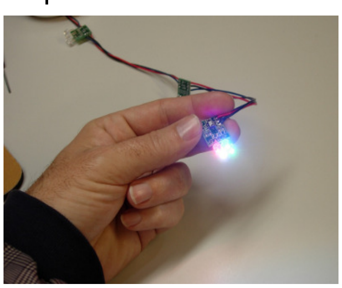
Hardware Summary:

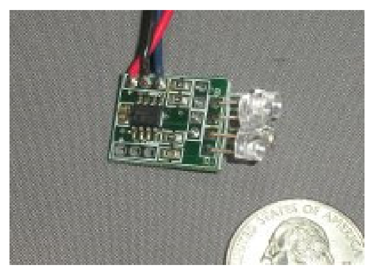
- Each lamp contains one pre-programmed Microchip 12F609 microprocessor.
    - Each lamp is capable of driving four LEDs. Each LED must draw no more than 25mA of current.
    - Each LED has a current limit resistor so that maximum brightness may be achieved.
    - Each lamp should have a unique network node address. Otherwise, two identical nodes will try to communicate on the bus at the same time.
- All the lamps connect together into a string of lamps.
    - Three wires connect all the lamps together into a string of lamps.
    - Each lamp required 5Vdc on two of the wires. The third wire provides for communications.



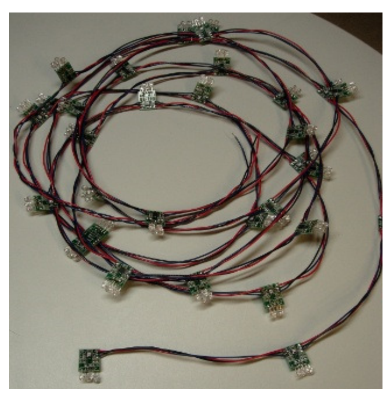
Kemper LED Lamp

Software Summary:

- Each lamp responds to half-duplex 9600 message packets.
    - Each packet contains a lamp address, command, data, and checksum.
    - Any packet directed to a specific lamps causes the lamp to respond with an acknowledgment.
    - There are almost a dozen commands available to control the lamps.
    - Each lamp behaves as a slave on the bus.
    - A master host must provide the commands to instruct the lamps.



Single Lamp



String of 32 Lamps

Device Overview & Description

The Kemper LED Lamp (KLL) is a purpose programmed microprocessor targeted at simplifying the task of controlling and driving multiple LED (light emitting diode) lights. Multiple lamps can be connected together in a daisy chain fashion. The lamp system is scalable, in that, as more lamps are connected together more embedded MIPS horsepower is added. In effect, the Kemper LED Lamp is an embedded ASIC capable of driving four channels of LEDs using programmable PWM signals that ramp, dwell, and decay as directed by a master node.

Each lamp is hard-coded with a embedded node address. Also, all lamps will respond to a global node address. Finally, each lamp is capable of being assigned three soft node addresses. Using soft node addresses, the lamps may be grouped. Groupings make it very easy to affect a subset of lamps on a single bus.

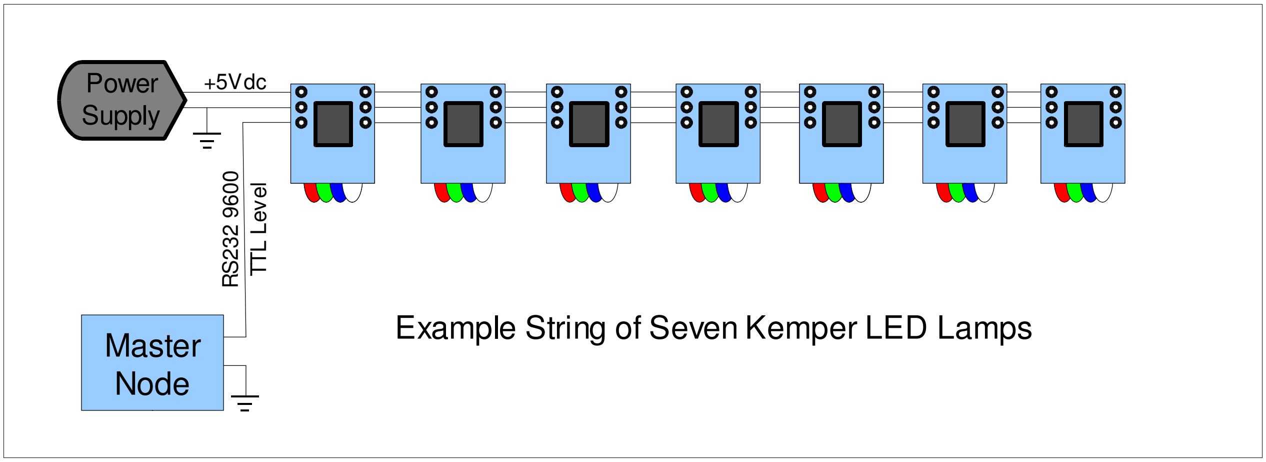
The main design goals were to keep the cost-per-lamp as low as possible, design the communication bus to facilitate easy wiring, and make the command protocol easy to use.

The Kemper LED Lamp offloads the burden of generating multiple PWM signals from the host computer. As an example, a demonstration system was shown that had 64 lamps connected to a single RS232 / USB port on a PC. With four LEDs on each lamp, that equates to 256 LEDs! Each LED was individually controllable using a PWM signal. By sending just four bytes of information for the PC, all the LEDs in the system could be commanded to perform an operation.

One of the unique features of the Kemper LED Lamp is the “Turbo Ping” command. Typically, the master node sends a global command to all lamps to start a Turbo Ping. Each lamp, in turn, broadcast their node address back on the bus to the master node. To eliminate collisions, each lamp uses their own unique node address to determine when to transmit on the bus. Each node, after receiving the Ping Command, starts a dwell timer that equals (1ms * node address). For example, node #10 would dwell for 20ms before transmitting. With a maximum of 254 nodes the Turbo Ping Command takes a maximum of 512ms to complete. Not bad considering that with just one command the PC can discover all the lamps on a bus within ½ second. So, kick off the command, wait ½ second, and then inspect the Rx buffer on the PC. Each byte in the PC's Rx buffer is a node address.

License

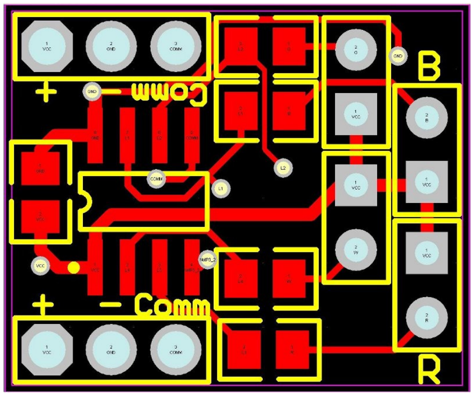
Electrical Specifications



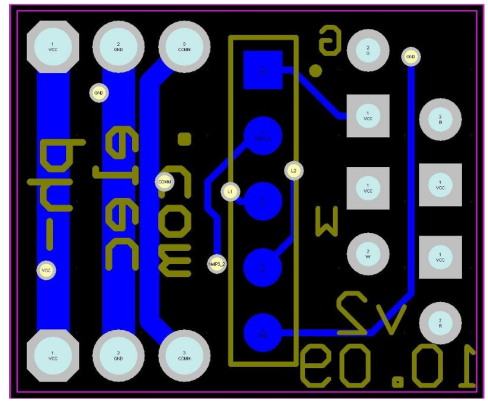
Hardware Description:

	Nominal	Max / Min Values / Notes
Voltage Supply:	+5Vdc	Max: 6.0V / Min: 2.5V
Node Current Consumption:	40mA	Max: 82mA / Min: 2mA
Temperature Range:		Max: 85℃ / Min: -40℃
Communications:	TTL	1 Start / 8 Data / 1 Stop
Comm Baud Rate:	9600	Max: 9650 / Min: 9550
Physical Size:	20mm x 15mm	±1mm
Microprocessor:	Microchip 12F609	
In-circuit Programming:	Two LEDs must be removed before reprogramming can begin. Programming in-circuit pads are located on bottom of PCB.	
LEDs:	Blue: 350mcd / 18 cents / 3.32V @ 20mA Green: 1500mcd / 22 cents / 3.06V @ 20mA White: 1500mcd / 25 cents / 3.55V @ 20mA Red: 350mcd / 17 cents / 2.00V @ 20mA	
LED Source:	http://besthongkong.com	

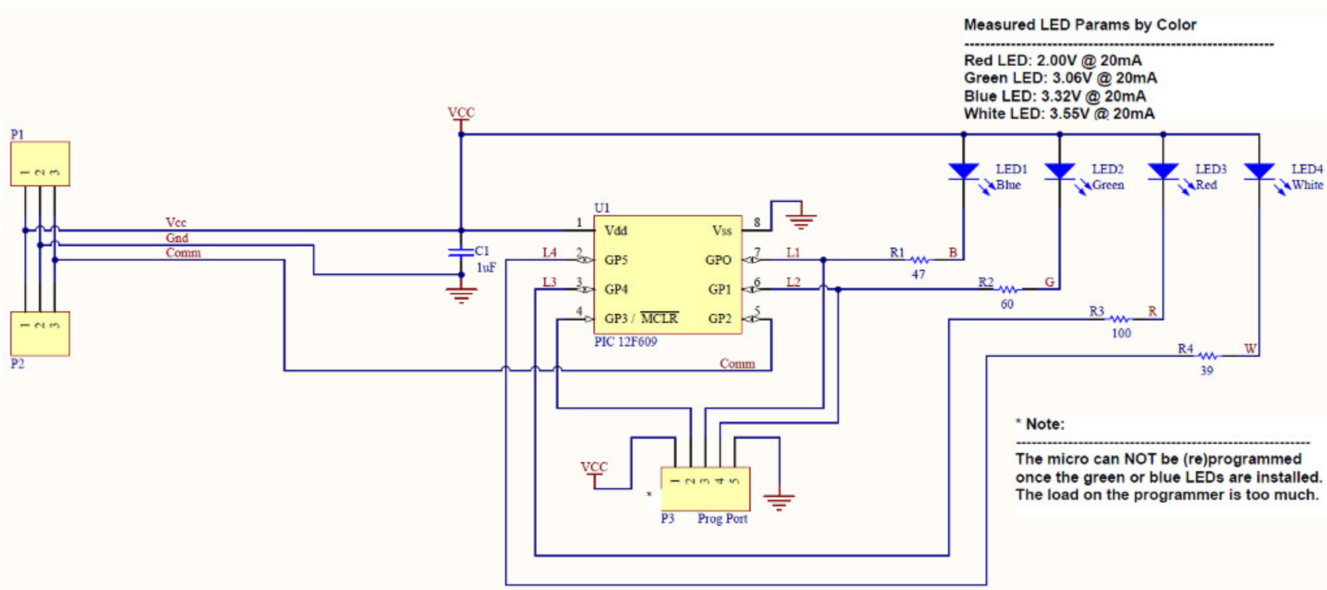
Circuit Board Design:



PCB Top Layer



PCB Bottom Layer



Kemper LED Lamp - Schematic

Kemper LED Lamp Bus Protocol

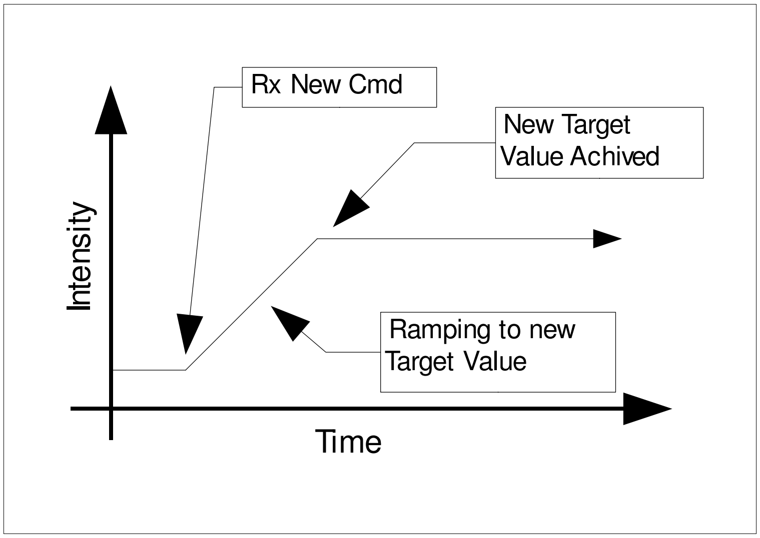
The Kemper LED Lamp (KLL) protocol consists of several bytes sent at 9600 baud, half duplex, at TTL logic level. A packet consists of a node address byte followed by a command byte and then several data bytes with a final checksum byte. The checksum byte is the modulo 256 addition of all preceding bytes. The command / data payload may very from a single byte up to six bytes. Therefore, a single packet can very from three bytes (node address + command byte + checksum) up to six bytes. Once a packet is started all remaining bytes must be transmitted within a timeout period of 60mS. This makes it possible to guarantee that all nodes are ready to receive a new packet – simply dwell for more than 60mS and all nodes are guaranteed to be ready to receive a new packet.

Command #	Command	Length
0	Reset	3
1	Set Pullup	3
2	Set LED On / Off	4
3	Quick Set LED Level	5
4	Set LED Ramp / Decay	5
5	Set LED Level	4
6	Start Turbo Ping	3
7	Sync PWM Timer	3
8	Set Ack Mode	4
9	Set 2nd Address	6
10	Pulse LED	4

Note: The length specified in the table includes the overhead of the node address and the trailing checksum. As an example, the reset command consists of (node address) + (reset command) + (checksum) for a total of three bytes.

Example Packet:

Node Address	Command	Data 0	Checksum
0x10	0xF5	0xFF	0x04



This example would command node 16 (0x10 hexadecimal) to drive all four LED output channels to 0xFF level. The command byte is split into two parts. The lower nibble in the command byte is the actual command. The upper nibble in the command byte is a bit-field where each bit maps to a LED channel. Upon successfully accepting the packet, node 16 will immediately acknowledge the packet by echoing his node address on the bus. Packet acknowledgment does not happen when the global address is used.

After a new command is received, the new target output value is compared to the existing output value. If the values are different, either the ramp or decay values are used to drive the output value to the new target value.

Each LED channel has a current output value, target output value, ramp rate, and decay rate. The target, ramp, and decay values may be set using commands. The Kemper LED Lamp simply drives the current output value to the target output value using the ramp / decay values.

It's important to remember that each LED channel can have an independent target, ramp, and decay rate. If a new command is received while the LED channel is already ramping, the new target and ramp/decay rate will take over. Also, each LED channel is independent of the other channels. So, even if one channel is ramping/decaying that will not affect another LED channel ramping/decaying to some other target value.

The ramp/decay tick rate is 60Hz. For example, ramping from zero to 255 at a ramp rate of one will take 255 * (1 / 60) = 4.25 seconds. This would be the slowest ramp rate. The default ramp/decay rate is 0x15 (21 decimal) – this equates to a ramp from 0 to 255 in 255 / 21 * (1/60) = 202mS. The fastest ramp/decay rate would be 255 – this equates to 255 / 255 * (1/60) = 16.6mS.

Node Address:

NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0

b7 b0

bit 7-0: **NA<7:0>:** Node Address
The Node Address specifies which node on the bus the following data packet is directed to. Each node has an embedded, hard coded, address. Each node also responds to a global address of 0xFF.

The node address is always a single byte. The value can range from zero to 255. However, the value 255 (or 0xFF in hexadecimal) is reserved as a global match address. This means that one bus can have as many as 254 nodes with each being individually addressable.

Command:

b7	b6	b5	b4	CMD3	CMD2	CMD1	CMD0

b7 b0

bit 3-0 : **CMD<3:0>:** Command
The packet command is formed in the lower nibble of the command byte. Using four bits, only the first 11 commands are defined. The remaining, possible, five commands are unused and are ignored. The upper nibble is used based on the command that is being executed. See specifics for each command below.

bit 7-4: The definition of bits 4 thru 7 depend on the type of command being executed.

Checksum:

CS7	CS6	CS5	CS4	CS3	CS2	CS1	CS0

b7 b0

bit 7-0 : **CS<7:0>:** Checksum
The Node Address specifies which node on the bus the following data packet is directed to. Each node has an embedded, hard coded, address. Each node also responds to a global address of 0xFF.

The node address is always a single byte. The value can range from zero to 255. However, the value 255 (or 0xFF in hexadecimal) is reserved as a global match address. This means that one bus can have as many as 254 nodes with each being individually addressable.

Command Definitions:

Reset: **Command = 0**

b7							b0
na	na	na	na	0	0	0	0

bit 3-0 : **CMD<3:0>:** Reset Command = 0
After receiving the reset command, the node will execute the same initialization function used after a power-up reset. The follow items are initialized.

1) All LED output channels are turned off.
2) The comm pin weak pull-up resistor is turn on.
3) Ramp and decay rates are reset to 0x15.
4) Alternate node addresses are reset to 0xFF.
5) Node acknowledge is turned on.

All initialization is completed in less than 1mS. After initialization the node is ready to accept the next command.

bit 7-4: Bit 4 thru 7 are ignored and not used.

Set Pullup: **Command = 1**

b7							b0
PU3	PU2	PU1	PU0	0	0	0	1

bit 3-0 : **CMD<3:0>:** Set Pullup Command = 1
This command controls the comm pin pull-up resistor. After power-up (or a reset command), the pull-up resistor is turned on by default.

bit 7-4: **PU<7:4>:** Pullup Control
If PU > 0 then the comm pin pull-up resistor is switch on. Else, on PU = 0 then the pull-up resistor is switched off. Having the pull-up on will reduce noise on the comm bus. However, for very long strings, a single terminating resistor may be used instead with all node pull-up resistors turned off.

Set LED On / Off: **Command = 2**

R	G	B	W	0	0	1	0
IND3	IND2	IND1	IND0	SR	SG	SB	SW

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Set LED On / Off = 2
This command can be used to jump one, or more, LED channels to a new target level. The target levels are defined from the table below. The table values were chosen to give a linear light output appearance.

bit 4: **W:** White If true, this command will affect the white LED channel.
bit 5: **B:** Blue If true, this command will affect the blue LED channel.
bit 6: **G:** Green If true, this command will affect the green LED channel.
bit 7: **R:** Red If true, this command will affect the red LED channel.
If the color bit is cleared, the LED channel is left untouched.

Byte 1:

bit 0: **SW:** Set white LED channel.
1 = Set white LED target level to to index value.
0 = Set white LED target level to zero.
bit 1: **SB:** Set blue LED channel.
1 = Set blue LED target level to to index value.
0 = Set blue LED target level to zero.
bit 2: **SG:** Set green LED channel.
1 = Set green LED target level to to index value.
0 = Set green LED target level to zero.
bit 3: **SR:** Set red LED channel.
1 = Set red LED target level to to index value.
0 = Set red LED target level to zero.

Bit 7-4: **IND<7:4>:** Index values

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	4	13	20	28	35	48	59
9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
72	88	104	120	143	167	199	255

The index values are used to jump the LED channel target value to a new level.

* Tip: Some LED channels can be set to an index value while other channels can be set to zero. Use the SR, SG, SB, and SW bit to indicate if an index value is used or a zero value is used. So, with one command, all the LED channels can be set to either an index value or zero.

Quick Set Level: **Command = 3**

R	G	B	W	0	0	1	1
IND3	IND2	IND1	IND0	na	na	na	na

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Quick Set Level = 3
This command can be used to jump one, or more, LED channels to a new target level. The target levels are defined from the table below. The table values were chosen to give a linear light output appearance.

bit 4: **W:** White If true, this command will affect the white LED channel.
bit 5: **B:** Blue If true, this command will affect the blue LED channel.
bit 6: **G:** Green If true, this command will affect the green LED channel.
bit 7: **R:** Red If true, this command will affect the red LED channel.
If the color bit is cleared, the LED channel is left untouched.

Byte 1:

bit 0-3: not used

bit 7-4: **IND<7:4>:** Index values

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	4	13	20	28	35	48	59
9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
72	88	104	120	143	167	199	255

The index values are used to jump the LED channel target value to a new level.

Set Ramp / Decay Level: **Command = 4**

R	G	B	W	0	1	0	0
R7	R6	R5	R4	R3	R2	R1	R0
D7	D6	D5	D4	D3	D2	D1	D0

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Set Ramp / Decay Level = 4
This command is used to change the ramp / decay rate of a LED channel(s). Note, each LED channel maintains a separate ramp / decay set of values. Therefore, each LED channel may be set to different values.

bit 4: **W:** White If true, this command will affect the white LED channel.
bit 5: **B:** Blue If true, this command will affect the blue LED channel.
bit 6: **G:** Green If true, this command will affect the green LED channel.
bit 7: **R:** Red If true, this command will affect the red LED channel.
If the color bit is cleared, the LED channel is left untouched.

Byte 1:

bit 7-0 : R**<7:0>:** Ramp Rate

Byte 2:

bit 7-0 : D**<7:0>:** Decay Rate

The ramp / decay rate controls how quickly the current LED PWM value is changed. The ramp / decay rate is based on a 60Hz clock. Thus, the tick rate of the clock is $16^{2/3}$ms.

Here is an example, suppose the current LED value is zero and the ramp rate is set to 5. Then, a new command is received to set the desired (target) LED value to 0xFF (fully on at 255). The firmware in the lamp starts slewing the current LED output towards the target value of 0xFF. At each tick, the firmware adds 5 to the current value. This continues until the target value is meet. In this example, a ramp from zero to 0xFF at 5 counts per tick would take (Finish Value - Start Value) / ((60cnts / sec) * 5) = (0xFF – 0) / (60 * 5) = 0.85 seconds.

Set LED Level: **Command = 5**

R	G	B	W	0	1	0	1
T7	T6	T5	T4	T3	T2	T1	T0

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Set LED Level = 5
This command is used to change the target value of a LED channel(s). Note, each LED channel maintains a separate target value.

bit 4: **W:** White If true, this command will affect the white LED channel.
bit 5: **B:** Blue If true, this command will affect the blue LED channel.
bit 6: **G:** Green If true, this command will affect the green LED channel.
bit 7: **R:** Red If true, this command will affect the red LED channel.
If the color bit is cleared, the LED channel is left untouched.

Byte 1:

bit 7-0 : T**<7:0>:** Target LED Output Value

The lamp firmware contains an infinite loop that continually drives the current LED value toward the target LED value. The ramp / decay rate is used to determine the amount of adjustment that happens during each update tick (60Hz).

Even if a LED channel is slewing toward a new target value, if a new target value is received, the LED channel will start to slew to the new value upon the next update tick. Therefore, even if a LED is slewing, new values may be sent to the LED to change the direction, or rate.

Start Turbo Ping: **Command = 6**

na	na	na	na	0	1	1	0

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Start Turbo Ping = 6
This command starts the turbo ping response process.

Note, once this command is received no other command may be received for 512ms.

The turbo ping process is as follows, the node starts a dwell loop (while maintaining the LED output channels). The length of the dwell is based on the node's hard-coded address multiplied by 2ms. So, as an example, if the node address was 33, the dwell would last 66ms. After the dwell time is complete the node broadcasts the node address back onto the bus.

Usually, the turbo ping command is issued with the global node address. This results in all nodes on the bus receiving, and executing, the turbo ping command at the same time. As a result, each node broadcast their node address during a given timeslot on the bus that is guaranteed not to conflict with another node.

Reset PWM Timer: **Command = 7**

na	na	na	na	0	1	1	1

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Reset PWM Timer = 7
This command causes the internal PWM timer set to get reset.

Using the global node address, this command can be used to synchronize all the node's PWM timers. This becomes important if video taping the lamps. A typical video camera records at 30 frames per second – with the LED PWM timer operating at 60Hz a “beat frequency” gets established between the LED light and the video camera. There can be moments where, while viewing on the video recorder’s LCD monitor, that the LED is not on. However, by looking at the LED, one can clearly see that the LED is, in deed, truly on. The LED sends a burst of light between recording frames of the video recorder. With this command, at least all the lamps will appear to be synchronized.

It could be possible to synchronize the lamps to a running camcorder. Our camcorder outputs a standard video signal while anything is being displayed on the LCD monitor. By triggering on the sync pluse in the video signal it would be possible to synchronize the Kemper LED Lamps be issueing a “Reset PWM Timer” command in sync with the camcorder. In this way, all the Kemper LED Lamps on a given string would be synchronized with the camcorder frame rate.

Set Ack Response: **Command = 8**

FLG7	FLG6	FLG5	FLG4	1	0	0	0
DWL7	DWL6	DWL5	DWL4	DWL3	DWL2	DWL1	DWL0

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Set Ack Response = 8
Used to either enable, or, disable the command acknowledgment. After power-up (or a reset command), command acknowledgment is enabled by default.

bit 7-4: **FLG<7:4>:** Acknowledge Flag
FLG > 0 then the command acknowledgment is switch on. Else, on FLG = 0 then the command acknowledgment is switched off.

Byte 1:

DWL<7:0>: Pulse Dwell Time
The pulse dwell time is used by the "Pulse LED" command (see Pulse LED command #10).

After a packet is successfully received (good checksum) the node sends an acknowledgment by responding on the bus by transmitting the hard-coded node address. Note, the node only acknowledges packets addressed to the hard-coded node address. The node does not acknowledge packets addressed to the global node address or soft node addresses.

Set Additional Addresses: **Command = 9**

na	na	na	na	1	0	0	1
2nd 7	2nd 6	2nd 5	2nd 4	2nd 3	2nd 2	2nd 1	2nd 0
3rd 7	3rd 6	3rd 5	3rd 4	3rd 3	3rd 2	3rd 1	3rd 0
4th 7	4th 6	4th 5	4th 4	4th 3	4th 2	4th 1	4th 0

b7 b0

Byte 0:
bit 3-0 : **CMD<3:0>:** Set Additional Addresses = 9

Byte 1:
bit 7-0 : **2nd <7:0>:** Set 2nd Address

Byte 2:
bit 7-0 : **3rd <7:0>:** Set 3rd Address

Byte 3:
bit 7-0 : **4th <7:0>:** Set 4th Address

After power-up, by default, a node will respond to either the global node address or the hard-coded address. Using this command, it is possible to assign three more soft node addresses. This makes it possible to place nodes into groups and address them all together. Thus, a subset of the nodes make be directed (with a single quick command) to act independently of the remaining nodes.

Note, after power-up (or a reset command), the additional node addresses are cleared back to match the global node address. This, in effect, turns them off. Also know, no acknowledgments are sent when using soft node addresses.

Pulse LED: **Command = 10**

R	G	B	W	1	0	1	0
TAR7	TAR6	TAR5	TAR4	TAR3	TAR2	TAR1	TAR0

b7 b0

Byte 0:

bit 3-0 : **CMD<3:0>:** Pulse LED = 10

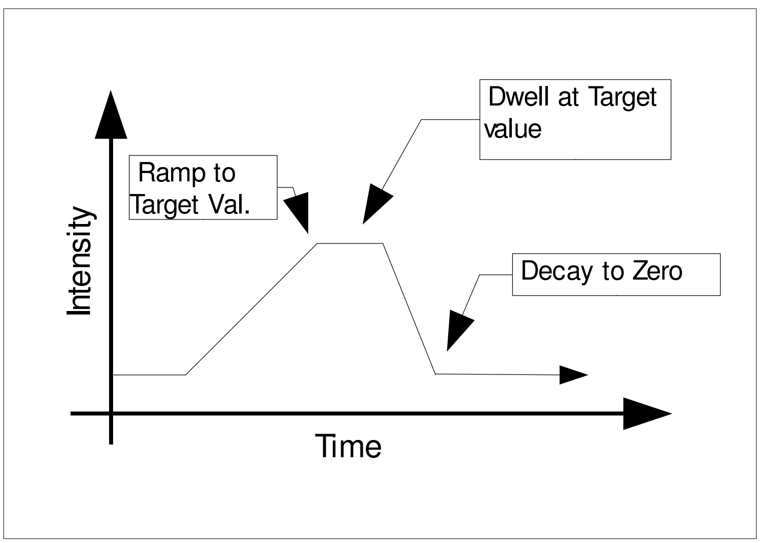
bit 4: **W:** White If true, this command will affect the white LED channel.
bit 5: **B:** Blue If true, this command will affect the blue LED channel.
bit 6: **G:** Green If true, this command will affect the green LED channel.
bit 7: **R:** Red If true, this command will affect the red LED channel.
If the color bit is cleared, the LED channel is left untouched.

Byte 1:

bit 7-0 : **TAR<7:0>:** Target Pulse Level

This command is design to make it quick and easy to pulse LEDs.

Upon receiving this command, the LED target value to updated with the value in byte 1. The firmware drives the current LED value towards the target value using ramp / decay values. Once the target value is achieved, a dwell time begins (see command #8 to set the dwell timer). When the dwell timer ends, the LED target value is set to zero.



Default dwell time is 16 counts (16 / 60Hz = 266ms). Use command #8 to adjust dwell time. The ramp / decay rates are the normal values used throughout and may be adjusted using command #4.