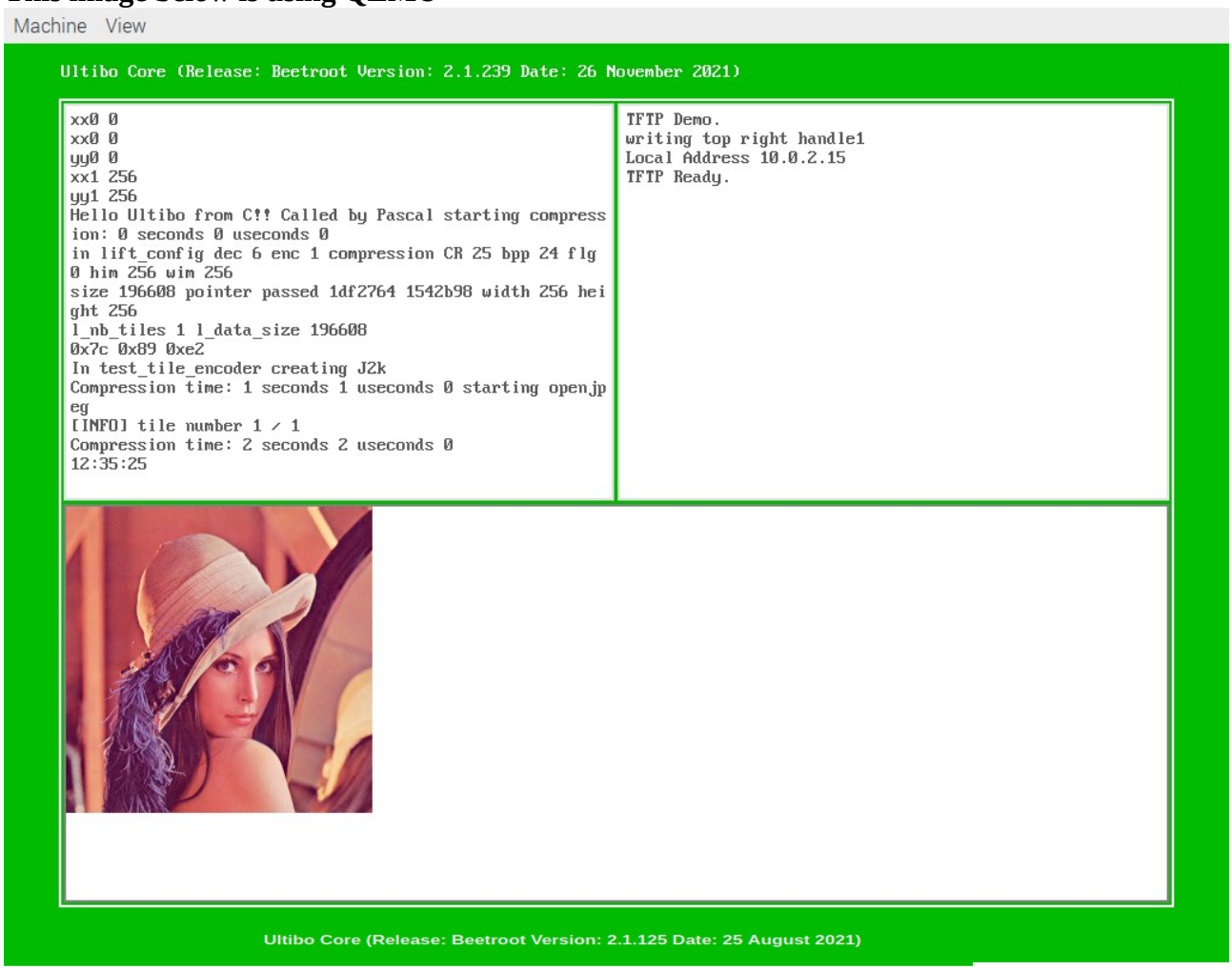
*****Default*****
Adding WiFI to DWT openjpeg
12/06/21
*****Default*****

Objective: To start with Ultibo members pjde & ric355 to create a Rpi WIFI that provides remote shell, webstatus. tftp, openjpeg 2000 over WIFI.

This will also integrate C with Ultibo.

This project starts with 2 of Ulitbo_Projects

This image below is using QEMU



General	**Platform**	
Platform		
Memory	Board Type:	BOARD_TYPE_RPI_ZERO_W
Heap Blocks	Board Model:	0
CPU	Board Serial:	0x000000005B2008C5
FPU	Board Revision:	0x009000C1
GPU		
RTL	Chip Revision:	0x00000000
Clock		
Locale	Firmware Revision:	0x608C2879 (1619798137)
Threading		
Thread List	Machine Type:	MACHINE_TYPE_BCM2708
Scheduler		
Devices	Memory Base:	0x00000000
Drivers	Memory Size:	536870912
Handles		
USB	Page Size:	4096
PCI	Large Page Size:	65536
MMC / SD / SDIO		
Network	Section Size:	1048576
Storage		
Filesystem	Power State	
Disk Cache	POWER_ID_MMC0:	POWER_STATE_ON
Keyboard	POWER_ID_MMC1:	POWER_STATE_OFF
Mouse	POWER_ID_MMC2:	POWER_STATE_OFF
Touch	POWER_ID_MMC3:	POWER_STATE_OFF

Status: Currently the Ultibo window appears and then disapeas.

Steps to create a kernel.img

Need to create a library of the openjpeg sources.

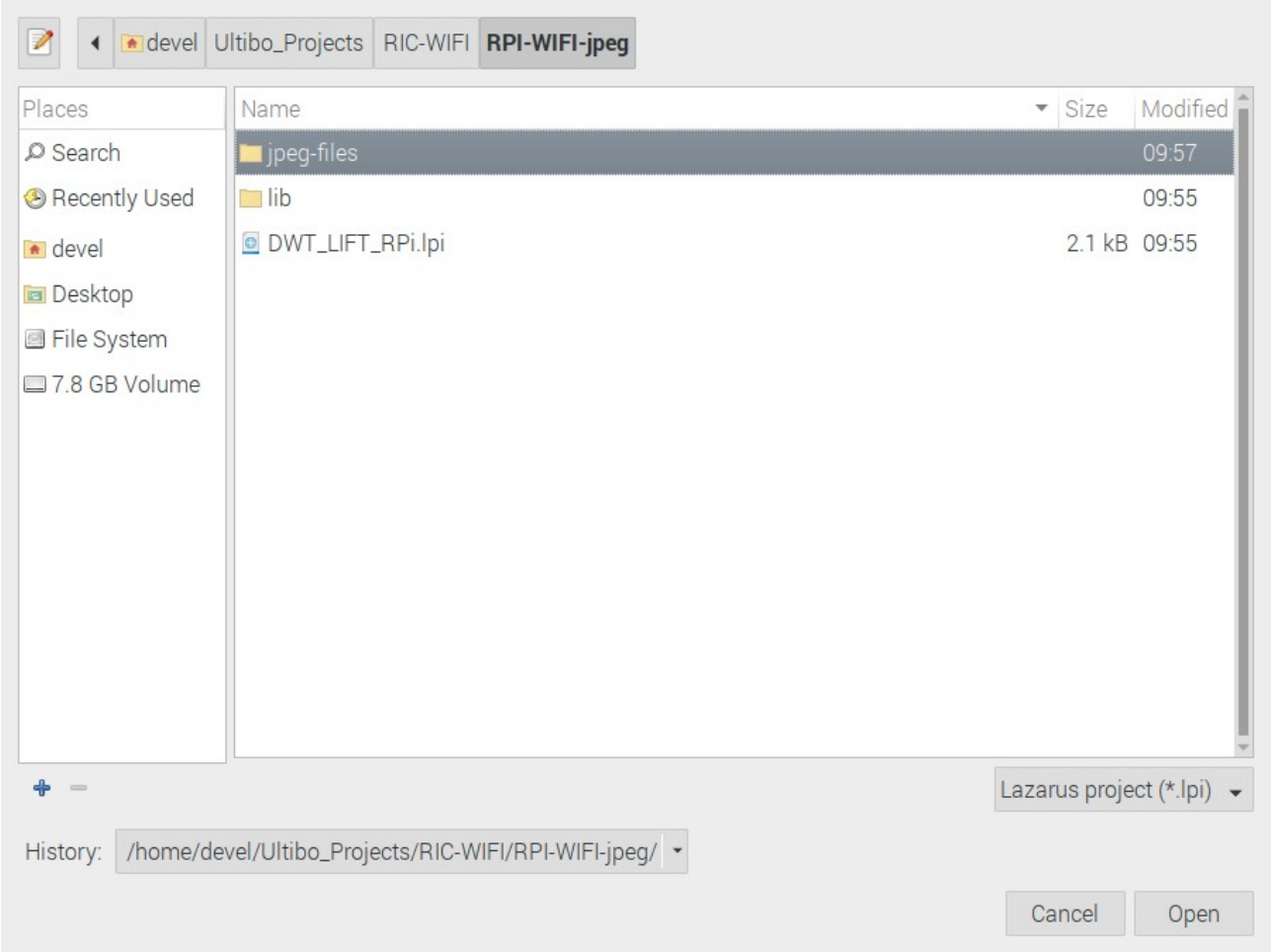
cd Ultibo_Projects/RIC-WIFI/src
The next step creates libopenjp2.a & libopenjp2_obj.txt
./compile_ultibo.sh
The word count here should be 22
the word count in /home/pi/jpeg-2000-test/bare-metal/openjp
when ./libbuild.sh is executed should be 22
22 22 182 libopenjp2_obj.txt

Need to create a library for Ultibo using libopenjp2.a from previous step.

cd ../RPI-WIFI-jpeg/
./libbuild.sh
dwtlift.c: In function 'decompress':
dwtlift.c:658:3: warning: implicit declaration of function 'octave_write_byte'; did you mean 'opj_write_tile'? [-Wimplicit-function-declaration]

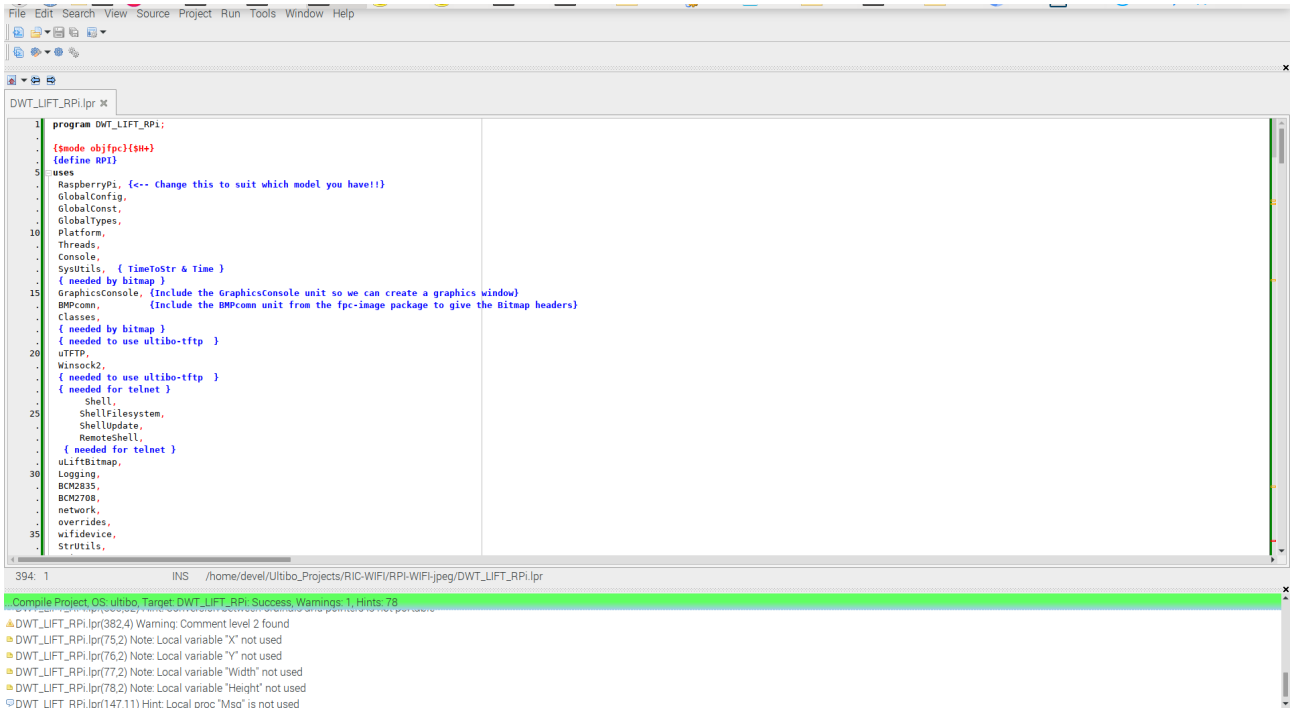
```
octave_write_byte(r_decompress_fn,r_decompress,da_x1*da_y1);
^~~~~~~~~~~~~~~~~~
opj_write_tile
```

Using Lazarus IDE (Ultibo-Edition)
From the main menu Project/Open



Depress Open

From the main menu Run/Compile



The green bar indicates the kernel.img was created.

https://ultibo.org/forum/viewtopic.php?f=10&t=1604

Using a new approach, starting with wifi.lpr, which I had already added
TFTP support. Now a the top window was split into two windows
topwindow := ConsoleWindowCreate(ConsoleDeviceGetDefault,
CONSOLE_POSITION_TOPLEFT,TRUE);
jpegHandle:=ConsoleWindowCreate(ConsoleDeviceGetDefault,CONSOLE_POSITION_TOPRIGHT,True)
https://github.com/develone/Ultibo_Proj ... _RPi3.lpr

https://github.com/develone/Ultibo_Proj ... T_RPi.lpr
Openjpeg WIFI Demo
xx0 0
xx0 0
xx0 0
yy0 0
xx1 256
yy1 256
In the Top Left Window the WIFI IP
In Bottom Network information.

telnet xx.xx.xx.xx

```
Ultibo Core (Release: Beetroot Version: 2.1.239 Date: 26 November 2021)
 (Type HELP for a list of available commands)
  Directory of C:\

 29-11-21 16:34:40               5442  fixup4.dat
 29-11-21 16:34:40               7313  fixup.dat
```

```
 5-12-21 16:53:04       <DIR>          firmware
 5-12-21 18:55:54             3140500  kernel7.img
 4-12-21 13:29:46              196730  MyBitmap.bmp
 4-12-21 13:28:40                  24  testfile
 5-12-21 14:36:58             3095412  kernel.img
 29-11-21 16:34:40            2231712  start4.elf
 29-11-21 16:34:42            2955936  start.elf
 4-12-21 13:28:40                  24  256com
 4-12-21 13:29:04                  24  256decom
 29-11-21 16:34:36              52456  bootcode.bin
 29-11-21 20:21:14                 56  cmdline.txt
 29-11-21 20:18:36                 13  config.txt
 4-12-21 13:29:46              196730  lena_rgb_256.bmp
          14 file(s) 11882372 bytes
          1 dir(s)
```

Getting Transfer timed out messages on large files using tftp between RPi & RPi3B+ WIFI tftp and RPi4B running Raspberry Pi O/S BullsEye.
In the Top Right Window I now see the following:

tftp xx.xx.xx.xx

```
tftp> binary
tftp> trace on
Packet tracing on.
tftp> get fixup4.dat
sent RRQ <file=fixup4.dat, mode=octet>
received DATA <block=1, 512 bytes>
sent ACK <block=1>
received DATA <block=2, 512 bytes>
sent ACK <block=2>
received DATA <block=3, 512 bytes>
sent ACK <block=3>
received DATA <block=4, 512 bytes>
sent ACK <block=4>
received DATA <block=5, 512 bytes>
sent ACK <block=5>
received DATA <block=6, 512 bytes>
sent ACK <block=6>
received DATA <block=7, 512 bytes>
sent ACK <block=7>
received DATA <block=8, 512 bytes>
sent ACK <block=8>
received DATA <block=9, 512 bytes>
sent ACK <block=9>
received DATA <block=10, 512 bytes>
sent ACK <block=10>
received DATA <block=11, 322 bytes>
Received 5442 bytes in 0.1 seconds
tftp> get fixup.dat
sent RRQ <file=fixup.dat, mode=octet>
received DATA <block=1, 512 bytes>
sent ACK <block=1>
received DATA <block=2, 512 bytes>
sent ACK <block=2>
received DATA <block=3, 512 bytes>
sent ACK <block=3>
received DATA <block=4, 512 bytes>
sent ACK <block=4>
```

```
received DATA <block=5, 512 bytes>
sent ACK <block=5>
received DATA <block=6, 512 bytes>
sent ACK <block=6>
received DATA <block=7, 512 bytes>
sent ACK <block=7>
received DATA <block=8, 512 bytes>
sent ACK <block=8>
received DATA <block=9, 512 bytes>
sent ACK <block=9>
received DATA <block=10, 512 bytes>
sent ACK <block=10>
received DATA <block=11, 512 bytes>
sent ACK <block=11>
received DATA <block=12, 512 bytes>
sent ACK <block=12>
received DATA <block=13, 512 bytes>
sent ACK <block=13>
received DATA <block=14, 512 bytes>
sent ACK <block=14>
received DATA <block=15, 145 bytes>
Received 7313 bytes in 0.1 seconds
tftp> get MyBitmap.bmp
sent RRQ <file=MyBitmap.bmp, mode=octet>
received DATA <block=1, 512 bytes>
sent ACK <block=1>
received DATA <block=2, 512 bytes>
sent ACK <block=2>
received DATA <block=3, 512 bytes>
sent ACK <block=3>
received DATA <block=4, 512 bytes>
sent ACK <block=4>
received DATA <block=5, 512 bytes>
sent ACK <block=5>
received DATA <block=6, 512 bytes>
sent ACK <block=6>
received DATA <block=7, 512 bytes>
sent ACK <block=7>
received DATA <block=8, 512 bytes>
sent ACK <block=8>
received DATA <block=9, 512 bytes>
sent ACK <block=9>
received DATA <block=10, 512 bytes>
sent ACK <block=10>
received DATA <block=11, 512 bytes>
sent ACK <block=11>
received DATA <block=12, 512 bytes>
sent ACK <block=12>
received DATA <block=13, 512 bytes>
sent ACK <block=13>
received DATA <block=14, 512 bytes>
sent ACK <block=14>
received DATA <block=15, 512 bytes>
sent ACK <block=15>
received DATA <block=16, 512 bytes>
sent ACK <block=16>
received DATA <block=17, 512 bytes>
sent ACK <block=17>
received DATA <block=18, 512 bytes>
sent ACK <block=18>
received DATA <block=19, 512 bytes>
sent ACK <block=19>
received DATA <block=20, 512 bytes>
sent ACK <block=20>
```

```
received DATA <block=21, 512 bytes>
sent ACK <block=21>
received DATA <block=22, 512 bytes>
sent ACK <block=22>
received DATA <block=23, 512 bytes>
sent ACK <block=23>
received DATA <block=24, 512 bytes>
sent ACK <block=24>
received DATA <block=25, 512 bytes>
sent ACK <block=25>
received DATA <block=26, 512 bytes>
sent ACK <block=26>
received DATA <block=27, 512 bytes>
sent ACK <block=27>
received DATA <block=28, 512 bytes>
sent ACK <block=28>
received DATA <block=29, 512 bytes>
sent ACK <block=29>
received DATA <block=30, 512 bytes>
sent ACK <block=30>
received DATA <block=31, 512 bytes>
sent ACK <block=31>
sent ACK <block=31>
sent ACK <block=31>
sent ACK <block=31>
sent ACK <block=31>
Transfer timed out.

tftp> get start4.elf
sent RRQ <file=start4.elf, mode=octet>
received DATA <block=1, 512 bytes>
sent ACK <block=1>
received DATA <block=2, 512 bytes>
sent ACK <block=2>
received DATA <block=3, 512 bytes>
sent ACK <block=3>
received DATA <block=4, 512 bytes>
sent ACK <block=4>
received DATA <block=5, 512 bytes>
sent ACK <block=5>
received DATA <block=6, 512 bytes>
sent ACK <block=6>
received DATA <block=7, 512 bytes>
sent ACK <block=7>
received DATA <block=8, 512 bytes>
sent ACK <block=8>
received DATA <block=9, 512 bytes>
sent ACK <block=9>
received DATA <block=10, 512 bytes>
sent ACK <block=10>
received DATA <block=11, 512 bytes>
sent ACK <block=11>
received DATA <block=12, 512 bytes>
sent ACK <block=12>
received DATA <block=13, 512 bytes>
sent ACK <block=13>
received DATA <block=14, 512 bytes>
sent ACK <block=14>
received DATA <block=15, 512 bytes>
sent ACK <block=15>
received DATA <block=16, 512 bytes>
sent ACK <block=16>
received DATA <block=17, 512 bytes>
sent ACK <block=17>
```

```
received DATA <block=18, 512 bytes>
sent ACK <block=18>
received DATA <block=19, 512 bytes>
sent ACK <block=19>
received DATA <block=20, 512 bytes>
sent ACK <block=20>
received DATA <block=21, 512 bytes>
sent ACK <block=21>
received DATA <block=22, 512 bytes>
sent ACK <block=22>
received DATA <block=23, 512 bytes>
sent ACK <block=23>
received DATA <block=24, 512 bytes>
sent ACK <block=24>
received DATA <block=25, 512 bytes>
sent ACK <block=25>
received DATA <block=26, 512 bytes>
sent ACK <block=26>
received DATA <block=27, 512 bytes>
sent ACK <block=27>
received DATA <block=28, 512 bytes>
sent ACK <block=28>
received DATA <block=29, 512 bytes>
sent ACK <block=29>
received DATA <block=30, 512 bytes>
sent ACK <block=30>
received DATA <block=31, 512 bytes>
sent ACK <block=31>
received DATA <block=32, 512 bytes>
sent ACK <block=32>
received DATA <block=33, 512 bytes>
sent ACK <block=33>
received DATA <block=34, 512 bytes>
sent ACK <block=34>
received DATA <block=35, 512 bytes>
sent ACK <block=35>
received DATA <block=36, 512 bytes>
sent ACK <block=36>
received DATA <block=37, 512 bytes>
sent ACK <block=37>
received DATA <block=38, 512 bytes>
sent ACK <block=38>
received DATA <block=39, 512 bytes>
sent ACK <block=39>
received DATA <block=40, 512 bytes>
sent ACK <block=40>
received DATA <block=41, 512 bytes>
sent ACK <block=41>
received DATA <block=42, 512 bytes>
sent ACK <block=42>
received DATA <block=43, 512 bytes>
sent ACK <block=43>
received DATA <block=44, 512 bytes>
sent ACK <block=44>
received DATA <block=45, 512 bytes>
sent ACK <block=45>
received DATA <block=46, 512 bytes>
sent ACK <block=46>
received DATA <block=47, 512 bytes>
sent ACK <block=47>
received DATA <block=48, 512 bytes>
sent ACK <block=48>
received DATA <block=49, 512 bytes>
sent ACK <block=49>
```

```
received DATA <block=50, 512 bytes>
sent ACK <block=50>
received DATA <block=51, 512 bytes>
sent ACK <block=51>
received DATA <block=52, 512 bytes>
sent ACK <block=52>
received DATA <block=53, 512 bytes>
sent ACK <block=53>
received DATA <block=54, 512 bytes>
sent ACK <block=54>
received DATA <block=55, 512 bytes>
sent ACK <block=55>
received DATA <block=56, 512 bytes>
sent ACK <block=56>
received DATA <block=57, 512 bytes>
sent ACK <block=57>
received DATA <block=58, 512 bytes>
sent ACK <block=58>
received DATA <block=59, 512 bytes>
sent ACK <block=59>
received DATA <block=60, 512 bytes>
sent ACK <block=60>
received DATA <block=61, 512 bytes>
sent ACK <block=61>
sent ACK <block=61>
sent ACK <block=61>
sent ACK <block=61>
sent ACK <block=61>
Transfer timed out.

tftp> get 256com
sent RRQ <file=256com, mode=octet>
received DATA <block=1, 24 bytes>
Received 24 bytes in 0.0 seconds
tftp> get 256decom
sent RRQ <file=256decom, mode=octet>
received DATA <block=1, 24 bytes>
Received 24 bytes in 0.0 seconds
tftp> quit
```

These are source I used with Lazarus IDE (Ultibo Edition)

https://github.com/develone/Ultibo_Proj ... T_RPi3.lpr

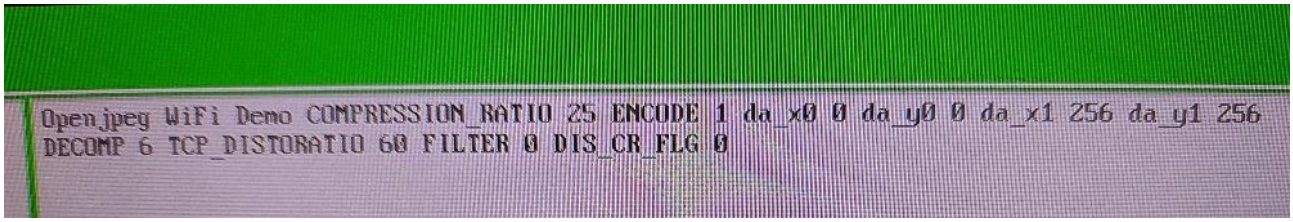
https://github.com/develone/Ultibo_Proj ... FT_RPi.lpr

Removing the trace in the files cmdsftp helps in the kernel transfer using tftp.
https://github.com/develone/Ultibo_Proj ... g/cmdstftp

https://github.com/develone/Ultibo_Proj ... g/cmdstftp
The kernel transfer works better on RPi Zero than RPi3B+
This is to RPi-Zero W
tftp xx.xx.xx.xx < cmdstftp
tftp> tftp> Sent 3474492 bytes in 129.3 seconds
This is to RPi3B+
tftp xx.xx.xx.xx < cmdstftp
tftp> tftp> Transfer timed out.

Working files used in testing WIFI on RPi Zero -W & RPi3B+
https://github.com/develone/Ultibo_Proj ... FI/wkgrpi3
You need a cmdline.txt file or your site.
WIFISCAN=1 SSID=<SSID> KEY=<KEY> COUNTRY=<COUNTRY>
This is what you will get in right top window. These are what will be used by C to do openjpeg.

At the time of my last test WIFI, remote shell, TFTP, Webstatus are working on RPi Zero -W & RPi3B



On the RPi3B+ the openjpeg now works using WIFI

These projects now demostrate Ultibo WIFI calling C. with TFTP, remote shell,
Webstatus HTTP.
This is on the RPi Zero-W
In top left window C debug
Hello Ultibo from C!!
.
.
.
In test_tile_encoder creating J2k
Exception: Undefined instruction at 003011FB8

The file 30-12-99 00:00:21 7848 test.j2k
Which is the openjpeg results ok is created.
The file 4-12-21 13:29:46 196730 MyBitmap.bmp
was compresed 25:1 which is the first binary value in the file
256com and displayed in the top right window as COMPRESSION_RATIO.

This error Exception: Undefined instruction at 003011FB8
does not occur on the RPi3B+.
30-12-99 00:00:21 7848 test.j2k
Ultibo Core (Release: Beetroot Version: 2.1.239 Date: 26 November 2021)
(Type HELP for a list of available commands)
Directory of C:\

29-11-21 16:34:40 5442 fixup4.dat
29-11-21 16:34:40 7313 fixup.dat
5-12-21 16:53:04 <DIR> firmware
6-12-21 16:44:30 3519580 kernel7.img
4-12-21 13:29:46 196730 MyBitmap.bmp
4-12-21 13:28:40 24 testfile
5-12-21 14:36:58 3095412 kernel.img
29-11-21 16:34:40 2231712 start4.elf

29-11-21 16:34:42 2955936 start.elf
4-12-21 13:28:40 24 256com
4-12-21 13:29:04 24 256decom
29-11-21 16:34:36 52456 bootcode.bin
29-11-21 20:21:14 56 cmdline.txt
29-11-21 20:18:36 13 config.txt
4-12-21 13:29:46 196730 lena_rgb_256.bmp
30-12-99 00:00:21 7848 test.j2k
15 file(s) 12269300 bytes
1 dir(s)
The C disasmbly is found at
https://raw.githubusercontent.com/devel ... g/dis.txt

https://raw.githubusercontent.com/devel ... penjp2.txt

On a RPi4B+ git@github.com:develone/openjpeg.git which has
opj_decompress.
196730 MyBitmap.bmp
7848 test.j2k

tftp the file test.j2k

tftp xx.xx.xx.xx IP of RPi3B+ running Ultibo that read MyBitmap.bmp and compressed it 25:1
tftp> binary
tftp> get test.j2k
Received 7848 bytes in 1.2 seconds
tftp> quit

./build/bin/opj_decompress -i test.j2k -o wifi.bmp
[INFO] Start to read j2k main header (0).
[INFO] Main header has been correctly decoded.
[INFO] No decoded area parameters, set the decoded area to the whole image
[INFO] Header of tile 1 / 1 has been read.
[INFO] Generated Outfile wifi.bmp
decode time: 24 ms
lena_rgb_256.bmp -> MyBitmap.bmp
MyBitmap.bmp -> test.j2k
lena_rgb_256.bmp converted to png for upload to forum with gimp.
wifi.bmp converted to png for upload to forum with gimp

Steps for building openjpeg in folder t_ultibo.

git clone git@github.com:develone/openjpeg.git t_ultibo

cd t_ultibo/

mkdir build

cd build/

cmake ../

-- The C compiler identification is GNU 10.2.1
-- The CXX compiler identification is GNU 10.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Check if the system is big endian
-- Searching 16 bit integer
-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for stddef.h
-- Looking for stddef.h - found
-- Check size of unsigned short
-- Check size of unsigned short - done
-- Searching 16 bit integer - Using unsigned short
-- Check if the system is big endian - little endian
-- Looking for string.h
-- Looking for string.h - found
-- Looking for memory.h
-- Looking for memory.h - found
-- Looking for stdlib.h
-- Looking for stdlib.h - found
-- Looking for stdio.h
-- Looking for stdio.h - found
-- Looking for math.h
-- Looking for math.h - found
-- Looking for float.h
-- Looking for float.h - found
-- Looking for time.h
-- Looking for time.h - found
-- Looking for stdarg.h
-- Looking for stdarg.h - found
-- Looking for ctype.h
-- Looking for ctype.h - found
-- Looking for assert.h
-- Looking for assert.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for inttypes.h
-- Looking for inttypes.h - found
-- Looking for strings.h
-- Looking for strings.h - found
-- Looking for sys/stat.h
-- Looking for sys/stat.h - found

```
-- Looking for unistd.h
-- Looking for unistd.h - found
-- Checking for 64-bit off_t
-- Checking for 64-bit off_t - present with _FILE_OFFSET_BITS=64
-- Checking for fseeko/ftello
-- Checking for fseeko/ftello - present
-- Large File support - found
-- Looking for include file malloc.h
-- Looking for include file malloc.h - found
-- Looking for _aligned_malloc
-- Looking for _aligned_malloc - not found
-- Looking for posix_memalign
-- Looking for posix_memalign - found
-- Looking for memalign
-- Looking for memalign - found
-- Found ZLIB: /usr/lib/arm-linux-gnueabihf/libz.so (found version "1.2.11")
-- Your system seems to have a Z lib available, we will use it to generate PNG lib
-- Found PNG: /usr/lib/arm-linux-gnueabihf/libpng.so (found version "1.6.37")
-- Your system seems to have a PNG lib available, we will use it
-- Found TIFF: /usr/lib/arm-linux-gnueabihf/libtiff.so (found version "4.2.0")
-- Your system seems to have a TIFF lib available, we will use it
-- Could NOT find LCMS2 (missing: LCMS2_LIBRARY LCMS2_INCLUDE_DIR)
-- Could NOT find LCMS (missing: LCMS_LIBRARY LCMS_INCLUDE_DIR)
-- LCMS2 or LCMS lib not found, activate BUILD_THIRDPARTY if you want build it
-- Configuring done
-- Generating done
-- Build files have been written to: /home/devel/t_ultibo/build

make
Scanning dependencies of target openjp2_static
[  1%] Building C object src/lib/openjp2/CMakeFiles/openjp2_static.dir/thread.c.o
[  2%] Building C object src/lib/openjp2/CMakeFiles/openjp2_static.dir/bio.c.o
                    .
                    .
                    .
[97%] Building C object src/bin/jp2/CMakeFiles/opj_decompress.dir/converttif.c.o
[ 98%] Building C object src/bin/jp2/CMakeFiles/opj_decompress.dir/convertpng.c.o
[100%] Linking C executable ../../../bin/opj_decompress
[100%] Built target opj_decompress
ls bin
libopenjp2.a   libopenjp2.so.2.3.1  opj_compress    opj_dump
libopenjp2.so  libopenjp2.so.7      opj_decompress
```