Effective: June, 2023
Supersedes: MSG33-5021-M3, Ed. 07/01/2021

PHD User Guide

API Reference

Parker

ENGINEERING YOUR SUCCESS.

Parker Hannifin Canada
Electronic Motion & Controls Division
1305 Clarence Avenue
Winnipeg, MB R3T 1T4 Canada
office +1 204 452 6776
Fax +1 204 478 1749

http://www.parker.com/ecd

Copyright 2021 © Parker Hannifin Corporation . All rights reserved. No part of this work may be reproduced, published, or distributed in any form or by any means (electronically, mechanically, photocopying, recording, or otherwise), or stored in a database retrieval system, without the prior written permission of Parker Hannifin Corporation in each instance.

Contents

Publication History

The following table provides an overview of the changes made to this document over the course of its publication history.

Release	Description of Change, Date
Rev. 001	First release of this document, 10/19/2017
Rev. 002	Edits throughout 09/11/2018
Rev. 003	Add encoder information, update images, 10/23/2018
Rev. 004	USB cable note, update images, 06/4/2019
Rev. 005	Add deinterlace notes to Chapter 3, 7/30/2019
Rev. 006	Add cos.ext_od_result responses to Appendix B, 9/5/2019
Rev. 007	Change nxs.adc *TEMP_SENSE* key value range, 10/15/2019
Rev. 008	Add to nxs.rtc note, and receive DTC note, 11/1/2019
Rev. 009	Fix *update_interval* parameter in section 3.1, 11/20/2019
Rev. 010	Changes to Auto-Installer section 2.7, 2/17/2020
Rev. 011	Fix *year* parameter in section 2.5.11, 4/7/2020
Rev. 012	Fix *nxs.adc* key values in section 2.5.1, change *publish_unmatched_filename* to *publish_all_filename* and add USB stick notes in Section 2.7.1-2.7.3, 4/16/2020
Rev. 013	Add *nxs.set_auto_brightness* notes on ambient light sensor in section 2.5.12, 4/20/2020
Rev. 014	Add *nxs.din* GPIO09 notes for PHD50 in section 2.5.3, 4/29/2020
Rev. 015	Add Section 3.1.1 Camera Settings, 4/15/2021
Rev. 015	Restructure of entire document, Edits through section 5.1 and beyond 7/1/2021

1. Introduction

These instructions are to be used as a reference tool for the vehicle manufacturer's design, production, and service personnel.
The user of these instructions should have basic knowledge in the handling of electronic equipment.

1.1. Overview

The PHD family of displays are general purpose color displays with capacitive touchscreens suitable for a wide range of industry applications.

The PHD software platform consists of various levels of software that support each other to create a complete working system. Access to the lower level functionality of this system is provided to end users via the Dynamic Communications Platform (DCP).

This document details the events provided to the Crank Storyboard Engine via the PHD API events.evt file. This file, when placed into a Crank Storyboard Designer project's events folder makes those events available to application developers working within the Crank Storyboard Designer.

This document also details all the events available to UI developers for interacting with the system. They are grouped by Storyboard Engine IO channel which also serves to divide them into functional groups.

Simple Lua function examples are provided. For most of these examples, validation of input parameters is not given, as this is simple to do. Any Lua variables referenced in sample functions that are not defined in the function can be assumed to be defined at a higher scope. There would also be opportunity, in a larger project, to combine a lot of these Lua functions into more generic event sending or receiving functions with logic to handle multiple events.

The platform software is consistent across PHD28, PHD50 and PHD70. Where it is not, those differences are called out. In all cases, all events are available across all platforms, but may never occur (for externally generated events) or may be ignored when sent on platforms where they are not supported.

1.2. Definitions

The terminology used in this manual are defined in the following table.

Abbreviation	Explanation
Channel	A channel is an input mechanism for Storyboard IO applications, via which they can receive events.
Crank Storyboard Designer	A desktop development suite that enables user interface designers to easily prototype the look and feel of a product and then deploy it to a target equipped with the Crank Storyboard Engine.
Crank Storyboard Engine	The runtime component that runs on the target device and presents the content developed in Crank Storyboard Designer.

Abbreviation	Explanation
Crank Storyboard IO or Storyboard IO	This is a library and API provided by Crank Software that allows other processes to send events to, and receive events from, the Crank Storyboard Engine, or any other process using the Crank Storyboard IO library. Crank Storyboard IO was previously known as, and often still referred to as “greio”. Every Crank Storyboard IO process creates a named channel for listening and can send events to named channels created by other Crank Storyboard IO processes.
DCP	Dynamic Communications Platform This is an API that abstracts a platform or OS-specific IPC protocol. It is the IPC mechanism via which all of the Parker middleware communicates. It is also the method by which system data is converted into Crank Storyboard IO events for use by the user application.
Event	An identifier used by the Storyboard IO library and the Storyboard Engine to trigger an action. Events contain an event name and an event payload.
Event payload	Data associated with an event. The event payload is described using a format string.
Format string	Provides a description of how to interpret an event payload. The format string uses the format: [number of bytes][signed/unsigned][number of elements][ ] [name] For example, standard C data types would be formatted as follows: **1s0 --> Special null terminated string** **1s1|1u1 --> 1 byte integer (int8_t uint8_t)** **2s1|2u1 --> 2 byte integer (int16_t uint16_t)** **4s1|4u1 --> 4 byte integer (int32_t uint32_t)** **4f1 --> 4 byte floating point (IEEE754 float)** **8s1|8u1 --> 8 byte integer (int64_t uint64_t)** So a structure such as: **struct {** **int32_t itemA;** **uint16_t itemB;** **}** would have the format string: *4s1 itemA 2u1 itemB.* When using numeric values, this API has, where ever possible, adopted a 32-bit signed or unsigned value. This is because ARM processors are more efficient when using 32-bit quantities than other data sizes.
GPL	The GNU General Public License, the license under which Linux kernel source code and ancillary GPL utilities are released.
IPC	Inter-Process Communications. A mechanism for moving a data between two (or more) processes that may reside in separate address spaces.
Key	An identifier paired with a value, attached to an event. A key-value pair is represented in this document as (key, value). Those parentheses serve to delimit the text and are not used during programming.
NCC	Non-volatile customer configuration (or ncc) is the component that handles access to the non-volatile memory available on the system. It uses Storyboard I/O channel “ncc”.

Abbreviation	Explanation
Nexus	Nexus (or nxs) is the system hardware interface component that bundles together much of the low-level functionality (ADC, backlight, RTC) for presentation to Crank via DCP. It uses Storyboard I/O channel “nxs”.
Storyboard application	An application developed using the Crank Storyboard Designer that can run on a target using the Crank Storyboard Engine. The Storyboard application communicates with lower-level functionality via DCP using the events described in this API.
UI	User Interface. This is the graphical component of a Storyboard application used to interact with the device.
Value	Data associated with a key. Supported types are signed and unsigned integers (8, 16, 32 or 64 bits), floats (32-bit only), arrays of integers or floats, and null-terminated strings. A key-value pair is represented in this document as (key, value). Those parentheses serve to delimit the text and are not used during programming.

1.3. Product documentation

The following publications are relevant for users of this product.

- PHD Catalog datasheets MSG33-5021/US thru MSG33-5023/US
- PHD User Guide/Instruction book MSG33-5021-IB/US
- PHD Application Packager manual MSG33-5021-M1/US
- PHD J1939 Stack Generation manual MSG33-5021-M4/US

All documentation may be found on our web pages, located at www.parker.com/ecd.

Contact the manufacturer if there is anything you are not sure about or if you have any questions regarding the product and its handling or maintenance.

The term “manufacturer” refers to Parker-Hannifin Corporation.

2. Application Programming Interface

2.1. Description

This section describes the events available to Storyboard Designer for sending data to and receiving data from the rest of the system.

The API definitions are divided into functional blocks, although there are a number of DCP management events that are available across all I/O channels. Those are described first, in the next section.

All DCP-related content is sent from the Storyboard application via events.

For many events, multiple keys are supported. When sending a multiple key event, not all keys need to be specified, only those of interest.

For example, if sending an nxs.set_dout message to set DOUTs GPIO2 and P5V0, only those two keys, along with their values, would need to be sent. Any keys not specified would retain their current state.

PHD specific events should not be subscribed to in the gre.init event. It is recommended that PHD specific events be subscribed to in the gre.screenshow.post event for whatever the main entry screen into the application is during startup.

In some conditions, Lua script allows for direct access to certain registers on the microprocessor. Writing to registers in this fashion bypasses the API set and is strongly discouraged. Doing so may result in undocumented and undesirable effects in the operation and performance of the PHD.

2.2. DCP Management Events

2.2.1. phd.subscribe

Event	*phd.subscribe*
Event Data	
Keys	Values
event	<event_name>
receiver	<IO channel>
Direction	Output
Channel	any
Notes	Sent by the application to tell DCP it wishes to subscribe to a particular event or class of events. For example, to subscribe to the system hardware interface backlight events, send the *phd. subscribe* event to the system hardware interface channel (“nxs”) with data containing the event name and the IO channel name of the receiver (usually the Storyboard application name). (“event”, “<event_name>”) (“receiver”, “storyboard_app.gapp”) If the application does not subscribe to a class of events, it does not receive them. This avoids the problem of DCP flooding the application with every DCP message in the system. Subsystems can communicate with each other via DCP without disturbing the application. Please note that Lua examples for received events assume that the Storyboard application has subscribed to the event via this mechanism and reception of the event is set to trigger a Lua function. Also note you do not need to subscribe to a channel to send events to it.

Lua Example	**function register**(mapargs)
	    local data = {}
	    data[“event”] = “nxs.backlight”
	    data[“receiver”] = “storyboard_app_name.gapp”
	    gre.send_event_data(“**phd.subscribe**”,
	                    “1s0 event 1s0 receiver”,
	                    data, “nxs”)
	end

2.3. Non-volatile customer configuration channel events

2.3.1. ncc.command

Event	*ncc.command*	
Event Data		
Keys	**Values**	**Description**
cmd	“read” or “write”	Desired command for non-volatile memory.
addr	<0..32767>	Address, in bytes, from which to read or write.
len	<1..4096>	Number of bytes to read or write.
fmt	“string”	A Crank-style format specifier, e.g. 1s0, 4u1 used to pack the values.
data	value	Used when writing only. Size and format of the data depends on size and format being written.
Direction	output	
Channel	ncc	
Notes	Sent by the application to issue a command to the non-volatile memory. “read” command: Assuming all keys are present and their values are valid, the data is extracted from memory and the *ncc.response* event is posted. You would need to subscribe to this event to see the result of your “read” command. The read example below demonstrates a request to read 4 bytes of data from offset 1024. Note that *fmt* is set to 4u1, so *ncc.response* returns 1 element of 4 bytes. “write” command: The write example below demonstrates modifying a 4-byte value in the non-volatile memory at offset 0.	

Lua Example	function **request_read**(mapargs)
	    local data = {}
	    data[“cmd”] = “read”
	    data[“addr”] = 1024
	    data[“len”] = 4
	    data[“fmt”] = “4u1”
	    gre.send_event_data(“ncc.command”,
	                    “1s0 cmd 4u1 addr 4u1 len 1s0 fmt”, data, “ncc”)
	end
	function **write_ncc**(mapargs)
	    local data = {}
	    data[“cmd”] = “write”
	    data[“addr”] = 1024
	    data[“len”] = 4
	    data[“fmt”] = “4u1”
	    data[“data”] = 0xDEADBEEF
	    gre.send_event_data(“ncc.command”,
	                    “1s0 cmd 4u1 addr 4u1 len 1s0 fmt”, data, “ncc”)
	end	

2.3.2. ncc.response

Event	*ncc.response*	
Event Data		
Keys	**Values**	**Description**
addr	<0..32767>	The returned data’s address in the non-volatile memory.
data	<0..4095>	The returned data from the non-volatile memory.
Direction	input	
Channel	ncc	
Notes	Received by the application in response to an *ncc.command* “read” command. The example below assumes the Storyboard application is set up to call Lua function **ncc_response** when the *ncc.response* event occurs. It further assumes a simple 4-byte integer value was requested. Event data is available through the *mapargs* argument.	

Lua Example	`function ncc_response(mapargs)`
	`    local ev = mapargs.context_event_data`
	`    local addr = ev[“addr”]`
	`    local data = ev[“data”]`
	`    if (addr ~= nil and data ~= nil) then`
	`        print(“Read data ”..data..“ from address ”..addr)`
	`    end`
	`end`	

2.4. Version supervisor events

2.4.1. serial

Event	*serial*	
Event Data		
Keys	**Values**	**Description**
name	<string>	Name of device (e.g. PHD28, PHD50, PHD70).
pcb_serial	<string>	Serial number of circuit board.
pcb_mfgdate	<string>	Manufacturing date of circuit board, in YYYYMMDD format.
serial	<string>	Serial number of the device.
mfgdate	<string>	Manufacturing date, in YYYYMMDD format.
sap	<string>	SAP number of the device.
variant	<string>	Model and revision of the device
Direction	input	
Channel	version	
Notes	Received by the application when subscribed to. The *serial* event contains the name, serial number and manufacturing date for the device. The key names are *name, pcb_serial, pcb_mfgdate, serial, and mfgdate*. The example below assumes the Storyboard application is set up to call Lua function **serial_receive** when the *serial* event occurs. Event data is available through the *mapargs* argument.	

Lua Example

```
function serial_receive(mapargs)
    local ev = mapargs.context_event_data
    local name = ev[“name”]
    local serial = ev[“serial”]
    local mfgdate = ev[“mfgdate”]
    local pcb_serial = ev[“pcb_serial”]
    local pcb_mfgdate = ev[“pcb_mfgdate”]

    if (name ~= nil and serial ~= nil and mfgdate ~= nil
        and pcb_serial ~= nil and pcb_mfgdate ~= nil) then
        print(“Model: “. . ev[“name”])
        print(“PCB serial number: “. . ev[“pcb_serial”])
        print(“PCB manufactured: “. . ev[“pcb_mfgdate”])
        print(“Serial number: “. . ev[“serial”])
        print(“Manufactured: “. . ev[“mfgdate”])
    end
end
```

2.4.2. Version

Event	*version*	
Event Data		
Keys	**Values**	**Description**
count	<0..100>	Number of version items in message.
partnumX	<0..4294967295>	Software part number for entry X.
nameX	<string>	Software name for entry X.
majorX	<0..4294967295>	Software major version number for entry X.
minorX	<0..4294967295>	Software minor version number for entry X.
Direction	input	
Channel	version	
Notes	Received by the application when subscribed to. The *version* event contains count version items, each consisting of the four keys, *partnumX*, *nameX*, *majorX* and *minorX*, where X is a number from 0 to count. The example below assumes the Storyboard application is set up to call Lua function **version_receive** when the *version* event occurs. Event data is available through the *mapargs* argument. Note that Lua “for loops” are inclusive, so the example below is written to loop from 0 through count -1.	

Lua Example	function version_receive(mapargs)
	    local ev = mapargs.context_event_data
	    local count = ev[“count”] - 1
	    local i
	    for i = 0,count do
	        print(“Name: “. . ev[“name”..i])
	        print(“Part number: “. . ev[“partnum”..i])
	        print(“Version: V”..ev[“major”..i]..”.”..
	            string.format(“%02d”, ev[“minor”..i]))
	    end
	-- Input validation of partnumX against nil left as an exercise
	end	

2.4.3. version.set

Event	*version.set*	
Event Data		
Keys	**Values**	**Description**
partnum	<0..4294967295>	Software part number. This key is required.
name	<string>	Friendly name of the software. This key is optional. “unknown” will be substituted if name is not provided.
major	<0..4294967295>	Software major version number. This key is optional. 0 will be substituted if *major* is not provided.
minor	<0..4294967295>	Software minor version number. This key is optional. 0 will be substituted if *minor* is not provided.
Direction	output	
Channel	version	
Notes	Sent by a program to register its part number, name and version information with the version supervisor. The *version.set* event consists of four keys, *partnum*, *name*, *major* and *minor*. While this event can certainly be used by a Crank application to register its own part number, name and version with the version supervisor, it will primarily be used by other applications on the system to register themselves. This will enable Crank to read the device’s complete version information by registering for the *version* event.	
Lua Example	function register_version(mapargs) local data = {} data[“partnum”] = 1042699 data[“name”] = “Heavy Lift App” data[“major”] = 1 data[“minor”] = 3 gre.send_event_data(“version.set”, “4u1 partnum 1s0 name 4u1 major 4u1 minor”, data, “version”) -- Output data left as an exercise end	

2.5. System hardware interface channel events

2.5.1. nxs.adc

Event	*nxs.adc*	
Event Data		
Keys	**Values**	**Description**
AMB_LIGHT	<0..1023> PHD70 <0..4095> PHD28/50	Ambient light sensor.
DISPBKLT_VOUT	<0..36000>	Display backlight voltage, millivolts. Only available on PHD70.
OUTPUT1_SENSE	<0..VBATT in mV>	Low-side output 1sense, millivolts.
OUTPUT2_SENSE	<0..VBATT in mV>	Low-side output 2 sense, millivolts.
P12V0_OUT_SENSE	<0..12000>	12V output sense, millivolts.
P5V0_OUT_SENSE	<0..5000>	5V output sense, millivolts.
TEMP_SENSE	<-40000..130000>	Temperature, millidegrees Celsius.
VBATT_SENSE	<0..36000>	Battery voltage, millivolts.
GPIOx	<0..5000>	GPIO1 – GPIO10 input voltage, millivolts. Replace x with the GPIO number of interest. Not all GPIOs may be present or enabled on all platforms.
Direction	input	
Channel	nxs	
Notes	Received by the application in response to an *nxs.request_adc* event. The example below assumes the Storyboard application is set up to call Lua function **adc_receive** when the *nxs.adc* event occurs. Event data is available through the *mapargs* argument. The state of every ADC channel is returned with every response to an *nxs.request_adc* event.	

Lua Example	`function adc_receive(mapargs)`
	`    local ev = mapargs.context_event_data`
	`    local temperature = ev[“TEMP_SENSE”]`
	`        print(“Received ADC update: “. . temperature)`
	`    -- Other values left as an exercise`
	`end`	

2.5.2. nxs.backlight

Event	*nxs.backlight*	
Event Data		
Keys	**Values**	**Description**
display_level	<0..100>	0 = 0% backlight, 100 = 100% backlight.
Direction	input	
Channel	nxs	
Notes	Received by the application whenever the backlight level changes. The values range from 0 to 100. The example below assumes the Storyboard application is set up to call Lua function **backlight_receive** when the *nxs.backlight* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	function backlight_receive(mapargs)
	local ev = mapargs.context_event_data
	local level = ev[“display_level”]
	if (level ~= nil) then
	print(“Received brightness: “..level)
	end
	end	

2.5.3. nxs.din

Event	*nxs.din*	
Event Data		
Keys	**Values**	**Description**
GPIOx	<0..10>	Digital input GPIOx, where x is a number from 1 to 11 representing the digital input number. Key names are GPIO1 through GPIO11 for platforms with 11 digital inputs. Not all GPIOs may be present or enabled on all platforms.
GPIO9	<0..Max Hz>	If the FIN is enabled on the PHD50 for GPIO9, the GPIO09 key will return the frequency value. When the FIN for GPIO9 is enabled, the values will be sent out Analog input Sample Rate value setup in the PHD Packager during load time GPIO09 key will return the frequency value. **PHD50 only.**
Direction	input	
Channel	nxs	
Notes	Received by the application when initially subscribed to, and whenever the DIN states change. The example below assumes the Storyboard application is set up to call Lua function **din_receive** when the *nxs.din* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	`function din_receive(mapargs)`
	`    local ev = mapargs.context_event_data`
	`    local gpio1 = ev[“GPIO1”]`
	`    if (gpio1 ~= nil) then`
	`        print(“Received DIN update: “..gpio1)`
	`    end`
	`    -- Values GPIO2 through GPIO10 left as an exercise`
	`end`	

2.5.4. nxs.dout

Event	*nxs.dout*	
Event Data		
Keys	**Values**	**Description**
OUTPUT1	<0..1>	Low side output 1. 0 = disabled. 1 = enabled.
OUTPUT2	<0..1>	Low side output 2. 0 = disabled. 1 = enabled.
P12V0	<0..1>	12V output. 0 = disabled. 1 = enabled.
P5V0	<0..1>	5V output. 0 = disabled. 1 = enabled.
GPIOx	<0..1>	Digital output *GPIOx* where *x* is a number from 2 to 10 representing the digital output number. Key names are GPIO2 through GPIO10 for platforms with 9 digital outputs. Not all GPIOs may be present or enabled on all platforms. Keypad configuration may mean some GPIOs are not available as DOUTs.
HOLDPWR	<0..1>	Only available on PHD50 with enhanced power manaagement circuit. 0 = disabled. 1 = enabled (i.e. PHD50 will continue operation when GPIO1 goes low.
Direction	input	
Channel	nxs	
Notes	Received by the application when subscribed to. Used to query the status of the douts The example below assumes the Storyboard application is set up to call Lua function **dout_receive** when the *nxs.dout* event occurs. Event data is available through the *mapargs* argument.	

Lua Example

```
function dout_receive(mapargs)
    local ev = mapargs.context_event_data
    local output1 = ev[“OUTPUT1”]
    if (output1 ~= nil) then
        print(“State of OUTPUT1 is”)
      if (output1 == 0) then
          print(“disabled”)
      else
          print(“enabled”)
      end
    end
    -- Other DOUT values left as an exercise
end
```

2.5.5. nxs.dout_errors

Event	*nxs.dout_errors*	
Event Data		
Keys	**Values**	**Description**
dout_errors	0x0004 – OUTPUT1_SHORT_BATT 0x0008 – OUTPUT1_SHORT_GND 0x0010 – OUTPUT2_SHORT_BATT 0x0020 – OUTPUT2_SHORT_GND 0x0040 – P5V0_SHORT_BATT 0x0080 – P5V0_SHORT_GND 0x0100 – P12V0_SHORT_BATT 0x0200 – P12V0_SHORT_GND	A 16-bit value with each bit flagging a different dout error.
Direction	input	
Channel	nxs	
Notes	Received by the application in response to an *nxs.dout_errors* event. More than one bit may be set at a time. The bit field is only published when it changes; this includes when there is no longer an error condition. The example below assumes the Storyboard application is set up to call Lua function **dout_errors_receive** when the *nxs.dout_errors* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	function **dout_errors_receive**(mapargs)
	local ev = mapargs.context_event_data
	local errors = ev[“dout_errors”]
	if (errors ~= nil) then
	if (**bit32**.band(errors,0x0004) ~= 0x0000) then
	print(“OUTPUT1_SHORT_BATT error bit set.”)
	end
	end
	-- Other DOUT errors left as an exercise
	end	

2.5.6. nxs.encoder_val

Event	*nxs.encoder_val*	
Event Data		
Keys	**Values**	**Description**
encoder_value	<0..4294967295>	Encoder value.
Direction	input	
Channel	nxs	
Notes	Received by the application when the encoder is rotated clockwise or counter-clockwise. For each clockwise “click” of the encoder, the value increases by one. For each counter-clockwise “click” of the encoder, the value decreases by one. The initial Encoder value is 0xFFFFFFFF / 2 or 2147483647. This value minimizes the likelihood of a roll over or roll under of the count, but the possibility of a roll over or roll under should still be accounted for. The example below assumes the Storyboard application is set up to call Lua function **encoder_receive** when the *nxs.encoder_value* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	**function encoder_receive**(mapargs)
	local ev = mapargs.**context_event_data**
	if (ev[“encoder_value”] ~= **nil**) **then**
	print(“Encoder updated to value” ..ev[“encoder_value”])
	end
	end	

** NOTE ** Only the PHD50 and PHD70 support encoder input.		

2.5.7. nxs.key_press

Event	*nxs.key_press*	
Event Data		
Keys	**Values**	**Description**
keypress	<key_string>	A string containing the name of the key that was pressed.
Direction	input	
Channel	nxs	
Notes	Received by the application when a keypad key is pressed. The key_string published is read from the keypad configuration XML file. The example below assumes the Storyboard application is set up to call Lua function **key_press** when the *nxs.key_press* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	function **key_press**(mapargs)
	    local ev = mapargs.context_event_data
	    local key = ev[“keypress”]
	    if (key ~= nil) then
	        print(“Key pressed:”..key)
	    end
	end	

2.5.8. nxs.key_release

Event	*nxs.key_release*	
Event Data		
Keys	**Values**	**Description**
keyrelease	<key_string>	A string containing the name of the key that was released.
Direction	input	
Channel	nxs	
Notes	Received by the application when a keypad key is released. The key_string published is read from the keypad configuration XML file. The example below assumes the Storyboard application is set up to call Lua function **key_release** when the *nxs.key_release* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	function **key_release**(mapargs)
	    local ev = mapargs.context_event_data
	    local key = ev[“keyrelease”]
	    if (key ~= nil) then
	        print(“Key released: ”..key)
	    end
	end	

2.5.9. nxs.request_adc

Event	*nxs.request_adc*	
Event Data		
Keys	**Values**	**Description**
adc	<string>	Any data can be provided
Direction	output	
Channel	nxs	
Notes	Sent by the application to request ADC values. The example below assumes the Storyboard application is set up to call Lua function *request_adc* periodically to request ADC values. Note that the “adc” key merely has to be provided to confirm that you are interested in the ADC data. The value provided for the key is not relevant, as all ADCs are returned with each *nxs.adc* event.	

Lua Example	function **request_adc**(mapargs)
	    local data = {}
	    data[“adc”] = “1”
	    gre.send_event_data(“nxs.request_adc”, “1s0 adc”, data, “nxs”)
	end	

2.5.10. nxs.request_rtc

Event	*nxs.request_rtc*	
Event Data		
Keys	**Values**	**Description**
rtc_get	*<string>*	Real time clock.
Direction	output	
Channel	nxs	
Notes	Sent by the application to request the RTC values. The example below assumes the Storyboard application is set up to call Lua function **request_rtc** periodically to request RTC values. Note that the value of rtc_get is not used therefore it can be any string.	

Lua Example	function **request_rtc**(mapargs)
	    local data = {}
	    data[“rtc_get”] = “1”
	    gre.send_event_data(“**nxs.request_rtc**”, “**1s0 rtc_get**”, data, “**nxs**”)
	end	

** NOTE ** Only the standard PHD70 supports RTC, not the reduced memory PHD70.		

2.5.11. nxs.rtc

Event	*nxs.rtc*	
Event Data		
Keys	**Values**	**Description**
seconds	<0..59>	
minutes	<0..59>	
hours	<0..23>	
day	<1..31>	
month	<0..11>	January=0. . December=11
year	<2000..2099>	
timezone	<string>	
Direction	input	
Channel	nxs	
Notes	Received by the application in response to an *nxs.request_rtc* event. Note that if subscribed to, this event generates a new value every second. In addition, the application is not expected to use the RTC to keep track of time. Rather, it should use the system time, which is set by the PHD using the RTC on each startup. The example below assumes the Storyboard application is set up to call Lua function **rtc_receive** when the *nxs.rtc* event occurs. Event data is available through the *mapargs* argument.	
Lua Example	function **rtc_receive**(mapargs) local ev = mapargs.context_event_data local sec = ev[“second”] local min = ev[“minute”] local hr = ev[“hour”] local day = ev[“day”] local mon = ev[“month”] + 1 local yr = ev[“year”] local tz = ev[“timezone”] **print**(yr..“-”..mon..“-”..day..” “..hr..“:”..min..“:”..sec) **print**(“Time zone: ”..tz) end	
** NOTE ** Only the standard PHD70 supports RTC, not the reduced memory PHD70. VBATT must stay on to the PHD to maintain the RTC settings. It is recommended to use the wake/sleep functionality to turn “Off” the PHD rather than cycle power to maintain the RTC settings.		

2.5.12. nxs.set_auto_brightness

Event	*nxs.set_auto_brightess*	
Event Data		
Keys	**Values**	**Description**
auto_brightness	<on..off>	'on' means auto brightness is enabled, 'off' means auto brightness is disabled.
Direction	output	
Channel	nxs	
Notes	Sent by the application when the application wishes to control the auto brightness feature. Using the value 'on' will enable the auto brightness feature and using the value 'off' will disable it. The auto brightness feature is 'off' by default. If set, the auto brightness will automatically dim the backlight per the following: **PHD28, PHD50**: If the ambient light sensor returns a value above 3840, then the backlight will adjust to 20% If the ambient light sensor returns a value below 1024, then the backlight will adjust to 100%. **PHD70:** If the ambient light sensor is above 800, then the backlight will adjust to 20%. If the ambient light sensor is below 800, then the backlight will adjust to 90%.	

Lua Example	`function enable_auto_brightness(mapargs)`
	`    local data = {}`
	`    data["auto_brightness"] = "on"`
	`    gre.send_event_data("nxs.set_auto_brightness",`
	`                        "1s0 auto_brightness", data, "nxs")`
	`end`	

2.5.13. nxs.set_backlight

Event	*nxs.set_backlight*	
Event Data		
Keys	**Values**	**Description**
display_level	<0..100>	0 = 0% backlight, 100 = 100% backlight.
Direction	output	
Channel	nxs	
Notes	Sent by the application to adjust the backlight level. The values range from 0 to 100. The example below assumes that the Storyboard application has, based on user interaction with a screen brightness control, called a Lua function called **set_backlight** with an argument of value and a number from 0 to 100.	

Lua Example

```
function set_backlight(mapargs)
      local level = mapargs["value"]
      local data = {}
      if (level ~= nil) then
          print("Set brightness: "..level)
          data["display_level"] = level
          gre.send_event_data("nxs.set_backlight",
                             "4u1 display_level", data, "nxs")
      end
end
```

** NOTE ** If nxs.set_auto_brightness is set to “on”, this value won’t be set.

2.5.14. nxs.set_dout

Event	*nxs.set_dout*	
Event Data		
Keys	**Values**	**Description**
OUTPUT1	<0..1>	Low side output 1. 0 = disabled. 1 = enabled.
OUTPUT2	<0..1>	Low side output 2. 0 = disabled. 1 = enabled.
P12V0	<0..1>	12V output. 0 = disabled. 1 = enabled.
P5V0	<0..1>	5V output. 0 = disabled. 1 = enabled.
GPIO2	<0..1>	GPIO output 2. 0 = disabled. 1 = enabled.
GPIO3	<0..1>	GPIO output 3. 0 = disabled. 1 = enabled.
GPIO4	<0..1>	GPIO output 4. 0 = disabled. 1 = enabled.
GPIO5	<0..1>	GPIO output 5. 0 = disabled. 1 = enabled.
GPIO6	<0..1>	GPIO output 6. 0 = disabled. 1 = enabled.
HOLDPWR	<0..1>	Only available on the PHD50 with enhanced power management circuit. Set to 1 to allow the PHD50 to continue operating after GPIO1 goes low.
Direction	output	
Channel	nxs	
Notes	Sent by the application to set the state of digital outputs. The example below assumes the Storyboard application has called the function **enable_outputs** which set enables OUTPUT1 and disables OUTPUT2. Note that all digital outputs need not be present in every *nxs.set_dout* event. Only those digital outputs specified in the message are set. A “1” enables an output and a “0” disables it.	

Lua Example	function **enable_outputs**(mapargs)
	    local data = {}
	    data[“OUTPUT1”] = “1”
	    data[“OUTPUT2”] = “0”
	    gre.send_event_data(“**nxs.set_dout**”,
	                    “4u1 OUTPUT1 4u1 OUTPUT2”, data, “nxs”)
	end

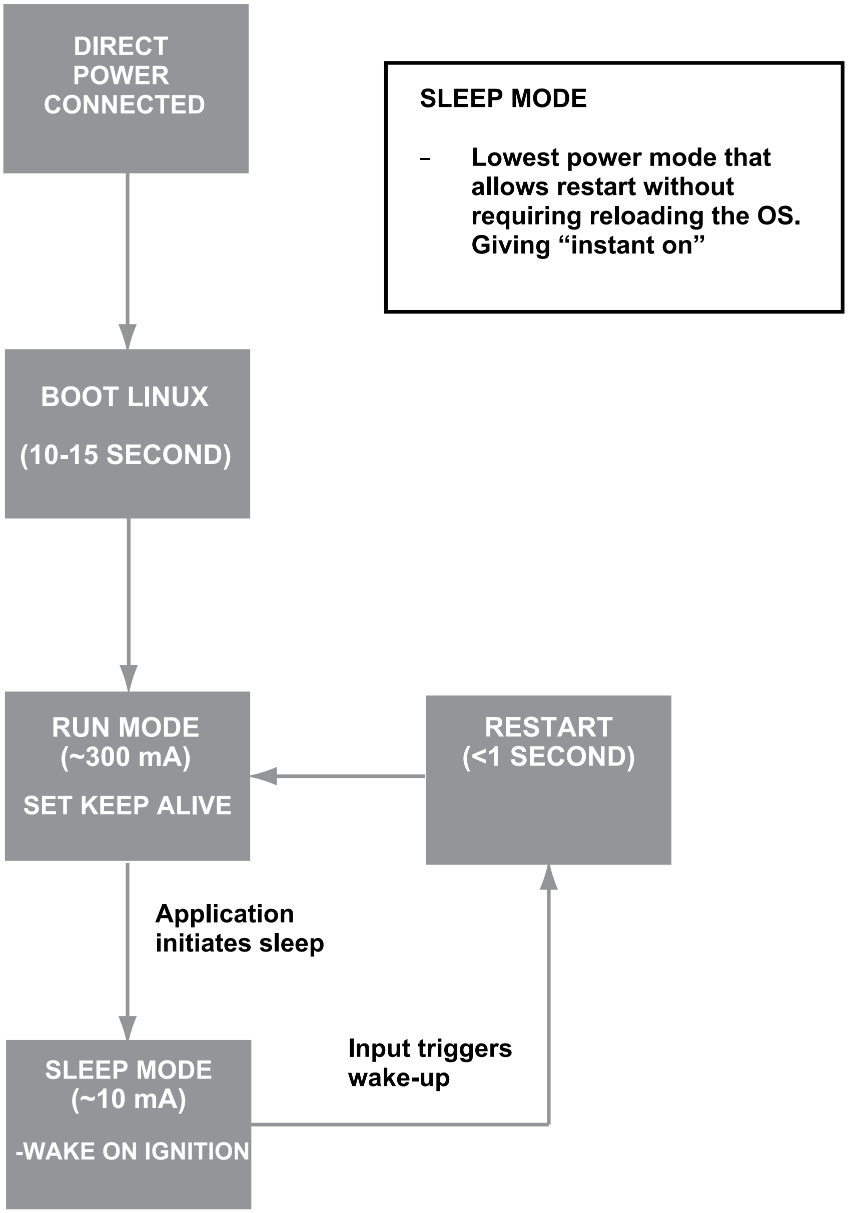
2.5.15. nxs.set_heater

Event	*nxs.set_heater*	
Event Data		
Keys	**Values**	**Description**
heater_level	<0..100>	0 = 0% heater, 100 = 100% heater.
Direction	output	
Channel	nxs	
Notes	Sent by the application to adjust the heater level. The values range from 0 to 100. The example below assumes that the Storyboard application has, based on user interaction with a heater level control, called a Lua function called **set_heater_level** with an argument of value and a number from 0 to 100.	

Lua Example	function **set_heater_level**(mapargs)
	    local level = mapargs[“**value**”]
	    local data = {}
	    if (level ~= nil) then
	      print(“**Set heater level:** “..level)
	      data[“**heater_level**”] = level
	      gre.send_event_data(“**nxs.set_heater**”,
	                    “4u1 heater_level”, data, “nxs”)
	    end
	end	

** NOTE ** Only the PHD28 and PHD50 support the internal heater.		

Standard PHD50 power management states and transitions

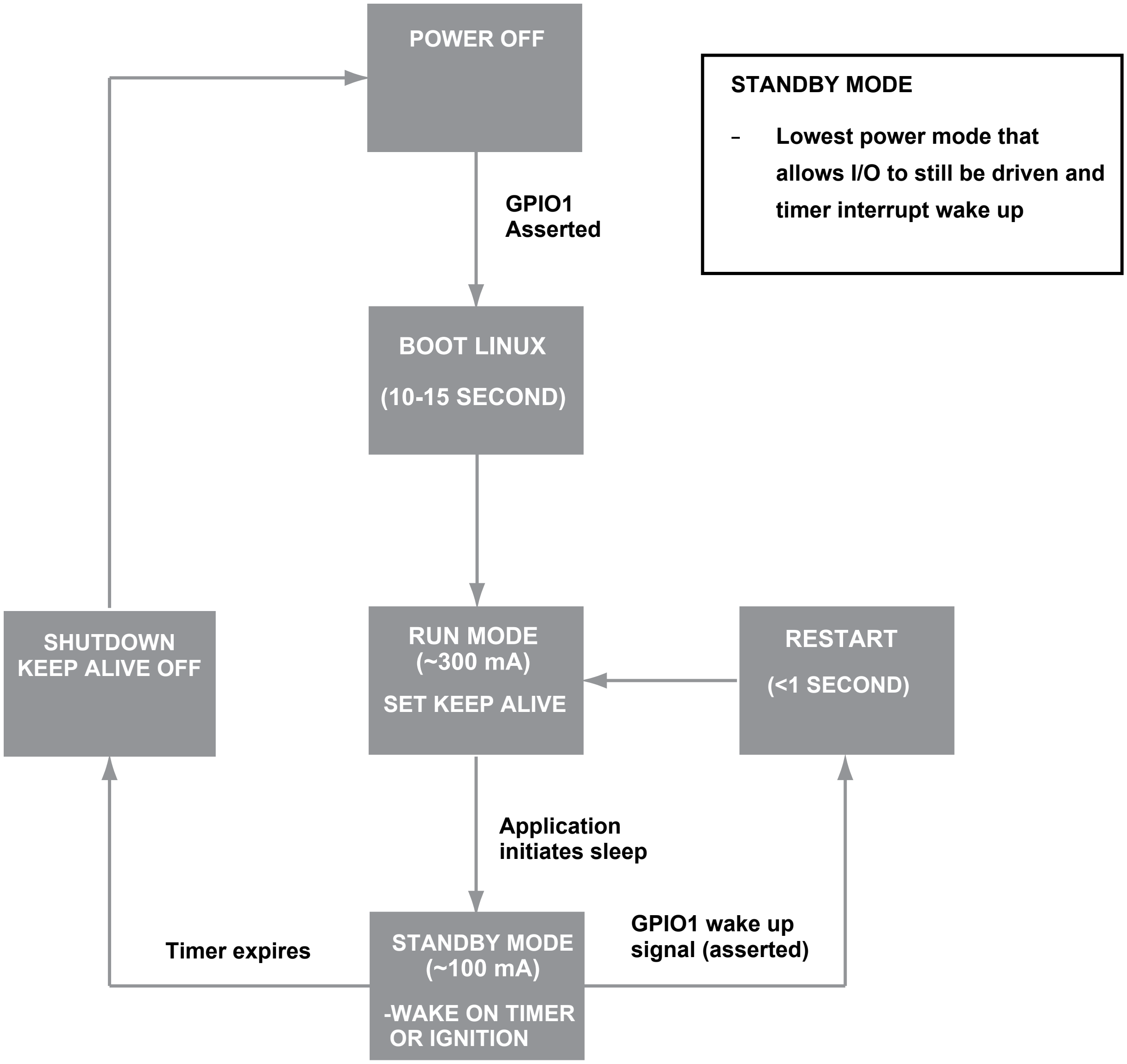


SLEEP MODE

- **Lowest power mode that allows restart without requiring reloading the OS. Giving "instant on"**

The PHD50 with Enhanced Power Management Circuitry (EPMC) supports a second low-power mode (standby). In this mode, the display consumes more power while asleep, but it can be awoken by an externaltrigger or an internal timer. If the timer wakes the display it will then transition to OFF.

PHD50 with EPMC power management states and transitions


POWER OFF
STANDBY MODE
- Lowest power mode that allows I/O to still be driven and timer interrupt wake up
GPIO1 Asserted
BOOT LINUX
(10-15 SECOND)
SHUTDOWN KEEP ALIVE OFF
RUN MODE
(~300 mA)
SET KEEP ALIVE
RESTART
(<1 SECOND)
Application initiates sleep
Timer expires
STANDBY MODE
(~100 mA)
-WAKE ON TIMER OR IGNITION
GPIO1 wake up signal (asserted)

2.5.16. nxs.set_rtc

Event	*nxs.set_rtc*	
Event Data		
Keys	**Values**	**Description**
seconds	<0..59>	
minutes	<0..59>	
hours	<0..23>	
day	<1..31>	
month	<0..11>	January=0. . December=11
year	<1900..no limit>	
timezone	<string>	
Direction	output	
Channel	nxs	
Notes	Sent by the application to set the real-time clock. The example below assumes that the Storyboard application has, based on user interaction with a screen RTC control, called a Lua function called **set_rtc** with arguments of *seconds, minutes, hours, day, month, year* and *timezone*. Time zone must always be provided even if you don't wish to change it. The example below forces the time zone to "US/Eastern". Other time zone strings may be substituted. Please see Appendix C.	

Lua Example	**function** **set_rtc**(mapargs)
	    local data = {}
	    data["seconds"] = mapargs["seconds"]
	    data["minutes"] = mapargs["minutes"]
	    data["hours"] = mapargs["hours"]
	    data["day"] = mapargs["day"]
	    data["month"] = mapargs["month"]
	    data["year"] = mapargs["year"]
	    data["timezone"] = "US/Eastern"
	    gre.send_event_data("**nxs.set_rtc**",
	                    "4u1 seconds 4u1 minutes 4u1 hours
	                    4u1 day 4u1 month 4u1 year 1s0 timezone",
	                    data, "nxs")
	end	

** NOTE ** Only the standard PHD70 supports RTC, not the reduced memory PHD70.		

2.5.17. nxs.set_timezone

Event	*nxs.set_timezone*	
Event Data		
Keys	**Values**	**Description**
timezone	<string>	
Direction	output	
Channel	nxs	
Notes	Sent by the application to set the time zone. The example below sets the time zone to “US/Eastern”. Other time zone strings may be substituted. Please see Appendix C. Note that setting the time zone will result in the reported system time changing to reflect the new time zone.	
Lua Example	function **set_timezone**(mapargs) local data = {} data[“timezone”] = “US/Eastern” **gre**.send_event_data(“**nxs.set_timezone**”, “1s0 timezone”, data, “nxs”) end	

2.5.18. nxs.sleep

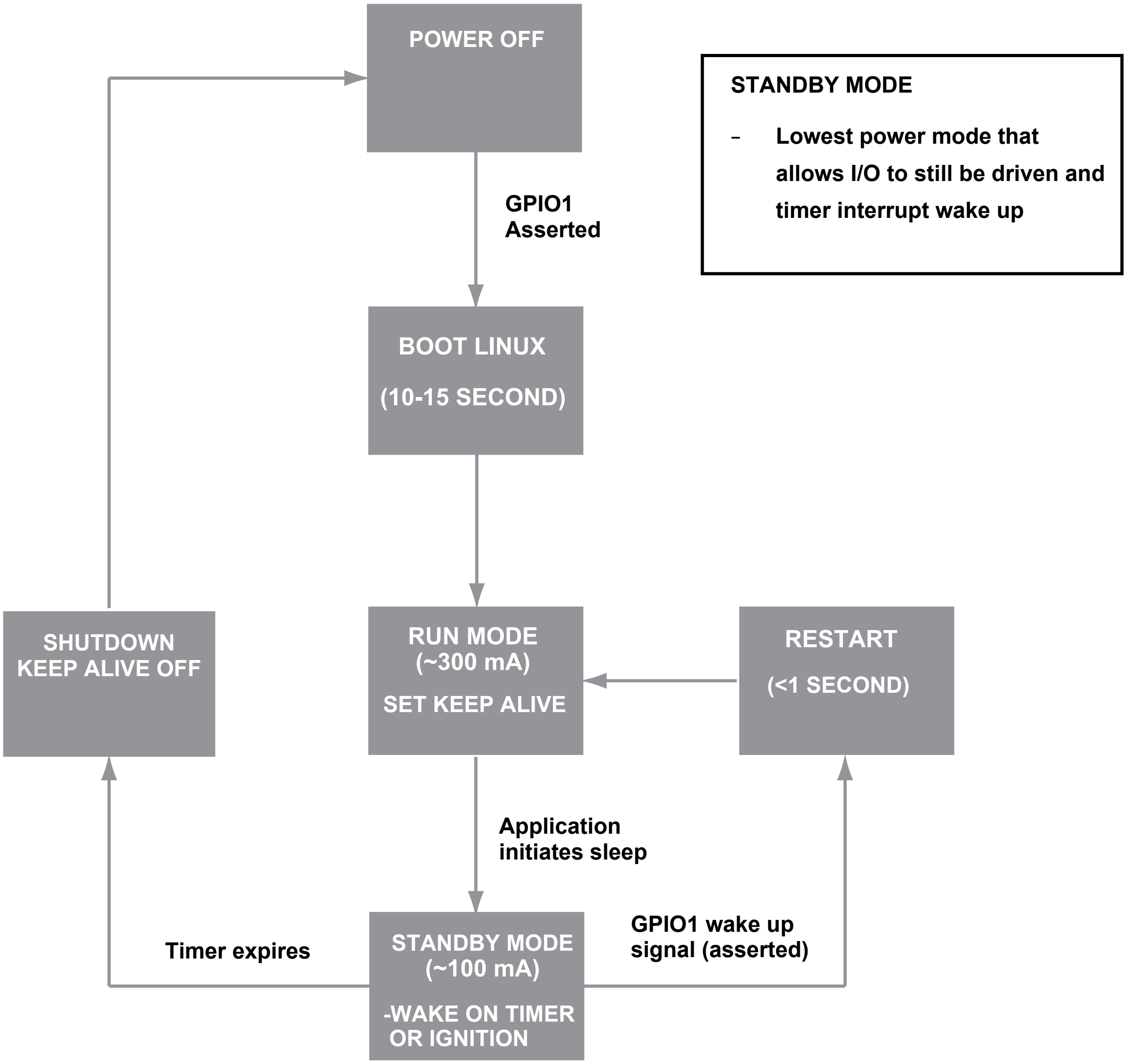
Event	*nxs.sleep*	
Event Data		
Keys	**Values**	**Description**
sleep_mode	<string>	String which sets the type of sleep to put the unit into. A value of “mem” puts the unit into memory sleep, which is the standard sleep mode without any timeout. Any other value puts the unit into enhanced sleep mode, which is only available on hardware with the enhanced power management circuit. After the timer runs out, the unit will power off.
Direction	output	
Channel	nxs	
Notes	Sent by the application to put the display into sleep mode.	

Lua Example	-- put the display into memory sleep
	function **suspend**(mapargs)
	    local data = {}
	    data[“sleep_mode”] = “mem”
	    gre.send_event_data(“**nxs.sleep**”,
	                        “1s0 sleep_mode”, data, “nxs”)
	end	

** NOTE ** It is the user’s responsibility to ensure that the application puts the system outputs into the desired state on resume from sleep. This means setting DOUTs, PWMs and so on to desired levels upon resume. Camera playback may not start automatically on resume, so camera playback should be disabled using *gra.media.stop* command before issuing the *nxs.sleep* command. When the application is being updated, and loaded to flash, the current application in RAM is still running. The application programmer needs to take care that the application does not go to sleep while an updated application is being received by checking the ai.install.state event. When the PHD enters sleep mode, the 5VDC and 12VDC power supplies will turn off so devices that are intended to wake the PHD should not use these power supplies as they will be powered off when the PHD is in sleep mode.		

Enhanced Sleep Mode Operation

Enhanced sleep mode offers a latent power down feature after the PHD enters sleep mode and a timer has elapsed. It is activated only on GPIO1 and is usually used as the Ignition input from the key switch. Vehicle power should still be applied to the VBATT pins to power the PHD. The state of GPIO1 must be monitored by the PHD application. When GPIO1 goes low, a nxs.sleep() event should be sent to put the PHD into enhanced sleep mode. Any corresponding clean up and application sleep preparation steps need to be taken before the nxs.sleep() event is sent. The enhanced sleep operation is shown in the block diagram below. While in enhanced sleep mode, before the timer runs out, the PHD can be woken through GPIO1 going high. After the timer runs out, the PHD will be rebooted through GPIO1 going high.


POWER OFF
GPIO1 Asserted
BOOT LINUX
(10-15 SECOND)
STANDBY MODE
- Lowest power mode that allows I/O to still be driven and timer interrupt wake up
SHUTDOWN
KEEP ALIVE OFF
RUN MODE
(~300 mA)
SET KEEP ALIVE
RESTART
(<1 SECOND)
Application initiates sleep
Timer expires
STANDBY MODE
(~100 mA)
-WAKE ON TIMER OR IGNITION
GPIO1 wake up signal (asserted)

2.5.19. nxs.timezone

Event	*nxs.timezone*	
Event Data		
Keys	**Values**	**Description**
timezone	<string>	
Direction	input	
Channel	nxs	
Notes	Received by the application when subscribed to and upon change. This event contains the current system time zone. The example below assumes the Storyboard application is set up to call Lua function **tz_receive** when the *nxs.timezone* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	function **tz_receive**(mapargs)
	    local ev = mapargs.**context_event_data**
	    local tz = ev[“timezone”]
	
	    print(“Time zone: ”**..**tz)
	end	

2.5.20. nxs.wakeup_reason

Event	*nxs.wakeup_reason*	
Event Data		
Keys	**Values**	**Description**
wakeup_reason	0x0000 – Wake aborted 0x0001 – GPIO6 0x0002 – GPIO2 0x0004 – GPIO3 0x0008 – GPIO4 0x0010 – GPIO5 0x0020 – GPIO1 0x0040 – GPIO7 0x0080 – GPIO8 0x0200 – TOUCH 0x0400 – CAN1_RX 0x0800 – CAN2_RX	A 16-bit value with each bit representing the wakeup source.
Direction	input	
Channel	nxs	
Notes	Received by the application in response to being woken up. The example below assumes the Storyboard application is set up to call Lua function **wakeup_reason_receive** when the *nxs.wakeup_reason* event occurs. Event data is available through the *mapargs* argument.	

Lua Example

```lua
function wakeup_reason_receive(mapargs)
    local ev = mapargs.context_event_data
    local wakeup = ev[“wakeup_reason”]
    if (wakeups ~= nil) then
        if (bit32.band(wakeup,0x0001) ~= 0x0000) then
            print(“Woken up by GPIO6.”)
        end
    end
    -- Other Wakeup reasons left as an exercise
end
```

2.6. Persistent data storage (PDS)

2.6.1. pds.command

Event	*pds.command*	
Event Data		
Keys	**Values**	**Description**
command	“save” or “restore”	Desired command for PDS.
key	“string”	The key you wish to save or restore from.
value	value	If saving, the value you wish to save to PDS. If restoring, the default value to use if the data does not exist in PDS. Can be any valid Crank data type.
Direction	output	
Channel	pds	
Notes	Sent by the application to issue a command to the PDS module. “save” command: Saves the value in the key *value* into PDS under the key *key*. “restore” command: Retrieves a value from PDS using the key *key*. If the key does not exist in PDS, it will be created and saved with the value provided in *value*. A **pds.response** event containing the restored key and value will be sent in either case. Note that flooding PDS with **pds.command** requests may fill up its input queue or the Crank application’s queue with responses. We recommend limiting PDS commands to bursts of 100 requests of 32-bit sized data at a time. If the keys requested are typically larger than 32-bits, the burst size should be reduced accordingly. The example below shows an attempt to restore a saved backlight value from key “backlight”.	

Lua Example	**function restore_values**(mapargs)
	    local data = {}
	    data[“command”] = “restore”
	    data[“key”] = “backlight”
	    data[“value”] = 60 -- Provide a default level
	    gre.send_event_data(“pds.command”,
	                “1s0 command 1s0 key 4u1 value”, data, “pds”)
	end	

2.6.2. pds.response

Event	*pds.response*	
Event Data		
Keys	**Values**	**Description**
command	“string”	The command this response is meant for. This tells you if the response is the result of a save or restore command.
key	“string”	The key that is being returned
value	value	The restored value as read from PDS.
Direction	input	
Channel	pds	
Notes	Received by the application in response to a pds.command “restore” command. The example below assumes the Storyboard application is set up to call Lua function **pds_response** when the *pds.response* event occurs. Event data is available through the *mapargs* argument. The example demonstrates the result of restoring a PDS key of “backlight”.	

Lua Example	function pds_response(mapargs)
	  local ev = mapargs.context_event_data
	  if (ev[“key”] ~= nil and ev[“value”] ~= nil
	    and ev[“command”] ~= nil) then
	   if (ev[“key”] == “backlight”) then
	    if (ev[“command”] == “save”) then
	     print(“Saved backlight value of ”
	        ..tostring(ev[“value”]))
	  elseif (ev[“command”] == “restore”) then
	     print(“Restored backlight value of ”
	        ..tostring(ev[“value”]))
	     g_backlight = ev[“value”]
	   end
	  end
	 end
	end	

2.7. Auto-installer channel events

Autoinstaller is a DCP-based program that watches for notification of updates (via USB, CAN or other mechanisms) and allows the installation of updated software on the unit. It uses DCP to provide feedback to a user application if desired.

For USB device configurations, autoinstaller updates consist of getting a package file into the right location (/tmp) with the proper name (by default, usb_archive.file).

For USB host configurations, autoinstaller publishes a list of files that match a mask set in the autoinstaller XML configuration file via the *ai.available_releases* event. It then expects to be notified of a selected file via the *ai.selected_release* event.

2.7.1. ai.set_option

Event	*ai.set_option*	
Event Data		
Keys	**Values**	**Description**
publish_all_filename	<0. . 1>	Set to 1 to enable or set to 0 to disable the option to publish the filenames of the files that do not match the auto install filename format (filename mask).
Direction	output	
Channel	ai	
Notes	Sent by the application to the autoinstaller option **publish_all_filename**. By default, this option is disabled. If enabled, the autoinstaller will publish the filename of all the files found on the USB stick. In addition, if the USB stick doesn't contain any files that match the auto install filename format (mask), the USB stick will not be automatically unmounted. In this case, the application **must** send a "STATE_CANCEL" event to unmount the USB stick. The example below assumes the application calls the function set_ai_options when needed.	

Lua Example	`function set_ai_options(mapargs)`
	`local data = {}`
	`data[“publish_all_filename”] = 1`
	`gre.send_event_data(“ai.set_option”,`
	`“4u1 publish_all_filename”, data, “ai”)`
	`end`	

2.7.2. ai.available_releases

Event	*ai.available_releases*	
Event Data		
Keys	**Values**	**Description**
count	<0. . n>	A value from 0 to n representing the number of files found on the USB stick that match the autoinstaller file mask.
filenameX	<string>	A string representing the filename associated with the count value X. Note that the key *count* value X, is appended to the key *filename*.
unmatched_count	<0. . n>	Shown if **ai.set_option** is enabled. A value from 0 to n representing the number of files found on the USB stick that do not match the auto install filename format (mask)
unmatched_FilenameX	<string>	Shown if **ai.set_option** is enabled. A string containing the filename of the file found on the USB stick that does not match the auto install filename format (mask) where X is from 0 to (unmatchedCount – 1).
Direction	input	
Channel	ai	
Notes	Received by the application when subscribed to. Applications can use this to display a filename menu when configured for USB host for packager updates. The *ai.available_releases* event contains the key *count* for the number of filenames, and the key *filenameX* containing the string of the filename for the X count element. Note that the key *count* value X, is appended to the key *filename*. For example, assume there are 2 files available on the USB stick. The first file is FILEABC and the second is FILEXYZ. The key *count* would be 2. The key *filename0* would contain the string FILEABC and the key *filename1* would contain the string FILEXYZ. If the autoinstall option “**publish_all_filename**” in the *ai.set_option* event is enabled, the ai.available_releases event will also contain a nonzero unmatchedCount value and the unmatchedFileNameX string. The USB stick must be inserted after the ai.set_option, publish_all_filenames is enabled. If **publish_all_filename** is disabled, or if the USB stick is inserted before the publish_all_filename is enabled, then unmatchedCount will be zero. Note the mount path for the USB stick on the PHD is “/media/sda1”. The example below assumes the Storyboard application is set up to call Lua function **pop_releases** when the *ai.available_releases* event occurs. Event data is available through the mapargs argument. The example also assumes that list is a global table that maintains a list of the packages available on the USB stick. The *pop_table* function shown is assumed to populate an on-screen control with the data from the list table. The variable *tcount* is a global that tracks the number of items in the file list table.	

Lua Example	function **pop_releases**(mapargs)
	local ev = mapargs.context_event_data
	local i
	if ev[“count”] ~= nil then
	tcount = ev[“count”]
	
	for i = 1,tcount do
	if ev[“fileName”..i-1] ~= nil then
	list[i] = ev[“fileName”..i-1]
	end
	end
	pop_table()
	end
	end	

2.7.3. ai.install_state

Event	*ai.install_state*	
Event Data		
Keys	**Values**	**Description**
ai.install_state	<0. . 12>	A value from 0 to 12 representing the current autoinstaller state.
Direction	input	
Channel	ai	
Notes	Received by the application to read the installer state information. Applications can use this to display status or progress on the application screen during installs. 0 = STATE_FILE_CHECK 1 = STATE_USB_FILE_DETECTED 2 = STATE_CAN_FILE_DETECTED 3 = STATE_INVALID_FILE 4 = STATE_INSTALLING 5 = STATE_INSTALL_SUCCESS 6 = STATE_INSTALL_FAILED 7 = STATE_USB_QUERY 8 = STATE_USB_MOUNT_FAIL 9 = STATE_USB_NO_FILES_FOUND 10 = STATE_USB_WAITING_FOR_FILE_SELECTION 11 = STATE_CANCEL 12 = STATE_FILE_SELECTED States 7 through 12 are particular to the USB host update method. States 1 and 2 are specific to USB device or CAN update methods. States 3 through 6 are common to all. If the USB stick does not contain any filenames matching either the autoinstaller filename format (mask), or if the **publish_all_filename** option in the *ai.set_option* event is not enabled, then the USB stick will be unmounted immediately. If the **publish_all_filename** option is enabled, the state will automatically change from STATE_USB_QUERY to STATE_USB_WAITING_FOR_FILE_SELECTION when only unmatched files are detected. While in the STATE_USB_WAITING_FOR_FILE_SELECTION state, the application must either send the STATE_CANCEL event to unmount the USB Stick, or the filename selected for installation with the *ai. selected release* event. Note the USB stick must be reinserted to start a new installer session after receiving the STATE_CANCEL request. The example below assumes the Storyboard application is set up to call Lua function **ai_state** when the *ai.install_state* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	function ai_state(mapargs)
	  local ev = mapargs.context_event_data
	  local state = ev[“ai_install_state”]
	  if (state ~= nil) then
	    print(“Installing, state is “.. state)
	  end
	end	

2.7.4. ai.selected_release

Event	*ai.selected_release*	
Event Data		
Keys	**Values**	**Description**
fileName	<string>	A file release name selected from a USB stick that the user has chosen to install.
Direction	output	
Channel	ai	
Notes	Sent by the application to inform autoinstaller which file from the available releases has been selected by the user. Autoinstaller will attempt to install the selected file. In the Lua example below, “fileName” is a global variable that contains a file selected by the application for installation.	

Lua Example	function send_release(mapargs)
	local data = {}
	
	if (fileName ~= nil and fileName ~= “ ”) then
	data[“fileName”] = fileName
	gre.send_event_data(“ai.selected_release”,
	“1s0 fileName”, data, “ai”)
	end
	end	

3. Enabling a Camera via Crank

For devices with camera capabilities (PHD50 and PHD70), we rely on Linux's gstreamer service in order to enable and disable the camera feed. Crank's Storyboard Engine supports integration with gstreamer, which means that the cameras can be manipulated by sending built-in Crank events.

3.1. Requirements for enabling the camera

The basic Crank actions to use in your Crank application are used for starting and stopping the video stream.

First, make sure that the area in which you wish to display the camera image is transparent in Crank. This is so that the video, which runs "lowest" in the vertical stacking of Crank elements, is visible "through" the Crank application. Make sure any background images show up as transparent in that area rather than any type of image or solid colour background. Any images or text which are displayed over the camera area (and not hidden or set to an alpha of 0) will display over the camera feed. Transparent areas show up in Crank Storyboard Designer with a checkboard pattern, as demonstrated below.

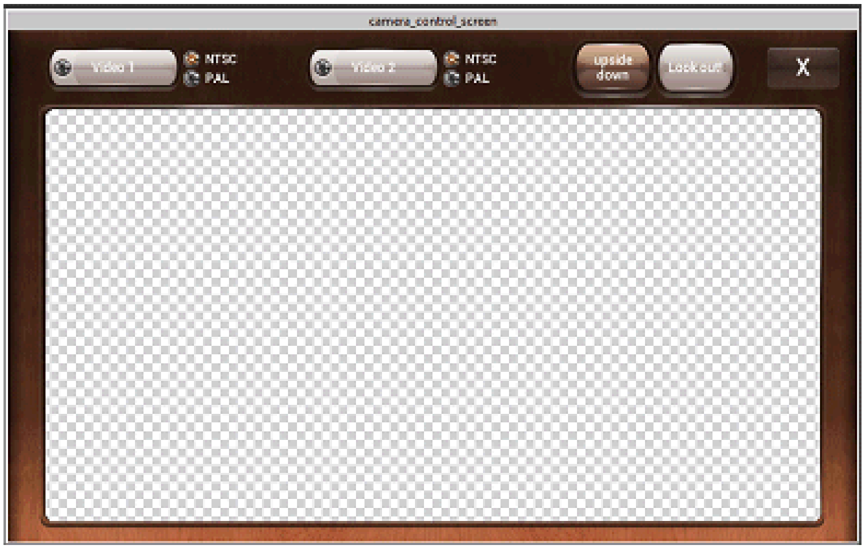


Figure 3.1. Sample camera layout

Next, using Crank's "Add" menu item, add a render extension to your project. This render extension should be of type "external". Give its Object Path parameter a unique name (e.g. "/video"). Set the size of the extension to your desired video size. This extension will create a solid pink rectangle within Crank. This pink rectangle will not appear when running. You can use this as a visual reminder of where your camera will display, but you must still set axis-left, axis-top, disp-width and disp-height in the extra_data field to place and size your camera view. The following image shows the same camera layout as above but with the external render object added in. Note that behind the pink rectangle, Crank is still transparent.

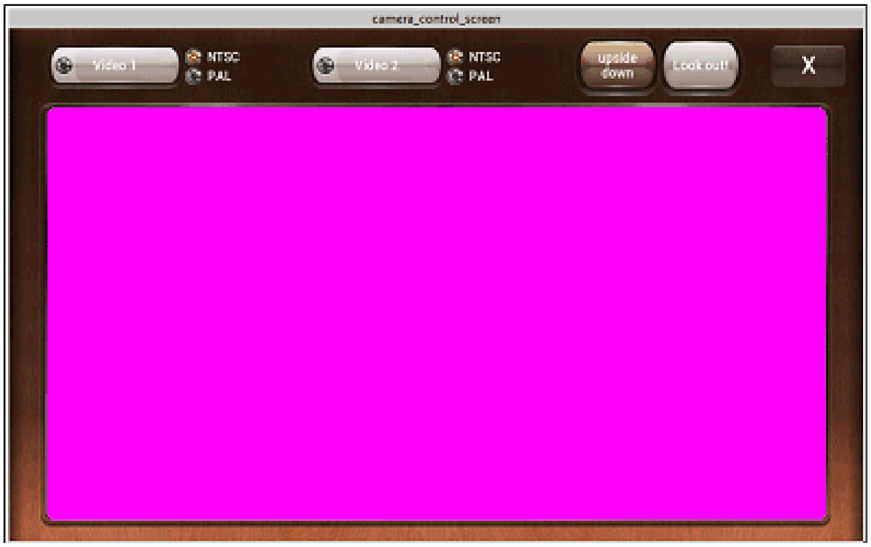


Figure 3.2. Camera layout with external render extension

For the PHD50 only, also add a Crank "fill" render extension over top, with its color set to #FF00FF. This will also appear as a pink box. The size of this pink fill object controls how much of the video is visible in Crank. Any Crank objects placed over top of this pink fill will partially obscure the video.

Starting a new video stream requires sending of the *gra.media.new.video* action. This action must be initiated when the application wishes to start video.

This action contains a number of built-in parameters which must be set as shown below.

Notes:
- For PHD50*, output_depth set to 2, this indicates 16-bit depth.*
- *the object_name parameter must match the external render extension's Object Path parameter (e.g. "/video").*
- media.sbat *must be present in the Crank project's templates folder.*

channel_name	**video1**
media_name	
volume	**0**
update_interval	**1000**
emit_time_events	**0**
external_buffer_name	**video**
object_name	**/video**
output_width	**0**
output_height	**0**
output_depth	**2**

For the PHD50, the camera input is always used full-size. The PHD50 extra_data value should be left blank. Whether the camera type is PAL or NTSC does not matter, the PHD50 will auto-detect the camera type. The camera must be powered at start up for auto-detect to work.

The *extra_data* parameter differs from PHD70 to PHD50 as the pipeline to construct the camera feed is different.
For the PHD70, match the width and height to the output_width and output_height you specified earlier. These should both match the size you used for the external render object in the Crank GUI.

Notes:
- the extra_data parameter should be all on one line. Substitute spaces for the linebreaks shown for that item.
- the extra_data parameter contains the Gstreamer pipeline and this can be setup to show different cameras or enable different options
- ***For PHD70****, output_width and output_height values can be zero. Instead, disp-width and disp-height in the extra_data parameter are used to control the output size of the video on the screen.*

PHD70 extra_data value:
"pipeline:imxv4l2src device=/dev/video0 capture-mode=0 ! mfw_isink rotation=X axis-top=102 axis-left=35 disp-width=546 disp-height=364 disp-deinterlace=X " Note the space after X, before the closing quotation mark.
Valid options for the "rotation" setting are:
0: NONE
1: VERTICAL FLIP
2: HORIZONTAL FLIP
3: ROTATE 180
4: ROTATE 90 RIGHT
5: ROTATE 90 RIGHT VFLIP
6: ROTATE 90 RIGHT HFLIP
7: ROTATE 90 LEFT

Valid options for the "capture-mode" setting are:
0: NTSC Format camera
1: PAL Format camera

Valid options for the "disp-deinterlace" setting are:
0: Disable deinterlacing
1: Enable deinterlacing

- Camera input selection: device=/dev/video0 declares video input #1 and device=/dev/video1 declares video input #2.

To halt a video stream, the *gra.media.stop* action must be sent. The *channel_name* specified must match the channel_name for the *gra.media.new_media* stream you wish to stop.

channel_name	**video1**
emit_state_event	**0**

3.1.1. Camera Settings

The PHD50 camera implementation includes a component that takes advantage of features of the Vybrid chip to allow reading and setting of key camera controls via the gre.media.settings event.
This event can be both sent by the user's Crank application to set or request the camera controls, and is also received by the user's Crank application in reply.

gre.media.settings **Event**	***gre.media.settings***	
Event data:		
Keys	**Values**	**Description**
channel	<IO channel>	Storyboard IO channel to receive these events. You must provide this parameter so that the media settings component knows where to send a reply. *Channel* needs to be sent only the first time you communicate with the media settings component. We recommend an initial "query" of the settings to provide the channel name.

action_result	<0..1>	Tells the media settings component if you wish to set or retrieve media settings. 0 = set, 1 = get. Crank 4s1 value.
contrast	<1..255>	Provides the camera contrast level to set. For output, only applicable if *action_result* is 0. Crank 4s1 value.
hue	<-128..+127>	Provides the camera hue level to set. Only applicable if *action_result* is 0. Crank 4s1 value.
saturation	<0..255>	Provides the camera saturation level to set. Only applicable if *action_result* is 0. Crank 4s1 value.
brightness	<-128..+127>	Provides the camera brightness level to set. Only applicable if *action_result* is 0. Crank 4s1 value.
mirror_h	<0..1>	Enables or disables camera horizontal mirroring. 0 = disable, 1 = enable. Only applicable if *action_result* is 0. Crank 4s1 value.
Direction	Output	
Channel	com.crank.media_backend	
Notes	Sent by the application to set or query the values of the PHD50 media settings. The first example below assumes the Storyboard application is set up to call Lua function **gre_media_query** on startup. Substitute your application's IO channel name for *my_camera_app.gapp.* The second example below assumes that the Crank application has called the sat_adjust function with the parameter *value* representing a new desired saturation value. *Channel* is not sent in the second example, as it assumes that has already been communicated via a call to gre_media_query.	

Lua example

```
-- This function queries the media settings.
function gre_media_query()
    local data = {}
  data["action_result"] = 0
  data["channel"] = "my_camera_app.gapp"
  print("gre_media_query: querying settings")
  gre.send_event_data("gre.media.settings", "4s1 action_result 1s0
  channel", data, "com.crank.media_backend")
end

-- This function sets saturation to the provided value.
function sat_adjust(mapargs)
  local value = tonumber(mapargs.value)
  local data = {}
  data["action_result"] = 1
  data["saturatio   n"] = value
  gre.send_event_data("gre.media.settings", "4s1 action_result 4s1
saturation", data, "com.crank.media_backend")
end
```

3.2. Contents of media.sbat

If you do not have media.sbat in your Crank Storyboard Suite “Samples” directory, you should be able to cut and paste the following into a text editor such as Notepad and save it to your application’s template folder as “media.sbat”. Use Notepad or a similar text editor, not a word processor like Word or Wordpad.

```
<actiontemplates>
       <template name=”gra.media.new.audio”>
    <arguments>
      <element name=”channel_name” type=”string” />
        <element name=”media_name” type=”string” />
        <element name=”volume” type=”integer” />
        <element name=”update_interval” type=”integer” />
        <element name=”emit_time_events” type=”integer” />
        <element name=”extra_data” type=”string” />
    </arguments>
  </template>
  <template name=”gra.media.new.video”>
    <arguments>
      <element name=”channel_name” type=”string” />
        <element name=”media_name” type=”string” />
        <element name=”volume” type=”integer” />
        <element name=”update_interval” type=”integer” />
        <element name=”emit_time_events” type=”integer” />
        <element name=”external_buffer_name” type=”string” />
        <element name=”object_name” type=”string” />
        <element name=”output_width” type=”integer” />
        <element name=”output_height” type=”integer” />
        <element name=”output_depth” type=”integer” />
        <element name=”extra_data” type=”string” />
    </arguments>
  </template>
  <template name=”gra.media.volume”>
    <arguments>
      <element name=”channel_name” type=”string” />
        <element name=”volume” type=”integer” />
        <element name=”emit_volume_event” type=”integer” />
        </arguments>
    </template>
    <template name=”gra.media.seek”>
        <arguments>
            <element name=”channel_name” type=”string” />
                <element name=”seek_num” type=”integer” />
                <element name=”emit_state_event” type=”integer” />
        </arguments>
    </template>
    <template name=”gra.media.stop”>
        <arguments>
            <element name=”channel_name” type=”string” />
                <element name=”emit_state_event” type=”integer” />
        </arguments>
    </template>
    <template name=”gra.media.playpause”>
        <arguments>
            <element name=”channel_name” type=”string” />
                <element name=”emit_state_event” type=”integer” />
        </arguments>
    </template>
    <template name=”gra.media.connect”>
    </template>
</actiontemplates>
```

4. RTC and time zone support

For devices with RTC capabilities (PHD70 standard), we rely on Linux's time zone database support to handle local clocks and daylight savings changes.

There are two clocks in an RTC-capable PHD70 device. There is the hardware RTC and there is the Linux system time.

Time zones are also supported and the time zone can be set or retrieved via API events. The device ships configured for the UTC time zone. This is effectively the Greenwich Mean Time zone, but the UTC time zone never shifts for daylight savings.

To set a different time zone, you would send the nxs.set_timezone event along with one of the accepted time zone strings (see Appendix C).

When you send the nxs.set_rtc event, the system assumes that you are specifying a local time that you wish to set, along with the desired time zone. It sets the time you specify as the system time and then adjusts the time to UTC and writes it into the RTC. If you issue a nxs.request_rtc event, you will receive the RTC value expressed in local time. Ideally though, you should use the Lua os.date() function to retrieve the system time, as it is set in lockstep with the RTC and always expressed relative to your set time zone.

The RTC is set in UTC so that the RTC time does not change during daylight savings time. If the RTC is correct, you can change time zones at any time, whether in or out of daylight savings and the system will automatically account for local daylight savings conditions.

The following list of 37 zones are the most commonly-used ones, for the full list, see Appendix C.

4.1. Most commonly used time zones

CET	Iceland	Poland
CST6CDT	Iran	Portugal
Cuba	Israel	ROC
EET	Jamaica	ROK
EST5EDT	Japan	Singapore
Egypt	Kwajalein	Turkey
Eire	Libya	UTC
GB	MST7MDT	Universal
GB-Eire	NZ	W-SU
GMT	NZ-CHAT	WET
Greenwich	Navajo	Zulu
HST	PRC	
Hongkong	PST8PDT	

5. Crank Storyboard Best Practices

5.1. Introduction

When developing and simulating a Crank Storyboard application on a Windows host, the simulator is actually running the Crank Storyboard engine. As a result, simulated applications may have much more resources available to them (RAM, processor speed, graphics acceleration) than on the target platforms.

As a result, application design choices may perform fine under the simulator but perform poorly on the target platforms.
In order to reduce this discrepancy, there are certain design guidelines to consider during development.

5.2. Crank Storyboard design guidelines

5.2.1. Application color depth

Make sure to choose an appropriate color depth for your application. This can be specified when creating a new application in Crank Storyboard Designer. It can also be changed on an existing application by selecting the Application from the Application Model window and then modifying the properties.

The best color depth depends on the target platform.
For PHD28 and PHD50, 16-bit (565) color depth must be used.
For PHD70, 32-bit (8888) color depth must be used.

5.2.2. Image color depth

When adding images to the user interface it is always preferable to create them in the desired color depth. If the application will be running in 16-bit color then the most efficient image to render will be a 16-bit image. If alpha blending/transparency is not required when this image is rendered then it is advisable to create images in the application color depth or at least remove the alpha channel in the image. If alpha blending is desired for an image, it should be left as a 32-bit image.

5.2.3. Image rotation and scaling

Image rotation and scaling can cause performance issues, as they require extra processing and memory allocations to perform these operations. If an image does not require real-time scaling while the application is running, then it should be resized externally and used at the target resolution. For example, if you have a 64x64 sized telltale control in the application, use a 64x64 image. Likewise, if an image does not require real-time rotation while the application is running, but is simply loaded at an angle, rotate it outside of Storyboard Designer so that it can be loaded without the need to rotate it inside Storyboard Designer. Also keep in mind the maximum size of the target displays, as both the resolution and the physical display size may dictate what size of images to use.

The PHD70 is a 7” 800x480 screen. This means there are 133 pixels per inch, or in other words, an image that is 48 pixels wide will take up less than 3/8 of an inch on the screen.

The PHD50 is a 5” 800x480 screen. This translates to 188 pixels per inch, so, an image that is 48 pixels wide will take up about ¼ of an inch on the screen. The PHD28 is a 2.8” 240x320 screen. This means there are 141 pixels per inch, or in other words, an image that is 48 pixels wide will take up less than 3/8 of an inch on the screen—however, for the PHD28, this would represent 15 – 20% of the screen width, depending on whether it’s used in portrait or landscape orientation.

5.2.4. Image cropping and positioning

In order to reduce unnecessary processing, make sure to create all images with as tight a border as possible. For example, do not create a large full-screen image composed of transparency with a small 48x48 icon in the middle, as this will result in waste of memory and processing power when the image is loaded, even if it’s not being rotated.

In general, the dimensions of the image file should be very close to the dimensions of the visible portion of the image. For images that will need alpha-blending with other elements, leave one row of absolutely blank pixels around its perimeter. Most image tools like GIMP or Photoshop will do this when using the autocrop or trim feature.

For any images that will be rotated in real-time, make sure the original source image is drawn at 0, 90, 180 or 270 degrees. Using an arbitrary angle in the original source image will result in more empty space around the image.

To picture this, please see the two images below. This needle will be rotated during application use, so one might assume that bringing it in at an angle doesn’t matter. However, the number of transparent pixels that have to be alpha-blended in each case is quite different. The non-rotated image is 42x119 pixels (4,998 pixels total), but the source image that is pre-rotated is 114x114 pixels (12,996 pixels total). Note that the images have had a black background added to them simply to illustrate the extent of the transparent pixels.

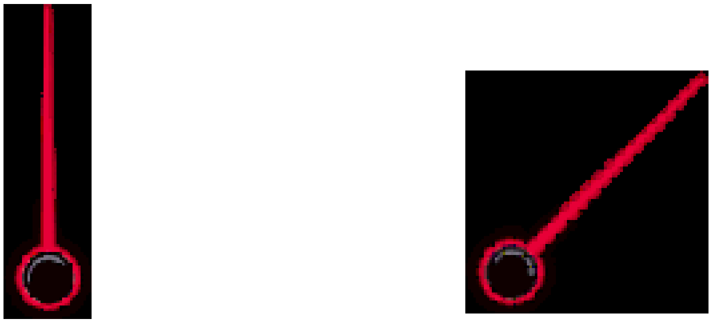


Figure 5.1. Needle image example

5.2.5. Image transparency

If an image does not require transparency then create it opaque, i.e. with the alpha channel removed. When this image is loaded by Storyboard Engine, it converts it to the color depth of the application, which allows Storyboard Engine to optimize its rendering to the framebuffer.

Further improvements may also be made by taking icons and screen elements that do not change and flattening them with the background color or image to create an opaque image.

Most popular images tool like GIMP or Photoshop support removing the alpha channel from an image. To do this, load the image into the tool and choose the option to Flatten the image. This will remove the transparency from the image. In most cases, the image transparency will be replaced by the active foreground color. You could then change it to an appropriate color for your background, or even combine it directly with the background image first, and then flatten both together.

5.2.6. Fill or circle control replacements for graphics

Any graphics that are simple box or circle shapes of a single color should not be used as images. Instead, remove the images from the project and replace them with Crank fill controls. This will reduce both the size of the application and its memory usage while running.

5.2.7. Polygon replacement for graphics

In some cases, some images could be replaced by Crank's polygon control. With the polygon control, shapes can be drawn by specifying drawing points.

Even images that must be rotated in real-time can be replaced in this way.

The images below show screenshots from two slightly different Crank applications. One screenshot is from an application using images for the four needles on the screen and the other uses Crank's polygon control for each needle.

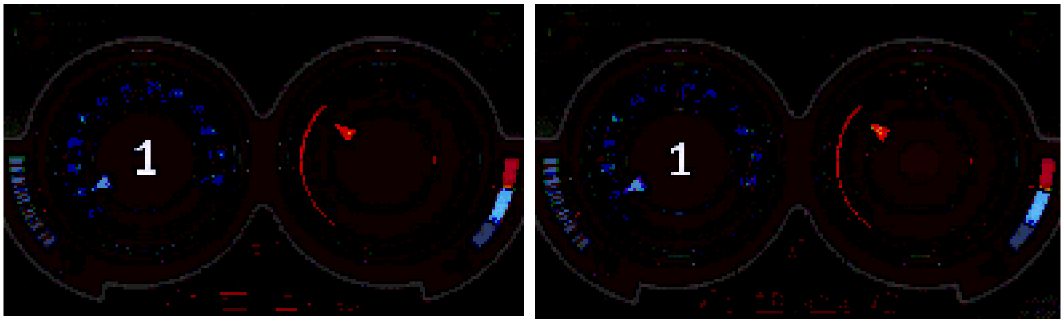


Figure 5.2. The image on the left uses the polygon controls for the four needles.

Note:

Make sure to close the polygon shape when intending to use it for rotation. This can be accomplished by repeating the initial x,y point of the polygon as the final point.

For example, here's a polygon defined with points (x,y) 190:200 205:200 200:20. It looks okay on the left screen. But when rotated, as shown on the right, it goes out of bounds.

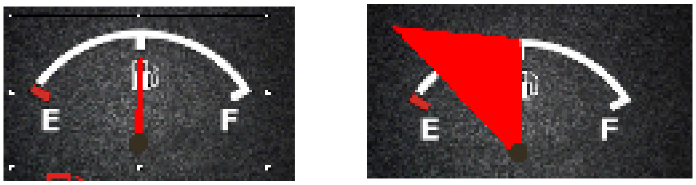


Figure 5.3. Polygon out of bounds.

By defining the polygon with points (x,y) 190:200 205:200 200:20 190:200. It made the problem go away.

5.2.8. Layer alpha blending

Modifying an entire layer's alpha property to make it visible or invisible is a very expensive operation and should be avoided.
It is better to have each image on the layer tied to a single alpha variable. That way, when the variable is modified, all those elements will adjust their alpha. This is less demanding on the device than blending the whole layer.

If using alpha to fade from one layer to another layer on a screen, it is more efficient to create an additional screen and use the "Screen Fade" action to fade between the screens.

For example, for a logo screen that fades in, use an initial screen with a background color and a second screen with the logo and background color. The initial screen can transition to the logo screen using screen fade and it will have the same appearance as using the alpha on the logo and fading it in.

5.2.9. Animation rate

The system may not be able to keep up with multiple animations with high frames per second rates. If this happens, Storyboard Engine will drop frames, doing extra work to do so. The end result may be an animation with even lower frames per second than could have been achieved by just choosing an appropriate frames per second. An animation rate of 14 frames per second will look good enough on most simple animations, though the results may vary depending on the target of the animation, the length of the animation, and the over- or under-laying screen components.

Different animation frame rates should be tested on the target hardware to strike a balance between animation appearance and performance.

6. Software Licensing Agreement

6.1. Licensing

The software running on the PHD devices makes use of code built using various Open Source Public Licenses as noted below.

All user-facing application programs must make mention of the underlying use of GNU code accessible through a menu somewhere in the program and the accompanying full text of the GNU General Public License version 2.0 should be available through a further screen. This does not mean that the application program must be released under the GPL. Rather, it must make mention that GPL code is in use on the device.

All Parker-generated code is copyright Parker-Hannifin Corporation, all rights reserved. No part of this Parker code may be reproduced, published, or distributed in any form or by any means (electronically, mechanically, photocopying, recording or otherwise), or stored in a database or retrieval system, without the prior written permission of Parker-Hannifin Corporation in each instance.

As per the GNU General Public License, version 2, section 3b, Parker offers that for a period of 3 years, the source code to the GPL items that have been modified by Parker (the kernel and U-Boot) can be provided when asked.

6.1.1. BSD COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. “Contributor” means each individual or entity that creates or contributes to the creation of Modifications.

1.2. “Contributor Version” means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. “Covered Software” means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. “Executable” means the Covered Software in any form other than Source Code.

1.5. “Initial Developer” means the individual or entity that first makes Original Software available under this License.

1.6. “Larger Work” means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. “License” means this document.

1.8. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. “Modifications” means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. “Original Software” means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. “Patent Claims” means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. “Source Code” means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. “You” (or “Your”) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, “You” includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw

Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" (as that term is defined at 48 C.F.R. ¤ 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

6.1.2. Dropbear

Dropbear contains a number of components from different sources, hence there are a few licenses and authors involved. All licenses are fairly non-restrictive.

The majority of code is written by Matt Johnston, under the license below.

Portions of the client-mode work are © 2004 Mihnea Stoenescu, under the same license:

Copyright © 2002-2013 Matt Johnston
Portions copyright © 2004 Mihnea Stoenescu
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS

BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LibTomCrypt and LibTomMath are written by Tom St Denis, and are Public Domain.

sshpty.c is taken from OpenSSH 3.5p1,
Copyright © 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland. All rights reserved

“As far as I am concerned, the code I have written for this software can be used freely for any purpose. Any derived versions of this software must be clearly marked as such, and if the derived work is incompatible with the protocol description in the RFC file, it must be called by a name other than “ssh” or “Secure Shell”. “
loginrec.c
loginrec.h
atomicio.h
atomicio.c
and strlcat() (included in util.c) are from OpenSSH 3.6.1p2, and are licensed under the 2 point BSD license.

loginrec is written primarily by Andre Lucas, atomicio.c by Theo de Raadt.

strlcat() is © Todd C. Miller

Import code in keyimport.c is modified from PuTTY’s import.c, licensed as follows:

PuTTY is copyright 1997-2003 Simon Tatham.

Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry,
Justin Bradford, and CORE SDI S.A.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.1.3. Expat

Copyright © 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright © 2001, 2002, 2003, 2004, 2005, 2006 Expat maintainers.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.1.4. Libffi

libffi - Copyright © 1996-2014 Anthony Green, Red Hat, Inc and others. See source files for details.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS

BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.1.5. Libjpeg

LEGAL ISSUES

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright © 1991-2014,
Thomas G. Lane, Guido Vollbeding.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:
(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

6.1.6. Libpng

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

This code is released under the libpng license.

libpng versions 1.2.6, August 15, 2004, through 1.6.17, March 26, 2015, are Copyright © 2004, 2006-2015 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.2.5 with the following individual added to the list of Contributing Authors:
Cosmin Truta

libpng versions 1.0.7, July 1, 2000, through 1.2.5 - October 3, 2002, are Copyright © 2000-2002 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:
Simon-Pierre Cadieux
Eric S. Raymond
Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright © 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:
Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997
Andreas Dilger, distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:
John Bowler
Kevin Bracey

Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright © 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS".

The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png_get_copyright" function is available, for convenient use in "about" boxes and the like:

printf("%s",png_get_copyright(NULL));

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson
glennrp at users.sourceforge.net
26-Mar-15

6.1.7. Libxml

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar license but with different Copyright notices) all the files are:

Copyright © 1998-2003 Daniel Veillard.
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

6.1.8. Orc

The majority of the source code and the collective work is subject to the following license:

Copyright 2002 - 2009 David A. Schleef <ds@schleef.org>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The source code implementing the Mersenne Twister algorithm is subject to the following license:

Copyright © 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.1.9. Zlib

zlib.h -- interface of the 'zlib' general purpose compression library version 1.2.8, April 28th, 2013

Copyright © 1995-2013 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files http://tools.ietf.org/html/rfc1950 (zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).

6.1.10. LGPL-2.1

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright © 1991, 1999 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

**GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the

Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright © <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library;
if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston,
MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

6.1.11. LGPL-3

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. < https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.
As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.
"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.
An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.
A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".
The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.
The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.
You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.
If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:
a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.
The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors,

or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:
a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.
You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:
a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
d) Do one of the following:
0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.
You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
6. Revised Versions of the GNU Lesser General Public License.
The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.
If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library

6.1.12. GPL2

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided

that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.> Copyright © <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © year
name of author
Gnomovision comes with ABSOLUTELY
NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions;
type `show c’ for details.

The hypothetical commands `show w’ and `show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w’ and `show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision’ (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

6.1.13. GPL3

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. https://fsf.org/ Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright © <year> <name of author>

This program is free software: you can redistribute it and/

or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright © <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <https://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <https://www.gnu.org/licenses/why-not-lgpl.html>.

7. Appendix A

7.1. J1939 Service

The PHD J1939 service exists to provide the system with access to one or more J1939 CAN buses. Inter-process communication is provided via DCP to permit access from other system services.

7.1.1. Features

- Send and receive of 8-byte messages
- Send and receive of long messages (up to and including 1785 byte payloads) using the following mechanisms.
  - Point to point - Long message exchange between precisely two connected parties.
  - Broadcast - Long message exchange to all interested parties within the broadcast domain of the local network segment.
- J1939 network management, including address claim concession and defense.
- Support for SOFT (software identification) message.
- Automatic handling of RQST (request) messages.

7.2. Architecture

The service is formed of two primary components:

- The J1939 library (1042616, libj1939). This connects to the CAN device drivers to provide send and receive communications with multiple CAN buses. The library provides a full abstraction of the lower level interfaces to present an external interface at the J1939 level in the protocol stack. Both C++ (native) and C (wrapper) interfaces are provided.
- The J1939 service (1042618, j1939d). This component is the daemon which provides the front end communication service and is the point of interaction for other communicating processes. Operation is driven by a configuration file, allowing per-application customization without recompilation of the core executable.

To promote code reuse, a common shared component is employed between the J1939 library and the J1939 service. This contains a library of common “helper” functions and common definitions to be used by both components.

The J1939 service also requires the use of an external library to allow the user configuration file to be read. The user configuration is specified in the form of a JavaScript Object Notation (JSON) file and so a library capable of reading this is necessary; this is currently implemented in libjson using JsonCpp which is released into the public domain under the MIT license with only a requirement to include the author’s copyright information.

7.3. Components

The J1939 service/library introduces several implementation-specific components.

7.3.1. J1939 Interface

A J1939 Interface represents a single physical CAN interface on the target device. Its operation is managed by a J1939 Interface Manager and is described in the configuration file by an interface descriptor. The interface manager acts as the single flow control point for all messages entering (received by) the interface, and all messages exiting (sent from) the interface.

7.3.2. J1939 Controller Application (CA)

A J1939 Controller Application is a logical component which resides within an interface. Multiple controller applications are permitted within a single interface. Each controller application owns a single J1939 source address and must have an associated NAME field. A controller application has a list of messages which it will receive, and a list of messages which it is permitted to transmit.

7.4. Usage

The J1939 service can be invoked from the command line. A number of switches can be specified to modify its operation, as shown in the table below.

Argument	Function
--help	Prints usage information to the console.
--ver	Prints version information to the console.
-c	Specifies the configuration file to use. If no file is specified, the service will use the file at /usr/local/parker/config/j1939.json. Example: **j1939 –c /root/testconfig.json** will cause the configuration file 'testconfig.json' located in /root/ to be used
-a	Specifies the DCP channel name to use for communication. If no channel name is specified, the service will fail to start. Example: **j1939d –a test** will cause the service to use 'test' as the channel name.
-v	Sets the debug verbosity level. A higher value results in more verbose debug messages. Example: **j1939d – v 5**
-d	Instructs the service to configure the CAN devices specified in the configuration file appropriately.

7.5. J1939 Configuration

The J1939 service is configured at runtime using a text-based configuration file. This allows the service to be customized without requiring recompilation. The file is defined in JavaScript Object Notation (JSON) format, a data-oriented hierarchical data description language based around JavaScript with simple translation to object-oriented systems.

7.5.1. File Structure

An overview of the J1939 configuration file structure is shown below.

- Configuration File Root
  - interfaces
    - deviceName
    - controllerApplications
      - address
      - name
        - arbAddressCapable
        - identityNumber
        - industryGroup
        - vehicleSystemInstance
        - vehicleSystem
        - function
        - functionInstance
        - ecuInstance
        - manufacturerCode
      - receiveDM1s
        - sa
        - spn
        - fmi
        - priority
        - user1
        - user2
      - transmitDM1s
        - id
        - spn
        - fmi
      - receiveMessages
        - pgn
        - ignoreSourceAddress
        - sourceAddress
        - notifyStale
        - staleTimeoutPeriod
        - length
        - rateLimit
        - parameters
          - name
          - startPosition
          - length
          - datatype
          - endian

- transmitMessages
    - pgn
    - destinationAddress
    - transmitMode
    - transmitRate
    - length
    - priority
    - parameters • name
        - startPosition
        - length
        - datatype
        - endian

7.5.2. File Components

7.5.3. File Root

Object	*File Root*	**Parent**	***None***
Attributes	**Type**	**Required**	**Description**
interfaces	Interface array	Yes	An array of physical CAN interface descriptions.
Notes	The top level of the document may only contain an array of interface descriptors. Each physical interface must only be described once.		

JSON Example	{
	“interfaces” : [
	{
	… interface description
	}
	]
	}		

7.5.4. Interfaces

Object	*Interface*	**Parent**	*Root*
Attributes	**Type**	**Required**	**Description**
deviceName	String	Yes	The name of the device as it is identified by the operating system.
controllerApplications	Controller application array	Yes	An array of controller applications which operate on the current interface.
Notes	An interface description object.		

JSON Example	{
	“interfaces” : [
	{
	“deviceName” : “can0”,
	“controllerApplications” : [
	… controller applications
	]
	}
	]
	}		

7.5.5. Controller Application

Object	*controllerApplication*	**Parent**	*interface*
Attributes	**Type**	**Required**	**Description**
address	Number	Yes	The J1939 source address of the controller application in decimal notation.
Name	Array	Yes	The J1939 NAME field. See section X for child attributes.
DM1	String	Yes	The fully-qualified path and filename to the DM1 config file for this controller application.
receiveMessages	Message array	Yes	An array of J1939 messages which should be received by this CA.
transmitMessages	Message array	Yes	An array of J1939 messages which should be transmitted by this CA.
Notes	A controller application description object.		
JSON Example	“controllerApplications” : [“address” : 72, “name” : [… J1939 NAME field attributes], “DM1” : “/root/app/config/dm1_ca40.json”, “receiveMessages” : [… receive messages], “transmitMessages” : [… transmit messages]]		

7.5.6. Message - receiveMessages

Object	*message*	**Parent**	*controllerApplication*
Attributes	**Type**	**Required**	**Description**
pgn	Number	Yes	The J1939 Parameter Group Number in decimal format. Valid values are 0 - 65535.
ignoreSourceAddress	Boolean	Yes	A Boolean value indicating whether the source address of the messages should be ignored. If set to 'true', *sourceAddress* property must be present.
sourceAddress	Number	No	The J1939 source address of the message. If *ignoreSourceAddress* is set to 'false', the message is filtered on source address as well as PGN.
notifyStale	Boolean	Yes	A Boolean value indicating whether the subscriber will be notified if the message becomes stale. Requires *staleTimeoutPeriod*. Only valid for receive type messages.
staleTimeoutPeriod	Number	No	Sets the stale timeout period for the message. The subscriber will be notified if the specified period of time elapses following the last valid receipt of the message. Required if *notifyStale* is set to 'true'. Specified in milliseconds.
length	Number	Yes	The length of the message in bytes.
rateLimit	Number	No	Sets the rate at which new values for the receive message are published. Setting the value to 0 or not specifying this attribute removes the rate limit. Specified in milliseconds.
ignoreDuplicate	Boolean	Yes	When set to true, an update to a message that does not change the contents of any SPNs will not cause an event to be emitted (for a received message) or a J1939 message to be sent over CAN (for a transmitted message).
parameters	Boolean array	Yes	An array of parameter objects which form the message data.
Notes	A message description object.		

JSON Example	{
	"pgn" : **54528**,
	"ignoreSourceAddress" : false,
	"sourceAddress" : **36**,
	"length" : **8**,
	"rateLimit" : **200**,
	"notifyStale" : false,
	
	"parameters" : [
	{
	… parameter definition
	}
	]
	}		

7.5.7. Message - transmitMessages

Object	*message*	Parent	*controllerApplication*
Attributes	**Type**	**Required**	**Description**
pgn	Number	Yes	The J1939 Parameter Group Number in decimal format. Valid values are 0 - 65535.
destinationAddress	Number	Yes	The J1939 destination address of the message.
length	Number	Yes	The length of the message in bytes.
priority	Number	Yes	Sets the priority of the message to be transmitted.
ignoreDuplicate	Boolean	Yes	When set to true, an update to a message that does not change the contents of any SPNs will not cause an event to be emitted (for a received message) or a J1939 message to be sent over CAN (for a transmitted message).
transmitMode	Enumeration	Yes, for transmit	Sets the mode by which a message will be transmitted. Only valid for transmit messages. Permitted values are: “periodic : The message will be sent at the interval defined in the period property. “periodicOnChange : The message will be sent at the interval defined in the period property in addition to being sent when the message data changes. “onRequest : The message is only sent upon request. “onChange : The message is sent when the message data is changed.
transmitRate	Number	Yes, for transmit	The rate at which the transmit message is sent, should *transmitMode* be set to an appropriate mode. Specified as a period, in milliseconds.
parameters	Boolean array	Yes	An array of parameter objects which form the message data.
Notes	A message description object.		

JSON Example	{
	“pgn” : **54528**,
	“ignoreSourceAddress” : false,
	“sourceAddress” : **36**,
	“length” : **8**,
	“rateLimit” : **200**,
	“notifyStale” : false,
	“parameters” : [
	{
	… parameter definition
	}
	]
	}		

7.5.8. Parameter

Object	*parameter*	**Parent**	*message*
Attributes	**Type**	**Required**	**Description**
Name	Number	Yes	The identifying name of the parameter. A special paremeter name of “multiplexor” is also supported. If the first paremeter for a message is named “mulitplexor” then it definies the bits, starting at 0, where the multiplexor ID is stored. Subsequent parameters must have the “mux’ keyword specified to indicate which variant of the message it is associated with.
startPosition	Number	Yes	The bit position (0 indexed) at which the parameter starts. The start position and length combination must be within the bounds of the parent message length. The special multiplexor must have a start position of 0.
length	Number	Yes	The length of the parameter, in bits. For raw and string types, this must be byte aligned (length % 8 = 0). For integer types, this must be 0 < length < 64. The special multiplexor must have a length of 8.
endian	Enumeration	Yes	The endianness of the parameter. Must be one of the following: “big “little
dataType	Enumeration	Yes	The data type of the parameter. Must be one of the following: “integer : The parameter represents an unsigned single primitive type whose size is described by the rule 2(3,6). “raw : The parameter represents a consecutive series (array) of bytes. “string : The parameter represents an ASCII string with or without a NULL terminator.
Notes	A parameter description object.		

JSON Example	“parameters” : [
	{
	“name” : “spn190”,
	“startPosition” : 24,
	“length” : 16,
	“endian” : “little”,
	“dataType” : “integer”
	}
	]
	In the multiplexor example below, this PGN is a multiplexed message that consists of parameters Device and Failure for mux 0, and parameters Function and Store for mux 1. We do not change the PGN definition, merely the parameters:		

JSON Example (Cont.)

```
{
"parameters":
    [
            {
"name":"multiplexor",
"startPosition":0,
"length":8,
"endian":"little",
"dataType":"integer"

},
{
"name":"Device",
"mux":0,
"startPosition":8,
"length":8,
"dataType":"integer","endian":"little"
 },
 {
"name":"Function","mux":1,
"startPosition":8,
"length":8,
"dataType":"integer","endian":"little"
},
{
"name":"Failure",
"mux":0,
"startPosition":16, "length":8,
"dataType":"integer","endian":"little"
},
{
"name":"Store",
"mux":1,
"startPosition":16, "length":16,
"dataType":"integer","endian":"little"
}
   ]
},
```

7.5.9. J1939 NAME

Object	*parameter*	**Parent**	*message*
Attributes	**Type**	**Required**	**Description**
arbAddressCapable	Number	Yes	The address arbitration capable field. The J1939 service does not support address arbitration, so this should always be set to 0.
identityNumber	Number	Yes	The identity number field.
industryGroup	Number	Yes	The industry group field.
vehicleSystemInstance	Number	Yes	The vehicle system field.
vehicleSystem	Number	Yes	The vehicle system instance field.
function	Number	Yes	The function field
functionInstance	Number	Yes	The function instance field.
manufactureCode	Number	Yes	The manufacturer code field.
ecuInstance	Number	Yes	The ECU instance field.
Notes	A J1939 NAME field description.		
JSON Example	**“name” :** **{** **“arbAddressCapable”** : 0, **“identityNumber” :** 1244, **“industryGroup” :** 3, **“vehicleSystemInstance” :** 1, **“vehicleSystem” :** 0, **“function” :** 24, **“functionInstance” :** 0, **“ecuInstance” :** 1, **“manufacturerCode” :** 71 **}**		

7.6. DM1 File Structure

- Configuration File Root
  - receiveDM1s
    - sa
    - spn
    - fmi
    - priority
    - user1
    - user2
  - transmitDM1s
    - id
    - spn
    - fmi

7.7. DM1 File Components

7.7.1. File Root

Object	*File Root*	**Parent**	*none*
Attributes	**Type**	**Required**	**Description**
receiveDM1s	Receive DTC array	Yes	An array of fault codes and descriptions.
transmitDM1s	Transmit DTC array	Yes	An array of fault codes and identifiers.
Notes	The top level of the document must contain exactly one instance of the receiveDM1s and transmitDM1s sections. The array for each of these sections is allowed to be empty.		

JSON Example	{
	  "receiveDM1s" : [
	    {
	      … receive DTC
	    }
	  ],
	  "transmitDM1s" : [
	    {
	      … transmit DTC
	    }
	  ]
	}		

7.7.2. Receive DTC

Object	*receiveDTC*	**Parent**	*receiveDM1s*
Attributes	**Type**	**Required**	**Description**
sa	Number	Yes	The source address of the device emitting the DM1 message.
spn	Number	Yes	The spn field of a dtc code in the DM1 message (see SAE J1939-73 specification document for details).
fmi	Number	Yes	The fmi field of a dtc code in the DM1 message (see SAE J1939-73 specification document for details).
priority	Number	Yes	The importance of this dtc. 0 = highest, 31 = lowest.
user1	string	No	User-defined text.
user2	string	No	User-defined text.
Notes	The top level of the document must contain exactly one instance of the receiveDM1s and transmitDM1s sections. The array for each of these sections is allowed to be empty. Only Conversion Method 4, CM bit set to 0, is supported in parsing the DTC parameters.		

JSON Example	{
	“sa” : 0
	“spn” : 1076
	“fmi” : 5
	“priority” : 4
	“user1” : “S1076F5”
	“user2” : “Engine Problem”
	}		

7.7.3. Transmit DTC

Object	*transmitDTC*	**Parent**	*transmitDM1s*
Attributes	**Type**	**Required**	**Description**
id	Number	Yes	The unique identifier for this fault code.
spn	Number	Yes	The spn field of a dtc code in the DM1 message (see SAE J1939-73 specification document for details).
fmi	Number	Yes	The fmi field of a dtc code in the DM1 message (see SAE J1939-73 specification document for details).
Notes	The combination of spn and fmi must be unique. The id field should be a short unique text for this error.		
JSON Example	**{** “id” : **PHD1-7** “spn” : **1** “fmi” : **7** **}**		

8. External Interface

Inter-process communication is provided by a common DCP layer employed across the Parker middleware. For Crank-based display systems, the underpinning layer is Storyboard I/O. Inter-process communication hinges around two major components:

- Publisher - a publisher exposes data in the current processes' address space which enables it to be accessed from other processes
- Subscriber- A subscriber exposes data in teh address space of one process, allowing it to be accessed from another

A more in-depth description of DCP may be found in document 1042F26.

The J1939 service publishes and subscribes to data using strict, pre-defined paths. The following text provides examples of these. Note that the examples are provided in Luo form (for use within Crank Storyboard) and assume that the application channel name is set to 'j1939'. The path names still hold for other C/C++ based services and may be adjusted to suit.

8.1. Enabling J1939

Before any J1939 data can be sent or received, an external application must enable J1939 using the following mechanism.

8.1.1. j1939.canX.caZZ.setmode

Event	*j1939.canX.caZZ.setmode*	
Event Data		
Keys	**Values**	**Description**
mode	<0. . 1>	0 = J1939 transmission and reception disabled. 1 = J1939 transmission and reception enabled.
Direction	output	
Channel	J1939	
Notes	Sent from the application to the J1939 service. The path is described generically where: "canX" is the name of the interface "caZZ" is the source address of the target CA. Note that all PGN and CA values must be specified in decimal notation (base 10). When *mode* is set to 1, the timers for all stale messages will be reset to avoid immediately flooding the application with "stale" events, followed shortly thereafter with "fresh" events. Whenever *mode* is set to 1, the interface will send out an address claim message.	
Lua Example	**function enable_J1939**(mapargs) **local** data = {} data["**mode**"] = 1 **gre.send_event_data**("**j1939.can0.ca40.setmode**", "4u1 mode", data, "j1939") **end**	

8.1.2. Receive subscriptions

An external application may obtain J1939 message data using the following mechanism.

8.1.3. j1939.canx.pgnYYYYY.[muxAAA][saZZZ]

Event	*j1939.canx.pgnYYYYY.[muxAAA][saZZZ]*	
Event Data		
Keys	**Values**	**Description**
parameters	<parameter data>	Each parameter in the message is accessible using the parameter name as the key.
rawCanId	,<0..1073741823>	Unparsed CAN identifier of the message
Direction	input	
Channel	J1939	
Notes	Sent by the J1939 service to other services which have subscribed to the event. The path is described generically where: • “canX” is the name of the interface • “pgnYYYYYY” is the PGN of interest • muxAAA represents the multiplexor ID of the message • “saZZZ” is the address of the sender The square brackets ([ ]) in the event name represent optional items. When using those optional items, the square brackets should be left off of the event name. For example, data for PGN 61444 from source address 0 on interface can0, the path would be: j1939.can0.pgn61444.sa0. Note that it is possible to exclude the ‘saZZ’ argument in order to receive PGN YYYYYY from any source address. Be aware however that this requires a corresponding message entry in the J1939 CAN configuration file structure where the *‘ignoreSourceAddress’* attribute has been set to ‘true’. The muxAAA value only needs to be provided for PGNs that are defined with a multiplexor parameter. The AAA values should be a 4u1 parameter field named "multiplexor" containing the AAA value. For each multiplexor ID defined in the configuration file, a corresponding event will be created. Note that all PGN, MUX and SA values must be specified in decimal notation (base 10). The rawCanId identifier is returned with every message in the raw, unparsed form. The application will need to parse the value as needed. Note that the example below assumes the application has been configured to call cb_update_rpm upon receipt of PGN 61444.	

Lua Example	function cb_update_rpm(mapargs)
	  local ev = mapargs.context_event_data
	  local data = {}
	  local rpm = ev[“spn190”] / 8
	  local rpm_rot = (rpm / 10000) * 49
	
	  data[“tach_exterior.pointer_tach_exterior.rot”] = rpm_rot
	
	  gre.set_data(data)
	end	

8.1.4. Transmit message data

The application is able to control the data transmitted by defined messages using the following mechanism.

8.1.5. j1939.canX.pgnYYYYYY.daZZ

Event	*j1939.canX.pgnYYYYYY.daZZ*	
Event Data		
Keys	**Values**	**Description**
parameter	<parameter data>	Each parameter in the message is accessible using the parameter name as the key.
Direction	output	
Channel	J1939	
Notes	Sent from the application to the J1939 service. The path is described generically where: "canX" is the name of the interface "pgnYYYYYY" is the PGN of interest "daZZ" is the address of the receiver. For example, data for PGN 61444 being sent to address 0 on interface can0, the path would be: j1939.can0.pgn61444.da0. Note that all PGN and DA values must be specified in decimal notation (base 10). Note that the example adjusts the testParameter parameter of PGN 65535. testParameter is an unsigned 32-bit value.	
Lua Example	function cb_update_65535(mapargs) local data = {} data["test"] = sendData gre.send_event_data("j1939.can1.pgn65535.da10", "4u1 test", data, "j1939") sendData = sendData + 1 end	

8.1.6. Send Message

The application is able to send pre-defined J1939 CAN messages using the following mechanism.

8.1.7. j1939.canX.caZZ.send

Event	*j1939.canX.caZZ.send*	
Event Data		
Keys	**Values**	**Description**
pgnList	<0. .65535>	An array of the PGNs to transmit.
Direction	output	
Channel	J1939	
Notes	Sent from the application to the J1939 service. The path is described generically where: • “canX” is the name of the interface • “caZZ” is the address of the receiver. For example for SA 0x00 on interface can0, the path would be: j1939.can0.sa0. Note that all PGN and SA values must be specified in decimal notation (base 10).	

Lua Example	function **cb_send_ffff**(mapargs)
	  local data = **{}**
	  data[“**pgnList**”] = 65535
	  gre.send_event_data(“**j1939.can1.sa72.send**”,
	        “**4u1 pgnList**”, data, “j1939”)
	end	

8.1.8. Receive Status Information

The application can obtain status information from an interface using the following mechanism.

8.1.9. j1939.canX.caZZ.status

Event	*j1939.canX.caZZ.status*	
Event Data		
Keys	**Values**	**Description**
sa	<0. .255>	The source address of the pgn whose state has changed. This key will report 254 if the receive message was defined with ignoreSourceAddress set to "true".
pgn	<0. .65535>	The PGN of the message
status	<0. .1>	0 = message has gone stale. 1 = message was stale, has now been received.
Direction	input	
Channel	J1939	
Notes	Sent by the J1939 service to other services which have subscribed to the event. The path is described generically where: • "canX" is the name of the interface • "caZZ" is the address of the CA defined in the json configuration file. For example, status information for CA 0x00 on interface can0, the path would be: j1939.can0.ca0.status. Note that all PGN and SA values must be specified in decimal notation (base 10).	
Lua Example		

8.1.10. Version Information

The application can publish its version information to the J1939 service using the following mechanism.

8.1.11. j1939.set_app_version_info

Event	*j1939.set_app_version_info*	
Event Data		
Keys	**Values**	**Description**
appVersionMajor	<ascii string>	The application major version number. Should be numeric for full compliance with Flashloader version information.
appVersionMinor	<ascii string>	The application minor version number. Should be numeric for full compliance with Flashloader version information.
appVersionBuildNumber	<ascii string>	The application build number. Should be numeric for full compliance with Flashloader version information.
appPartNumber	<ascii string>	The application part number. Should be numeric for full compliance with Flashloader version information.
appManufacturer	<ascii string>	The manufacturer of the application. Should be 5 characters or less for J1939 CI message compliance.
appModel	<ascii string>	The application model.
appUnitNumber	<ascii string>	The application unit number.
kernelPartNumber	<ascii string>	The kernel part number.
coproPartNumber	<ascii string>	The part number of the Co-processor software.
coproVersion	<ascii string>	The version string for the Co-processor software.
faultTablePartNumber	<ascii string>	The part number of the fault table.
faultTableVersion	<ascii string>	The version string for the fault table.
imagesPartNumber	<ascii string>	The part number of the images application.
imagesVersion	<ascii string>	The version string of the images application.
parametersPartNumber	<ascii string>	The part number for the parameters project.
parametersVersion	<ascii string>	The version string for the parameters project.
Direction	output	
Channel	J1939	
Notes		
Lua Example	function set_app_ver_info(mapargs) print(“Sending app version information...”) local data = {} data[“appVersionMajor”] = “2” data[“appVersionMinor”] = “48” data[“appVersionBuildNumber”] = “2401” data[“appPartNumber”] = “1042618” data[“appManufacturer”] = “PHC” data[“appModel”] = “J1939 daemon test” data[“appUnitNumber”] = “54” data[“kernelPartNumber”] = “1042605” gre.send_event_data(“j1939.set_app_version_info”, “1s0 appVersionMajor 1s0 appVersionMinor 1s0 appVersionBuildNumber 1s0 appPartNumber 1s0 appManufacturer 1s0 appModel 1s0 appUnitNumber 1s0 kernelPartNumber”, data, “j1939”) end	

8.1.12. DM1 Receive Information

The application can obtain DM1 information from the J1939 service using the following mechanism.

8.1.13. j1939.canX.dm1_dtc.caZZ

Event	*j1939.canX.dm1_dtc.caZZ*	
Event Data		
Keys	**Values**	**Description**
sa	<0. .255>	Address of the device that sent the DM1
spn	<0. .524287>	The SPN field
fmi	<0. .31>	The FMI field
dtc	<0. .16777215>	The combined value of the spn and fmi fields.
status	<0. .1>	1 = dtc is active, 0 = dtc inactive.
priority	<0. .31>	The importance of the dtc. 0 = highest, 31 = lowest.
oc	<0. .127>	The occurrence count of the dtc.
user1	<text string>	The “user1” string from the configuration file.
user2	<text string>	The “user2” string from the configuration file.
Direction	input	
Channel	J1939	
Notes	Sent by the J1939 service to other services which have subscribed to the event. The path is described generically where: “canX” is the name of the interface “caZZ” is the source address of the controller application that received the DM1. If the sa is set to 255, all DTC will be reviewed. For example, if the controller application is defined as using address 40 on can0, the event name would be path would be: j1939.can0.dm1_dtc.ca40. Note that the example assumes the application has been configured to call cb_update_faults upon receipt of j1939.can0.dm1_dtc.ca40.	

Lua Example	function cb_update_faults(mapargs)
	local ev = mapargs.context_event_data
	local dtc = ev[“dtc”]
	local oc = ev[“oc”]
	local status = ev[“status”]
	
	if (status == 1) then
	data[dtc] = oc
	else
	data[dtc] = nil
	end
	end	

8.1.14. j1939.canX.dm1_lamp_status.caZZ

Event	j1939.canX.dm1_lamp_status.caZZ	
Event Data		
Keys	Values	Description
awl	<0..3>	Amber Warning Light, 0 = off, 1 = on, 2 = reserved, 3 = no data
rsl	<0..3>	Red Safety Light, 0 = off, 1 = on, 2 = reserved, 3 = no data
mil	<0..3>	Malfunction Ind. Light, 0 = off, 1 = on, 2 = reserved, 3 = no data
protect	<0..3>	Engine Protect Light, 0 = off, 1 = on, 2 = reserved, 3 = no data
awl_fl	<0..3>	Amber Warning Light Flash, 0 = off, 1 = on, 2 = reserved, 3 = no data
rsl_fl	<0..3>	Red Safety Light Flash, 0 = off, 1 = on, 2 = reserved, 3 = no data
mil_fl	<0..3>	Malfunction Ind. Light Flash, 0 = off, 1 = on, 2 = reserved, 3 = no data
protect_fl	<0..3>	Engine Protect Light Flash, 0 = off, 1 = on, 2 = reserved, 3 = no data
sa	<0..255>	The source address of the pgn whose state has changed. This key will report 254 if the receive message was defined with ignoreSourceAddress set to "true".
Direction	input	
Channel	J1939	
Notes	Sent by the J1939 service to other services which have subscribed to the event. The path is described generically where: "canX" is the name of the interface "caZZ" is the source address of the controller application that received the DM1. For example, if the controller application is defined as using address 40 on can0, the event name would be path would be: j1939.can0.dm1_dtc.ca40. Note that the example assumes the application has been configured to call cb_update_lamps upon receipt of j1939.can0.dm1_lamp_status.ca40.	

Lua Example	function cb_update_lamps(mapargs)
	 local data = {}
	 local ev = mapargs.context_event_data local awl = ev["awl"]
	 local awl_fl = ev["awl_fl"] if (awl = 1) then
	  if (awl_fl = 3) then
	   print("Amber Warning Lamp ON")	

8.1.15. DM1 Transmit Information

The application is able to send DM1 information using the following mechanism.

8.1.16. j1939.canX.caZZ.alarm

Event	*j1939.canX.caZZ.alarm*	
Event Data		
Keys	**Values**	**Description**
id	<text string>	An identifier defined in the dm1 configuration json file.
status	<0. .1>	1 = fault is active, 0 = fault is inactive.
Direction	output	
Channel	J1939	
Notes	Sent to other services which have subscribed to the event. The path is described generically where: “canX” is the name of the interface “caZZ” is the source address of the controller application that received the DM1. For example, if the controller application is defined as using address 40 on can0, the event name would be path would be: j1939.can0.dm1_dtc.ca40. Note that the example assumes the application has been configured to call cb_update_faults upon receipt of j1939.can0.dm1_dtc.ca40.	
Lua Example	function **cb_set_fault**(mapargs) local data= {} data[“id”] = mapargs[“id”] data[“status”] = mapargs[“status”] gre.send_event_data(“j1939.can0.ca40.alarm”, “1s0 id 4u1 status”, data, “j1939”) end	

9. Appendix B

9.1. CANopen Service

The PHD CANopen service exists to provide the system with access to one or more CANopen CAN buses. Inter-process communication is provided via DCP to permit access from other system services.

Standard PHDs support J1939 only. Units must be special ordered to support CANopen.

9.2. CANopen file transfer support

9.2.1. Introduction

The CANopen stack includes a block transfer feature to allow files to the sent to or from the PHD. With this feature, one or more SDOs can be used to transfer files to and/or from the PHD (SDO block download and upload).
The SDO object's index must be in the manufacturer-specific range (0x2000 to 0x5FFF) or the device-specific range (0x6000 to 0x9FFF).
Upon successful reception of a file, the CANopen stack will signal the appropriate cos.rxNNNN event.

9.2.2. Feature constraints

- The SDOs used for file transfer must have their data type set to DOMAIN in the EDS file (i.e. DataType = 0x000F). They must also have a Filename parameter set to a valid file in the EDS file.
- The maximum supported file-size is 64kB.
- The PHD will abort a file transfer if the file is not accessible, too large or there is not enough space available at the destination.
- The PHD acts only as a CANopen slave for file transfers.
- The CANopen stack will not allow executable files to be transferred.
- The default destination path will be /usr/local/parker/data.

An example of an object in the EDS file that can receive a CDF file is shown belo using the object index 5000, but you can use any index from 2000 to AFFF.

```
[5000]
ParameterName=CDF Data
ObjectType=0x7
DataType=0x000F
AccessType=rww
DefaultValue=0
PDOMapping=0
```

After the CDF has been transferred to the PHD, the contents are available for reading by a Crank application

9.2.3. EDS file transfer example

Here are some sample EDS elements detailing file transfer support. This EDS file specifies 5 possible file transfer targets, defining them within SDO index 0x4444.

```
[4444]
ParameterName=File Transfer Test Objects
ObjectType=0x8
SubNumber=4
ObjFlags=0x0

[4444sub0]
ParameterName=Number of subindexes
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=3
PDOMapping=0
ObjFlags=0x0

[4444sub1]
ParameterName=File Transfer 1 read-write
ObjectType=0x7
DataType=0x000F
AccessType=rww
DefaultValue=0
PDOMapping=0
ObjFlags=0x0
Filename=fx_444401.txt
[4444sub2]
ParameterName=File Transfer 2 write-only
ObjectType=0x7
DataType=0x000F
AccessType=wo
DefaultValue=0
PDOMapping=0
LowLimit=0
HighLimit=100
ObjFlags=0x0
Filename=fx_444402.txt

[4444sub3]
ParameterName=File Transfer 3 read-only
ObjectType=0x7
DataType=0x000F
AccessType=ro
DefaultValue=0
PDOMapping=0
LowLimit=0
HighLimit=100
ObjFlags=0x0
Filename=fx_444403.txt

[4445]
ParameterName=File Transfer 4 no subs
ObjectType=0x7
DataType=0x000F
AccessType=rww
DefaultValue=0
PDOMapping=0
Filename=fx_444500.txt
```

9.3. CANopen channel events

9.3.1. cos.ext_od_access

<table>
<tr><td><b>Event</b></td><td colspan="2"><i>cos.ext_od_access</i></td></tr>
<tr><td colspan="3"><b>Event Data</b></td></tr>
<tr><td><b>Keys</b></td><td><b>Values</b></td><td><b>Description</b></td></tr>
<tr><td><i>access</i></td><td><0..65535></td><td>0 = read request from other node.
Non-zero = write value to other node.</td></tr>
<tr><td><i>index</i></td><td><0..65535></td><td>Index of the item in the external object dictionary</td></tr>
<tr><td><i>subindex</i></td><td><0..65535></td><td>Subindex of the item in the external object dictionary</td></tr>
<tr><td><i>value</i></td><td><0..4294967295></td><td>Value to write to the external object dictionary. Ignored for reads.</td></tr>
<tr><td><b>Direction</b></td><td colspan="2">output</td></tr>
<tr><td><b>Channel</b></td><td colspan="2">cos</td></tr>
<tr><td><b>Notes</b></td><td colspan="2">Sent by the application to read or write an external object dictionary entry by index and subindex.
The examples below assume the Storyboard application is set up to call the given Lua functions with arguments of <i>index</i> and <i>subindex</i>. The <b>canopen_write_ext_od</b> takes an additional argument of <i>value</i>.</td></tr>
<tr><td><b>Lua Example</b></td><td colspan="2">

```
function canopen_read_ext_od(mapargs)
    local index = mapargs[“index”]
    local subindex = mapargs[“subindex”]
    if (index ~= nil and subindex ~= nil) then
    -- This is a READ access.
    data[“access”] = 0
    data[“index”] = index
    data[“subindex”] = subindex
    gre.send_event_data(“cos.ext_od_access”,
                        “4u1 access 4u1 index 4u1 subindex”, data, “cos”)
    end
end
function canopen_write_ext_od(mapargs)
    local data = {}
    if (mapargs[“index”] ~= nil and mapargs[“subindex”] ~= nil
    and mapargs[“value”] ~= nil) then
    -- This is a WRITE access.
    data[“access”] = 1
    data[“index”] = mapargs[“index”]
    data[“subindex”] = mapargs[“subindex”]
    data[“value”] = mapargs[“value”]
    gre.send_event_data(“cos.ext_od_access”,
                        “4u1 access 4u1 index 4u1 subindex 4s1 value”,
                          data, “cos”)
    end
end
```

</td></tr>
</table>

9.3.2. cos.ext_od_result

Event	*cos.ext_od_result*	
Event Data		
Keys	**Values**	**Description**
access	<0..65535>	0 = this is a read request result. Non-zero = this is a write request result.
index	<0..65535>	Index of the item in the external object dictionary
result	<0..65535>	The result of the previous cos.ext_od_access operation. A value of 0 inidicates the previous operation was successful. A non-zero result indicates that the SDO transfer was not successful. Supported responses: COP_k_OK - 0 (Success) COP_k_ERR - 1 (Error) COP_k_NO - 2 (Unsupported SDO) COP_k_TIMEOUT - 255 (Timeout) This list does not include all of the possible error/abort codes defined in the CANopen spec, but they do represent the possible results at the DCP layer.
subindex	<0..65535>	Subindex of the item in the external object dictionary
value	<0..4294967295>	Contains the value returned from the other node in response to a read operation if it was successful. This parameter should be ignored for successful write operations. If the operation was unsuccessful where the value in result is non-zero, this parameter will contain either the CANopen SDO abort error code or an internal error code.
Direction	input	
Channel	cos	
Notes	Received by the application in response to a *cos.ext_od_access* event. If external object dictionary data was written, this result will merely be the ACK or NACK of that data. *Value* will be 0 if the data was written successfully and non-zero if the write failed. The example below assumes the Storyboard application is set up to call Lua function **canopen_ext_od_receive** when the *cos.ext_od_result* event occurs. Event data is available through the *mapargs* argument.	

Lua Example

```
function canopen_ext_od_receive(mapargs)
    local ev = mapargs.context_event_data
    if (ev[“access”] ~= nil and ev[“access”] ~= 0) then
        print(“Data written successfully to external OD.”)
    elseif (ev[“access”] == 0) then
        print(“Read ”..ev[“value”]..“from external OD: ”
                    ..ev[“index”]..”, “..ev[“subindex”]..“.”)
    end
end
```

9.3.3. cos.info

<table>
<tr><td><b>Event</b></td><td colspan="2"><i>cos.info</i></td></tr>
<tr><td colspan="3"><b>Event Data</b></td></tr>
<tr><td><b>Keys</b></td><td><b>Values</b></td><td><b>Description</b></td></tr>
<tr><td><i>version</i></td><td>&lt;version string&gt;</td><td>A string containing the CANopen stack version.</td></tr>
<tr><td><b>Direction</b></td><td colspan="2">input</td></tr>
<tr><td><b>Channel</b></td><td colspan="2">cos</td></tr>
<tr><td><b>Notes</b></td><td colspan="2">Received by the application to read the CANopen version information.
The example below assumes the Storyboard application is set up to call Lua function <b>canopen_info_receive</b> when the <i>cos.info</i> event occurs. Event data is available through the <i>mapargs</i> argument.</td></tr>
<tr><td><b>Lua Example</b></td><td colspan="2"><pre><code>function canopen_info_receive(mapargs)
    local ev = mapargs.context_event_data
    local version = ev[“version”]
    if (version ~= nil) then
        print(“Received CANopen version: ”..version)
    end
end</code></pre></td></tr>
</table>

9.3.4. cos.od

Event	*cos.od*	
Event Data		
Keys	**Values**	**Description**
index	<0..65535>	Index of the object dictionary element
subindex	<0..65535>	Subindex of the object dictionary element
datatype	<0..65535>	Data type of the object dictionary element
access	<0..65535>	Access level of the object dictionary element
value	<0..4294967295>	Value of the object dictionary element
Direction	input	
Channel	cos	
Notes	Received by the application in response to a *cos.request_od* event. The example below assumes the Storyboard application is set up to call Lua function **canopen_receive_od** when a *cos.od* event occurs. Event data is available through the mapargs argument.	

Lua Example	**function canopen_receive_od**(mapargs)
	    local ev = mapargs.context_event_data
	    local index = ev[“index”]
	    local subindex = ev[“subindex”]
	    local value = ev[“value”]
	    if (ev[“index”] ~= **nil and** ev[“subindex”] ~= **nil and** ev[“value”] ~= **nil**)
	    then
	        print(“Received ”..ev[“value”]..“from OD: ”
	                        ..ev[“index”]..”, “..ev[“subindex”]..“.”)
	    end
	end	

9.3.5. cos.request_od

Event	*cos.request_od*	
Event Data		
Keys	**Values**	**Description**
index	<0..65535>	Index of the desired object dictionary element
subindex	<0..65535>	Subindex of the desired object dictionary element
Direction	output	
Channel	cos	
Notes	Sent by the application to request an object dictionary entry by index and subindex. The example below assumes the Storyboard application is set up to call Lua function **canopen_request_od** with arguments of *index* and *subindex* with values of 0 through 65535.	
Lua Example	function **canopen_request_od**(mapargs) local data = {} local index = mapargs["index"] local subindex = mapargs["subindex"] if (index ~= nil and subindex ~= nil) then data["index"] = index data["subindex"] = subindex **gre**.send_event_data("cos.request_od", "4u1 access 4u1 index 4u1 subindex", data, "cos") end end	

9.3.6. cos.rxNNNN

Event	*cos.rxNNNN* where *NNNN* is equal to the object dictionary item's index in hexadecimal.	
Event Data		
Keys	**Values**	**Description**
index	<0..65535>	Index of the object dictionary element
subindex	<0..65535>	Subindex of the object dictionary element
value	<0..4294967295>	New value of the object dictionary element.
Direction	input	
Channel	cos	
Notes	Received by the application when an item in the object dictionary is updated by an SDO or a PDO message.	
Lua Example	**function cos_rx6000**(mapargs) -- This event handler is called when this application receives the -- cos/cos.rxNNNN event – in this case object 6000. **local** ev = mapargs.context_event_data **local** index = mapargs[“index”] **local** subindex = mapargs[“subindex”] **local** value = mapargs[“value”] **if** (ev[“index”] ~= nil and ev[“subindex”] ~= **nil and** ev[“value”] ~= **nil**) **then** index = ev[“index”] subindex = ev[“subindex”] value = ev[“value”] **print**(“Rcvd [“..index..“:”..subindex..”]“..”=”..value) -- Now do something with the new data if needed. **end** **end**	

9.3.7. cos.rx_emcy

Event	*cos.rx_emcy*	
Event Data		
Keys	**Values**	**Description**
error_code	<0..65535>	Error code
error_reg	<0..255>	Error register
count	<0..255>	Occurrence count
value	<0..65535>	Data for the EMCY message.
Direction	input	
Channel	cos	
Notes	EMCY Receive Event. This event is published by cos when an EMCY message is received from the VSM using COB-ID 0x083. The example below assumes the Storyboard application is set up to call Lua function **canopen_rx_emcy** when the *cos.rx_emcy* event is received.	

Lua Example	`function canopen_rx_emcy(mapargs)`
	`    local ev = mapargs.context_event_data`
	`    local code = ev[“error_code”]`
	`    local reg = ev[“error_reg”]`
	`    local count = ev[“count”]`
	`    local value = ev[“value”]`
	`        print(“Received EMCY message.”)`
	`        print(“__Error code: ”. . code)`
	`        print(“Error register: ”. . reg)`
	`        print(“Occurrences: ”. . count)`
	`        print(“__.___Value: ”. . value)`
	`end`	

9.3.8. cos.set_error

Event	*cos.set_error*	
Event Data		
Keys	**Values**	**Description**
error_code	<0..65535>	Error code
error_reg	<0..255>	Error register
count	<0..255>	Occurrence count
value	<0..65535>	Data for the EMCY message.
Direction	output	
Channel	cos	
Notes	EMCY Send Event. This event is sent by the application to set the error code in the CANopen stack for the PHD. The example below assumes the Storyboard application is set up to call Lua function **canopen_set_error** with arguments of *code, reg*, count and *value*.	
Lua Example	function **canopen_rx_emcy**(mapargs) local ev = mapargs.context_event_data local code = ev[“error_code”] local reg = ev[“error_reg”] local count = ev[“count”] local value = ev[“value”] local data = {} if (code ~= nil and reg ~= nil and count ~= nil and value ~= nil) then data[“code”] = code data[“reg”] = reg data[“count”] = count data[“value”] = value **gre.**send_event_data(“cos.set_error”, “2u1 error_code 1u1 error_reg 1u1 count 2u1 value”, data, “cos”) end end	

9.3.9. cos.set_od

Event	*cos.set_od*	
Event Data		
Keys	**Values**	**Description**
index	<0..65535>	Index of the object dictionary element
subindex	<0..65535>	Subindex of the object dictionary element
value	<0..4294967295>	New value of the object dictionary element.
Direction	output	
Channel	cos	
Notes	Sent by the application to set the value of a given object dictionary item. The example below assumes the Storyboard application is set up to call Lua function **canopen_set_od** with arguments of **index** and **subindex** with values of 0 through 65535 and *value* with values of 0 through 4294967295.	

Lua Example	`function canopen_set_od(mapargs)`
	`    local index = mapargs[“index”]`
	`    local subindex = mapargs[“subindex”]`
	`    local value = mapargs[“value”]`
	`    local data = {}`
	`    if (index ~= nil and subindex ~= nil and value ~= nil) then`
	`    data[“index”] = index`
	`    data[“subindex”] = subindex`
	`    data[“value”] = value`
	`    gre.send_event_data(“cos.set_od”,`
	`                    “4u1 index 4u1 subindex 4u1 value”, data, “cos”)`
	`    end`
	`end`	

9.3.10. cos.state

Event	*cos.state*	
Event Data		
Keys	**Values**	**Description**

state	0x0001 – Rx queue overflow 0x0002 – CAN overrun 0x0004 – CAN BUS-OFF 0x0008 – CAN error 0x0010 – CAN error 0x0020 – Tx queue overflow 0x0040 – Rx queue overflow 0x0100 – CANopen node started 0x0200 – CANopen node stopped 0x0400 – CANopen guard error 0x0800 – CANopen guard error 0x1000 – CANopen node pre-operational 0x2000 – CANopen node reset comm. request 0x4000 – CANopen node reset request 0x8000 – CANopen PDO error	A 16-bit value with each bit representing a different CANopen state.
app_state	0x0000 – Reset state - cos will reinitialize its internalstate machines and the object dictionary, but manufacturer objects (0x2000 to 0x5fff) are only initialized once 0x0001 – Init state - cos is waiting for init to finish 0x0002 – Init finished - transitions directly to state 3 from here 0x0003 – Check CAN bus - cos is actively looking for a working CAN bus -- if an error occurrs while waiting it transitions to state 0 0x0004 – CAN bus connected 0x0005 – Initialize application - CSDO channel and Emergency Message handler are reinitialized 0x0006 – Application is running normally 0x0007 – Reserved 0x0008 – Prepare for reset - this is triggered when a NMT reset command is received -- cos flushes buffers 0x0009 – Resetting 0x000A – Reinitialize 0x0010 – Connected after BUS OFF - successful BUS OFF handling here prevents entire cos reinitialization 0x00FF – CANopen Error	A 16-bit value with each bit representing a different CANopen application state.
Direction	input	
Channel	cos	
Notes	Received by the application to read the CANopen network state information. The example below assumes the Storyboard application is set up to call Lua function **canopen_state** when the *cos.state* event occurs. Event data is available through the *mapargs* argument.	

Lua Example	`function canopen_state(mapargs)`
	`    local ev = mapargs.context_event_data`
	`    local state = ev[“state”]`
	`    if (state ~= nil) then`
	`        if (bit32.band(state,0x0001) ~= 0x0000) then`
	`            print(“COS: Rx queue overflow.”)`
	`        end`
	`    end`
	`    -- Other COS states left as an exercise`
	`end`	

9.3.11. Reprogramming over CAN

The CANopen stack on the PHDs is able to receive files for reprogramming the display when transmitted over CAN. This feature uses the CANopen standard objects as defined in specification CiA DSP-302, where the PHD implements the objects in the table below. The transfer from the PC is a 2 stage process where the PHD package is transferred first and then the MD5hash of the project is transferred to subindex 3.

Index	Subindex	Data Type	Description
1F50	1	8-bit	Contains the number of programs; set to 2.
1F50	2	DOMAIN variable size with max of 16MB	This entry is for receiving the package for reprogramming the PHD. This file transfer needs to be completed before sending the MD5 hash/ checksum for subindex 3.
1F50	3	DOMAIN (16 bytes)	This entry is for receiving the MD5 hash of the file in subindex 2. When the 16-byte hash is received,the file transferred in subindex 2 will be verified and if the hash matches, the file will be installed.

The PHD's CANopen stack can receive the reprogramming package in either operational or non-operational mode. It may prove easier to reprogram the device if the PHD is in a known predetermined state. If this is the case, there will be a requirement to enter non-operational mode when reprogramming.

10. Appendix C

10.1. DCPCAN, Generic CAN service

The PHD DCPCAN service exists to provide the system with access to any 11 or 29 bit ID message on the bus. This allows all CAN messages to be interpreted by the application. Inter-process communication is provided via DCP to permit access from other system services.

10.1.1. DCPCAN Features

Send and receive of 8-byte messages using 11 or 29 bit IDs regardless of identifier or SA.
Currently, the DCPCAN functionality is only available on the first CAN port.

10.1.2. Bus Loading Constraints

Since the DCPCAN service allows all CAN messags to be passed to the application, all CAN messages will generate a corresponding event and the need to process the message in the application itself. Care must be taken to ensure that the application is not overloaded by Bus traffic.

10.1.3. dcpcan.can1rx

Event	*dcpcan.can1rx*	
Event Data		
Keys	**Values**	**Description**
id	<0..2047> <2147483648..4294967295>	0 = read request from other node. Non-zero = write value to other node.
length	<0..8>	Index of the item in the external object dictionary
data_lo	<0..4294967295>	Subindex of the item in the external object dictionary
data_high	<0..4294967295>	Value to write to the external object dictionary.
Direction	input	
Channel	dcpcan	
Notes	Received by the application when a CAN message is received. The example below assumes the Storyboard application is set up to call Lua function **can_receive** when a *dcpcan.can1rx* event occurrs. Event data is available through the *mapargs* argument.	

Lua Example

```
function can_receive(mapargs)
  local ev = mapargs.context_event_data
  local id = ev[“id”]
  local data_size = ev[“length”]
  local data_lo = ev[“data_lo”]
  local data_hi = ev[“data_hi”]
  if (id ~= nil and data_size ~= nil) then
    print(“Received from ID: ”. . id)
    print(“Length: ”. . data_size)
    if (data_size > 0) then
      print(“data_lo: ”. . data_lo)
    end
    if (data_size > 4) then
      print(“data_hi: ”. . data_hi)
    end
  end
end
```

10.1.4. dcpcan.can1tx

Event	*dcpcan.can1tx*	
Event Data		
Keys	**Values**	**Description**
id	<0..2047> <2147483648..4294967295>	0 = read request from other node. Non-zero = write value to other node.
length	<0..8>	Index of the item in the external object dictionary
data_lo	<0..4294967295>	Subindex of the item in the external object dictionary
data_high	<0..4294967295>	Value to write to the external object dictionary.
Direction	output	
Channel	dcpcan	
Notes	Sent by the application to cause a CAN message to be emitted. The example below assumes the Storyboard application is set up to call Lua function **can_send** when a *dcpcan.can1tx* event occurrs. Event data is available through the *mapargs* argument.	
Lua Example	function can_send(mapargs) local data = {} local id = mapargs[“id”] local data_size = mapargs[“length”] local data_lo = mapargs[“data_lo”] local data_hi = mapargs[“data_hi”] if (id ~= nil and data_size ~= nil and data_lo ~= nil and data_hi ~= nil) then data[“id”] = id data[“length”] = data_size data[“data_lo”] = data_lo data[“data_hi”] = data_hi gre.send_event_data(“dcpcan.can1tx”, “4u1 id 4u1 length 4u1 data_lo 4u1 data_hi”, data, “dcpcan”) end end	

11. Appendix D

11.1. Time zones

The following is a list of settable time zones on the PHD70 platform. You can choose to support as many or as few of these time zones as you wish, but this list represents all the valid strings that can be passed to the nxs.set_timezone and nxs.set_rtc events. Sending the nxs.set_timezone event with one of these strings will cause the device to adhere to the time zone and daylight savings rules for that time zone. Setting the time zone will also cause the system time to update to reflect the time zone.

We recommend choosing a small subset of time zones to support that meets your market needs. As the device ships configured for UTC, we suggest you support that time zone as well. It is useful for cases where users do not wish to have the device autocorrect for daylight savings time.

11.1.1. List of time zones

Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara
Africa/Asmera
Africa/Bamako
Africa/Bangui
Africa/Banjul
Africa/Bissau
Africa/Blantyre
Africa/Brazzaville
Africa/Bujumbura
Africa/Cairo
Africa/Casablanca
Africa/Ceuta
Africa/Conakry
Africa/Dakar
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Douala
Africa/El_Aaiun
Africa/Freetown
Africa/Gaborone
Africa/Harare
Africa/Johannesburg
Africa/Juba
Africa/Kampala
Africa/Khartoum
Africa/Kigali
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Lome
Africa/Luanda
Africa/Lubumbashi
Africa/Lusaka
Africa/Malabo
Africa/Maputo
Africa/Maseru
Africa/Mbabane
Africa/Mogadishu
Africa/Monrovia
Africa/Nairobi
Africa/Ndjamena
Africa/Niamey
Africa/Nouakchott
Africa/Ouagadougou
Africa/Porto-Novo
Africa/Sao_Tome
Africa/Timbuktu
Africa/Tripoli
Africa/Tunis
Africa/Windhoek
America/Adak
America/Anchorage
America/Anguilla
America/Antigua
America/Araguaina
America/Argentina/Buenos_Aires
America/Argentina/Catamarca
America/Argentina/ComodRivadavia
America/Argentina/Cordoba
America/Argentina/Jujuy
America/Argentina/La_Rioja
America/Argentina/Mendoza
America/Argentina/Rio_Gallegos
America/Argentina/Salta
America/Argentina/San_Juan
America/Argentina/San_Luis
America/Argentina/Tucuman
America/Argentina/Ushuaia
America/Aruba
America/Asuncion
America/Atikokan
America/Atka
America/Bahia
America/Bahia_Banderas
America/Barbados
America/Belem
America/Belize
America/Blanc-Sablon
America/Boa_Vista
America/Bogota
America/Boise
America/Buenos_Aires
America/Cambridge_Bay
America/Campo_Grande
America/Cancun

List of time zones continued

America/Caracas
America/Catamarca
America/Cayenne
America/Cayman
America/Chicago
America/Chihuahua
America/Coral_Harbour
America/Cordoba
America/Costa_Rica
America/Creston
America/Cuiaba
America/Curacao
America/Danmarkshavn
America/Dawson
America/Dawson_Creek
America/Denver
America/Detroit
America/Dominica
America/Edmonton
America/Eirunepe
America/El_Salvador
America/Ensenada
America/Fort_Wayne
America/Fortaleza
America/Glace_Bay
America/Godthab
America/Goose_Bay
America/Grand_Turk
America/Grenada
America/Guadeloupe
America/Guatemala
America/Guayaquil
America/Guyana
America/Halifax
America/Havana
America/Hermosillo
America/Indiana/Indianapolis
America/Indiana/Knox
America/Indiana/Marengo
America/Indiana/Petersburg
America/Indiana/Tell_City
America/Indiana/Vevay
America/Indiana/Vincennes
America/Indiana/Winamac
America/Indianapolis
America/Inuvik
America/Iqaluit
America/Jamaica
America/Jujuy
America/Juneau
America/Kentucky/Louisville
America/Kentucky/Monticello
America/Knox_IN
America/Kralendijk
America/La_Paz
America/Lima
America/Los_Angeles
America/Louisville
America/Lower_Princes
America/Maceio
America/Managua
America/Manaus
America/Marigot
America/Martinique
America/Matamoros
America/Mazatlan
America/Mendoza
America/Menominee
America/Merida
America/Metlakatla
America/Mexico_City
America/Miquelon
America/Moncton
America/Monterrey
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau
America/New_York
America/Nipigon
America/Nome
America/Noronha
America/North_Dakota/Beulah
America/North_Dakota/Center
America/North_Dakota/New_Salem
America/Ojinaga
America/Panama
America/Pangnirtung
America/Paramaribo
America/Phoenix
America/Port-au-Prince
America/Port_of_Spain
America/Porto_Acre
America/Porto_Velho
America/Puerto_Rico
America/Rainy_River
America/Rankin_Inlet
America/Recife
America/Regina
America/Resolute
America/Rio_Branco
America/Rosario
America/Santa_Isabel
America/Santarem
America/Santiago
America/Santo_Domingo
America/Sao_Paulo
America/Scoresbysund
America/Shiprock
America/Sitka
America/St_Barthelemy
America/St_Johns
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Swift_Current
America/Tegucigalpa
America/Thule
America/Thunder_Bay
America/Tijuana
America/Toronto
America/Tortola
America/Vancouver
America/Virgin
America/Whitehorse
America/Winnipeg
America/Yakutat
America/Yellowknife
Antarctica/Casey
Antarctica/Davis
Antarctica/DumontDUrville
Antarctica/Macquarie
Antarctica/Mawson
Antarctica/McMurdo
Antarctica/Palmer
Antarctica/Rothera
Antarctica/South_Pole

List of time zones continued

Antarctica/Syowa
Antarctica/Troll
Antarctica/Vostok
Arctic/Longyearbyen
Asia/Aden
Asia/Almaty
Asia/Amman
Asia/Anadyr
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Baghdad
Asia/Bahrain
Asia/Baku
Asia/Bangkok
Asia/Beirut
Asia/Bishkek
Asia/Brunei
Asia/Calcutta
Asia/Chita
Asia/Choibalsan
Asia/Chongqing
Asia/Chungking
Asia/Colombo
Asia/Dacca
Asia/Damascus
Asia/Dhaka
Asia/Dili
Asia/Dubai
Asia/Dushanbe
Asia/Gaza
Asia/Harbin
Asia/Hebron
Asia/Ho_Chi_Minh
Asia/Hong_Kong
Asia/Hovd
Asia/Irkutsk
Asia/Istanbul
Asia/Jakarta
Asia/Jayapura
Asia/Jerusalem
Asia/Kabul
Asia/Kamchatka
Asia/Karachi
Asia/Kashgar
Asia/Kathmandu
Asia/Katmandu
Asia/Khandyga
Asia/Kolkata
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuching
Asia/Kuwait
Asia/Macao
Asia/Macau
Asia/Magadan
Asia/Makassar
Asia/Manila
Asia/Muscat
Asia/Nicosia
Asia/Novokuznetsk
Asia/Novosibirsk
Asia/Omsk
Asia/Oral
Asia/Phnom_Penh
Asia/Pontianak
Asia/Pyongyang
Asia/Qatar
Asia/Qyzylorda
Asia/Rangoon
Asia/Riyadh
Asia/Saigon
Asia/Sakhalin
Asia/Samarkand
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Srednekolymsk
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Tel_Aviv
Asia/Thimbu
Asia/Thimphu
Asia/Tokyo
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Urumqi
Asia/Ust-Nera
Asia/Vientiane
Asia/Vladivostok
Asia/Yakutsk
Asia/Yekaterinburg
Asia/Yerevan
Atlantic/Azores
Atlantic/Bermuda
Atlantic/Canary
Atlantic/Cape_Verde
Atlantic/Faeroe
Atlantic/Faroe
Atlantic/Jan_Mayen
Atlantic/Madeira
Atlantic/Reykjavik
Atlantic/South_Georgia
Atlantic/St_Helena
Atlantic/Stanley
Australia/ACT
Australia/Adelaide
Australia/Brisbane
Australia/Broken_Hill
Australia/Canberra
Australia/Currie
Australia/Darwin
Australia/Eucla
Australia/Hobart
Australia/LHI
Australia/Lindeman
Australia/Lord_Howe
Australia/Melbourne
Australia/NSW
Australia/North
Australia/Perth
Australia/Queensland
Australia/South
Australia/Sydney
Australia/Tasmania
Australia/Victoria
Australia/West
Australia/Yancowinna
Brazil/Acre
Brazil/DeNoronha
Brazil/East
Brazil/West
CET
CST6CDT

List of time zones continued

Canada/Atlantic
Canada/Central
Canada/East-Saskatchewan
Canada/Eastern
Canada/Mountain
Canada/Newfoundland
Canada/Pacific
Canada/Saskatchewan
Canada/Yukon
Chile/Continental
Chile/EasterIsland
Cuba
EET
EST
EST5EDT
Egypt
Eire
Etc/GMT
Etc/GMT+0
Etc/GMT+1
Etc/GMT+10
Etc/GMT+11
Etc/GMT+12
Etc/GMT+2
Etc/GMT+3
Etc/GMT+4
Etc/GMT+5
Etc/GMT+6
Etc/GMT+7
Etc/GMT+8
Etc/GMT+9
Etc/GMT-0
Etc/GMT-1
Etc/GMT-10
Etc/GMT-11
Etc/GMT-12
Etc/GMT-13
Etc/GMT-14
Etc/GMT-2
Etc/GMT-3
Etc/GMT-4
Etc/GMT-5
Etc/GMT-6
Etc/GMT-7
Etc/GMT-8
Etc/GMT-9
Etc/GMT0
Etc/Greenwich
Etc/UCT
Etc/UTC
Etc/Universal
Etc/Zulu
Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belfast
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
Europe/Brussels
Europe/Bucharest
Europe/Budapest
Europe/Busingen
Europe/Chisinau
Europe/Copenhagen
Europe/Dublin
Europe/Gibraltar
Europe/Guernsey
Europe/Helsinki
Europe/Isle_of_Man
Europe/Istanbul
Europe/Jersey
Europe/Kaliningrad
Europe/Kiev
Europe/Lisbon
Europe/Ljubljana
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Mariehamn
Europe/Minsk
Europe/Monaco
Europe/Moscow
Europe/Nicosia
Europe/Oslo
Europe/Paris
Europe/Podgorica
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/San_Marino
Europe/Sarajevo
Europe/Simferopol
Europe/Skopje
Europe/Sofia
Europe/Stockholm
Europe/Tallinn
Europe/Tirane
Europe/Tiraspol
Europe/Uzhgorod
Europe/Vaduz
Europe/Vatican
Europe/Vienna
Europe/Vilnius
Europe/Volgograd
Europe/Warsaw
Europe/Zagreb
Europe/Zaporozhye
Europe/Zurich
GB
GB-Eire
GMT
GMT+0
GMT-0
GMT0
Greenwich
HST
Hongkong
Iceland
Indian/Antananarivo
Indian/Chagos
Indian/Christmas
Indian/Cocos

Indian/Comoro
Indian/Kerguelen
Indian/Mahe
Indian/Maldives
Indian/Mauritius
Indian/Mayotte
Indian/Reunion
Iran
Israel
Jamaica
Japan
Kwajalein

List of time zones continued

Libya
MET
MST
MST7MDT
Mexico/BajaNorte
Mexico/BajaSur
Mexico/General
NZ
NZ-CHAT
Navajo
PRC
PST8PDT
Pacific/Apia
Pacific/Auckland
Pacific/Bougainville
Pacific/Chatham
Pacific/Chuuk
Pacific/Easter
Pacific/Efate
Pacific/Enderbury
Pacific/Fakaofo
Pacific/Fiji
Pacific/Funafuti
Pacific/Galapagos
Pacific/Gambier
Pacific/Guadalcanal
Pacific/Guam
Pacific/Honolulu
Pacific/Johnston
Pacific/Kiritimati
Pacific/Kosrae
Pacific/Kwajalein
Pacific/Majuro
Pacific/Marquesas
Pacific/Midway
Pacific/Nauru
Pacific/Niue
Pacific/Norfolk
Pacific/Noumea
Pacific/Pago_Pago
Pacific/Palau
Pacific/Pitcairn
Pacific/Pohnpei
Pacific/Ponape
Pacific/Port_Moresby
Pacific/Rarotonga
Pacific/Saipan
Pacific/Samoa
Pacific/Tahiti
Pacific/Tarawa
Pacific/Tongatapu
Pacific/Truk
Pacific/Wake
Pacific/Wallis
Pacific/Yap
Poland
Portugal
ROC
ROK
Singapore
Turkey
UCT
US/Alaska
US/Aleutian
US/Arizona
US/Central
US/East-Indiana
US/Eastern
US/Hawaii
US/Indiana-Starke
US/Michigan
US/Mountain
US/Pacific
US/Pacific-New
US/Samoa
UTC
Universal
W-SU
WET
Zulu

© 2023 Parker Hannifin Corporation

Parker Hannifin Corporation
Electronic Motion and Controls Division
850 Arthur Ave
Elk Grove Village, IL 60007
phone 800 221 9257
www.parker.com/emc