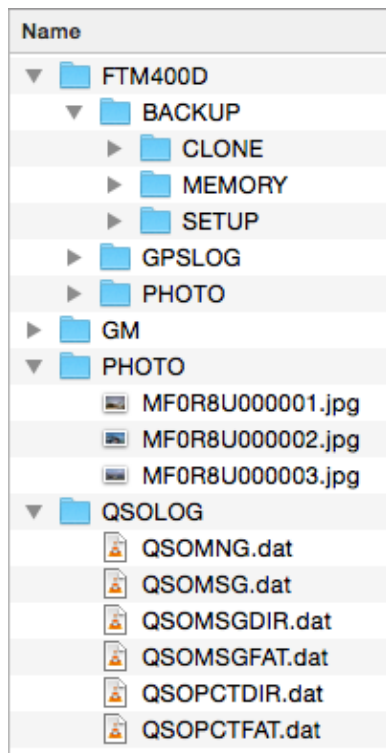


YAESU FTM-400D – SDCARD EXPLORATION – AUGUSTUS 2015

SDCARD STRUCTURE, PHOTO SIZE & FORMAT



On the left-hand side you see the directory structure of the micro SDcard. When formatted you will see this structure, with the exception of the 3 files in the PHOTO folder.

In this paper I will explain how you can add photos to the SDcard and then send them via System Fusion/FTM-400D. Yes, without the Yaesu Camera Mic (MH-85).

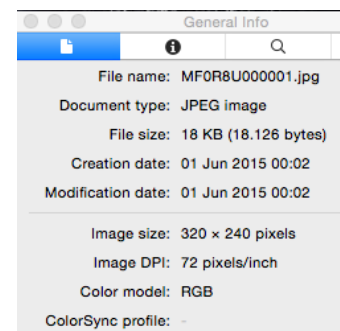
First thing to do is to choose a photo and convert this photo from its original pixel size into 320 x 240 pixels at 72dpi. The picture should then be save as JPEG image. The filename should be like MF0R8U000001.jpg.

Filename format:

M = Microphone

xxxxx = your radio ID (ref. GM menu)

000001 = sequence number



QSOLOG FOLDER

Files in the folder:

QSOMNG.dat → keeps track of the number of messages, photos, groups etc

QSOMSG.dat → contains your message in plain text, each message is 80bytes (max 80 characters) in size

QSOMSGDIR.dat → message directory, contains information like date sent, date received, date created, radio ID, callsign TO/FROM and GPS position.

QSOMSGFAT.dat → File Allocation Table (FAT), contains reference where the messages are located in the QSOMSGDIR.dat file.

QSOPCTDIR.dat → picture directory, contains information like date sent, date received, date created, radio ID, callsign TO/FROM, GPS position and picture filename.

QSOPCTFAT.dat → File Allocation Table (FAT), contains reference where the pictures are located in the QSOPCTDIR.dat file.

In the now following pages we will decode these files in detail...

YAESU FTM-400D – SDCARD EXPLORATION – AUGUSTUS 2015

QSOMNG.DAT

3 pictures, 3 test messages stored on de SDcard.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	b0	03	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	0.YYYYYYYYYYYYYYYY
00000010	00	03	00	01	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FFYYYYYYYYYYYYYY

address 00-01h = total text message counter, equals the number of 80byte messages in the QSOMSG.dat file.

address 10-11h = total picture counter, equals the number of pictures in the PHOTO folder.

address 12-13h = total GM group counter?? (needs more investigation...)

2 pictures, 2 messages (deleted 1 message and 1 picture using the FTM-400D) stored on the SDcard

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	b0	03	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	0.YYYYYYYYYYYYYYYY
00000010	00	03	00	01	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FFYYYYYYYYYYYYYY

As you can see deleting text messages and/or pictures does not affect the content of this file at all.

QSOMSG.DAT

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	74	65	73	74	31	20	20	20	20	20	20	20	20	20	20	20	Test1
00000010	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000020	20	20	20	20	20	20	20	20	20	20	34	31	20	20	20	20	41
00000030	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000040	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000050	66	61	74	20	6F	70	20	73	64	63	61	72	64	20	75	69	fat op sdcard ui
00000060	74	70	6C	75	69	7A	65	6E	20	20	20	20	20	20	20	20	tpluizen
00000070	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000080	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000090	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000A0	37	33	2C	20	50	48	30	50	50	4C	20	2D	20	50	65	74	73, PHOPPL - Pet
000000B0	65	72	20	20	20	20	20	20	20	20	20	20	20	20	20	20	er
000000C0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000D0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000E0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	

In this file your text messages are stored in plain text. Each message takes up 80bytes (4Fh) → recognize the text blocks msg#1 = 00-4Fh, msg#2 = 50-9Fh, msg#3 = A0-EFh etc.

When text messages are deleted with the FTM-400D this file is unchanged → the actual message is NOT deleted from the SD card (see QSMMSGFAT.dat section for the explanation!).

YAESU FTM-400D – SDCARD EXPLORATION – AUGUSTUS 2015

QSOMSGDIR.DAT

FD A0 QSOMSGDIR.dat

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 61 00 00 04 20 20 20 20 20 41 4C 4C 20 20 20 20 a... ALL
00000010 20 20 20 20 20 20 20 20 20 46 30 52 38 55 50 48      FOR8UPH
00000020 30 50 50 4C FF FF FF FF 20 20 20 20 20 20 15 08 OPPLyyyy ..
00000030 14 16 44 00 15 08 14 16 44 00 15 08 14 16 44 00 ..D.....D.....D.
00000040 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000050 00 00 00 00 20 20 20 20 20 20 20 20 20 20 20 20 ....
00000060 20 20 20 20 4E 20 35 32 32 34 30 36 35 30 45 30      N 52240650E0
00000070 30 35 31 39 37 35 36 30 30 00 00 00 00 00 00 00 051975600.....
00000080 61 00 00 05 20 20 20 20 20 41 4C 4C 20 20 20 20 a... ALL
00000090 20 20 20 20 20 20 20 20 20 46 30 52 38 55 50 48      FOR8UPH
000000A0 30 50 50 4C FF FF FF FF 20 20 20 20 20 20 15 08 OPPLyyyy ..
000000B0 14 16 45 00 15 08 14 16 45 00 15 08 14 16 45 00 ..E.....E.....E.
000000C0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000D0 00 00 00 50 20 20 20 20 20 20 20 20 20 20 20 20 ...P
000000E0 20 20 20 20 4E 20 35 32 32 34 30 36 36 30 45 30      N 52240660E0
000000F0 30 35 31 39 37 35 34 30 30 00 00 00 00 00 00 00 051975400.....
00000100 61 00 00 06 20 20 20 20 20 41 4C 4C 20 20 20 20 a... ALL
00000110 20 20 20 20 20 20 20 20 20 46 30 52 38 55 50 48      FOR8UPH
00000120 30 50 50 4C FF FF FF FF 20 20 20 20 20 20 15 08 OPPLyyyy ..
00000130 14 16 46 00 15 08 14 16 46 00 15 08 14 16 46 00 ..F.....F.....F.
00000140 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000150 00 00 00 A0 20 20 20 20 20 20 20 20 20 20 20 20 ...
00000160 20 20 20 20 4E 20 35 32 32 34 30 36 36 30 45 30      N 52240660E0
00000170 30 35 31 39 37 35 34 30 30 00 00 00 00 00 00 00 051975400.....

```

This file is a little more complex, so first I will explain the overall structure. Every text message reference in this directory file takes up 128bytes. This means that information about the first message can be found in message block 00-7Fh, the 2nd message 80-FFh and the 3rd message 100-170Fh.

Now let's zoom in on the message #1 details:

address 09-0Bh shows "ALL" → GM message group identifier
 address 19-23h shows "FOR8UPHOPPL" → which is my radio ID + callsign concatenated
 address 2E-32h shows "1508141644" → which is a date and time stamp (??)
 address 34-38h shows "1508141744" → again a date and time stamp (sent??)
 address 3A-3Eh shows "1508141644" → again a date and time stamp (received??)
 address 64-7Fh shows "N 52240650E0051975600" → your GPS location

QSOMSGFAT.DAT

3 test messages stored on the SDcard.

FD A0 QSOMSGFAT.dat

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 40 00 00 00 40 00 00 80 40 00 01 00      a...a...a...

```

YAESU FTM-400D – SDCARD EXPLORATION – AUGUSTUS 2015

So we have 3 text message on the SDCard and this is how the File Allocation Table (FAT) looks like. Every message uses 4bytes in the FAT, recognize the blocks msg #1 = 00-03h, msg#2 = 04-07h and msg#3 = 08-0Bh.

The format of a FAT entry is like "40 00 00 80", 1st byte = "40h" fixed, 2nd, 3rd and 4th byte = msg block identifier. Every message block can hold 2 message references. The 1st message reference of the 1st message block shows in the FAT as "40 00 00 00", the 2nd message reference of the 1st message block shows in the FAT as "40 00 00 80". Message references in a 4th message block show as "40 00 04 00" for the 1st and "40 00 05 80" for the 2nd message reference, which means FFh x FFh = 65536 message can be stored/allocated with this FAT.

2 messages (deleted 1 message using the FTM-400D) stored on the SDCard.

QSOMSGFAT.dat

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 40 00 00 00 FF FF FF FF 40 00 01 00 ...yyyy@...
```

I deleted the middle (2nd) message using the FTM-400D and the FAT looks like above. The second FAT entry is reset to "FF FF FF FF", remember the message itself is NOT deleted from the SDCard (so by manipulating the QSOMSGFAT.dat with a hex editor you can restore accidentally deleted messages!).

The QSOMSGFAT.dat file grows with the number of messages it will not shrink when message are deleted.

QSOPTDIR.DAT

QSOPTDIR.dat

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 70 2E 2E 2E 20 20 20 20 20 41 4C 4C 20 20 20 20 ... ALL
00000010 20 20 20 20 20 20 20 20 20 46 30 52 38 55 50 48 ... FOR8UPH
00000020 30 50 50 4C FF FF FF FF 20 20 20 20 20 20 15 08 OPPLyyyy ..
00000030 14 18 42 00 15 08 14 19 42 00 15 08 13 16 49 2E ..B.....B.....I.
00000040 32 30 31 35 2F 30 38 2F 31 33 20 31 36 3A 34 39 2015/08/13 16:49
00000050 2E 2E 46 CE 4D 46 30 52 38 55 30 30 30 30 30 31 ..FIMFOR8U0000001
00000060 2E 6A 70 67 4E 20 35 32 32 34 36 38 30 30 45 30 .jpgN 52246800E0
00000070 30 35 31 39 34 37 30 30 30 2E 2E 2E 2E 2E 2E 2E 051947000.....
00000080 70 2E 2E 2E 20 20 20 20 20 41 4C 4C 20 20 20 20 p... ALL
00000090 20 20 20 20 20 20 20 20 20 46 30 52 38 55 50 48 ... FOR8UPH
000000A0 30 50 50 4C FF FF FF FF 20 20 20 20 20 20 15 08 OPPLyyyy ..
000000B0 14 18 43 00 15 08 14 19 43 00 15 08 13 16 50 00 ..C.....C.....P.
000000C0 32 30 31 35 2F 30 38 2F 31 33 20 31 36 3A 35 30 2015/08/13 16:50
000000D0 2E 2E 46 CE 4D 46 30 52 38 55 30 30 30 30 30 32 ..FIMFOR8U0000002
000000E0 2E 6A 70 67 4E 20 35 32 32 34 36 38 30 30 45 30 .jpgN 52246800E0
000000F0 30 35 31 39 34 37 30 30 30 2E 2E 2E 2E 2E 2E 2E 051947000.....
00001000 70 2E 2E 2E 20 20 20 20 20 41 4C 4C 20 20 20 20 p... ALL
00001010 20 20 20 20 20 20 20 20 20 46 30 52 38 55 50 48 ... FOR8UPH
00001020 30 50 50 4C FF FF FF FF 20 20 20 20 20 20 15 08 OPPLyyyy ..
00001030 14 18 44 00 15 08 14 19 44 00 15 08 13 16 51 00 ..D.....D.....Q.
00001040 32 30 31 35 2F 30 38 2F 31 33 20 31 36 3A 35 31 2015/08/13 16:51
00001050 2E 2E 46 CE 4D 46 30 52 38 55 30 30 30 30 30 33 ..FIMFOR8U0000003
00001060 2E 6A 70 67 4E 20 35 32 32 34 36 38 30 30 45 30 .jpgN 52246800E0
00001070 30 35 31 39 34 37 30 30 30 2E 2E 2E 2E 2E 2E 2E 051947000.....
```

YAESU FTM-400D – SDCARD EXPLORATION – AUGUSTUS 2015

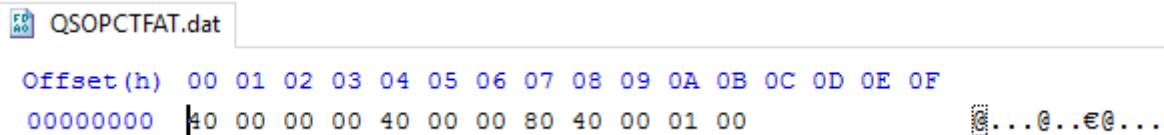
This file is a little more complex, so first I will explain the overall structure. Every picture reference in this directory file takes up 128bytes. This means that information about the first picture can be found in picture message block 00-7Fh, the 2nd picture message 80-FFh and the 3rd picture message 100-170Fh.

Now let's zoom in on the message #1 details:

address 09-0Bh shows "ALL" → GM message group identifier
address 19-23h shows "F0R8UPH0PPL" → which is my radio ID + callsign concatenated
address 2E-32h shows "1508141842" → which is a date and time stamp (received minus 1 hour??)
address 34-38h shows "1508141942" → again a date and time stamp (received)
address 3A-3Eh shows "1508131649" → again a date and time stamp (picture was taken)
address 64-7Fh shows "52246800E0051947000" → your GPS location

QSOPCTFAT.DAT

3 test messages stored on the SDcard.

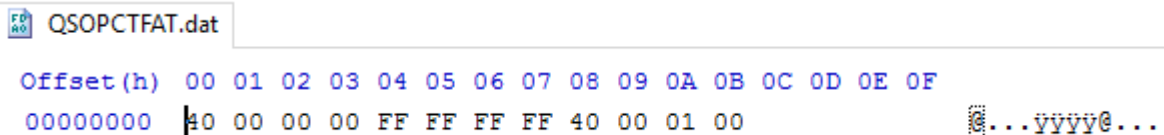


```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 40 00 00 00 40 00 00 80 40 00 01 00
```

So we have 3 pictures on the SDcard and this is how the File Allocation Table (FAT) looks like. Every picture reference uses 4bytes in the FAT, recognize the blocks msg #1 = 00-03h, msg#2 = 04-07h and msg#3 = 08-0Bh.

The format of a FAT entry is like "40 00 00 80", 1st byte = "40h" fixed, 2nd, 3rd and 4th byte = picture msg block identifier. Every picture message block can hold 2 picture message references. The 1st picture message reference of the 1st picture message block shows in the FAT as "40 00 00 00", the 2nd picture message reference of the 1st picture message block shows in the FAT as "40 00 00 80". Picture message references in a 4th picture message block show as "40 00 04 00" for the 1st and "40 00 05 80" for the 2nd picture message reference, which means FFh x FFh = 65536 picture messages (pictures) can be stored/allocated with this FAT.

2 messages (deleted 1 message using the FTM-400D) stored on the SDcard.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 40 00 00 00 FF FF FF FF 40 00 01 00
```

I deleted the middle (2nd) picture using the FTM-400D and the FAT looks like above. The second FAT entry is reset to "FF FF FF FF", remember the picture itself (MF0R8U000002.jpg) is NOT deleted from the SDcard (so by manipulating the QSOPCTFAT.dat with a hex editor you can restore accidentally deleted pictures!).

The QSOPCTFAT.dat file grows with the number of picture messages it will not shrink when picture messages are deleted.

FINAL WORD ABOUT THIS DOCUMENT

Disclaimer

Use the information in this document at your own risk. Editing of files can corrupt them, make a backup before you start editing files.

It is allowed to share the document in its completeness. Please mention the author (me) when you do so.

I would like to thank Michelle, W5NYV, for the information provided via this url:

<http://www.bigideatrouble.com/SystemFusionExploration.pdf>.

Special thanks to David, KG5EIU, for sending me some samples of the information stored on his SDcard.

73, PHOPPL – Peter

Date	Version	Changes
12-AUG-2015	1.0	Initial document
15-AUG-2015	1.1	Typo's corrected (and added some new probably...)