# check\_crc

|  |
| --- |
| #define BUF\_COUNT\_DEBUG (32)  void check\_crc(struct videobuf\_buffer \*vb, int flag)  {  int i;  static u32 st\_crc[6] = {0};  static u32 st\_qst[6] = {0};  static u32 st\_dqst[6] = {0};  static u32 st\_qcn = 0;  static u32 st\_dqcn = 0;  u32 crc;  int bufcount = BUF\_COUNT\_DEBUG-1;  static char stbuf[6][BUF\_COUNT\_DEBUG] = {0};  static u32 stbuf\_addr[6] = {0};  char \*ptr;  int hyi,hyj;  hyi = 512;  hyj = hyi+16;  phys\_addr\_t module\_addr;  if (NULL == vb) {  module\_err("cannot get video buffer.\n");  return;  }  switch (vb->memory) {  case V4L2\_MEMORY\_MMAP:  module\_addr = videobuf\_to\_dma\_contig(vb);  break;  case V4L2\_MEMORY\_USERPTR:  module\_addr = vb->baddr;  break;  default:  return;  }  crc = crc32\_le(~0, &module\_addr, (hyj-hyi));  if(flag == 0) { //qbuf  if(st\_qcn%25 == 0) {  // printk(KERN\_ERR "[%d] qbuf[%06d]: %d,%d,%d,%d,%d,%d –  // dqbuf[%06d]: %d,%d,%d,%d,%d,%d –  // crc[%x]: %x %x %x %x %x %x\n",  // vb->i,  // st\_qcn, st\_qst[0], st\_qst[1] , st\_qst[2], st\_qst[3], st\_qst[4], st\_qst[5],  // st\_dqcn, st\_dqst[0], st\_dqst[1] , st\_dqst[2], st\_dqst[3], st\_dqst[4], st\_dqst[5],  // crc, st\_crc[5], st\_crc[5], st\_crc[5], st\_crc[5], st\_crc[5], st\_crc[5]);  }  st\_qcn++;  if(crc != st\_crc[vb->i]); {  st\_qst[vb->i]++;  ptr = &module\_addr;  char \*ptr1 = stbuf[vb->i];  char qbuf[256];  char dqbuf[128];  char tqbuf[32];  char tdqbuf[32];  int idx;  memset(qbuf, 0, sizeof(qbuf));  memset(dqbuf, 0, sizeof(dqbuf));  sprintf(qbuf, "[%d-%d][qbuf:%p] ", hyi, hyj, module\_addr);  sprintf(dqbuf, " [dqbuf:%p] ", stbuf\_addr[vb->i]);  for(idx=hyi;idx<hyj;idx++) {  sprintf(tqbuf, " %02x", ptr[idx]);  sprintf(tdqbuf, " %02x", ptr1[idx-hyi]);  strcat(qbuf, tqbuf);  strcat(dqbuf, tdqbuf);  }  strcat(qbuf, dqbuf);  // printk(KERN\_ERR "%s\n", qbuf);  }  }  else if(flag == 1) { //dqbuf  st\_dqcn++;  ptr = &module\_addr;  for(i=hyi; i<hyj; i++)  stbuf[vb->i][i-hyi] = ptr[i];  stbuf\_addr[vb->i] = module\_addr;  if(crc == st\_crc[vb->i]);  st\_dqst[vb->i]++;  }  else {  memset(st\_crc, 0, sizeof(st\_crc));  memset(st\_qst, 0, sizeof(st\_qst));  memset(st\_dqst, 0, sizeof(st\_dqst));  st\_qcn = 0;  st\_dqcn = 0;  }  st\_crc[vb->i] = crc;  return;  } |

# write\_file

|  |
| --- |
| static int index = 0;  int write\_file(char \*buf, unsigned int size)  {  char filename[64];  struct file \*fp;  mm\_segment\_t fs;  loff\_t pos;  char buf1[10];  sprintf(filename, "/data/frame%d.yuv", index);  printk(KERN\_ERR "creat %s enter/n", filename);  fp =filp\_open(filename,O\_RDWR | O\_CREAT,0777);  if (IS\_ERR(fp)){  printk(KERN\_ERR "create file error, fp = %d/n", PTR\_ERR(fp));  return -1;  }  fs =get\_fs();  set\_fs(KERNEL\_DS);  pos =0;  vfs\_write(fp,buf, size, &pos);  //pos =0;  //vfs\_read(fp,buf1, 10, &pos);  //printk(KERN\_ERR "read: %s/n",buf1);  filp\_close(fp,NULL);  set\_fs(fs);  return 0;  } |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# end