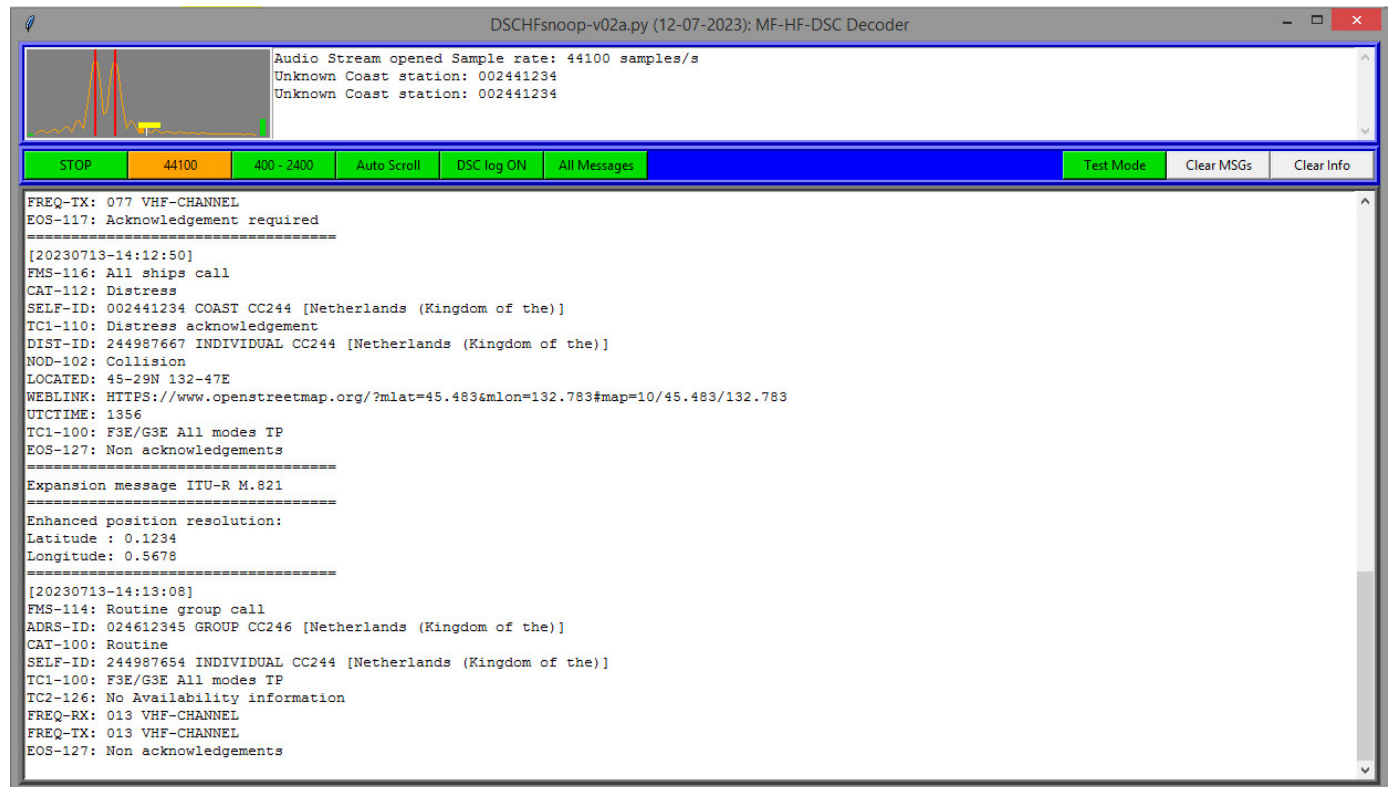


## RECEPTION OF DIGITAL SELECTIVE CALLING ON MF-HF (2023)

### What is Digital Selective Calling or DSC

Digital selective calling or DSC is a standard for transmitting predefined digital messages via the medium-frequency (MF), high-frequency (HF) and very-high-frequency (VHF) maritime radio systems. This program can decode DSC signals on MF and HF. Most used frequencies are 8414.5 kHz and 2187.5 kHz.



*The Python program for the reception of Digital Selective Calling on MF-HF.*

### Simple

This program can be executed with a Raspberry Pi Version 4. It can be used with Windows and Linux!

Connect your PC to the SSB shortwave receiver. Download the file -23hfdsource.zip- from this website and unzip all files in one directory. Execute the Python program DSCsnoop-v02a.py, if you do not know how, that is explained later.

Press the -START- button, select the desired audio input and the DSC data will be displayed.

### Buttons:

#### -START-

Press this button to start the reception or to stop the reception.

#### -44100-

This is the sample rate of your audio device. If you have a slow PC, then you can also select 11025 samples/sec., that will reduce the processing power a lot! But then you need to use an audio device that supports this sample rate or you have to use downsampling software like Pulse Audio for Linux.

#### -400 - 2400-

This is the audio range that is scanned for usable signals. Normally it is set to 400 - 2400 Hz. But if you have some unwanted signals in the spectrum you can select other ranges. For 1500 Hz center frequency you can select 1000 - 2000 Hz or 1200 - 1800 Hz. For a 1500 Hz center frequency, you have to tune the SSB receiver 1500 Hz lower than the DSC frequency. So 8413 kHz for a 8414.5 kHz DSC frequency and 2186 kHz for a DSC frequency of 2187.5 kHz. For a 1700 Hz center frequency you can select 1400 - 2000 Hz. Selection is done by pressing the button, the range is given by the button text.

#### -Auto Scroll-

Normally activated, the last message is shown on the display.

#### -DSC log ON-

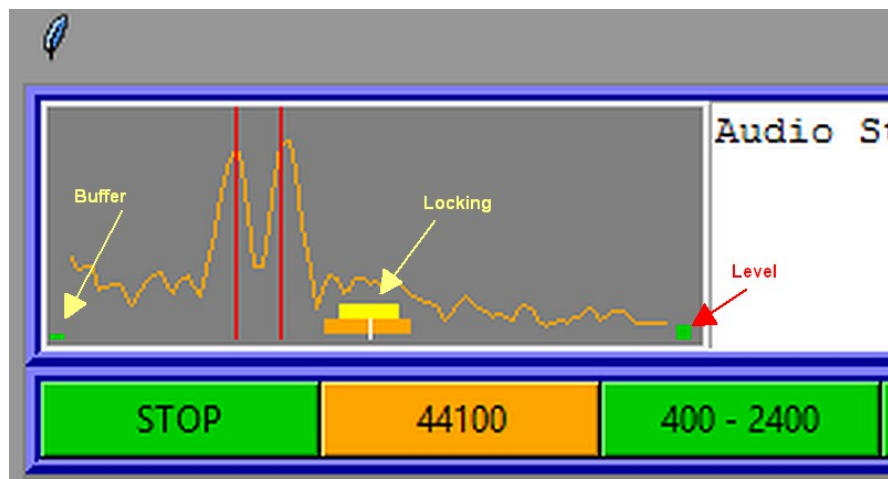
If OFF, the saving of the DSC data to the files is stopped.

#### -All Messages-

Almost all messages are test messages. If the reception of All Messages is disabled, the test messages are not decoded anymore.

#### -Test Mode-

Normally switched off. You can select Level 1 debug messages (orange button) or Level 2 debug messages (red button). But Level 2 will display much useless information.



The spectrum display

### The spectrum display

The selected spectrum is displayed. The vertical red lines are the shift frequencies of the DSC signal.

At the left there is a vertical green line. It displays the audio buffer size. If your PC is too slow, it will go up and become red colored.

At the right there is also a vertical green line. It displays the audio level and will become red colored if the audio level is too high.

When a phasing signal is detected, a horizontal orange line is displayed at the bottom. If the Signal to Noise is good, the line will be small, when you have a noisy signal, it will be a wide line. It moves a little around the white marker, that is the locking routine.

And just above the orange line, there is a yellow line, that moves left - right, depending of the bit synchronization being forwards or backwards.

```
#####
# Configuration
SAMPLERate = 44100          # Sample rate of soundcard, 11025 or 44100 preferred
FTPtime = 30                # FTP interval time in minutes, integer of 60!!!
TITLE1 = "DSC ALL MESSAGES" # Title of the file
FileName1 = "dscall.txt"    # File name for all messages
DIR1 = "./DSCall/"          # Directory of the file
TITLE2 = "DSC WITHOUT TEST MESSAGES" # Title of the file
FileName2 = "dscminustest.txt" # File name for messages except test messages
DIR2 = "./DSCminustest/"    # Directory of the file
TITLE3 = "DSC SPECIAL MESSAGES" # Title of the file
FileName3 = "dscspecial.txt" # File name for special messages like for example Distress or a special MMSI
DIR3 = "./DSCspecial/"     # Directory of the file
DIRday = "./DSCday/"        # Directory for the daily files
DIRcoast = "./DSCcoast/"    # Directory for the mmsi coast station files
DIRship = "./DSCship/"      # Directory for the mmsi ship station files
DayOfMonth = False          # If False, Day Of Week is selected for Day Name (0=Sunday) instead of Day Of Month
DBcoast = 2                 # Database Coast: 0=none; 1=MultiPSK; 2=YADD
DBship = 2                  # Database Ship: 0=none; 1=MultiPSK; 2=YADD
FTPfilename = "FTPuploads.txt"
FREQband = 0                # 0, 1, 2, 3 Frequency band
SAX = 200                   # Width of the spectrum screen
SAY = 80                    # Height of the spectrum screen
SAMargin = 10               # Margin left - right for audio buffer and level of spectrum screen
ButtonWidth = 12            # Width of the buttons
HLINE = "-----"          # Message separation line
```

Configuration

### Configuration

In the first lines of the Python program you can find various settings. Perhaps that you want to change some of them. Especially the DBcoast and DBship might be interesting for you if you want to choose a data base for Coast and Ship stations. It are not my data bases. They are from the excellent DSC decoder programs MultiPSK (can decode much more) and the great YADD program.

Data bases can be found at

GM4SLV : MF/HF DSC Database:

[http://www.yaddnet.org/pages/php/test/mmsi\\_coast.php](http://www.yaddnet.org/pages/php/test/mmsi_coast.php)

[http://www.yaddnet.org/pages/php/mmsi\\_shipname.php](http://www.yaddnet.org/pages/php/mmsi_shipname.php)

Rename the data bases to YADDcoast.txt and YADDship.txt.

For more information about the reception of DSC signals, visit the DSC group website: <https://groups.io/g/dsc-list>

Name	Date modified	Type	Size
DSCall	8-7-2023 20:54	File folder	
DSCcoast	8-7-2023 20:54	File folder	
DSCday	8-7-2023 20:54	File folder	
DSCminustest	8-7-2023 20:54	File folder	
DSCship	8-7-2023 20:54	File folder	
DSCspecial	8-7-2023 20:54	File folder	
DSCHFnoop-v02a.py	8-7-2023 20:35	PY File	108 KB
FTPnoop-v01a.py	17-6-2023 14:58	PY File	7 KB
FTPuploads.txt	29-6-2023 15:30	Text Document	1 KB
MultiPSKcoast.txt	8-7-2023 17:43	Text Document	43 KB
MultiPSKship.txt	8-7-2023 17:43	Text Document	2.329 KB
YADDcoast.txt	8-7-2023 19:33	Text Document	28 KB
YADDship.txt	7-7-2023 21:37	Text Document	2.121 KB

The file structure

**File structure**

And of course we want to save the received data in files! There are directories DSCall, DSCminustest and DSCspecial. In the first directory, there is a file with all the DSC messages and that are really many during a day! The second directory contains only the DSC signals without the test messages and that are not so many. The third directory shows only special messages, only distress messages at the moment. But you can edit the Python program so that for example a special MMSI is saved in this directory!

The directory DSCday contains the daily files. They are updated every half hour and can for example be uploaded to your DSC website. You can copy a MMSI and look at the internet what kind of ship it is and where it is.

The directories DSCcoast and DSCship contain files with the Coast and Ship MMSI and how many times they have been logged in a month. The first line is January, the last line December.

And in the latest version, the directory DSCpos is added, it contains the monthly files with received ship positions.

```

1  #!/bin/sh
2
3  # Start DSC
4  cd /home/pa2ohh/P3-DSCHFsnop/Python3/
5  lxterminal -l -e 'python3 DSCHFsnop-v02a.py ; /bin/sh ' &
6  lxterminal -l -e 'python3 FTPsnop-v01a.py ; /bin/sh ' &

```

*The script that is used to start DSCHFsnop on a Raspberry Pi*

**Script to start DSCHFsnop.py on a Raspberry Pi**

Make the script on the desktop and make it executable. You have to edit the script to make it suitable for your own PC of course!

Execute the script and the program starts. But... you can see that also a second program is started here! It is FTPsnop.py! This can be used to upload files to your website with FTP. Click the link here below to see how that works:

[DSC signals on 8414.5 kHz](#)

The text files are uploaded and also converted to .htm files. But you cannot use DSCHFsnop.py straight away, you have to edit it to set your own FTP login codes.

**Downsampling**

You can use the program with the Raspberry Pi version 4. The CPU percentage is approximately 25%, so it should also be possible to execute it with a 3x slower Raspberry Pi version 3. But it might be that you have to downsample the audio stream. Before pressing Start, press the orange "44100" button and the sample rate will be reduced to 11025. Then the CPU percentage is lower than 10% on my Raspberry Pi version 4 and it should certainly be possible to execute it with a Raspberry Pi version 3. My USB sound device supports a sample rate of 11025. It is a Behringer type UFO202 and even supports a 8000 Hz sampling rate. But if your device does not support it, you can do the downsampling with Pulseaudio. Then you have to install the "Pulse" audio platform on the Raspberry Pi. Pulse is then selected as audio input device and is used for the downsampling of the audio stream. Type the following commands in a terminal window:

```

sudo apt-get install pulseaudio
sudo apt-get install pavucontrol

```

**Error handling**

Every character is transmitted twice, once in the DX position and once 5 characters later in the RX position. And every byte of 10 bits is specially coded. If one bit or more bits of the same "Y" or "B" value is disturbed, it will become an invalid character. This decoder program accepts wrong RX data, but only valid characters. To eliminate problems with fading, two zero crossings are combined ("YB" and "BY" crossings) for the bit synchronisation.

[Click this link to download the ZIP file with the Python programs and other scripts](#)

**FTPsnop.py**

```

File Edit Format Run Options Window Help
# FTPsnop-v01a.py(w) (17-06-2023)
# For uploading of the data files of XXXsnop.py
# For Python version 3
# Made by Onno Hoekstra (pa2ohh)

# External modules: None

import os
import time
import ftplib

from time import gmtime, strftime

# ===== !!!Settings that have to be changed by the user!!! =====
FTPHost = "ftp.qsl.net"
FTPuser = "pa2ohh"
FTPdir = "./dsc"
FTPpassword = "password"

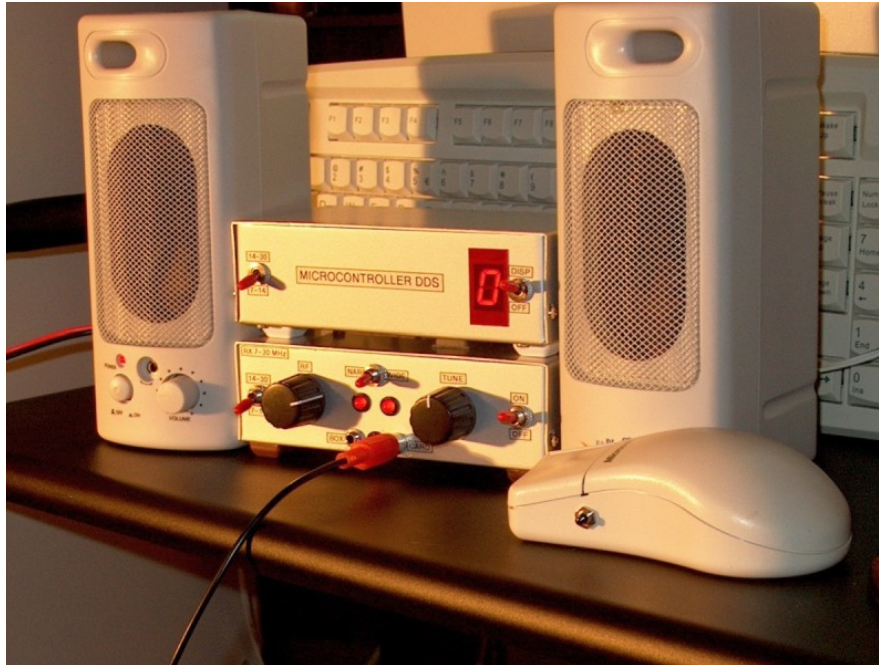
TXTupload = True # Do not upload the text files if False
HTMLupload = True # Do not upload the html files if False

```

*You have to edit the first lines with login codes of FTPsnop.py.*

**FTPsnop.py**

FTPsnop.py is started after DSCHFsnop.py. Every 30 minutes, DSCHFsnop.py updates the daily text files with DSC messages in the DSCday directory. It saves the file names in the file FTPuploads.txt. FTPsnop.py checks if this file exists. If it exists, it will upload the files to your FTP site. If you have set HTMLupload=True, then the text files are converted to HTML files and uploaded.



*Simple direct conversion receiver that was used to receive the DSC messages on 8414.5 kHz*

---

## SOFTWARE

Before you are using this program, you have to install Python. That is very simple. But read first something about Python by clicking the following link:

[WHAT IS PYTHON AND HOW DO YOU INSTALL PYTHON](#)

As the source code of Python is written in ASCII, it is very simple to modify the program to you own requirements. Think for example about the size of the screen, the colors etc.

**Required Python version:**

- Python version 3

**Required external modules (site-packages for the correct Python version!):**

- Numpy
- pyaudio

---

[Click this link to download the ZIP file with the Python programs and other scripts](#)

---

[Index PA2OHH](#)