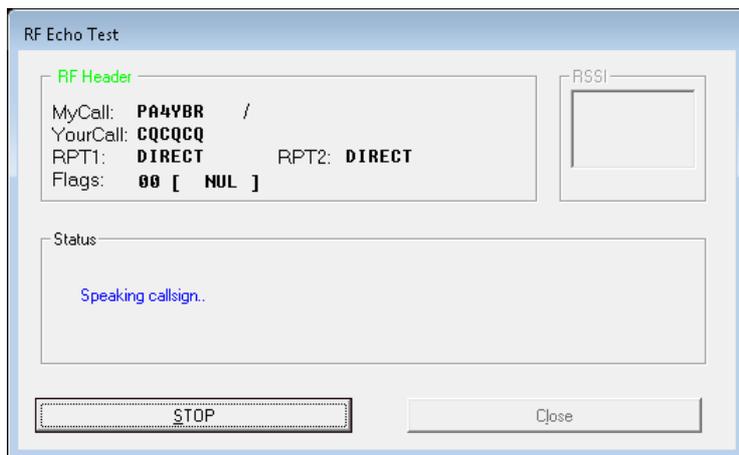
Re-run the VR2 tuning procedure. If you still cannot find a working setting, this would be a good time to get some coffee, relax, and then ask for help on the *GMSK Node Adapter* group on Yahoo!

If the audio levels and COS setting are correctly configured, you should see your call-sign appear in the RF Header section, and you should see streams of hexadecimal data fly by in the Data section of the screen.

A quick check to see if the data is correct is by watching it fly by, and every now and then, you will see a blank line, followed by a line with part of its data being displayed in either green or red. The usual applies: **green is good, red's bad**. Red indicates that the data contains (too many) errors, and this is usually caused by the audio levels not being at the proper levels. Use the VR2 tuning procedure to adjust until you see mostly (if not all) green in this screen.



When the receiver seems to be working well, it is time to also test the transmitter part of the board, and for this, we have the other utility in this program: the Echo Test server. Click on the **Echo Test** button, and you should see this screen:



Click on the **START** button to fire it up, and it should display a message stating that it is waiting for a new transmission.

Key up the radio again, and say something, then un-key. You should see the screen receive your call-sign, and if all was set up right, you should now hear audio coming from your radio, as the Echo Server is spelling out your call-sign, followed by a

message spoken by yours truly, and then followed by a recording of what you just said (hence the name of the program.)

If you can see that the radio is receiving something (and the TX (RED) LED on the board lights up), but you hear no audio at all, you may have to change the setting of the TX Invert flag - see the Configuration section of this chapter on how to change this setting.

If you hear audio, but it's quite garbled, you may have to adjust the audio level from the audio going from the board to the transmitting radio, which is handled by the VR1 trimpot. Adjust it slowly until audio quality is good and constant.

When all is working as expected, you can close the program, and move on to the next level - installing and setting up the PC application you are going to use the board with.

#### 4.4 Setting Up DVAR HotSpot

With all hardware working, you can now proceed to installing the **HotSpot** application on your PC, as per its documentation. If all is well, you will end up with a screen looking similar to this:



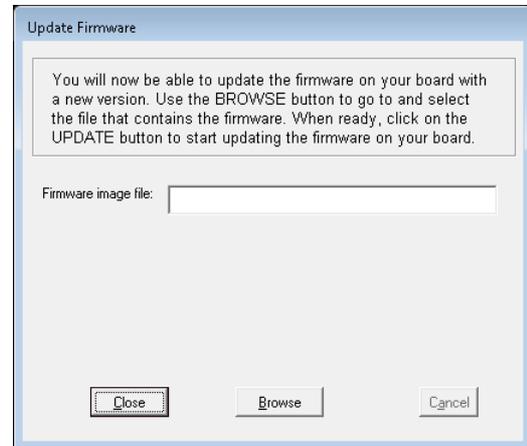
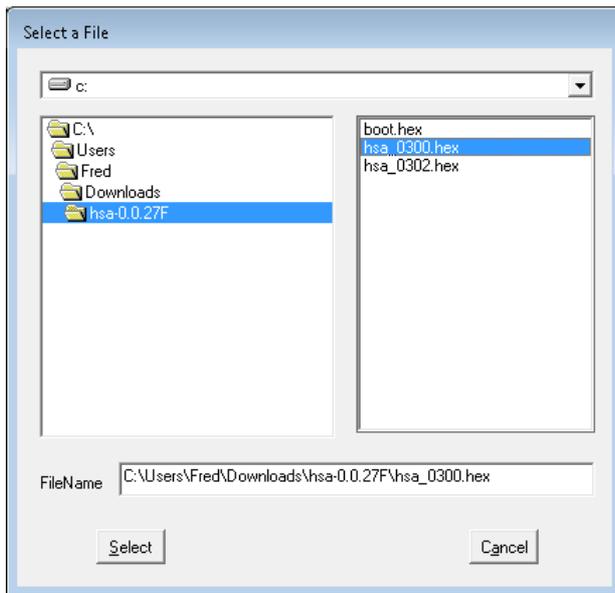
which is the screen of an active HotSpot system in operation.

## 5. Updating the Firmware

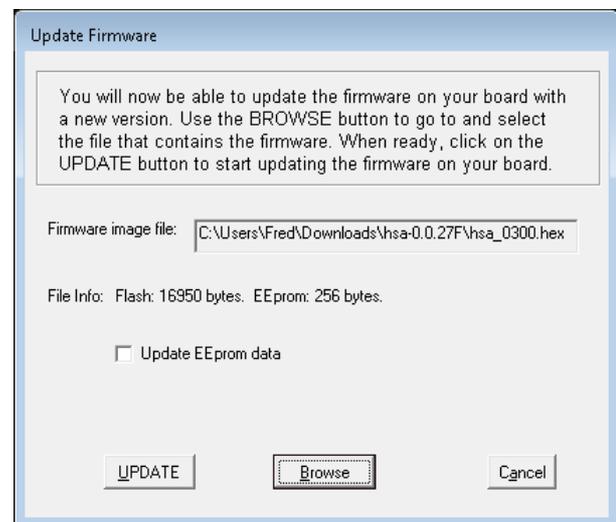
Updating the firmware on your board is fairly simple.

From the NAWinCFG program, select the **Update Firmware** item on the **Tools** menu →

**Browse** to the firmware image file, and select it:



When you have selected the file to be used, click on **Select** to go to the next screen:

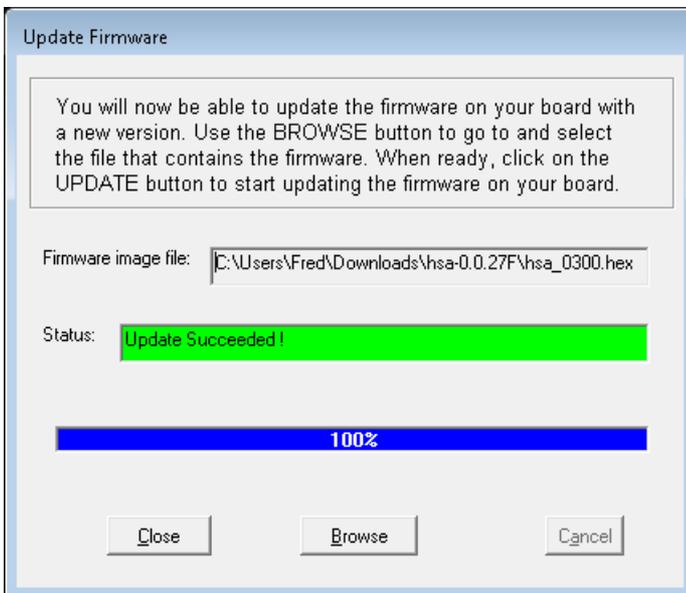
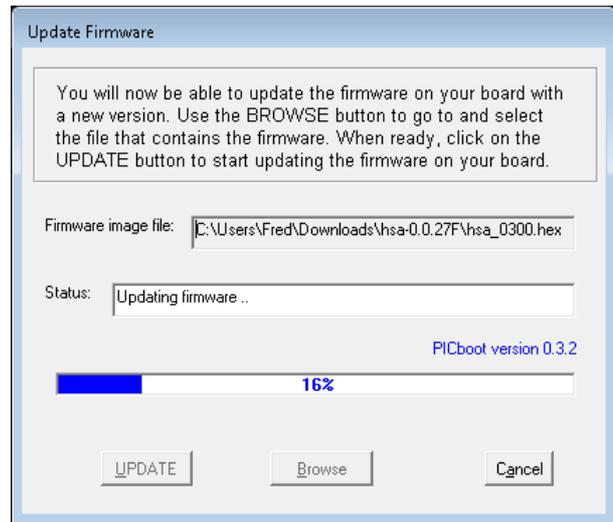


In this screen, some information about the firmware image file shown, and you gave the option of also updating (re-programming) the EEPROM data of the chip, which holds the various configuration settings of the board.

Normally, this only has to be done the very first time you install the firmware (because the entire chip will be empty), or if you suspect that the EEPROM settings have been corrupted.

If a firmware upgrade requires that you re-program these parameters, the README file accompanying that file will explicitly say so !

If you're all set and ready to program the chip, click on **UPDATE** and you'll see the next screen go by: →

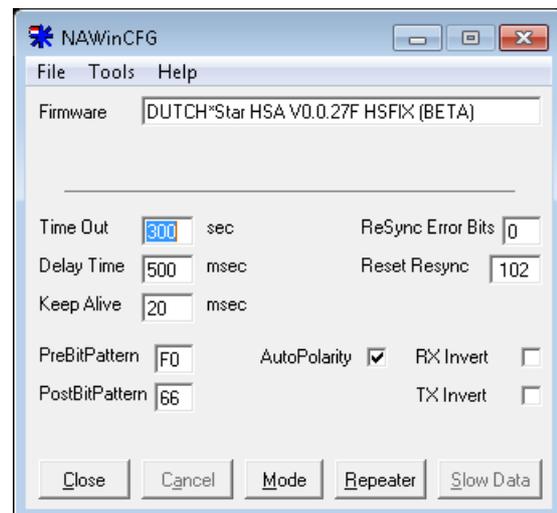


If everything went as planned, you should see a screen similar to the one on the left. ←

The **green** color here is essential: if any error occurred, it will tell you which error, and the color will be **red**.

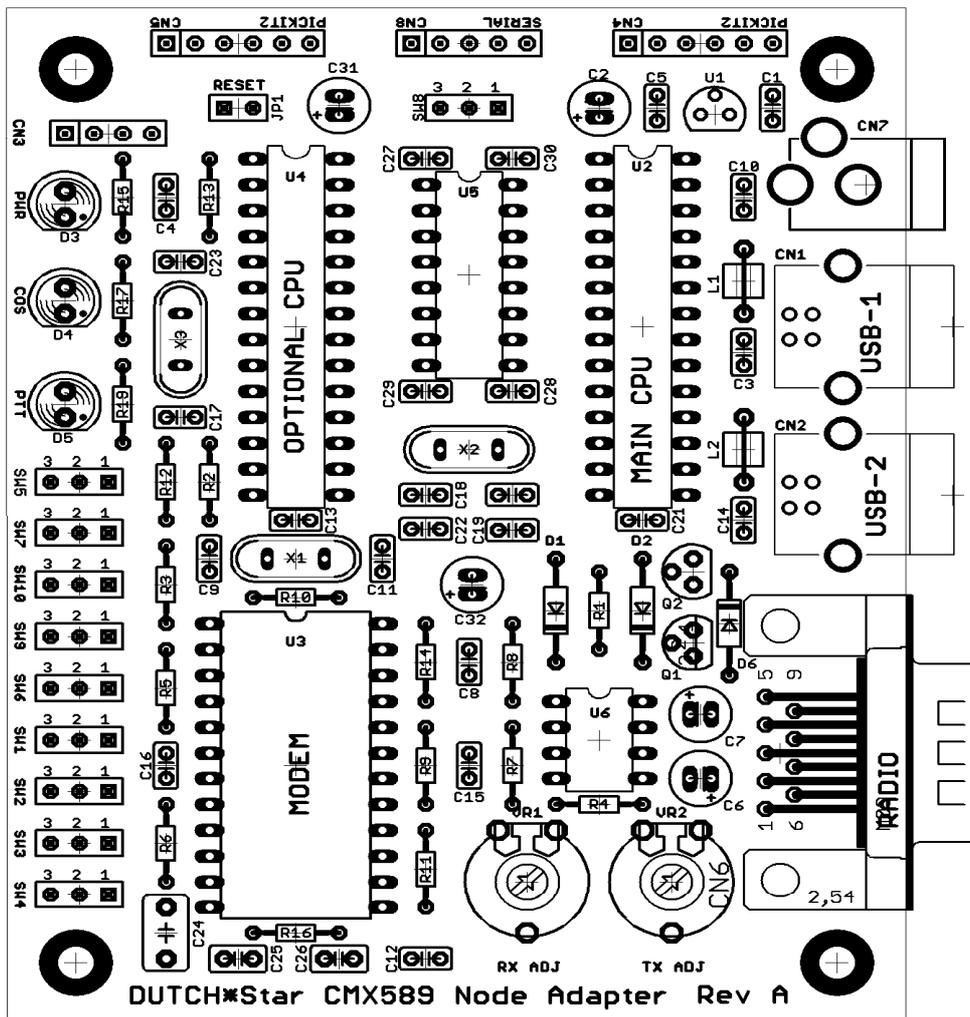
After clicking on the **Close** button, the Update Firmware procedure closes, and we will be back in the main screen of the NAWinCFG program.

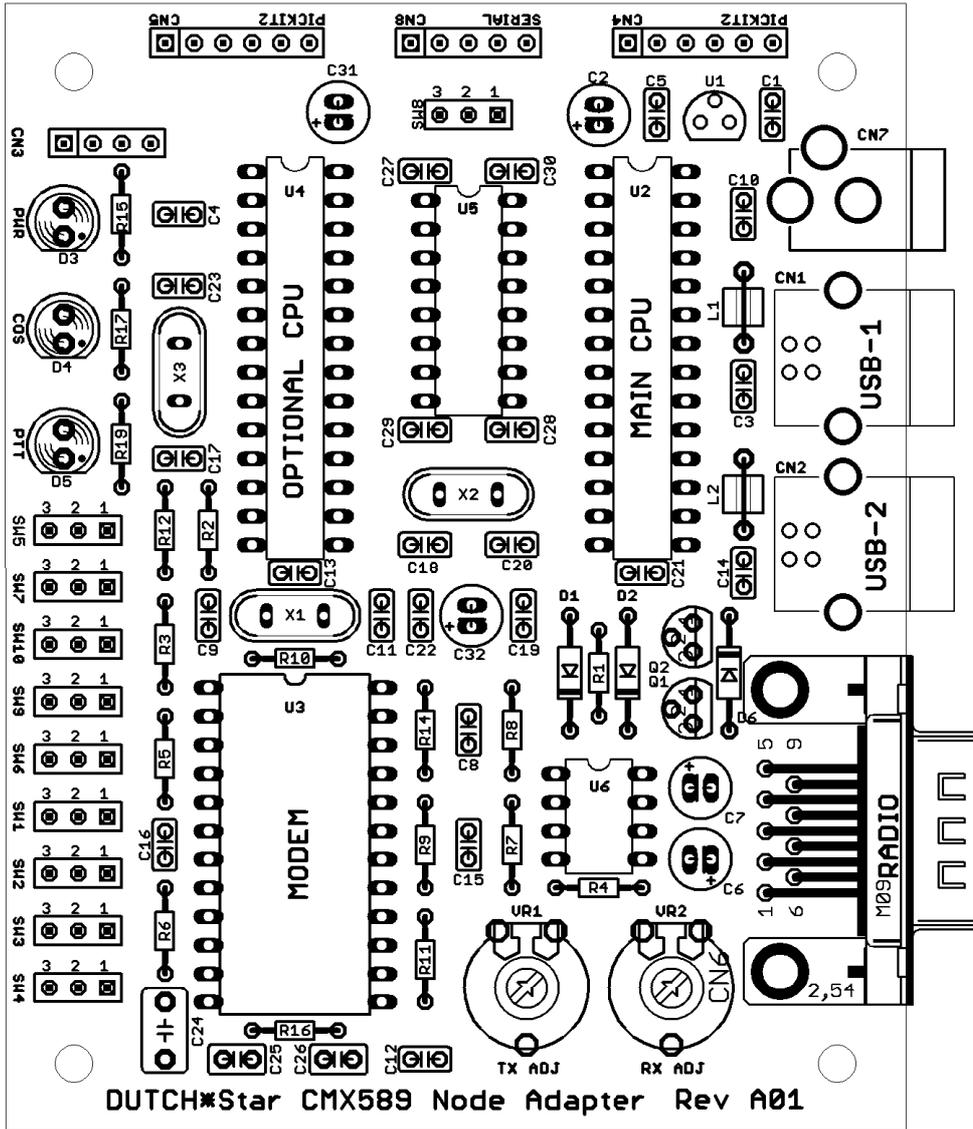
Normally, it should now see a board, and the firmware version info should correspond with what you had just programmed into it:



# Appendix A: PCB Parts Placement

These are enlarged versions of the PCB's to give you a better view of where the various parts are on the board. Make sure you are looking at the correct revision of the board; parts may have been changed around between revisions!





## Appendix B: Parts List

### B.1 Parts for Rev.A

Your kit should contain the following items:

2	PinHeader	1x6	CN4,5		
2	PCB Socket	USB-B	CN1,2	2	Resistor 330 R17,19
1	PCB Socket	DB9-M	CN6	1	Resistor 470 R15
1	PinHeader	1x4	CN3	5	Resistor 47K R1,2,3,5,12
1	DC SOCKET	NEB-21R	CN7	4	Resistor 100K R4,6,11,14
1	PinHeader	1x5	CN8	1	Resistor 180K R8
10	PinHeader	1x3	SW1,2,3,4,5,6,7,8,9,10	2	Resistor 470K R7,9
				2	Resistor 1M R10,16
2	Ferrite Bead	FB-110	L1,2		
				1	Trimpot 500K VR2
7	Capacitor	22p	C9,11,16,17,18,20,23	1	Trimpot 1M VR1
1	Capacitor	330p	C8		
1	Capacitor	470p	C12	1	LED 5mm BLUE D3
2	Capacitor	22n	C25,26	1	LED 5mm GREEN D4
13	Capacitor	100n	C1,3,4,5,10,14,15,19,22,27,28,29,30	1	LED 5mm RED D5
2	Capacitor	470n	C13,21		
1	Capacitor	1uF	C24	3	Diode 1N4148 D1,2,6
				2	Transistor BC547 Q1,2
1	Elco	4.7u	C32		
2	Elco	10u	C6,31	2	IC SOCKET28P-3 U2,4
2	Elco	47u	C2,7	1	IC SOCKET24P-6 U3
				1	IC SOCKET16P U5
1	IC 78L05Z		U1	1	IC SOCKET8P U6
2	IC PIC18F2550 - I/SP		U2,4		
1	IC CMX589AP4		U3	1	XTAL 4mm 4.9152 X1
1	IC MAX232A		U5	2	XTAL 4mm 24.0000 X2,3
1	IC RC4558P		U6		
				10	JUMPER 2.54mm RED,BLACK
				1	PCB Node Adapter

If all parts are not in your kit, **STOP NOW** and contact the people at DUTCH\*Star!

Although all efforts have been made to get it right, mistakes do happen, and those can usually be fixed easily!



## Appendix C: Corrections

### C.1 Corrections for Rev. A boards.

If you have a Rev. A board:

- Diode **D1** must be placed in reverse polarity.
- Do not place resistor **R13 (10K)** and jumper **JP1 (RESET)** onto the PCB. Discard these parts from your kit if it is present.
- The labels **RX** and **TX** near the two trimpots **VR1** and **VR2** have been swapped... **VR1** is actually **TX**, and **VR2** is actually **RX**. Keep this in mind when adjusting the audio levels!
- Transistor **Q2** has been changed from a 2N7000 FET to a regular BC547 NPN type; if your kit contains 2pcs of BC547, the second one goes in place of Q2, but **YOU MUST PLACE IT IN REVERSE PIN ORDER** compared to the drawing on the PCB!
- Solder a wire between the collector of **Q1** and **Pin4** of **CN6** (the DB9 connector.)
- Solder a wire between **Pin25 (RB4)** of **U4** (Option PIC) and **Pin3 (RA1)** of **U2** (Main PIC). Also, solder a wire between **Pin24 (RB3)** of **U4** and **Pin1** of **SW9**.

The comments regarding R13 and JP1 above are valid if the board is used with firmware other than DUTCH\*Star HSA, such as the software written by Satoshi Yasuda. HSA firmware DOES require these parts to be installed on the board!

These issues have been corrected in Rev.A01 of both the schematics and the PCB. If you have a more recent PCB revision, you can ignore the above corrections.

