pihat

Release 0.0.6

Dec 11, 2019

Contents:

The Raspberry Pi HAT specification defines the existence of an ID EEPROM that is connected to pins 27 and 28 of the 40-way connector. This ID EEPROM contains a machine-readable description of the HAT hardware in a form that allows the Pi to automatically configure the GPIOs and load appropriate device drivers.

The `pihat` library provides a mechanism for easily extracting and manipulating the contents of the ID EEPROM. For example:

```
from pihat.eeprom import EepromDevice

with EepromDevice() as eeprom:
    print("Board UUID is %s" % eeprom.uuid)
    print("Board uses %d GPIOs" %
          sum(x.used for x in eeprom.pins))
```

CHAPTER 1

pihat package

1.1 Subpackages

1.1.1 pihat.eeprom package

Submodules

pihat.eeprom.cli module

Pi Hat command line interface

class `pihat.eeprom.cli.`**`Command`**(*args: dataclasses.InitVar*)

Bases: `object`

Extract/merge/replace Pi Hat ID EEPROM

`dump`()

Dump existing EEPROM contents to stdout

`eeprom`(***kwargs*)

Construct EEPROM object

`execute`()

Execute command

`extract`()

Extract existing EEPROM content

`merge`()

Merge with existing EEPROM content

`parser`()

Argument parser

`read_desc`()

Read description

read_fdt()
Read devicetree

replace()
Replace existing EEPROM content

write_desc(*desc*)
Write description

write_fdt(*fdt*)
Write devicetree

pihat.eeprom.constants module

Pi Hat EEPROM constants

class pihat.eeprom.constants.**EepromSignature**
Bases: enum.IntFlag

Magic signature

RPI = 1766862162

class pihat.eeprom.constants.**EepromVersion**
Bases: enum.IntFlag

Structure version

V1 = 1

class pihat.eeprom.constants.**EepromAtomType**
Bases: enum.IntFlag

Atom type

CUSTOM = 4

DTBO = 3

GPIO = 2

INFO = 1

class pihat.eeprom.constants.**EepromGpioDrive**
Bases: enum.IntEnum

GPIO bank drive

DEFAULT = 0

MA_10 = 5

MA_12 = 6

MA_14 = 7

MA_16 = 8

MA_2 = 1

MA_4 = 2

MA_6 = 3

MA_8 = 4

RESERVED_10 = 10

RESERVED_11 = 11

RESERVED_12 = 12

RESERVED_13 = 13

RESERVED_14 = 14

RESERVED_15 = 15

RESERVED_9 = 9

class pihat.eeprom.constants.**EepromGpioSlew**

Bases: enum.IntEnum

GPIO bank slew

DEFAULT = 0

LIMITED = 1

RESERVED = 3

UNLIMITED = 2

class pihat.eeprom.constants.**EepromGpioHysteresis**

Bases: enum.IntEnum

GPIO bank hysteresis

DEFAULT = 0

DISABLED = 1

ENABLED = 2

RESERVED = 3

class pihat.eeprom.constants.**EepromGpioBackPower**

Bases: enum.IntEnum

Board back powering

MA_1300 = 1

MA_2000 = 2

NONE = 0

RESERVED = 3

class pihat.eeprom.constants.**EepromGpioFunction**

Bases: enum.IntEnum

GPIO pin function

ALT0 = 4

ALT1 = 5

ALT2 = 6

ALT3 = 7

ALT4 = 3

ALT5 = 2

INPUT = 0

OUTPUT = 1

class pihat.eeprom.constants.**EepromGpioPull**

Bases: enum.IntEnum

GPIO pull direction

DEFAULT = 0

DOWN = 2

NONE = 3

UP = 1

pihat.eeprom.crc module

CRC-16 algorithm

pihat.eeprom.crc.**crc16**(*data, crc=0, table=array('H', [0, 49345, 49537, 320, 49921, 960, 640, 49729, 50689, 1728, 1920, 51009, 1280, 50625, 50305, 1088, 52225, 3264, 3456, 52545, 3840, 53185, 52865, 3648, 2560, 51905, 52097, 2880, 51457, 2496, 2176, 51265, 55297, 6336, 6528, 55617, 6912, 56257, 55937, 6720, 7680, 57025, 57217, 8000, 56577, 7616, 7296, 56385, 5120, 54465, 54657, 5440, 55041, 6080, 5760, 54849, 53761, 4800, 4992, 54081, 4352, 53697, 53377, 4160, 61441, 12480, 12672, 61761, 13056, 62401, 62081, 12864, 13824, 63169, 63361, 14144, 62721, 13760, 13440, 62529, 15360, 64705, 64897, 15680, 65281, 16320, 16000, 65089, 64001, 15040, 15232, 64321, 14592, 63937, 63617, 14400, 10240, 59585, 59777, 10560, 60161, 11200, 10880, 59969, 60929, 11968, 12160, 61249, 11520, 60865, 60545, 11328, 58369, 9408, 9600, 58689, 9984, 59329, 59009, 9792, 8704, 58049, 58241, 9024, 57601, 8640, 8320, 57409, 40961, 24768, 24960, 41281, 25344, 41921, 41601, 25152, 26112, 42689, 42881, 26432, 42241, 26048, 25728, 42049, 27648, 44225, 44417, 27968, 44801, 28608, 28288, 44609, 43521, 27328, 27520, 43841, 26880, 43457, 43137, 26688, 30720, 47297, 47489, 31040, 47873, 31680, 31360, 47681, 48641, 32448, 32640, 48961, 32000, 48577, 48257, 31808, 46081, 29888, 30080, 46401, 30464, 47041, 46721, 30272, 29184, 45761, 45953, 29504, 45313, 29120, 28800, 45121, 20480, 37057, 37249, 20800, 37633, 21440, 21120, 37441, 38401, 22208, 22400, 38721, 21760, 38337, 38017, 21568, 39937, 23744, 23936, 40257, 24320, 40897, 40577, 24128, 23040, 39617, 39809, 23360, 39169, 22976, 22656, 38977, 34817, 18624, 18816, 35137, 19200, 35777, 35457, 19008, 19968, 36545, 36737, 20288, 36097, 19904, 19584, 35905, 17408, 33985, 34177, 17728, 34561, 18368, 18048, 34369, 33281, 17088, 17280, 33601, 16640, 33217, 32897, 16448]))*

Calculate CRC-16

pihat.eeprom.device module

EEPROM device

class pihat.eeprom.device.**EepromDeviceOverlay**(*bus: int = 99, autocreate: bool = True, autoremove: bool = False, timeout: float = 2.0, interval: float = 0.1*)

Bases: object

EEPROM devicetree overlay

autocreate = True

autoremove = False

bus = 99

data
Overlay devicetree blob

directory
Overlay directory

dtbo
Overlay file

eeprom
EEPROM device path

install()
Install overlay

interval = 0.1

name
Overlay name

remove()
Remove overlay

timeout = 2.0

wait()
Wait for EEPROM device to exist

class pihat.eeprom.device.**EepromDevice**(*file: Union[None, os.PathLike, IO] = None, mode: str = 'r+b', header: pihat.eeprom.layout.EepromHeader = <factory>, atoms: List[pihat.eeprom.layout.EepromAtom] = <factory>, autoload: bool = True, autosave: bool = False, autouuid: bool = True, bus: dataclasses.InitVar = None, overlay: pihat.eeprom.device.EepromDeviceOverlay = <factory>*)

Bases: *pihat.eeprom.file.EepromFile*

EEPROM stored in an i2c EEPROM device

autouuid = True

bus = None

open(*file=None, mode=None*)
Open file

pihat.eeprom.exceptions module

Pi Hat EEPROM exceptions

exception pihat.eeprom.exceptions.**EepromValueError**

Bases: ValueError

EEPROM value error

exception pihat.eeprom.exceptions.**EepromSignatureError**
Bases: *pihat.eeprom.exceptions.EepromValueError*

EEPROM magic signature error

exception pihat.eeprom.exceptions.**EepromCrcError**
Bases: *pihat.eeprom.exceptions.EepromValueError*

EEPROM CRC error

exception pihat.eeprom.exceptions.**EepromLengthError**
Bases: *pihat.eeprom.exceptions.EepromValueError*

EEPROM length error

exception pihat.eeprom.exceptions.**EepromVerificationError**
Bases: *pihat.eeprom.exceptions.EepromValueError*

EEPROM verification error

pihat.eeprom.file module

EEPROM file

class pihat.eeprom.file.**EepromFile**(*file: Union[None, os.PathLike, IO] = None, mode: str = 'r+b', header: pihat.eeprom.layout.EepromHeader = <factory>, atoms: List[pihat.eeprom.layout.EepromAtom] = <factory>, autoload: bool = True, autosave: bool = False, autouuid: bool = False*)
Bases: *pihat.eeprom.layout.Eeprom*, pihat.eeprom.file.OpenableFile

EEPROM stored in a file

autoload = True

autosave = False

autouuid = False

load(*file=None, mode='rb'*)
Load EEPROM from file

pack(*fixup=True*)
Pack structure to binary data

save(*file=None, mode='wb', verify=False*)
Save EEPROM to file

pihat.eeprom.layout module

Pi Hat EEPROM layout

class pihat.eeprom.layout.**EepromHeader**(***kwargs*)
Bases: pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM header

eeplen
Structure/Union member

numatoms
Structure/Union member

signature
An enumerated type field within an EEPROM structure

version
An enumerated type field within an EEPROM structure

class pihat.eeprom.layout.**EepromVendorInfo**(***kwargs*)
Bases: pihat.eeprom.layout.EepromAtomData, pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM vendor information data

fixup()
Fix up fields for consistency

pack(*fixup=True*)
Pack structure to binary data

pid
Structure/Union member

pstr
A string field within an EEPROM structure

pver
Structure/Union member

type = 1

unpack(*raw*)
Unpack structure from binary data

uuid
A UUID field within an EEPROM structure

vstr
A string field within an EEPROM structure

class pihat.eeprom.layout.**EepromGpioBank**(***kwargs*)
Bases: pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM GPIO bank configuration

drive
An enumerated type field within an EEPROM structure

hysteresis
An enumerated type field within an EEPROM structure

slew
An enumerated type field within an EEPROM structure

class pihat.eeprom.layout.**EepromGpioPower**(***kwargs*)
Bases: pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM GPIO powering

back_power
An enumerated type field within an EEPROM structure

class pihat.eeprom.layout.**EepromGpioPin**(***kwargs*)
Bases: pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM GPIO pin description

function
An enumerated type field within an EEPROM structure

pull
An enumerated type field within an EEPROM structure

used
An enumerated type field within an EEPROM structure

class pihat.eeprom.layout.**EepromGpioPins**
Bases: pihat.eeprom.layout.EepromArray

EEPROM GPIO pin descriptions

class pihat.eeprom.layout.**EepromGpioMap**(***kwargs*)
Bases: pihat.eeprom.layout.EepromAtomData, pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM GPIO map

bank
Structure/Union member

pins
Structure/Union member

power
Structure/Union member

type = 2

class pihat.eeprom.layout.**EepromAtom**(***kwargs*)
Bases: pihat.eeprom.layout.EepromLittleEndianStructure

EEPROM atom

count
Structure/Union member

data = b''

fixup()
Fix up fields for consistency

pack(*fixup=True*)
Pack structure to binary data

type
An enumerated type field within an EEPROM structure

unfixed_len
Recorded length (ignoring any potential data changes)

unpack(*raw*)
Unpack structure from binary data

class pihat.eeprom.layout.**Eeprom**(*header: pihat.eeprom.layout.EepromHeader = <factory>, atoms: List[pihat.eeprom.layout.EepromAtom] = <factory>*)
Bases: pihat.eeprom.layout.EepromStructure

EEPROM content

atom(*type*)
Find first atom of a specified type

bank
An EEPROM attribute located inside the GPIO map atom

dtbo
Device tree overlay atom

fdt
An EEPROM attribute located inside the device tree blob atom

fixup()
Fix up fields for consistency

gpio
GPIO map atom

has_dtbo
Check for presence of device tree overlay atom

info
Vendor information atom

pack(*fixup=True*)
Pack structure to binary data

pid
An EEPROM attribute located inside the vendor information atom

pins
An EEPROM attribute located inside the GPIO map atom

power
An EEPROM attribute located inside the GPIO map atom

pstr
An EEPROM attribute located inside the vendor information atom

pver
An EEPROM attribute located inside the vendor information atom

unpack(*raw*)
Unpack structure from binary data

uuid
An EEPROM attribute located inside the vendor information atom

vstr
An EEPROM attribute located inside the vendor information atom

Module contents

Pi Hat EEPROM

1.2 Module contents

Pi Hat

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W