

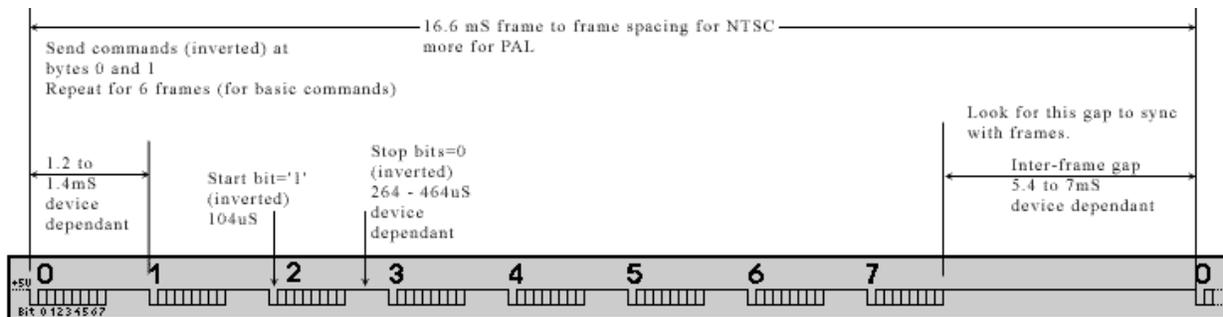
## Sony Lan-C & Control-L

To start your program will first look for (0) or high +3 to +5 vdc on LanC line (controlled by camera), for 1200 mS to 1500 mS this high is an Interframe gap that starts after the 8<sup>th</sup> byte and ends before the start of the 1<sup>st</sup> byte of the next frame set. The camera then pulls the Lanc line Low (1) 0 vdc for 104uS, This is the first start bit and where your 1<sup>st</sup> command byte should be sent which would be ( \$18, %0001,1000 ). You have 1.2 to 1.4 mS to send your 8 bits ( which take about 832uS ) then you wait for a second start bit ( sent by camera ) and after this second start bit, you send your 2<sup>nd</sup> command byte (on, off, rec., etc.). You will send your 1<sup>st</sup> command byte and second command byte three times on three consecutive frame sets.

Looking at the camera's 8 byte data stream byte's 0 and 1 are not used by the camera to send data. These two bytes are used by the camera to look for incoming data. Byte 0 is your first command byte slot, and byte 1 is your second command byte slot.

Byte 0	% 0001,1000	command byte	\$18	Hex18	Dec 24
Byte 1	% 0101,1110	power off	\$5E	Hex5E	Dec 94

You send the right digit first and move left until all eight bits are sent. Remember, both bytes must be sent each time. The \$18 must be sent first, followed by the \$5E in this example. This should be done 3 times.



( One complete telegram or frame takes about 20 mS. 1.2 to 1.4 mS per byte X 8 = 11.2 + 5.4 to 7 mS. Interframe gap= 18.2 mS camera's vary ).

An example of sending the camera a command, this is for the first command byte of \$18 would be as follows ( this example is using bit banging for ease in understanding the sequence ).

```

Command_byte:          'tells camera that a command is to follow (control command)
pauseus 80             'pause for camera's start bit to finish
input portb.2         'first byte to send (LSB 1ST) Hex 18 %0001,1000
pauseus 312
low portb.2
pauseus 208
input portb.2
pauseus 312
input portb.2         'Stop bits set high (pulled high by camera)
Return

```

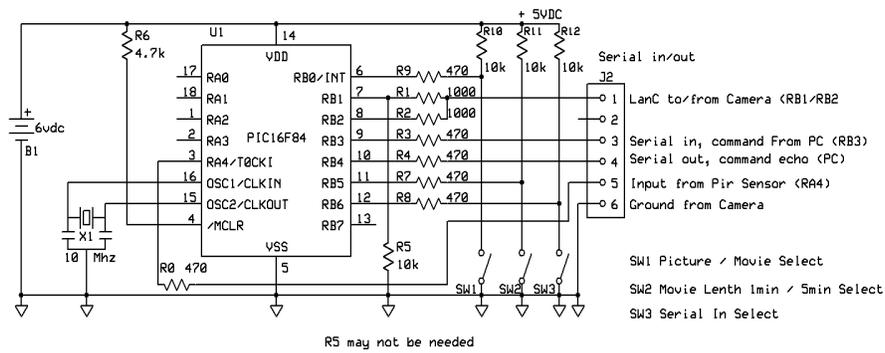
In this example the pauseus 80 was used instead of 104 uS, this was used because the program section prior to this, was looking for the start bit ( sent by camera ), and this program overhead pushed the start of my first bit over too far as seen on the oscilloscope, so I adjusted this wait period to fit. Now after this pause ( or start bit completes) the command byte of %0001,1000 or hex 18, \$18 is sent, the first bit is at the far right. This first statement input portb.2 for 312 uS keeps the lanc line pulled high ( pulled high by camera) for three bit cycles or 104 uS times 3 = 312 uS, which you see is 000 starting from the right. The next command is a low portb.2 which pulls the lanc line low to 0 vdc this is a 1, for 208 uS this will be the two 1's, each bit is 104 uS long.

Camera will not power on? I found that giving the LanC line a low (1) or 0 vdc command from 500mS starts the camera's data Frame, this data Frame only lasts for a few seconds then shuts down again. While this data stream is active, the camera will accept commands. My camera would not wake from a power switch off. The power on command only works after a 5 minute inactive timeout, or after issuing a power off command \$5E or \$2A.

Nightshot Plus (Sony) on my camera when I issue a power off command this also shuts off the nightshot plus when the switch is left in the on position. So, it's your choice, leave it on or off.

#### Lanc Commands that worked with my video camera

Camera Command	Hex #	Dec #	Binary
1.) Power On	\$5C	92	%0101,1100
<b>2.) Power Off</b>	<b>\$5E</b>	<b>94</b>	<b>%0101,1110</b>
<b>3.) Record On/Off</b>	<b>\$33</b>	<b>51</b>	<b>%0011,0011</b>
<b>4.) Photo Capture</b>	<b>\$39</b>	<b>57</b>	<b>%0011,1001</b>
<b>5.) Record On</b>	<b>\$27</b>	<b>39</b>	<b>%0010,0111</b>
<b>6.) Record Off</b>	<b>\$29</b>	<b>41</b>	<b>%0010,1001</b>
<b>7.) Photo Delay</b>	<b>\$2B</b>	<b>43</b>	<b>%0010,1011</b>
<b>8.) Power Off</b>	<b>\$2A</b>	<b>42</b>	<b>%0010,1010</b>



<b>Shelby Control Systems</b>		
<b>Sony LanC Controller</b>		
A.L.G.	Rev 1.0 9/1/2009	Page # 1