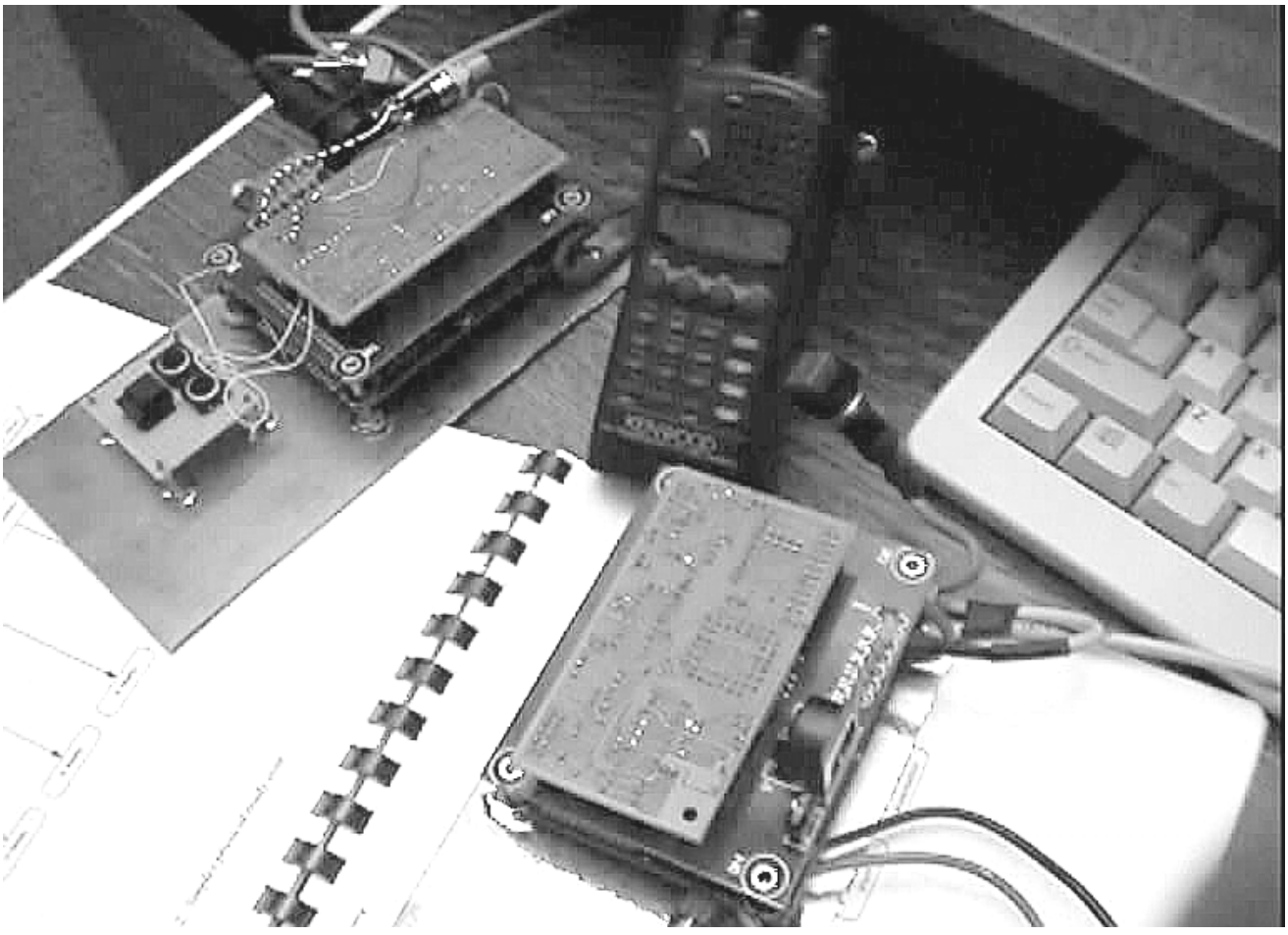
MCB152
Micro Controller Board 80c152

MCB152

Micro Controller Bord including Software and Development Tools. Perfectly suited for DATACOMMUNICATION. Especially for Radio-Amateurs and students in Micro-electronics.

Engage yourself into the wonderful world of microcontrollers and discover the secrets of the Digital Highway.



I. INTRODUCTION

A. Historic

At the beginning of INTERNET, we started with nothing. Now can be proud to present our **MCB152** microcontroller board.

This controller board is only one part of subsequent innovative changes. Soon after we launched our local TCP server, we discovered that the current TNC types were too slow to use all the features TCP could give us. Moreover we observed that the development of specialised software for AX.25 - which would present the same user friendly environment as commercial TCP/IP software - would take us at least several years to develop. On the contrary, state of the art TCP/IP software is widely available.

By definition, TCP/IP is an open protocol. It is included in many different operating systems and provides us the possibility to transfer data between those systems.
Because Radio-amateurs are obligated to use AX.25 as data protocol, we will use AX.25 as our datalink layer.

To be able to use standard INTERNET software, we should develop a TNC that speaks to standard INTERNET software at one side and to the AX.25 Radio-network at the other side. That was our primary goal for the **MCB152** project.

B. Concept

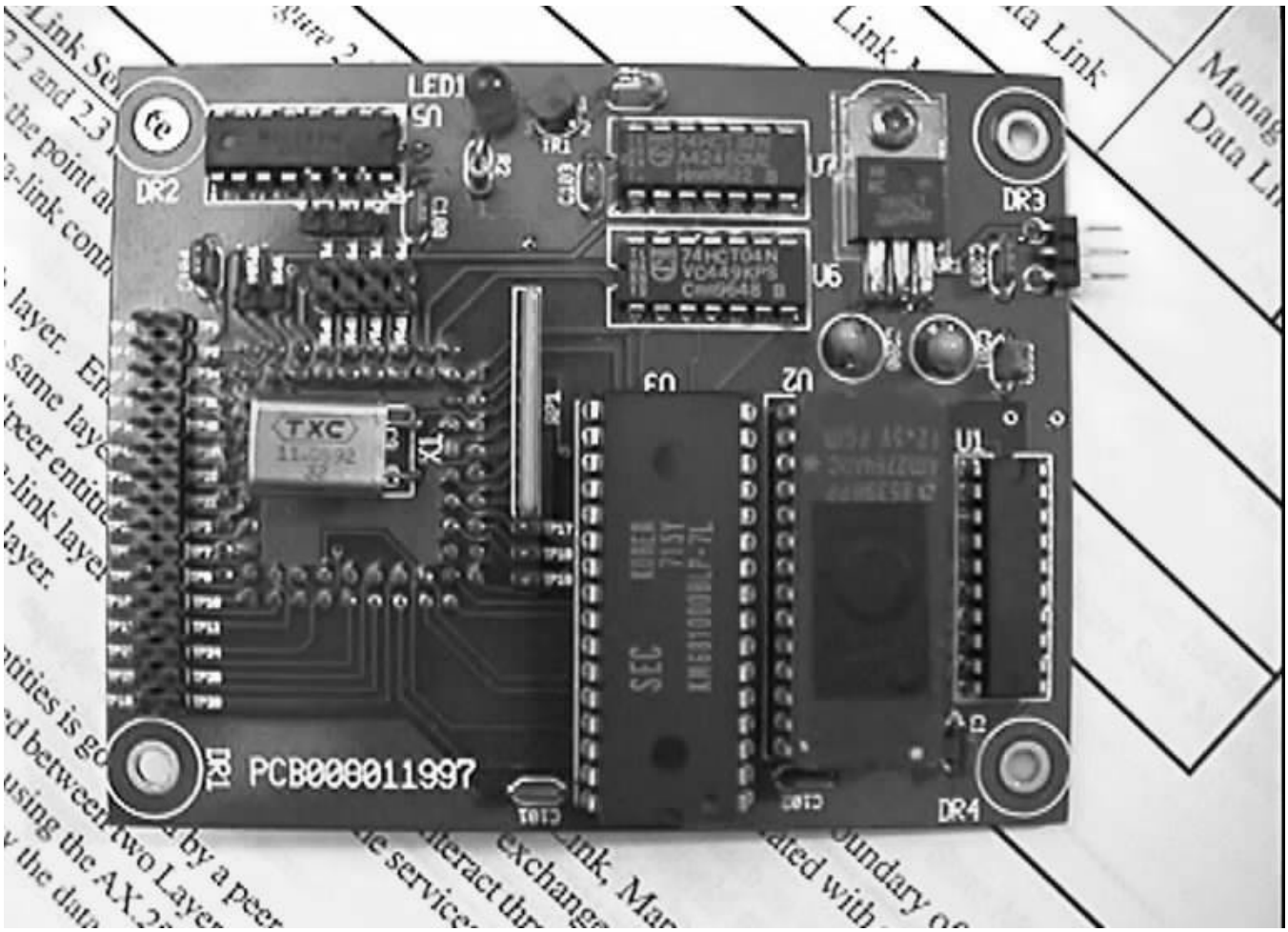
Because we hoped that not only Radio-Amateurs would use our project, we tried to make a universal Microcontroller development board.

1. Hardware

The hardware basic concept exists of separate boards that can be plugged on top of each other. This leaves us the possibility to expand and change the primary concept without reconstructing everything.
As a TNC, the project is composed of three different PCB's: The mainboard, the interconnection PCB and the USCC modem (DK9RR).
This is the primary project that can and will be extended with other versions of modems or I/O equipment.

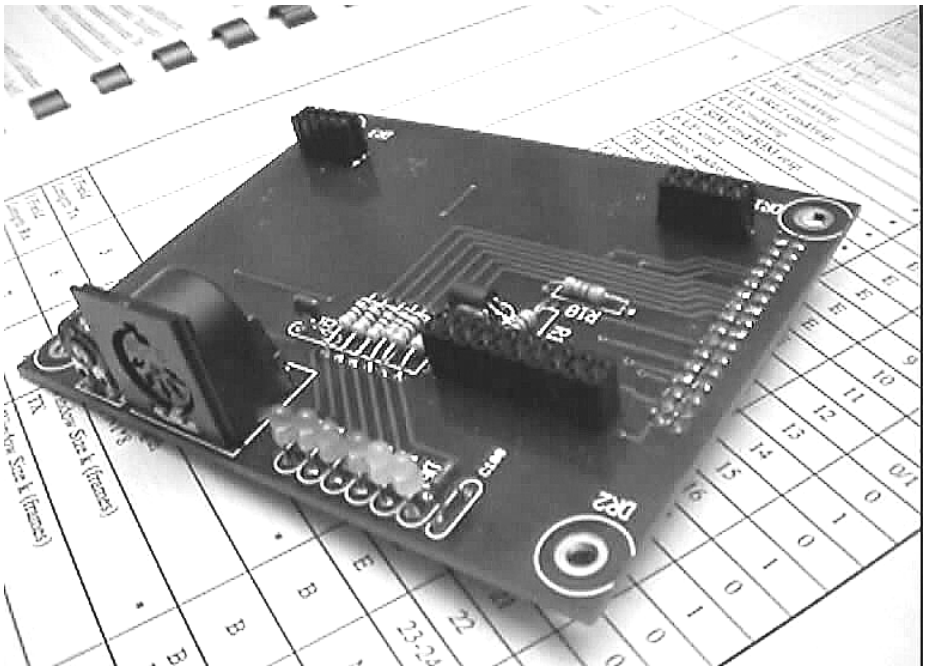
a. Mainboard

The mainboard contains a microcontroller Intel 80c152 and a few extra circuits. The RS-232 port is used to connect the **MCB152** to a serial communication port of a PC. The **MCB152** is equipped with the maximum available CODE- and DATA-memory, 64kB each, contained in one 128kB RAM. The on-board EPROM is programmed with the **KICKSTART** debug and upload environment. The 5 Volt regulator provides us the possibility to connect the board directly to a 12VDC power supply, commonly available in a radio-shack. There are connections available for the RESET button and for the RS-232 datacable.



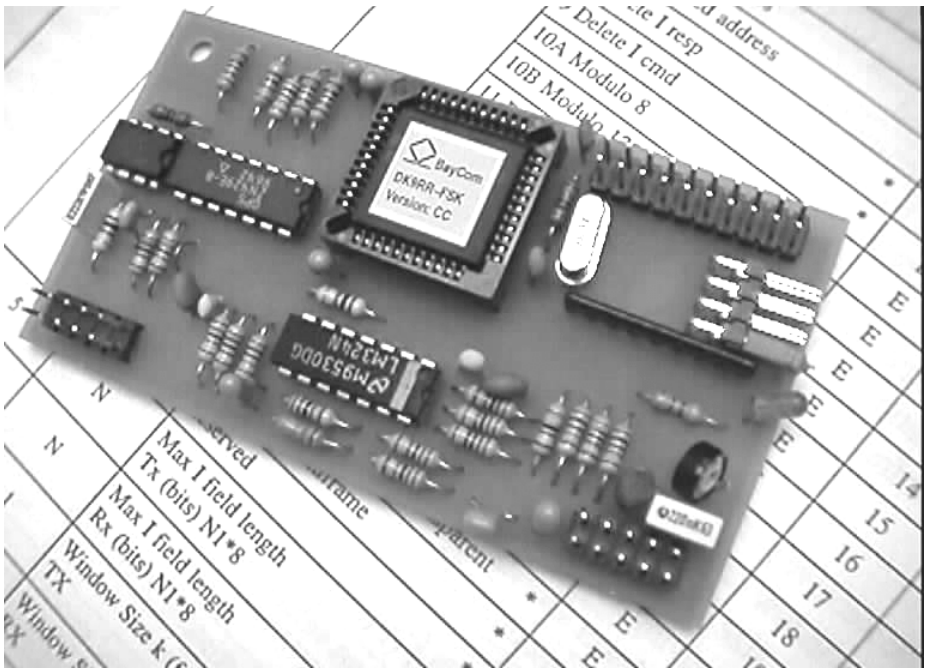
b. Interconnection PCB

This PCB contains very few components. Its only purpose is to make an interconnection between the I/O ports of the microcontrollerboard and the transceiver. On this board we also provided the connectors on which we plug the DK9RR USCC modem.



c. USCC modem

Here we use an existing modem made by the BAYCOM group in Germany. Besides powerful debug and test options, the modem is also capable to encode/decode 1200Bd AFSK, 4800bd and 9600bd G3RUH packet signals. The modem can be included with the package or you can order it yourself directly with Baycom in Germany.



2. Software

The concept is to meet the wishes of a wide public. Everyone who wants to get some experience with the MCS51 microcontroller series will be pleased with this design. The 80c152 is a 80c31 CPU with an extra communication and DMA controller. This gives us the opportunity to make FULL duplex communnication at a speed as high as 680kBaud.

To avoid the programming of EPROMs, we included the principle of uploadable software. The format of that software is an INTEL-HEX file, widely used as exportformat of many assemblers.
The **MCB152** is equipped with an automatic baudrate detection. The following baudrates can be used: 1k2, 2k4, 4k8, 9k6, 19k2, 28k8, 57k6.
You can upload the firmware with a simple 'copy' command from every computer equipped with an RS-232 communication port.
We forced ourselves to occupy as less resources as possible from the mainboard resulting in the maximum flexibility. So we can use all of the CODE- and DATA-memory as well as all I/O pins, except the memory addresses FFF0h through FFFFh and I/O pin P1.7. Those are used to switch from **KICKSTART** to the uploaded program.

The **KICKSTART** gives us the opportunity to debug a uploaded software in CODE memory. So we can first upload a program and then use the **KICKSTART** to disassemble the code with the 'D' command. With the 'M' command you can get an ASCII dump of the program-memory. To clear the program memory, we could use the 'C' command. The 'G' command lets us start the uploaded program or a routine somewhere in code memory. To change or control the contents of the registers, you could use the 'S' command. If your test of the uploaded program fails, you could use the RESET button to restart the **KICKSTART** and use it to debug the uploaded software. The software stays in CODE memory until you clear it or you have had a power down. The '?' or 'h' command gives you more explanation about the commands available in the **KICKSTART**.

The sequence of upload and execution is as follows. When the **KICKSTART** receives Intel-HEX lines, it starts decomposing these lines and puts them in CODE memory. If the **KICKSTART** gets an Intel-HEX line coded as an 'END-OF-FILE', it will switch automatically to the uploaded program and start executing. The red LED turns off at this time.
The 'END-OF-FILE' format in an INTEL-HEX file can be recognized by the code :0001.... as the first 4 digits and will most frequently be included at the end of the .HEX file.

We will deliver the board with fully functional software and an assembler with libraries for an IBM compatible PC. The uploadable software can be uploaded from every PC, which is capable of communicating with a RS-232 device.

3. Conclusion

Although this project describes a TNC as primary project for this controller board, it can also be used as a 'first step' for those who want to discover the wonderful world of microcontrollers. It will be the basic board for our future projects. We also hope to encourage those who want to experiment with microcontrollers and apply them with a fully functional development-system.

To achieve this goal, we forced ourselves to distribute all hardware and software diagrams, programs and all source codes for free. We hope that everyone working with this board will join our filosophy.
Please, if you distribute your source-code, include your name and contact-medium.

II. CONSTRUCTION

A. Precautions

Keep the following advise in mind when you want to complete your kit as professional as possible. The fall-out will decrease remarkably if you follow these safety procedures.

- Find an organized, clean and flat surface on your workbench, preferrably some conducting surface plate that is connected to the groud potential or to the same potential as your soldertip. I used a single-sided PCB board, connected with some copperwire to the metal part of my soldering iron, or, if the soldering iron isn't grouded, to the groundpin on the main supply.

  BEWARE: DO NOT USE any synthetic surface like plexiglass, a piece of carpet,.. These are extremely static sensitive materials and are easily inflammable.

- Provide sufficient ventilation in the room, since the fumes are toxic. An easy way to achieve this is to put a blower in front of you, blowing away from you. This will take the fumes away from your face and doesn't give to much airflow in your eyes.

- Use a soldering iron with a small long-live soldertip (1mm). It will avoid to make short-circuits and bad solderings.

- Use high quality solder with flux and a diameter no more than 1.5mm.

- Do not use extra flux. If a component can not be soldered, you should look for durt, oxidation or greasy connection leads. Remove the oxidation by scrapping with a knife or a fiber brush. This brush can be found in a bookstore and is normally used to correct things on drawing paper.

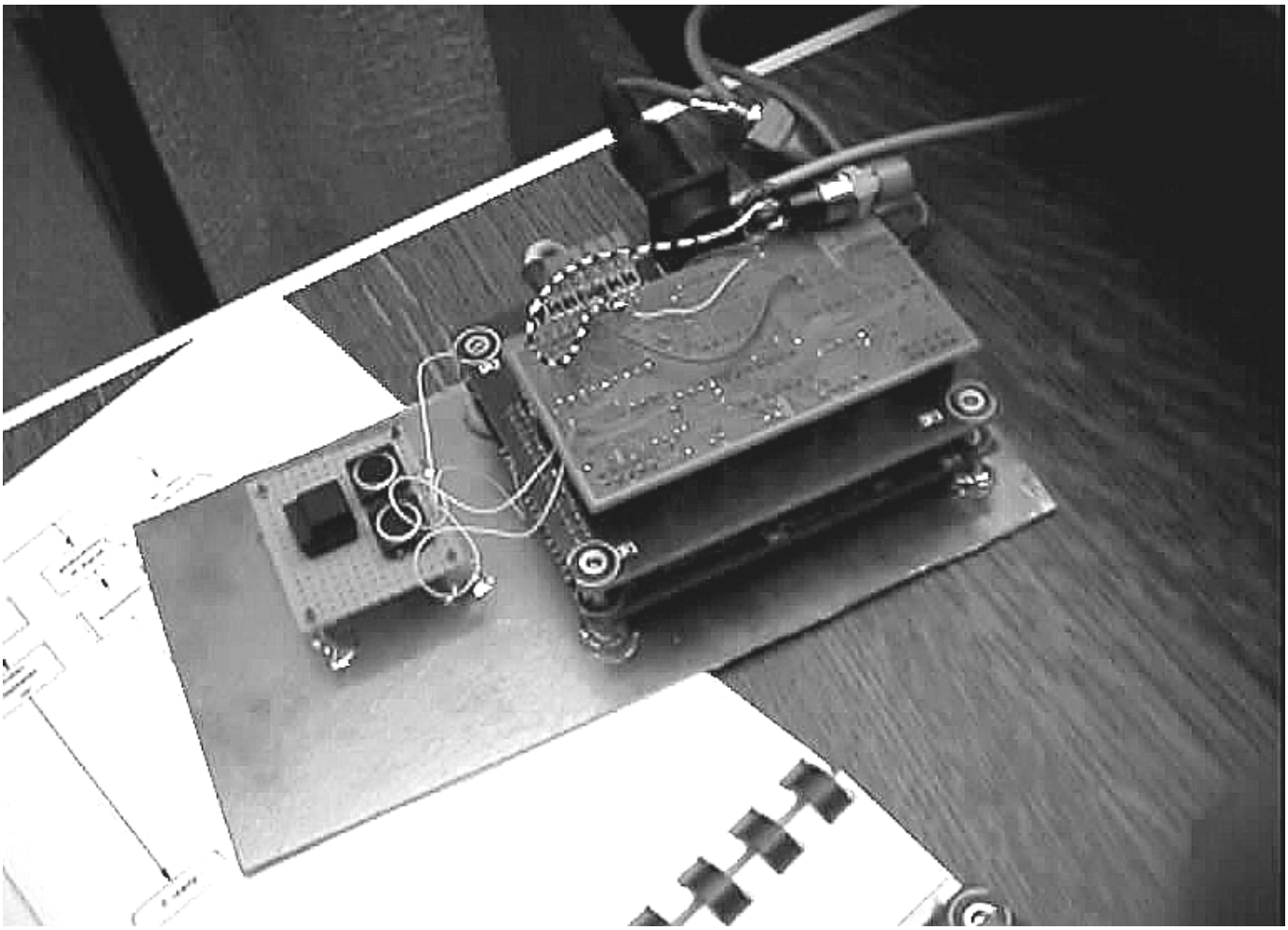
- The cleaning of the solderpoint is done on a moistured sponge (hard side of the sponge). As an alternative, you could use a piece of folded moistured cotton. You should clean your soldertip BEFORE you start soldering, NEVER AFTER.

- A good soldering can only be made if you heat up both the component leads and the solder island. The solder should be applied at the opposite side of the heating place. The solder should start melting due to the temperature of the component leads and the solder island. First remove the solder and then the soldering iron. Place the soldering iron in its stand without cleaning the tip.

- Cut the component leads shortly, but avoid to cut into the soldering itself.

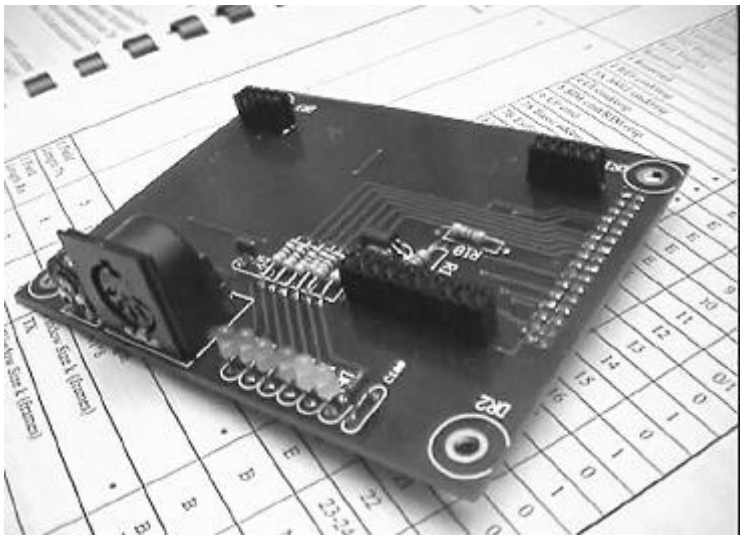
B. Mouning PCB

You should consider care of the mechanical construction and the mounting of the PCB's before you start soldering.
The picture should give you an impression on how the project should look like when it is finished. This should give you the possibility to construct a mounting base or to make the chassis for your project before the first soldering is done. Normally, all connections are present on the PCB, but you can also provide external connectors for the TRX plug and LED mounts on the outside of your chassis.
For those who want to experiment, they should also preserve a RESET button. Probably the easiest way is to mount the project on a piece of blanc PCB board, so every connection is accessible.

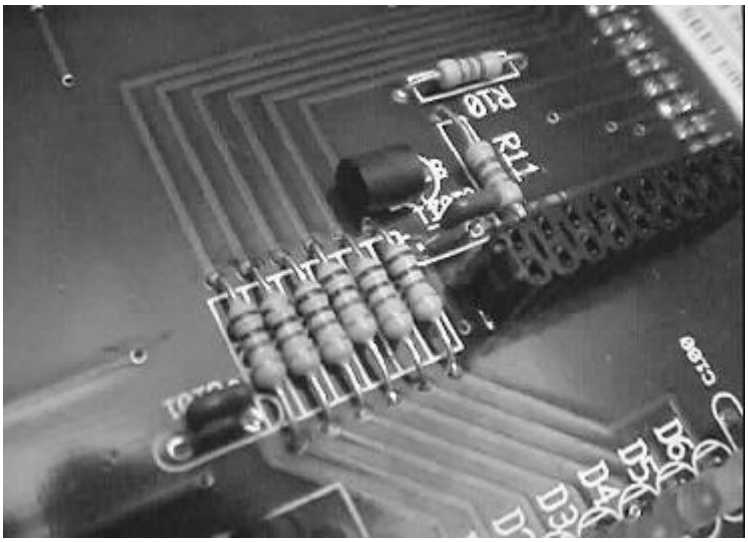


C. Constructing the interconnection PCB

Considered this PCB is less complicated than the other boards, we will start with this one. First we will mount the connectors.



- Put the DIN5 PCB connector on the PCB as marked on the component side.
- Turn over the PCB and keep the connector in place.
- Solder 1 lead to the PCB.
- Make sure that the connector is still in place.
- Solder all the other pins of the connector.
- Mount all the LEDs. For D1 to D4 we will use green ones, D5 and D6 are red ones. The long lead of the LED, indicating the anode, will be mounted near the outside of the PCB. Push the LED in place as far as possible and bend the leads on the solderside a little bit apart. Solder the leads and cut the remaining part of the leads.
- Mount the resistors R1 to R6 1k5 (Brown-green-red or brown-green-black-brown).
- Mount the resistors R10 to R11 4k7 (yellow-violet-red or yellow-violet-black-brown).
- Mount transistor T1 type BC547 (C547B) as marked on the PCB. Don't push it to deep in place. After soldering bend the transistor as shown on the picture.



- Mount the potentiometer P1 470 Ohm.
- Mount capacitor C100, C101 and C102 100nF (103).
- Cut the female double-row header to a length of 16X2, two times the length of 5X2 and once the length of 10x2.
- Place the 10x2 and the two 5x2 headers on the component side of the PCB. After soldering one lead, make sure that the connector is still in place. Solder all the leads.
- Place the female 16x2 header on the SOLDER side. You should solder the leads at the component side.

This will complete the construction of the Interconnection PCB.

D. Constructing the mainboard

1. PLCC68 socket

Before we mount the PLCC68 socket it is necessary to place some other components. Those will be placed and soldered at the component side of the PCB. Cut the leads as short as possible on the solderside.

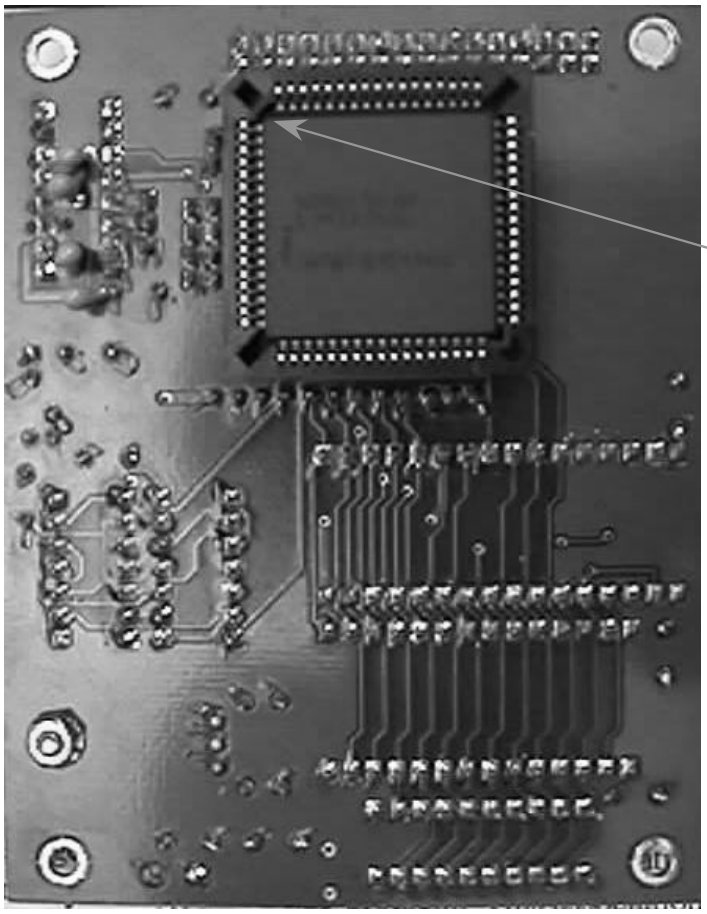
- Mount C10 100nF (104) and C2 470nF (474) on the component side and solder at the component side
- Mount C8 and C9 27p (270 or 27) on the component side and solder at the component side.

The Xtal will be placed later.

- Cut all the remaining leads at the solder side.

Now we will place the PLCC68 socket at the SOLDER side of the PCB.

- Look for Pin 1 on the PLCC68 socket. It is marked with a little triangle in the middle of one of the sides.



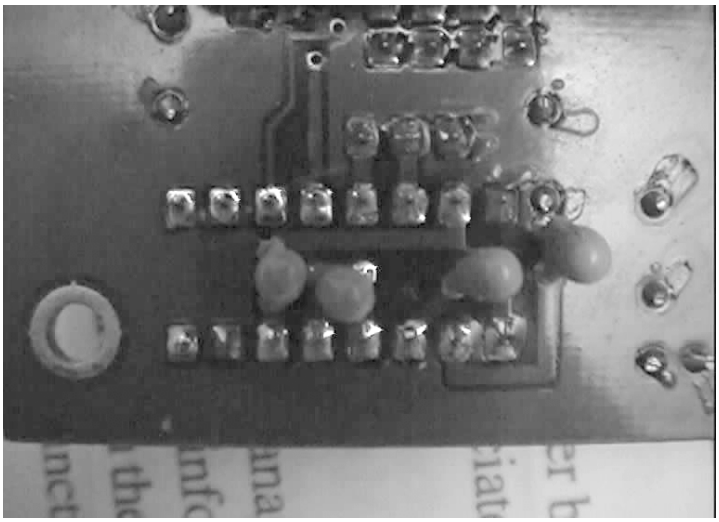
- Put the socket in place, the indication of Pin 1 toward the connections TP1..TP25.
  If we take the PCB board with the solder side toward us and the double-row connector away from us, you will find the non-rectangular shaped corner of the socket in the upper-left corner of the socket.

- Solder one lead. Make sure the socket is still in place before soldering the others.

2. Tantal condensators for the MAX232 RS232 convertor

These will be mounted on the solder side of the PCB and soldered at both sides. We will place C4 to C7 1 uF ("1. 35" stands for 1. uF 35 Volt). The positive lead is marked with a black strip or a + sign on the package. This lead is frequently longer than the other.

- Put the capacitors in place as shown on the picture. You should check the polarity of the connections with the schematic.



3. LED1 and resistors

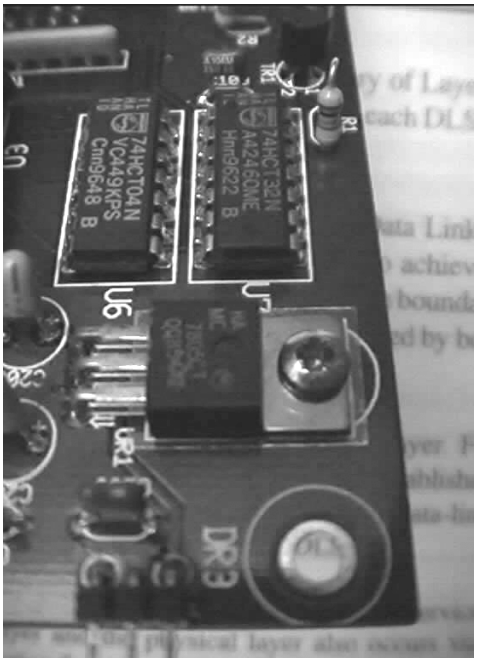
LED1, as well as all the other components, will be placed on the component side of the PCB board.
The flat side of the LED will be placed towards TR1.

Resistors R1 and R2 will be placed standing.
Placement of TR1 points out itself

4. Voltage regulator UR1

The placement of UR1 type 7805 should take some preparation.

- Take UR1 and bend the leads so that the component fits the drawing when it is placed. Especially watch for the mounting hole.
- Put the isolating mica plate underneath UR1. Make sure the back of UR1 is completely covered by the isolator.
- Fix UR1 with the screw. Make sure the plastic washer is in place. Solder all the leads at both sides of the PCB.
- Place C200 and C201. These are Tantal types 10mf (10.16 10 mf 16 volt). Check the polarisation.
- On the resistor network RP1, you will find a dot marking pin1. This network will be placed such that pin 1 will be placed beside TR1.



5. Header 2x16

This male header is placed on the component side of the PCB.

- Place the connector and solder just 1 lead.
- Make sure the connector is still in place before soldering all the other leads.
- Put the other connectors in place.

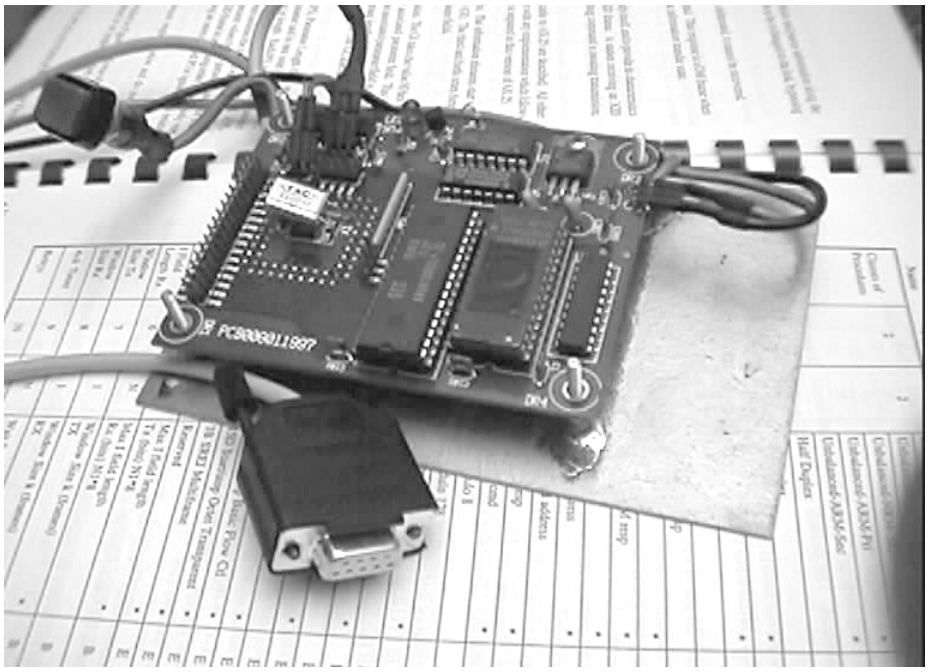
6. IC’s

The IC's U1, U5, U6, U7 are soldered directly in place. For U2 and U3 we will use IC sockets.
- The IC's U1, U5, U6, U7 are soldered directly in place. For U2 and U3 we will use IC sockets.
- Place the IC socket for U2. Make sure the solder is applied through the solder island to the top of the PCB. If not, try to correct the problem. Eventually solder those points at both sides.
- Put IC socket for U3 in place.
- Put U5 in place after cutting the leads of C4 and C7 as short as possible.
- Place IC's U1, U6 and U7 as marked.

7. Decoupling-capacitors

Those are all the remaining capacitors. These are 100nf ceramic types (104).
Make sure they are soldered on both sides.

8. Connection leads



a. Power supply

The power supply (minimal 8v DC) can be obtained from a standard 12VDC power supply, commonly available in most radio shacks. This will be reduced to 5V DC on the microcontroller board (UR1). The connections are TP201(GND) and TP200 (+PWR).
The upper surface of the PCB will carry the +5V DC and the bottom surface the GND.

b. Reset button

It is not nessesary to use a reset button. The processor is resetted during power on. For those who want to use the **MCB152** as a development system, it is advisable to provide a RESET button.

The reset button is a normal-open button connected across TP35 and TP104.

c. RS-232 connections

These connections are used to make datacommunication possible between a PC and this microcontroller board. The connection exists of three signals.

TP3	RXD	signal from PC to **MCB152**
TP4	TXD	signal from **MCB152** to PC
TP103	GND	Ground signal

At the PC side we use a DB9 connector.

- Make a connection on the DB9 connector between DCD(1), DTR(4) and DSR(6).
- Make a connection on the DB9 connector between CTS(8) and RTS(7).
- Connect the GND (TP103) of the **MCB152** to the Signal Ground (5) of the DB9.
- Connect TXD (TP4) of the **MCB152** to Receive Data (2)
- Connect RXD (TP3) of the **MCB152** to Transmit Data (3)
- Connection (9) on the DB9 is not used

9. Mounting IC's

Because most of the ICs are sensitive for static discharges, it is better to defer the mounting of these devices. You should avoid wearing synthetic clothing or sitting on synthetic materials. Try to ground yourself and the surface you are working on. Put all components in your vicinity, so you don't have to move around with unpackaged ICs.

You will notice that the connection leads of the ICs are too far apart to fit directly into their sockets. This is not a mistake. To make them closer you should bend them all at once. This can be achieved as follows:

- Take the component with two hands and place it with one side of the leads on the surface of your workBench, the leads pointing away from you.
- Push the component down and bend it forward. All leads will be bending equally.
- Try pushing it in to its socket.

a. U2 and U3

Make sure that the ICs are placed in the appropriate direction. The printed design on the PCB surface gives an indication.

b. U4 PLCC68 Microcontroller 80c152

Placement of the microcontroller into its socket demands some skills. we will explain one method of inserting the PLCC68 chip.

- Locate the position of Pin 1 on the chip. This pin is marked with a dot on the IC package.
- Place the IC on top of the socket, such that pin 1 is pointing to the 16x2 header. Be sure that all connection leads are covered by a connector lead in the socket.
- Push - with equal pressure - at both sides of the IC package until the IC is completely pushed into the socket.

BEWARE: If you do not push equally, you will destroy the socket!. The IC must fit easily and with a 'click'.

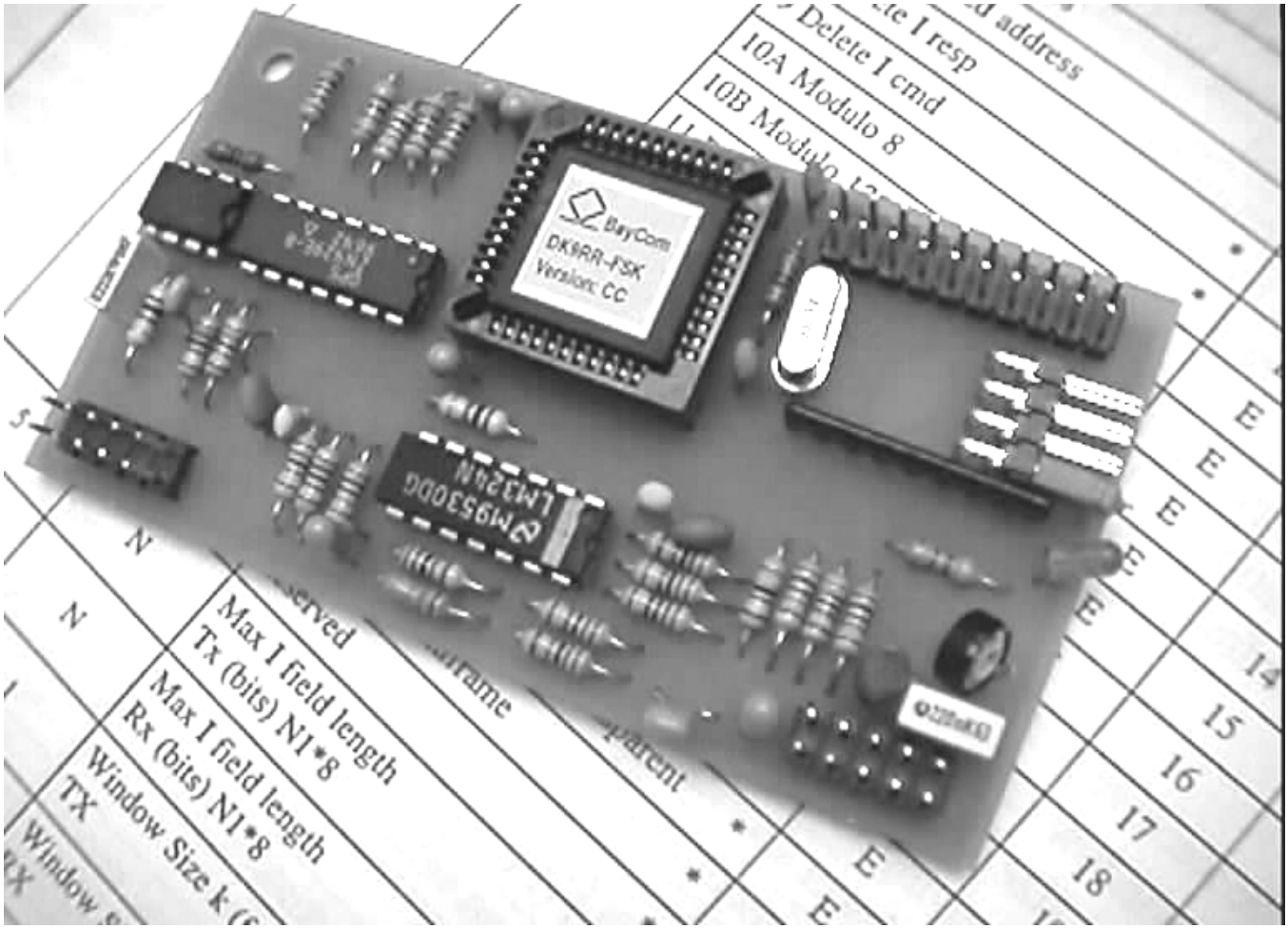
c. Mounting Xtal

Because this component is shock sensitive, it is placed as the last component on a PCB. The component will be soldered at the component side and bent toward the PCB. Avoid to bend too far so that the metal package of the Xtal touches the solder islands of the PLCC socket

This was the last component to be mounted on the mainboard.

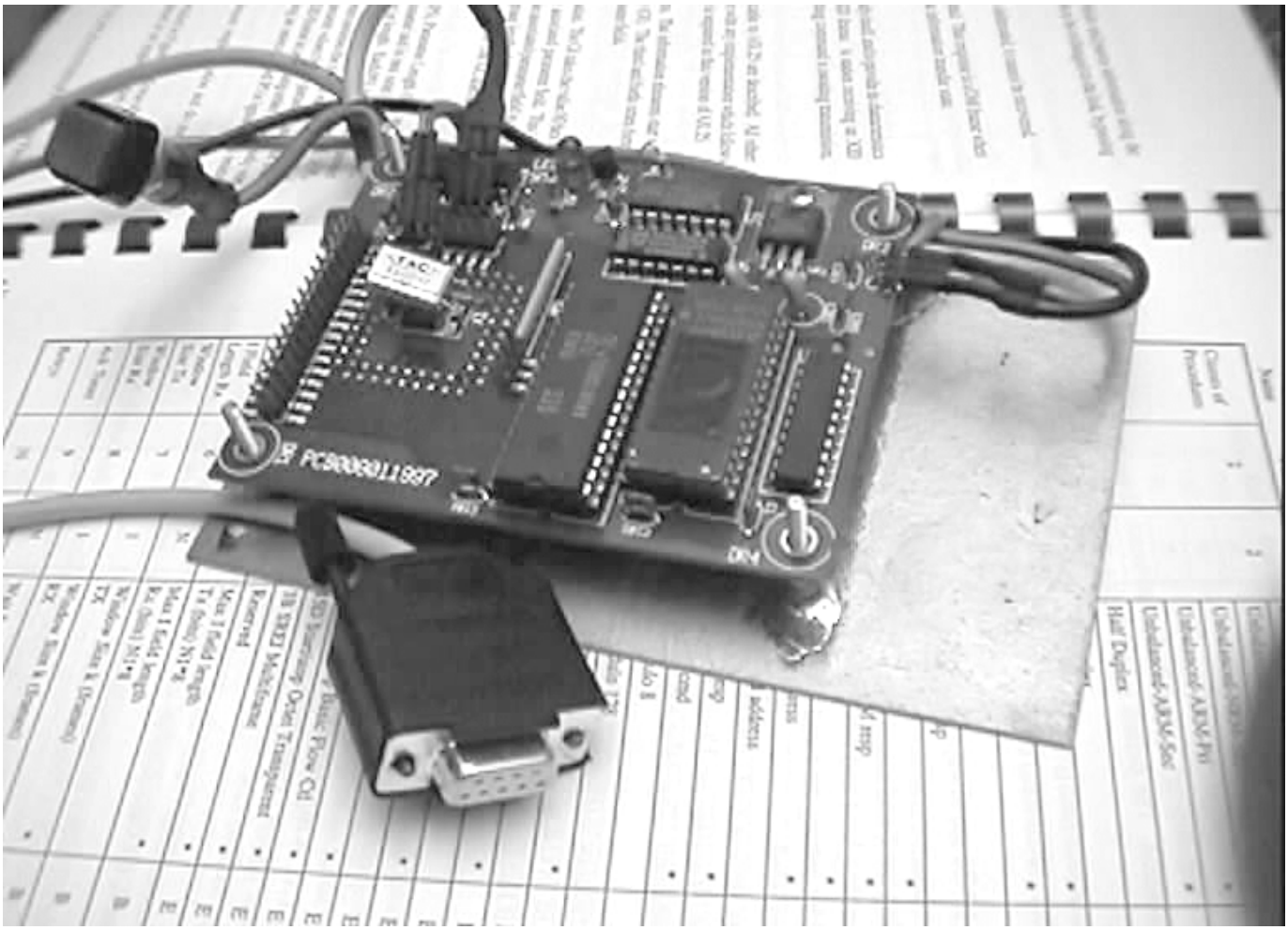
E. Baycom DK9RR modem

The construction of the modem is described in a separate paper from the Baycom group. Pay attention to the positioning of the PLCC socket and use the same precautions for mounting the sensitive IC's.



III. TESTING

A. Testing mainboard

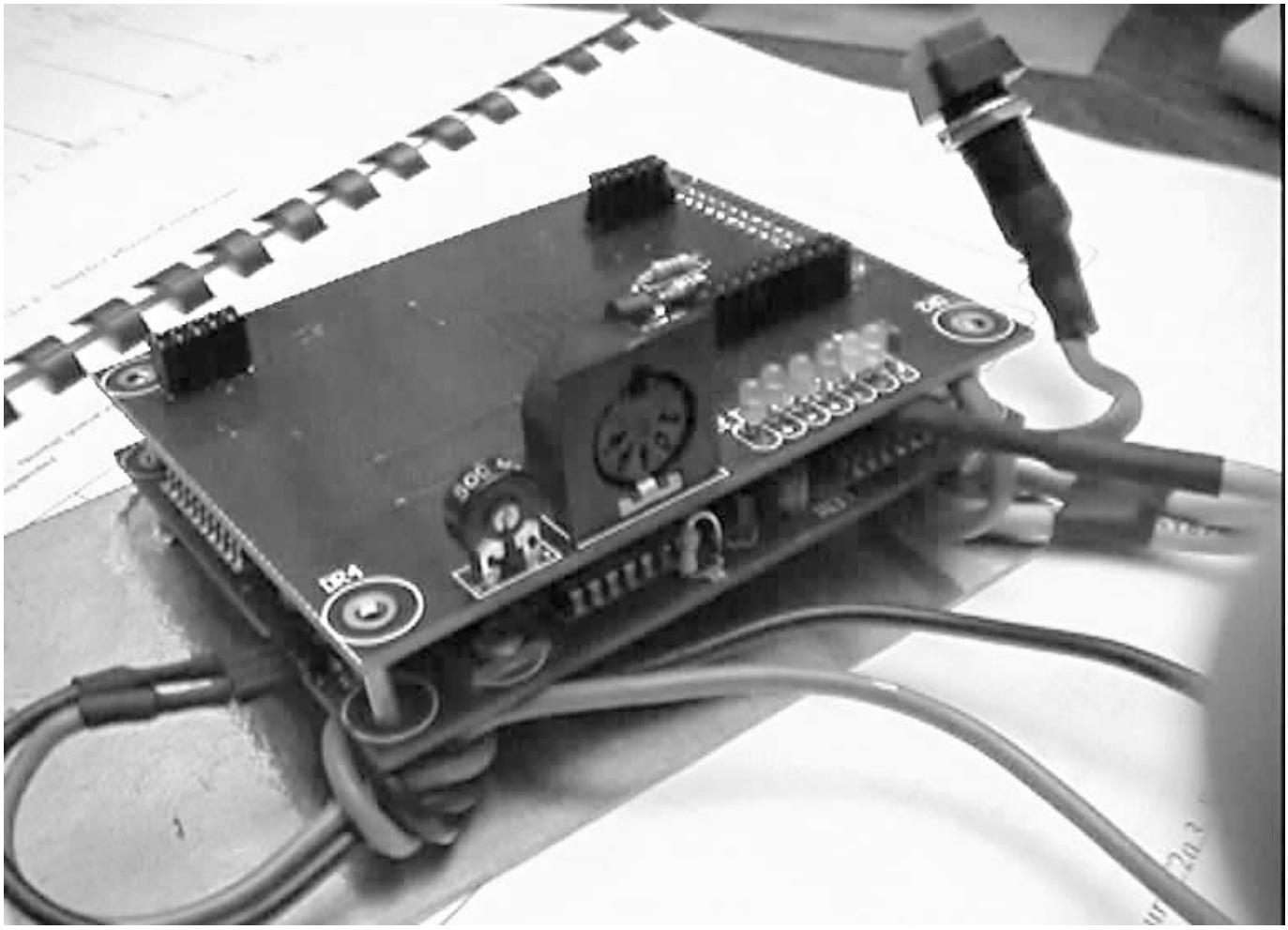


You should be familiar with some basic PC skills. We assume that you have some basic knowledge about the hardware and the command set of DOS. The initial program to be used is a terminal program like Hyperterminal under Windows.

- Connect the DB9 connector to a free COM-port.
- Start the terminal program and change the settings of the coresponding COM-port to the following: 8 bits, no parity and 1 stopbit. You may use one of the following baudrates: 1200, 2400, 4800, 9600, 19200, 28800, 57600.
- Select NO handshaking or Hardware handshaking, but certainly not Xon/Xoff.
- Supply the **MCB152** with power. The red LED on the Mainboard should glow.
- To start the communication with the **MCB152** press <SPACE> twice. This will indicate that you want to give manual commands to the **KICKSTART** firmware.

There should be a "0000>" prompt on your screen now. Now it is possible to start entering a set of different commands like ? or H for help. This will present you a list of all available commands in the **KICKSTART** Bootloader.This procedure completes the test of the main board. Now you are able to upload your own software or software developed by others. Our first project with this microcontroller board is a modern TNC. This software and hardware is included in this kit as well as some test and debug software, as you will see later on.

B. Testen interconnection board

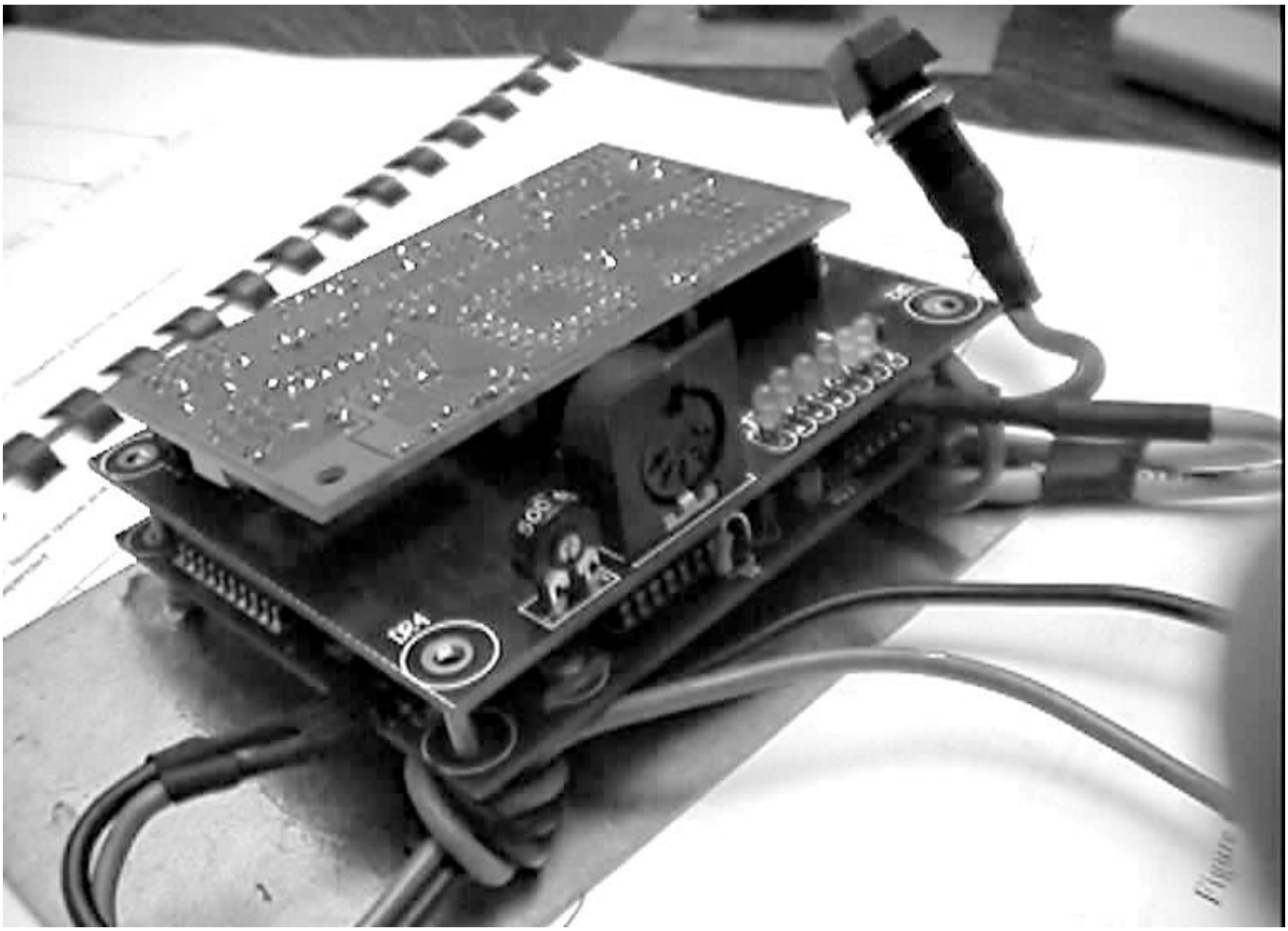


The LEDs on this board are connected to the I/O pins of the microcontroller through the big header. The solder islands of the header are relatively small. This will raise the chances for short circuits.
After a visual check, you can use the following testprogram to verify.

- Put the interconnection board on top of the main board.
- Power up.
- Start your terminal program and upload the LEDTEST.HEX file to the **MCB152** (type ASCII TEXT).

You will see a running light changing every 0.7 sec. Only one LED at a time may glow. If more than one LED lights up, you should look for short circuits.

C. Testing DK9RR modem



To be able to test the DK9RR you should already have made the connections to your Transceiver. The levels of the in and out signals of the DK9RR are high enough to be connected directly to the modulator and demodulator of your transceiver. Most modern transceivers have signals available on a plug, external to the device, known as DATA_IN and DATA_OUT. It is advisable to have a second transceiver at hand, so you can monitor your own TX signal.

The connections of the TRX plug are conform the BAYCOM standard.

1 MIC or DATA IN
4 LS or DATA OUT
2 GND Ground
5 Not connected
3 PTT (make contact to GND if active)

- Put the DK9RR on top of the interconnection board and set to 1200bd.
- Power up.
- Restart your terminal program and set the baudrate at 9600Bd.
- Upload the file DK9RRTST.HEX to the **MCB152**.

After uploading, you will see that all LEDs light up. Also the red LED on the mainboard will turn off, indicating that the currently executed program originates from the RAM and not out of the **KICKSTART** EPROM.

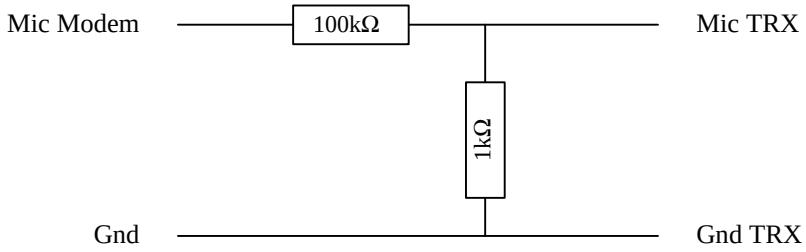
This test program sends a frame every 5 sec. If all connections to your transceiver are made correctly, your transceiver will go on air every time the RED LED on the DK9RR lights up. This LED will turn on if the microcontroller gives the command (D6 lights up). The trimmer on the DK9RR sets the TX audio level. The TX level should be set as low as possible (minimizing bandwidth of your TX signal). The trimmer on the interconnection board sets the RX level. This should be set experimentally. D4 indicates reception of data signals. D3 indicates a decoded frame is transmitted to the PC. If you have reception but no decoding, you should try another setting of the RX level. This is not the single possible problem but it could be one.

Description of the LEDs used by this program:

-	D6	red	PTT	Frame in transmission (on air).
-	D5	red	LSC_RX	Frame received from PC.
-	D4	green	DCD	Data signal detected.
-	D3	green	LSC_TX	Received frame decoded.

Remark:

- The DK9RR is a very sensitive decoder. If your receive audio level is too high, you will not see optimum decoding. Adjust your receive audio level properly.
- If you connect the TX audio to your MIC input, you should use an attenuator circuit like the one shown below. This circuit must attenuate the TX audio from the DK9RR to an acceptable level for the mic amplifier. You may include this circuit into the chassis of the vuC plug.



All received and decoded frames are transmitted to the PC. You should see them as partly unreadable ASCII text in your terminal program. If this is true, your TNC works fine.

This will end the initial testing of your **MCB152** micro controller board used as a TNC. From now on you can start experimenting and installing other software parts like KISS firmware to be used in standard AX.25 software (JNOS, TOP, GP enz...) or other firmware so you can use Netscape or any other browsers to do TCP/IP over the AX.25 packet network.

D. Epilogue

This includes the primary documentation for our **MCB152** board. You will find extra information on the Packet network, with the creators or on the included CD-ROM.

You may visit us every friday evening at 20h00 in our local clubhouse.

Oud Atheneum, first floor. UBA-RST
Zoutstraat 43 B-3800 Sint-Truiden, Belgium.

2m 145.275 MHz

Or via tel	++32-11-673480 (tel+fax)	Machiels Walter ON4AWM → Hardware.
	++32-11-674426	Elen Joachim ON1DDS → Software.

We hope to see you soon on packet. Send us a mail if you have any problem or question, we are glad to help you.

Best 73, Joachim & Walter. http://GALLERY.UUNET.BE/Gert.Leunen

Software:
Elen Joachim / Nulens Vicky
Tiensesteenweg 78 / 402
3800 Sint-Truiden
Tel: ++32 -11 - 67 44 26

Packet: on1dds@on4baf.ampr.org
on1dds@on4baf.lm.bel.eu
E-mail: j_elen@hotmail.com

Hardware:
Machiels Walter/ Cremers Lydia
Bloesemlaan 9
3800 Sint-Truiden
Tel: ++32 - 11 - 67 34 80
Fax: ++32 - 11 - 67 34 80
Packet: on4awm@on4baf.ampr.org
on4awm@on4baf.lm.bel.eu
E-mail:

Baycom group:
Baycom
Bert-Brecht-Weg 28
D-30890 Barsinghausen
Tel: ++49 - 5105 - 58 50 50
Fax: ++49 - 5105 - 58 50 60

Ref:	DK9RR kit 4540	109 DM (price 1997)
	DK9RR complete 4640	139 DM (price 1997)

Word of thanks
We wish to thank everyone for their support and cooperation, especially those listed here:

- **R&D International** - Naamsesteenweg 146, B-3800 Sint-Truiden.
- **UBA-RST,** Radioamateurs sectie Sint-Truiden.
- **Baycom Group**, Bert-Brecht-Weg 28, D-30890 Barsinghausen.