

FJRX-2FP

Software to make DF1FO's ARDF
FJRX 2 Receiver suitable for any style
of Fox Hunting.



PA2PIM

W.J. (Pim) Niessen

Status : Released
Author : PA2PIM, W.J. (Pim) Niessen
Approved :
Date : May 27, 2008
Distribution : Public
Filename : RP001_FJRX-2FP_V2.0.doc

Initials
Initials

■ Document history

Version	Date	Author	Changes
1.0	2008-01-28	PA2PIM, W.J. (Pim) Niessen	Initial release
1.1	2008-01-29		Added appendices paragraph, several minor textual changes
1.2	2008-02-03	PA2PIM, W.J. (Pim) Niessen	Updated file names in Distribution paragraph
1.3	2008-02-05	PA2PIM, W.J. (Pim Niessen)	Changed sequence of programming Flash memory and EEPROM
1.4	2008-02-12	PA2PIM, W.J. (Pim Niessen)	Added picture to cover page
1.5	2008-02-25	PA2PIM, W.J. (Pim Niessen)	Added LCD layout explanation
1.7	2008-03-25	PA2PIM, W.J. (Pim Niessen)	Added Mem prog sound and Attenuator cal in 5 dB paragraphs.
1.8	2008-03-26	PA2PIM, W.J. (Pim Niessen)	Corrected typos, made file names consistent
1.9	2008-05-14	PA2PIM, W.J. (Pim Niessen)	Jutberg changes
2.0	2008-05-27	PA2PIM, W.J. (Pim Niessen)	Completed documenting Jutberg changes. Updated wish list

■ Contents

1. Introduction	4
2. FJRX-2FP changed functionality	4
2.1 FP Hunting modes added	5
2.2 Dynamic Frequency Step size	5
2.3 AGC	5
2.4 RxA and RxB	5
2.5 Auto RxA/B switching	5
2.6 Display layout	6
2.7 Selecting Hunting modes	7
2.8 Long Press defined for the Encoder Pushbutton	7
2.9 Acoustic S-meter (AcSm)	7
2.10 Attenuator calibration in 5 dB steps.	7
2.11 Save Calibration	7
2.12 EEPROM init	8
2.13 Menus in 3 different languages	8
2.14 LCD only updated when information has changed.	8
2.15 Memory programming.	8
3. Loading software	8
3.1 Programming the Fuses	9
3.2 Programming the Flash memory	9
3.3 Programming the EEPROM	9
3.4 Calibrating the receiver	9
3.5 MEGA8 programming availability	9
4. Wish List	9
4.1 Checking battery voltage 10 sec after the start of Fox 1	10
4.2 Anti jitter mechanism when changing memories	10
4.3 Two different sound schemes	10
4.4 Adjustable dwell time for the AcSm.	10
4.5 Tuning frequency wrap around.	10
4.6 Implementing the wishes	10
5. Unwanted features, a.c.k.a. bugs	10
6. Software description	10
7. Distribution	11
8. Licensing	12

■ 1. Introduction

In the first half of 2007 Nick Roethe, DF1FO, published his FJRX 2 ARDF Fox hunting receiver. He also made a high quality kit available to build one. (<http://www.mydarc.de/df1fo>).

The FJRX 2 with the software from DF1FO is a 2m ARDF Fox hunting receiver under microprocessor control. It's more or less a standard double super Fox hunting receiver. It has an I²C controlled synthesizer as the first oscillator and a TCA440 as the 2nd mixer and IF amplifier. An Atmel AVR MEGA8 micro processor is used as the brains of the receiver.

The software from DF1FO for the FJRX 2 is fully geared towards ARDF foxhunting with its prescribed behaviour of foxes, 2 hour hunt limit, etc. For a description of the functionality please visit (<http://www.mydarc.de/df1fo>).

The fox hunting I like is not the ARDF style but the ones where there are, within reason, no rules. Among others, these style of hunts are organized in central Holland (www.pi4rcg.nl) or at the radio camping week of the VRZA (www.radiokampweek.nl)

When going to such a hunt you don't know:

- The number of Foxes
- Power of the Foxes
- Frequencies of the Foxes
- Modulation of the Foxes
- Whether or not the Foxes are on a timer
- Power of the Foxes
- Will there be Pancakes or Sausage Rolls at the Finish
- Etc.

The receiver requirements for such a Fox Hunt differ from the requirements for an ARDF Fox Hunt receiver. You should be able to tune the band rapidly and have Automatic Gain Control. A few easily accessible memories would be nice.

Since I liked the receiver design I adapted the FJRX 2 software to accommodate my required extra functionality on top of the existing ARDF functionality. All additions are done in software so no hardware changes are required to the FJRX 2 receiver.

This document is the description of the resulting software which I named FJRX-2FP.

By the way, the FP stands for Flying Pig club (<http://fpgrp.net/>).

I found their motto quite appropriate as a project name as well as the envisioned usage of the FJRX-2FP receiver.

"No Dues, No Rules, Just Fun - and if we don't like it we fix it!"

The FJRX-2FP software is an extension of the software developed by DF1FO V4.2. Also ideas, and code from Thomas Schiller, DC7GB, were used. (<http://www.mydarc.de/dc7gb/DF1FO-Peiler.html>).

■ 2. FJRX-2FP changed functionality

The following functionality was added or changed compared to DF1FO's V4.2 or DC7GB V3.3+ of the software for the FJRX 2.

2.1 FP Hunting modes added

FP Hunting mode.

In this hunting mode the encoder knob adjusts Frequency or RF Gain. A short press on the encoder knob switches between the two parameters to adjust.

FP Memory Hunting Mode.

This is a variation of DF1FO's 1x2<>3 ARDF mode. In the FP implementation the receiver will listen to (max.) 10 different timed foxes transmitting at any of (max.) 5 frequencies stored in a memory. Turning the encoder knob while pressing the shaft will select for the current Fox any of the 5 memories, (Turning left) or the homing transmitter stored in memory 6 (Turning right). It's still possible to change the Frequency or RF Gain as described in 2.2 below. Operation is a bit complicated so the usability of this Hunting mode will have to be proven one day.

2.2 Dynamic Frequency Step size

Frequency step size is 1.25 or 10 kHz. per encoder click in DF1FO's software. In the FJRX-2FP software it depends on the turning speed of the encoder knob. If you turn slowly, less than 1 click per 500 ms, the step size is 1.25 kHz/click. If you turn faster a weighting algorithm is used on the number of encoder clicks over the last expired 500 ms. If you turn the tuning knob fast enough the Rx frequency will change 50 kHz/step.

2.3 AGC

AGC was added to the two original gain modes defined by DF1FO, Manual Gain Control (MGC) and Auto Gain reduction. A long press on the switch pushbutton cycles through the 3 different Gain Modes in a FP Hunting mode. AGC is turned on in all Hunting modes when changing the frequencies stored in the memories. The Gain mode active prior to activating a menu is restored when returning to a hunting mode. To stay compatible with DF1FO's look and feel AGC is not allowed in any ARDF Hunting mode.

Every time the attenuator changes 5 dB a sound will be played and some users find this distracting. The sound for attenuator changes is shortened from 600 ms to 200 ms. If you still find this distracting, you can disable attenuator sounds when AGC is active in the Calibration menu.

2.4 RxA and RxB

Two receivers are implemented in the software named RxA and RxB. When in any of the FP hunting modes two or more consecutive short presses on the pushbutton switch will switch to the other receiver. RxA is the default receiver. Frequency, Gain Mode, Gain setting etc. are saved between RxA and RxB. Audible Morse code A or B is generated to indicate to the operator which receiver is activated.

2.5 Auto RxA/B switching

When in the FP Hunting mode Auto RxA/B switching can be enabled. RxA and RxB activation time is adjusted under the Settings menu. This feature is useful to determine when the homing beacon is turned on, keep an ear on the next Fox, etc.

2.6 Display layout

Frequency display

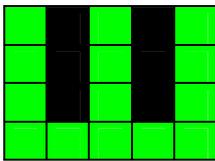
4 characters are used to display the Rx frequency down to a kHz. accuracy. The leading 100 and 10 MHz. digits are not displayed since they do not change.

Gain or Distance display

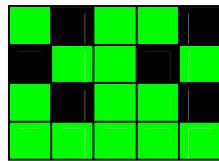
Nothing was changed compared with DF1FO's version of the software. 3 characters are used on the top line of the LCD to display the RF attenuator setting or the estimated distance to the Fox. For < 100 dB attenuation a compressed dB symbol is displayed. If the distance is displayed an m or km is displayed if space permits.

Gain Mode indication

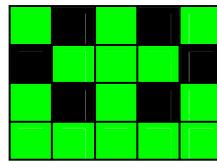
The frequency and Gain or Distance characters are separated by a single character which is programmed as needed.



Manual Gain Control



DF1FO's Auto Gain Reduction



Automatic Gain Control

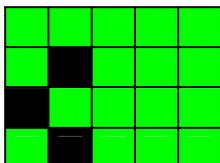
The 5*8¹ matrix of a character is split in two parts. The upper 3 rows of the matrix are used to display the Gain mode as shown on the left. The 4th row is reserved for separation with the memory symbol in the lower 4 rows of the matrix.

Memory indication

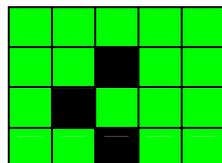
The bottom 4 rows of the same programmable character are used to display the selected memory. I used DC7GB ideas to use the Roman numerals | through V|.



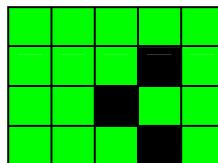
On the left is the symbol for Memory 3. In FP hunting modes the Rx frequency can easily be changed. As long as the VFO frequency differs less then 2.5 kHz from a frequency stored in a memory a roman numeral is displayed. As soon as the Rx frequency deviates by more then 2.5 kHz from a frequency stored in a memory a graphical character is displayed. Which character is used is determined by the sign and smallest absolute difference between the VFO frequency and any of the 6 memories. Below are the 4 symbols used when the VFO frequency is above a frequency stored in one of the memories.



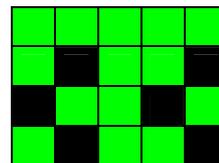
Nearest Memory frequency is ≤ 10 kHz. below the VFO frequency



Nearest Memory frequency is between 11.25 and 20 kHz. below the VFO frequency



Nearest Memory frequency is between 21.25 and 30 kHz. below the VFO frequency



Nearest Memory frequency is ≥ 30 kHz. below the VFO frequency

When increasing the VFO frequency and the next Memory frequency is approached, the direction of the arrows

is reversed. When turning the band the overall effect, is that you see an arrow moving across the lower part of a character on the LCD. The closer you get to a memory frequency the closer the arrow moves to the edge of the character. For frequencies differences larger then 30 kHz. two arrows are displayed.

¹ The LCD uses a 5*7 character matrix and an additional 8th row below it for the cursor. Character Generator Ram space is therefore 5*8. The cursor line commonly is a dash but it can be whatever you want to display on a single line.

In a FP Hunting mode the encoder knob adjusts either attenuation or frequency. A short press on the encoder knob toggles between adjusting these parameters. The parameter which gets adjusted is on the right side of the display, i.e. closest to the encoder knob.

2.7 Selecting Hunting modes

There are 3 ARDF and 2 FP hunting modes in the FJRX-2FP. DF1FO used some of the memories to select the 3 different ARDF hunting modes. In the FJRX-2FP the 5 different Hunting modes are selectable under the Settings menu. When changing hunting mode the receiver switches to RxA. The RxA parameters are also copied to RxB when changing hunting modes.

2.8 Long Press defined for the Encoder Pushbutton

In FP Hunting modes the encoder pushbutton had to be used to:

- Toggle between adjusting Frequency or RF Gain
- Toggling the Acoustic S-meter (AcSm) On and Off

Long press functionality, as implemented for the pushbutton switch, was added to the Encoder Pushbutton. A short press toggles Frequency or RF Gain adjustment. A long press toggles the AcSm On and off.

2.9 Acoustic S-meter (AcSm)

- The frequency of the AcSm in DF1FO's code is between 100 and 3400 Hz. I found the 3400 Hz a bit too high and was looking for freeing up program space. In a spreadsheet a logarithmic, exponential and linear curve was programmed. In the final curve the weighting of each type of curve as well as the minimum and maximum audio frequency can be set. The spreadsheet generates the program words to implement a lookup table from Received Signal Strength to Audio Frequency. The lookup table is programmed in EEPROM and copied to Ram during the start of the Receiver. The end result is a flexible tone generator using less program space.
- The maximum AcSm automatic switch off level was raised from 3/8 to 7/8 of full scale. I found the original maximum of 3/8 just a bit too low. In practise I usually have it set to 4/8. Settings above 5/8 are pretty useless but no code space was left to limit it to that value.
- The AcSM dwell time or "Nachlaufzeit" was changed from 3s to 1.2s. Together with the automatic switch-off level of the AcSm level this was found to be useful when trying to locate this tiny fox in camouflage colours fox hidden in these stingy shrubs.

2.10 Attenuator calibration in 5 dB steps.

DF1FO's software let you calibrate the Attenuator in 10 dB steps. Interpolation is used to let the attenuator operate in 5 dB steps. The attenuator is now calibrated in 5 dB steps which is using less code overall. In theory the attenuator is more accurate since no interpolation is used anymore but it's questionable if this is relevant. If you have already calibrated the receiver a worksheet in the spreadsheet accompanying the distribution does the interpolation for you. Take the calculated values and enter them in the receiver in Cal mode. If you still have to calibrate the receiver do so at a 5 dB interval. It only takes slightly longer than the original 10 dB interval and you only have to do it once

2.11 Save Calibration

V4.x software of DF1FO compares current receiver settings with the settings stored in EEPROM every second. With that feature implemented there is no need to still have a "Calibration Save" option in the

Calibration menu. Calibration settings will be saved within one second. The menu option and underlying code was removed.

2.12 EEPROM init

During start up the FJRX 2 software checks if the EEPROM is programmed. If not, it copies parameters stored in program space to the EEPROM. Also there was an option in the Calibration menu to force a reload of the EEPROM. Because program space was needed this functionality and menu option is removed.

In order to get the receiver operational a default set of parameters has to be loaded in EEPROM. The default parameters, together with the AcSm lookup table, are defined in the source file. The compiler generates the required xxx.eep file.

2.13 Menus in 3 different languages

In the FJRX-2FP the entire menu texts are in English and no other language can be selected. If you want menus in another language translate the current menus, recompile the code and reprogram the MEGA8. I removed the language selection option from the FJRX 2 software to gain the needed program space.

2.14 LCD only updated when information has changed.

Birdies are always an issue when a Microprocessor is operating in close vicinity of a sensitive receiver. And indeed a few can be heard at full RF Gain. The original software from DF1FO updates the LCD every 100 ms. Code was added to only update the LCD when the information to display is different from the information already on the LCD. This avoids putting spurious generating high speed data on the LCD data lines when it is not really needed. Thomas, DC7GB, reported that this change made quite a difference to the spurious level.

2.15 Memory programming.

In the original Memory programming mode the receiver frequency switches to the stored frequency if you select another memory. During an ARDF hunt you know which frequency or frequencies are in use so this is not an issue. In a FP style of hunt I found this effect to be an issue. I lost the frequency of the fox I had just tuned the receiver to. So the behaviour is changed. You tune the receiver to the selected frequency. Then flip up the switch and press the encoder to get to memory programming mode. In the lower right of the display the selected memory number is indicated. Optional you can press and turn the encoder to select another memory. The receiver frequency will not change. Finally a long press on the encoder programs the indicated memory

■ 3. Loading software

For the FJRX-2FP you need to (re)program the Flash memory **and** the EEPROM memory of the MEGA8

Make sure you program the Flash memory before you program the EEPROM!²

² If you program the EEPROM first it will be overwritten by the original software from DF1FO which is still in the Flash memory.

3.1 Programming the Fuses

- If not already done so program the fuses as listed in the FJRX-2FP-xx.ASM file using any ATMEL AVR programmer. When reprogramming a Mega8 from a functional DF1FO receiver skip to paragraph 3.2

```
EESAVE      =      0
CKSEL       =      1111
SUT         =      11
BODEN       =      0
BODLVL      =      0
```

The resulting Hex-Pattern = D13F

3.2 Programming the Flash memory

- If you have calibrated the FJRX 2 receiver before go back in calibration mode and write down the Attenuator, Frequency correction, Battery Voltage, Battery Warning settings and Calibration factors.
- Program the FJRX-2FP-xx.hex file in the Flash memory of the MEGA8 using any ATMEL AVR programmer.

3.3 Programming the EEPROM

- Program the FJRX-2FP-xx.eep file in the EEPROM using any ATMEL AVR programmer.

3.4 Calibrating the receiver

- If you had previously calibrated the FJRX 2 go into Calibration Mode:
 - Re-enter the noted down values of paragraph 3.3 for all parameters except the Attenuator calibration.
 - Take the spreadsheet FJRX-2FP.XLS and open the Attn. Calib. Transfer worksheet.
 - Fill in the noted down Attenuator Calibration values from paragraph 3.3 in the yellow column C9...C21.
 - Enter the interpolated values from E9...E33 as the calibration values for the Attenuator.

Your FJRX-2FP is operational again with added functionality.

- If you had not yet calibrated the receiver follow the instructions as provided by DF1FO in his fjr233.doc document at (<http://www.mydarc.de/df1fo>).

You can now go to any Fox hunt, ARDF style or otherwise!

3.5 MEGA8 programming availability

If you do not have any means at all to (re)program your MEGA8 please contact me at pa2pim at amsat dot org.

■ 4. Wish List

The current version of the software provides you with a very well useable receiver for all styles of fox hunting. But there is always something more on the wish list. And, compared to the latest version published by DF1FO some features of his software are also not implemented yet.

Here is the wish list.

4.1 Checking battery voltage 10 sec after the start of Fox 1

In this FJRX-2FP software version the battery voltage is checked every elapsed 2 min's of the stopwatch. A battery low alarm sound can coincide with the sounds accompanying changing to another memory or fox. Shifting the alarm 10s in time will alleviate this issue. (Implemented in DF1FO's V4.2)

4.2 Anti jitter mechanism when changing memories

The encoder reacts to every action. Handling the receiver with gloves or in the cold might result in unintended actions. (Implemented in DF1FO's V4.2)

4.3 Two different sound schemes

You can operate this receiver blindfolded. All kind of sounds will tell you what's going on or what you are doing without looking at the display. This is perfect for the handicap hunt, where you really have to wear a blindfold! In some other style of hunts the informative sounds can be distracting because they occupy time and you can not hear the fox. A brief beep in response to an action could be enough. Implementing two selectable different sound schemes will give you a choice.

4.4 Adjustable dwell time for the AcSm.

Currently the AcSm dwell time or "Nachlaufzeit" is fixed at 1.2s. An adjustable setting between 0.1 and 5.0 s would allow it to be adjusted to personal taste.

4.5 Tuning frequency wrap around.

When walking and searching for the next fox in the band you can not always look at the display to check the current frequency. When tuning and reaching the end of the 2m band it is helpful if you heard a sound and the Rx frequency would jump back to the other side of the band. This feature must be selectable by the user.

4.6 Implementing the wishes

I have optimized the code for flash memory usage just about to the limit. Squeezing any more bytes out of it to implement the above changes will take an inappropriate amount of time. And, it's still questionable if I can free up enough bytes to implement the above wishes.

I therefore have used an ATMEL MEGA168 as a pin compatible replacement for the MEGA8. It's only €2,- more expensive but has twice as much flash memory as the MEGA8.

■ 5. Unwanted features, a.c.k.a. bugs

No Known Bugs as of 2008-03-26. Let me know if you find any.

■ 6. Software description

The general principle of the software for the FJRX-2FP was not changed compared to the FJRX 2 software from DF1FO or DC7GB.

Here are the highlights of the software design.

- Timer 0 generates an interrupt every 320 μ s.
- In the Interrupt Service Routine for Timer 0 the Fox Timer, Stopwatch and various other time related flags are maintained. Also the debounce routines for the switch and the encoder are executed.
- The continuous looping Main routine monitors the encoder and switch status bytes and performs the appropriate actions. Which action needs to be performed is determined by a state machine.
- The implementation of the state machine is one of the major differences between the FJRX 2 and FJRX-2 FP software. The six possible encoder actions form the columns in a two dimensional matrix. The 30+ different operating modes of the receiver are the rows in the same matrix.
- At every Column/Row junction of this state machine matrix is the address of a routine which needs to be executed in this operational mode of the receiver when this encoder action occurs.
- The state machine principle overall is code efficient and very flexible. The state machine overview and changing rules are documented in the spreadsheet accompanying the FJRX-2FP software.
- Derived from the 320 μ s interrupt every 100 ms, 1s etc. the display is updated, audio frequency of the Tone generator is updated, RxA/B is toggled or the Battery Voltage is checked and the appropriate action is taken.
- The additional extra FP functionality required code space. This was obtained by optimizing the existing software. The following principles were used:
 - Macros usage was scrutinized for unneeded push and pop instructions.
 - Two registers are loaded with a 0 or 1 during receiver start-up. This avoids using the `ldi Rx, 0` and `ldi Rx, 1` instruction when setting the multitude of status flags and thus saves code space.
 - Quite a number of compare and branch instructions were changed to indexed lookup tables since they are more code efficient. The ultimate example is the state machine in the Main loop. The tables of the state machine require 415 bytes of code space. Overall it's still more code efficient than a whole load of compare and branch instructions. More than 100 bytes of code space were gained. Apart from that, changing functionality requirements in a state machine is a breeze. Instructions on how to go about that are in the spreadsheet accompanying the FJRX-2FP software.

■ 7. Distribution

The FJRX-2FP software is distributed in the FJRX-2FP_Xx.ZIP file which has the following contents.

File Name	Description
FJRX-2FP_17.ASM	The Assembly source code for the ATMEL AVR Studio 4 compiler. Xx denote the major and minor revision number
FJRX-2FP_macros.ASM	The Macro's called from the above file
FJRX-2FP-17.hex	The resulting code to be programmed in the MEGA8's Flash memory
FJRX-2FP-17.eep	The data to be programmed in the EEPROM of the MEGA8
FJRX-2FP_1.7.XLS	A spreadsheet with various info, see below
RP001_FJRX-2FP_V2.0_3.DOC	This file.

File Name	Description
BattChargerMax712.pdf	A battery NiMh battery charger based on the Maxim MAX712 chip. I build this one to charge the 1800 mAh battery of the receiver. It's powered by a laptop charger from a fleemarket and charges the battery in about 4 hours. (When asked Maxim provides free samples of the chip.)

The FJRX-2FP_1.7.XLS spreadsheet contains 10 worksheets:

Worksheet name	Description
Operating Short Form	A compact overview of the FJRX-2FP operation.
State Machine	Detailed overview of the State Machine in the Main Loop. The names of the routines to execute in every state and the strings to display are listed. This worksheet was used when writing the code to verify if everything was implemented.
Changing State Machine	Instructions on how to change functions of the state machine
Acoustic S meter Calculation	The ACSM lookup table is build here from 3 different curves and selectable minimum and maximum frequencies. The original DF1FO ACSM curve is also available for comparison.
ACSM Code	The result from the previous worksheet. Copy/Paste it in the source code to generate the required lines of code.
Mem_Symbol_Def	An overview of the symbols used to display the distance to the closest memory
Synthesizer Step Size (exp)	Calculation of the dynamic step size parameters for the VFO function. An exponential curve is used. The curve in the next worksheet provides a less "aggressive" response
Synthesizer Step Size (lin)	Calculation of the dynamic step size parameters for the VFO function. A linear curve is used. This curve is implemented in the FJRX-2FP_17 version of the software. Together with a maximum step size of 50 kHz. per encoder click it provides a smooth working tuning response.
Atten. Calib. Transfer	If you have previously calibrated your receiver at 10 dB's interval using DF1FO software fill in the Attenuator calibration values in the yellow column. Take the data out of the orange column and enter it in Cal mode in the receiver
PA2PIM Att Values	The Attenuator calibration data for two chips. An original Siemens TCA440 (saved in the junk box for around 25 years) and one produced by Halbleiterwerk Frankfurt/Oder supplied with the kit from DF1FO. Just an example.

This zip file is also available via http://home.planet.nl/~niess153/Fox_Hunting.htm

■ 8. Licensing

This software is licensed under the **BEER-WARE** License.
The following strictly applies:

- It is provided solely for non commercial use.

- **As long as you retain this notice you can do whatever you want with this software provided you do not make any financial gain out of it.**
- If you use this software notify me by sending me an E-mail or a QSL card.
- If there are any remarks or wishes about this software, speak up and let me know. My spare time is limited but some wishes are easy to implement. Miracles might take a bit longer but I will try if convinced of the added value of your wish.
- If we meet some day, and you think this software is worth it, you can buy me a beer in return for the countless hours I spend on writing and testing this software.

W.J. (Pim) Niessen, PA2PIM

PA2Pim at Amsat dot org

www.planet.nl/~niess153