



**InternationalLight**  
TECHNOLOGIES

10 Technology Drive  
Peabody, MA 01960  
Ph: 978-818-6180  
Fax: 978-818-6181  
Web: [www.intl-lighttech.com](http://www.intl-lighttech.com)

**THEORY OF OPERATION  
AND  
CUSTOM LABVIEW CODE  
FOR  
ILT1000/ILT5000**

## 1. Theory of operation:

### ILT1000 and ILT5000 “USB Serial Port” Programming / Theory of Operation

Below is the Theory of Operation for interfacing with the ILT1000 or ILT5000 over a serial port. “Device” below refers to either the ILT1000 or ILT5000, as the Theory of Operation is the same.

- The device responds to a number of commands (together known as the command-line API) delivered via the USB port.
  - o Please visit the ILT website for the full API  
<http://www.intl-lighttech.com/support/manuals-documentation>
- The USB port is configured on the device as a USB Serial Port. As a result, interfacing can be done with any standard terminal program (hyperterminal, putty, MAC terminal window, etc).
- The device behaves as follows and in sequence:
  - a. Perform analog/digital conversion and mathematical functions as part of optical level detection
    - i. Buffer up to 4 characters of any incoming commands while performing the processing above
  - b. Check for any characters in the buffer referenced above to indicate an incoming command and process any commands as required. Important programming notes:
    - i. It can take up to 25ms for the system to complete complex analog/digital tasks and process the remainder of the command that was not buffered.
    - ii. The device determines the completion of the command by detecting a “\r” character, i.e. ASCII code 13 decimal.
  - c. Start all over from the top at (a)...
- As a result of the above device behavior, the recommended method to program each device is to:
  - o The device operated at 115200 baud, 8 bits, 1 stop bit, no parity bits, no flow control.
  - o Always append commands with \r to indicate the command completion.
  - o Send the first character of the command, for example the “g” in “getcurrent\r”.
  - o Pause 50ms (this can be lowered for high speed access, but start with 50ms)
  - o Send the remainder of the command, i.e. “etcurrent\r”. There are no pauses required between characters as the device is rapidly draining the incoming serial port at this point.
  - o Immediately start sensing the response from the device.
    - The device will always send a response to acknowledge the status of the command completion as well as to return values for “get” commands.
    - The response will always be terminated with a “\r\n” (13 decimal, 10 decimal) sequence. This can be used to sense the end of the response and start sending the next command.
      - Regarding time-outs for command responses, “get” commands will typically respond within 100ms. “set” commands that store their configuration in flash memory can take up to 5 seconds to respond. Other special commands like “setfactorydark”, “setuserdark”, and “captureflash” can take longer to respond.

- Each ILT1000/ILT5000 device is single threaded, meaning that commands and responses need to be processed in sequence. A 2<sup>nd</sup> command cannot, for example, be initiated before the 1st command's response is fully processed. As a result, if a multi-threaded application is accessing the device, a programmatic lock must be placed around device command/response sequences to make sure multiple threads do not attempt to access the device at the same time.
  - o If multiple devices are being monitored, a single lock can be used for access to all devices. This is simpler from a coding perspective, but not as efficient as it does not allow multiple devices to operate in parallel. For the best performance it is recommended that a per-device lock be established. This allows all devices to be accessed in parallel.
  - o As a further performance benefit when monitoring multiple devices, the delay after the first character can be performed in parallel across all devices. For example, if sending "getcurrent\r" to 5 devices, one would:
    - Send "g" to all 5 devices
    - Wait 50ms
    - Send "etcurrent\r" to all devices
    - Process the reply from all devices

## 2. Labview code, Sample program:

You can download a copy of our labview vi from the ILT software page on the ILT website:

<http://www.intl-lighttech.com/support/software>

To send command to the ILT1000, you need to send the first character, wait 50ms, then send the remaining characters. You can do this in a couple of ways:

1. Send the characters 1 by 1 and leave a 50ms delay between each. This is how the example Labview code works to simplify the block diagram.
2. Send the characters 1 by 1, leave a 50ms delay after the first characters, then send the others without delay.

Others notes:

1. \r needs to be the only trailing characters to indicate the end of the command
2. On receipt, read until you get a \n

For this device, the only setup should be for serial port: 115200,N,8,1,no HW or SW handshakes.

Please refer to the full API for a list of all available commands.

<http://www.intl-lighttech.com/support/manuals-documentation>