```
#!/usr/bin/perl

# Copyright (c) Mauro Carvalho Chehab <mchehab@infradead.org>
# Released under GPLv2
#
# In order to use, you need to:
#       1) Download the windows driver with something like:
#       Version 2.4
#               wget http://www.twinhan.com/files/AW/BDA T/20080303_V1.0.6.7.zip
#               or wget http://www.stefanringel.de/pub/20080303_V1.0.6.7.zip
#       Version 2.7
#               wget
http://www.steventoth.net/linux/xc5000/HVR-12x0-14x0-17x0_1_25_25271_WHQL.zip
#       2) Extract the files from the zip into the current dir:
#               unzip -j 20080303_V1.0.6.7.zip 20080303_v1.0.6.7/UDXTTM6000.sys
#               unzip -j HVR-12x0-14x0-17x0_1_25_25271_WHQL.zip
Driver85/hcw85bda.sys
#       3) run the script:
#               ./extract_xc3028.pl
#       4) copy the generated files:
#               cp xc3028-v24.fw /lib/firmware
#               cp xc3028-v27.fw /lib/firmware

#use strict;
use IO::Handle;

my $debug=0;

sub verify ($$)
{
        my ($filename, $hash) = @_;
        my ($testhash);

        if (system("which md5sum > /dev/null 2>&1")) {
                die "This firmware requires the md5sum command - see
http://www.gnu.org/software/coreutils/\n";
        }

        open(CMD, "md5sum ".$filename."|");
        $testhash = <CMD>;
        $testhash =~ /([a-zA-Z0-9]*)/;
        $testhash = $1;
        close CMD;
                die "Hash of extracted file does not match (found $testhash,
expected $hash!\n" if ($testhash ne $hash);
}

sub get_hunk ($$)
{
        my ($offset, $length) = @_;
        my ($chunklength, $buf, $rcount, $out);

        sysseek(INFILE, $offset, SEEK_SET);
        while ($length > 0) {
        # Calc chunk size
                $chunklength = 2048;
```

```
                $chunklength = $length if ($chunklength > $length);

                $rcount = sysread(INFILE, $buf, $chunklength);
                die "Ran out of data\n" if ($rcount != $chunklength);
                $out .= $buf;
                $length -= $rcount;
        }
        return $out;
}

sub write_le16($)
{
        my $val = shift;
        my $msb = ($val >> 8) &0xff;
        my $lsb = $val & 0xff;

        syswrite(OUTFILE, chr($lsb).chr($msb));
}

sub write_le32($)
{
        my $val = shift;
        my $13 = ($val >> 24) & 0xff;
        my $12 = ($val >> 16) & 0xff;
        my $11 = ($val >> 8)  & 0xff;
        my $10 = $val         & 0xff;

        syswrite(OUTFILE, chr($10).chr($11).chr($12).chr($13));
}

sub write_le64($$)
{
        my $msb_val = shift;
        my $lsb_val = shift;
        my $17 = ($msb_val >> 24) & 0xff;
        my $16 = ($msb_val >> 16) & 0xff;
        my $15 = ($msb_val >> 8)  & 0xff;
        my $14 = $msb_val         & 0xff;

        my $13 = ($lsb_val >> 24) & 0xff;
        my $12 = ($lsb_val >> 16) & 0xff;
        my $11 = ($lsb_val >> 8)  & 0xff;
        my $10 = $lsb_val         & 0xff;

        syswrite(OUTFILE,
                 chr($10).chr($11).chr($12).chr($13).
                 chr($14).chr($15).chr($16).chr($17));
}

sub write_hunk($$)
{
        my ($offset, $length) = @_;
        my $out = get_hunk($offset, $length);

        printf "(len %d) ",$length if ($debug);
```

```
        for (my $i=0;$i<$length;$i++) {
                printf "%02x ",ord(substr($out,$i,1)) if ($debug);
        }
        printf "\n" if ($debug);

        syswrite(OUTFILE, $out);
}

sub write_hunk_fix_endian($$)
{
        my ($offset, $length) = @_;
        my $out = get_hunk($offset, $length);

        printf "(len_fix %d) ",$length if ($debug);

        for (my $i=0;$i<$length;$i++) {
                printf "%02x ",ord(substr($out,$i,1)) if ($debug);
        }
        printf "\n" if ($debug);

        my $i=0;
        while ($i<$length) {
                my $size = ord(substr($out,$i,1))*256+ord(substr($out,$i+1,1));
                syswrite(OUTFILE, substr($out,$i+1,1));
                syswrite(OUTFILE, substr($out,$i,1));
                $i+=2;
                if ($size>0 && $size <0x8000) {
                        for (my $j=0;$j<$size;$j++) {
                                syswrite(OUTFILE, substr($out,$j+$i,1));
                        }
                        $i+=$size;
                }
        }
}

sub main_firmware_24($$$$)
{
        my $out;
        my $j=0;
        my $outfile = shift;
        my $name    = shift;
        my $version = shift;
        my $nr_desc = shift;

        for ($j = length($name); $j <32; $j++) {
                $name = $name.chr(0);
        }

        open OUTFILE, ">$outfile";
        syswrite(OUTFILE, $name);
        write_le16($version);
        write_le16($nr_desc);

        #
        # Firmware 0, type: BASE FW   F8MHZ (0x00000003), id:
(0000000000000000), size: 6635
```

```
        #

        write_le32(0x00000003);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(6635);                       # Size
        write_hunk_fix_endian(257752, 6635);

        #
        # Firmware 1, type: BASE FW   F8MHZ MTS (0x00000007), id:
(0000000000000000), size: 6635
        #

        write_le32(0x00000007);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(6635);                       # Size
        write_hunk_fix_endian(264392, 6635);

        #
        # Firmware 2, type: BASE FW   FM (0x00000401), id: (0000000000000000),
size: 6525
        #

        write_le32(0x00000401);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(6525);                       # Size
        write_hunk_fix_endian(271040, 6525);

        #
        # Firmware 3, type: BASE FW   FM INPUT1 (0x00000c01), id:
(0000000000000000), size: 6539
        #

        write_le32(0x00000c01);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(6539);                       # Size
        write_hunk_fix_endian(277568, 6539);

        #
        # Firmware 4, type: BASE FW   (0x00000001), id: (0000000000000000),
size: 6633
        #

        write_le32(0x00000001);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(6633);                       # Size
        write_hunk_fix_endian(284120, 6633);

        #
        # Firmware 5, type: BASE FW   MTS (0x00000005), id: (0000000000000000),
size: 6617
        #

        write_le32(0x00000005);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(6617);                       # Size
        write_hunk_fix_endian(290760, 6617);
```

```
        #
        # Firmware 6, type: STD FW    (0x00000000), id: PAL/BG A2/A
(0000000100000007), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000001, 0x00000007);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(297384, 161);

        #
        # Firmware 7, type: STD FW    MTS (0x00000004), id: PAL/BG A2/A
(0000000100000007), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000001, 0x00000007);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(297552, 169);

        #
        # Firmware 8, type: STD FW    (0x00000000), id: PAL/BG A2/B
(0000000200000007), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000002, 0x00000007);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(297728, 161);

        #
        # Firmware 9, type: STD FW    MTS (0x00000004), id: PAL/BG A2/B
(0000000200000007), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000002, 0x00000007);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(297896, 169);

        #
        # Firmware 10, type: STD FW    (0x00000000), id: PAL/BG NICAM/A
(0000000400000007), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000004, 0x00000007);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(298072, 161);

        #
        # Firmware 11, type: STD FW    MTS (0x00000004), id: PAL/BG NICAM/A
(0000000400000007), size: 169
        #
```

```
        write_le32(0x00000004);                 # Type
        write_le64(0x00000004, 0x00000007);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(298240, 169);

        #
        # Firmware 12, type: STD FW    (0x00000000), id: PAL/BG NICAM/B
(0000000800000007), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000008, 0x00000007);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(298416, 161);

        #
        # Firmware 13, type: STD FW    MTS (0x00000004), id: PAL/BG NICAM/B
(0000000800000007), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000008, 0x00000007);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(298584, 169);

        #
        # Firmware 14, type: STD FW    (0x00000000), id: PAL/DK A2
(00000003000000e0), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(298760, 161);

        #
        # Firmware 15, type: STD FW    MTS (0x00000004), id: PAL/DK A2
(00000003000000e0), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(298928, 169);

        #
        # Firmware 16, type: STD FW    (0x00000000), id: PAL/DK NICAM
(0000000c000000e0), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x0000000c, 0x000000e0);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(299104, 161);

        #
```

```
        # Firmware 17, type: STD FW    MTS (0x00000004), id: PAL/DK NICAM
(0000000c000000e0), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x0000000c, 0x000000e0);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(299272, 169);

        #
        # Firmware 18, type: STD FW    (0x00000000), id: SECAM/K1
(0000000000200000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(299448, 161);

        #
        # Firmware 19, type: STD FW    MTS (0x00000004), id: SECAM/K1
(0000000000200000), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(299616, 169);

        #
        # Firmware 20, type: STD FW    (0x00000000), id: SECAM/K3
(0000000004000000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x04000000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(299792, 161);

        #
        # Firmware 21, type: STD FW    MTS (0x00000004), id: SECAM/K3
(0000000004000000), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x04000000);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(299960, 169);

        #
        # Firmware 22, type: STD FW    D2633 DTV6 ATSC (0x00010030), id:
(0000000000000000), size: 149
        #

        write_le32(0x00010030);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
```

```
        write_le32(149);                        # Size
        write_hunk_fix_endian(300136, 149);

        #
        # Firmware 23, type: STD FW    D2620 DTV6 QAM (0x00000068), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000068);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(300296, 149);

        #
        # Firmware 24, type: STD FW    D2633 DTV6 QAM (0x00000070), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000070);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(300448, 149);

        #
        # Firmware 25, type: STD FW    D2620 DTV7 (0x00000088), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000088);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(300608, 149);

        #
        # Firmware 26, type: STD FW    D2633 DTV7 (0x00000090), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000090);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(300760, 149);

        #
        # Firmware 27, type: STD FW    D2620 DTV78 (0x00000108), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000108);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(300920, 149);

        #
        # Firmware 28, type: STD FW    D2633 DTV78 (0x00000110), id:
(0000000000000000), size: 149
```

```
	#

	write_le32(0x00000110);			# Type
	write_le64(0x00000000, 0x00000000);	# ID
	write_le32(149);			# Size
	write_hunk_fix_endian(301072, 149);

	#
	# Firmware 29, type: STD FW    D2620 DTV8 (0x00000208), id:
(0000000000000000), size: 149
	#

	write_le32(0x00000208);			# Type
	write_le64(0x00000000, 0x00000000);	# ID
	write_le32(149);			# Size
	write_hunk_fix_endian(301232, 149);

	#
	# Firmware 30, type: STD FW    D2633 DTV8 (0x00000210), id:
(0000000000000000), size: 149
	#

	write_le32(0x00000210);			# Type
	write_le64(0x00000000, 0x00000000);	# ID
	write_le32(149);			# Size
	write_hunk_fix_endian(301384, 149);

	#
	# Firmware 31, type: STD FW    FM (0x00000400), id: (0000000000000000),
size: 135
	#

	write_le32(0x00000400);			# Type
	write_le64(0x00000000, 0x00000000);	# ID
	write_le32(135);			# Size
	write_hunk_fix_endian(301554, 135);

	#
	# Firmware 32, type: STD FW    (0x00000000), id: PAL/I
(0000000000000010), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000000, 0x00000010);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(301688, 161);

	#
	# Firmware 33, type: STD FW    MTS (0x00000004), id: PAL/I
(0000000000000010), size: 169
	#

	write_le32(0x00000004);			# Type
	write_le64(0x00000000, 0x00000010);	# ID
	write_le32(169);			# Size
	write_hunk_fix_endian(301856, 169);
```

```
        #
        # Firmware 34, type: STD FW    (0x00000000), id: SECAM/L AM
(0000001000400000), size: 169
        #

        #
        # Firmware 35, type: STD FW    (0x00000000), id: SECAM/L NICAM
(0000000c00400000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x0000000c, 0x00400000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(302032, 161);

        #
        # Firmware 36, type: STD FW    (0x00000000), id: SECAM/Lc
(0000000000800000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00800000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(302200, 161);

        #
        # Firmware 37, type: STD FW    (0x00000000), id: NTSC/M Kr
(0000000000008000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(302368, 161);

        #
        # Firmware 38, type: STD FW    LCD (0x00001000), id: NTSC/M Kr
(0000000000008000), size: 161
        #

        write_le32(0x00001000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(302536, 161);

        #
        # Firmware 39, type: STD FW    LCD NOGD (0x00003000), id: NTSC/M Kr
(0000000000008000), size: 161
        #

        write_le32(0x00003000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(302704, 161);
```

```
        #
        # Firmware 40, type: STD FW    MTS (0x00000004), id: NTSC/M Kr
(0000000000008000), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(302872, 169);

        #
        # Firmware 41, type: STD FW    (0x00000000), id: NTSC PAL/M PAL/N
(000000000000b700), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(303048, 161);

        #
        # Firmware 42, type: STD FW    LCD (0x00001000), id: NTSC PAL/M PAL/N
(000000000000b700), size: 161
        #

        write_le32(0x00001000);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(303216, 161);

        #
        # Firmware 43, type: STD FW    LCD NOGD (0x00003000), id: NTSC PAL/M
PAL/N (000000000000b700), size: 161
        #

        write_le32(0x00003000);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(303384, 161);

        #
        # Firmware 44, type: STD FW    (0x00000000), id: NTSC/M Jp
(0000000000002000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00002000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(303552, 161);

        #
        # Firmware 45, type: STD FW    MTS (0x00000004), id: NTSC PAL/M PAL/N
(000000000000b700), size: 169
        #

        write_le32(0x00000004);                 # Type
```

```
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(303720, 169);

        #
        # Firmware 46, type: STD FW    MTS LCD (0x00001004), id: NTSC PAL/M
PAL/N (000000000000b700), size: 169
        #

        write_le32(0x00001004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(303896, 169);

        #
        # Firmware 47, type: STD FW    MTS LCD NOGD (0x00003004), id: NTSC PAL/M
PAL/N (000000000000b700), size: 169
        #

        write_le32(0x00003004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(304072, 169);

        #
        # Firmware 48, type: SCODE FW  HAS IF (0x60000000), IF = 3.28 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3280);                       # IF
        write_le32(192);                        # Size
        write_hunk(309048, 192);

        #
        # Firmware 49, type: SCODE FW  HAS IF (0x60000000), IF = 3.30 MHz id:
(0000000000000000), size: 192
        #

#       write_le32(0x60000000);                 # Type
#       write_le64(0x00000000, 0x00000000);     # ID
#       write_le16(3300);                       # IF
#       write_le32(192);                        # Size
#       write_hunk(304440, 192);

        #
        # Firmware 50, type: SCODE FW  HAS IF (0x60000000), IF = 3.44 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3440);                       # IF
        write_le32(192);                        # Size
        write_hunk(309432, 192);
```

```
        #
        # Firmware 51, type: SCODE FW  HAS IF (0x60000000), IF = 3.46 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3460);                       # IF
        write_le32(192);                        # Size
        write_hunk(309624, 192);

        #
        # Firmware 52, type: SCODE FW  DTV6 ATSC OREN36 HAS IF (0x60210020), IF
= 3.80 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60210020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3800);                       # IF
        write_le32(192);                        # Size
        write_hunk(306936, 192);

        #
        # Firmware 53, type: SCODE FW  HAS IF (0x60000000), IF = 4.00 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4000);                       # IF
        write_le32(192);                        # Size
        write_hunk(309240, 192);

        #
        # Firmware 54, type: SCODE FW  DTV6 ATSC TOYOTA388 HAS IF (0x60410020),
IF = 4.08 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60410020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4080);                       # IF
        write_le32(192);                        # Size
        write_hunk(307128, 192);

        #
        # Firmware 55, type: SCODE FW  HAS IF (0x60000000), IF = 4.20 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4200);                       # IF
        write_le32(192);                        # Size
        write_hunk(308856, 192);
```

```
        #
        # Firmware 56, type: SCODE FW  MONO HAS IF (0x60008000), IF = 4.32 MHz
id: NTSC/M Kr (0000000000008000), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le16(4320);                       # IF
        write_le32(192);                        # Size
        write_hunk(305208, 192);

        #
        # Firmware 57, type: SCODE FW  HAS IF (0x60000000), IF = 4.45 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4450);                       # IF
        write_le32(192);                        # Size
        write_hunk(309816, 192);

        #
        # Firmware 58, type: SCODE FW  MTS LCD NOGD MONO IF HAS IF (0x6002b004),
IF = 4.50 MHz id: NTSC PAL/M PAL/N (000000000000b700), size: 192
        #

        write_le32(0x6002b004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le16(4500);                       # IF
        write_le32(192);                        # Size
        write_hunk(304824, 192);

        #
        # Firmware 59, type: SCODE FW  LCD NOGD IF HAS IF (0x60023000), IF =
4.60 MHz id: NTSC/M Kr (0000000000008000), size: 192
        #

        write_le32(0x60023000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le16(4600);                       # IF
        write_le32(192);                        # Size
        write_hunk(305016, 192);

        #
        # Firmware 60, type: SCODE FW  DTV6 QAM DTV7 DTV78 DTV8 ZARLINK456 HAS
IF (0x620003e0), IF = 4.76 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x620003e0);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4760);                       # IF
        write_le32(192);                        # Size
        write_hunk(304440, 192);

        #
```

```
        # Firmware 61, type: SCODE FW  HAS IF (0x60000000), IF = 4.94 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4940);                       # IF
        write_le32(192);                        # Size
        write_hunk(308664, 192);

        #
        # Firmware 62, type: SCODE FW  HAS IF (0x60000000), IF = 5.26 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5260);                       # IF
        write_le32(192);                        # Size
        write_hunk(307704, 192);

        #
        # Firmware 63, type: SCODE FW  MONO HAS IF (0x60008000), IF = 5.32 MHz
id: PAL/BG A2 NICAM (0000000f00000007), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x0000000f, 0x00000007);     # ID
        write_le16(5320);                       # IF
        write_le32(192);                        # Size
        write_hunk(307896, 192);

        #
        # Firmware 64, type: SCODE FW  DTV7 DTV78 DTV8 DIBCOM52 CHINA HAS IF
(0x65000380), IF = 5.40 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x65000380);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5400);                       # IF
        write_le32(192);                        # Size
        write_hunk(304248, 192);

        #
        # Firmware 65, type: SCODE FW  DTV6 ATSC OREN538 HAS IF (0x60110020), IF
= 5.58 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60110020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5580);                       # IF
        write_le32(192);                        # Size
        write_hunk(306744, 192);

        #
        # Firmware 66, type: SCODE FW  HAS IF (0x60000000), IF = 5.64 MHz id:
```

```
PAL/BG A2 (0000000300000007), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000003, 0x00000007);     # ID
        write_le16(5640);                       # IF
        write_le32(192);                        # Size
        write_hunk(305592, 192);

        #
        # Firmware 67, type: SCODE FW  HAS IF (0x60000000), IF = 5.74 MHz id:
PAL/BG NICAM (0000000c00000007), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x0000000c, 0x00000007);     # ID
        write_le16(5740);                       # IF
        write_le32(192);                        # Size
        write_hunk(305784, 192);

        #
        # Firmware 68, type: SCODE FW  HAS IF (0x60000000), IF = 5.90 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5900);                       # IF
        write_le32(192);                        # Size
        write_hunk(307512, 192);

        #
        # Firmware 69, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.00 MHz
id: PAL/DK PAL/I SECAM/K3 SECAM/L SECAM/Lc NICAM (0000000c04c000f0), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x0000000c, 0x04c000f0);     # ID
        write_le16(6000);                       # IF
        write_le32(192);                        # Size
        write_hunk(305576, 192);

        #
        # Firmware 70, type: SCODE FW  DTV6 QAM ATSC LG60 F6MHZ HAS IF
(0x68050060), IF = 6.20 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x68050060);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(6200);                       # IF
        write_le32(192);                        # Size
        write_hunk(306552, 192);

        #
        # Firmware 71, type: SCODE FW  HAS IF (0x60000000), IF = 6.24 MHz id:
PAL/I (0000000000000010), size: 192
```

```
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000010);     # ID
        write_le16(6240);                       # IF
        write_le32(192);                        # Size
        write_hunk(305400, 192);

        #
        # Firmware 72, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.32 MHz
id: SECAM/K1 (0000000000200000), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le16(6320);                       # IF
        write_le32(192);                        # Size
        write_hunk(308472, 192);

        #
        # Firmware 73, type: SCODE FW  HAS IF (0x60000000), IF = 6.34 MHz id:
SECAM/K1 (0000000000200000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le16(6340);                       # IF
        write_le32(192);                        # Size
        write_hunk(306360, 192);

        #
        # Firmware 74, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.50 MHz
id: PAL/DK SECAM/K3 SECAM/L NICAM (0000000c044000e0), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x0000000c, 0x044000e0);     # ID
        write_le16(6500);                       # IF
        write_le32(192);                        # Size
        write_hunk(308280, 192);

        #
        # Firmware 75, type: SCODE FW  DTV6 ATSC ATI638 HAS IF (0x60090020), IF
= 6.58 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60090020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(6580);                       # IF
        write_le32(192);                        # Size
        write_hunk(304632, 192);

        #
        # Firmware 76, type: SCODE FW  HAS IF (0x60000000), IF = 6.60 MHz id:
PAL/DK A2 (00000003000000e0), size: 192
        #
```

```
        write_le32(0x60000000);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le16(6600);                       # IF
        write_le32(192);                        # Size
        write_hunk(306168, 192);

        #
        # Firmware 77, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.68 MHz
id: PAL/DK A2 (00000003000000e0), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le16(6680);                       # IF
        write_le32(192);                        # Size
        write_hunk(308088, 192);

        #
        # Firmware 78, type: SCODE FW  DTV6 ATSC TOYOTA794 HAS IF (0x60810020),
IF = 8.14 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60810020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(8140);                       # IF
        write_le32(192);                        # Size
        write_hunk(307320, 192);

        #
        # Firmware 79, type: SCODE FW  HAS IF (0x60000000), IF = 8.20 MHz id:
(0000000000000000), size: 192
        #

#       write_le32(0x60000000);                 # Type
#       write_le64(0x00000000, 0x00000000);     # ID
#       write_le16(8200);                       # IF
#       write_le32(192);                        # Size
#       write_hunk(308088, 192);
}

sub main_firmware_27($$$$)
{
        my $out;
        my $j=0;
        my $outfile = shift;
        my $name    = shift;
        my $version = shift;
        my $nr_desc = shift;

        for ($j = length($name); $j <32; $j++) {
                $name = $name.chr(0);
        }

        open OUTFILE, ">$outfile";
        syswrite(OUTFILE, $name);
```

```
        write_le16($version);
        write_le16($nr_desc);

        #
        # Firmware 0, type: BASE FW   F8MHZ (0x00000003), id:
(0000000000000000), size: 8718
        #

        write_le32(0x00000003);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(8718);                       # Size
        write_hunk_fix_endian(813432, 8718);

        #
        # Firmware 1, type: BASE FW   F8MHZ MTS (0x00000007), id:
(0000000000000000), size: 8712
        #

        write_le32(0x00000007);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(8712);                       # Size
        write_hunk_fix_endian(822152, 8712);

        #
        # Firmware 2, type: BASE FW   FM (0x00000401), id: (0000000000000000),
size: 8562
        #

        write_le32(0x00000401);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(8562);                       # Size
        write_hunk_fix_endian(830872, 8562);

        #
        # Firmware 3, type: BASE FW   FM INPUT1 (0x00000c01), id:
(0000000000000000), size: 8576
        #

        write_le32(0x00000c01);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(8576);                       # Size
        write_hunk_fix_endian(839440, 8576);

        #
        # Firmware 4, type: BASE FW   (0x00000001), id: (0000000000000000),
size: 8706
        #

        write_le32(0x00000001);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(8706);                       # Size
        write_hunk_fix_endian(848024, 8706);

        #
        # Firmware 5, type: BASE FW   MTS (0x00000005), id: (0000000000000000),
size: 8682
```

```
	#

	write_le32(0x00000005);			# Type
	write_le64(0x00000000, 0x00000000);	# ID
	write_le32(8682);			# Size
	write_hunk_fix_endian(856736, 8682);

	#
	# Firmware 6, type: STD FW    (0x00000000), id: PAL/BG A2/A
(0000000100000007), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000001, 0x00000007);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(865424, 161);

	#
	# Firmware 7, type: STD FW    MTS (0x00000004), id: PAL/BG A2/A
(0000000100000007), size: 169
	#

	write_le32(0x00000004);			# Type
	write_le64(0x00000001, 0x00000007);	# ID
	write_le32(169);			# Size
	write_hunk_fix_endian(865592, 169);

	#
	# Firmware 8, type: STD FW    (0x00000000), id: PAL/BG A2/B
(0000000200000007), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000002, 0x00000007);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(865424, 161);

	#
	# Firmware 9, type: STD FW    MTS (0x00000004), id: PAL/BG A2/B
(0000000200000007), size: 169
	#

	write_le32(0x00000004);			# Type
	write_le64(0x00000002, 0x00000007);	# ID
	write_le32(169);			# Size
	write_hunk_fix_endian(865592, 169);

	#
	# Firmware 10, type: STD FW    (0x00000000), id: PAL/BG NICAM/A
(0000000400000007), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000004, 0x00000007);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(866112, 161);
```

```
        #
        # Firmware 11, type: STD FW    MTS (0x00000004), id: PAL/BG NICAM/A
(0000000400000007), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000004, 0x00000007);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(866280, 169);

        #
        # Firmware 12, type: STD FW    (0x00000000), id: PAL/BG NICAM/B
(0000000800000007), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000008, 0x00000007);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(866112, 161);

        #
        # Firmware 13, type: STD FW    MTS (0x00000004), id: PAL/BG NICAM/B
(0000000800000007), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000008, 0x00000007);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(866280, 169);

        #
        # Firmware 14, type: STD FW    (0x00000000), id: PAL/DK A2
(00000003000000e0), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(866800, 161);

        #
        # Firmware 15, type: STD FW    MTS (0x00000004), id: PAL/DK A2
(00000003000000e0), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(866968, 169);

        #
        # Firmware 16, type: STD FW    (0x00000000), id: PAL/DK NICAM
(0000000c000000e0), size: 161
        #
```

```
        write_le32(0x00000000);                 # Type
        write_le64(0x0000000c, 0x000000e0);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(867144, 161);

        #
        # Firmware 17, type: STD FW    MTS (0x00000004), id: PAL/DK NICAM
(0000000c000000e0), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x0000000c, 0x000000e0);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(867312, 169);

        #
        # Firmware 18, type: STD FW    (0x00000000), id: SECAM/K1
(0000000000200000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(867488, 161);

        #
        # Firmware 19, type: STD FW    MTS (0x00000004), id: SECAM/K1
(0000000000200000), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(867656, 169);

        #
        # Firmware 20, type: STD FW    (0x00000000), id: SECAM/K3
(0000000004000000), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x04000000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(867832, 161);

        #
        # Firmware 21, type: STD FW    MTS (0x00000004), id: SECAM/K3
(0000000004000000), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x04000000);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(868000, 169);

        #
```

```
        # Firmware 22, type: STD FW    D2633 DTV6 ATSC (0x00010030), id:
(0000000000000000), size: 149
        #

        write_le32(0x00010030);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868176, 149);

        #
        # Firmware 23, type: STD FW    D2620 DTV6 QAM (0x00000068), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000068);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868336, 149);

        #
        # Firmware 24, type: STD FW    D2633 DTV6 QAM (0x00000070), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000070);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868488, 149);

        #
        # Firmware 25, type: STD FW    D2620 DTV7 (0x00000088), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000088);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868648, 149);

        #
        # Firmware 26, type: STD FW    D2633 DTV7 (0x00000090), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000090);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868800, 149);

        #
        # Firmware 27, type: STD FW    D2620 DTV78 (0x00000108), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000108);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
```

```
        write_le32(149);                        # Size
        write_hunk_fix_endian(868960, 149);

        #
        # Firmware 28, type: STD FW    D2633 DTV78 (0x00000110), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000110);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(869112, 149);

        #
        # Firmware 29, type: STD FW    D2620 DTV8 (0x00000208), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000208);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868648, 149);

        #
        # Firmware 30, type: STD FW    D2633 DTV8 (0x00000210), id:
(0000000000000000), size: 149
        #

        write_le32(0x00000210);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(149);                        # Size
        write_hunk_fix_endian(868800, 149);

        #
        # Firmware 31, type: STD FW    FM (0x00000400), id: (0000000000000000),
size: 135
        #

        write_le32(0x00000400);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le32(135);                        # Size
        write_hunk_fix_endian(869584, 135);

        #
        # Firmware 32, type: STD FW    (0x00000000), id: PAL/I
(0000000000000010), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00000010);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(869728, 161);

        #
        # Firmware 33, type: STD FW    MTS (0x00000004), id: PAL/I
(0000000000000010), size: 169
```

```
	#

	write_le32(0x00000004);			# Type
	write_le64(0x00000000, 0x00000010);	# ID
	write_le32(169);			# Size
	write_hunk_fix_endian(869896, 169);

	#
	# Firmware 34, type: STD FW    (0x00000000), id: SECAM/L AM
(0000001000400000), size: 169
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000010, 0x00400000);	# ID
	write_le32(169);			# Size
	write_hunk_fix_endian(870072, 169);

	#
	# Firmware 35, type: STD FW    (0x00000000), id: SECAM/L NICAM
(0000000c00400000), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x0000000c, 0x00400000);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(870248, 161);

	#
	# Firmware 36, type: STD FW    (0x00000000), id: SECAM/Lc
(0000000000800000), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000000, 0x00800000);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(870416, 161);

	#
	# Firmware 37, type: STD FW    (0x00000000), id: NTSC/M Kr
(0000000000008000), size: 161
	#

	write_le32(0x00000000);			# Type
	write_le64(0x00000000, 0x00008000);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(870584, 161);

	#
	# Firmware 38, type: STD FW    LCD (0x00001000), id: NTSC/M Kr
(0000000000008000), size: 161
	#

	write_le32(0x00001000);			# Type
	write_le64(0x00000000, 0x00008000);	# ID
	write_le32(161);			# Size
	write_hunk_fix_endian(870752, 161);
```

```
        #
        # Firmware 39, type: STD FW    LCD NOGD (0x00003000), id: NTSC/M Kr
(0000000000008000), size: 161
        #

        write_le32(0x00003000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(870920, 161);

        #
        # Firmware 40, type: STD FW    MTS (0x00000004), id: NTSC/M Kr
(0000000000008000), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(871088, 169);

        #
        # Firmware 41, type: STD FW    (0x00000000), id: NTSC PAL/M PAL/N
(000000000000b700), size: 161
        #

        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(871264, 161);

        #
        # Firmware 42, type: STD FW    LCD (0x00001000), id: NTSC PAL/M PAL/N
(000000000000b700), size: 161
        #

        write_le32(0x00001000);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(871432, 161);

        #
        # Firmware 43, type: STD FW    LCD NOGD (0x00003000), id: NTSC PAL/M
PAL/N (000000000000b700), size: 161
        #

        write_le32(0x00003000);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(871600, 161);

        #
        # Firmware 44, type: STD FW    (0x00000000), id: NTSC/M Jp
(0000000000002000), size: 161
        #
```

```
        write_le32(0x00000000);                 # Type
        write_le64(0x00000000, 0x00002000);     # ID
        write_le32(161);                        # Size
        write_hunk_fix_endian(871264, 161);

        #
        # Firmware 45, type: STD FW    MTS (0x00000004), id: NTSC PAL/M PAL/N
(000000000000b700), size: 169
        #

        write_le32(0x00000004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(871936, 169);

        #
        # Firmware 46, type: STD FW    MTS LCD (0x00001004), id: NTSC PAL/M
PAL/N (000000000000b700), size: 169
        #

        write_le32(0x00001004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(872112, 169);

        #
        # Firmware 47, type: STD FW    MTS LCD NOGD (0x00003004), id: NTSC PAL/M
PAL/N (000000000000b700), size: 169
        #

        write_le32(0x00003004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le32(169);                        # Size
        write_hunk_fix_endian(872288, 169);

        #
        # Firmware 48, type: SCODE FW  HAS IF (0x60000000), IF = 3.28 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3280);                       # IF
        write_le32(192);                        # Size
        write_hunk(811896, 192);

        #
        # Firmware 49, type: SCODE FW  HAS IF (0x60000000), IF = 3.30 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3300);                       # IF
        write_le32(192);                        # Size
        write_hunk(813048, 192);
```

```
        #
        # Firmware 50, type: SCODE FW  HAS IF (0x60000000), IF = 3.44 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3440);                       # IF
        write_le32(192);                        # Size
        write_hunk(812280, 192);

        #
        # Firmware 51, type: SCODE FW  HAS IF (0x60000000), IF = 3.46 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3460);                       # IF
        write_le32(192);                        # Size
        write_hunk(812472, 192);

        #
        # Firmware 52, type: SCODE FW  DTV6 ATSC OREN36 HAS IF (0x60210020), IF
= 3.80 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60210020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(3800);                       # IF
        write_le32(192);                        # Size
        write_hunk(809784, 192);

        #
        # Firmware 53, type: SCODE FW  HAS IF (0x60000000), IF = 4.00 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4000);                       # IF
        write_le32(192);                        # Size
        write_hunk(812088, 192);

        #
        # Firmware 54, type: SCODE FW  DTV6 ATSC TOYOTA388 HAS IF (0x60410020),
IF = 4.08 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60410020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4080);                       # IF
        write_le32(192);                        # Size
        write_hunk(809976, 192);
```

```
        #
        # Firmware 55, type: SCODE FW  HAS IF (0x60000000), IF = 4.20 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4200);                       # IF
        write_le32(192);                        # Size
        write_hunk(811704, 192);

        #
        # Firmware 56, type: SCODE FW  MONO HAS IF (0x60008000), IF = 4.32 MHz
id: NTSC/M Kr (0000000000008000), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le16(4320);                       # IF
        write_le32(192);                        # Size
        write_hunk(808056, 192);

        #
        # Firmware 57, type: SCODE FW  HAS IF (0x60000000), IF = 4.45 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4450);                       # IF
        write_le32(192);                        # Size
        write_hunk(812664, 192);

        #
        # Firmware 58, type: SCODE FW  MTS LCD NOGD MONO IF HAS IF (0x6002b004),
IF = 4.50 MHz id: NTSC PAL/M PAL/N (000000000000b700), size: 192
        #

        write_le32(0x6002b004);                 # Type
        write_le64(0x00000000, 0x0000b700);     # ID
        write_le16(4500);                       # IF
        write_le32(192);                        # Size
        write_hunk(807672, 192);

        #
        # Firmware 59, type: SCODE FW  LCD NOGD IF HAS IF (0x60023000), IF =
4.60 MHz id: NTSC/M Kr (0000000000008000), size: 192
        #

        write_le32(0x60023000);                 # Type
        write_le64(0x00000000, 0x00008000);     # ID
        write_le16(4600);                       # IF
        write_le32(192);                        # Size
        write_hunk(807864, 192);

        #
```

```
        # Firmware 60, type: SCODE FW  DTV6 QAM DTV7 DTV78 DTV8 ZARLINK456 HAS
IF (0x620003e0), IF = 4.76 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x620003e0);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4760);                       # IF
        write_le32(192);                        # Size
        write_hunk(807288, 192);

        #
        # Firmware 61, type: SCODE FW  HAS IF (0x60000000), IF = 4.94 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(4940);                       # IF
        write_le32(192);                        # Size
        write_hunk(811512, 192);

        #
        # Firmware 62, type: SCODE FW  HAS IF (0x60000000), IF = 5.26 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5260);                       # IF
        write_le32(192);                        # Size
        write_hunk(810552, 192);

        #
        # Firmware 63, type: SCODE FW  MONO HAS IF (0x60008000), IF = 5.32 MHz
id: PAL/BG A2 NICAM (0000000f00000007), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x0000000f, 0x00000007);     # ID
        write_le16(5320);                       # IF
        write_le32(192);                        # Size
        write_hunk(810744, 192);

        #
        # Firmware 64, type: SCODE FW  DTV7 DTV78 DTV8 DIBCOM52 CHINA HAS IF
(0x65000380), IF = 5.40 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x65000380);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5400);                       # IF
        write_le32(192);                        # Size
        write_hunk(807096, 192);

        #
        # Firmware 65, type: SCODE FW  DTV6 ATSC OREN538 HAS IF (0x60110020), IF
```

```
= 5.58 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60110020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5580);                       # IF
        write_le32(192);                        # Size
        write_hunk(809592, 192);

        #
        # Firmware 66, type: SCODE FW  HAS IF (0x60000000), IF = 5.64 MHz id:
PAL/BG A2 (0000000300000007), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000003, 0x00000007);     # ID
        write_le16(5640);                       # IF
        write_le32(192);                        # Size
        write_hunk(808440, 192);

        #
        # Firmware 67, type: SCODE FW  HAS IF (0x60000000), IF = 5.74 MHz id:
PAL/BG NICAM (0000000c00000007), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x0000000c, 0x00000007);     # ID
        write_le16(5740);                       # IF
        write_le32(192);                        # Size
        write_hunk(808632, 192);

        #
        # Firmware 68, type: SCODE FW  HAS IF (0x60000000), IF = 5.90 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(5900);                       # IF
        write_le32(192);                        # Size
        write_hunk(810360, 192);

        #
        # Firmware 69, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.00 MHz
id: PAL/DK PAL/I SECAM/K3 SECAM/L SECAM/Lc NICAM (0000000c04c000f0), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x0000000c, 0x04c000f0);     # ID
        write_le16(6000);                       # IF
        write_le32(192);                        # Size
        write_hunk(808824, 192);

        #
        # Firmware 70, type: SCODE FW  DTV6 QAM ATSC LG60 F6MHZ HAS IF
(0x68050060), IF = 6.20 MHz id: (0000000000000000), size: 192
```

```
        #

        write_le32(0x68050060);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(6200);                       # IF
        write_le32(192);                        # Size
        write_hunk(809400, 192);

        #
        # Firmware 71, type: SCODE FW  HAS IF (0x60000000), IF = 6.24 MHz id:
PAL/I (0000000000000010), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000010);     # ID
        write_le16(6240);                       # IF
        write_le32(192);                        # Size
        write_hunk(808248, 192);

        #
        # Firmware 72, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.32 MHz
id: SECAM/K1 (0000000000200000), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le16(6320);                       # IF
        write_le32(192);                        # Size
        write_hunk(811320, 192);

        #
        # Firmware 73, type: SCODE FW  HAS IF (0x60000000), IF = 6.34 MHz id:
SECAM/K1 (0000000000200000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00200000);     # ID
        write_le16(6340);                       # IF
        write_le32(192);                        # Size
        write_hunk(809208, 192);

        #
        # Firmware 74, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.50 MHz
id: PAL/DK SECAM/K3 SECAM/L NICAM (0000000c044000e0), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x0000000c, 0x044000e0);     # ID
        write_le16(6500);                       # IF
        write_le32(192);                        # Size
        write_hunk(811128, 192);

        #
        # Firmware 75, type: SCODE FW  DTV6 ATSC ATI638 HAS IF (0x60090020), IF
= 6.58 MHz id: (0000000000000000), size: 192
        #
```

```
        write_le32(0x60090020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(6580);                       # IF
        write_le32(192);                        # Size
        write_hunk(807480, 192);

        #
        # Firmware 76, type: SCODE FW  HAS IF (0x60000000), IF = 6.60 MHz id:
PAL/DK A2 (00000003000000e0), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le16(6600);                       # IF
        write_le32(192);                        # Size
        write_hunk(809016, 192);

        #
        # Firmware 77, type: SCODE FW  MONO HAS IF (0x60008000), IF = 6.68 MHz
id: PAL/DK A2 (00000003000000e0), size: 192
        #

        write_le32(0x60008000);                 # Type
        write_le64(0x00000003, 0x000000e0);     # ID
        write_le16(6680);                       # IF
        write_le32(192);                        # Size
        write_hunk(810936, 192);

        #
        # Firmware 78, type: SCODE FW  DTV6 ATSC TOYOTA794 HAS IF (0x60810020),
IF = 8.14 MHz id: (0000000000000000), size: 192
        #

        write_le32(0x60810020);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(8140);                       # IF
        write_le32(192);                        # Size
        write_hunk(810168, 192);

        #
        # Firmware 79, type: SCODE FW  HAS IF (0x60000000), IF = 8.20 MHz id:
(0000000000000000), size: 192
        #

        write_le32(0x60000000);                 # Type
        write_le64(0x00000000, 0x00000000);     # ID
        write_le16(8200);                       # IF
        write_le32(192);                        # Size
        write_hunk(812856, 192);
}


sub extract_firmware {
        my $sourcefile_24 = "UDXTTM6000.sys";
        my $hash_24 = "cb9deb5508a5e150af2880f5b0066d78";
```

```
        my $outfile_24 = "xc3028-v24.fw";
        my $name_24 = "xc2028 firmware";
        my $version_24 = 516;
        my $nr_desc_24 = 77;
        my $out;

        my $sourcefile_27 = "hcw85bda.sys";
        my $hash_27 = "0e44dbf63bb0169d57446aec21881ff2";
        my $outfile_27 = "xc3028-v27.fw";
        my $name_27 = "xc2028 firmware";
        my $version_27 = 519;
        my $nr_desc_27 = 80;
        my $out;

        if (-e $sourcefile_24) {
                verify($sourcefile_24, $hash_24);

                open INFILE, "<$sourcefile_24";
                main_firmware_24($outfile_24, $name_24, $version_24,
$nr_desc_24);
                close INFILE;
        }

        if (-e $sourcefile_27) {
                verify($sourcefile_27, $hash_27);

                open INFILE, "<$sourcefile_27";
                main_firmware_27($outfile_27, $name_27, $version_27,
$nr_desc_27);
                close INFILE;
        }
}

extract_firmware;
printf "Firmwares generated.\n";
```