# 1 Overview

This document provides specifications for a *firefly simulator*. This device is intended to simulate the light displays of common fireflies, such as *Photinus pyralis*. The *firefly simulator* will provide means to control the brightness, duration, and other relevant parameters of the simulated light display. Light-emitting diodes (LEDs) will be used as the light source, and the color of the display can be altered by connecting different LED types to the simulator.

The *firefly simulator* can be configured via commands issued to it from a host computer, as shown in Fig. 1. The physical link between the host computer and the *firefly simulator* is an asynchronous serial communications interface, which can be accomplished with a common and inexpensive USB adapter. The *firefly simulator* also can return status information to the host computer via the same interface.

Once configured, the *firefly simulator* can be used in a stand-alone mode, without requiring a connection to a host computer. Pushbuttons on the *firefly simulator* can be used to activate light displays that were previously configured. When in the stand-alone mode, the simulator can be powered by a USB powerbank.

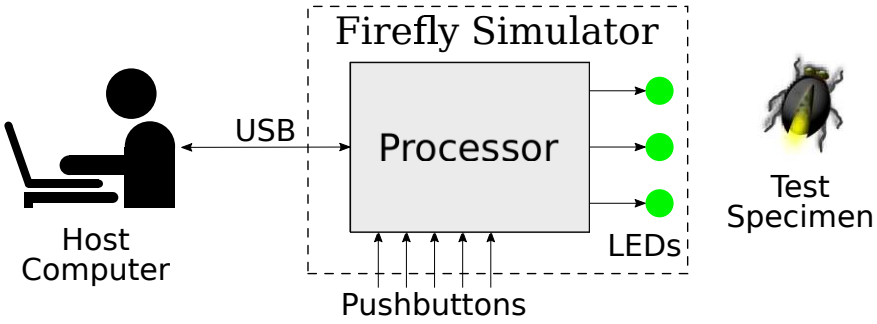


Figure 1: Firefly Simulator Block Diagram

This document describes the minimum required functionality of the *firefly simulator*. Possible future enhancements to the simulator include:

- The ability to sense the behavior of a test specimen and use that behavior to modify the parameters of a light display.
- The ability to record information about simulator activity on non-volatile memory, such as a removable Secure Digital (SD) memory card.

# 2 References

1. *IEEE Standard for Transitions, Pulses, and Related Waveforms*, IEEE Standard 181, 2011.

2. *Data elements and interchange formats  Information interchange  Representation of dates and times*, ISO 8601

# 3 Definitions

| | |
|---|---|
| abort | A pushbutton input to the *firefly simulator*. Pressing this button causes the simulator to stop any repeated *flashes* or *patterns*, and to bring the *illumination level* to 0% on all *channels*. |
| ASCII | The American Standard Code for Information Interchange. The letters in the English alphabet, the decimal digits, and common punctuation marks are assigned a unique 7-bit binary code. |

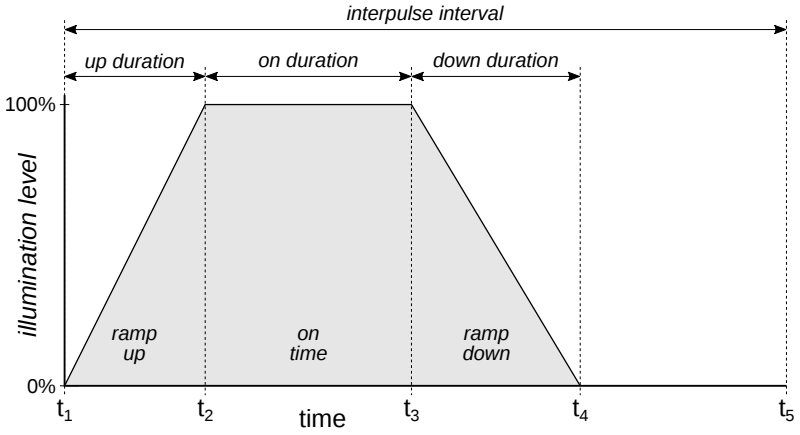


Figure 2: *flash* waveform

channel — The *channel* of an *LED* is an integer that specifies which physical output connector of the firefly simulator is connected to the physical LED. The value of *channel* is an integer and shall not be less than 1 and not greater than *max channel*.

Note that the *channel* is not the pin number of any particular microcontroller. An implementation of the firefly simulator must perform the appropriate mapping of an *LED*'s *channel* to an appropriate physical pin on the output device.

down duration — The length of the *ramp down* interval. The *down duration* is equal to $t_4 - t_3$, as shown in Fig. 2. The *down duration* shall be a non-negative integer value with units of milliseconds. The value of *down duration* shall not be less than 0 or greater than 32 767.

event — An external event that can be recognized by the *firefly simulator*. Details TBD.

flash — The process of bringing the *illumination level* of an *LED* from 0% to 100% then back to 0% illumination. A *flash* consists of a *ramp up*, followed by an *on time*, followed by a *ramp down*. A *flash* begins at $t_1$ and ends at $t_4$, as shown in Fig. 2. The interval from $t_1$ to $t_2$ is the *ramp up*. The interval from $t_2$ to $t_3$ is the *on time*. The interval from $t_3$ to $t_4$ is the *ramp down*. The interval from $t_4$ to $t_5$ is the *interpulse interval*.

Note the definition of a *flash* primarily specifies the *timing* behavior of the *flash*. The selection of a specific physical LED and its *max brightness* are part of the definition of an *LED*.

flash pattern interval — The total time duration of a *pattern*. This interval includes the time of all *flashes* in the pattern as well as the subsequent time when there are no flashes, as shown in Fig. 3. The *flash pattern interval* is a parameter of a *pattern*.

illumination level — The brightness of an *LED* at any given point in time, as a percentage of that *LED*'s *max brightness*. The *illumination level* and *max brightness* values indirectly translate to the average current passing through the physical LED.

At any given point in time, the average current for an LED is

$$I_{AVG} = \frac{illumination\,level}{100} \times \frac{max\,brightness}{100} \times I_{max}$$

| | |
|---|---|
| interpulse interval | The total time duration of a *flash*. This interval includes the time when the *LED* is illuminated as well as the subsequent time when the *LED* is not illuminated, from $t_1$ to $t_5$ in Fig. 2. The *interpulse interval* is a parameter of a *flash*. |
| message | A sequence of *message fields*, separated by the ASCII comma character (decimal 44) and terminated by either the ASCII Line Feed (decimal 10), the ASCII Carriage Return (decimal 13), or both the Line Feed and the Carriage Return. There shall not be a comma before the first field in a message nor after the last field. |
| message field | A sequence of one or more ASCII characters from the set of letters (`A` through `Z` and `a` through `z`), decimal digits (`0` through `9`), and the punctuation characters required for a *timestamp* (colon, minus, plus, period). A message field shall not include a comma. |
| max brightness | The maximum duty factor of the *pulse-width modulation* signal that controls the *illumination level* of an *LED*. The *max brightness* is a characteristic of an *LED*. The value of *max brightness* is an integer and shall be not less than 1 and not greater than 100. |
| max channel | The number of physical *LED channels* available on a particular implementation of a *firefly simulator*. The value of *max channel* shall not be less than 1 or greater than 127 for any implementation of a *firefly simulator*. |
| max event | The number of unique *event* configurations available on a particular implementation of a *firefly simulator*. The value of *max event* shall not be less than 0 or greater than 127 for any implementation of a *firefly simulator*. |
| max flash | The number of unique *flash* configurations available on a particular implementation of a *firefly simulator*. The value of *max flash* shall not be less than 1 or greater than 127 for any implementation of a *firefly simulator*. |
| max LED | The number of unique *LED* configurations available on a particular implementation of a *firefly simulator*. The value of *max LED* shall not be less than 1 or greater than 127 for any implementation of a *firefly simulator*. |
| max pattern | The number of unique *pattern* configurations available on a particular implementation of a *firefly simulator*. The value of *max pattern* shall not be less than 1 or greater than 127 for any implementation of a *firefly simulator*. |
| on duration | The length of the *on time* interval. The *on duration* is equal to $t_3 - t_2$, as shown in Fig. 2. The value of *on duration* shall be a positive, non-zero integer with units of milliseconds. The value of *on duration* shall not be less than 1 or greater than 32 767. |
| on time | The period of time during a *flash* when the *LED* is constantly at an illumination level of 100%. |
| pattern | A sequence of up to 16 *flashes*, possibly followed by a period of time where there are no *flashes*. Note that the definition of a *pattern* specifies only the sequence of *flashes* that should occur as well as the total duration of the *pattern*. |
| ramp down | The process of linearly decreasing the *illumination level* of an *LED* from 100% to 0%. |
| ramp up | The process of linearly increasing the *illumination level* of an *LED* from 0% to 100%. |
| time stamp | A representation of the current date and time used as a *message field*. The *time stamp* shall conform to ISO 8601. The format of the *time stamp* is `YYYY-MM-DDTHH:MM:SSZ`. The '`-`' character delimits the year, month, and day fields; the '`T`' delimits the date from the time; the '`:`' character delimits the hours, minutes, and seconds fields. All time values are reported using 24-hour UTC (Zulu) time rather than the local timezone. |

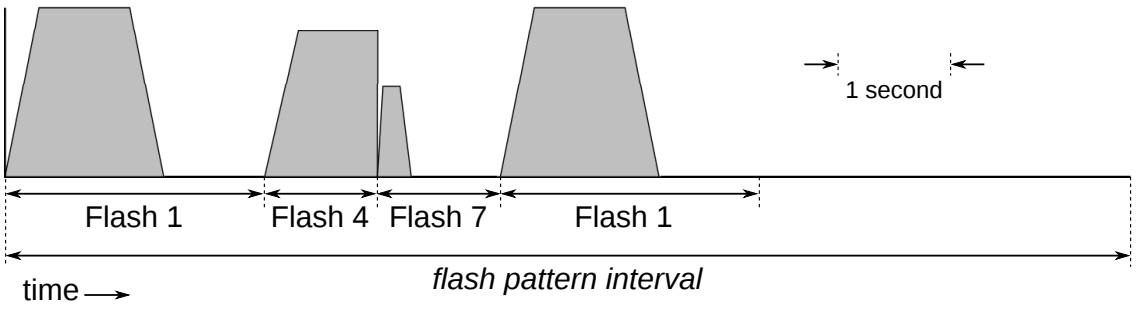


Figure 3: Example *pattern* timeline

up duration — The length of the *ramp up* interval. The *up duration* is equal to $t_2 - t_1$, as shown in Fig. 2. The value of *up duration* shall be a non-negative integer with units of milliseconds. The value of *up duration* shall not be less than 0 or greater than 32 767.

# 4 Resolution and Accuracy

## 4.1 Brightness

The *firefly simulator* does not directly control the brightness of a physical LED. Instead, the simulator controls the average current provided to the LED. The simulator's configuration messages allow the actual average current ($I_{AVG}$) of an LED to be specified with a resolution of $\pm 1\%$ of the maximum available current ($I_{MAX}$). The maximum available current, and the precision which with the current can be specified, will be determined by the circuitry associated with a given physical LED and need not be the same for all LEDs.

## 4.2 Time

Values that represent time shall have units of milliseconds and a resolution of 1 ms. The accuracy of all pulse durations and time delays generated by the *firefly simulator* over an *interpulse interval* shall have a maximum error of $\pm 10$ ms. The cumulative timing error over a *flash pattern interval* shall not exceed 200 ms.

# 5 Configuration Messages

The firefly simulator is configured via a serial communications interface to a host computer. The host computer can set the values of all parameters for *LEDs*, *flashes*, and *patterns*. A unique configuration message format is specified for configuring an *LED*, configuring a *flash*, or configuring a *pattern*.

## 5.1 *LED Configuration Message*

An *LED Configuration Message* is sent from the host computer to the firefly simulator. Every *LED Configuration Message* shall contain four *message fields*, as shown in Table 1.

As an example, the three messages below could be sent by the host computer in order to configure three *LEDs*. This example assumes that *LED* 2 uses physical *channel* 1 and has a *max brightness* of 100%. *LEDs* 3 and 5 use the same physical *channel* (i.e. the same physical LED) but with different levels of *max brightness*: 87% and 53%, respectively. The *LED* numbers, *channel* numbers, and *max brightness* levels shown here were chosen arbitrarily; the purpose of this example is only to illustrate the syntax of the *LED Configuration Message*.

```
L,2,1,100
L,3,6,87
```

```
L,5,6,53
```

Table 1: Definition of the *LED Configuration Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first character for an *LED Configuration Message* | This field shall be the uppercase letter '`L`'. |
| 2 | *LED* number | A unique identifier for each *LED* definition | This field shall contain a decimal integer from 1 to *max LED*. |
| 3 | LED *channel* | The physical channel associated with this *LED* | This field shall contain a decimal integer from 1 to *max channel*. |
| 4 | *max brightness* | The maximum brightness level for the *LED* | This field shall contain a decimal integer from 1 to 100. |

## 5.2 *Flash Configuration Message*

A *Flash Configuration Message* is sent from the host computer to the firefly simulator. This message provides the parameters for a single *flash* of an *LED*, as shown in Table 2.

As an example, the three messages below could be sent by the host computer in order to configure the three *flashes* shown in Fig. 3. This example assumes that *flash* 1 uses *LED* 2, *flash* 4 uses *LED* 3, and *flash* 7 uses *LED* 5. The *flash* and *LED* numbers were chosen arbitrarily; the purpose of this example is only to illustrate the syntax of the *Flash Configuration Message*.

```
F,1,2,300,800,300,2300
F,4,3,300,700,0,1000
F,7,5,50,150,100,1100
```

Table 2: Definition of the *Flash Configuration Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first character for a *Flash Configuration Message* | This field shall be the uppercase letter '`F`'. |
| 2 | *flash* number | A unique identifier for each *flash* definition | This field shall contain a decimal integer from 1 to *max flash*. |
| 3 | LED number | Identifier for the *LED* to be illuminated in this *flash* | This field shall contain a decimal integer from 1 to *max channel*. |
| 4 | *up duration* | The duration of the *ramp up* in milliseconds | This field shall contain a decimal integer from 0 to 32 767. |
| 5 | *on duration* | The duration of the *on time* in milliseconds | This field shall contain a decimal integer from 1 to 32 767. |
| 6 | *down duration* | The duration of the *ramp down* in milliseconds | This field shall contain a decimal integer from 0 to 32 767. |
| 7 | *interpulse interval* | The duration of the entire *flash* in milliseconds | This field shall contain a decimal integer from 0 to 32 767. |

### 5.3 *Pattern Configuration Message*

A *Pattern Configuration Message* is sent from the host computer to the firefly simulator. This message provides the parameters for a single *pattern* of one or more *flashes*, as shown in Table 3. Note that the same *flash* may be repeated within a pattern.

A *pattern* may contain from 1 to 16 *flashes*. Therefore, field 4 in the *Pattern Configuration Message* may be repeated to define the desired *flashes* in the pattern. If the *Pattern Configuration Message* specifies fewer than 16 *flashes* then the unspecified *flashes* shall have a default *flash* number of 0 (zero) and will be ignored during execution of the *pattern*.

As an example, the message below could be sent by the host computer in order to configure the *pattern* shown in Fig. 3. This particular *pattern* was arbitrarily designated as pattern 5. It contains four *flashes*, but one of the previously defined *flashes* was repeated. The total duration of this pattern (i.e. the *flash pattern interval*) is 10 s.

```
P,5,10000,1,4,7,1
```

Table 3: Definition of the *Pattern Configuration Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first character for a *Pattern Configuration Message* | This field shall be the uppercase letter '`P`'. |
| 2 | *pattern* number | A unique identifier for each *pattern* definition | This field shall be a decimal integer from 1 to *max pattern*. |
| 3 | *flash pattern interval* number | The total duration of the *pattern* in milliseconds | This field shall contain a decimal integer from 0 to 32 767. |
| 4 | *flash* list | The number of a *flash* to be included in the *pattern*. | This field shall be a decimal integer from 1 to *max flash*. |

### 5.4 *Random Pattern Set Configuration Message*

A *Random Pattern Set Configuration Message* is sent from the host computer to the firefly simulator. This message provides the parameters for a list of *patterns* that will be executed in a pseudorandom order **??**.

A set (or list) of random *patterns* may contain from 1 to 16 patterns. Therefore, field 3 in the *Random Pattern Set Configuration Message* may be repeated to define the desired patterns. If the *Random Pattern Set Configuration Message* specifies fewer than 16 *patterns* then the unspecified *patterns* shall have a default *pattern* number of 0 (zero) and will be ignored.

Table 4: Definition of the *Random Pattern Set Configuration Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first character for a *Random Pattern Set Configuration Message* | This field shall be the uppercase letter '`R`'. |
| 2 | *Random Pattern Set* number | A unique identifier for each random pattern set definition | This field shall be a decimal integer from 1 to *max pattern set*. |
| 3 | *pattern* list | The number of a *pattern* to be included in the set. | This field shall be a decimal integer from 1 to *max pattern*. |

### 5.5 *Time Configuration Message*

A *Time Configuration Message* is sent from the host computer to the firefly simulator. This message sets the current date and time for the real-time clock (RTC) in the firefly simulator, as shown in Table 5. The time information **must** be provided in a 24-hour format, where each day begins at 00:00:00 and ends at 23:59:59, and as Coordinated Universal Time (UTC) rather than in the local time zone.

The firefly simulator **does not** verify the validity of the date and time information. Configuring the RTC with invalid values may cause unpredictable behavior.

Table 5: Definition of the *Time Configuration Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first character for a *Time Configuration Message* | This field shall be the uppercase letter '`T`'. |
| 2 | Year | The current year | This field shall be a four-digit decimal integer from 2000 to 2099. |
| 3 | Month | The current month | This field shall be a decimal integer from 1 to 12. |
| 4 | Date | The current day of the month | This field shall be a decimal integer from 1 to 31. |
| 5 | Hour | The current hour, in 24-hour format | This field shall be a decimal integer from 0 to 23. |
| 6 | Minute | The current minutes | This field shall be a decimal integer from 0 to 59. |
| 7 | Seconds | The current seconds | This field shall be a decimal integer from 0 to 59. |

## 6 Command and Response Messages

The host computer can command the *firefly simulator* to turn on an *LED* at a specified *illumination level*, repeatedly execute a specific *flash*, repeatedly execute a specific *pattern*, or repeatedly execute a set of available *patterns* in a pseudorandom order.

The *Execute LED Message* may be sent repeatedly to activate one or more LEDs. However, once the *firefly simulator* has received and started executing an *Execute Flash Message*, *Execute Pattern Message*, or *Execute Random Pattern Message* it will not respond to any messages sent by the host computer, nor will it respond to pressing keys on the keypad. Each of these commands must be terminated by pressing the *abort* button before the *firefly simulator* will respond to messages from the host computer or keypad.

### 6.1 *Execute LED Message*

An *Execute LED Message* is sent from the host computer to the firefly simulator. This message can be used to turn an LED on for testing or calibration purposes. Every *Execute LED Message* shall contain three *message fields*, as shown in Table 6.

Table 6: Definition of the *Execute LED Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first two characters for an *Execute LED Message* | This field shall be the uppercase letters '`XL`'. |
| 2 | LED *channel* | A unique identifier for a physical LED | This field shall contain a decimal integer from 1 to *max channel*. |
| 3 | *illumination level* | Sets a constant value for the *illumination level* of the *LED* | This field shall contain a decimal integer from 0 to 100. A value of 100 causes the *illumination level* of the LED to be set at a constant value of 100% (of the *LED*'s *max brightness* level). A '0' shall cause the *illumination level* of the LED to be set at 0% (completely dark). |

## 6.2 *Execute Flash Message*

An *Execute Flash Message* is sent from the host computer to the firefly simulator. This message can be used to cause the firefly simulator to repeatedly execute a specific *flash*. Every *Execute Flash Message* shall contain two *message fields*, as shown in Table 7.

Table 7: Definition of the *Execute Flash Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first two characters for an *Execute flash Message* | This field shall be the uppercase letters '`XF`'. |
| 2 | *flash* number | A unique identifier for a *flash* | This field shall contain a decimal integer from 1 to *max flash*. |

## 6.3 *Execute Pattern Message*

An *Execute Pattern Message* is sent from the host computer to the firefly simulator. This message can be used to cause the firefly simulator to repeatedly execute a specific *pattern*. Every *Execute pattern message* shall contain two *message fields*, as shown in Table 8.

Table 8: Definition of the *Execute Pattern Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first two characters for an *Execute pattern Message* | This field shall be the uppercase letters '`XP`'. |
| 2 | *pattern* number | A unique identifier for a *pattern* | This field shall contain a decimal integer from 1 to *max pattern*. |

### 6.4 *Execute Random Pattern Message*

An *Execute Random Pattern Message* is sent from the host computer to the *firefly simulator*. This message can be used to cause the simulator to pseudo-randomly select and execute patterns from the set of patterns listed in an earlier *Random Pattern Set Configuration Message*.

The *firefly simulator* shall send a *Pattern Start Message* to the host computer before beginning each pattern.

Table 9: Definition of the *Execute Random Pattern Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first two characters for an *Execute Random Pattern Message* | This field shall be the uppercase letters '`XR`'. |
| 2 | *Random Pattern Set* number | A unique identifier for a *Random Pattern Set* | This field shall contain a decimal integer from 1 to *max pattern set*. |

## 7 Response Messages

In some circumstances the firefly simulator will send a message to the host computer without first receiving a message from the host. An *Event Response Message* will be sent to the host if an external event occurs. If the firefly simulator is repeatedly executing *patterns* then a *Pattern Start Message* will be sent before each *pattern* begins execution.

By default, the response messages sent from the firefly simulator to the host computer are designed to simplify any parsing and processing that will be done on the host computer. These messages are intentionally terse. However, the firefly simulator software has a compile-time option that causes human-readable messages to be sent instead. The human-readable messages include labels for the message fields, which can make them easier to use for debugging. The formats of these human-readable messages are not specified here.

### 7.1 *Event Response Message*

The *firefly simulator* shall respond to an *event* by sending an *Event Response Message* to the host computer, as shown in Table 10.

Table 10: Definition of the *Event Response Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | message type | Unique identifier for this message type | This field shall contain the lowercase letter '`e`'. |
| 2 | *time stamp* | The current data and time | This field is a *time stamp* |
| 3 | temperature | The current ambient temperature, in degrees Celsius. | This field shall contain a decimal integer from 0 to 127. |
| 4 | *pattern* | The *event number* of the *event* that occurred. | This field shall contain a decimal integer from 1 to 127. |

### 7.2 *Pattern Start Message*

The *firefly simulator* will send a *Pattern Start Message* to the host computer, as shown in Table 11, before the execution of any *pattern* begins. If a single *pattern* is executed repeatedly then the *Pattern Start Message* will also

be sent repeatedly.

Table 11: Definition of the *Pattern Start Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | message type | Unique identifier for this message type | This field shall contain the lowercase letter 'p'. |
| 2 | *time stamp* | The current data and time | This field is a *time stamp* |
| 3 | temperature | The current ambient temperature, in degrees Celsius. | This field shall contain a decimal integer from 0 to 127. |
| 4 | *pattern* | The pattern number of the *pattern* that will be executed. | This field shall contain a decimal integer from 1 to 127. |

# 8 Information Query Messages

The firefly simulator can be implemented on several different hardware platforms, and each platform will have inherent limits in its capabilities. The host computer can send a *Capacity Query* message to the firefly simulator in order to determine how many physical *LED* channels are available, as well as the platform's capacity to store various types of configuration messages.

The host computer may also send messages to the firefly simulator in order to determine current configuration of all *LEDs*, *flashes*, *patterns*, and *random pattern sets* that are known to the simulator. These "dump" messages do not require that the *abort* button be pressed before another command is issued.

By default, the response messages sent from the firefly simulator to the host computer are designed to simplify any parsing and processing that will be done on the host computer. These messages are intentionally terse. However, the firefly simulator software has a compile-time option that causes human-readable messages to be sent instead. The human-readable messages include labels for the message fields, which can make them easier to use for debugging. The formats of these human-readable messages are not specified here.

## 8.1 *Capacity Query/Response Messages*

The *Capacity Query Message* can be used by the host computer to determine the capabilities of a *firefly simulator*.

Table 12: Definition of the *Capacity Query Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first character for a *Capacity Query Message* | This field shall be the uppercase letter 'C'. |

The *firefly simulator* will respond to the *Capacity Query Message* by sending a *Capacity Response Message* to the host computer. Note that the values returned indicate the overall capacity of the firefly simulator hardware, but they do not reflect what portion of that capacity may already be used.

Table 13: Definition of the *Capacity Response Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | message type | Unique identifier for this message type | This field shall contain the lowercase letter '`c`'. |
| 2 | *time stamp* | The current data and time | This field is a *time stamp* |
| 3 | temperature | The current ambient temperature, in degrees Celsius. | This field shall contain a decimal integer from 0 to 127. |
| 4 | *max channel* | The number of physical LED channels available | This field shall contain a decimal integer from 1 to 127. |
| 5 | *max LED* | The number of *LED* definitions that may be stored | This field shall contain a decimal integer from 1 to 127. |
| 6 | *max flash* | The number of *flash* definitions that may be stored | This field shall contain a decimal integer from 1 to 127. |
| 7 | *max event* | The number of *event* definitions that may be stored | This field shall contain a decimal integer from 1 to 127. |
| 8 | *max pattern* | The number of *pattern* definitions that may be stored | This field shall contain a decimal integer from 1 to 127. |
| 9 | *max pattern set* | The number of *random pattern set* definitions that may be stored | This field shall contain a decimal integer from 1 to 127. |

## 8.2 *Dump LEDs Message/Response*

A *Dump LEDs Message* is sent from the host computer to the firefly simulator. The firefly simulator responds by sending information about all defined *LEDs* back to the host computer.

The *Dump LEDs Message* shall contain one *message field*, as shown in Table 14.

Table 14: Definition of the *Dump LEDs Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first characters for a *Dump LEDs Message* | This field shall be the two uppercase letters '`DL`'. |

The firefly simulator responds to the *Dump LEDs Message* by sending one text line for each *LED* that is currently configured. The default format of each line in the *Dump LEDs Response* is identical to the format for the *LED Configuration Message*, as shown in Table 1, except that the first field is the lowercase letter '`l`' instead of an uppercase character.

## 8.3 *Dump Flashes Message/Response*

A *Dump Flashes Message* is sent from the host computer to the firefly simulator. The firefly simulator responds by sending information about all defined *flashes* back to the host computer.

The *Dump Flashes Message* shall contain one *message field*, as shown in Table 15.

Table 15: Definition of the *Dump Flashes Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first characters for a *Dump Flashes Message* | This field shall be the two uppercase letters '`DF`'. |

The firefly simulator responds to the *Dump Flashes Message* by sending one text line for each *flash* that is currently configured. The default format each line of the *Dump Flashes Response* is identical to the format for the *Flash Configuration Message*, as shown in Table 2, except that the first field is the lowercase letter '`f`' instead of an uppercase character.

## 8.4 *Dump Patterns Message/Response*

A *Dump Patterns Message* is sent from the host computer to the firefly simulator. The firefly simulator responds by sending information about all defined *patterns* back to the host computer.

The *Dump Patterns Message* shall contain one *message field*, as shown in Table 16.

Table 16: Definition of the *Dump Patterns Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first characters for a *Dump Patterns Message* | This field shall be the two uppercase letters '`DP`'. |

The firefly simulator responds to the *Dump Patterns Message* by sending one text line for each *pattern* that is currently configured. The default format of each line of the *Dump Patterns Response* is identical to the format for the *Pattern Configuration Message*, as shown in Table 3, except that the first field is the lowercase letter '`p`' instead of an uppercase character.

## 8.5 *Dump Random Pattern Sets Message/Response*

A *Dump Random Pattern Sets Message* is sent from the host computer to the firefly simulator. The firefly simulator responds by sending information about all defined *random pattern sets* back to the host computer.

The *Dump Random Pattern Sets Message* shall contain one *message field*, as shown in Table 17.

Table 17: Definition of the *Dump Random Pattern Sets Message*

| Field Number | Field Name | Description | Format |
|---|---|---|---|
| 1 | Header | Unique first characters for a *Dump Patterns Message* | This field shall be the two uppercase letters '`DR`'. |

The firefly simulator responds to the *Dump Random Pattern Sets Message* by sending one text line for each *random pattern set* that is currently configured. The default format of each line of the *Dump Random Pattern Sets Response* is identical to the format for the *Random Pattern Set Configuration Message*, as shown in Table 4, except that the first field is the lowercase letter '`r`' instead of an uppercase character.

# 9 Keypad Interface

The firefly simulator can also be controlled by using the attached 12-key keypad. The keypad has keys for each of the digits from '0' to '9' and the characters '*' and '#'.

The keypad can be used to cause the firefly simulator to execute a single *pattern* repeatedly. The user must first press the '*' key and then a single digit key corresponding to the desired *pattern* number. Only *patterns* 1 through 9 may be executed in this way. Note that the firefly simulator will respond by sending *Pattern Start Messages*, just as if the host computer had sent an *Execute Pattern Message*.

The keypad can be used to cause the firefly simulator to execute a single *random pattern set* repeatedly. The user must first press the '#' key and then a single digit key corresponding to the desired *random pattern set* number. Only *random pattern sets* 1 through 9 may be executed in this way. Note that the firefly simulator will respond by sending *Pattern Start Messages*, just as if the host computer had sent an *Execute Random Pattern Set Message*.

# 10 Revision History

- From version 2.1 to version 2.2:
  - Added *Time Configuration Message*
  - Added *max pattern set* to *Capacity Response Message*
  - Changed *Execute Random Pattern* message
  - Added *Dump Random Pattern Sets* command and response
- From version 2.0 to version 2.1:
  - Real-time clock functionality added
  - Added *Dump LEDs*, *Dump Flashes*, and *Dump Patterns* command/response messages.
  - Added definition of a *time stamp*
  - Added keypad interface
- From version 1.0 to version 2.0:
  - Added ISO 8601 to references
  - Added definitions of terms *abort*, *flash pattern interval*, *interpulse interval*, *max event*, *max LED*, *time stamp*
  - The term *blink* is replaced with the term *flash*.
  - Added *time stamp* and *temperature* fields to the *capacity response message*
  - Added examples of typical messages
  - Definition and configuration message for *pattern* was significantly changed.
  - Removed the repeat count field from all *execute* commands; these commands must now be terminated by the *abort*
  - Updated diagram of *flash* timing to include *interpulse interval*.
  - Updated diagram of *pattern* timing to include *flash pattern interval*.
  - Added definition of *start pattern message*.