

*****Draft*****

Pico Lifting Step with CRC and head & tail code.

11/20/21

*****Draft*****

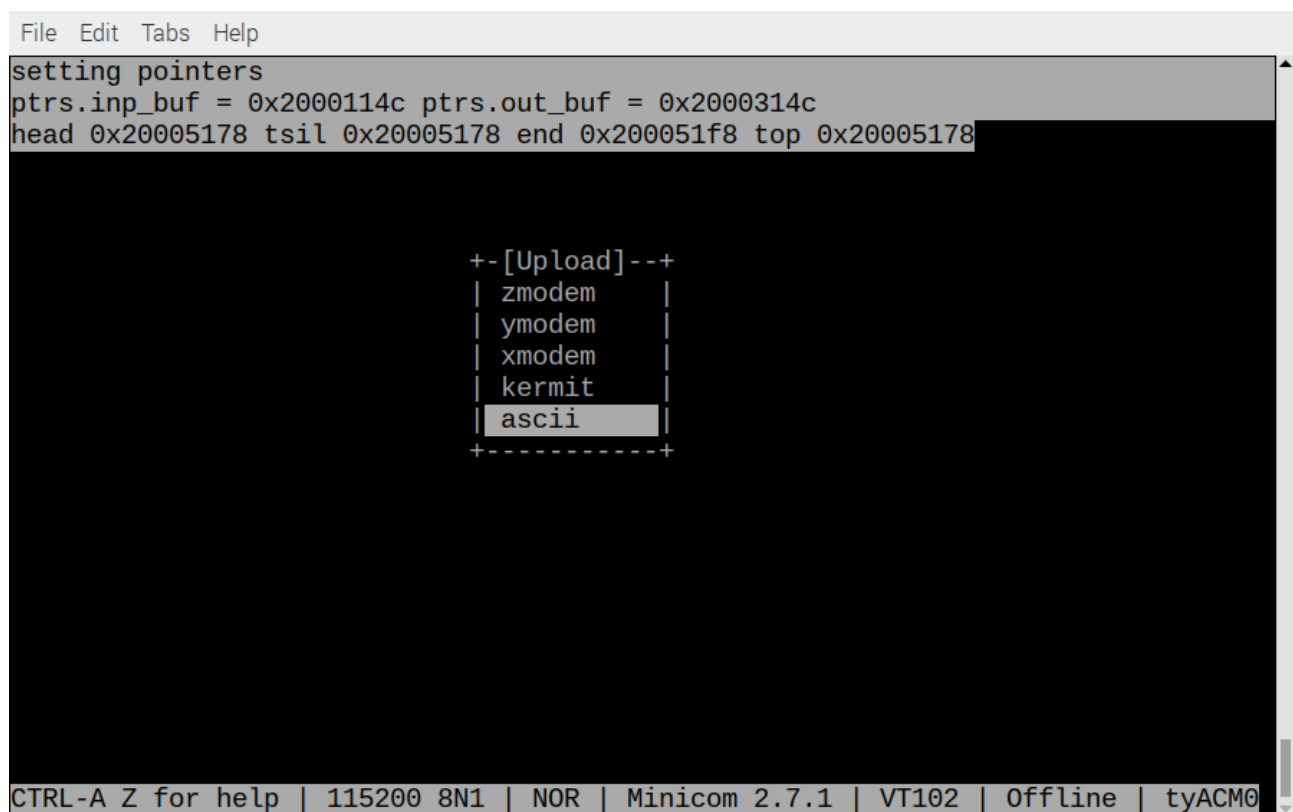
Now DBU4 performs the read and lifting step.

setting pointers

ptrs.inp_buf = 0x2000114c ptrs.out_buf = 0x2000314c

head 0x20005178 tsil 0x20005178 end 0x200051f8 top 0x20005178

ctrl A S



```
File Edit Tabs Help
setting pointers
ptrs.inp_buf = 0x2000114c ptrs.out_buf = 0x2000314c
head 0x20005178 tsil 0x20005178 end 0x200051f8 top 0x20005178

+--[Upload]--+
| zmodem      |
| ymodem      |
| xmodem      |
| kermit      |
| ascii       |
+-----+

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | tyACM0
```

Select the file a64.bin

```
File Edit Tabs Help

We+-----[Select a file for upload]-----+
|Directory: /root
OP| [..]
Co| [.cache]
Po| [.config]
| [.local]
Pr| [.ssh]
| [.vnc]
se| .bash_history
pt| .bash_history-01185.tmp
he| .bashrc
| .lessht
| .profile
| a64.bin
| aa.bin
| bb.bin
| cc.bin
|
| ( Escape to exit, Space to tag )
+-----+

[Goto] [Prev] [Show] [Tag] [Untag] [Okay]

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | tyACM0
```

Data is received.

```
File Edit Tabs Help

Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Aug 13 2017, 15:25:34.
Port /dev/ttyACM0, 07:02:23
+-----[ascii upload - Press CTRL-C to quit]-----+
Press CTRL|ASCII upload of "a64.bin"
|
|
|setting p|4.1 Kbytes transferred at 4160 CPS... Done.
ptrs.inp_|
head 0x20| READY: press any key to continue...
|
|
+-----+

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | tyACM0
```

User prompted

File Edit Tabs Help

Press CTRL-A Z for help on special keys

setting pointers

ptrs.inp_buf = 0x2000114c ptrs.out_buf = 0x2000314c

head 0x20005178 tsil 0x20005178 end 0x200051f8 top 0x20005178

104 104 105 107 105 103 103 103 103 104 104 103 102 103 103 105 107 106 105 101

107 93 96 80 88 69 52 62 64 64 62 64 78 71 52 66 94 67 56 59 57 53 51 62 64 67

recCRC 0x0 0x200051f2 0x200051f2 0x200051f8 0x20005178 0x200018118 146 148 157

recCRC 0xd 0x2000519a 0x2000519a 0x200051f8 0x20005178 0x20002fcc

0x7c

145 154 151 143 136 166 196 194 196 204 206 206 208 210 212 214 212 212 201 158

113 94 126 130 102 51 56 58 54 53 52 54 68 93 136 140 154 158 157 156 156 157 1

recCRC 0x19 0x200051da 0x200051da 0x200051f8 0x20005178 0x2000304c

0x7a

142 141 131 170 191 191 200 208 207 209 210 212 213 212 208 209 198 85 66 89 10

61 94 126 130 102 51 56 58 54 53 52 54 68 93 136 140 154 158 157 156 156 157 15

recCRC 0x46 0x20005199 0x20005199 0x200051f8 0x20005178 0x200030cc

0x71

142 141 131 170 191 191 200 208 207 209 210 212 213 212 208 209 198 85 66 89 10

61 142 53 51 55 54 50 50 52 48 103 149 146 161 162 157 157 155 153 154 156 156

recCRC 0x31 0x200051d9 0x200051d9 0x200051f8 0x20005178 0x2000314c

Command (1 = Send or 0 = Wait):

Enter 1

File Edit Tabs Help

0,1,-1,1,5,-5,1,-1,4,4,-3,-4,8,-5,7,3,13,8,17,-4,-10,-29,2,22,23,2,-1,0,17,-2,-,
1,1,0,-3,-9,5,0,6,3,-2,2,4,-5,5,-17,-1,-2,6,3,41,-42,34,17,-14,-14,-5,6,32,-16,,
0,0,1,3,0,-3,1,5,-8,5,3,-5,4,0,4,1,-1,-3,-2,-14,50,-23,-17,-1,1,-3,1,-20,-8,2,1,
3,2,-1,9,-5,-2,2,-23,11,1,-2,3,9,3,-9,3,1,-9,10,27,-19,4,2,1,0,0,-1,-1,0,-3,10,,
1,3,-5,19,1,-2,-2,24,1,1,8,5,-27,0,16,-15,2,23,-7,-19,5,3,0,-5,1,-3,1,1,6,6,-17,
1,2,-7,22,2,8,-8,-7,4,-3,11,0,9,-7,-18,5,20,-16,4,1,0,1,4,3,4,-2,2,5,-5,-4,6,0,,
2,3,3,14,-2,-9,12,4,-3,2,7,-3,3,-19,24,11,-30,18,8,1,1,-1,1,10,-14,-1,2,0,3,1,1,
1,1,1,-27,19,3,-5,1,5,6,2,-22,-8,1,-3,-15,-62,11,10,1,0,8,5,2,-8,0,0,-1,0,0,0,0,
2,1,0,7,-21,9,0,-1,4,-2,6,-70,18,0,1,30,-20,5,3,2,-2,4,-1,5,2,-3,1,1,0,-1,2,2,4,
0,2,0,1,-7,9,5,2,-3,6,-6,8,0,-1,1,6,-3,2,0,-5,-30,40,22,12,-4,6,0,0,0,2,-1,1,1,,
-1,1,1,2,2,-7,8,12,-2,0,-7,5,-3,-3,2,-3,16,-5,3,1,-14,19,10,-5,-44,0,-1,1,1,0,0,
12,-5,0,1,1,0,-6,-15,24,-12,9,2,19,7,5,29,-46,15,4,1,5,2,-2,0,-25,58,-4,0,0,1,-,
-5,3,1,-2,-1,1,0,3,-6,14,-6,1,-34,5,-5,10,-14,-6,9,1,-1,-1,-3,1,5,-41,20,3,-2,1,
-4,3,1,1,5,-6,1,10,-15,6,-1,48,18,-4,1,0,3,-1,0,1,1,-1,-1,-1,3,9,-2,5,2,2,5,0,6,
-1,0,1,2,0,6,-2,-48,33,-4,6,-61,56,-31,2,4,-1,4,-4,-2,3,1,2,2,-3,1,-2,0,18,-13,,
8,-2,3,0,-1,1,7,25,-2,10,7,37,-31,6,2,-2,-1,-1,3,0,1,1,1,3,-15,7,1,0,7,-32,6,5,,
13,-3,-1,1,-1,-1,6,6,4,9,3,-3,7,-2,-6,4,1,3,1,2,-1,0,0,1,11,4,3,-1,-2,4,-4,4,0,,
-33,11,13,-8,-9,12,-8,-15,8,-7,-3,-3,-1,0,5,0,0,-1,1,1,0,-1,0,2,13,-14,0,7,-1,-,
8,-9,-12,15,28,-17,4,3,4,-5,3,3,5,3,-9,3,1,2,-1,1,-1,1,1,-2,3,-20,-4,8,2,15,-37,
2,1,8,1,-11,4,0,-7,-14,8,4,0,1,0,-5,6,1,2,-3,-6,4,1,1,0,4,1,5,-1,0,-11,4,-10,22,
0,5,5,-8,3,1,-8,6,7,-2,0,0,1,1,6,-1,2,-2,4,4,0,1,-1,0,2,-1,-4,0,-1,5,-3,4,-24,2,
-9,8,-6,0,2,-11,9,3,2,0,0,-1,-1,1,4,-7,4,-18,5,3,1,-1,3,3,17,-15,7,-1,1,-1,3,8,,
86,-7,2,-1,-28,22,5,-2,0,0,2,0,0,0,-1,0,0,19,-4,-2,-1,0,0,19,-76,0,2,2,7,-7,5,-,

First 64 values found in the array a[]

161,157,156,157,159,162,162,166,172,165,148,117,93,94,94,94,
100,104,103,102,100,102,103,107,109,110,110,108,107,106,105,105,

102,101,105,102,94,82,80,84,85,82,77,66,60,57,53,52,
52,84,131,127,82,51,42,39,40,40,39,77,91,72,57,52,

```
gcc crc-ex-1.c -o crc-ex-1
```

```
./crc-ex-1 > xx.txt
```

```
sudo cp aa.bin /root
```

File written with crc-ex-1.c

aa.bin

```
00000000 A1 9D 9C 9D 9F A2 A2 A6 AC A5 94 75 5D 5E 5E 5E .....u]^^^
00000010 64 68 67 66 64 66 67 6B 6D 6E 6E 6C 6B 6A 69 69 dhgfdgkmmnlkjii
00000020 66 65 69 66 5E 52 50 54 55 52 4D 42 3C 39 35 34 feif^RPTURMB<954
00000030 34 54 83 7F 52 33 2A 27 28 28 27 4D 5B 48 39 34 4T..R3*>('M[H94
```

The file aa.bin is sent using minicom to the raspberry pi pico.

```
./crc-ex-1 > xx.txt data writen to xx.txt
```

```
a19d9c9d9fa2a2a6aca594755d5e5e5e646867666466676b6d6e6e6c6b6a69696665696665e525054
55524d423c3935343454837f52332a272828274d5b483934
0x3c
```

raspberry pi pico

0x3c

```
161 157 156 157 159 162 162 166 172 165 148 117 93 94 94 94 100 104 103 102 100 102 103 107
109 110 110 108 107 106 105
102 101 105 102 94 82 80 84 85 82 77 66 60 57 53 52 52 84 131 127 82 51 42 39 40 40 39 77 91
72 57 52
```

In the first shell.

```
export PATH=~/.local/openocd/bin/:$PATH
```

```
cd /opt/pico-examples/build/
```

```
openocd -f interface/raspberrypi-swd.cfg -f target/rp2040.cfg -c "program
hello_world/usb/hello_usb.elf verify reset exit"
```

Open On-Chip Debugger 0.10.0+dev-g71510a7-dirty (2021-08-15-17:08)

Licensed under GNU GPL v2

For bug reports, read

<http://openocd.org/doc/doxygen/bugs.html>

adapter speed: 1000 kHz

Info : Hardware thread awareness created

Info : Hardware thread awareness created

Info : RP2040 Flash Bank Command

Info : BCM2835 GPIO JTAG/SWD bitbang driver

Info : clock speed 1001 kHz

Info : SWD DPIDR 0x0bc12477

Info : SWD DLPIDR 0x00000001
Info : SWD DPIDR 0x0bc12477
Info : SWD DLPIDR 0x10000001
Info : rp2040.core0: hardware has 4 breakpoints, 2 watchpoints
Info : rp2040.core1: hardware has 4 breakpoints, 2 watchpoints
Info : starting gdb server for rp2040.core0 on 3333
Info : Listening on port 3333 for gdb connections
target halted due to debug-request, current mode: Thread
xPSR: 0xf1000000 pc: 0x000000ee msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0xf1000000 pc: 0x000000ee msp: 0x20041f00
** Programming Started **
Info : RP2040 B0 Flash Probe: 2097152 bytes @10000000, in 512 sectors

target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
Info : Writing 36864 bytes starting at 0x0
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
** Programming Finished **
** Verify Started **
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x00000178 msp: 0x20041f00
** Verified OK **
** Resetting Target **
shutdown command invoked

In a 2nd shell.

cd /opt/pico-examples/build/hello_world/usb/

make

```
[ 0%] Performing build step for 'ELF2UF2Build'
[100%] Built target elf2uf2
[ 0%] No install step for 'ELF2UF2Build'
[ 0%] Completed 'ELF2UF2Build'
[ 0%] Built target ELF2UF2Build
[ 0%] Built target bs2_default
[ 0%] Built target bs2_default_padded_checksummed_asm
[100%] Built target hello_usb
```

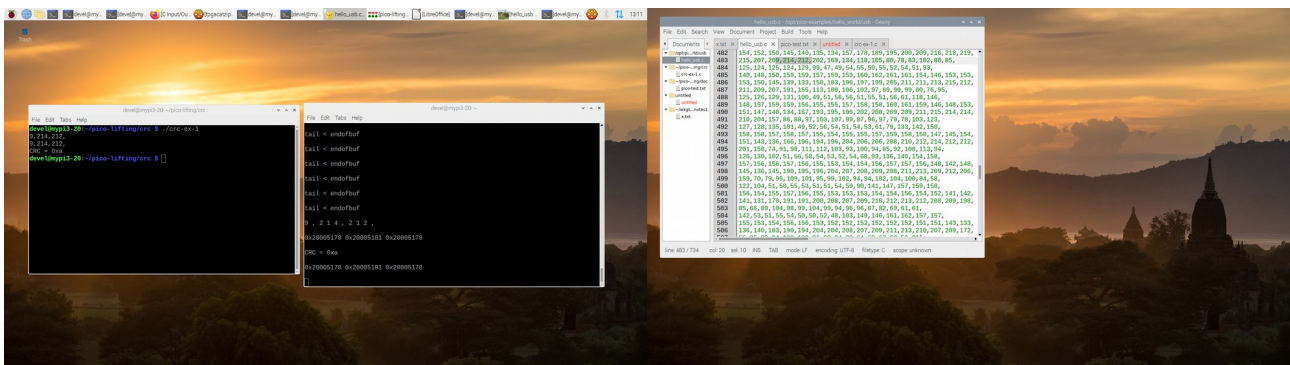
In a 3rd shell.

sudo minicom -s

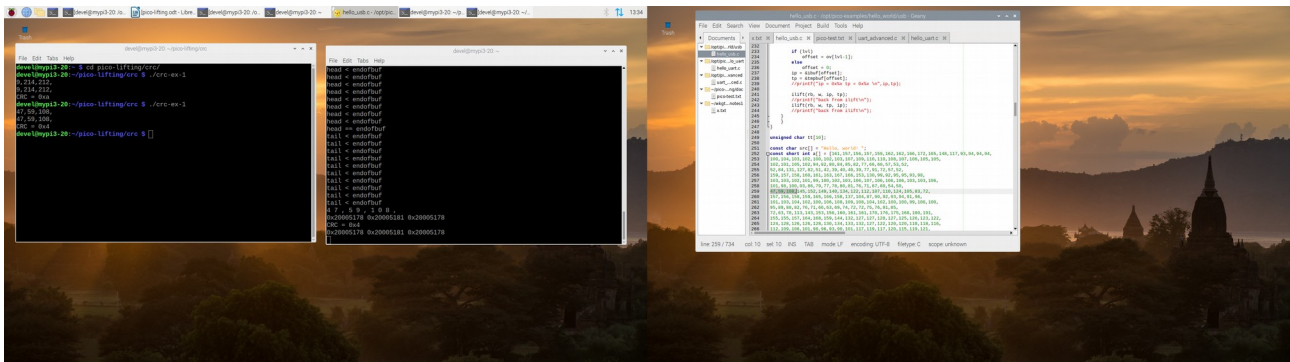
In a 4th shell

```
cd pico-lifting/crc/
./crc-ex-1
9,214,212,
9,214,212,
CRC = 0xa
```

and in 3rd shell.



```
./crc-ex-1
47,59,108,
47,59,108,
CRC = 0xa4
and in 3rd shell.
```



Now have all of code to receive data over the serial port checking the data received was correct using a CRC. Once all of the data is received compute the lifting step. And send over the serial port the results of the lifting step.

Sending 151,147,140,134,167,193,195,19 in groups of 10 and computing the CRC

Added control for sending results. Command (1 = Send or 0 = Wait):

The methods bump_head, bump_tail, dec_head, and dec_tail now require modification for receiving single bytes instead of short int.

bump_head, bump_tail, dec_head, and dec_tail code.

For testing head tail bump & dec

```
sleep_ms(50);
```

```
#define DEBUG 1
```

```
#define DEBUG1 0
```

```
#define DEBUG2 0
```

```
#define imgsize 512
```

```
#define DEBUG 0
```

```
#define DEBUG1 1
```

```
#define DEBUG2 0
```

```
#define imgsize 4096
```

```
sleep_ms(8000);
```

```
head = 0x2000114c tail = 0x2000114c end = 0x2000154c top = 0x2000114c
```

```
head = 0x2000114c tail = 0x2000114c 0x2000154c 0x2000114c
```

```
head < endofbuf
```

```
tail < endofbuf
```

```
head < endofbuf
```

```
tail < endofbuf
```

```
head < endofbuf
```

```
tail < endofbuf
```

```
head = 0x20001152 tail = 0x20001152 0x2000154c 0x2000114c
```

```
head < topofbuf
```

```
tail < topofbuf
```

```
head < topofbuf
```

```
tail < topofbuf
```

```
head < topofbuf
```

```
tail < topofbuf
```

```
head = 0x2000114c tail = 0x2000114c 0x2000154c 0x2000114c
```

Add code to do a CRC.

<https://www.pololu.com/docs/0J44/6.7.6>

Simple Example

The following example program shows how to compute a CRC byte in the C language. The outer loop processes each byte, and the inner loop processes each bit of those bytes. In the example main() routine, this is applied to generate the CRC byte in the message 0x83, 0x01, that was used in Section 6.5. The getCRC() function will work without modification in both Arduino and Orangutan programs. "pico-lifting/crc/crc-ex.c"

```
"gcc crc-ex.c -o crc-ex1"
```

```
./crc-ex  
83 1 17  
83 2 45
```

Advanced Example

The following example program shows a more efficient way to compute a CRC in the C language. The increased efficiency is achieved by pre-computing the CRCs of all 256 possible bytes and storing them in a lookup table, which can be in RAM, flash, or EEPROM. These table values are then XORed together based on the bytes of the message to get the final CRC. In the example main() routine, this is applied to generate the CRC byte in the message 0x83, 0x01, that was used in Section 6.5. "pico-lifting/crc/crc-ex-1.c"

The 2 bytes 0xd3, 0x01 result in CRC8 0x4e

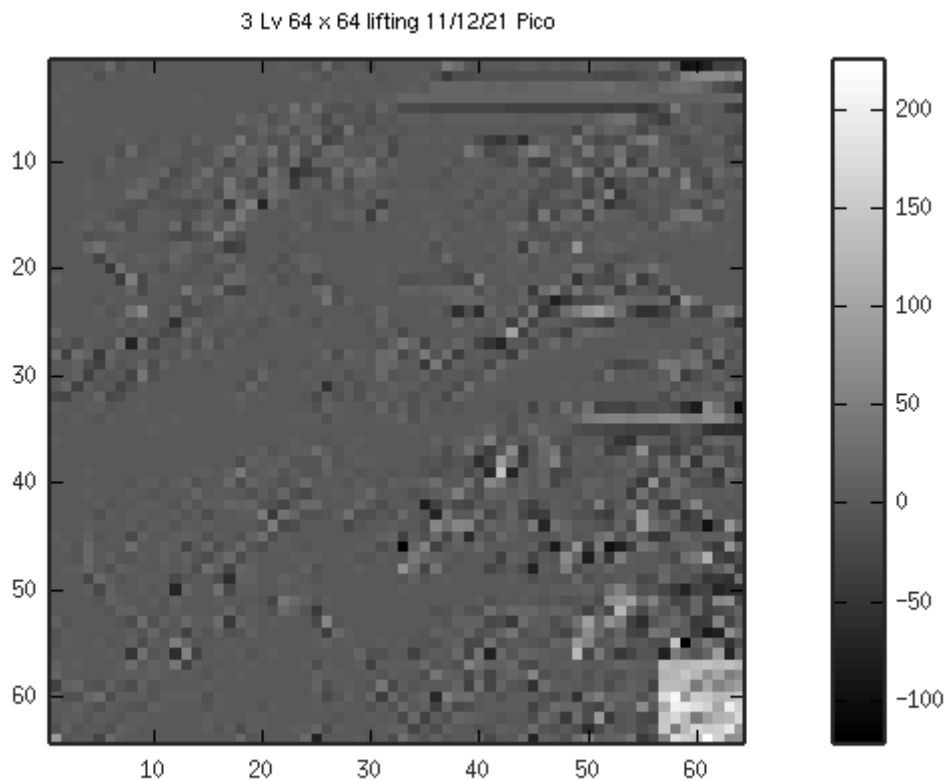
```
message[3] = {0xd3, 0x01, 0x00};
```

```
11001011 10000000 01110100
```

<https://github.com/lammertb/libcrc.git>
lena



The pico computes the 64 x 64 lifting step. In the image below



$\mu_2 = 246.411$

Add the C code to perform the lifting step DWT to the hello_usb.c

```
cp testfiles/64/* .
./buildpi_lift.sh
rm -rf dwt.bin ; ./pi_jpeg 0 1
```

```
octave
GNU Octave, version 4.4.1
Copyright (C) 2018 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.
```

Octave was configured for "arm-unknown-linux-gnueabi".

Additional information about Octave is available at <https://www.octave.org>.

Please contribute if you find this software useful.
For more information, visit <https://www.octave.org/get-involved.html>

Read <https://www.octave.org/bugs.html> to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

```
octave:1> rgb
```

Need to find how to insert the 4096 values in the program hello_usb.c
Create 4096 values of 8 bit from the file "r-8.bin" using myformat.

```

00000000 A1 9D 9C 9D 9F A2 A2 A6 AC A5 94 75 5D 5E 5E 5E .....u]^^
00000010 64 68 67 66 64 66 67 6B 6D 6E 6E 6C 6B 6A 69 69 dhgfdfgkmnnlkjii
00000020 66 65 69 66 5E 52 50 54 55 52 4D 42 3C 39 35 34 feif^RPTURMB<954
00000030 34 54 83 7F 52 33 2A 27 28 28 27 4D 5B 48 39 34 4T..R3*('([H94
00000040 9F 9D 9E A0 A1 A3 A7 A6 99 82 63 5C 5F 5F 5D 62 .....c\_]b
00000050 67 67 66 65 63 64 66 67 6A 6B 6A 6A 6A 67 67 6A ggfecdfgkjijggj
00000060 65 62 64 5D 56 4F 4D 4E 50 51 4C 47 43 3C 36 32 ebd]VOMNPQLGC<62
00000070 2F 3B 6C 91 98 95 8C 86 7A 70 6B 6E 7C 69 53 48 /;l.....zpkn|iSH
00000080 9D 9C 9E 9F A5 A6 9E 89 68 57 5A 5C 5D 5E 5B 60 .....hWZ\)]^
00000090 65 67 68 66 64 6A 6C 6D 6C 68 66 64 64 63 64 64 eghfdjlmhfdcdcd
000000A0 5F 59 58 52 4C 47 42 3F 45 4A 48 48 4B 4C 5B 55 _YXRLGB?EJHHKL[U
000000B0 48 3F 4E 71 8F 99 9C A0 A1 A1 AA B0 AF A8 B4 BF H?Nq.....
000000C0 9B 9B 9D A4 A8 9F 90 84 7F 7F 80 7F 7D 7E 7B 7A .....}~{z
000000D0 7C 80 7E 7E 80 82 86 85 84 7F 7A 78 78 76 76 74 |.~.....zxxvvt
000000E0 70 6D 6A 65 62 60 5D 5A 65 75 77 75 78 73 77 79 pmjeb`)Zeuwuxswy
000000F0 70 5B 48 46 46 3E 3F 47 4E 59 65 75 82 86 A2 B5 p[HFF>?GNYeu....
00000100 A0 A3 A6 A6 A5 A0 9F A2 A3 A4 A3 A1 9F 9E 9E 9F .....
00000110 9F 9F A2 A3 A3 A4 A5 A5 A3 A4 A2 A1 A2 A1 A0 A1 .....
00000120 A0 A2 A1 A0 9F 9E 9D 9D 9D A0 A1 A2 A3 9F 9B 9C .....
00000130 9D 98 94 95 97 96 96 95 92 8E 8A 88 87 85 85 86 .....
00000140 AA AA A5 A0 9F A1 A3 A5 A6 A7 A8 A7 A6 A5 A4 A7 .....
00000150 A8 AA AE AD AB AD AE AD AC AD AD AD AD AE AF AF .....
00000160 B1 B1 B3 B3 B2 B0 AD AF AD AA AB AE B1 B2 B0 AF .....

.
.
.

00000FB0 55 42 59 68 62 63 68 63 5E 60 60 57 52 45 3D 3D UBYhbhc^``WRE==
00000FC0 8E 35 33 37 36 32 32 34 30 67 95 92 A1 A2 9D 9D .53762240g.....
00000FD0 9B 99 9A 9C 9C 99 98 98 98 98 98 98 97 97 8F 85 .....
00000FE0 88 8C B7 BE C2 CC CC D0 CF D1 D3 D5 D2 CF D1 AC .....
00000FF0 37 55 59 5E 66 66 60 58 5E 52 40 3E 43 3F 3B 5B 7UY^ff`X^R@>C?;[

```

```
gcc myformat.c -o myformat
```

```

./myformat
161,157,156,157,159,162,162,166,172,165,148,117,93,94,94,94,
100,104,103,102,100,102,103,107,109,110,110,108,107,106,105,105,
102,101,105,102,94,82,80,84,85,82,77,66,60,57,53,52,
52,84,131,127,82,51,42,39,40,40,39,77,91,72,57,52,
159,157,158,160,161,163,167,166,153,130,99,92,95,95,93,98,
103,103,102,101,99,100,102,103,106,107,106,106,106,103,103,106,
101,98,100,93,86,79,77,78,80,81,76,71,67,60,54,50,
47,59,108,145,152,149,140,134,122,112,107,110,124,105,83,72,
157,156,158,159,165,166,158,137,104,87,90,92,93,94,91,96,
101,103,104,102,100,106,108,109,108,104,102,100,100,99,100,100,
95,89,88,82,76,71,66,63,69,74,72,72,75,76,91,85,
72,63,78,113,143,153,156,160,161,161,170,176,175,168,180,191,

.
.
.

```

85,66,89,104,98,99,104,99,94,96,96,87,82,69,61,61,
142,53,51,55,54,50,50,52,48,103,149,146,161,162,157,157,
155,153,154,156,156,153,152,152,152,152,152,151,151,143,133,
136,140,183,190,194,204,204,208,207,209,211,213,210,207,209,172,
55,85,89,94,102,102,96,88,94,82,64,62,67,63,59,91,

The following was added to "hello_usb.c"

```
struct PTRs {  
    /*This is the buffer for inp & output  
    2048 x 2048 = 4194304  
    256 x 256 = 65536  
    */  
    short int inbuf[4096*2];  
    short int *inp_buf;  
    short int *out_buf;  
    short int flag;  
    short int w;  
    short int h;  
    short int *fwd_inv;  
    short int fwd;  
    short int *red;  
} ptrs;
```

```
const short int a[] = {161,157,156,157,159,162,162,166,172,165,148,117,93,94,94,94,  
    .  
    .  
    .  
142,53,51,55,54,50,50,52,48,103,149,146,161,162,157,157,  
155,153,154,156,156,153,152,152,152,152,152,151,151,143,133,  
136,140,183,190,194,204,204,208,207,209,211,213,210,207,209,172,  
55,85,89,94,102,102,96,88,94,82,64,62,67,63,59,91};
```

```
for(i = 0; i < 4096;i++) ptrs.inp_buf[i] = a[i];
```

```
cd tmp  
git clone git@github.com:develone/pico-lifting.git  
cd pico-lifting  
cp testfiles/2048/* .  
./buildpi_lift.sh  
rm -f dwt.bin;./pi_jpeg 0 1
```

Next instead of reading the files and writing the results in dwt.bin The goal is use as serial tx & rx to send the data to the program. Then transmit the result over the serial tx.

The folder testfiles/2048/ has the files to perform a 2048 lifting step.
The folder testfiles/256/ has the files to perform a 256 lifting step.

The following command compiles the code ./buildpi_lift.sh

There is a define in pi_jpeg.c that turns off the debug

```
rm -f dwt.bin ; ./pi_jpeg 0 1
0x0 0x22048 0x1022048
ptrs.fwd_inv = 0x2022060
reading r.bin
fwd lifting step only
w = 0x800 ptrs.inp_buf wptr = 0x22048 alt = 0x1022048 ptrs.fwd_inverse = 0x2022060
ptrs.fwd_inverse = 0x1
starting red dwt
finished ted dwt
octave
GNU Octave, version 4.4.1
Copyright (C) 2018 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.
```

Octave was configured for "arm-unknown-linux-gnueabi".

Additional information about Octave is available at <https://www.octave.org>.

Please contribute if you find this software useful.

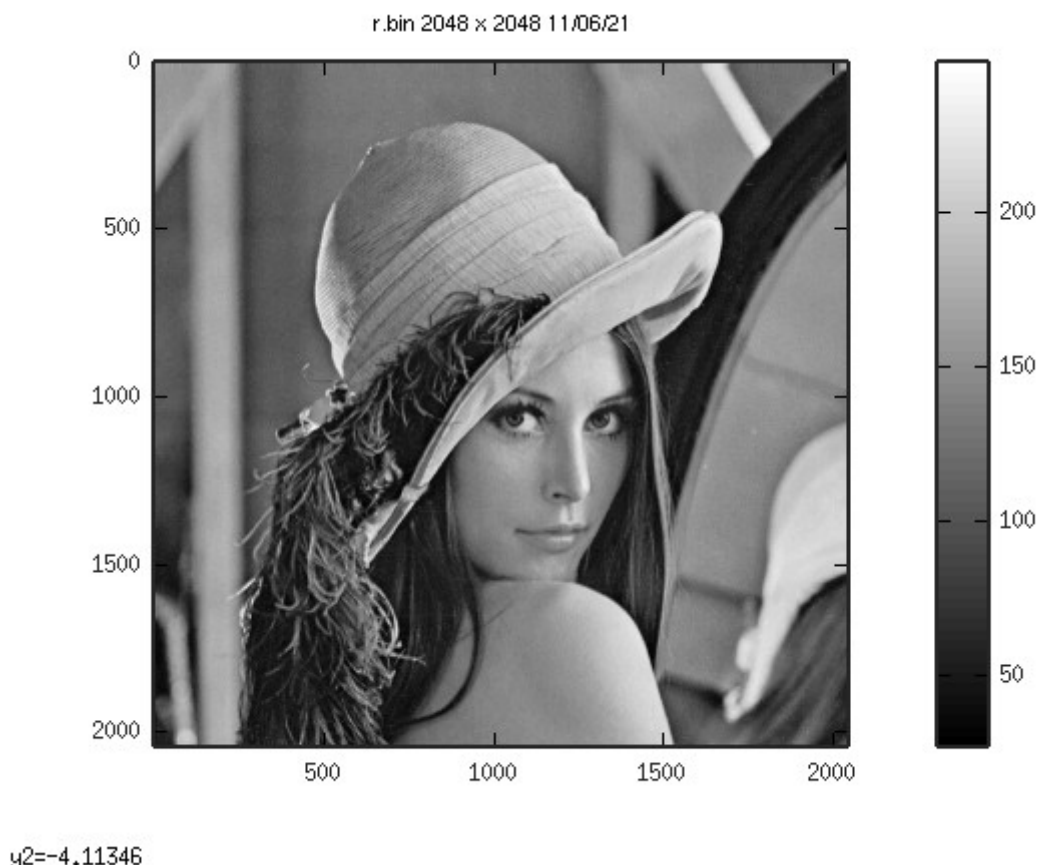
For more information, visit <https://www.octave.org/get-involved.html>

Read <https://www.octave.org/bugs.html> to learn how to submit bug reports.

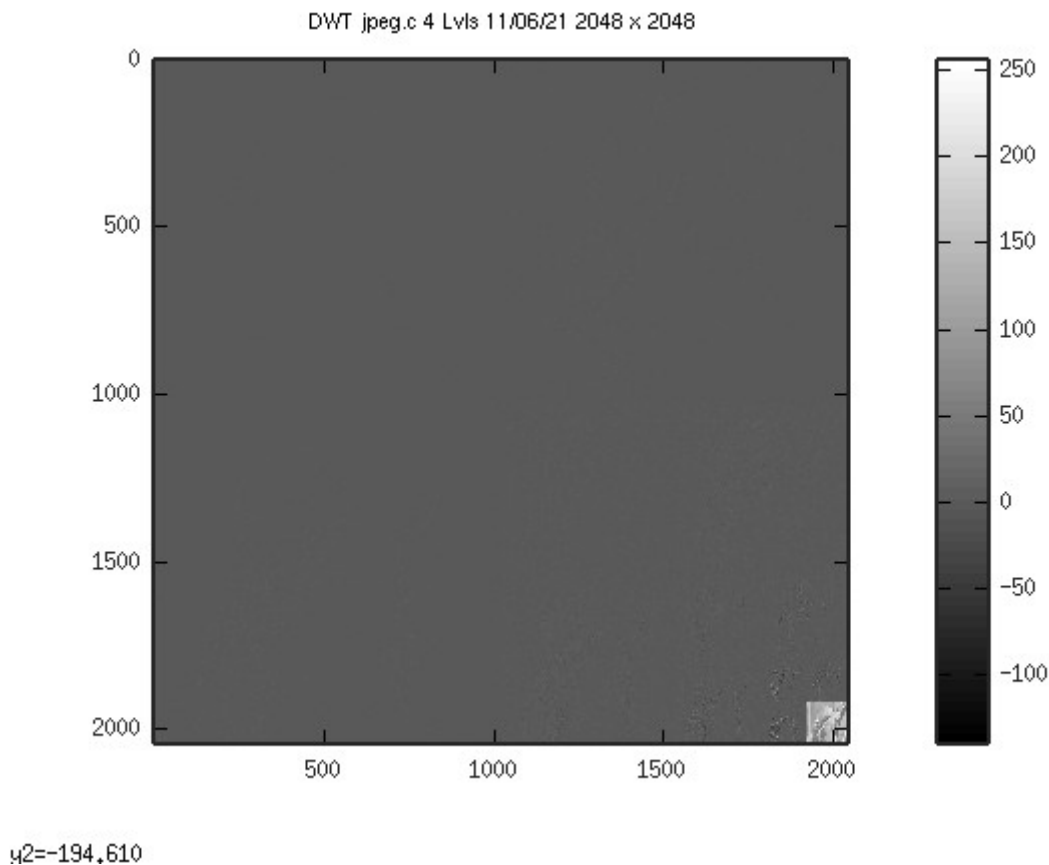
For information about changes from previous versions, type 'news'.

octave:1> rgb

The input was a pgm file 2048 x 2048



4 lvs 2048 x 2048 lifting step.



```
cp testfiles/256/* .
./buildpi_lift.sh
devel@mypi3-20:~/pico-lifting $ rm -f dwt.bin ; ./pi_jpeg 0 1
0x0 0x22048 0x62048
ptrs.fwd_inv = 0xa2060
reading r.bin
fwd lifting step only
w = 0x100 ptrs.inp_buf wptr = 0x22048 alt = 0x62048 ptrs.fwd_inverse = 0xa2060
ptrs.fwd_inverse = 0x1
starting red dwt
finished ted dwt
```

octave
GNU Octave, version 4.4.1
Copyright (C) 2018 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "arm-unknown-linux-gnueabi".

Additional information about Octave is available at <https://www.octave.org>.

Please contribute if you find this software useful.
For more information, visit <https://www.octave.org/get-involved.html>

Read <https://www.octave.org/bugs.html> to learn how to submit bug reports.

For information about changes from previous versions, type 'news'.

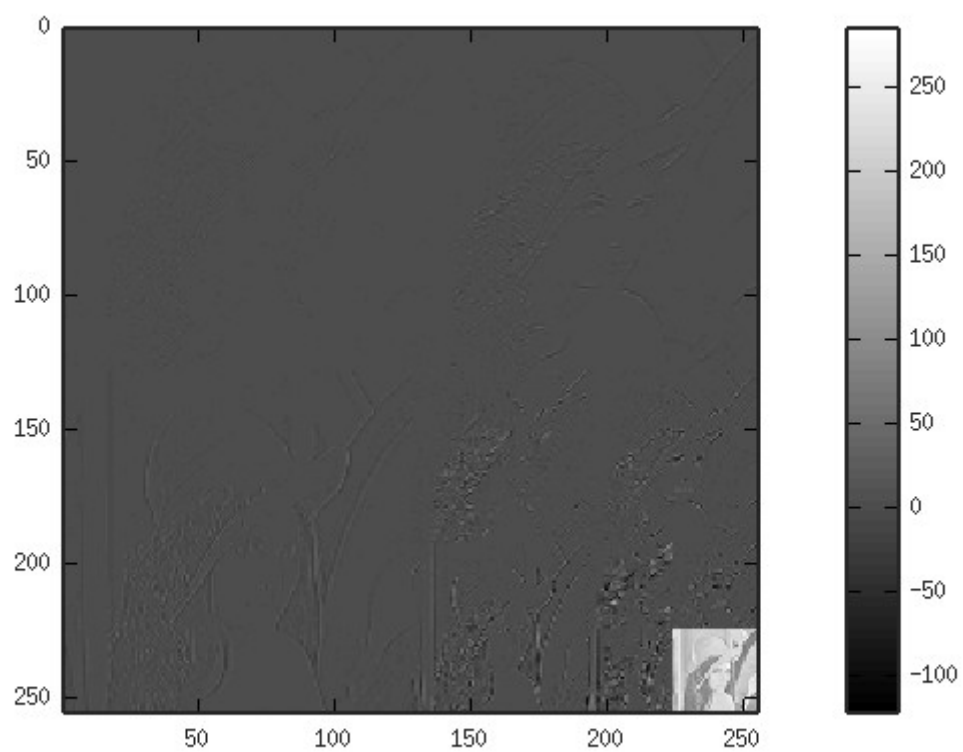
```
octave:1> rgb
```

The input was a pgm file 256 x 256



$\mu_2 = 174.001$

3 lvs 256 x 256 lifting step.



$\mu_2 = -178.095$