The Sony Control-L LANC Protocol

I have been collecting info on LANC for several years. Below is a collection of what I have for your fi les.

```
======================================================================
      A small hardware Summary of the SONY LANC Control L Protocol
               October 27, 1994  (dmeed@nbnet.nb.ca)
======================================================================

  For the inside scoop on the Sony LANC Control L order the Protocol
manual for Control L/Lf from Sony.  I ordered it from:

     Sony Service Company
     Parts Division
     8281 NW 107th Terrace
     Kansas City, MO 64153
     (816) 891-7550

     Part # 9-972-453-11   "Protocol of Control L/LF"

  You get to find out where to order it in your country, but I'm sure
they will send you a copy from the address above if you give them a
call.  About $20US or so (but don't quote me!)  (Somebody let me know
if they do manage to get it from the US overseas.)

  Control L is a two-way serial open collector 9600 baud protocol.
Cameras (control-L) use a three pin sub-mini jack that has ground on
the sleeve, power (up to 100ma unregulated 5-9v) at the tip and LANC
signal on the ring.

  VCRs with the five pin mini-DIN connector have DC out on pin 1,
LANC bus on pin 4 and GND on pin 5.  Pin 3 may be a power switch line
(ground to switch power on and off) and pin 2 may put out a square wave
locked to the video frames.  (pins 2 & 3 are optional, not used on
consumer equipment.)

  The LANC bus is open collector so it is normally pulled high to about
5v and is pulled low to send commands or status information. You can
hook the LANC signal directly to the input of a 1488 RS232 line driver
and feed that into your PC serial port and capture the 9600 baud data
stream.  It will have to be inverted before you use it (00 will read as
FF).

  The data stream is 8 bytes, then a gap (1.7ms? until the end of the
current frame) then 8 bytes for the next frame, another gap and so on.
If you can write your serial driver to sync to that gap you can read it
easily.  I wrote a program in modula 2 to display the signal, but I
don't know enough about PC timing to detect that gap in the data
stream.

The camcorder puts out an 8 byte data packet with each video frame.
The first two bytes are for controllers to command the camera and are
usually 00 00. The next two are for tuners and are also usually 00 00.
The last four bytes are for the VCR status and carry the counter and
several other status bits.

  Reading is the easy part . .  .

  To send a command you have to wait until you detect the start bit of
the first byte in a frame then impose your signal on the LANC line
(remember it is open collector and inverted, so 00 is all high - you
just pull low for your command bits). I haven't the foggiest how to
send a byte on the serial port in response to an external start bit.  I
think you might almost have to bit bang it in software.  Wouldn't be
bad - only need to send 2 bytes in each frame. Oh yeah - and in at
least 5 consecutive frames to make sure the VCR hears and understands
your command.

I've got a Sony RM95 wired remote here.  These are the codes I see by
pressing the buttons on the remote and eavesdropping on the LANC line
with a PC @9600 baud.

 Button on remote    Code sent on bytes 1 & 2
 ================    ========================
 (these buttons work in camera mode)
     Zoom tele            28 35
     Zoom wide            28 37
     focus                28 41 (toggle between manual and autofocus)
     focus farther        28 45
     focus closer         28 47
     start/stop rec       18 33
     edit search -        18 65
     edit search +        18 67
     rec review           18 69
     power                18 5E

  (these buttons work in VTR mode)
     stop                 18 30
     pause                18 32
     play                 18 34
     rewind               18 36
     fast forward         18 38
     record               18 3A
     slow                 18 46
     frame advance        18 62
     counter reset        18 8C
     data screen          18 B4

I'm looking for the codes for slow zoom, fade, index, and all the other
goodies on the side of my V101 camera.  Would be nice to be able to
press those buttons without having to touch the camera jiggle jiggle).
If you find them please let me know.  You could always try sending
all the possibilities and see what you get...


Update - 95/07
I've just managed to write a simple program for the PIC16C84 processor
which will control a camcorder remotely - Sort of a homemade version of
the RM100 remote control

David Meed       dmeed@nbnet.nb.ca
```

```
======================================================================
Additional Resources (haven't been there to check it out yet):
>>LANC info can be found in the file collection of the Amer. Soc. of
>>Reverse Engineers.  It can be accessed via ftp to tracker.vet.uga.edu.
>>The files are in the /pub/asre directory.  In addition to several files
>>on LANC, they have a file on the VCRPLUS protocol.
>>
>>Dave Dunfield has some files on LANC in association with his DBOX project
>>(a do-it-yourself LANC controller).  His files are on his BBS at
>>613-256-6289.
======================================================================
And the rest is some stuff I was sent from "NEIL HIGGINS, SEQEB"

======================================================================


Xref: un rec.video:39939
Path: un!bunyip.cc.uq.oz.au!munnari.oz.au!spool.mu.edu!howland.reston.ans.net!europa.eng.gtefsd.com!news.umbc.edu!haven.umd.edu!purdue!mozo.cc.purdue
From: laird@pasture.ecn.purdue.edu (Kyler Laird)
Newsgroups: rec.video
Subject: Re: LANC Protocol
Date: 29 Sep 1994 01:23:10 GMT
Organization: Purdue University
Lines: 852
Message-ID: <36d4tu$buq@mozo.cc.purdue.edu>
References:
NNTP-Posting-Host: pasture.ecn.purdue.edu
X-Newsreader: NN version 6.5.0 (NOV)

nh007@un.seqeb.gov.au ( NEIL HIGGINS) writes:

>Can anyone direct me to a specification for the LANC protocol, used for
>edit control, etc. on Sony cameras? The local rep. tells me the protocol
>specification is not available. I find that hard to believe.

Reps can be so full of it, can't they...

Here's the unedited brain dump.

--kyler


Xref: noose.ecn.purdue.edu comp.multimedia:6121 comp.sources.wanted:14178 comp.sys.amiga.multimedia:2644
Path: noose.ecn.purdue.edu!mentor.cc.purdue.edu!purdue!haven.umd.edu!darwin.sura.net!zaphod.mps.ohio-state.edu!sdd.hp.com!portal!cup.portal.com!DonD
From: DonD@cup.portal.com (Don Robert DeCosta)
Newsgroups: comp.multimedia,comp.sources.wanted,comp.sys.amiga.multimedia
Subject: Re: sony Control-L documentation
Message-ID: <78191@cup.portal.com>
Date: Wed, 24 Mar 93 22:37:07 PST
Organization: The Portal System (TM)
References:
Lines: 759

>Does anyone have any information on where I can get documentation for the
>SONY (I guess) control-L protocol for talking to VCRs, camcorders, etc.?
>

Greeting LANC fans,

I just went through this back in September... Here's the repsonses
I got.


Don DeCosta           |The nice thing about sanity is|  VM/Nomad2
DonD@cup.portal.com   |you can lose it more than once|  Amiga/Imagine
==========================================================================
==========================================================================
==========================================================================
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

You can try to get information from Sony although there is little chance
to get an up-to-date specification.

You can also find some partial information on mrcnext.cso.uiuc.edu from
someone who made a Control-L interface.

Regards,
--
Michael van Elst
UUCP:     universe!local-cluster!milky-way!sol!earth!uunet!unido!mpirbn!p554mve
Internet: p554mve@mpirbn.mpifr-bonn.mpg.de
                                "A potential Snark may lurk in every tree."
==========================================================================


What kind of system are you working on?

I have a Mac with some Hypercard XCMDs of a company called Video Production
Controls.
Control L is essentially a 9600 baud serial line with an external (ie.
dictated by the VCR) clock.  This means reading is easy: just a rs-232
9600 baud hookup works, writing commands is harder because you will have
to 'clock in' to the VCR.

Sony provided me with lists with all the codes.
I would send if I had them here.

Lex van Sonderen
lex@aimla.com


==========================================================================
~Date: Tue, 15 Sep 92 17:28:07 PDT
~From: jphealy@luna.dpl.scg.hac.com (John P. Healy)
Message-Id: <9209160028.AA07356@luna.DPL.SCG.HAC.COM>
To: DonD@cup.portal.com
~Subject: Re: Sony LANC/Control L
~Newsgroups: comp.multimedia
In-Reply-To: <65824@cup.portal.com>
Organization: Hughes Aircraft Co., El Segundo, CA
Cc:

Have you already looked at Mark Abbate's VideoTolkit? It may do
everything you want (and more).

==========================================================================
```

Hi,

we are going to sell the Video Workshop, a PC based Video cut system,
where you have a Sony Control L interface and a Panasonic 5 pin
interface to control a player and a recorder VCR.
It starts from 999.-DM and comes with the PC-Titler juniour program to
generate titles during cuting.

Best regards Stefan Hartmann.
email to: leo@zelator.in-berlin.de

--

****
* Stefan Hartmann   This is how to contact me: *
* EMAIL: leo@zelator.in-berlin.de   *
* Phone : ++ 49 30 344 23 66   FAX : ++ 49 30 344 92 79 *
****

==========
========== LANC1.txt ==========
==========
~Date: Tue, 19 Feb 91 14:32:11 PST
~From: root@shuksan (Operator)
Message-Id: <9102192232.AA13431@>
~Subject: control L code

 Hi there -- here is my C code and asm code for my camera
control stuff. It is uuencoded to keep the mailers happy
(I hope). The code is somewhat hardware dependant and is
not the cleanest in the world (that is what you get when
it "evolves" via a 'scope and bit fiddling!). I am running
a 10mhz AST 286 with a VGA monitor. Feel free to do what
you want with this code. When figuring the loop timing, if
you are using the assy ref. guide with TASM, the number of
clocks shown for some instructions are NOT correct! I discovered
for example on some of the jump instructions, it lists for
example 4 clocks - in reality, it is 4 clocks + the number
of bytes in the following instruction .... just a slight
difference! The interface looks very sim. to the drawing
in the ctl L manual (if you want, I will get it and describe
it for you --- I forgot it at home today!)

I will have my schematic in postscript form if you are interested.
My schematic capture routine generates postscript for a plotter
(I use a laserjet II with the Adobe ps cart.)

Mikey (yes "he likes it!")
===   Mike Fields
==> uunet!bcstec!shuksan!mikey (206) 657-6136 [work]   12022 NE 138th Pl.
uw-beaver!ssc-vax!shuksan!mikey (206) 821-3492 [home]   Kirkland, Wa. 98034

==========
========== LANC2.txt ==========
==========

~Subject: Control-L Info (was Re: LANC interface?)
~From: lindh@uhasun.hartford.edu (Andrew Lindh)
Path: ux1.cso.uiuc.edu!moe.ksu.ksu.edu!zaphod.mps.ohio-state.edu!usc!apple!news
.bbn.com!noc.near.net!uhasun!lindh
~Reply-To: lindh@uhasun.hartford.edu (Andrew Lindh)
~Newsgroups: rec.video
Message-ID: <708@ultrix.uhasun.hartford.edu>
~References: <1991Sep17.172210.12851@gtc.com> <1991Sep17.234413.01054822@locus.c
om> <1991Sep22.150545.11650@bilver.uucp>
~Date: 29 Sep 91 01:41:55 GMT

I have a (old, 1986) copy of the Control-L protocol. It does talk about
Beta and Video 8 (8mm), but NOT VHS. Here is some short info:

Pin	Color	Function
1	Red	DC OUT (5.9 to 9V DC 100mA)
2	White	Option CTL (Lf Mode ONLY)
		_____
3	Black	Power SW
4	Yellow	Serial Bus
5	Blue	Ground

The Bus Line is a bi-directional Serial line (Async 9600 baud/1 start bit
2 stop?) The start is syncronized with each FIELD and is 8 words (8 bits each)
Pin 2 is not used in consumer units, it has to do with tape direction
you don't need it....

There must always be a Commander and a Slave. A VTR or Camera is a
slave, a Computer, Editor, VTR (with built-in a Edit Controller) is
a Commander. The commander uses the first 2 words to send commands
and the last 6 are for returnned info.

But there are other setups:

UNIT	WORDS
	0 1 2 3 4 5 6 7
Console VTR	R R W W W W W W
CTR with tuner	R R W W W W W W
Portable VTR	R R R R W W W W
Camcorder	R R R R W W W W
AC Adapter	- - W W R R R R
TT unit	- - W W R R R R
Camera	A W B - R R R R A=R(Sometimes)/W B=R(sometimes)
Editor	W W - - R R R R
Computer	W W R R R R R R
Remote	W W - - R R R R

Word 0	Device Code (4 bits)/Guide Code (4)
Word 2	Device Code (4)/Guide Code
Word 1/3	Basic Command to VTR (8 bits)
Word 1/3	Command To Camera (8)
Word 3	TV channel (8)
Word 4	VTR Mode (8)
Word 5	VTR Status (4)/Guide Code (4)
Word 6	VTR Status

```
        Word 7          VTR Status
        Word 7          Insert Status
        Word 6/7        4-digit counter
        Word 6/7        Hour/Minute/Second

If you are connected from a computer to a VTR:
Shorting 3 to ground turns the VTR off.
Start bit is sent from the VTR
First 2 words (0 and 1) are commands from the computer to the VTR
the last 6 are from the VTR to the computer


                     +5V
                      |
                      ~4.7K=VTR or 10K in other
                      |         47k   /------INPUT
pint 4--+------+--|<--+-->|-----~-+-|
        |      |  signal diode    |  \>+
        |      |                  |    |
        _      |                  ~----+--G
        ^      \          47k
        |       |----+----~------OUTPUT
        G
~References: <1991Sep17.234413.01054822@locus.com> <3230@news.chips.com>
~Date: 1 Oct 91 15:33:08 GMT

>Uh, I have a Sony CCDV101 Hi8 Camcorder with control-L on a mini-phono jack...
>Is this just the pin 4 serial bus?  Or is it a Tip-Ring(stereo mini) with
>power or something else on the ring?

It's a Tip-Ring.  One is power and the other is the TTL level serial
bus.  I don't remember which is which.  The power is the unregulated 6V
power from the battery.

The serial signal is sending data at 9600 baud in a standard 1 start bit,
8 data bits, and 1 stop bit format.  To feed this into a standard RS-232
port you just need an RS-232 level converter (and maybe an inverter?).
The level converter needs +-12 volts, you you will also need an external
power source.  Sending data back to the VCR is harder.

The serial line is normally high (+5V), the start bit is low (0V). The
data bits are inverted, i.e. 1 -> 0V, 0 -> +5V, the stop bit is high (+5V).

The VCR sends out a set of 8 bytes once every field (i.e., 60 times a
second for NTSC VCRs).  Other than the stop and start bits, there is
no time gap between each of the 8 bytes.  There is a gap between each
set of 8 bytes.  To know when the set of 8 bytes begins, you are supposed
to watch for this time gap.  However, because of the consistent format
of the data, you should be able to determine were the set begins just by
examining the data from multiple sets of bytes.

Bytes 0 and 1 (The first two bytes in the 8 byte set), are used by
"Camera, personal computer and editor" to send commands to the VCR.

Bytes 2 and 3 are used buy "Tuner and timer" to send commands to VCR
and "Status to exterior".

Bytes 4, 5, 6, and 7 are used by the VCR to send out status and counter
information.

The serial line is an open-collector type data line.  This means that
the VCR normally holds the line at +5V with a pull-up resistor.  An
external unit that wants to send data uses an open-collector type
TTL gate that does nothing to send a +5V signal, and grounds the
line to send a 0V signal.

The VCR will send the start bit for the first 4 words, but will leave
the line at +5V (all zeros) for the data.  Command bytes of all zero
are defined to be a no-op.

A unit (like an edit controller) that wants to send a command must
wait for the VCR to send the start bit, and then start transmitting
the correct 8 data bits, in sync with the VCR.  Making a serial port
on a PC do this is hard.  I got e-mail from one guy that said he did
it all in software.  He connected the serial data line to something
like the CD line, and used something like DTR for output.  He then
wrote a timing loop in software to read and transmit the bits at
the correct time.  Simple hardware but really tricky and machine
dependent software.

The serial line can also have multiple controllers connected to it.
Because of this, any controller should do bit checking.  This is where
you read the commands bits as you send them to check and see if another
controller is sending a command at the same time. (i.e., you must
look for collisions.)

If two devicess send a 5V, or 0V signal at the same time, you can't see
the collision.  But if one sends 5V and one sends 0V, then the 0V
signal will "win".  If you try to send a 5V signal, but read back a 0V
level on the line, then you know there was a collision and you should
stop sending and let the other guy complete his command.

I don't now for sure, but I think this might happen in a device like
a camcorder if you press a button (like stop) on the unit at the same
time your computer is trying to send a command.

The manual also says that you must send evey command multiple times
to make sure the VCR responds to it.  It says: "In order that the VTR
makes the command effective, it is necessary to transmit the
same code continuously over 4 fields.  Therefore, it is required for the
peripheral side to transmit over 5 fields of the same code."  I don't
quite understand the logic in that statement, but I think it's saying
you have to transmit every command for 5 fields.

The tables that define all the bits, and all the commands take up
6 pages (double sided) in the manual, and I don't intend to type
them all in, but here is enough to get you started:

Bits are numbered 0 to 7, with 0 being the first bit sent.

Byte 0

  B0-3
  0000 Command from Camera
  0001 Command from Commander

  B4-7
```

```
  0000 Use prohibited (i.e. no-op)
  1000 Basic VTR command
  0100 Camera comand
  1000 Special VTR command

Byte 1 (Depends on B4-7 of Byte 0)

 Basic VTR commands - Byte 0 = 0001-1000 (This is not the complete list)
  00000000 CH-1/1       (I think this is like pressing the 1 button)
  01000000 CH-2/2
  00100000 CH-3/3
  ...
  01001000 CH-10/0
  more strange channel buttons.

  01010100 Power on/off
  00111010 Power on
  01111010 Power off

  00001100 STOP
  01001100 PAUSE
  00101100 PB (Play Back??)
  01101100 REWIND
  00011100 FF
  01011100 REC
  00000010 STILL

  00000110 REVERSE
  01000110 FOWARD
  00011110 PAY (should this be PLAY???)

 Special VTR command - Byte 0 = 0001-1100
  10001011 + Frame advance
  10011011 - Frame advance

 Camera comand - Byte 0 = 0001-0100
  11010100 Camera Power on/off
  10101100 Zoom Tele
  11101100 Zoom Wide

Byte 4 - VTR mode
  (the xxx bits have different meaning for each mode which I don't
   list here.  For example 00100110 is REC with VIDEO INSERT)
  1000xxxx Tape Ejected
  0100xxxx Stoped
  1100xxx0 FF
  1100xxx1 RWD
  0010xxxx REC
  0001xxxx More REC modes
  0110xxxx PB (Play Back)
  1110xxxx SLOW/STILL

Byte 5 - VTR status and guide code
  B0-3 status bits
  B0   1 -> undefined command. (you sent a command that this VCR doesn't know)

  B4-7 (defines the contents of the last two bytes (6 and 7))
  0000 Use prohibited
  1000 8mm VTR status
  0100 Decimal 4-digit counter
  1100 first half of hour, minute and second counter
  0010 second half of hour, minute and second counter
  1110 Beta VTR status

Byte 6 for decimal 4-digit counter
  B0-3 first digit
  0000 0
  1000 1
  0100 2
  1100 3
  ...
  1001 9

  B4-7 second digit

Byte 7 for decimal 4-digit counter
  B0-3 third digit
  B4-7 forth digit

Byte 6 for first half of HMS counter
  B0-3 units digit of seconds (encoded like above)
  B4-7 tens digit of seconds
Byte 7 for first half of HMS counter
  B0-3 units digit of minutes
  B4-7 tens digit of minutes

Byte 6 for second half of HMS counter
  B0-3 units digit of hours
  B4-7 tens digit of hours
Byte 7 for second half of HMS counter (only 1 bit used)
  B7
  0 +
  1 -

On your V101, the single frame record can not be activated with the
play/rec button on the camara.  It can only be activated with the play/rec
buttons on the remote.  I don't know what it will do if you sent it
the REC command.  Will it start a single frame record cycle?  If it doesn't
then maybe they defined a new command for the V101 to make that work.

I own a V101 and I too want to test this some day.

Curt Welch
curt@oasys.dt.navy.mil
Code 3531
David Taylor Research Center (A Navy Lab)
Bethesda, MD
(301) 227-1428
~Subject: Sony Control-L Protocol Hacke!
~From: pierce@chips.com (John Pierce)
Path: ux1.cso.uiuc.edu!uwm.edu!cs.utexas.edu!swrinde!mips!pacbell.com!iggy.GW.V
italink.COM!nocsun.NOC.Vitalink.COM!indetech!daver!news.chips.com!pierce
~Reply-To: pierce@chips.com (John Pierce)
~Newsgroups: rec.video
Message-ID: <3493@news.chips.com>
~References:
~Date: 29 Oct 91 01:39:19 GMT
```

Well, after reading a couple of postings here about Sony Control-L protocol
we called Sony and got the official spec on Control-L. Its bad reading!
Direct translation? Lotsa typos. But. We made it WORK!

Following is a hacked up piece of C for a IBM-PC parallel port. Take a
DB-25P. Connect a sub-mini MONO jack to ground (pin 18 will do) and
pin 17 ("Select"). Voila' Build following piece of C with MS C6.0 using
at least -Oi (Optimize Intrinsics--- put I/O inline!) bunches of oneletter
commands will make my Sony CCD-V101 do all kinda tricks! Lotsa fun.

Please for to excuse hacked style of program ... This was written for
the heck of it on a Saturday afternoon, after several beers!

Oh. This has only been tested on 20Mhz 386DX (no cache) and faster boxes.
I don't know how slow a box it might work on, but it is speed independant
as long as your box is fast enuf. Note the Delay(t104/4) after start bit
detection. This can be trimmed to better align the transmitted bits with
the remote devices timing. If your computer was infinitely fast, it s/b
Delay(t104/2).... i.e. sample in mid-bit. But since pc's take time
to do anything, we just tweaked it til it looked good on a scope...

Also note that we make no attempt to display the status while there is
a command being actively sent. This is cuz its written with STDIO instead
of clever PC screen blasting, and we were losing too many messages...

Obviously this whole shabang should be in assembler.

==========cut-here==========

```
#include
#include

// Copyright (C) 1991, Acme Software Inc.
// All rights reserved EXCEPT the right to use for NON-COMMERCIAL
//      PURPOSES.  I.E.  Hack away but DON'T sell this!
// This copyhack must accompany all copies of this source code.


#define dataport 0x37A
// #define dataport 0x3BE

#define outbit(b) outp(dataport,((inp(dataport)&(0xFF^bit)) | (((b)&1)<< t);
        rt += t;
        }


void framesync(void)
        {
        int i;
        while(1)                // frame sync
                {
                rt = readpic();
                while ((readpic()-rt) < t1700 && (i=(inp(dataport)&bit)==0));
                if(i) break;
                }
        }

unsigned DoByte(unsigned char O)
        {
        static int x=0;
        unsigned d,i;

        rt = readpic();
        while((inp(dataport)&bit)==0)   // wait for start bit...
                if((readpic()-rt) > 45000) return 0;    //   OOPS, TIMEOUT!!!

        rt = readpic();
        delay(t104/4);          // halfway (almost) into start bit
        d = 0;
        for(i=0;i<8;i++)
                {
                d >>= 1;
                delay(t52);     // wait start of next bit
                outbit(O);
                O >>= 1;
                delay(t52);     // wait middle of bit
                if (inp(dataport)&bit) d |= 0x80;
                }
        delay(t104*2);  // wait for stop bit
        return d;
        }

void DoFrame()
        {
        int i;

        _asm {cli};
        framesync();
        for(i=0;i<8;i++)
                In[i] = DoByte(Out[i]);
        _asm{sti};
        }

void main(int argc, char *argv[])
        {
        int i;
        unsigned char data[8];
        unsigned char Command = ' ';
        long frame=0;
        int f=0;
        int Lapse = 0;
        int Time = 300;

        outbit(0);      // clear serial line

        printf("\n      p - Play                r - Record");
        printf("\n      , - Rewind              . - FF");
        printf("\n      s - Stop                S - Pause");
        printf("\n      < - Shuttle <<          > - Shuttle >>");
        printf("\n      t - Tele Zoom           w - Wide Zoom");
        printf("\n      e - Eject               l - Time Lapse");
        printf("\n      d - Display             q - Quit");
        printf("\n      x - Start/Stop");
        printf("\n       - Clear command");
        printf("\n");
```

```
while (1)
        {
        if (kbhit())
                {
                Command = getch();
                switch(Command)
                        {
                case 'd':       // Display ON/OFF
                        Out[0] = 0x18;
                        Out[1] = 0xB4;
                        f = 5;
                        break;
                case 'l':
                        Lapse = Time;
                case 'r':       // VTR Record
                        Out[0] = 0x18;
                        Out[1] = 0x3A;
                        f = 5;
                        break;
                case 'p':       // VTR Playback
                        Out[0] = 0x18;
                        Out[1] = 0x34;
                        f = 5;
                        break;
                case 's':       // VTR Stop
                        Out[0] = 0x18;
                        Out[1] = 0x30;
                        f = 5;
                        break;
                case 'S':       // VTR Pause
                        Out[0] = 0x18;
                        Out[1] = 0x32;
                        f = 5;
                        break;
                case '>':       // Fast Forward >> continous
                        Out[0] = 0x18;
                        Out[1] = 0x38;
                        f = -1;
                        break;
                case '<':       // Fast Rewind << continuous
                        Out[0] = 0x18;
                        Out[1] = 0x36;
                        f = -1;
                        break;

                case '.':       // Fast Forward >>
                        Out[0] = 0x18;
                        Out[1] = 0x38;
                        f = 5;
                        break;
                case ',':       // Fast Rewind <<
                        Out[0] = 0x18;
                        Out[1] = 0x36;
                        f = 5;
                        break;
                case 'e':       // Eject
                        Out[0] = 0x18;
                        Out[1] = 0x2C;
                        f = 5;
                        break;
                case 'x':       // Start/Stop
                        Out[0] = 0x18;
                        Out[1] = 0x33;
                        f = 5;
                        break;
                case 'w':       // Wide Zoom
                        Out[0] = 0x28;
                        Out[1] = 0x37;
                        f = -1;
                        break;
                case 't':       // Tele Zoom
                        Out[0] = 0x28;
                        Out[1] = 0x35;
                        f = -1;
                        break;
                case 'q':
                        exit(0);
                        break;
                case ' ':
                default:
                        Out[0] = 0x00;
                        Out[1] = 0x00;
                        Command = ' ';
                        f = 0;
                        break;
                        }
                }
        DoFrame();
        frame++;
        if(f>0)
                {
                f--;
                if (f == 0)
                        {
                        Out[0] = 0x00;
                        Out[1] = 0x00;
                        if (Command != 'l') Command = ' ';
                        f = 0;
                        }
                }
        if(Command == ' ')
                {
                if((In[5]&0xF0)==0x30)
                        {
                        ss = In[6];
                        mm = In[7];
                        }
                else if((In[5]&0xF0)==0x40)
                        {
                        hh = In[6];
                        sign = ((In[7]&0x80)?'-':' ');
                        }
                for(i=0;;i++)
                        {
                        if(Status[i].code == 0) break;
                        if(In[4]==Status[i].code) break;
                        }
```

```
                        printf("\r%c%2X:%02X:%02X %-16.16s",
                                sign,hh,mm,ss,
                                Status[i].name);

                        if(Status[i].code == 0)
                                 for(i=0;i<8;i++)
                                        printf(" %02X", In[i]);

                        }
                if(Command == 'l')
                        {
                        Lapse--;
                        if(Lapse <= 0)
                                {
                                Lapse = Time;
                                Out[0] = 0x18;
                                Out[1] = 0x3A;
                                f = 5;
                                }
                        }
                }
        }
```

==========

Try 1800-352-SONY. Its the VISCA Developers hotline

PS I have nothing to do with this stuff except personal interest. I work
for a different division of Sony.

Bob Berger - SONY Advanced Video Technology Center
685 River Oaks Parkway San Jose, CA 95134 408-944-4964 FAX: 408-954-1027
INTERNET: berger@sfc.sony.com UUCP: [uunet,mips]!sonyusa!sfcsun!berger
JAPAN: berger@sfc.sony.co.jp--

(Message inbox:279)
Return-Path: o-intelhouse@dlb.com
Received: from bank.ecn.purdue.edu by pasture.ecn.purdue.edu (5.65/1.32jrs)
id AA23943; Mon, 10 Jan 94 00:50:06 -0500
Received: from daver.bungi.com by bank.ecn.purdue.edu (5.65/1.32jrs)
id AA27269; Mon, 10 Jan 94 00:49:57 -0500
Received: from dlb by daver.bungi.com with bsmtp
(Smail3.1.28.1 #1) id m0pJEzB-00009Fa; Sun, 9 Jan 94 21:16 PST
Received: from bucksm by dlb.com with bsmtp
(Smail3.1.27.1 #4) id m0pJDtk-0000BNa; Sun, 9 Jan 94 20:06 PST
Received: by bucksm.dlb.com (Smail3.1.27.1 #6)
id m0pJC1L-000Apea; Sun, 9 Jan 94 19:06 MST
Message-Id:
From: daver!dlb!bucksm!dlb!netcomsv.netcom.com!wheaton.wheaton.edu!sga!dougs
To: o-intelhouse@dlb.com
Date: Sun, 9 Jan 94 15:48:12 CST
Subject: RE: Sony remote control protocol
X-L2L: dlb.com
Precedence: bulk
Errors-To: o-intelhouse@dlb.com

> > The following is a reference guide to using a microcomputer to
> >control Sony equipment via the SIRCS protocol. This can occur either
> >via an infrared interface, or with a Control-S port. It is being
> >released in the hope that it will be useful to some of you. Apparently
> >there is no documentation on the protocol available from Sony (at least,
> >that's what their publications office said).
>
> You spoke to the wrong person/people. You need to ask for "Sony
> Remote Control Systems" (2RM383-1). Mine was $5. SIRCS is described
> on pages 17-27. (SIRCS intro from 17-22, System III on 22, Beta on 23,
> and Audio on 25-27)

The codes are also the same as documented in the service manual "Protocol
of Control L/LF" (86C0943-1). However the timing is different. The
control L frame is 16.7ms consisting of 8 words. The first few words
send the command and the remaining words return a status from the device.
I wonder if a similar reason is why Ed didn't find the need to have the
packet take a full 45ms but Scott did? Could it be that room must be left
for a status response when using a wired interface?

By the way, does anyone know if a device with a Control-L port will
accept Control-S commands? My camcorder has a Control-L port but there
is almost no mention of it in the manual. According to the Control-L
service manual, Control-L is bidirectional and is primarily used
between a VTR and peripherals. Control-S is uni-directional and is
used to remote control all devices.

The Control L manual is dated March, 1986 even though I purchased it only
a year ago. It is missing information on current equipment but covers
Beta and Video 8 fairly well.

Do you have the phone number to order documents? I can't seem to find it
and would like to order the manual you mentioned.

> Good work, guys! I started observing these signals (with an
> MC68HC11), but after I got them and was able to send them, I
> bought the manual. Much easier.

I would be glad to document the codes used on my camcorder (CCD-V101)
and receiver (STR-D1011) however, I don't have access to an O'scope at
home. Does anyone know of a simple program to poll a PC paralell port
bit and display the result? That should do the job when attached to
a Sharp IR cube.

-- Doug Smith, SGA * Loves Park, IL * dougs@sga.uucp
-- AppleLink: G0231 * CompuServe: 72727,3532

--

==========

Neil Higgins E-Mail: esa_neilh@seqeb.gov.au
Network Automation Engineer Ph: +61 7 223 4327
The South East Queensland Electricity Board Fax: +61 7 210 0149
G.P.O. Box 1461, Brisbane, Q. 4001 AUSTRALIA

See also PicLanc.asm - A rough assembly listing for bit banging LANC on the PIC16c84 from Microchip

Return to home page

I'd love to have your feedback: Email dmeed@nbnet.nb.ca

Last modified: November 8, 1998.
