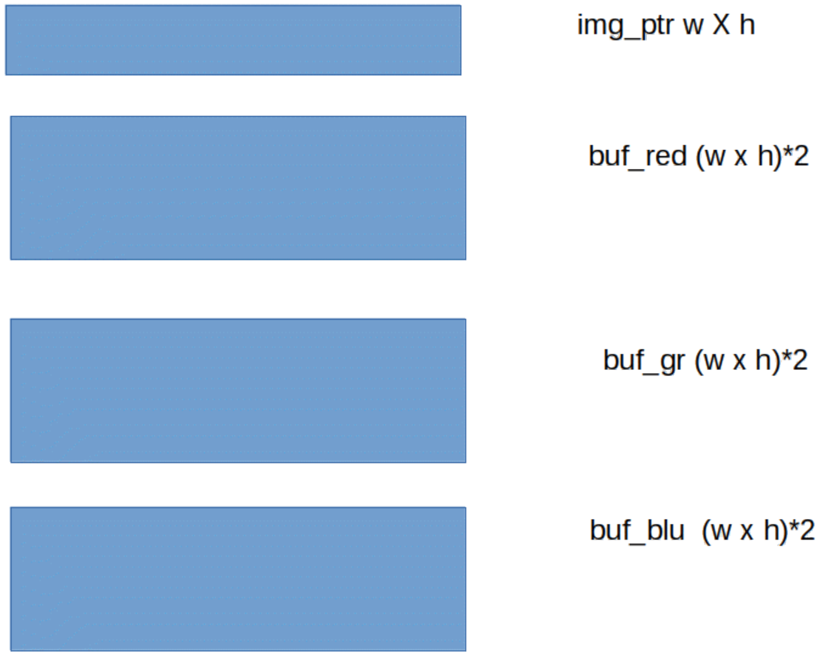
*****************Draft*****************
Zipcpu Catboard using catzip.bin
running jpeg which was compiled jpeg.c with lifting.c debug
11/29/18
*****************Draft*****************

The program “rd_bytes” created from “rd_bytes.c”  reads the file “rgb.bin” and writes “rgb_pack.bin” to be pushed to the catboard SDRAM using the program “arm-wrsdram rgb_pack.bin”.



JPEG memory buffers for r g b packed and r g b

The program arm-wrsdram reads the file rgb_pack.bin and writes to catboard sdram at address img_ptr allocated with the img_ptr = (int *)malloc(sizeof(int)*w*h);.  These 32 bit words have the r g b packed into a single word.
This requires the program to split 256 x 256 words to 3 buffers buf_red buf_red = (int *)malloc(sizeof(int)*w*h*2);, buf_grn buf_gr = (int *)malloc(sizeof(int)*w*h*2);, and buf_bl buf_bl = (int *)malloc(sizeof(int)*w*h*2);.

```
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./buildsdramscope.sh
#define         DUMPMEM  0x20098a8     img_ptr
#define         DUMPMEM  0x20498b0     buf_red
```

Transfer data from the RPi3B to SDRAM of the catboard.

```
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./arm-wrsdram rgb_pack.bin
```

The size of the buffer is 0x00ffff or 65535 words

```
READ-COMPLETE
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./arm-zipload -v ../board/jpeg
Halting the CPU
Memory regions:
        Block RAM: 01400000 - 01402000
        SDRAM     : 02000000 - 04000000
Loading: ../board/jpeg
Section 0: 02000000 - 0200989c
Writing to MEM: 02000000-0200989c
Clearing the CPUs registers
Setting PC to 02000000
The CPU should be fully loaded, you may now
start it (from reset/reboot) with:
> wbregs cpu 0x0f

CPU Status is: 0000060f

pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./arm-wbregs cpu 0x0f
```

Below is the jpeg debug output.  This shows the pointers that were allocated with the malloc. Five words of buf_red  at 3 different address are displayed.   Five words of buf_gr  at 3 different address are displayed. These are followed by, five words of buf_bl  at 3 different address are displayed.  Then the split 5 values of split r g b are displayed to verify the packed data is separated correctly.

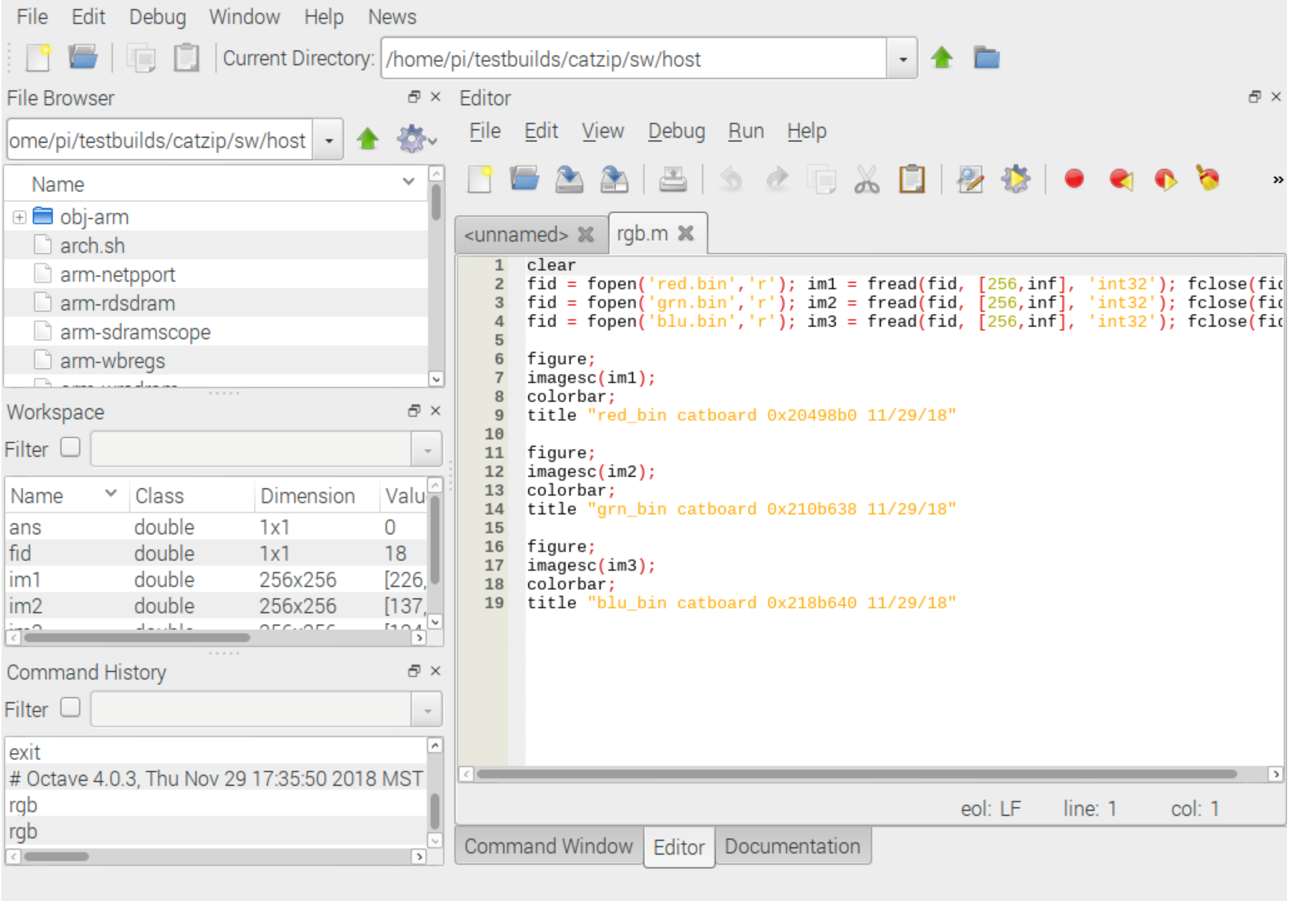
```
. img_ptr = 0x20098a8
. buf_red = 0x20498b0
. buf_gr = 0x210b638
. buf_bl = 0x218b640
. fwd_inv = 0x20c98b8
. Start of JPEG DWT!
. w = 0x100 h = 0x100
. img_ptr = 0x20098a8 *img_ptr = 0xe22247c
. img_ptr = 0x20098ac *img_ptr = 0xde22083
. img_ptr = 0x20098b0 *img_ptr = 0xe221475
. img_ptr = 0x20098b4 *img_ptr = 0xe32207b
. img_ptr = 0x20098b8 *img_ptr = 0xe12287a
.
. img_ptr = 0x2026d7c *img_ptr = 0xb611851
. img_ptr = 0x2026d80 *img_ptr = 0xb812051
. img_ptr = 0x2026d84 *img_ptr = 0xce1845b
. img_ptr = 0x2026d88 *img_ptr = 0xc312c53
. img_ptr = 0x2026d8c *img_ptr = 0xc31585a
.
. img_ptr = 0x2049070 *img_ptr = 0x8b11058
. img_ptr = 0x2049074 *img_ptr = 0x800f054
. img_ptr = 0x2049078 *img_ptr = 0x7409c44
. img_ptr = 0x204907c *img_ptr = 0x680783b
. img_ptr = 0x2049080 *img_ptr = 0x6608c43
.
. im_s_ptr = 0x20098a8
. buf_red = 0x20498b0 *buf_red = 0xe2
. buf_red = 0x20498b4 *buf_red = 0xde
. buf_red = 0x20498b8 *buf_red = 0xe2
. buf_red = 0x20498bc *buf_red = 0xe3
. buf_red = 0x20498c0 *buf_red = 0xe1
.
. buf_gr = 0x210b638 *buf_gr = 0x89
. buf_gr = 0x210b63c *buf_gr = 0x88
. buf_gr = 0x210b640 *buf_gr = 0x85
. buf_gr = 0x210b644 *buf_gr = 0x88
. buf_gr = 0x210b648 *buf_gr = 0x8a
.
. buf_bl = 0x218b640 *buf_bl = 0x7c
. buf_bl = 0x218b644 *buf_bl = 0x83
. buf_bl = 0x218b648 *buf_bl = 0x75
. buf_bl = 0x218b64c *buf_bl = 0x7b
. buf_bl = 0x218b650 *buf_bl = 0x7a
.
. w = 0x100 buf_red wptr = 0x20498b0 alt =  0x20898b0 fwd_inverse =  0x20c98b8 f. _inverse = 0x1
. w = 0x100 buf_gr wptr1 = 0x210b638 alt1 =  0x214b638 fwd_inverse =  0x20c98b8 . d_inverse = 0x1
. w = 0x100 buf_bl wptr2 = 0x218b640 alt2 =  0x21cb640 fwd_inverse =  0x20c98b8 . d_inverse = 0x1
. all pointers for r g b dwt should be setup correctly
. starting red dwt
. in lifting
. in singlelift
```

Transfer data from the SDRAM of the catboard to RPi3B.

```
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./arm-rdsdram red.bin

Write-COMPLETE
The size of the buffer is 0x00ffff or 65535 words
```

The octave console that creates the 3 figures below.


File Edit Debug Window Help News
Current Directory: /home/pi/testbuilds/catzip/sw/host
File Browser
ome/pi/testbuilds/catzip/sw/host
Name
obj-arm
arch.sh
arm-netpport
arm-rdsdram
arm-sdramscope
arm-wbregs
Workspace
Filter
Command History
Filter
Editor
File Edit View Debug Run Help
<unnamed>
rgb.m
eol: LF line: 1 col: 1
Command Window Editor Documentation

| Name | Class | Dimension | Valu |
|---|---|---|---|
| ans | double | 1x1 | 0 |
| fid | double | 1x1 | 18 |
| im1 | double | 256x256 | [226, |
| im2 | double | 256x256 | [137, |

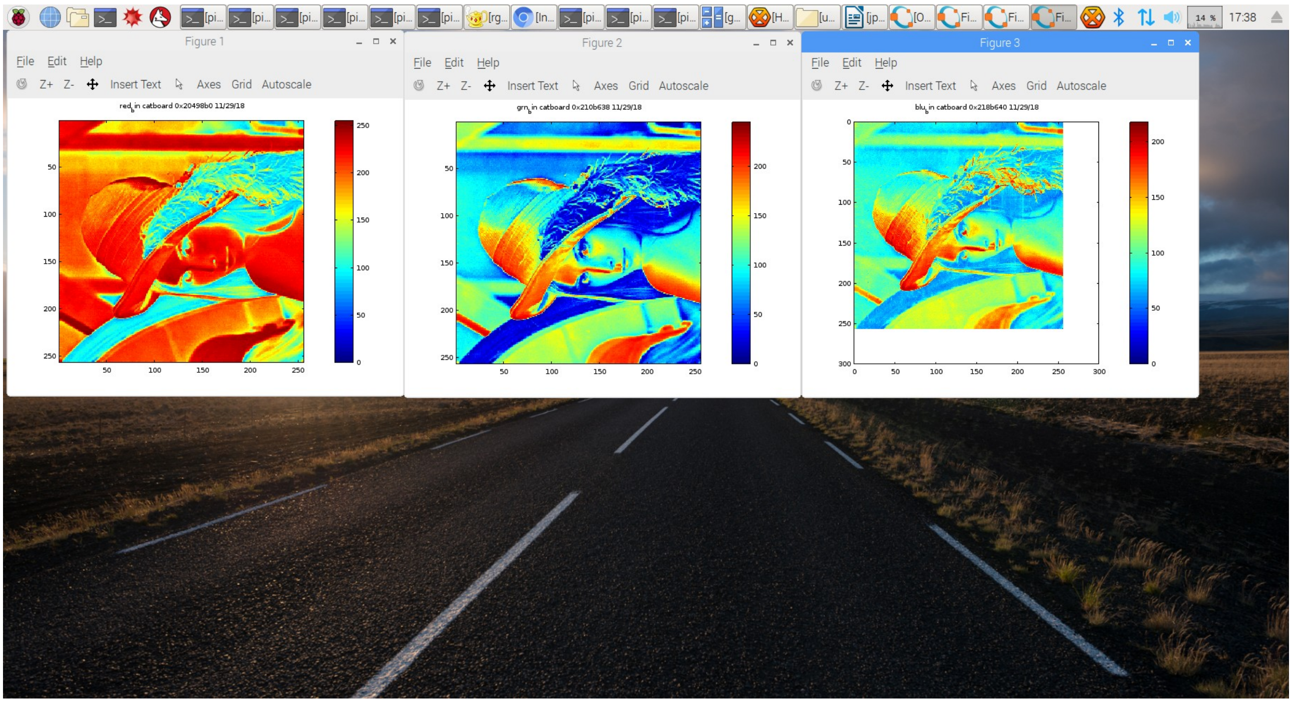
```
exit
# Octave 4.0.3, Thu Nov 29 17:35:50 2018 MST
rgb
rgb
```

```
clear
fid = fopen('red.bin','r'); im1 = fread(fid, [256,inf], 'int32'); fclose(fi
fid = fopen('grn.bin','r'); im2 = fread(fid, [256,inf], 'int32'); fclose(fi
fid = fopen('blu.bin','r'); im3 = fread(fid, [256,inf], 'int32'); fclose(fi

figure;
imagesc(im1);
colorbar;
title "red_bin catboard 0x20498b0 11/29/18"

figure;
imagesc(im2);
colorbar;
title "grn_bin catboard 0x210b638 11/29/18"

figure;
imagesc(im3);
colorbar;
title "blu_bin catboard 0x218b640 11/29/18"
```



Need to compile rdsdram.cpp with the address of the buf_gr

```
#define         DUMPMEM   0x210b638
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./buildsdramscope.sh

pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./arm-rdsdram grn.bin

Write-COMPLETE
The size of the buffer is 0x00ffff or 65535 words
```

grn_image

Need to compile rdsdram.cpp with the address of the buf_gr

```
#define         DUMPMEM   0x210b638
```

Need to compile rdsdram.cpp with the address of the buf_bl

```
#define         DUMPMEM   0x218b640
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./buildsdramscope.sh
pi@mypi3-1:~/testbuilds/catzip/sw/host $ ./arm-rdsdram blu.bin

Write-COMPLETE
The size of the buffer is 0x00ffff or 65535 words
```