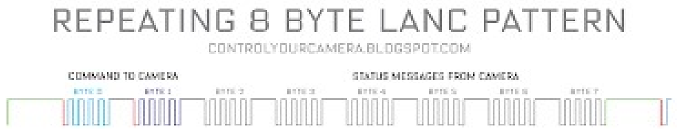
CONTROL YOUR CAMERA

MAC OS X SOLUTIONS FOR COMPUTER CONTROLLED PHOTO AND VIDEO SHOOTING.

THURSDAY, FEBRUARY 3, 2011

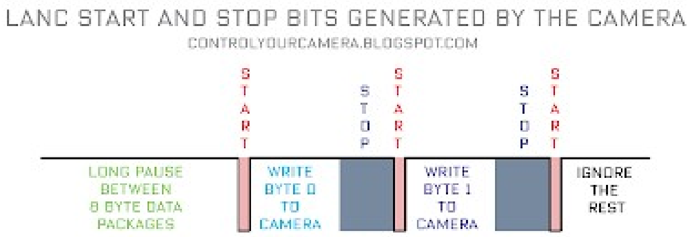
Arduino controlled video recording using the LANC port

LANC is a SONY development and stands for "Local Application Control Bus". It is a two-way serial open collector 9600 baud protocol with inverted logic. The LANC line is pulled high to about +5v and is pulled low to send commands or status information.


REPEATING 8 BYTE LANC PATTERN
CONTROLYOURCAMERA.BLOGSPOT.COM
COMMAND TO CAMERA
STATUS MESSAGES FROM CAMERA
BYTE 0
BYTE 1
BYTE 2
BYTE 3
BYTE 4
BYTE 5
BYTE 6
BYTE 7

The data stream is 8 bytes, followed by a longer than 5 millisecond pause until the end of the current frame. Then come another 8 bytes for the next frame and another pause and so on.
There's a short pause between individual bytes. Each byte is preceded by a start bit.

LANC is a single wire serial connection and because of this timing is critical. The camera can only listen to commands for the first half of the signal pattern.
The camera listens to the first 4 bytes and sends status information in the last 4 bytes. Only the first two bytes are needed to control a video camera. The rest can be ignored.


LANC START AND STOP BITS GENERATED BY THE CAMERA
CONTROLYOURCAMERA.BLOGSPOT.COM
START
STOP
START
STOP
START
LONG PAUSE BETWEEN 8 BYTE DATA PACKAGES
WRITE BYTE 0 TO CAMERA
WRITE BYTE 1 TO CAMERA
IGNORE THE REST

To send a command to the camera the command has to be synchronized to the LANC signal from the camera. The camera puts out a start bit before and a stop bit after each byte. The first

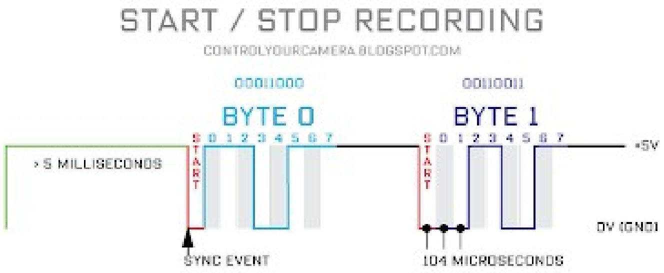
BLOG ARCHIVE

▼ 2011 (6)
- ► April (1)
- ► March (1)
- ▼ February (4)
  - Finding out LANC remote commands
  - USB to LANC Interface Adapter by Applied Logic Eng...
  - Arduino powered simple LANC remote
  - Arduino controlled video recording using the LANC ...

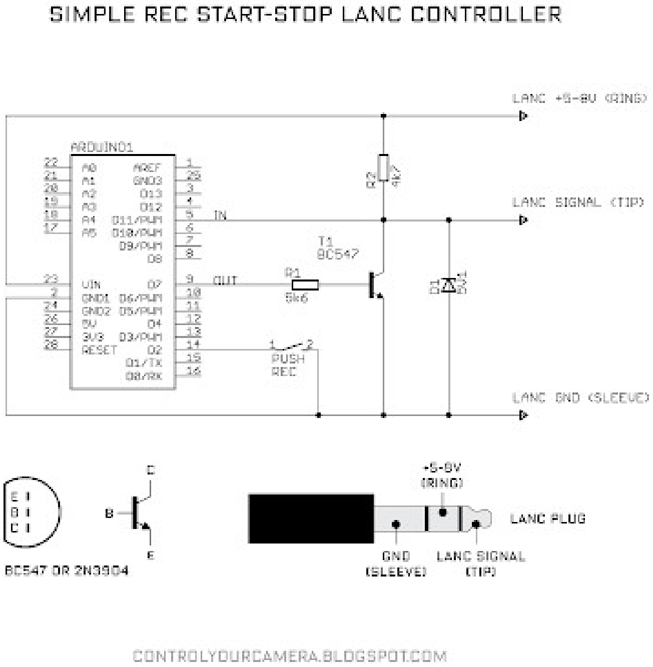
► 2010 (13)

► 2009 (2)

command byte from the controller (Arduino) has to be transmitted exactly between the start bit that follows the long pause between the 8 byte data packages and the following short stop bit. Then the second byte must be send to the camera. After sending those two bytes the LANC signal must be be left alone and put back to LOW i.e +5V.



The commands for starting and stopping video recording are 18 and 33 in hexadecimal format or 00011000 and 00110011 in binary format. The bytes must be put out with the least significant, right-most bit (bit 0) first i.e. 00110011 is put out 11001100. Note that in a LANC signal LOW is +5V and HIGH is 0V. LANC commands must be repeated 4 times in order to be accepted by the camera.



Since we have to read from and write to a single wire an interface is required. Thanks to Ariel Rocholls simple interface circuit writing and reading can be done independently. D1 is a 5.1V zener diode.

Arduino sketch:
Copy and paste it into the Arduino editor for a better reading experience.

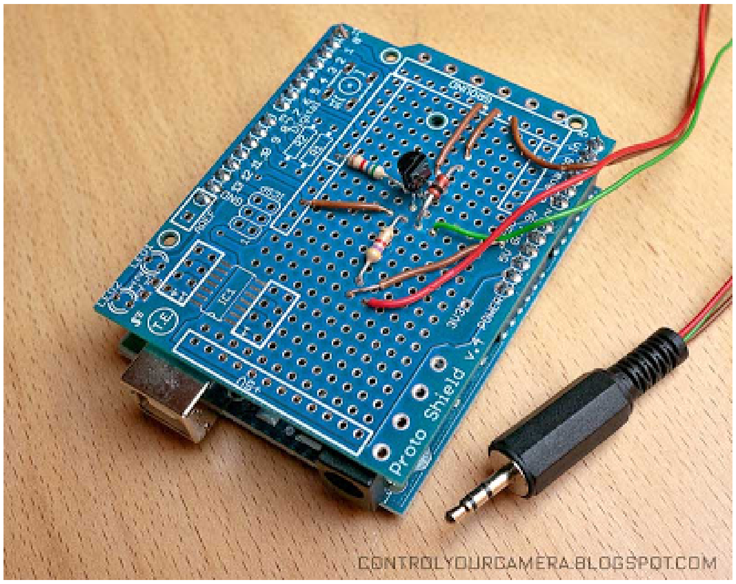
```
/*
Send a Start/Sop Recording command to the LA
Tested with a Canon XF300 camcorder
This code requires a simple interface see ht
Feel free to use this code in any way you wa

Comprehensive LANC info: www.boehmel.de/lanc

"LANC" is a registered trademark of SONY.
CANON calls their LANC compatible port "REMO

2011, Martin Koch
```

I've tested the Arduino sketch with a Canon XF300 camcorder and I suppose that the timing will work with the entire XF family. Other cameras may require experimenting with different "delayMicroseconds" values.



The prototype LANC interface is built onto a shield board that sits on top of the Arduino. The Arduino is powered by the LANC port. Although the XF300 puts out only +5V the Arduino worked without problems. During development I had the USB cable connected to the camera at the same time as the Interface was connected to the camera. I suppose since the Arduino chooses automatically to use either the external or the +5V USB power supply this should be safe.

Always turn the camera off before plugging or unplugging the LANC cable!

"LANC" is a registered trademark of SONY. CANON calls their LANC compatible port "REMOTE". Comprehensive LANC info can be found at boehmel.de/lanc.htm

POSTED BY MARTIN KOCH AT 2:23 AM

44 COMMENTS:

Robert C. Fisher July 12, 2011 at 3:00 PM

I am shooting the last space shuttle landing and need to control a Sony CX700 or 560. The Arduino will have a RTC so I can use it to turn on power to camera and start/stop recording. the cameras will be put out a day ahead and will be autonomous until we can reach them again sometime after the landing. I am not a coder, how would I accomplish this. Also about the power, can I not connect the power from the camera to VIN, leave everything as is?

Thanks

Reply

Martin Koch July 13, 2011 at 5:09 AM

Robert, I'm sorry but I have no experience with Sony cameras. For such an important historical event I suggest you better find someone to help you in person and make it foolproof.

Reply

Robert C. Fisher July 13, 2011 at 6:17 AM

Isn't LANC supposed to be a standard more or less? I have been trying to find someone to help for months ad your blog is one of the few resources. I do have a person helping with some coding and he is great but doesn't have any LANC experience.

Reply

Martin Koch July 13, 2011 at 12:51 PM

OK,if your camera has a LANC input and not the other SONY protocol (http://www.boehmel.de/slink.htm) then the solution above should work. A contact from the RTC (I suppose this means real time clock) will have to power camera and Arduino which after a delay will start recording over LANC.

The best option would be putting the camera into sleep mode and waking it up over the LANC line. I don't know if your camera has this but if yes you need a SONY LANC controller to find the command using this method: http://controlyourcamera.blogspot.com/2011/02/finding-out-lanc-remote-commands.html or maybe you find the command at http://www.boehmel.de/lanc.htm

Of course you can power the Arduino from the camcorder battery instead from the LANC line.

Good luck!

Reply

Georgia September 28, 2011 at 9:21 AM

Hello,

I have built the prototype and uploaded the code to my Arduino. When I plug into the camera, it just keeps turning the camera on and off over and over again. Can anyone help?

Thanks

Reply

Replies

julian March 7, 2012 at 9:33 AM

hello, thanks for the info in your blog.
the best i found in web.
it explains clear the info put in http://www.boehmel.de/lanc.htm
i want to implement the rest of the codes like white balance or focus.
have you try this?

thanks again

julian

Martin Koch March 7, 2012 at 12:33 PM

Thanks. No I haven't. It's a lot of trial and error. Good luck!

Reply

Georgia September 29, 2011 at 2:29 PM

Hi,

I am looking for the LANC commands to turn the camera on and off. I have a HXR-MC1.

Thanks for the great blog.

Reply

Martin Koch September 30, 2011 at 1:36 AM

Sorry, I can't help. You can try to find the command yourself using this:
http://controlyourcamera.blogspot.com/2011/02/finding-out-lanc-remote-commands.html Good luck.

Reply

Nishant Sood June 8, 2012 at 4:02 AM

I have made it work with the Vixia HF S20

Reply

Bert Meelberg July 17, 2012 at 7:00 AM

Did you have any luck finding the special commands of the ZR-2000?

Reply

Martin Koch July 18, 2012 at 12:51 PM

Hi Bert, sorry no luck.

Reply

Robert Culp July 31, 2012 at 8:07 PM

Any Similar sites out there to access the Panasonic Focus/Iris and Zoom/StartStop remote ports?

Reply

Daniel Nethery December 26, 2012 at 1:58 PM

Martin this blog has been an amazing resource. But now I'm trying to work out using a Lanc Controller to send commands and control devices trough my an Arduino.

Any thoughts on how I would make the Arduino emulate a video camera? When I do the sniffing on the controller to the camera. I can see the camera sending a constant 00000000 00000000 to the controller setting the sync.

But when I try to build a sequence for of code in the void loop function acting like the camera, then listening for a response from the controller. I can't seem to get any thing back. I don't know if it's my timing or the circuity.

I'm using the "Simple Start Stop" circuit layout shown above. But I have pulled out the start stop button.

Thanks for your thoughts and Help!

Reply

Replies

Martin Koch December 30, 2012 at 12:50 AM

Daniel I'm afraid that I don't understand what you are trying to do.

Reply

Andrew Hunter December 29, 2012 at 4:43 PM

I can confirm that this code works with the Canon C300.

Cheers and thanks for the well written article.

Reply

Replies

Martin Koch December 30, 2012 at 12:58 AM

Thanks for the confirmation Andrew.

Reply

Balázs Kelemen January 2, 2013 at 2:50 PM

The zener diode is to protect the Arduino from voltages higher than 5.1 Volts. Am I right?

Reply

Replies

Martin Koch January 3, 2013 at 2:09 AM

Correct. For a confirmation just follow the link to Ariel Rocholls page from where I got the circuit.

Reply

Marcus Wolschon November 13, 2013 at 11:38 AM

I'm reverse engineering the LANC command codes of the Blackmagic Pocket Cinema Camera.
http://marcuswolschon.blogspot.de/2013/11/bmpcc-lanc-and-sysclk-dx-usbee-dx-clone.html

I have already found 3 codes not mentioned anywhere else for iris+, iris- and autofocus.
I know there is a command for auto-iris but I haven't found it yet.

Reply

Marcus Wolschon November 13, 2013 at 11:40 AM

..oh and you can get rid of your -8ms adjustment by using the fastWrite macro I found. It makes the remote work much more reliable.
..and clean up your bit-banging while at it with the send_8() method I posted. (Link in the comment above.)

Reply

Martin Koch November 13, 2013 at 1:54 PM

Hi Marcus, it's great to see your work on the BMPCC. Thanks for the tips. Why is it that camera manufacturers make such a secret about their LANC implementation. I had no luck with

Canons support they said somthing like licensed work from Sony and that they can't give me any details. Well we find the codes nevertheless. Good luck!

Reply

Martin Koch November 13, 2013 at 2:12 PM

For "normal" LANC signals I had success using this: http://controlyourcamera.blogspot.co.at/2011/02/finding-out-lanc-remote-commands.html

Reply

潘承伟 December 9, 2013 at 11:43 AM

Hello, thank you for sharing your blog.

Now, I'm looking for the information of startbit and bit-banging. In the Arduino codes above, I can not find/understand the startbit that you write, for example by Byte0, after "while (pulseIn(lancPin, HIGH) < 5000 " comes a "delayMicroseconds(bitDuration)" where is the startbit? is this "delayMicroseconds(bitDuration)" means startbit?

When we connect the camera and the arduino like the circuit above, whether the camera send any signal to LANC line （Automatic）？

Do you have any idea or data about this, the information from http://www.boehmel.de/lanc.htm is not comprehensive.
and my E-mail is : pcw007pcw@hotmail.com
Thank you

Reply

Martin Koch December 11, 2013 at 2:38 AM

Hi, if you read the description above carefully again you will see that the start bit happens after the LANC line goes low after a long delay. So after waiting delayMicroseconds(bitDuration) - the start bit duration - you can write the two REC command bytes.

The camera always and automatically sends the 8 byte pattern over the LANC line. After pushing the push switch the code above detects the "long" delay between the patterns, waits 1 start bit and then sends the REC command before the next status bytes come. I hope it is clearer now.

Reply

Volker Kuchelmeister January 30, 2014 at 10:43 PM

I asked Black Magic about their LANC codes, this is the reply from product manager:

Please see the following LANC commands supported on our cameras:

Name bytes[1:0]

```
---------------------------------
Nop = 0x0000,
RecordStart = 0x3318,
RecordStop = 0x198C,
IrisIncrement = 0x5528, // Used for IDLE state
IrisDecrement = 0x5328, // Used for IDLE state
IrisRecIncrement = 0x2A94, // Used for RECORD state
IrisRecDecrement = 0x2994, // Used for RECORD state
IrisAutoAdjust = 0xAF28,
FocusShuttleFar = 0xE028, // Used for IDLE state (value mask 0x0F00: 1 3 5 7 9 B D F)
FocusShuttleNear = 0xF028, // Used for IDLE state (value mask 0x0F00: 1 3 5 7 9 B D F)
FocusShuttleRecFar = 0x7094, // Used for RECORD state (value mask 0x0700: 0 1 2 3 4 5 6 7)
FocusShuttleRecNear = 0x7894, // Used for RECORD state (value mask 0x0700: 0 1 2 3 4 5 6 7)
FocusFar = 0x4528, // Used for IDLE state
FocusNear = 0x4728, // Used for IDLE state
FocusRecFar = 0x2394, // Used for RECORD state
FocusRecNear = 0x2294, // Used for RECORD state
```

Reply

Replies

Martin Koch January 31, 2014 at 12:03 AM

Hi Volker,
that's very nice from you and also Blackmagic Design. Thanks for posting this.

Rebelj12a February 20, 2014 at 9:16 AM

Ive found that on the Blackmagic Camera it seems that when the chord is plugged in it does not send out 5v. When using the Arduino not plugged into USB its only sending out 0.81-0.82 v of power. Once the Arduino is on, however, it does send at least 4.5v which is lower powered, but enough to power the Arduino and start the program loop.

Is there an initial starter pulse or power that is needed in order to start up the LANC power or does it just draw that much. Thanks!

tinkthank May 5, 2014 at 4:27 AM

i think you did something wrong. my BMPCC outputs 5v and powers the arduino (nano) fine.

Reply

Zardox February 27, 2014 at 3:22 PM

Great contribution Volker,
I need to find out if there is a code to control the zoom for powered zoom lens in black magic pocket camera.

Reply

Jon McPhalen April 10, 2014 at 9:25 PM

If you find yourself running out of power with the Arduino you might consider the Propeller -- very powerful (with eight 32-bit cores), and easier to program than most know. I recently wrote a LANC object for a friend's multi-camera controller; it just showed well at NAB and has some government agencies interested.

With the Power of the Propeller I'm able to send commands as well as read the rest of the stream back -- all in its own processor so the mainline code is not affected. On Saturday I was able to extract timecode data from the stream, something I wanted to do as soon as I started on LANC.

There's thread you may find interesting in the Parallax Propeller forum

http://forums.parallax.com/showthread.php/148752-LANC-control/page2

Reply

BrentGossett July 3, 2014 at 9:06 PM

I'm having trouble with the sniffer. I'm able to get the code to load, I have the remote connected to the camera and the arduino. The remote controls the camera when the buttons a pushed; however, there is nothing displayed in the serial monitor.

Reply

rlajos November 20, 2014 at 7:13 AM

Its only me who cannot post is this forum?

Reply

Davey IJzermans November 7, 2015 at 10:47 AM

Thanks you all that have contributed to this resource. It's very helpful. I do have a question: if I were to scale this down to work on an ATtiny45 or similar, would I need to adjust the timings to go from a 20MHz clock to 8MHz/1MHz clock? Does anyone know or can estimate the changes needed to the delays in the code?

Reply

Replies

Martin Koch November 8, 2015 at 1:17 AM

Hi Davey, you'll have to try. Generally the clock rate should not influence the required BitDuration of 104 microseconds. 20 MHz are just more accurate.

Amer Tahir June 12, 2016 at 5:45 AM

I can report it working on Digispark (http://digistump.com/products/1) Attiny85 microcontroller just fine.

Reply

Jenny Hayes December 5, 2015 at 2:54 AM

Thank you so very much buddies! I like your all best ever posts daily here. My hat is off to you on your special working.
pop over to these guys

Reply

Phil July 29, 2017 at 10:08 PM

Hi Martin, not sure if you are still working on this but I'm trying to figure out how to capture responses from the Camera. Is there a way to get on-going record status from a camera via lanc?

Reply

Chu Nguyen August 12, 2017 at 1:33 AM

Thank for your amazing article. Please help me. I have a problem that i debug this line while (pulseIn(lancPin, HIGH) < 5000) then always get pulse to zero. I made following the circuit above and code. But it seems dont have any pulsein return. I used the sony alpha 5100. and modify the micro usb to get pin 8 into D11. Thank so much.

Reply

Unknown November 22, 2017 at 6:15 AM

This comment has been removed by the author.

Reply

Skredfare November 22, 2017 at 6:16 AM

This comment has been removed by the author.

Reply

Skredfare November 22, 2017 at 6:16 AM

Thank you so much for this brilliant guide and code! I'm using it to control several different camera traps to film wildlife: since no one seems to know how to control cameras with the new multi protocol, I'm using this code and an adaptor to translate lanc to multi — and it seems to work well most of the time.

However, sometimes the cameras seems not to respond to the rec-command. They start up all right, but when the Arduino sends the rec command, the camera does not respond, and eventually goes to sleep again. The Arduino is still stuck, waiting for a reply from the camera: if I start the camera manually and press record, the Arduino proceeds in the program I've written.

I'm therefore wondering where in the code the program is actually waiting for a reply from the camera, and if there is a way for this to time out, and try again. Does anyone know?

Reply

ardugnome March 26, 2018 at 7:43 PM

Great post, confirm that it works with Canon Vixia HF-G20 and G30.
Is there any way to read the zoom bits from a camera in order to find out if it is in tele or wide mode? Tried http://www.boehmel.de/lanc.htm but no luck finding out how to read zoom from the camera.
I want to use the information to pass it to an Arduino controller to control motor pan speed.

Reply

Tyler Winegarner April 17, 2018 at 8:37 PM

Thank you so much for the project and documentation! Do you know if the circuit needs to change at all for 3.3v logic?

Reply

Enter your comment...

Comment as: Matthew White

Sign out

Publish

Preview

□ Notify me

Newer Post

Home

Older Post

Subscribe to: Post Comments (Atom)