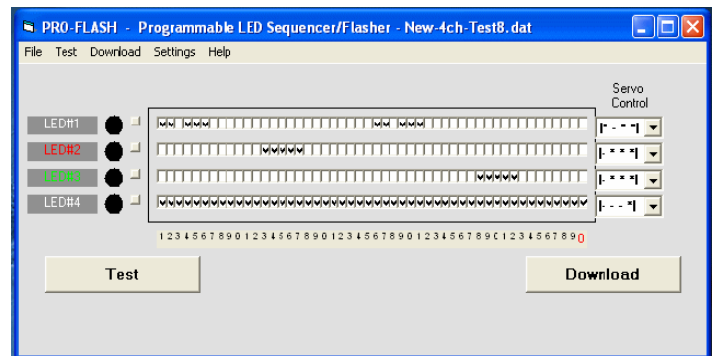


# ***User's Manual***

## ***"Mini-Flash"***

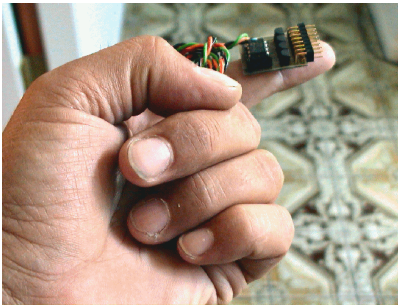
### ***Programmable, R/C-controlled LED Flasher/Sequencer!***

The ***"Mini-Flash"***, is a miniature, light-weight, programmable LED Sequencer/Flasher that is based around a small programmed PIC microcontroller. The Mini-Flash controller includes 4 programmable LED sequencing channels, although it allows for a total of 8 LEDs to be connected (2 LEDs per channel). Probably the most unique, cutting-edge feature about the Mini-Flash is the ability to customize the LED flashing sequence using a [PC-based \(i.e. Windows\) program](#). This programming software includes a unique and intuitive Graphical User Interface. Mini-Flash's servo wire which is normally plugged into a spare R/C receiver channel for control, is also used to reprogram the Mini-Flash



flasher/sequencer. A computer's serial port (9-pin) connection and programming cable is only needed to reprogram the Mini-Flash. The free downloadable programming software also allows the user to "test" the LED sequence they develop by use of software-based graphical representations of the 4 LED channels. The sequencer speed can also be programmed by the user, as well as the sequencer end marker (you don't need to use all 50 events in your sequence). Another great feature of the Mini-Flash controller is that each of the 4 LED channels can be controlled (on/off) using a spare R/C receiver channel. This gives the user the ability to remotely turn on or off any of the 4 LED channels, even while your model is being operated. How neat is that?

# Mini-Flash Controller Specifications

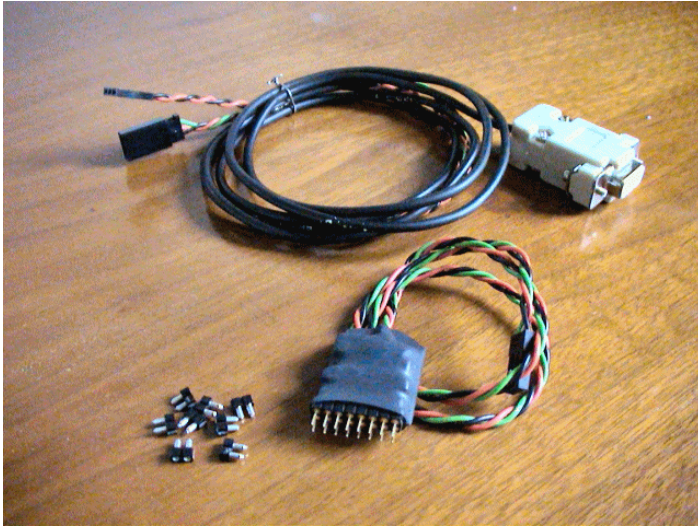


Weighing merely 9 grams and roughly 1 square inch by 1/4" thick, the Mini-Flash is based around a tiny 8-pin programmable microcontroller chip. The controller can get its power from either an R/C receiver or an external battery (normally, it is plugged into a receiver). The Mini-Flash is designed to operate off of a 4.8-5 volt power source therefore do not attempt to run it off of anything else or you will ruin the microcontroller. The Mini-Flash

provides for 4 independent LED sequencer channels that are driven using individual power output transistors (2N2222). These output transistors can typically provide up to 400 milliamps of total current, therefore, you are capable of connecting multiple LEDs to each channel if required. In fact, theoretically speaking, you should be able to connect at least 10-15 LEDs to each channel (assuming each LED draws around 25 milliamps each). Each of the 4 Mini-Flash LED channels can have up to a maximum of 50 individual events (an event either turns the LED on or off).

# of LED Channels	4 LED Channels (although Mini-Flash has two connectors available for each of the 4 channels)
# Events/Channel	50 maximum (User can set for less if desired)
Operating Voltage	4.8 volts to 5.0 volts (a filtered/regulated source is recommended, such as a BEC)
Operating Current	With out LEDs connected, roughly 15mA (LEDs draw roughly 25mA each)
Total Output Current	Each Transistor handles at least 400mA (so 400mA x 4 = 1.6 amps total)
Dimensions	Roughly 1 square inch, not counting the connector (a little over 1/4" thick)
Weight	Roughly 9 grams <u>total</u> weight (includes completed controller, servo lead and LED connector)
Programming Software	Tested on Windows 98, 2000, ME and XP (developed using MS Visual Basic 5.0)

# Mini-Flash Kit Contents



The picture to the left indicates exactly what is included in the Mini-Flash kit : (1 controller, one programming cable and eight 2-pin LED mating connectors). Remove the contents from their plastic packing bags and be sure to not loose the small miniature LED mating connectors. These connectors mate to the controller's LED connections. To use these connectors, you will need to solder on some small gauge, two conductor wire, a series resistor ([see below](#)) and finally your

LED. If you require more mating connectors, these can be purchased through DIYRC.com (send an email).

**Note:** Some pre-manufactured LEDs (for systems such as Curtek) can easily plug right onto Mini-Flash's LED connection! The output pins currently shown on the Mini-Flash above are "machine pin" headers whereas Curtek uses a normal "square pin" header. At your request, I can solder on this type of connection instead of the machine pin header (although, the machine pin headers are higher quality and in fact lighter in weight). Email me with your request in any case.

**Note:** If you are using a BEC to power up your receiver, and you have the Mini-Flash plugged into the receiver for power (and control), you will need to be sure the BEC is rated for the total current that will be drawn, that being the combination of the connected servos (normally powered from Rx), the receiver itself, the Mini-Flash itself and the LEDs connected to the Mini-Flash.

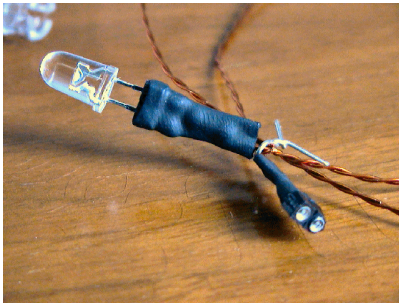


# Mini-Flash Connections

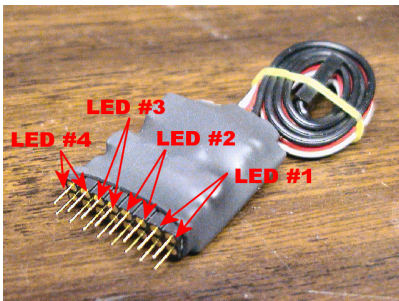
Assemble your LEDs using the supplied mating connectors. Also be sure to add an appropriate series resistor in series with one of the LED leads as you need to limit the current supplied to the LED as to not exceed its maximum specified operating current, otherwise LED damage will occur. ([See Notes section below](#))



The LED mating connectors supplied provide a nice tight connection to the LED connector pins located on the Mini-Flash. If you are going to heat shrink these ends, be sure you do not cover the black portion of the connector as this will not allow you to plug these connectors side-by-side onto the Mini-Flash (heat shrink makes the connectors too wide). If you are only using 4 LEDs (one per channel), then you can skip every other connector.



Only apply heat shrink around the metal portion of the connectors and connecting wires, especially on the connector side that will connect to the Mini-Flash controller (see picture). Also notice that this LED harness has female connectors on both sides. This allows you to replace LEDs (i.e. change a color). The LED connectors include a 100 ohm resistor (inside the black heat shrink tubing).



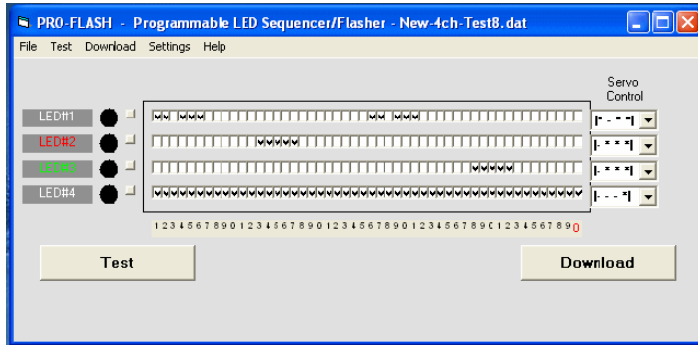
The picture to the left indicates the LED channel outputs. Notice that each of the 4 channels has two outputs, allowing you to connect two LEDs per channel. Add more LEDs to a channel by using a "Y" adapter or simply solder multiple LED lead wires onto the connector. As shown in the picture, the pins along the top are ground (-) while the pins along the bottom are positive (+).

The servo-like cable/plug leading out of the Mini-Flash serves two purposes. One, it is used to download a sequence from your PC (via programming cable) to your Mini-Flash and two, it is used to get servo information and power from your receiver during normal use such as in your R/C model.

**NOTE:** Be sure that when you connect the Mini-Flash cable to either your receiver or programming cable, you pay close attention to voltage polarity. As with most servo wires, the black or brown wire is ground, the orange or red wire is +V while the green or white wire is the servo signal. Plugging this cable in wrong could and probably would damage the Mini-Flash.

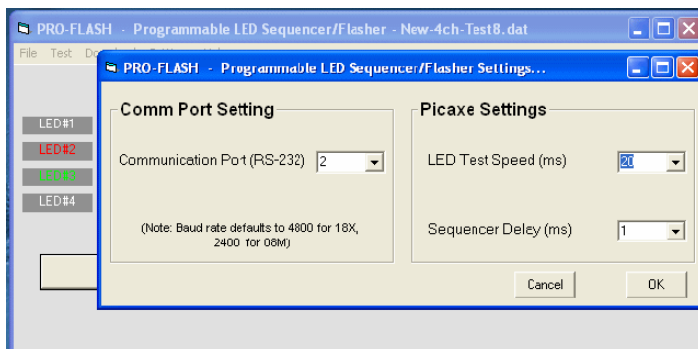
# Programming the Mini-Flash

Probably the neatest feature of the Mini-Flash is the ability for the user to completely customize the flashing sequence for each of the 4 LED



channels, and then download it to the Mini-Flash. Not only are these sequences programmable, but the user can also program the overall LED sequencing speed. Another really neat and very useful feature is the "Test" feature where you can actually visually test the sequence you programmed even with out the Mini-Flash connected. Pressing the test button in the software will run the sequence while visually displaying the actually sequence on GUI-based LEDs. This test function also allows you to test servo control function too by popping up a slider control which emulates the R/C transmitter's stick/slider control. The following sections explain each of the programming functions such as general Program Settings (communications port and sequencer speeds), setting the graphical LED colors, setting LED Channel Sequences, using the LED channel options button, setting the end of the sequence, assigning servo control functions, testing the sequence, opening, merging and saving sequence files on your PC and finally downloading your sequence to the Mini-Flash.

## Changing the General Program Settings (Comm Port and Sequencer Speeds):

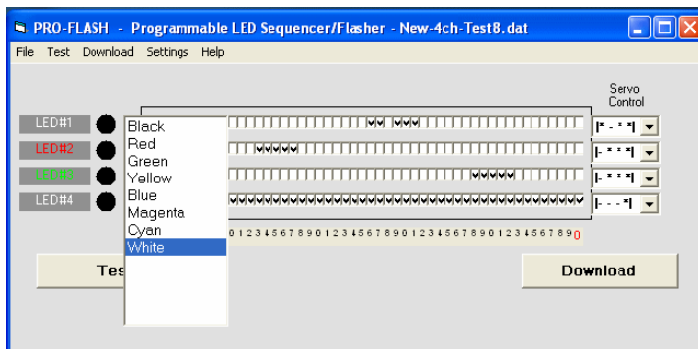


The Mini-Flash programming software allows the user to alter the Mini-Flash sequencer speed. Additionally, the software also allows the user to select the proper communications port to which the download cable is connected to. These settings can be changed by selecting the "Settings" menu option. Selecting this option will bring up a new

Settings window. On the far left side, there exists an options frame that allows the user to change the communications port (1->6). Use the pull-down and select the port # to which you have the programming cable connected (it might in fact default to port 2 which you might not have; typically, this would be set to port#1). Once you set this correctly, this setting is saved in a configuration file and will never need to be changed again unless you plug the programming cable onto a different port. On the far right of the Settings window, you will see two modifiable settings mainly "LED Test Speed" and "Sequencer Delay". The first, "LED Test Speed" allows the user to fine

tune the sequencer's speed when using the "Test" button. Although the software uses timers, I have found that some older computer can run sluggish. This setting usually defaults to 20 (20 milliseconds) and should suffice on most modern computer. You can change this setting (up or down) to alter the test speed, such that it closely matched that of the real Mini-Flash sequencer. The "Sequencer Delay" option allows the user to alter the actual running speed on the Mini-Flash unit (this value is also sent to the Mini-Flash" when downloading a sequence). This "delay" value indicated the actual delay between each of the 50 programmed events. A delay value of 1 (1ms) will make the Mini-Flash run as fast as it can. Increasing this delay value will slow down the running of the sequence (increases the delay time between events). Experiment with these values until you are happy with the results. Typically, a delay setting of 10-15ms will suffice for most applications.

### **Setting GUI LED Colors:**



The Mini-Flash programming software emulates the LEDs connected to the Mini-flash controller and these LEDs (LED#1 - 4) are shown towards the left of each of the 4 sequence event rows. Initially, when the software is loaded, these color default to black and you will need to set these. To select an LED color, click on the LED name (LED#1-

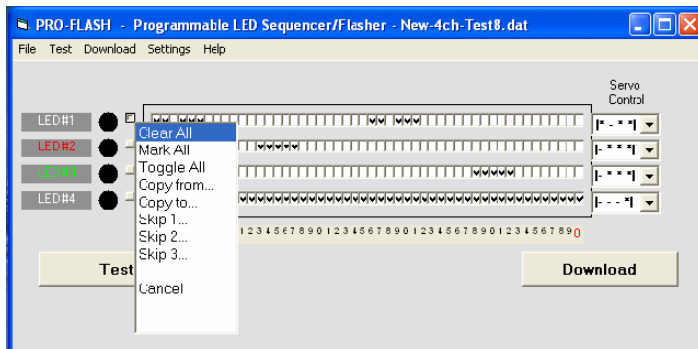
4) with your mouse and a window will pop up with various color names. Simply select the appropriate color (maybe even the color LED you plan to actually hook up to your Mini-Flash). Upon selecting the color, the LED text color will also change indicating the color the graphical LED will appear when triggered. Do this for the remaining graphical LEDs.

**NOTE:** If you programmed a sequence, pressed the "Test" button and do not see any graphical LEDs flashing, you might have forgotten to set the graphical LED colors. Stop the test, set the colors and rerun the test.

### **Setting LED Channel Sequence/Events:**

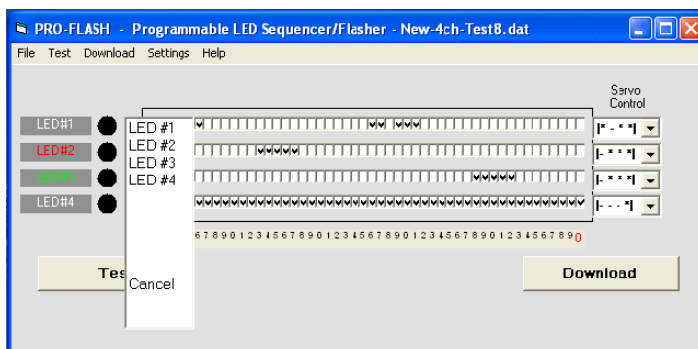
The user has 50 events per channel that are capable of being programmed where an event is either On or Off, as indicated with a check mark if On. You can set the sequence for each channel by using your mouse and mouse button (you can even hold and drag across the row to mark/unmark). Click on the individual checkboxes to either set or unset the individual sequence event. Another quicker method to set a particular channel's sequence is to use the Channel Options button (small square button just to the right of the graphical LEDs) which is explained below.

## Using the LED Channel Options button:



The Channel Options button allows you to manipulate the selected channel's sequence in that you can quickly:

<b>Clear All</b>	Turns OFF all events on selected channel
<b>Set All</b>	Turns ON all events on selected channel
<b>Toggle All</b>	Toggles the state of all selected channels events (if event is on, it is turned off, if event is off, it is turned on)
<b>Copy To</b>	Copy the selected channel (all events) to another channel (user then selects channel to copy to)
<b>Copy From</b>	Copy another channel's events into the selected channel (user then selects channel to copy from)
<b>Skip 1</b>	Turns on every other event on selected channel
<b>Skip 2</b>	Turns on every 3rd event on selected channel
<b>Skip 3</b>	Turns on every 4th event on selected channel
<b>Cancel</b>	Cancels the Channel Options action, returns to GUI



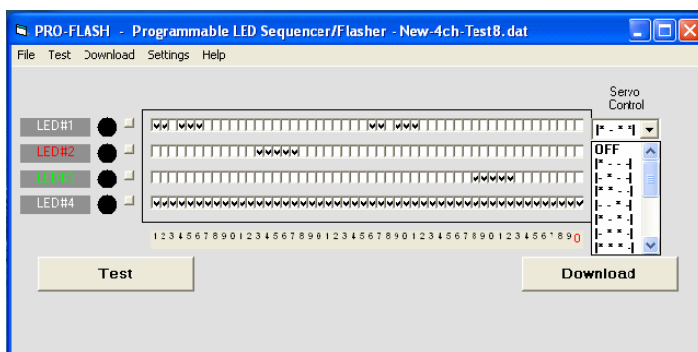
If you selected either the "Copy To" or the "Copy From" option, you will see this pop-up window shown to the left, allowing you to choose a LED channel (LED#) for which to either copy to or to copy from. Upon selecting one, you will see the selected channel's sequence events update accordingly.

## Setting the LED Sequence "End Marker":

Normally, the Mini-Flash programming assumes that the user will use all 50 events when developing a new sequence. There might be a time when the user might only want to use 25 events instead of 50. If you were to look closely at the programming software, there exists a grayish line across the bottom of the sequencer that has numbers (1->10 or really 1->0) that cycle 5 times, indicating the 50 individual events.

This "event bar" is also used to indicate the displayed events as they are being played back during testing. The programming software indicates the sequence "end mark" by painting a number on this bar RED. Normally, when starting a new sequence, you will see the far most "0" painted red (event #50). This indicates that this event will be the last event in the sequence and will then loop back around to the first event (Event #1), repeating indefinitely. To change this setting, simply "double-click" on another number located on this bar until you see it change to red/bold text. The sequence end mark has now been changed. If you were to now press "Test", you will see that the test sequencer now cycles up to the newly selected end marker and then loops back to the first event. This End Marker setting is also sent to the Mini-Flash when ever a sequence is downloaded.

### **The All-Important "Servo Control":**



The Mini-Flash programming software includes a "servo control" for each of the 4 channels that allow the user to use an R/C servo output (from an R/C receiver) as a "switch" control for each LED channel. This feature allows the user to control (turn On or Off) each of the 4 channels in response to the position of the servo control. If the

user uses a stick/slider as the servo control, the user will be able to define one of 16 control settings for each LED channel. These settings are based upon the servo position, that consisting of 4 unique "Zones". The zones are 0->1/4, 1/4->1/2, 1/2->3/4, and 3/4 ->Full (with full being the full deflection of the switch/slider). But how do you get to choose 1 of 16 positions? Well, you are also allowed to select one of the 16 possible "binary" combinations of these zones. For instance, you could program a LED channel to be active only when the stick is in the first zone (0->1/4) OR in the forth zone (3/4->Full). This binary interpretation would be "1001". All 16 combinations are available to the user. The servo pulse position determines the binary "mask" applied to each LED's sequence set by the user. A mask of "\*-\*-" (as selected in the programming software) would mean the sequence would only be active (turned on) when the servo is in zone 2 (1/4->1/2) or in zone 4 (full). As another last example, a mask of "\*\*\*-" would mean the LED sequence would be active for zones 2 thru 4, except for zone 1 (0->1/4), where the channel would be turned off. If the servo output used for the control is a toggle switch instead of a variable slider/joystick, you would then obviously have only two zones (on or off, or, 0->1/4 and 3/4->Full). The following chart indicates how the servo control setting works for each of the LED channels.

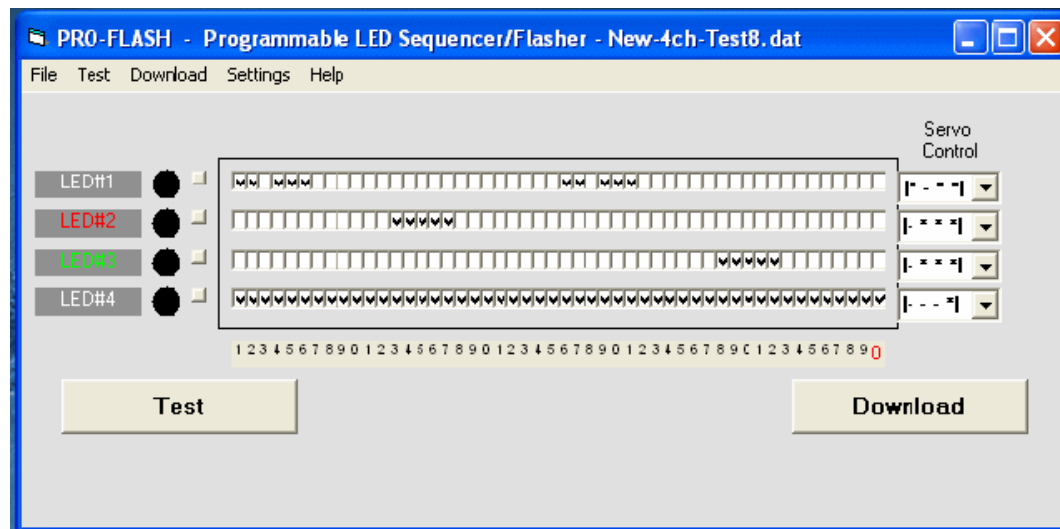


Servo Control Setting	Position of Transmitter Stick/Slider & Effect on LED Channel			
	Zone #1 (0 -> 1/4)	Zone #2 (1/4 -> 1/2)	Zone #3 (1/2 -> 3/4)	Zone #4 (3/4 -> Full)
---- (Off) (i.e. Always On)	off	off	off	off
---*	on	off	off	off
--*_	off	on	off	off
__**	on	on	off	off
_*__	off	off	on	off
_*_*	on	off	on	off
_**_	off	on	on	off
_***	on	on	on	off
*___	off	off	off	on
*__*	on	off	off	on
*_*_	off	on	off	on
*_**	on	on	off	on
**__	off	off	on	on
**_*	on	off	on	on
***_	off	on	on	on
**** (None) (i.e. Always On)	on	on	on	on

Included with the Mini-Flash software download are a few example files that show you some sample sequences/controls you can develop. Use your own imagination... the possibilities are almost endless. As for one example (see New-4ch-Test8.dat), this is how each of the channels are set up

Ch#1 = White LED, 1 double blink (on-off-on-off) every second (Tail Strobe)  
Ch#2 = Red LED, one longer blink (on-off) every 2 seconds; alternates with Ch#3  
Ch#3 = Green LED, one longer blink every 2 seconds; alternates with Ch#2 (Wingtip)  
Ch#4 = White LED, always ON (Landing light) - but controlled by servo control (see below)

This can be seen on the following screen shot:



As for servo controls, the following can also be programmed (assuming you are controlling Mini-Flash using a spare slider/stick channel, or even tapping off of the throttle servo wire):

When stick is 3/4 - to- Full deflection, Only CH#1 is active (i.e. flashing strobe on Tail)

When stick is 1/2 - to- 3/4 deflection, Only Ch#2 and Ch#3 are active (i.e. red & green wing tip light)

When stick is 1/1 - to- 1/2 deflection, Only Ch#1, Ch#2 and Ch#3 are active (i.e. Both strobe and wingtip lights)

When stick is 0 - to- 1/4 deflection, All channels (1-4) are active (i.e. Strobe, wing tip & Landing light)

The Servo Controls for each channel are set as follows:

Ch#1 = "\*-\*\*\*" (ON only when servo signal is in zones 1, 2 and 4)

Ch#2 = "-\*\*\*\*" (ON only when servo signal is in zones 1, 2 and 3)

Ch#3 = "-\*\*\*\*" (ON only when servo signal is in zones 1, 2 and 3)

Ch#4 = "---\*" (ON only when servo signal is in zone 1)

The 4 servo "zones" are as described:

Zone #1 = Stick in 0 to 1/4 deflection

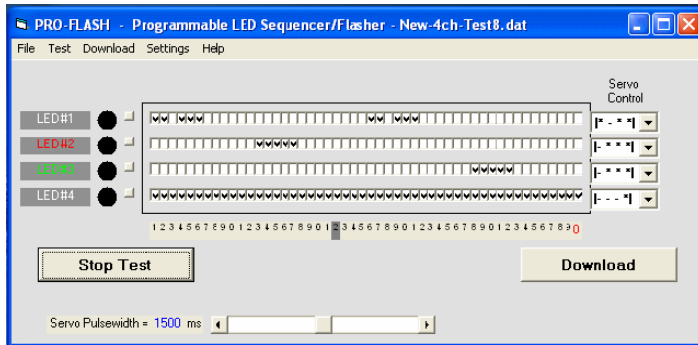
Zone #2 = Stick in 1/4 to 1/2 deflection

Zone #3 = Stick in 1/2 to 3/4 deflection

Zone #4 = Stick in 3/4 to full deflection

The more you experiment, the more versatile you will find the Mini-Flash programmer!

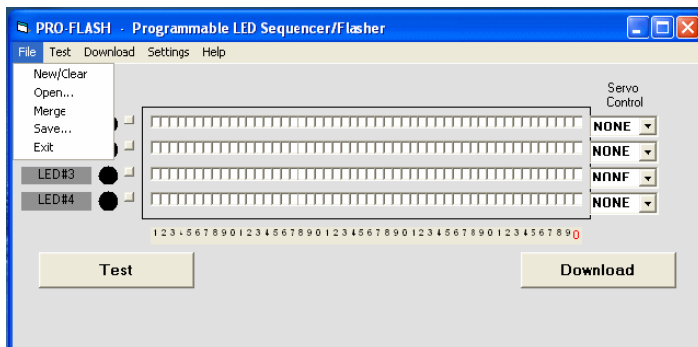
## Testing the programmed LED Sequence:



The programming software allows the user to "test" the current programmed sequence, including the servo control functions programmed by the user. To initiate the test, press the "Test" button in the programming software. Upon pressing this button, you will see the test sequencer display the programmed sequence on the

graphical LEDs in the program. If you have selected a servo control for any of the 4 channels, you will also see a scrollbar along the bottom of the page. This bar is used as if it was the transmitter's stick/switch, allowing you to test the servo control functions you have programmed for each LED channel. You can use the left/right arrow keys to alter the scrollbar setting or you can use your mouse to move the slider. As you change this slider, you will notice the servo pulse width value changing, just as if it was a servo signal coming from the receiver. As you change this slider, you should see its effect on the programmed LED channels. If no servo controls are programmed for any of the channels, this scrollbar type control will not be displayed, since it is not needed.

## New/Clear, Open, Merge, or Save Sequence File:



The programming software allows the user to develop a **New** LED sequence (clearing all current sequence events), **Open** an existing sequence file (previously saved sequence files like the sample files included), **Merge** an existing sequence file into the currently displayed sequence or **Save** the current sequence. These options are

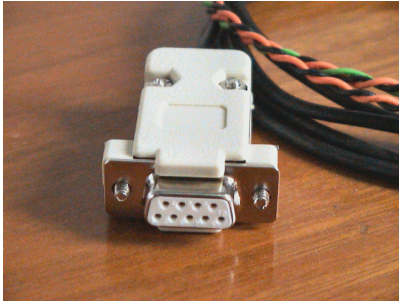
easily selected under the "File" menu in the programming software and are fairly intuitive (these are general dialog boxes most MS Window users are already familiar with).

## Programming the Mini-Flash ! :

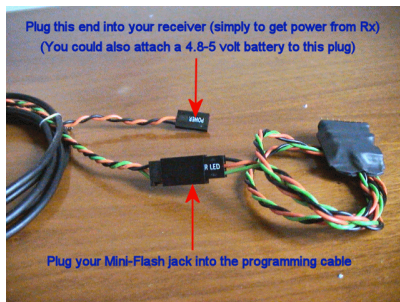
After you are satisfied with your programmed LED sequence, you can easily download it to your Mini-Flash, and this normally takes less than 10 seconds. It is very important though that you follow the instructions below very carefully. Please read carefully before attempting to download the sequence.

# STEP-BY-STEP PROGRAMMING INSTRUCTIONS

**Step 1:** Unplug the Mini-flash from its power source, that being a battery or a receiver that is powered up (assumes you have LEDs connected to controller)



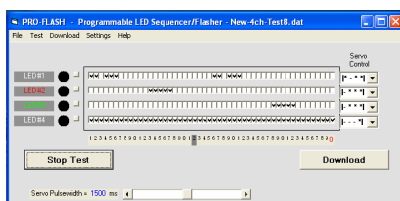
**Step 2:** Connect the programming cable's 9-pin connector (rs-232 serial cable) to an empty RS-232 serial communications port on your computer. Make note of the actual com port number, (usually 1 or 2, some PCs could have more) as you will need to define this port in the programming software later if you already haven't changed it.



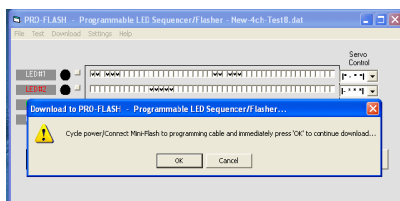
**Step 3:** Connect the POWER end of the programming cable (female plug) to your receiver and be sure the receiver is powered up (this connection to the Receiver simply provides power to the Mini-Flash during the downloading process; you could always use a 4.8v-5v battery instead of connecting it to your receiver)

**Note:** Always observe polarity when plugging the Mini-Flash into the receiver!

**Note:** DO NOT connect the Mini-Flash to the programming cable quite yet (you will do so in step 6 below)



**Step 4:** Now, run the software and configure the LED channels as you like them. You could also load a supplied example file for test purposes. If you haven't done so already, you could "virtually" test the sequence by hitting the "Test" button in the software (hit "Stop Test" button to stop testing)

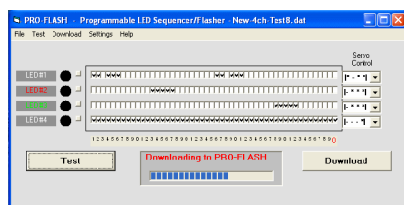


**Step 5:** When happy with the sequence and controls, hit the "Download" button and a message will pop up. This message tells you to cycle power to the Mini-Flash or connect the Mini-Flash to the programming cable.

**STOP! The next step (#6) is very important so please read thoroughly!**



**Step 6:** With this message still displayed, connect the Mini-Flash's servo connector to the male mating connector on the programming cable. As soon as you do this, you will see the all the LEDs on the Mini-Flash flash briefly. Eight (8) seconds later (gives time for Rx/PCM to configure if needed), all LEDs light up and all stay on for approximately 3 seconds. This 3 second period is when you need to then press the OK button to finally complete the download. If the LEDs go out (and start flashing) before you had time to hit the OK button, disconnect the Mini-Flash from the programming cable and repeat starting at the top of Step 6. It is really important that you hit the "OK" button in the download windows WHILE the LEDs are all lit. This is the only time the Mini-Flash can be programmed by the PC.



**Step 7:** You will notice that immediately after you hit the "OK" button, Mini-Flash's LEDs all go out (all LED extinguish). At the same time, you will see the downloading progress as indicated by the software's download progress bar along the bottom of the screen. This download process typically takes on the order of 7-10

seconds. When the download is completed, you will see the download bar disappear and then see the LEDs on the Mini-Flash begin flashing. They should now be flashing in accordance to how you programmed them, BUT... you will not be able to test any of the servo control functions until you connect the Mini-Flash to your Rx.

**Step 8:** Disconnect the Mini-Flash from the programming cable and disconnect the programming cable from your R/C receiver. Now, with your transmitter and receiver turned on (both powered), plug the Mini-Flash's servo connector directly to the R/C receiver channel which you want to use as a control. As soon as you plug it in, your Mini-Flash should turn on all briefly, eight (8) seconds later it should light all LEDs for 3 seconds, then it will be fully operational (LEDs flashing and servo controls functioning). You are now done! Simply repeat this process to reprogram.

**NOTE:** In order for the Mini-Flash to use these programmed setting and be fully operational each and every time you power it up, you will need to be sure that your R/C transmitter is powered ON before you apply power to the Mini-Flash (this is normal recommended operation for most R/C models, as it is safer on your model's servos and control surfaces). If the transmitter is not turned on while the Mini-Flash is powered, it will eventually go into a "free-running" mode where all sequences are active, regardless of the servo signal. Even if you then turn on your transmitter, it will stay in this free-running mode until power is cycled to the Mini-Flash.

**WARNING!:** DO NOT turn on your transmitter when the Mini-Flash has all its LED lit (i.e. during the 3 second programming period). Doing this will damage the previously programmed sequence and you will be required to reprogram the Mini-Flash via the com port.

# **Operating the Mini-Flash**

The following instructions describe how the user would normally operate the Mini-Flash once installed in your model. Be sure to thoroughly read these instructions before proceeding.

## **STEP-BY-STEP OPERATING INSTRUCTIONS**

1. With all devices off (no power to transmitter, receiver or Mini-Flash), power up your transmitter and be sure it is outputting RF (some Tx take a second or two).
2. With your receiver off, connect the Mini-Flash to its appropriate channel (normally, it is already connected in your model)
3. With your R/C Transmitter on, and the Mini-Flash plugged into your receiver, finally power on the receiver.
4. You should see all LEDs light and extinguish quickly. Eight (8) seconds later, all LEDs come on for 3 seconds then all go off. Only then is the Mini-Flash fully operational (LEDs flashing along with servo control functionality)
5. Power off sequence is not that important but normally (and for safety sake), it's always good to power off your receiver before powering off your transmitter.

## Other Notes

### DON'T FORGET THE ALL IMPORTANT SERIES RESISTOR INLINE WITH ALL LEDS !!!

The LEDs you will connect to the MINI-FLASH need to have a resistor in series with one of the LED's leads (I usually use the + lead). The resistor value depends on the LED's forward voltage and the LED's maximum current rating. These specs are usually supplied by the LED manufacturer. The basic formula for calculating this required series LED resistance is as follows:

$$\text{Required Resistance} = (\text{Voltage Source} - \text{LED Forward Voltage}) / \text{Max LED Current}$$

or

$$R = (V_s - V_f) / I_{max}$$

As an example, a particular LED has a 3.0 volt forward voltage ( $V_f$ ) and a maximum current rating of 20mA. It is to be driven with a 5.0 volt source. Therefore,

$$R = (5.0 - 3.0) / .020 \quad \text{or} \quad R = 2.0 / .020$$

or

$$\underline{R = 100 \text{ Ohms}}$$

Some LED Calculator Links:

<http://linear1.org/ckts/led.php> (simple LED resistor calculator)

<http://led.linear1.org/led.wiz> (another LED calculator for multiple LED in series or parallel)

<http://www.besthongkong.com/clrc.htm> (a neat calculator hosted by a big LED distributor in Hong Kong)

Some Interesting Visual Basic coding information (Picaxe to/from PC serial port) @ <http://www.rentron.com/VisualBasic.htm> (Reynolds Electronics)

# MINI-FLASH FAQ

## Q1. How do you compensate for differing forward voltages and current drawn by differing LED's?

**A1.** Ah.... good question....

The input voltage to LEDs is all not that important (usually 5v is fine for all). It is the current that you push through the LED that is important, as you do not want to drive them with too much current. You usually always need to put a series resistor inline with one of the LED leads such to limit the current. I use a neat calculator, I even have a link for it on my webpages. You really need to know the specs on the LEDs, particularly the LEDs rated current (typical 5mm LEDs run around 20-25 milliamps). Here's the calculator:

<http://linear1.org/ckts/led.php>

You simply enter the supply voltage (in the controller's case, 5 volts), the LEDs forward voltage (this varies from LED to LED) then enter the rated LED current (typically 20-25 milliamps). Then hit the "find R" button and the program calculates the resistor value you need for that LED (typically a 68 - 120 ohm resistor is required). Be careful also as there exists some LED that already have the series resistor incorporated in the LED (not all that common though). Once the resistor value is determined, I usually then solder it to the end of one of the LED leads. The wires then leading from this LED assembly is then connected directly to the controller using a miniature machine-pin female socket (I will provide at least 8 with every controller). Putting the series resistors on the PCB would take up space (unless they were SMD maybe).

## Q2. I have a question: So tell me how you could control different sequences with a TX like a 6102 which is a 6 channel but the 2 extras are more like landing gear, and flaps?

**A2.** Great question.... ( I knew I'd eventually have to write about this, as most will probably use a switch and some will have the luxury of having a spare variable channel, like a rotating knob or lever).

Is the landing gear channel on your system controlled by a switch ? (most likely). How about the flaps, are these a switch too or is this a rotating knob (like I think my Futaba 9C has) ??

If the channel you have the flasher plugged into is controlled (Tx) using a toggle



switch, you will still be able to control the 4 LED channels/sequences, but you will only have two possible servo control positions, on (\*---) and off (---\*). If you were to look at the "Servo Control" pull-down in the software, you have 16 different "servo zone" combinations you can chose. All 16 work only when you are using a variable Rx channel, such as a channel being controlled by a knob, slider or stick. If you are only using a "switched" Rx channel and you want to simply use the switch to turn the channel on or off, you would set up the "servo control" such that it is "\*---". What this would do is make the selected LED channel active only when the switch is in the up position (actually, it is as if the stick is in the position, 3/4->Full deflection). Now catch this, if you were to choose the "\*\*\*--" option or the "\*\*\*\*-" option, this would still have the same effect. When using a toggle switch on your Tx to control the LED flasher, the receiver will never output the middle two zones (-\*\*-). Therefore selecting these zones in a channel's servo zone will do nothing.

"PICAXE is a trademark of Revolution Education Ltd ([www.picaxe.co.uk](http://www.picaxe.co.uk))"

Mini-Flash Rev 2.0 User's Manual – Copyrighted by DIYRC.com (2007)  
(Reproduction by permission only)