ARC4 KEY FINDER
for
RASPBERRY PI 1/2/3/4

<u>Disclaimer</u>: This software must be used on your own radios in your test lab. Do not use this software on radios that do not belong to you, it is illegal in most countries. Check your local legislation.

This software allows you to find the key for Motorola/Hytera/Anytone DMRA EP/ARC4-40.

I am not the author of this software, I explain to you what I understood about how it works.

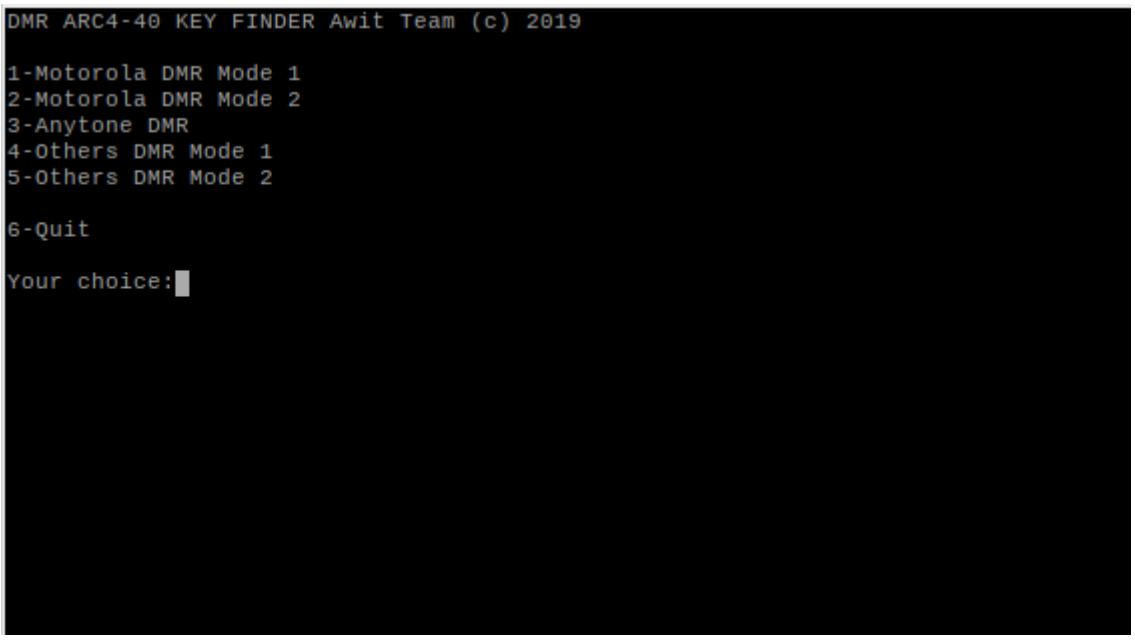
There are two pieces of software: a 32-bit version and a 64-bit version, depending on the operating system you have on your Rasbperry Pi.
The 64-bit version is faster than the 32-bit version.

Verify that the program has execution rights. If it does not have them, set the execution rights, such as:

sudo chmod 777 arc4keyfinder

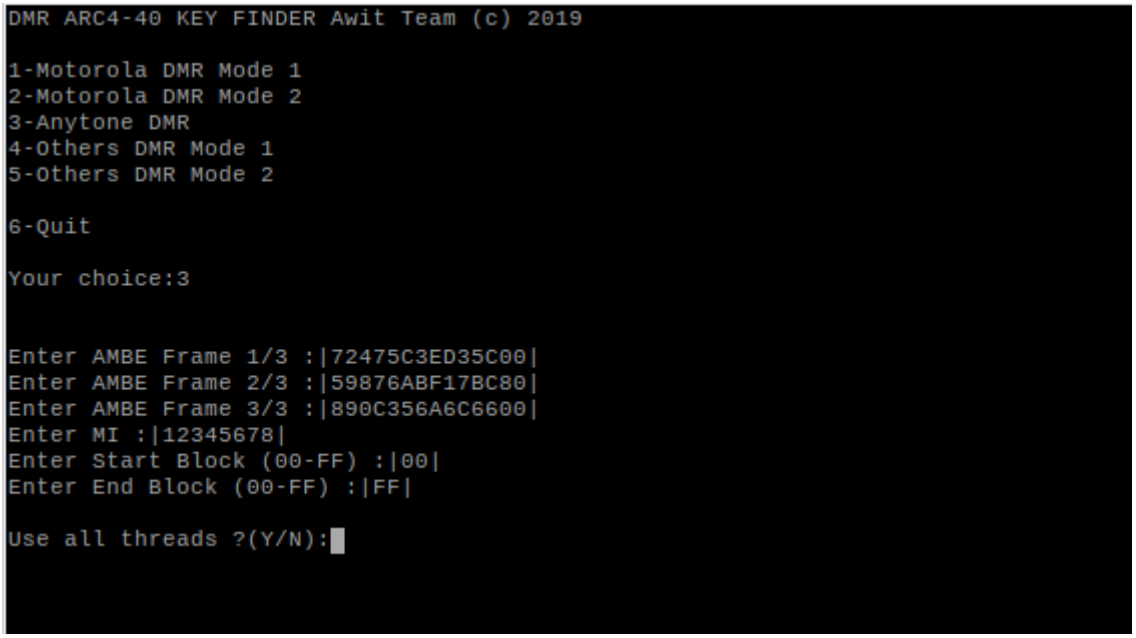
If the program finds a key, it writes it to the **keys_found.txt** file (the file does not exist and will be created when a key is found). Check that write permissions are possible by the program or run the program in root mode:

sudo ./arc4keynfinder

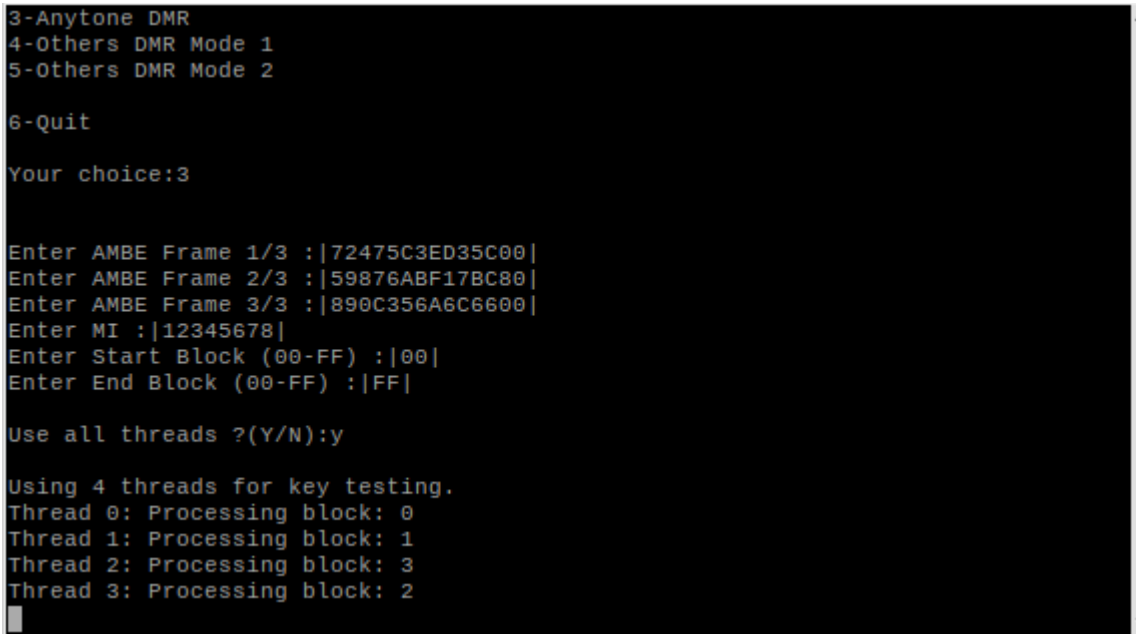


Anytone:

Enter the 3 frames:



The MI, Start Block : 00 and End Block : FF

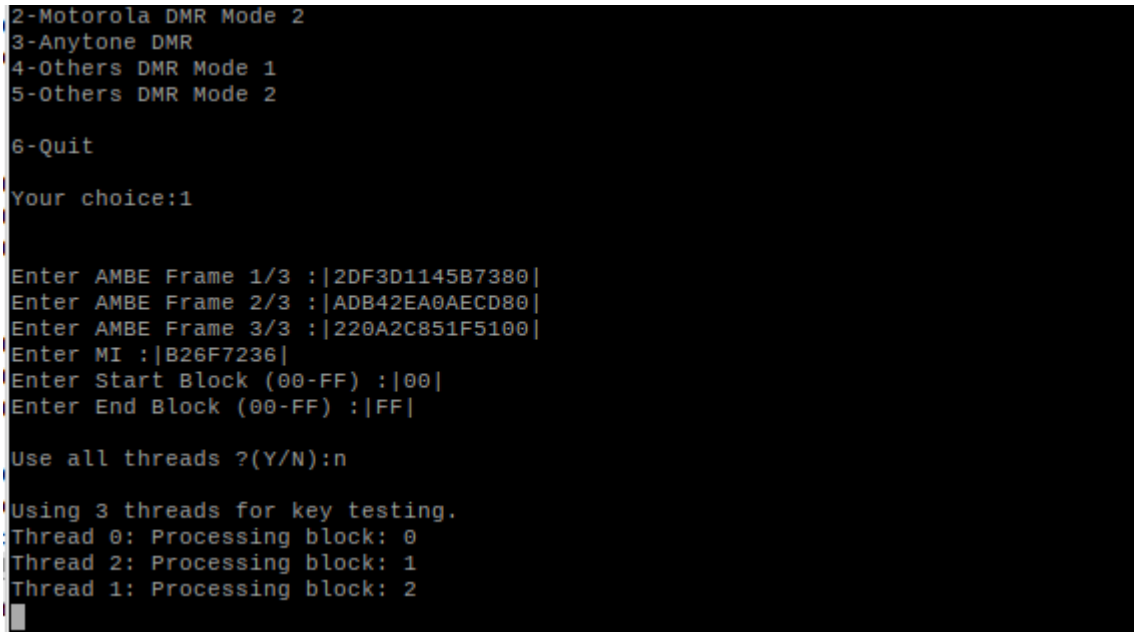


You can use all the threads available on your Raspberry Pi: answer **Yes**

Or all threads minus one: answer **No** (if you want to continue using the Raspberry Pi to do something else).

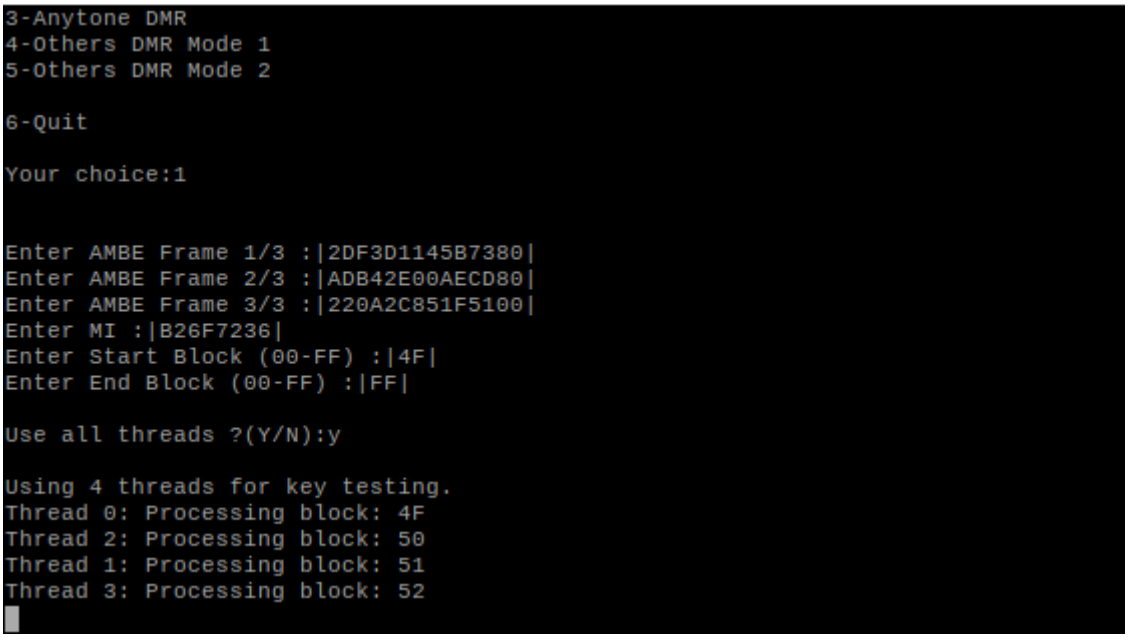
Motorola :

Enter the 3 frames:



You can stop the search and resume it later. For example, you searched for blocks from 00 to 4F (hexadecimal) and stopped the search and turned off the Raspberry Pi.

Another day you can resume the search from 4F. You put 4F in the start block (you have to put the frames back because the program doesn't keep them in memory):



If the program finds a key, it writes it to the **keys_found.txt** file (the file does not exist and will be created when a key is found). Check that write

permissions are possible by the program or run the program in root mode:

sudo ./arc4keynfinder

How do I get ARC4 frames?

Use DSD-FME.

To get help:

dsd-fme.exe -h

dsd-fme -Z

If you have too many receive errors, DSD-FME allows you to capture directly from an RTL-SDR dongle:

Frequency capture 432.153 Mhz :

dsd-fme.exe -Z -i rtl:1:432.153M:22:-2:12:0:6021

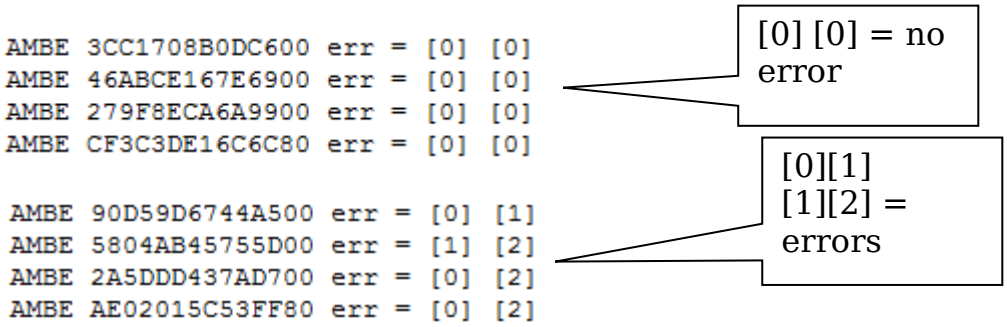
You may need to fine-tune the frequency : 432.151, 432.152, 432.154, 432.155 ...

You can use 2>log.txt to save the screen output to a text file.

dsd-fme.exe -Z -i rtl:1:432.153M:22:-2:12:0:6021 2>log.txt

You must use the first 3 or 18 frames of an ARC4 transmission, but be careful, you have to start at first superframe of the transmission.

It is necessary to ensure that the errors are at 0:



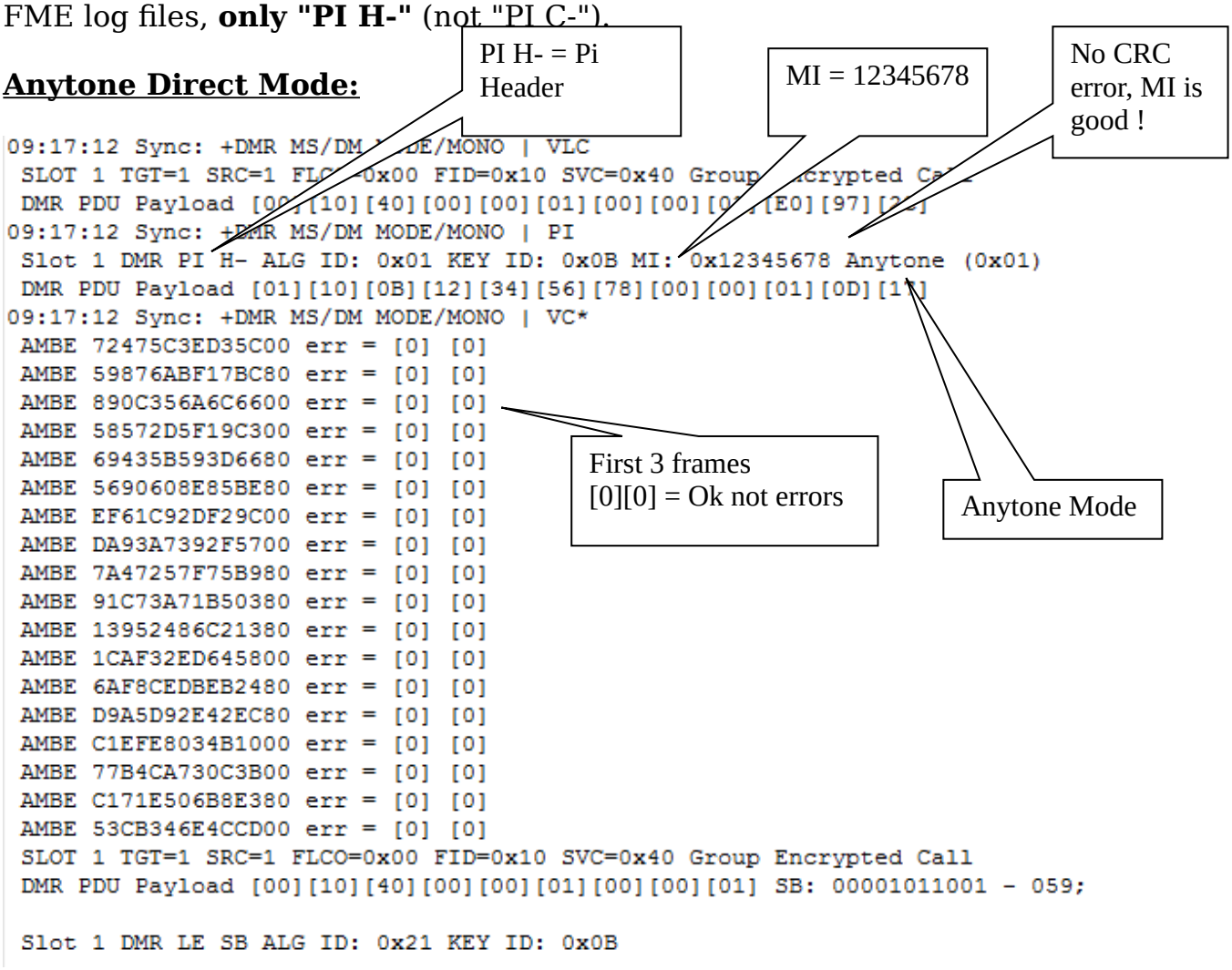
If you have errors : start again, don't waste your time with frames with errors. The ARC4 is very sensitive, a simple error in receiving can prevent the key from being found.

You need to look for the first 3 frames of the first superframe of a transmission or the first 18 frames (for options 4 and 5).

If you capture frames in direct mode, the display by DSD-FME is not the same as if you capture frames in relay mode.

You should always look for "**PI H-**" (this is the PI HEADER) in the DSD-FME log files, **only "PI H-"** (not "PI C-").

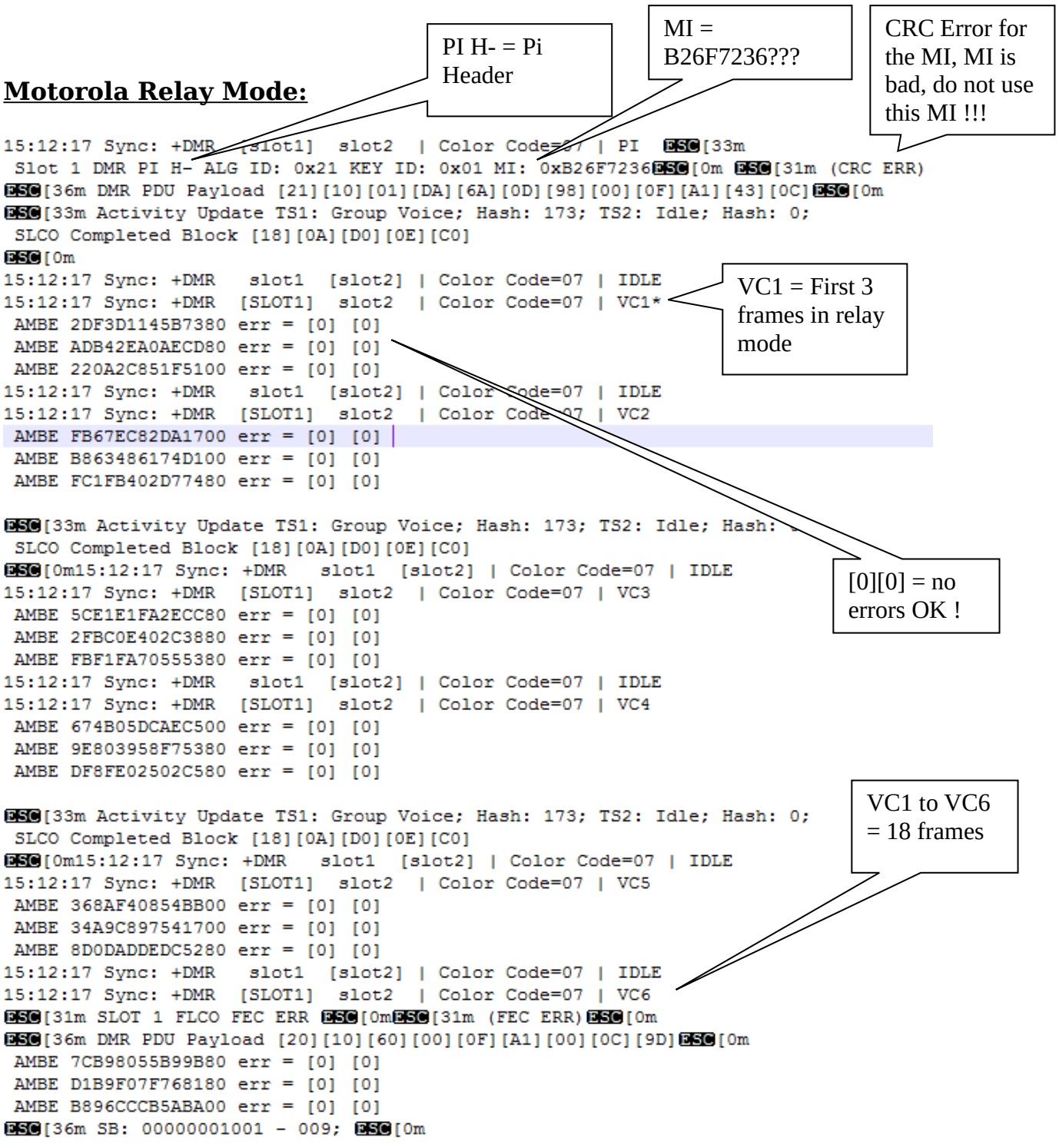
Anytone Direct Mode:




```
09:17:12 Sync: +DMR MS/DM MODE/MONO | VLC
 SLOT 1 TGT=1 SRC=1 FLCO=0x00 FID=0x10 SVC=0x40 Group Encrypted Call
 DMR PDU Payload [00][10][40][00][00][01][00][00][01][E0][97][2E]
09:17:12 Sync: +DMR MS/DM MODE/MONO | PI
 Slot 1 DMR PI H- ALG ID: 0x01 KEY ID: 0x0B MI: 0x12345678 Anytone (0x01)
 DMR PDU Payload [01][10][0B][12][34][56][78][00][00][01][0D][1A]
09:17:12 Sync: +DMR MS/DM MODE/MONO | VC*
 AMBE 72475C3ED35C00 err = [0] [0]
 AMBE 59876ABF17BC80 err = [0] [0]
 AMBE 890C356A6C6600 err = [0] [0]
 AMBE 58572D5F19C300 err = [0] [0]
 AMBE 69435B593D6680 err = [0] [0]
 AMBE 5690608E85BE80 err = [0] [0]
 AMBE EF61C92DF29C00 err = [0] [0]
 AMBE DA93A7392F5700 err = [0] [0]
 AMBE 7A47257F75B980 err = [0] [0]
 AMBE 91C73A71B50380 err = [0] [0]
 AMBE 13952486C21380 err = [0] [0]
 AMBE 1CAF32ED645800 err = [0] [0]
 AMBE 6AF8CEDBEB2480 err = [0] [0]
 AMBE D9A5D92E42EC80 err = [0] [0]
 AMBE C1EFE8034B1000 err = [0] [0]
 AMBE 77B4CA730C3B00 err = [0] [0]
 AMBE C171E506B8E380 err = [0] [0]
 AMBE 53CB346E4CCD00 err = [0] [0]
 SLOT 1 TGT=1 SRC=1 FLCO=0x00 FID=0x10 SVC=0x40 Group Encrypted Call
 DMR PDU Payload [00][10][40][00][00][01][00][00][01] SB: 00001011001 - 059;

 Slot 1 DMR LE SB ALG ID: 0x21 KEY ID: 0x0B
```

Motorola Relay Mode:




```
15:12:17 Sync: +DMR   [slot1]  slot2  | Color Code=07 | PI  ESC[33m
 Slot 1 DMR PI H- ALG ID: 0x21 KEY ID: 0x01 MI: 0xB26F7236ESC[0m ESC[31m (CRC ERR)
ESC[36m DMR PDU Payload [21][10][01][DA][6A][0D][98][00][0F][A1][43][0C]ESC[0m
ESC[33m Activity Update TS1: Group Voice; Hash: 173; TS2: Idle; Hash: 0;
 SLCO Completed Block [18][0A][D0][0E][C0]
ESC[0m
15:12:17 Sync: +DMR   slot1  [slot2] | Color Code=07 | IDLE
15:12:17 Sync: +DMR  [SLOT1]  slot2  | Color Code=07 | VC1*
 AMBE 2DF3D1145B7380 err = [0] [0]
 AMBE ADB42EA0AECD80 err = [0] [0]
 AMBE 220A2C851F5100 err = [0] [0]
15:12:17 Sync: +DMR   slot1  [slot2] | Color Code=07 | IDLE
15:12:17 Sync: +DMR  [SLOT1]  slot2  | Color Code=07 | VC2
 AMBE FB67EC82DA1700 err = [0] [0]
 AMBE B863486174D100 err = [0] [0]
 AMBE FC1FB402D77480 err = [0] [0]

ESC[33m Activity Update TS1: Group Voice; Hash: 173; TS2: Idle; Hash:
 SLCO Completed Block [18][0A][D0][0E][C0]
ESC[0m15:12:17 Sync: +DMR   slot1  [slot2] | Color Code=07 | IDLE
15:12:17 Sync: +DMR  [SLOT1]  slot2  | Color Code=07 | VC3
 AMBE 5CE1E1FA2ECC80 err = [0] [0]
 AMBE 2FBC0E402C3880 err = [0] [0]
 AMBE FBF1FA70555380 err = [0] [0]
15:12:17 Sync: +DMR   slot1  [slot2] | Color Code=07 | IDLE
15:12:17 Sync: +DMR  [SLOT1]  slot2  | Color Code=07 | VC4
 AMBE 674B05DCAEC500 err = [0] [0]
 AMBE 9E803958F75380 err = [0] [0]
 AMBE DF8FE02502C580 err = [0] [0]

ESC[33m Activity Update TS1: Group Voice; Hash: 173; TS2: Idle; Hash: 0;
 SLCO Completed Block [18][0A][D0][0E][C0]
ESC[0m15:12:17 Sync: +DMR   slot1  [slot2] | Color Code=07 | IDLE
15:12:17 Sync: +DMR  [SLOT1]  slot2  | Color Code=07 | VC5
 AMBE 368AF40854BB00 err = [0] [0]
 AMBE 34A9C897541700 err = [0] [0]
 AMBE 8D0DADDEDC5280 err = [0] [0]
15:12:17 Sync: +DMR   slot1  [slot2] | Color Code=07 | IDLE
15:12:17 Sync: +DMR  [SLOT1]  slot2  | Color Code=07 | VC6
ESC[31m SLOT 1 FLCO FEC ERR ESC[0mESC[31m (FEC ERR)ESC[0m
ESC[36m DMR PDU Payload [20][10][60][00][0F][A1][00][0C][9D]ESC[0m
 AMBE 7CB98055B99B80 err = [0] [0]
 AMBE D1B9F07F768180 err = [0] [0]
 AMBE B896CCCB5ABA00 err = [0] [0]
ESC[36m SB: 00000001001 - 009; ESC[0m
```

A particularity of the DMR in relay mode exists: if the walkie-talkie is far from the relay, its transmission can arrive with errors.
The relay will send this data back with errors but as it is a new emission it sends the data received with error into data without error (but without really correcting them).

If you listen to the output relay you may believe that there are no errors (display err = [0] [0] when there are errors inside the data because the frames received by the relay were bad.

The key will then never be found with these frames.

To be sure that there are no errors, you have to listen to the input relay (before the relay rebroadcasts). If you capture frames with err = [0] [0] in the input relay then you can be sure that there are no errors.