Sense-Emu Documentation

Release 1.1

Raspberry Pi Foundation

Jul 09, 2018

Contents

CHAPTER 1

Installation

1.1 Raspbian installation

If you are using the Raspbian[1] distro, it is best to install the Sense HAT Emulator using the system's package manager: apt. This will ensure that the emulator is easy to keep up to date, and easy to remove should you wish to do so. To install using apt simply:

```
$ sudo apt-get update
$ sudo apt-get install python-sense-emu python3-sense-emu sense-emu-tools
```

These three packages contain the following things:

sense-emu-tools This package contains the Sense HAT Emulator application.

python-sense-emu This is the Python 2 version of the Sense HAT Emulator library.

python3-sense-emu This is the Python 3 version of the Sense HAT Emulator library.

To upgrade your installation when new releases are made you can simply use apt's normal upgrade procedure:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

If you ever need to remove your installation:

```
$ sudo apt-get remove python-sense-emu python3-sense-emu sense-emu-tools
```

1.2 Ubuntu installation

To install from the author's PPA[2]:

```
$ sudo add-apt-repository ppa://waveform/ppa
$ sudo apt-get update
$ sudo apt-get install python-sense-emu python3-sense-emu sense-emu-tools
```

[1] http://www.raspbian.org/
[2] https://launchpad.net/~waveform/+archive/ppa

To upgrade your installation when new releases are made you can simply use apt's normal upgrade procedure:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

To remove the installation:

```
$ sudo apt-get remove python-sense-emu python3-sense-emu sense-emu-tools
```

1.3 macOS installation

The following installation instructions assume you are using Homebrew[3] as your macOS package manager:

```
$ brew install pygobject3 gtk+3
```

If you are using Python virtual environments for the Sense HAT Emulator, the `system-site-packages` option must be enabled within the virtualenv.

For existing virtual environments:

```
$ rm ${VIRTUAL_ENV}/lib/python*/no-global-site-packages.txt
```

For new virtual environments:

```
$ virtualenv --system-site-packages
```

Activate your virtual environment and install the Sense HAT Emulator:

```
$ pip install sense-emu
```

1.4 Alternate platforms

On platforms other than Raspbian or Ubuntu, it is probably simplest to install system wide using Python's `pip` tool:

```
$ pip install sense-emu
```

To upgrade your installation when new releases are made:

```
$ pip install -U sense-emu
```

If you ever need to remove your installation:

```
$ pip uninstall sense-emu
```

Note: The emulator application requires PyGObject and cairo to be installed (GTK3 bindings for Python), but this cannot be obtained from PyPI; install PyGObject manually from your operating system's package manager (e.g. python-gi, python3-gi, python-gi-cairo, and python3-gi-cairo on Raspbian/Ubuntu).

Also note that installation via `pip` won't create short-cuts for the emulator application in your desktop's start menu. Instead you will have to launch it manually by running `sense_emu_gui` from the command line.

[3] http://brew.sh/

CHAPTER 2

Examples

2.1 Introduction

The Sense HAT emulator exactly mirrors the official Sense HAT API. The only difference (required because both the emulator and actual library can be installed simultaneously on a Pi) is the name: `sense_emu` instead of `sense_hat`. It is recommended to import the library in the following manner at the top of your code:

```
from sense_emu import SenseHat
```

Then, when you want to change your code to run on the actual HAT all you need do is change this line to:

```
from sense_hat import SenseHat
```

To run your scripts under the emulator, first start the emulator application, then start your script.

Several example scripts, with varying degrees of difficulty, are available from the *File* → *Open example* menu within the emulator. Selecting an example from this menu will open it in Python's IDLE environment.

Note: The example will be opened directly from the installation. To edit the example for your own purposes, use *File* → *Save As* in IDLE to save the file under your home directory (e.g. `/home/pi`).

A selection of example scripts is given in the following sections.

2.2 Temperature

Displays the current temperature reading on the Sense HAT's screen:

```
from sense_emu import SenseHat

sense = SenseHat()

red = (255, 0, 0)
blue = (0, 0, 255)

while True:
```

(continues on next page)

(continued from previous page)

```
    temp = sense.temp
    pixels = [red if i < temp else blue for i in range(64)]
    sense.set_pixels(pixels)
```

2.3 Humidity

Displays the current humidity reading on the Sense HAT's screen:

```
from sense_emu import SenseHat

sense = SenseHat()

green = (0, 255, 0)
white = (255, 255, 255)

while True:
    humidity = sense.humidity
    humidity_value = 64 * humidity / 100
    pixels = [green if i < humidity_value else white for i in range(64)]
    sense.set_pixels(pixels)
```

2.4 Joystick

Scrolls a blip around the Sense HAT's screen in response to joystick motions:

```
from sense_emu import SenseHat

x = y = 4
hat = SenseHat()

def update_screen():
    hat.clear()
    hat.set_pixel(x, y, 255, 255, 255)

def clamp(value, min_value=0, max_value=7):
    return min(max_value, max(min_value, value))

def move_dot(event):
    global x, y
    if event.action in ('pressed', 'held'):
        x = clamp(x + {
            'left': -1,
            'right': 1,
            }.get(event.direction, 0))
        y = clamp(y + {
            'up': -1,
            'down': 1,
            }.get(event.direction, 0))

update_screen()
while True:
    for event in hat.stick.get_events():
        move_dot(event)
        update_screen()
```

An alternative way to write this example using the joystick's event handler attributes is given below:

```python
from sense_emu import SenseHat
from signal import pause

x = y = 4
hat = SenseHat()

def update_screen():
    hat.clear()
    hat.set_pixel(x, y, 255, 255, 255)

def clamp(value, min_value=0, max_value=7):
    return min(max_value, max(min_value, value))

def move_dot(event):
    global x, y
    if event.action in ('pressed', 'held'):
        x = clamp(x + {
            'left': -1,
            'right': 1,
            }.get(event.direction, 0))
        y = clamp(y + {
            'up': -1,
            'down': 1,
            }.get(event.direction, 0))

update_screen()
hat.stick.direction_up = move_dot
hat.stick.direction_down = move_dot
hat.stick.direction_left = move_dot
hat.stick.direction_right = move_dot
hat.stick.direction_any = update_screen
pause()
```

2.5 Rainbow

Scrolls a rainbow of colours across the Sense HAT's pixels:

```python
from colorsys import hsv_to_rgb
from time import sleep
from sense_emu import SenseHat

# Hues represent the spectrum of colors as values between 0 and 1. The range
# is circular so 0 represents red, ~0.2 is yellow, ~0.33 is green, 0.5 is cyan,
# ~0.66 is blue, ~0.84 is purple, and 1.0 is back to red. These are the initial
# hues for each pixel in the display.
hues = [
    0.00, 0.00, 0.06, 0.13, 0.20, 0.27, 0.34, 0.41,
    0.00, 0.06, 0.13, 0.21, 0.28, 0.35, 0.42, 0.49,
    0.07, 0.14, 0.21, 0.28, 0.35, 0.42, 0.50, 0.57,
    0.15, 0.22, 0.29, 0.36, 0.43, 0.50, 0.57, 0.64,
    0.22, 0.29, 0.36, 0.44, 0.51, 0.58, 0.65, 0.72,
    0.30, 0.37, 0.44, 0.51, 0.58, 0.66, 0.73, 0.80,
    0.38, 0.45, 0.52, 0.59, 0.66, 0.73, 0.80, 0.87,
    0.45, 0.52, 0.60, 0.67, 0.74, 0.81, 0.88, 0.95,
    ]

hat = SenseHat()

def scale(v):
```

(continues on next page)

(continued from previous page)

```
    return int(v * 255)

while True:
    # Rotate the hues
    hues = [(h + 0.01) % 1.0 for h in hues]
    # Convert the hues to RGB values
    pixels = [hsv_to_rgb(h, 1.0, 1.0) for h in hues]
    # hsv_to_rgb returns 0..1 floats; convert to ints in the range 0..255
    pixels = [(scale(r), scale(g), scale(b)) for r, g, b in pixels]
    # Update the display
    hat.set_pixels(pixels)
    sleep(0.04)
```

CHAPTER 3

Sense HAT API

The main class which is used to interact with the Sense HAT emulator is `SenseHat` (page 7). This provides accesss to all sensors, the LED pixel display, and the joystick. It is recommended that you import the library using the following idiom:

```
from sense_emu import SenseHat
```

This way, when you wish to deploy your code on an actual Sense HAT the only change you need to make is to this line, changing it to:

```
from sense_hat import SenseHat
```

3.1 SenseHat

class sense_emu.**SenseHat**(*imu_settings_file='RTIMULib'*, *text_assets='sense_hat_text'*)

The main interface the Raspberry Pi Sense HAT.

This class provides properties to query the various sensors on the Sense HAT (`temp`, `pressure`, `humidity`, `gyro`, etc.) and methods to control the LED "screen" on the HAT (`set_pixel()` (page 8), `set_pixels()` (page 8)).

The *imu_settings_file* parameter specifies the base name of the configuration file used to calibrate the sensors on the HAT. An ".ini" suffix will be implicitly added to this filename. If a file with the resulting name is present in `~/.config/sense_hat`, it will be used in the configuration. Otherwise, the file will be located within `/etc`, and will be copied to `~/.config/sense_hat` before use.

The *text_assets* parameter provides the base name of the PNG image and text file which will be used to define the font used by the `show_message()` (page 9) method.

clear(**args*)

Clears the LED matrix with a single colour, default is black / off

e.g. ap.clear() or ap.clear(r, g, b) or colour = (r, g, b) ap.clear(colour)

flip_h(*redraw=True*)

Flip LED matrix horizontal

flip_v(*redraw=True*)

Flip LED matrix vertical

gamma_reset()
Resets the LED matrix gamma correction to default

get_accelerometer()
Gets the orientation in degrees from the accelerometer only

get_accelerometer_raw()
Accelerometer x y z raw data in Gs

get_compass()
Gets the direction of North from the magnetometer in degrees

get_compass_raw()
Magnetometer x y z raw data in uT (micro teslas)

get_gyroscope()
Gets the orientation in degrees from the gyroscope only

get_gyroscope_raw()
Gyroscope x y z raw data in radians per second

get_humidity()
Returns the percentage of relative humidity

get_orientation_degrees()
Returns a dictionary object to represent the current orientation in degrees, 0 to 360, using the aircraft principal axes of pitch, roll and yaw

get_orientation_radians()
Returns a dictionary object to represent the current orientation in radians using the aircraft principal axes of pitch, roll and yaw

get_pixel(*x*, *y*)
Returns a list of [R,G,B] representing the pixel specified by x and y on the LED matrix. Top left = 0,0 Bottom right = 7,7

get_pixels()
Returns a list containing 64 smaller lists of `[R,G,B]` pixels representing what is currently displayed on the LED matrix

get_pressure()
Returns the pressure in Millibars

get_temperature()
Returns the temperature in Celsius

get_temperature_from_humidity()
Returns the temperature in Celsius from the humidity sensor

get_temperature_from_pressure()
Returns the temperature in Celsius from the pressure sensor

load_image(*file_path*, *redraw=True*)
Accepts a path to an 8 x 8 image file and updates the LED matrix with the image

set_imu_config(*compass_enabled*, *gyro_enabled*, *accel_enabled*)
Enables and disables the gyroscope, accelerometer and/or magnetometer input to the orientation functions

set_pixel(*x*, *y*, **args*)
Updates the single `[R,G,B]` pixel specified by x and y on the LED matrix Top left = 0,0 Bottom right = 7,7

e.g. ap.set_pixel(x, y, r, g, b) or pixel = (r, g, b) ap.set_pixel(x, y, pixel)

set_pixels(*pixel_list*)
Accepts a list containing 64 smaller lists of `[R,G,B]` pixels and updates the LED matrix. R,G,B elements must intergers between 0 and 255

set_rotation(*r=0*, *redraw=True*)
Sets the LED matrix rotation for viewing, adjust if the Pi is upside down or sideways. 0 is with the Pi HDMI port facing downwards

show_letter(*s*, *text_colour=[255, 255, 255]*, *back_colour=[0, 0, 0]*)
Displays a single text character on the LED matrix using the specified colours

show_message(*text_string*, *scroll_speed=0.1*, *text_colour=[255, 255, 255]*, *back_colour=[0, 0, 0]*)
Scrolls a string of text across the LED matrix using the specified speed and colours

stick
A `SenseStick` (page 9) object representing the Sense HAT's joystick.

3.2 SenseStick

class sense_emu.**SenseStick**
Represents the joystick on the Sense HAT.

get_events()
Returns a list of all joystick events that have occurred since the last call to `get_events()` (page 9). The list contains events in the order that they occurred. If no events have occurred in the intervening time, the result is an empty list.

wait_for_event(*emptybuffer=False*)
Waits until a joystick event becomes available. Returns the event, as an `InputEvent` (page 10) tuple.

If *emptybuffer* is `True` (it defaults to `False`), any pending events will be thrown away first. This is most useful if you are only interested in "pressed" events.

direction_any
The function to be called when the joystick is used. The function can either take a parameter which will be the `InputEvent` (page 10) tuple that has occurred, or the function can take no parameters at all.

This event will always be called *after* events associated with a specific action. Assign `None` to prevent this event from being fired.

direction_down
The function to be called when the joystick is pushed down. The function can either take a parameter which will be the `InputEvent` (page 10) tuple that has occurred, or the function can take no parameters at all.

Assign `None` to prevent this event from being fired.

direction_left
The function to be called when the joystick is pushed left. The function can either take a parameter which will be the `InputEvent` (page 10) tuple that has occurred, or the function can take no parameters at all.

Assign `None` to prevent this event from being fired.

direction_middle
The function to be called when the joystick middle click is pressed. The function can either take a parameter which will be the `InputEvent` (page 10) tuple that has occurred, or the function can take no parameters at all.

Assign `None` to prevent this event from being fired.

direction_right
The function to be called when the joystick is pushed right. The function can either take a parameter which will be the `InputEvent` (page 10) tuple that has occurred, or the function can take no parameters at all.

Assign `None` to prevent this event from being fired.

direction_up
The function to be called when the joystick is pushed up. The function can either take a parameter which will be the *InputEvent* (page 10) tuple that has occurred, or the function can take no parameters at all.

Assign None to prevent this event from being fired.

3.3 InputEvent

class sense_emu.**InputEvent**
A namedtuple()[4] derivative representing a joystick event. The following attributes are present:

timestamp
The time at which the event occurred, represented as the number of seconds since the UNIX epoch (same output as time()[5]).

direction
The direction in which the joystick was pushed (or released from), as one of the constants *DIRECTION_UP* (page 10), *DIRECTION_DOWN* (page 10), *DIRECTION_LEFT* (page 10), *DIRECTION_RIGHT* (page 10), *DIRECTION_MIDDLE* (page 10)

action
The action that occurred, as one of the constants *ACTION_PRESSED* (page 10), *ACTION_RELEASED* (page 10), or *ACTION_HELD* (page 10).

3.4 Constants

sense_emu.**DIRECTION_UP**

sense_emu.**DIRECTION_DOWN**

sense_emu.**DIRECTION_LEFT**

sense_emu.**DIRECTION_RIGHT**

sense_emu.**DIRECTION_MIDDLE**
Constants representating the direction in which the joystick has been pushed. *DIRECTION_MIDDLE* (page 10) refers to pressing the joystick as a button.

sense_emu.**ACTION_PRESSED**

sense_emu.**ACTION_RELEASED**

sense_emu.**ACTION_HELD**
Constants representing the actions that can be applied to the joystick.

[4] https://docs.python.org/3.5/library/collections.html#collections.namedtuple
[5] https://docs.python.org/3.5/library/time.html#time.time

CHAPTER 4

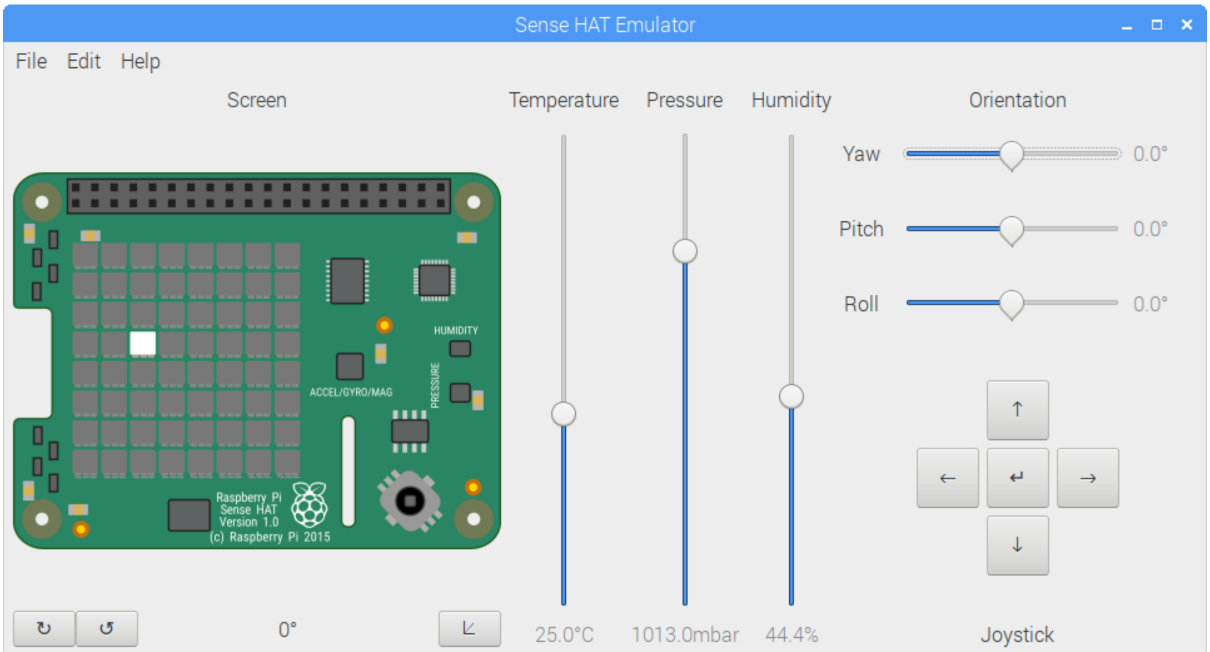
Sense Emulator

The Sense HAT emulator application. This GTK application provides an interactive interface for emulating the Raspberry Pi Sense HAT.

4.1 Synopsis

```
sense_emu_gui
```

4.2 Usage



The main window is divided into four parts. At the left is a visual representation of the Sense HAT. Scripts using the emulator library (sense_emu (page 7)) to set the HAT's LEDs will display the result here. Immediately below the LEDs are the rotation buttons which rotate the view of the HAT. These buttons also affect the action of the joystick buttons (covered below).

To the right of the LEDs are three sliders representing the temperature, pressure, and humidity of the emulated HAT's environment.

Note: The emulation does not *precisely* reflect the settings of the temperature, pressure, and humidity sliders. Random errors are introduced that scale according to the sensor specifications, and as the sliders are adjusted, the sensor value will gradually drift towards the new setting at a similar rate to the sensors on the real HAT.

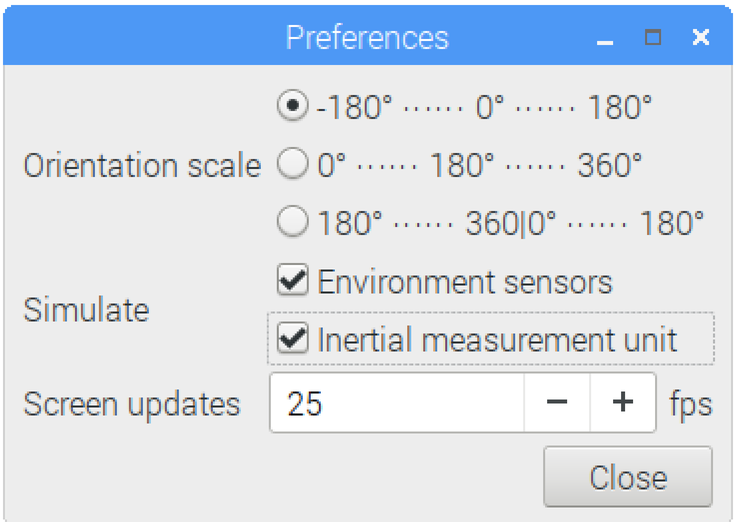
On the far right of the window, three sliders provide the orientation of the emulated HAT in terms of yaw (rotation around the vertical Z axis), pitch (rotation around the Y axis), and roll (rotation around the X axis). Adjusting these sliders affect the accelerometer, gyroscope, and magnetometer (compass) sensors on the emulated HAT. The emulated HAT assumes gravity runs vertically in the direction of the Z axis (as in the real HAT), and that North is in the direction of the X axis.

Finally, at the bottom right of the window, a series of buttons are provided to emulate the joystick on the HAT. The buttons will simulate *press*, *release*, and *hold* events generated by the real joystick. When the view of the HAT is rotated, the joystick buttons will act in the new orientation of the HAT. For example, initially the "up" button will send "up" events. After the HAT is rotated 90° the "up" button will send "right" events. After another 90° rotation, the "up" button will send "down" events, and so on.

Note: The emulator must be run prior to starting any scripts which expect to use the `sense_emu` (page 7) library. However, the emulator can be terminated (and restarted) while scripts using the library are running (obviously, when the emulator isn't running sensor errors can't be emulated and all sensor readings will appear fixed). Attempting to spawn more than one instance of the emulator will simply activate the existing instance.

4.3 Preferences

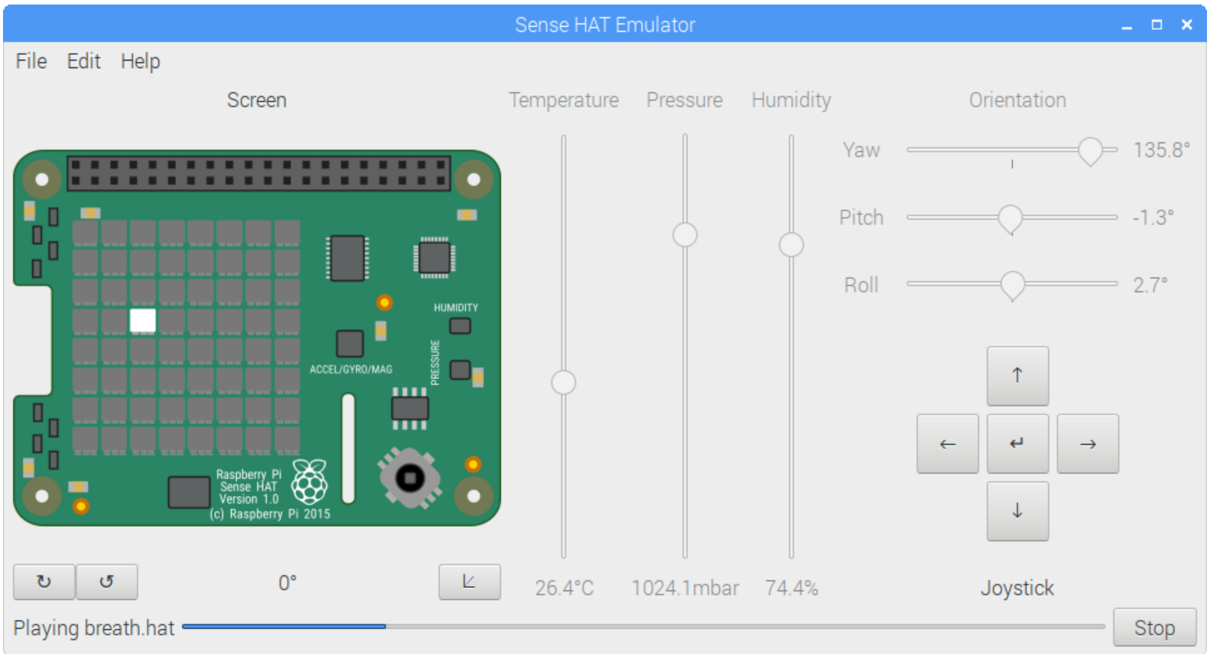
On slower Pis in particular, you may wish to disable some aspects of the emulation for performance purposes. From the "Edit" menu, select "Preferences". In the window that appears you can control which aspects of the emulation are enabled, and what speed the screen updates will be limited to.



You can also control the appearance of angles for the orientation sliders (note that this is a purely visual preference; it doesn't affect the output of the emulated sensors in any way).

4.4 Replay

Recordings of actual sensor readings from a Sense HAT can be replayed within the emulator. From the "File" menu, select "Replay recording". From the file selection dialog that appears, select the recording you wish to replay and click "Open". The replay will immediately begin, with progress displayed in a bar at the bottom of the main window. You can click "Stop" (next to the progress bar) to terminate playback of the recording.



During playback, the sensor sliders will move according to the data in the recording but will be disabled (to prevent the user affecting the replay). At the end of the replay (or immediately after termination of playback), the sliders will be left at their present positions and re-enabled.

CHAPTER 5

sense_rec

Records sensor readings from the Raspberry Pi Sense HAT in real time, outputting the results to a file for later playback or analysis. This is most useful for preparing records of experiments for use with the Sense HAT emulator. For example, a recording of a Sense HAT being dropped, a recording of a HAB[6] flight, a recording of the cycle of temperature over a few days, etc.

5.1 Synopsis

```
sense_rec [-h] [--version] [-q] [-v] [-l FILE] [-P] [-c CONFIG]
          [-d DURATION] [-f] output
```

5.2 Description

-h, --help
show this help message and exit

--version
show this program's version number and exit

-q, --quiet
produce less console output

-v, --verbose
produce more console output

-l FILE**, --log-file** FILE
log messages to the specified file

-P, --pdb
run under PDB (debug mode)

-c FILE**, --config** FILE
the Sense HAT configuration file to use (default: `/etc/RTIMULib.ini`)

-d SECS**, --duration** SECS
the duration to record for in seconds (default: record until terminated with `Control-C`)

[6] https://en.wikipedia.org/wiki/High-altitude_balloon

-i SECS, **--interval** SECS
the delay between each reading in seconds (default: the IMU polling interval, typically 0.003 seconds)

-f, --flush
flush every record to disk immediately; reduces chances of truncated data on power loss, but greatly increases disk activity

5.3 Examples

To record an experiment with the Sense HAT, simply execute **sense_rec** with the filename you wish to record the results:

```
$ sense_rec experiment.hat
```

By default, the recording will continue indefinitely. Press `Control-C` to terminate the recording. If you want to record for a specific duration, you can use the *--duration* (page 15) option to specify the number of seconds:

```
$ sense_rec --duration 10 short_experiment.hat
```

This tool can be run simultaneously with scripts that use the Sense HAT. Simply start your script in one terminal, then open another to start **sense_rec**. Alternatively, you can use the shell's job control facilities to start recording in the background:

```
$ sense_rec experiment.hat &
$ python experiment.py
...
$ kill %1
```

> **Warning:** Be aware that other scripts attempting to use the HAT's sensors will likely obtain different readings than they would have if run standalone. Some of the HAT's sensors are affected by their query-rate, and **sense_rec** drives all sensors at close to their maximum rate.

If – is specified as the output file, **sense_rec** will write its output to stdout. This can be used to reduce the disk space required for long output by piping the output through a compression tool like **gzip**:

```
$ sense_rec - | gzip -c - > experiment.hat.gz
```

When compressed in this manner the data typically uses approximately 3Kb per second (without **gzip** the recording will use approximately 10Kb of disk space per second). Use **gunzip** to de-compress the data for playback or analysis:

```
$ gunzip -c experiment.hat.gz | sense_play -
```

Another method of reducing the data usage is increasing the interval between readings (the default is the IMU polling interval which is an extremely short 3ms). Obviously a longer interval will reduce the "fidelity" of the recording; you will only see the sensors update at each interval during playback, however it can be extremely useful for very long recordings. For example, to record with a 1 second interval between readings for 24 hours:

```
$ sense_rec -i 1 -d $((24*60*60)) one_day_experiment.hat
```

Finally, you can use pipes in conjunction with **sense_csv** to produce CSV output directly:

```
$ sense_rec - | sense_csv - experiment.csv
```

Be warned that CSV data is substantially larger than the binary format (CSV data uses approximately 25Kb per second at the default interval).

CHAPTER 6

sense_play

Replays readings recorded from a Raspberry Pi Sense HAT, via the Sense HAT emulation library.

6.1 Synopsis

```
sense_play [-h] [--version] [-q] [-v] [-l FILE] [-P] input
```

6.2 Description

-h, --help
show this help message and exit

--version
show this program's version number and exit

-q, --quiet
produce less console output

-v, --verbose
produce more console output

-l FILE**, --log-file** FILE
log messages to the specified file

-P, --pdb
run under PDB (debug mode)

6.3 Examples

To play back an experiment recorded from the Sense HAT, simply execute **sense_play** with the filename you wish to play back:

```
$ sense_play experiment.hat
```

Playback will start immediately and continue in real-time (at the recording rate) until the file is exhausted. If you wish to start an emulated script at the same time as playback, you can use the shell's job control facilities:

```
$ sense_play experiment.hat & python experiment.py
```

If - is specified as the input file, **sense_play** will read its from stdin. This can be used to play back compressed recordings (see Examples under **sense_rec**) without using any disk space for decompression:

```
$ gunzip -c experiment.hat.gz | sense_play -
```

Note: If playback is going too slowly (e.g. because the Pi is too busy with other tasks, or because the data cannot be read quickly enough from the SD card), **sense_play** will skip records and print a warning to the console at the end of playback with the number of records skipped.

CHAPTER 7

sense_csv

Converts a Sense HAT recording to CSV format, for the purposes of debugging or analysis.

7.1 Synopsis

```
sense_csv [-h] [--version] [-q] [-v] [-l FILE] [-P]
          [--timestamp-format TIMESTAMP_FORMAT] [--header] input output
```

7.2 Description

-h, --help
show this help message and exit

--version
show this program's version number and exit

-q, --quiet
produce less console output

-v, --verbose
produce more console output

-l FILE**, --log-file** FILE
log messages to the specified file

-P, --pdb
run under PDB (debug mode)

--timestamp-format FMT
the format to use when outputting the record timestamp (default: ISO8601 format, which is "%Y-%m-%dT%H:%M:%S.%f"; see *strftime(3)* for information on valid format parameters)

--header
if specified, output column headers at the start of the output

7.3 Examples

To convert a recording to CSV, simply run **sense_csv** with the recorded file as the first filename, and the output CSV file as the second:

```
$ sense_csv experiment.hat experiment.csv
```

By default, only the data is output, with the columns defined as follows:

1. Timestamp - the moment in time at which the readings were taken (note that as the Pi lacks a real-time clock, this is likely to be inaccurate unless the clock has been set with NTP).
2. Pressure - the reading from the pressure sensor in hectopascals (hPa).
3. Temperature - the temperature reading from the pressure sensor in degrees celsius (°C).
4. Humidity - the reading from the humidity sensor in % humidity.
5. Temperature - the temperature reading from the humidity sensor in degrees celsius (°C).
6. Accelerometer X-axis - the acceleration reading along the X-axis of the HAT in g.
7. Accelerometer Y-axis.
8. Accelerometer Z-axis.
9. Gyroscope X-axis - the angular rate of change around the X-axis of the HAT in degrees per second.
10. Gyroscope Y-axis.
11. Gyroscope Z-axis.
12. Compass X-axis - the magnetometer reading along the X-axis in micro-teslas.
13. Compass Y-axis.
14. Compass Z-axis.
15. Orientation X-axis - the computed orientation of the HAT as radians rotation ($-\pi$ to $+\pi$) about the X-axis.
16. Orientation Y-axis.
17. Orientation Z-axis.

If you wish to include column headers as the first row of data, simply specify the *--header* (page 19) option:

```
$ sense_csv --header experiment.hat experiment.csv
```

If - is specified for either filename, **sense_csv** will read from stdin, or write to stdout. This can be used in conjunction with other standard command line utilities for all sorts of effects. For example, to produce a CSV file containing only the timestamps, humidity, and accelerometer readings:

```
$ sense_csv --header experiment.hat - | cut -d, -f1,4,6-8 > experiment.csv
```

CHAPTER 8

Change log

8.1 Release 1.1 (2018-07-07)

- Enforce a minimum width of window to ensure orientation sliders are never excessively small (#9[7])
- Various documentation updates (#12[8] etc.)
- Resizing of the display for high-resolution displays (#14[9])
- Orientation sliders had no effect when world simulation was disabled (#19[10])
- When the emulator was spawned by instantiating `SenseHat()` in an interpreter, pressing Ctrl+C in the interpreter would affect the emulator (#22[11])
- Make **sense_rec** interval configurable (#24[12])

Many thanks to everyone who reported bugs and provided patches!

8.2 Release 1.0 (2016-08-31)

- Initial release

[7] https://github.com/RPi-Distro/python-sense-emu/issues/9
[8] https://github.com/RPi-Distro/python-sense-emu/issues/12
[9] https://github.com/RPi-Distro/python-sense-emu/issues/14
[10] https://github.com/RPi-Distro/python-sense-emu/issues/19
[11] https://github.com/RPi-Distro/python-sense-emu/issues/22
[12] https://github.com/RPi-Distro/python-sense-emu/issues/24

CHAPTER 9

License

Copyright 2016 Raspberry Pi Foundation[13].

The Sense HAT Emulator package is licensed under the terms of the GNU Lesser General Public License version 2.1[14] or above, with the exception of the **sense_emu_gui**, **sense_rec**, **sense_play**, and **sense_csv** applications which are each licensed under the terms of the GNU General Public License version 2.0[15] or above. The full texts of these licenses can be found in the following sections.

9.1 GNU Lesser General Public License v2.1 or later

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 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.

9.1.1 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.

[13] http://raspberrypi.org/
[14] https://www.gnu.org/licenses/old-licenses/lgpl-2.1.en.html
[15] https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

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.

9.1.2 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

9.1.3 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 an idea of what it does. Copyright (C) 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!

9.2 GNU General Public License v2.0 or later

9.2.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public 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.

9.2.2 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

9.2.3 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 an idea of what it does. Copyright (C) yyyy 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 (C) 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

Python Module Index

s

Index

Symbols

A

C

D

F

G