Spam Protocol

[Version 1.3]
May 19, 2015

[Alhad Palkar]
System Software Engineering, Firmware
[apalkar@apple.com]

Contents

List of Figures

List of Tables

1 Introduction

This is a specification for the Spam protocol. Its a point-to-point packet protocol, which will be used by the device to communicate the status of on-going tests to a host. The host would in-turn use this to visually indicate the test status to the operator. It uses the EzLink protocol for communication. You can think of the Spam protocol as an application layer specification .

2 Packet

Every packet consists of 3 parts; the EzLink header, followed by the application header, and the actual payload.

EzLink Header	Spam Header	Payload...

For more information on EzLink see the following:

<rdar://problem/16541301>'EzLink' host-library

<rdar://problem/16541321>'EzLink' diags-side library

3 Goals

The primary goal of this protocol is to send status updates from the device to the host. The device in this case is the unit that is undergoing testing, traditionally known as DUT(Device Under Test). The host could be a computer, or even a micro-controller on the cable that is snooping data being sent from the DUT.

3.1 Device Goals

1. Signal start of communication using the EzLinkSetup API.
2. Send a configuration packet that establishes the heartbeat interval 'h' seconds.
3. Send a status packet every 'h' seconds.

3.2 Host Goals

1. Parse the status packet and give an indication to the operator about the device state.
2. In case of an unresponsive device, give an indication to the operator. A device is deemed unresponsive(hung/panicked etc.) if the host has not received any packet for more than '$2 \times h$' seconds.
3. If a packet is not supported ignore it without giving erring out. Even if the packet is unsupported this still counts towards refreshing the heartbeat.

4 Communication

The communication is always initiated by the device; all that the host does is send Acks back to the device. The sequence diagram is shown below.

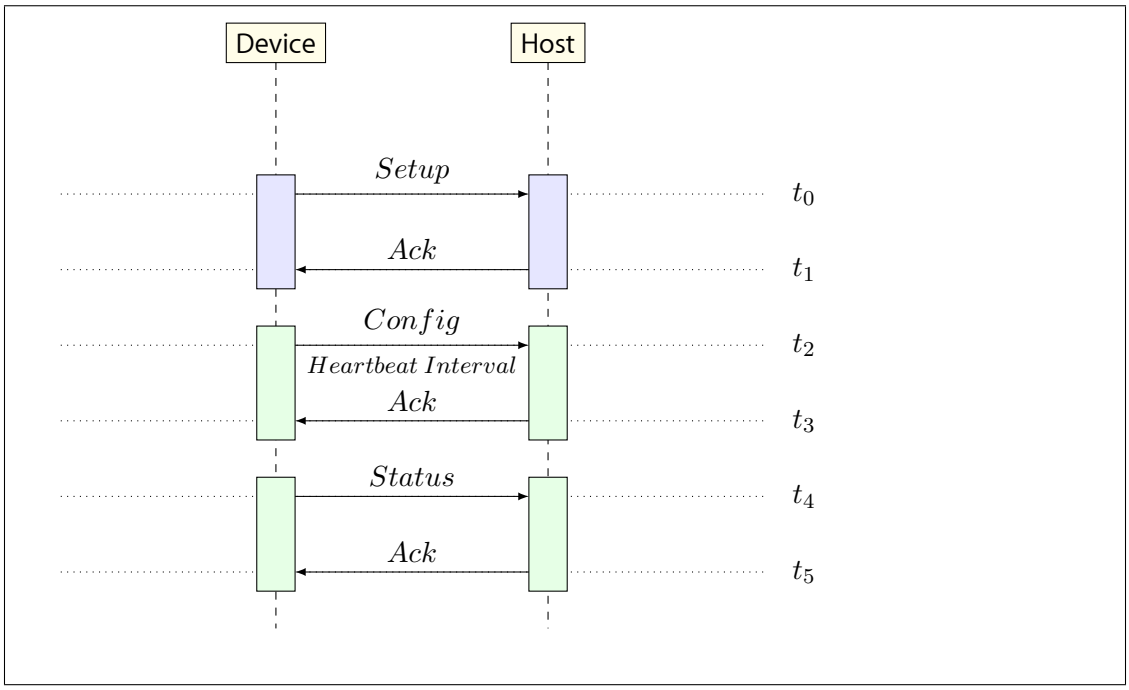

Figure 1: *SPAM Communication Sequence*

4.1 Setup

This is used by the device to discover the host and make sure its responsive. If this fails, the device will not initiate the Spam protocol

4.2 Configure

In this phase the device will setup the heartbeat interval. In between the setup and configure phases the host can assume a default heartbeat timeout that is sufficiently long (say 2min). Once the configure packet is received the host will update it's heartbeat timeout. The device can send configure packets at any point. For example if the host knows that it is going to be doing something during which time it might not be able to send packets to the host (one of the scenarios is when the device boots from EFI to OS), it can send a configure packet with a sufficiently long heartbeat interval.

4.3 Status Reporting

This is used by the device to report it's status to the host. To make implementation simple, since we know that the host in our case is going to control 3 LEDs we can have the packet contents directly control the LEDs. The plan is to add additional packet types later on.

5 Spam Packet Structure

5.1 Header

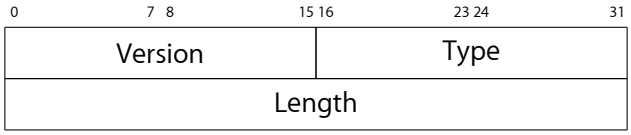

Figure 2: *Spam Header*

5.1.1. Version

The first 16 bits of the packet are the Spam protocol version number. This document describes version 1 of the Spam Protocol and thus the version field has a value of 0x01.

5.1.2. Type

This field is used to identify the types of packets. The possible values for this are:

Type	Description
0x0	Configure Packet
0x1	LED Control Packet
0x2	Heartbeat Packet

5.1.3. Length

This indicates the size of the payload, not including the size of the Spam Header itself.

5.2 Configure Packet

The configure packet payload will be nothing but a 32bit value that indicates the heartbeat interval in seconds

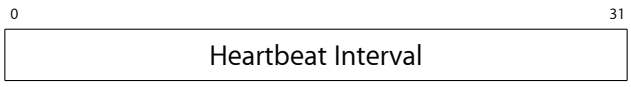

Figure 3: *Configure*

5.3 LED Control Packet

This will be used by the device to control the LEDs on the host side.

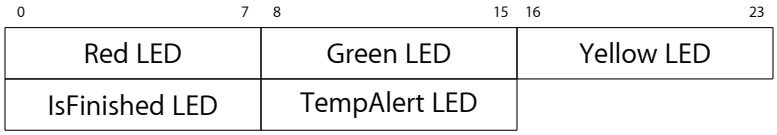

Figure 4: *LED Control*

Each LED can be controlled by it's corresponding 8bit value.

Value	Description
0x0	Off
0x1	On
0x2	Slow Pulse(every 3s)
0x3	Fast Pulse(every 0.5s)

5.4 Heartbeat Packet

This packet type has no payload. The 'Length' field for this packet should be set to 0. The Device is responsible for sending a heartbeat packet at the frequency dictated by the heartbeat interval.

6 LED Behavior

The Device is essentially in control of what packets it sends over to the Host, and hence controls which LEDs are lit up (there are some exceptions to this rule that will be highlighted below) The various LED states are as follows

1. **Plugged In**: The device has not yet sent the Setup packet. To indicate this the Host should set the Yellow LED to 'Slow Pulse' and all other LEDs to 'Off'.
2. **Ongoing**: Indicates that a test is in progress. We should set the Yellow LED to 'On', and all other LEDs to 'Off'.
3. **Pass**: Indicates that test has successfully finished. We should set Green LED and IsFinished LED to 'On'; all other LEDs should be 'Off'.
4. **Fail**: Indicates that test has failed. We should set Red LED and IsFinished to 'On'; all other LEDs should be 'Off'.
5. **Unresponsive**: In this scenario, the Host's Heartbeat / Setup Timer has expired and it should set Red LED to 'Fast Pulse' and IsFinished LED to 'On'.
6. **Temperature Alert**: The 'TempAlert' LED status indicates whether the temperature of the Device has gone above a maximum threshhold.

7 Host State Machine

This describes the host side state machine that drives the LEDs

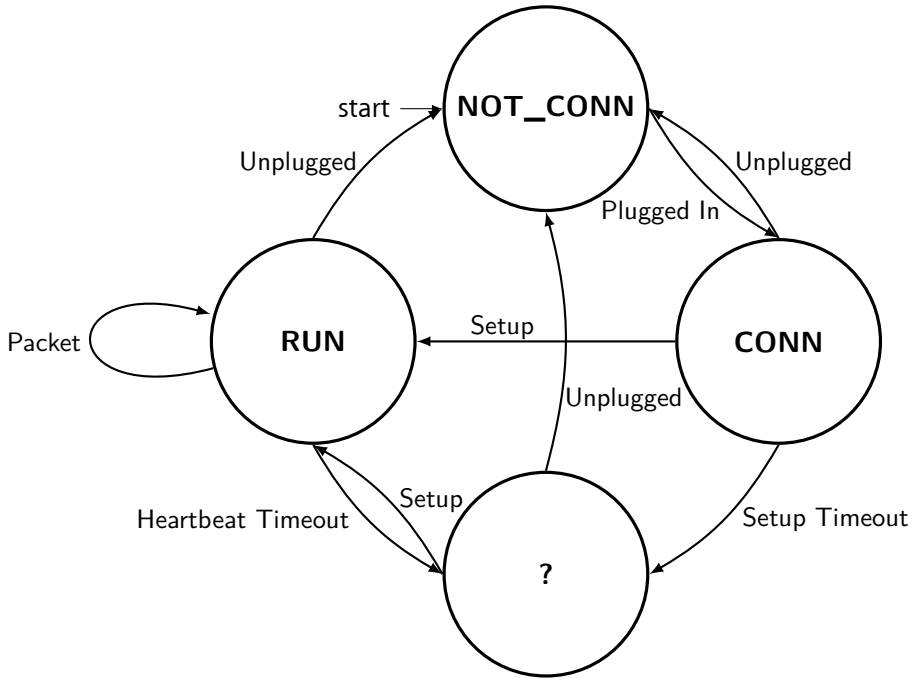

Figure 5: *Host State Machine*

1. '?' represents the Device that has paniced or has become unresponsive.
2. The 'Heartbeat timeout' is dictated by the Config packet sent by the Device.
3. The 'Setup Timeout' is useful to catch issues where the Device was connected to the Host but for some reason did not issue a Setup packet. This timeout should be 5min, which gives the Device more than enough time to boot into the OS and send a Setup packet.
4. If the Host receives a Setup packet in the '?' state, it should go to the 'RUN' state and set LEDs to represent *'Ongoing'*.

8 Device State Machine

This defines the Device side behavior.

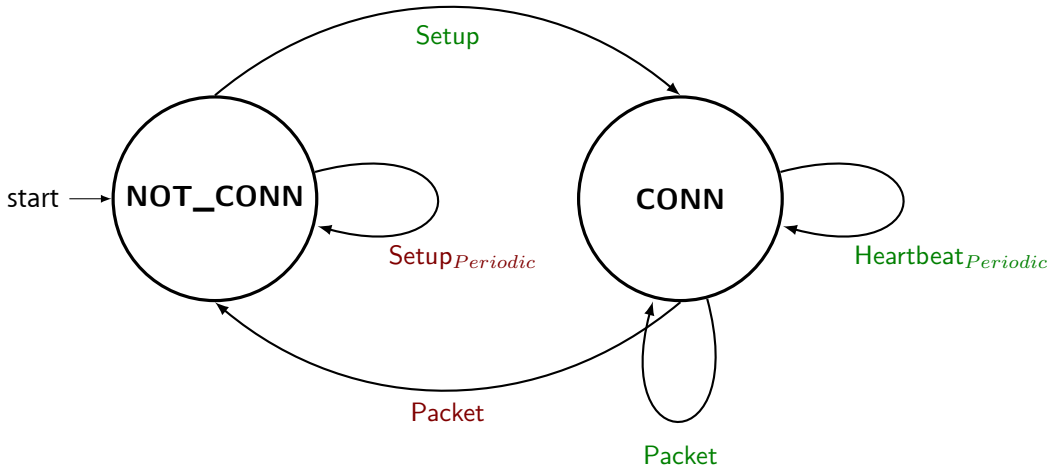

Figure 6: *Device State Machine*

1. Edges with 'green' labels indicate that the Device got an 'Ack' from the Host.
2. Edges with 'red' labels indicate that the Device failed to get an 'Ack' from the Host.
3. In the 'NOT_CONN' state the Device will keep sending periodic Setup packet until it finally gets an 'Ack' from the Host at which point it transistions to the 'CONN' state.
4. In the 'CONN' state the Device is supposed to periodically send the 'Heartbeat' packet as an 'I am alive' signal to the Host.
5. If the Device fails to get an 'Ack' for any packet that it sends to the Host, it transitions back to the 'NOT_CONN' state.