
insulaudit:decoding-carelink Documentation

Release 0.0.3

Insulaudit contributors

June 20, 2015

1	API	3
1.1	Link	3
1.2	Stick	3
1.3	Session	7
1.4	Commands	8
1.5	Download	25
1.6	History	25
1.7	lib	33
1.8	decocare	35
1.9	Modules	37
2	Indices and tables	39
	Python Module Index	41

Contents:

1.1 Link

1.1.1 link Module

```
class decocare.link.Link (port, timeout=None)
    Bases: object
    close ()
    open (newPort=False, **kwds)
    port = None
    read (c)
    readline ()
    readlines ()
    write (string)
```

1.2 Stick

This manages communications with the usb stick itself. This is really the heart of the whole package.

Given a [Link](#), with read/write methods, this module will allow using the usb stick as a radio to communicate with compatible medical devices.

1.2.1 stick Module

```
exception decocare.stick.BadCRC
    Bases: decocare.errors.StickError
decocare.stick.CRC8 (data)
class decocare.stick.InterfaceStats
    Bases: decocare.stick.StickCommand
    Abstract stats decoder.
    INTERFACE_IDX = 19
```

```
code = [5]
```

```
classmethod decode (klass, data)
```

Decode interface stats. The stick exposes 6 counters to monitor errors, crcs, naks, timeouts, rx, and tx packets. Very useful for debugging.

```
label = 'usb.interfaceStats'
```

```
parse (data)
```

```
    #>>>
```

```
class decocare.stick.LinkStatus
```

Bases: *decocare.stick.StickCommand*

Basic ACK type of command. Used to poll the modem's radio buffer. When the radio buffer is full, we can download a packet from the buffer. Otherwise, we need to be mindful of the state the radio is in. This opcode tells you the current state of the radio/stick.

```
code = [3]
```

```
parse (result)
```

```
    #>>>
```

```
reasons = ['OK']
```

```
record_error (result)
```

```
set_reason (status)
```

```
class decocare.stick.ProductInfo
```

Bases: *decocare.stick.StickCommand*

Get product info from the usb device.

Useful for identifying what kind of usb stick you've got; there are a few different kinds. Eg, European vs US regulatory domains require different frequencies for compliance.

```
SW_VER = 16
```

```
code = [4]
```

```
classmethod decode (klass, data)
```

```
classmethod decodeInterfaces (klass, L)
```

```
iface_key = {1: 'Paradigm RF', 3: 'USB'}
```

```
label = 'usb.productInfo'
```

```
parse (data)
```

```
    #>>>
```

```
rf_table = {0: '916.5Mhz', 1: '868.35Mhz', 255: '916.5Mhz'}
```

```
class decocare.stick.RadioStats
```

Bases: *decocare.stick.InterfaceStats*

Count of packets and stats on the radio side of the stick.

```
code = [5, 0]
```

```
class decocare.stick.ReadRadio (size)
```

Bases: *decocare.stick.StickCommand*

Read buffer from the radio.

Downloads a packet from the radio buffer.


```

code = [12, 0]
dl_size = 0
format ()
parse (raw)
    Detect BadCRC here. Also, look for eod set.
respond (raw)
size = 64

```

class `decocare.stick.SignalStrength`
 Bases: `decocare.stick.StickCommand`

This seems to be required to initialize communications with the usb stick. Also, you should wait until a minimum threshold is reached.

```

code = [6, 0]
parse (data)
    #>>>

```

class `decocare.stick.Stick (link)`
 Bases: `object`

The carelink usb stick acts like a buffer.

It has a variety of commands providing synchronous IO, eg, you may generally perform a read immediately after writing to it, and expect a response.

The commands operate on a local buffer used to facilitate exchanging messages over RF with the pump. RF communication with the pump happens asynchronously, requiring us to go through 3 separate phases for each message we'd like to exchange with the pumps: * transmit - send command * poll_size - loop * download - loop

Each command is usually only 3 bytes.

The protocol offers some facility for detecting and recovering from inconsistencies in the underlying transport of data, however, we are somewhat ignorant of them. The tricky bits are exactly how to recover from, eg CRC, errors that can occur. The “shape” and timing of these loops seem to mostly get the job done.

The Stick object provides a bunch of useful methods, that given a link, will represent the state of one active usb stick.

```

clear_buffer ()
    An alternative download solution. This can be helpful in scenarios where a prior run seems crashed your
    process, but the radio is still transmitting and receiving data. Running this method collects data from the
    radio until it's done receiving, more or less, at which point you should be free to try again.

```

```

static decode_hex (msg, Candidate)

```

```

download (size=None)
    Theory is to download anything and everything available off the radio buffer, and to wait if necessary.

```

```

download_packet (size)
    This is the tricky bit, where we stroke the radio and hope it gives us a buffer full of data.

```

```

interface_stats ()
    debug both sets of interface stats.

```

```

link = None

```

old_download_packet (*size*)

Naive version of downloading a packet. Didn't quite work right.

open ()

Open and get signal strength so everything is ready to go.

poll_size ()

query how many bytes are waiting in the radio buffer, ready to be downloaded

There seem to be a few sweet spots, where you want to download the data.

process ()

Working with the usb stick typically follows a pretty routine process: 1. send our opcode, get a response
2. use some custom logic, per opcode to respond to the stick's response 3. parse the response from that,
return result

This has to be done for each opcode.

product_info ()

Get the product info from the connected stick.

query (*Command*)

query - simplify the process of working with the stick, pass your command, get the result

radio_stats ()

just get radio stats.

read_status ()

Get current link status.

send_force_read (*retries=2, timeout=1*)

Pretty simple, try really hard to ensure that we've sent our bytes, and we get a response. This is probably overkill, but seems to get the job done.

signal_strength ()

just get signal strength from connected stick

transmit_packet (*command*)

Address a pump with a request.

usb_stats ()

just get usb stats.

class decocare.stick.StickCommand

Bases: object

Basic stick command

Each command is used to talk to the usb stick. The usb stick interprets the opcode, and then performs the function associated with the opcode. Altogether, the suite of opcodes that the stick responds to allows you to debug and track all packets you are sending/receiving plus allows you to send receive commands to the pump, by formatting your message into payloads with opcodes, and then letting the stick work on what you've given it. It's kind of like a modem with this funky binary interface.

code = [0]

delay = 0.001

format ()

format_c12 (*msg, a2=0, a3=0*)

label = 'example stick command'

parse (*data*)

respond (*raw*)

size = 64

class `decocare.stick.TransmitPacket` (*command*)

Bases: `decocare.stick.StickCommand`

Format a packet to send on the radio.

This commands formats a packet from usb, and shoves it into the radio buffer. The radio buffer is broadcast “over the air” so that any device sensitive to the packets you sent will respond accordingly (probably sending data back). For this reason, the serial number of the device you’d like to talk to is formatted into the packet.

calcRecordsRequired ()

code = [1, 0, 167, 1]

format ()

Formatting of the packet to be sent gets done here.

head = [1, 0, 167, 1]

parse (*results*)

respond (*raw*)

class `decocare.stick.UsbStats`

Bases: `decocare.stick.InterfaceStats`

Count of packets and stats on the usb side of the stick.

code = [5, 1]

The carelink usb stick acts like a modem. It responds to modem commands, in order to exchange data with a remote end-point. In this case, our little carelink usb modem responds to a variety of binary commands, which you must use in a co-ordinated flow to exchange data with the remote equipment. Also, in our case, the only remote equipment the carelink usb stick is known to talk to are the Me**ronic Par**igm series pumps.

The purpose of this module is to encapsulate the low level operations needed to use the stick in any practical sense. It serves as a set of guides and tutorials which should help people trying to understand or their therapy better. This module seems mostly correct, but things start to fall apart when downloading lots of data. The protocol is either sensitive to some timing somewhere that I’m unaware of, or when CRC errors or other things happen, I simply do the wrong thing to recover.

Please ask Medtronic for more information on how to learn what your device is doing.

1.3 Session

1.3.1 session Module

class `decocare.session.Pump` (*stick*, *serial*=‘208850’)

Bases: `decocare.session.Session`

execute (*command*)

power_control (*minutes*=None)

query (*Command*, ***kws*)

read_history_0 ()

read_model ()

```
    setModel (model=None, number=None)  
class decocare.session.Session (stick)  
    Bases: object  
    download ()  
    end ()  
    execute (command)  
    init (skip_power_control=False)  
    transfer ()
```

1.4 Commands

Each kind of medical device responds to different commands, but some of them respond to a suite of overlapping commands. Here's a discussion of some of the commands available.

(There are many more.)

1.4.1 commands Module

```
class decocare.commands.BaseCommand (code, descr, *args)  
    Bases: object  
    bytesPerRecord = 0  
    code = 0  
    descr = '(error)'  
    done ()  
    effectTime = 0  
    format ()  
    hexdump ()  
    maxRecords = 0  
    params = []  
    respond (data)  
    responded = False  
    retries = 2  
    timeout = 3  
  
class decocare.commands.KeypadPush (**kws)  
    Bases: decocare.commands.PumpCommand  
    classmethod ACT (klass, **kws)  
    classmethod DOWN (klass, **kws)  
    classmethod EASY (klass, **kws)  
    classmethod ESC (klass, **kws)
```

```

classmethod UP (klass, **kws)

code = 91

descr = 'Press buttons on the keypad'

maxRecords = 0

params = []

retries = 1

class decocare.commands.PowerControl (minutes=None, **kws)
    Bases: decocare.commands.PumpCommand

    >>> PowerControl(serial='665455').format() == PowerControl._test_ok
    True

code = 93

descr = 'RF Power On'

effectTime = 17

maxRecords = 0

params = [1, 10]

retries = 0

class decocare.commands.PowerControlOff (minutes=None, **kws)
    Bases: decocare.commands.PowerControl

    Here's an example where arguments clearly modify behavior.

    params = [0, 0]

class decocare.commands.PumpCommand (**kws)
    Bases: decocare.commands.BaseCommand

    allocateRawData ()

    bytesPerRecord = 64

    calcRecordsRequired ()

    data = bytearray(b'')

    effectTime = 0.5

    format ()

    getData ()

    log_name (prefix='')

    maxRecords = 1

    params = []

    retries = 2

    save (prefix='')

    serial = '208850'

class decocare.commands.PumpResume (**kws)
    Bases: decocare.commands.SetSuspend

    descr = 'Resume pump (cancel suspend)'

```

```
    params = [0]
class decocare.commands.PumpSuspend (**kws)
    Bases: decocare.commands.SetSuspend
    descr = 'Suspend pump'
    params = [1]
class decocare.commands.ReadBasalTemp (**kws)
    Bases: decocare.commands.PumpCommand
    MM511 - 120 MM512 and up - opcode 152 # strokes per basalunit = 40 - mm12, 10 in mm11
    code = 152
    descr = 'Read Temp Basal'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
class decocare.commands.ReadBatteryStatus (**kws)
    Bases: decocare.commands.PumpCommand
    code = 114
    descr = 'Read Battery Status'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
class decocare.commands.ReadContrast (**kws)
    Bases: decocare.commands.PumpCommand
    code = 195
    descr = 'Read Contrast'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
class decocare.commands.ReadCurPageNumber (**kws)
    Bases: decocare.commands.PumpCommand
    code = 157
    descr = 'Read Cur Page Number'
    getData ()
    maxRecords = 1
    pages = 'unknown'
    params = []
```

```
respond (data)
```

```
retries = 2
```

```
class decocare.commands.ReadErrorStatus (**kws)
```

```
Bases: decocare.commands.PumpCommand
```

```
>>> ReadErrorStatus(serial='665455').format() == ReadErrorStatus._test_ok
True
```

```
code = 117
```

```
descr = 'Read Error Status any current alarms set?'
```

```
maxRecords = 1
```

```
params = []
```

```
retries = 2
```

```
class decocare.commands.ReadFirmwareVersion (**kws)
```

```
Bases: decocare.commands.PumpCommand
```

```
code = 116
```

```
descr = 'Read Firmware Version'
```

```
getData ()
```

```
maxRecords = 1
```

```
params = []
```

```
retries = 2
```

```
class decocare.commands.ReadGlucoseHistory (page=None, **kws)
```

```
Bases: decocare.commands.ReadSensorHistoryData
```

```
>>> ReadGlucoseHistory(page=1).params
[0, 0, 0, 1]
>>> list(ReadGlucoseHistory(page=1).format())
[1, 0, 167, 1, 32, 136, 80, 128, 4, 0, 2, 2, 0, 154, 34, 0, 0, 0, 1, 155]
>>> ReadGlucoseHistory(page=2).params
[0, 0, 0, 2]
>>> ReadGlucoseHistory(page=3)
<ReadGlucoseHistory: size[1024]: [page] [3]: data[0]:>
>>> list(ReadGlucoseHistory(page=3).format())
[1, 0, 167, 1, 32, 136, 80, 128, 4, 0, 2, 2, 0, 154, 34, 0, 0, 0, 3, 54]
>>> ReadGlucoseHistory(page=3).params
[0, 0, 0, 3]
>>> ReadGlucoseHistory(params=[1]).params
[1]
>>> ReadGlucoseHistory(params=[2]).params
[2]
>>> ReadGlucoseHistory(params=[3]).params
[3]
```

```
code = 154
```

```
descr = 'read glucose history'
```

```
params = []
```

```
class decocare.commands.ReadHistoryData (page=None, **kws)
```

```
Bases: decocare.commands.PumpCommand
```

```
>>> ReadHistoryData(serial='208850', params=[ 0x03 ]).format() == ReadHistoryData._test_ok
True
>>> ReadHistoryData(params=[ 0x01 ]).params
[1]
>>> ReadHistoryData(params=[ 0x02 ]).params
[2]
>>> ReadHistoryData(params=[ 0x03 ]).params
[3]
>>> ReadHistoryData(page=0x01).params
[1]
>>> ReadHistoryData(page=0x02).params
[2]
>>> ReadHistoryData(page=0x03).params
[3]
```

code = 128

data = bytearray(b'')

descr = 'Read History Data'

done ()

effectTime = 0.1

getData ()

log_name (prefix='')

maxRecords = 16

page = None

params = []

respond (raw)

retries = 2

class decocare.commands.**ReadPumpID** (**kws)

Bases: *decocare.commands.PumpCommand*

code = 113

descr = 'Read Pump ID'

getData ()

maxRecords = 1

params = []

retries = 2

class decocare.commands.**ReadPumpModel** (**kws)

Bases: *decocare.commands.PumpCommand*

```
>>> ReadPumpModel(serial='665455').format() == ReadPumpModel._test_ok
True
```

code = 141

descr = 'Read Pump Model Number'

getData ()

maxRecords = 1


```

params = []
retries = 2

```

```

class decocare.commands.ReadPumpState (**kws)
    Bases: decocare.commands.PumpCommand

```

```

>>> ReadPumpState(serial='665455').format() == ReadPumpState._test_ok
True

```

```

code = 131
descr = 'Read Pump State'
maxRecords = 1
params = []
retries = 2

```

```

class decocare.commands.ReadPumpStatus (**kws)
    Bases: decocare.commands.PumpCommand

```

```

code = 206
descr = 'Read Pump Status'
getData ()
maxRecords = 1
params = []
retries = 2

```

```

class decocare.commands.ReadRTC (**kws)
    Bases: decocare.commands.PumpCommand

```

```

code = 112
descr = 'Read RTC'
getData ()
maxRecords = 1
params = []
retries = 2

```

```

class decocare.commands.ReadRadioCtrlACL (**kws)
    Bases: decocare.commands.PumpCommand

```

```

code = 118
descr = 'Read Radio ACL'
getData ()
maxRecords = 1
params = []
retries = 2

```

```

class decocare.commands.ReadRemainingInsulin (**kws)
    Bases: decocare.commands.PumpCommand

```

```

code = 115

```

```
    descr = 'Read Remaining Insulin'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
class decocare.commands.ReadSettings (**kws)
    Bases: decocare.commands.PumpCommand
    XXX: changed in MM512 to 192
    alarm (alarm)
    byte_map = {}
    code = 192
    descr = 'Read Settings'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
    temp_basal_type (data)
class decocare.commands.ReadTotalsToday (**kws)
    Bases: decocare.commands.PumpCommand
    code = 121
    descr = 'Read Totals Today'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
class decocare.commands.SetSuspend (**kws)
    Bases: decocare.commands.PumpCommand
    code = 77
    descr = 'Set Pump Suspend/Resume status'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
decocare.commands.PushEASY (**kws)
decocare.commands.PushUP (**kws)
decocare.commands.PushDOWN (**kws)
decocare.commands.PushACT (**kws)
```

```
decocare.commands.PushESC (**kws)

class decocare.commands.TempBasal (**kws)
    Bases: decocare.commands.PumpCommand

    classmethod Program (klass, rate=None, duration=None, temp=None, **kws)

    code = 76

    descr = 'Set temp basal'

    classmethod format_params (klass, rate, duration)

    classmethod format_percent_params (klass, rate, duration)

    getData ()

    params = [0, 0, 0]

    retries = 0

class decocare.commands.ManualCommand (**kws)
    Bases: decocare.commands.PumpCommand

    getData ()

    log_name (prefix='')

class decocare.commands.ReadCurGlucosePageNumber (**kws)
    Bases: decocare.commands.PumpCommand

    code = 205

    descr = 'Read Cur Glucose Page Number'

    getData ()

    maxRecords = 1

    params = []

    retries = 2

class decocare.commands.SetAutoOff (**kws)
    Bases: decocare.commands.PumpCommand

    code = 78

    maxRecords = 0

class decocare.commands.SetEnabledEasyBolus (**kws)
    Bases: decocare.commands.PumpCommand

    code = 79

    maxRecords = 0

class decocare.commands.SetBasalType (**kws)
    Bases: decocare.commands.PumpCommand

    code = 104

class decocare.commands.TempBasalPercent (**kws)
    Bases: decocare.commands.TempBasal

    code = 105

    descr = 'Set temp basal by percent'
```

```
    params = [0, 0]
    retries = 0
class decocare.commands.Bolus (**kws)
    Bases: decocare.commands.PumpCommand
    Bolus some insulin.
    XXX: Be careful please. Best trying this not connected to the pump until you trust it.
    code = 66
    descr = 'Bolus'
    getData ()
    params = []
class decocare.commands.ReadErrorStatus508 (**kws)
    Bases: decocare.commands.PumpCommand
    code = 38
    descr = 'error status'
    params = []
class decocare.commands.ReadBolusHistory (**kws)
    Bases: decocare.commands.PumpCommand
    code = 39
    descr = 'bolus history'
    params = []
class decocare.commands.ReadDailyTotals (**kws)
    Bases: decocare.commands.PumpCommand
    code = 40
    descr = '...'
    params = []
class decocare.commands.ReadPrimeBoluses (**kws)
    Bases: decocare.commands.PumpCommand
    code = 41
    descr = '...'
    params = []
class decocare.commands.ReadAlarms (**kws)
    Bases: decocare.commands.PumpCommand
    code = 42
    descr = '...'
    params = []
class decocare.commands.ReadProfileSets (**kws)
    Bases: decocare.commands.PumpCommand
    code = 43
```

```
    descr = '...'
    params = []
class decocare.commands.ReadUserEvents (**kws)
    Bases: decocare.commands.PumpCommand
    code = 44
    descr = '...'
    params = []
class decocare.commands.ReadRemoteControlID (**kws)
    Bases: decocare.commands.PumpCommand
    code = 46
    descr = '...'
    params = []
class decocare.commands.Read128KMem (**kws)
    Bases: decocare.commands.PumpCommand
    code = 55
    descr = '...'
    params = []
class decocare.commands.Read256KMem (**kws)
    Bases: decocare.commands.PumpCommand
    code = 56
    descr = '...'
    params = []
class decocare.commands.ReadBasalTemp508 (**kws)
    Bases: decocare.commands.PumpCommand
    code = 64
    descr = 'Read Temp Basal 508 (old)'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
class decocare.commands.ReadTodayTotals508 (**kws)
    Bases: decocare.commands.PumpCommand
    code = 65
    descr = 'Read Totals Today'
    getData ()
    maxRecords = 1
    params = []
    retries = 2
```

```
class decocare.commands.ReadSensorSettings (**kws)
    Bases: decocare.commands.PumpCommand
```

```
    code = 153
```

```
    descr = 'Read sensor settings'
```

```
    params = []
```

```
    retries = 2
```

```
class decocare.commands.ReadSensorHistoryData (page=None, **kws)
    Bases: decocare.commands.ReadHistoryData
```

```
class decocare.commands.ReadISIGHistory (page=None, **kws)
    Bases: decocare.commands.ReadSensorHistoryData
```

```
>>> ReadISIGHistory(page=0).params
[0, 0, 0, 0]
```

```
>>> ReadISIGHistory(page=1).params
[0, 0, 0, 1]
```

```
>>> ReadISIGHistory(page=2).params
[0, 0, 0, 2]
```

```
    code = 155
```

```
    descr = 'read ISIG history'
```

```
    maxRecords = 32
```

```
    params = []
```

```
class decocare.commands.FilterHistory (begin=None, end=None, **kws)
    Bases: decocare.commands.PumpCommand
```

```
    classmethod ISO (klass, begin=None, end=None, **kws)
```

```
    begin = None
```

```
    code = None
```

```
    end = None
```

```
    getData ()
```

```
class decocare.commands.FilterGlucoseHistory (begin=None, end=None, **kws)
    Bases: decocare.commands.FilterHistory
```

```
>>> FilterGlucoseHistory.ISO(begin='2014-04-13', end='2014-04-14').params
[7, 222, 4, 13, 7, 222, 4, 14]
```

```
    code = 168
```

```
class decocare.commands.FilterISIGHistory (begin=None, end=None, **kws)
    Bases: decocare.commands.FilterHistory
```

```
>>> FilterISIGHistory.ISO(begin='2014-04-13', end='2014-04-14').params
[7, 222, 4, 13, 7, 222, 4, 14]
```

```
    code = 169
```

```
class decocare.commands.ReadProfiles511_STD (**kws)
    Bases: decocare.commands.PumpCommand
```

```
    code = 122

class decocare.commands.ReadProfiles511_A(**kws)
    Bases: decocare.commands.PumpCommand

    code = 123

class decocare.commands.ReadProfiles511_B(**kws)
    Bases: decocare.commands.PumpCommand

    code = 124

class decocare.commands.Model511_ExperimentOP125(**kws)
    Bases: decocare.commands.PumpCommand

    code = 125

class decocare.commands.Model511_ExperimentOP126(**kws)
    Bases: decocare.commands.PumpCommand

    code = 126

class decocare.commands.ReadSettings511(**kws)
    Bases: decocare.commands.PumpCommand

    code = 127

class decocare.commands.ReadPumpTrace(**kws)
    Bases: decocare.commands.PumpCommand

    code = 163

    maxRecords = 16

class decocare.commands.ReadDetailTrace(**kws)
    Bases: decocare.commands.PumpCommand

    code = 164

    maxRecords = 16

class decocare.commands.Model511_Experiment_OP165(**kws)
    Bases: decocare.commands.PumpCommand

    code = 165

class decocare.commands.ReadNewTraceAlarm(**kws)
    Bases: decocare.commands.PumpCommand

    code = 166

    maxRecords = 16

class decocare.commands.ReadOldTraceAlarm(**kws)
    Bases: decocare.commands.PumpCommand

    code = 167

    maxRecords = 16

class decocare.commands.WriteGlucoseHistoryTimestamp(**kws)
    Bases: decocare.commands.PumpCommand

    code = 40

class decocare.commands.ReadLanguage(**kws)
    Bases: decocare.commands.PumpCommand
```

```
code = 134
```

```
class decocare.commands.ReadBolusWizardSetupStatus (**kws)
    Bases: decocare.commands.PumpCommand

    code = 135
```

```
class decocare.commands.ReadCarbUnits (**kws)
    Bases: decocare.commands.PumpCommand

    code = 136

    getData ()
```

```
class decocare.commands.ReadBGUnits (**kws)
    Bases: decocare.commands.PumpCommand

    code = 137

    getData ()
```

```
class decocare.commands.ReadCarbRatios (**kws)
    Bases: decocare.commands.PumpCommand

    code = 138

    classmethod decode_ratios (klass, data, units=0)

    getData ()

    item_size = 3

    num_items = 8
```

```
class decocare.commands.ReadCarbRatios512 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 138

    classmethod decode_ratios (klass, data, units=0)

    getData ()

    item_size = 2

    num_items = 8
```

```
class decocare.commands.ReadInsulinSensitivities (**kws)
    Bases: decocare.commands.PumpCommand

    code = 139

    getData ()

    resp_1 = bytearray(b'\x01\x00-\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00')
```

```
class decocare.commands.ReadBGTTargets (**kws)
    Bases: decocare.commands.PumpCommand

    code = 140
```

```
class decocare.commands.ReadBGTTargets515 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 159

    getData ()
```



```
class decocare.commands.ReadBGAlarmCLocks (**kws)
    Bases: decocare.commands.PumpCommand

    code = 142
```

```
class decocare.commands.ReadReservoirWarning (**kws)
    Bases: decocare.commands.PumpCommand

    code = 143
```

```
class decocare.commands.ReadBGReminderEnable (**kws)
    Bases: decocare.commands.PumpCommand

    code = 144
```

```
class decocare.commands.ReadSettings512 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 145
```

```
class decocare.commands.ReadProfile_STD512 (**kws)
    Bases: decocare.commands.PumpCommand
```

```
>>> import json
>>> schedule = ReadProfile_STD512.decode(ReadProfile_STD512._test_result_1)
>>> len(schedule)
4
>>> print json.dumps(schedule[0])
{"i": 0, "start": "00:00:00", "rate": 0.8, "minutes": 0}
>>> print json.dumps(schedule[1])
{"i": 1, "start": "06:30:00", "rate": 0.9500000000000001, "minutes": 390}
>>> print json.dumps(schedule[2])
{"i": 2, "start": "09:30:00", "rate": 1.1, "minutes": 570}
>>> print json.dumps(schedule[3])
{"i": 3, "start": "14:00:00", "rate": 0.9500000000000001, "minutes": 840}
```

```
code = 146
```

```
static decode (data)
```

```
getData ()
```

```
class decocare.commands.ReadProfile_A512 (**kws)
    Bases: decocare.commands.ReadProfile_STD512

    code = 147
```

```
class decocare.commands.ReadProfile_B512 (**kws)
    Bases: decocare.commands.ReadProfile_STD512

    code = 148
```

```
class decocare.commands.ReadLogicLinkIDS (**kws)
    Bases: decocare.commands.PumpCommand

    code = 149
```

```
class decocare.commands.Model512Experiment_OP150 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 150
```

```
class decocare.commands.ReadBGAlarmEnable (**kws)
    Bases: decocare.commands.PumpCommand
```

```
code = 151

class decocare.commands.GuardianSensorSettings (**kws)
    Bases: decocare.commands.PumpCommand

code = 209

class decocare.commands.GuardianSensorSettings (**kws)
    Bases: decocare.commands.PumpCommand

code = 209

class decocare.commands.GuardianSensorDemoGraphTimeout (**kws)
    Bases: decocare.commands.PumpCommand

code = 210

class decocare.commands.GuardianSensorAlarmSilence (**kws)
    Bases: decocare.commands.PumpCommand

code = 211

class decocare.commands.GuardianSensorRateChangeAlerts (**kws)
    Bases: decocare.commands.PumpCommand

code = 212

class decocare.commands.ReadSavedSettingsDate (**kws)
    Bases: decocare.commands.PumpCommand

code = 193

class decocare.commands.ReadBolusReminderEnable (**kws)
    Bases: decocare.commands.PumpCommand

code = 197

class decocare.commands.ReadBolusReminders (**kws)
    Bases: decocare.commands.PumpCommand

code = 198

class decocare.commands.ReadFactoryParameters (**kws)
    Bases: decocare.commands.PumpCommand

code = 199

class decocare.commands.ReadCalibrationFactor (**kws)
    Bases: decocare.commands.PumpCommand

code = 156

class decocare.commands.ReadVCNTRHistory (page=None, **kws)
    Bases: decocare.commands.ReadSensorHistoryData

code = 213

class decocare.commands.ReadOtherDevicesIDS (**kws)
    Bases: decocare.commands.PumpCommand

code = 240

class decocare.commands.PumpTraceSelect (**kws)
    Bases: decocare.commands.PumpCommand

code = 160
```

```
class decocare.commands.PumpEnableDetailTrace (**kws)
    Bases: decocare.commands.PumpTraceSelect

    params = [1]

class decocare.commands.PumpDisableDetailTrace (**kws)
    Bases: decocare.commands.PumpTraceSelect

    params = [0]

class decocare.commands.Experiment_OP161 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 161

class decocare.commands.Experiment_OP162 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 162

class decocare.commands.Model511_Experiment_OP119 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 119

class decocare.commands.Model511_Experiment_OP120 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 120

class decocare.commands.Model511_Experiment_OP121 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 121

class decocare.commands.Model511_Experiment_OP122 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 122

class decocare.commands.Model511_Experiment_OP123 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 123

class decocare.commands.Model511_Experiment_OP124 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 124

class decocare.commands.Model511_Experiment_OP125 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 125

class decocare.commands.Model511_Experiment_OP126 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 126

class decocare.commands.Model511_Experiment_OP127 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 127

class decocare.commands.Model511_Experiment_OP128 (**kws)
    Bases: decocare.commands.PumpCommand
```

```
    code = 128

class decocare.commands.Model511_Experiment_OP129 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 129

class decocare.commands.Model511_Experiment_OP130 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 130

class decocare.commands.SelectBasalProfile (**kws)
    Bases: decocare.commands.PumpCommand

    code = 74

class decocare.commands.SelectBasalProfileSTD (**kws)
    Bases: decocare.commands.SelectBasalProfile

    params = [0]

class decocare.commands.SelectBasalProfileA (**kws)
    Bases: decocare.commands.SelectBasalProfile

    params = [1]

class decocare.commands.SelectBasalProfileB (**kws)
    Bases: decocare.commands.SelectBasalProfile

    params = [2]

class decocare.commands.PumpExperiment_OP69 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 69

class decocare.commands.PumpExperiment_OP70 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 70

class decocare.commands.PumpExperiment_OP71 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 71

class decocare.commands.PumpExperiment_OP72 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 72

class decocare.commands.PumpExperiment_OP73 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 73

class decocare.commands.PumpExperiment_OP75 (**kws)
    Bases: decocare.commands.PumpCommand

    code = 75
```

1.5 Download

1.5.1 download Module

```
class decocare.download.Downloader (stick=None, device=None, log_format='logs/')
    Bases: object

    download ()
        Download a single page, copy paste from elsewhere.

    log_format = 'logs/'

class decocare.download.PageDownloader (stick=None, device=None, log_format='logs/')
    Bases: decocare.download.Downloader

    download ()

    download_page (x)

    log_format = 'logs/'

    read_current ()
```

1.6 History

1.6.1 history Module

This module provides some basic helper/formatting utilities, specifically targeted at decoding ReadHistoryData data.

```
class decocare.history.BGReceived (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    body_length = 3

    decode ()

    opcode = 63

class decocare.history.BasalProfileStart (head, larger=False)
    Bases: decocare.records.base.KnownRecord

    body_length = 3

    decode ()

    opcode = 123

class decocare.history.Battery (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    opcode = 26

class decocare.history.BigBolusWizardChange (head=bytearray(b''), model=None,
                                                larger=False)
    Bases: decocare.records.base.KnownRecord

    body_length = 143

    opcode = 90
```

```
class decocare.history.ChangeAlarmNotifyMode (head=bytearray(b''),          model=None,
                                              larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 0
    opcode = 99

class decocare.history.ChangeAudioBolus (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    decode ()
    opcode = 95

class decocare.history.ChangeBGReminderEnable (head=bytearray(b''),          model=None,
                                              larger=False)
    Bases: decocare.records.base.KnownRecord
    decode ()
    opcode = 96

class decocare.history.ChangeBasalProfile_new_profile (head=bytearray(b''),
                                                         model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 145
    decode ()
    opcode = 9

class decocare.history.ChangeBasalProfile_old_profile (head, larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 145
    decode ()
    opcode = 8

class decocare.history.ChangeBolusWizardSetup (head=bytearray(b''),          model=None,
                                              larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 40
    opcode = 79

class decocare.history.ChangeCaptureEventEnable (head=bytearray(b''),          model=None,
                                              larger=False)
    Bases: decocare.records.base.KnownRecord
    opcode = 131

class decocare.history.ChangeParadigmLinkID (head=bytearray(b''),          model=None,
                                              larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 14
    decode ()
    opcode = 60

class decocare.history.ChangeRemoteID (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    opcode = 39
class decocare.history.ChangeTempBasalType (head=bytearray(b''), model=None,
                                             larger=False)
    Bases: decocare.records.base.KnownRecord
    decode ()
    opcode = 98
class decocare.history.ChangeTime (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    opcode = 23
class decocare.history.ChangeTimeDisplay (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    opcode = 100
class decocare.history.ClearAlarm (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    opcode = 12
class decocare.history.EnableDisableRemote (head=bytearray(b''), model=None,
                                             larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 14
    opcode = 38
class decocare.history.HistoryPage (raw, model)
    Bases: decocare.history.PagedData
    clean (data)
    decode (larger=False)
class decocare.history.Ian50 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 34
    opcode = 80
class decocare.history.Ian54 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 57
    opcode = 84
class decocare.history.Ian69 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    body_length = 2
    opcode = 105
class decocare.history.IanA8 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
    head_length = 10
    opcode = 168
```

```
class decocare.history.JournalEntryMealMarker (head=bytearray(b''), model=None,
                                              larger=False)
```

Bases: decocare.records.base.KnownRecord

Capture Event > Meal marker

body_length = 2

decode ()

opcode = 64

```
class decocare.history.JournalEntryOtherMarker (head=bytearray(b''), model=None,
                                              larger=False)
```

Bases: decocare.records.base.KnownRecord

Capture Event > Other

body_length = 0

opcode = 67

```
class decocare.history.LowBattery (head=bytearray(b''), model=None, larger=False)
```

Bases: decocare.records.base.KnownRecord

opcode = 25

```
class decocare.history.LowReservoir (head=bytearray(b''), model=None, larger=False)
```

Bases: decocare.records.base.KnownRecord

```
>>> rec = LowReservoir( LowReservoir._test_1[:2] )
>>> decoded = rec.parse(LowReservoir._test_1)
>>> print str(rec)
LowReservoir 2012-12-07T11:02:43 head[2], body[0] op[0x34]
```

```
>>> print pformat(decoded)
{'amount': 20.0}
```

decode ()

opcode = 52

```
class decocare.history.MResultTotals (head, larger=False)
```

Bases: decocare.records.base.InvalidRecord

On 722 this seems like two records.

date_length = 2

date_str ()

head_length = 5

opcode = 7

parse_time ()

```
class decocare.history.Model522ResultTotals (head=bytearray(b''), model=None,
                                              larger=False)
```

Bases: decocare.records.base.KnownRecord

body_length = 40

date_length = 2

date_str ()

head_length = 1


```

    opcode = 109
    parse_time()
class decocare.history.NewTimeSet(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    opcode = 24
class decocare.history.NoDelivery(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    head_length = 4
    opcode = 6
class decocare.history.OldBolusWizardChange(head, larger=False)
    Bases: decocare.records.base.KnownRecord

    body_length = 117
    decode()
    opcode = 90
class decocare.history.PagedData(raw, model)
    Bases: object

    PagedData - context for parsing a page of cgm data.

    clean(data)
    eat_nulls(data)
class decocare.history.PumpResume(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    opcode = 31
class decocare.history.PumpSuspend(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    opcode = 30
class decocare.history.Rewind(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    opcode = 33
class decocare.history.Sara6E(head, larger=False)
    Bases: decocare.history.Model522ResultTotals

    Seems specific to 722?

    body_length = 48
    opcode = 110
class decocare.history.SelectBasalProfile(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    opcode = 20
class decocare.history.SensorAlert(head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord

    Glucose sensor alarms. The second byte of the head represents the alarm type. The third byte contains an
    alarm-specific value.

```

For example, a “Low Glucose” alarm type is: [

0x0b, # 11: Opcode 0x66, # 102: Low glucose subtype 0x50 # 80: Glucose level (For a pump configured to mg/dL)

]

alarm_types = {112: ‘Weak Signal’, 113: ‘Lost Sensor’, 115: ‘Low Glucose Predicted’, 101: ‘High Glucose’, 102: ‘Low

decode ()

head_length = 3

opcode = 11

class decocare.history.**SetAutoOff** (*head=bytearray(b’‘), model=None, larger=False*)

Bases: decocare.records.base.KnownRecord

opcode = 27

class decocare.history.**SetBolusWizardEnabled** (*head=bytearray(b’‘), model=None, larger=False*)

Bases: decocare.records.base.KnownRecord

decode ()

opcode = 45

class decocare.history.**SettingSomething57** (*head=bytearray(b’‘), model=None, larger=False*)

Bases: decocare.records.base.KnownRecord

opcode = 87

class decocare.history.**TempBasal** (*head=bytearray(b’‘), model=None, larger=False*)

Bases: decocare.records.base.KnownRecord

body_length = 1

decode ()

opcode = 51

class decocare.history.**TempBasalDuration** (*head=bytearray(b’‘), model=None, larger=False*)

Bases: decocare.records.base.KnownRecord

decode ()

opcode = 22

decocare.history.**decode_bg_targets** (*data, bg_units*)

decocare.history.**decode_carb_ratios** (*data*)

decocare.history.**decode_insulin_sensitivies** (*data*)

decocare.history.**decode_remote_id** (*msg*)

practice decoding some remote ids:

0x27

0x01 0xe2 0x40

0x03 0x42 0x2a

0x28 0x0c 0x89

0x92 0x00 0x00 0x00

```
>>> decode_remote_id(_remote_ids[0])
'123456'
```

```
>>> decode_remote_id(_remote_ids[1])
'213546'
```

```
>>> decode_remote_id(_remote_ids[2])
'821650'
```

```
decocare.history.decode_wizard_settings (data, num=8, model=None)
```

```
decocare.history.describe ()
```

```
decocare.history.describe_rate (offset, rate, q)
```

```
class decocare.history.hack51 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    opcode = 81
```

```
class decocare.history.hack52 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    opcode = 82
```

```
class decocare.history.hack53 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    body_length = 1
```

```
    opcode = 83
```

```
class decocare.history.hack55 (head, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    opcode = 85
```

```
class decocare.history.hack56 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    body_length = 5
```

```
    opcode = 86
```

```
class decocare.history.hack7d (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    body_length = 30
```

```
    opcode = 125
```

```
class decocare.history.hack82 (head=bytearray(b''), model=None, larger=False)
    Bases: decocare.records.base.KnownRecord
```

```
    body_length = 5
```

```
    opcode = 130
```

```
class decocare.history.old6c (head, model, **kwds)
    Bases: decocare.history.Model522ResultTotals
```

```
    opcode = 108
```

```
decocare.history.parse_midnight (data)
```

`decocare.history.parse_record` (*fd*, *head=bytearray(b'')*, *larger=False*, *model=None*)

Given a file-like object, and the head of a record, parse the rest of the record. Look up the type of record, read in just enough data to parse it, return the result.

class `decocare.history.questionable22` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 34

class `decocare.history.questionable23` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 35

class `decocare.history.questionable24` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 36

class `decocare.history.questionable2c` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 44

class `decocare.history.questionable3b` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 59

class `decocare.history.questionable5e` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 94

class `decocare.history.questionable61` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 97

class `decocare.history.questionable65` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 101

class `decocare.history.questionable66` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 102

class `decocare.history.questionable6f` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 111

class `decocare.history.questionable7c` (*head=bytearray(b'')*, *model=None*, *larger=False*)

Bases: `decocare.records.base.KnownRecord`

opcode = 124

`decocare.history.suggest` (*head*, *larger=False*, *model=None*)

Look in the known table of commands to find a suitable record type for this opcode.

`decocare.history.unmask_m_midnight` (*data*)

Extract date values from a series of bytes. Always returns tuple given a bytearray of at least 3 bytes.

Returns 6-tuple of scalar values year, month, day, hours, minutes, seconds.

1.7 lib

Sundry stuff to make working with data easier.

1.7.1 lib Module

This module provides some basic helper/formatting utilities.

```
>>> hexdump( bytearray( [0x00] ) )
'0000  0x00                                .'
```

```
>>> 0x00 == HighByte( 0x0F )
True
```

```
>>> 0x0F == LowByte( 0x0F )
True
```

```
>>> CRC16CCITT.compute( bytearray( [ 2, 6, 6, 3 ] ) )
16845
```

```
>>> CRC16CCITT.compute( bytearray( [ 0x02, 0x09, 0x00,
...                                0x05, 0x0D, 0x02, 0x03 ] ) )
29146
```

```
>>> BangInt( bytearray( [ 0x71, 0xDA ] ) )
29146
```

```
>>> BangInt( bytearray( [ 0x62, 0xC2 ] ) ) == CRC16CCITT.compute( bytearray( [ 2, 0x06, 0x08, 3 ] ) )
True
```

```
>>> CRC8.compute( bytearray( [ 0x00, 0xFF, 0x00 ] ) )
177
```

```
>>> BangInt( bytearray( [ 0x02, 0x02 ] ) )
514
```

```
>>> BangLong( bytearray( [ 0x0, 0x0, 0x02, 0x02 ] ) )
514L
```

```
decocare.lib.BangInt(ints)
```

```
decocare.lib.BangLong(bytez)
```

```
class decocare.lib.CRC16CCITT
```

```
    classmethod compute(klass, block)
```

```
    lookup = [0, 4129, 8258, 12387, 16516, 20645, 24774, 28903, 33032, 37161, 41290, 45419, 49548, 53677, 57806, 61935, 46575]
```

```
class decocare.lib.CRC8
```

```
    classmethod compute(klass, block)
```

```
    lookup = [0, 155, 173, 54, 193, 90, 108, 247, 25, 130, 180, 47, 216, 67, 117, 238, 50, 169, 159, 4, 243, 104, 94, 197, 43, 176, 13]
```

```
decocare.lib.HighByte(arg)
```

```
decocare.lib.LowByte(arg)
```

```
class decocare.lib.Timer
```

```
    Bases: object
```

```
    millis()
```

```
decocare.lib.basal_time(raw)
```

```
decocare.lib.decodeDC(msg)
```

```
>>> decodeDC(_decode_test_1) == bytearray(_decode_result_1)
True
>>> decodeDC(_decode_test_2) == bytearray(_decode_result_2)
True
```

```
decocare.lib.decodeDCByte(B)
```

```
decocare.lib.encodeDC(msg)
```

```
>>> encodeDC(_enc_test_1) == bytearray(_enc_result_1)
True
>>> encodeDC(_enc_test_2) == bytearray(_enc_result_2)
True
```

```
decocare.lib.epochize(dt)
```

```
decocare.lib.filter_date_today()
```

```
decocare.lib.format_filter_date(date)
```

```
>>> format_filter_date(parse.date('2014-04-09'))
[7, 222, 4, 9]
```

```
decocare.lib.hexbytes(hexstr)
```

```
decocare.lib.hexdump(src, length=8, indent=0)
```

Return a string representing the bytes in *src*, *length* bytes per line.

```
decocare.lib.int_dump(stream, indent=0)
```

```
>>> int_dump(bytearray([0x01, 0x02]))
'  1    2'
```

```
decocare.lib.makeByte(highNibble, lowNibble)
```

0 <= *highNibble* <= 15 0 <= *lowNibble* <= 15 0 <= *result* <= 255

```
class decocare.lib.parse
```

```
    static date(data)
```

```
>>> parse.date('2010-11-10T01:46:00').isoformat()
'2010-11-10T01:46:00'
```

```
>>> parse.date('2010-11-10 01:46:00').isoformat()
'2010-11-10T01:46:00'
```

```
>>> parse.date('2010-11-10 01:46PM').isoformat()
'2010-11-10T13:46:00'
```

```
>>> parse.date( '2010-11-10 13:46' ).isoformat( )
'2010-11-10T13:46:00'
```

```
>>> parse.date( '2010-11-10 1:46AM' ).isoformat( )
'2010-11-10T01:46:00'
```

1.8 decocare

1.8.1 decocare Package

decocare Package

decocare - a pure python module for serial communication with insulin pumps.

This module implements the communication protocol necessary to audit or command MM insulin pumps. Tested with 515, and 522.

See:

- <https://github.com/bewest/decoding-carelink>
- <https://github.com/bewest/insulaudit>
- <https://github.com/bewest/insulaudit-ssh-tools>
- <https://github.com/bewest/diabetes>
- <https://github.com/medevice-users/diabetes#projects>
- *Link*
- *Stick*
- *Session*
- *Commands*
- *Download*
- *History*
- *lib*

errors Module

exception `decocare.errors.AckError`

Bases: `decocare.errors.StickError`

exception `decocare.errors.BadDeviceCommError`

Bases: `decocare.errors.AckError`

exception `decocare.errors.StickError`

Bases: `exceptions.Exception`

tests Module

class `decocare.records.tests.BW722(head, model=None)`

Bases: `decocare.records.bolus.BolusWizard`

decode ()

```
class decocare.records.tests.TestSaraBolus
```

```
    bolus_1_ok = {'bg': 103, 'bg_target_low': 100, 'sensitivity': 60, 'bg_target_high': 120, 'unabsorbed_insulin_total': 0.5,
    bw_1_bytes = bytearray(b'[g\xa1Q\x0e\x04\r\rP\x00x<d\x00\x00(\x00\x00\x14\x00(x')
    bw_2_bytes = bytearray(b'[\xfc\xb7T\x0f\x04\r\rP\x00x<dX\x00\x00\x00\x00\x1c\x00<x')
    cal_bg_bytes = bytearray(b'\n\xfc\xb4T\x04\r')
    csv_breakdown = '\n 9/4/13 14:17:33,,,,,,Normal,1.0,1.0,,,,,,,,,,,,,,,,,,,,,BolusNormal\n "AMOUNT=1\n CONCENTRATIO
    hexdump = '\n 5b 67\n a1 51 0e 04 0d\n 0d 50 00 78\n 3c 64 00 00 28 00 00 14 00 28 78\n 5c 08 44 79 c0 3c 4b d0\n 01 00 28
    classmethod test_cal_bg (klass)
```

```
>>> TestSaraBolus.test_cal_bg( )
CalBGForPH 2013-09-04T15:20:52 head[2], body[0] op[0x0a]
{
    "amount": 252
}
```

```
decocare.records.tests.big_days(x=0)
# page 17, RECORD 11 >>> parse_date( big_days(0) ).isoformat( ) '2012-11-20T21:53:41'
# page 17, ~ RECORD 12 >>> parse_date( big_days(1) ).isoformat( ) '2012-11-20T22:07:38'
```

```
>>> parse_date( big_days(2) ).isoformat( )
'2012-11-20T21:53:41'
```

```
>>> parse_date( big_days(3) ).isoformat( )
'2012-11-20T22:07:38'
```

```
# page 16, RECORD ~15 >>> parse_date( big_days(4) ).isoformat( ) '2012-11-25T16:41:34'
```

```
>>> parse_date( big_days(5) ).isoformat( )
'2012-11-25T13:54:32'
```

```
# page 15 >>> parse_date( big_days(6) ).isoformat( ) '2012-11-29T20:25:37'
```

```
# page 0 >>> parse_date( big_days(7) ).isoformat( ) '2012-12-20T14:59:02'
```

```
>>> parse_date( big_days(8) ).isoformat( )
'2012-12-20T15:28:25'
```

```
decocare.records.tests.decode_wizard(data)
BYTE 01: 02: 03: 04: 05: 06: 07: 08: 09: 10: 12: 13: 14: 15: 16: 17: 18: 19: 20: 21: 22:
```

```
decocare.records.tests.dictlines(d)
```

```
decocare.records.tests.unsolved_bolus_wizard()
# >>> unsolved_bolus_wizard()
```

1.8.2 Modules

Each module can be run and tested indepdently. They will refuse to run if the tests do not pass.

1.9 Modules

The *decocare Package* implements the Medtronic Carelink protocol, sometimes known as ComLink2. The implementation is built using python modules as building blocks, each module allowing us to prove some specific part of the protocol.

Decoding-carelink implements a python module named decocare. The python module allows learning more about how Medtronic Insulin pumps and related devices work.

- *decocare Package*

Note: This information was gleaned through extensive testing and analysis of the behavior of actual devices, not from documentation. Please call and ask your vendor for the technical information needed to properly maintain these projects.

Indices and tables

- `genindex`
- `modindex`
- `search`

d

- `decocare.__init__`, 35
- `decocare.commands`, 8
- `decocare.download`, 25
- `decocare.errors`, 35
- `decocare.history`, 25
- `decocare.lib`, 33
- `decocare.link`, 3
- `decocare.records.tests`, 35
- `decocare.session`, 7
- `decocare.stick`, 3

A

AckError, 35
 ACT() (decocare.commands.KeypadPush class method), 8
 alarm() (decocare.commands.ReadSettings method), 14
 alarm_types (decocare.history.SensorAlert attribute), 30
 allocateRawData() (decocare.commands.PumpCommand method), 9

B

BadCRC, 3
 BadDeviceCommError, 35
 BangInt() (in module decocare.lib), 33
 BangLong() (in module decocare.lib), 33
 basal_time() (in module decocare.lib), 34
 BasalProfileStart (class in decocare.history), 25
 BaseCommand (class in decocare.commands), 8
 Battery (class in decocare.history), 25
 begin (decocare.commands.FilterHistory attribute), 18
 BGReceived (class in decocare.history), 25
 big_days() (in module decocare.records.tests), 36
 BigBolusWizardChange (class in decocare.history), 25
 body_length (decocare.history.BasalProfileStart attribute), 25
 body_length (decocare.history.BGReceived attribute), 25
 body_length (decocare.history.BigBolusWizardChange attribute), 25
 body_length (decocare.history.ChangeAlarmNotifyMode attribute), 26
 body_length (decocare.history.ChangeBasalProfile_new_profile attribute), 26
 body_length (decocare.history.ChangeBasalProfile_old_profile attribute), 26
 body_length (decocare.history.ChangeBolusWizardSetup attribute), 26
 body_length (decocare.history.ChangeParadigmLinkID attribute), 26
 body_length (decocare.history.EnableDisableRemote attribute), 27
 body_length (decocare.history.hack53 attribute), 31

body_length (decocare.history.hack56 attribute), 31
 body_length (decocare.history.hack7d attribute), 31
 body_length (decocare.history.hack82 attribute), 31
 body_length (decocare.history.Ian50 attribute), 27
 body_length (decocare.history.Ian54 attribute), 27
 body_length (decocare.history.Ian69 attribute), 27
 body_length (decocare.history.JournalEntryMealMarker attribute), 28
 body_length (decocare.history.JournalEntryOtherMarker attribute), 28
 body_length (decocare.history.Model522ResultTotals attribute), 28
 body_length (decocare.history.OldBolusWizardChange attribute), 29
 body_length (decocare.history.Sara6E attribute), 29
 body_length (decocare.history.TempBasal attribute), 30
 Bolus (class in decocare.commands), 16
 bolus_1_ok (decocare.records.tests.TestSaraBolus attribute), 36
 BW722 (class in decocare.records.tests), 35
 bw_1_bytes (decocare.records.tests.TestSaraBolus attribute), 36
 bw_2_bytes (decocare.records.tests.TestSaraBolus attribute), 36
 byte_map (decocare.commands.ReadSettings attribute), 14
 bytesPerRecord (decocare.commands.BaseCommand attribute), 8
 bytesPerRecord (decocare.commands.PumpCommand attribute), 9

C

cal_bg_bytes (decocare.records.tests.TestSaraBolus attribute), 36
 calcRecordsRequired() (decocare.commands.PumpCommand method), 9
 calcRecordsRequired() (decocare.stick.TransmitPacket method), 7
 ChangeAlarmNotifyMode (class in decocare.history), 25
 ChangeAudioBolus (class in decocare.history), 26

ChangeBasalProfile_new_profile (class in decocare.history), 26
ChangeBasalProfile_old_profile (class in decocare.history), 26
ChangeBGReminderEnable (class in decocare.history), 26
ChangeBolusWizardSetup (class in decocare.history), 26
ChangeCaptureEventEnable (class in decocare.history), 26
ChangeParadigmLinkID (class in decocare.history), 26
ChangeRemoteID (class in decocare.history), 26
ChangeTempBasalType (class in decocare.history), 27
ChangeTime (class in decocare.history), 27
ChangeTimeDisplay (class in decocare.history), 27
clean() (decocare.history.HistoryPage method), 27
clean() (decocare.history.PagedData method), 29
clear_buffer() (decocare.stick.Stick method), 5
ClearAlarm (class in decocare.history), 27
close() (decocare.link.Link method), 3
code (decocare.commands.BaseCommand attribute), 8
code (decocare.commands.Bolus attribute), 16
code (decocare.commands.Experiment_OP161 attribute), 23
code (decocare.commands.Experiment_OP162 attribute), 23
code (decocare.commands.FilterGlucoseHistory attribute), 18
code (decocare.commands.FilterHistory attribute), 18
code (decocare.commands.FilterLSIGHistory attribute), 18
code (decocare.commands.GuardianSensorAlarmSilence attribute), 22
code (decocare.commands.GuardianSensorDemoGraphTime attribute), 22
code (decocare.commands.GuardianSensorRateChangeAlert attribute), 22
code (decocare.commands.GuardianSensorSettings attribute), 22
code (decocare.commands.KeypadPush attribute), 9
code (decocare.commands.Model511_Experiment_OP119 attribute), 23
code (decocare.commands.Model511_Experiment_OP120 attribute), 23
code (decocare.commands.Model511_Experiment_OP121 attribute), 23
code (decocare.commands.Model511_Experiment_OP122 attribute), 23
code (decocare.commands.Model511_Experiment_OP123 attribute), 23
code (decocare.commands.Model511_Experiment_OP124 attribute), 23
code (decocare.commands.Model511_Experiment_OP125 attribute), 23
code (decocare.commands.Model511_Experiment_OP126 attribute), 23
code (decocare.commands.Model511_Experiment_OP127 attribute), 23
code (decocare.commands.Model511_Experiment_OP128 attribute), 23
code (decocare.commands.Model511_Experiment_OP129 attribute), 24
code (decocare.commands.Model511_Experiment_OP130 attribute), 24
code (decocare.commands.Model511_Experiment_OP165 attribute), 19
code (decocare.commands.Model511_Experiment_OP125 attribute), 19
code (decocare.commands.Model511_Experiment_OP126 attribute), 19
code (decocare.commands.Model512Experiment_OP150 attribute), 21
code (decocare.commands.PowerControl attribute), 9
code (decocare.commands.PumpExperiment_OP69 attribute), 24
code (decocare.commands.PumpExperiment_OP70 attribute), 24
code (decocare.commands.PumpExperiment_OP71 attribute), 24
code (decocare.commands.PumpExperiment_OP72 attribute), 24
code (decocare.commands.PumpExperiment_OP73 attribute), 24
code (decocare.commands.PumpExperiment_OP75 attribute), 24
code (decocare.commands.PumpTraceSelect attribute), 22
code (decocare.commands.Read128KMem attribute), 17
code (decocare.commands.Read256KMem attribute), 17
code (decocare.commands.ReadAlarms attribute), 16
code (decocare.commands.ReadBasalTemp attribute), 10
code (decocare.commands.ReadBasalTemp508 attribute), 17
code (decocare.commands.ReadBatteryStatus attribute), 10
code (decocare.commands.ReadBGAlarmClocks attribute), 21
code (decocare.commands.ReadBGAlarmEnable attribute), 21
code (decocare.commands.ReadBGReminderEnable attribute), 21
code (decocare.commands.ReadBGTargets attribute), 20
code (decocare.commands.ReadBGTargets515 attribute), 20
code (decocare.commands.ReadBGUnits attribute), 20
code (decocare.commands.ReadBolusHistory attribute), 16
code (decocare.commands.ReadBolusReminderEnable attribute), 22

code (decocare.commands.ReadBolusReminders attribute), 22

code (decocare.commands.ReadBolusWizardSetupStatus attribute), 20

code (decocare.commands.ReadCalibrationFactor attribute), 22

code (decocare.commands.ReadCarbRatios attribute), 20

code (decocare.commands.ReadCarbRatios512 attribute), 20

code (decocare.commands.ReadCarbUnits attribute), 20

code (decocare.commands.ReadContrast attribute), 10

code (decocare.commands.ReadCurGlucosePageNumber attribute), 15

code (decocare.commands.ReadCurPageNumber attribute), 10

code (decocare.commands.ReadDailyTotals attribute), 16

code (decocare.commands.ReadDetailTrace attribute), 19

code (decocare.commands.ReadErrorStatus attribute), 11

code (decocare.commands.ReadErrorStatus508 attribute), 16

code (decocare.commands.ReadFactoryParameters attribute), 22

code (decocare.commands.ReadFirmwareVersion attribute), 11

code (decocare.commands.ReadGlucoseHistory attribute), 11

code (decocare.commands.ReadHistoryData attribute), 12

code (decocare.commands.ReadInsulinSensitivities attribute), 20

code (decocare.commands.ReadISIGHistory attribute), 18

code (decocare.commands.ReadLanguage attribute), 19

code (decocare.commands.ReadLogicLinkIDS attribute), 21

code (decocare.commands.ReadNewTraceAlarm attribute), 19

code (decocare.commands.ReadOldTraceAlarm attribute), 19

code (decocare.commands.ReadOtherDevicesIDS attribute), 22

code (decocare.commands.ReadPrimeBoluses attribute), 16

code (decocare.commands.ReadProfile_A512 attribute), 21

code (decocare.commands.ReadProfile_B512 attribute), 21

code (decocare.commands.ReadProfile_STD512 attribute), 21

code (decocare.commands.ReadProfiles511_A attribute), 19

code (decocare.commands.ReadProfiles511_B attribute), 19

code (decocare.commands.ReadProfiles511_STD attribute), 18

code (decocare.commands.ReadProfileSets attribute), 16

code (decocare.commands.ReadPumpID attribute), 12

code (decocare.commands.ReadPumpModel attribute), 12

code (decocare.commands.ReadPumpState attribute), 13

code (decocare.commands.ReadPumpStatus attribute), 13

code (decocare.commands.ReadPumpTrace attribute), 19

code (decocare.commands.ReadRadioCtrlACL attribute), 13

code (decocare.commands.ReadRemainingInsulin attribute), 13

code (decocare.commands.ReadRemoteControlID attribute), 17

code (decocare.commands.ReadReservoirWarning attribute), 21

code (decocare.commands.ReadRTC attribute), 13

code (decocare.commands.ReadSavedSettingsDate attribute), 22

code (decocare.commands.ReadSensorSettings attribute), 18

code (decocare.commands.ReadSettings attribute), 14

code (decocare.commands.ReadSettings511 attribute), 19

code (decocare.commands.ReadSettings512 attribute), 21

code (decocare.commands.ReadTodayTotals508 attribute), 17

code (decocare.commands.ReadTotalsToday attribute), 14

code (decocare.commands.ReadUserEvents attribute), 17

code (decocare.commands.ReadVCNTRHistory attribute), 22

code (decocare.commands.SelectBasalProfile attribute), 24

code (decocare.commands.SetAutoOff attribute), 15

code (decocare.commands.SetBasalType attribute), 15

code (decocare.commands.SetEnabledEasyBolus attribute), 15

code (decocare.commands.SetSuspend attribute), 14

code (decocare.commands.TempBasal attribute), 15

code (decocare.commands.TempBasalPercent attribute), 15

code (decocare.commands.WriteGlucoseHistoryTimestamp attribute), 19

code (decocare.stick.InterfaceStats attribute), 3

code (decocare.stick.LinkStatus attribute), 4

code (decocare.stick.ProductInfo attribute), 4

code (decocare.stick.RadioStats attribute), 4

code (decocare.stick.ReadRadio attribute), 4

code (decocare.stick.SignalStrength attribute), 5

code (decocare.stick.StickCommand attribute), 6

code (decocare.stick.TransmitPacket attribute), 7

code (decocare.stick.UsbStats attribute), 7

compute() (decocare.lib.CRC16CCITT class method), 33

compute() (decocare.lib.CRC8 class method), 33

CRC16CCITT (class in decocare.lib), 33

CRC8 (class in decocare.lib), 33

CRC8() (in module decocare.stick), 3

csv_breakdown (decocare.records.tests.TestSaraBolus attribute), 36

D

data (decocare.commands.PumpCommand attribute), 9

data (decocare.commands.ReadHistoryData attribute), 12

date() (decocare.lib.parse static method), 34

date_length (decocare.history.Model522ResultTotals attribute), 28

date_length (decocare.history.MResultTotals attribute), 28

date_str() (decocare.history.Model522ResultTotals method), 28

date_str() (decocare.history.MResultTotals method), 28

decocare.__init__ (module), 35

decocare.commands (module), 8

decocare.download (module), 25

decocare.errors (module), 35

decocare.history (module), 25

decocare.lib (module), 33

decocare.link (module), 3

decocare.records.tests (module), 35

decocare.session (module), 7

decocare.stick (module), 3

decode() (decocare.commands.ReadProfile_STD512 static method), 21

decode() (decocare.history.BasalProfileStart method), 25

decode() (decocare.history.BGReceived method), 25

decode() (decocare.history.ChangeAudioBolus method), 26

decode() (decocare.history.ChangeBasalProfile_new_profile method), 26

decode() (decocare.history.ChangeBasalProfile_old_profile method), 26

decode() (decocare.history.ChangeBGReminderEnable method), 26

decode() (decocare.history.ChangeParadigmLinkID method), 26

decode() (decocare.history.ChangeTempBasalType method), 27

decode() (decocare.history.HistoryPage method), 27

decode() (decocare.history.JournalEntryMealMarker method), 28

decode() (decocare.history.LowReservoir method), 28

decode() (decocare.history.OldBolusWizardChange method), 29

decode() (decocare.history.SensorAlert method), 30

decode() (decocare.history.SetBolusWizardEnabled method), 30

decode() (decocare.history.TempBasal method), 30

decode() (decocare.history.TempBasalDuration method), 30

decode() (decocare.records.tests.BW722 method), 35

decode() (decocare.stick.InterfaceStats class method), 4

decode() (decocare.stick.ProductInfo class method), 4

decode_bg_targets() (in module decocare.history), 30

decode_carb_ratios() (in module decocare.history), 30

decode_hex() (decocare.stick.Stick static method), 5

decode_insulin_sensitivies() (in module decocare.history), 30

decode_ratios() (decocare.commands.ReadCarbRatios class method), 20

decode_ratios() (decocare.commands.ReadCarbRatios512 class method), 20

decode_remote_id() (in module decocare.history), 30

decode_wizard() (in module decocare.records.tests), 36

decode_wizard_settings() (in module decocare.history), 31

decodeDC() (in module decocare.lib), 34

decodeDCByte() (in module decocare.lib), 34

decodeInterfaces() (decocare.stick.ProductInfo class method), 4

delay (decocare.stick.StickCommand attribute), 6

descr (decocare.commands.BaseCommand attribute), 8

descr (decocare.commands.Bolus attribute), 16

descr (decocare.commands.KeypadPush attribute), 9

descr (decocare.commands.PowerControl attribute), 9

descr (decocare.commands.PumpResume attribute), 9

descr (decocare.commands.PumpSuspend attribute), 10

descr (decocare.commands.Read128KMem attribute), 17

descr (decocare.commands.Read256KMem attribute), 17

descr (decocare.commands.ReadAlarms attribute), 16

descr (decocare.commands.ReadBasalTemp attribute), 10

descr (decocare.commands.ReadBasalTemp508 attribute), 17

descr (decocare.commands.ReadBatteryStatus attribute), 10

descr (decocare.commands.ReadBolusHistory attribute), 16

descr (decocare.commands.ReadContrast attribute), 10

descr (decocare.commands.ReadCurGlucosePageNumber attribute), 15

descr (decocare.commands.ReadCurPageNumber attribute), 10

descr (decocare.commands.ReadDailyTotals attribute), 16

descr (decocare.commands.ReadErrorStatus attribute), 11

descr (decocare.commands.ReadErrorStatus508 attribute), 16

descr (decocare.commands.ReadFirmwareVersion attribute), 11

descr (decocare.commands.ReadGlucoseHistory attribute), 11

descr (decocare.commands.ReadHistoryData attribute),

12
 descr (decocare.commands.ReadISIGHistory attribute), 18
 descr (decocare.commands.ReadPrimeBoluses attribute), 16
 descr (decocare.commands.ReadProfileSets attribute), 16
 descr (decocare.commands.ReadPumpID attribute), 12
 descr (decocare.commands.ReadPumpModel attribute), 12
 descr (decocare.commands.ReadPumpState attribute), 13
 descr (decocare.commands.ReadPumpStatus attribute), 13
 descr (decocare.commands.ReadRadioCtrlACL attribute), 13
 descr (decocare.commands.ReadRemainingInsulin attribute), 13
 descr (decocare.commands.ReadRemoteControlID attribute), 17
 descr (decocare.commands.ReadRTC attribute), 13
 descr (decocare.commands.ReadSensorSettings attribute), 18
 descr (decocare.commands.ReadSettings attribute), 14
 descr (decocare.commands.ReadTodayTotals508 attribute), 17
 descr (decocare.commands.ReadTotalsToday attribute), 14
 descr (decocare.commands.ReadUserEvents attribute), 17
 descr (decocare.commands.SetSuspend attribute), 14
 descr (decocare.commands.TempBasal attribute), 15
 descr (decocare.commands.TempBasalPercent attribute), 15
 describe() (in module decocare.history), 31
 describe_rate() (in module decocare.history), 31
 dictlines() (in module decocare.records.tests), 36
 dl_size (decocare.stick.ReadRadio attribute), 5
 done() (decocare.commands.BaseCommand method), 8
 done() (decocare.commands.ReadHistoryData method), 12
 DOWN() (decocare.commands.KeypadPush class method), 8
 download() (decocare.download.Downloader method), 25
 download() (decocare.download.PageDownloader method), 25
 download() (decocare.session.Session method), 8
 download() (decocare.stick.Stick method), 5
 download_packet() (decocare.stick.Stick method), 5
 download_page() (decocare.download.PageDownloader method), 25
 Downloader (class in decocare.download), 25

E

EASY() (decocare.commands.KeypadPush class method), 8
 eat_nulls() (decocare.history.PagedData method), 29

effectTime (decocare.commands.BaseCommand attribute), 8
 effectTime (decocare.commands.PowerControl attribute), 9
 effectTime (decocare.commands.PumpCommand attribute), 9
 effectTime (decocare.commands.ReadHistoryData attribute), 12
 EnableDisableRemote (class in decocare.history), 27
 encodeDC() (in module decocare.lib), 34
 end (decocare.commands.FilterHistory attribute), 18
 end() (decocare.session.Session method), 8
 epochize() (in module decocare.lib), 34
 ESC() (decocare.commands.KeypadPush class method), 8
 execute() (decocare.session.Pump method), 7
 execute() (decocare.session.Session method), 8
 Experiment_OP161 (class in decocare.commands), 23
 Experiment_OP162 (class in decocare.commands), 23

F

filter_date_today() (in module decocare.lib), 34
 FilterGlucoseHistory (class in decocare.commands), 18
 FilterHistory (class in decocare.commands), 18
 FilterISIGHistory (class in decocare.commands), 18
 format() (decocare.commands.BaseCommand method), 8
 format() (decocare.commands.PumpCommand method), 9
 format() (decocare.stick.ReadRadio method), 5
 format() (decocare.stick.StickCommand method), 6
 format() (decocare.stick.TransmitPacket method), 7
 format_cl2() (decocare.stick.StickCommand method), 6
 format_filter_date() (in module decocare.lib), 34
 format_params() (decocare.commands.TempBasal class method), 15
 format_percent_params() (decocare.commands.TempBasal class method), 15

G

getData() (decocare.commands.Bolus method), 16
 getData() (decocare.commands.FilterHistory method), 18
 getData() (decocare.commands.ManualCommand method), 15
 getData() (decocare.commands.PumpCommand method), 9
 getData() (decocare.commands.ReadBasalTemp method), 10
 getData() (decocare.commands.ReadBasalTemp508 method), 17
 getData() (decocare.commands.ReadBatteryStatus method), 10
 getData() (decocare.commands.ReadBGTargets515 method), 20

`getData()` (decocare.commands.ReadBGUnits method), 20

`getData()` (decocare.commands.ReadCarbRatios method), 20

`getData()` (decocare.commands.ReadCarbRatios512 method), 20

`getData()` (decocare.commands.ReadCarbUnits method), 20

`getData()` (decocare.commands.ReadContrast method), 10

`getData()` (decocare.commands.ReadCurGlucosePageNumber method), 15

`getData()` (decocare.commands.ReadCurPageNumber method), 10

`getData()` (decocare.commands.ReadFirmwareVersion method), 11

`getData()` (decocare.commands.ReadHistoryData method), 12

`getData()` (decocare.commands.ReadInsulinSensitivities method), 20

`getData()` (decocare.commands.ReadProfile_STD512 method), 21

`getData()` (decocare.commands.ReadPumpID method), 12

`getData()` (decocare.commands.ReadPumpModel method), 12

`getData()` (decocare.commands.ReadPumpStatus method), 13

`getData()` (decocare.commands.ReadRadioCtrlACL method), 13

`getData()` (decocare.commands.ReadRemainingInsulin method), 14

`getData()` (decocare.commands.ReadRTC method), 13

`getData()` (decocare.commands.ReadSettings method), 14

`getData()` (decocare.commands.ReadTodayTotals508 method), 17

`getData()` (decocare.commands.ReadTotalsToday method), 14

`getData()` (decocare.commands.SetSuspend method), 14

`getData()` (decocare.commands.TempBasal method), 15

`GuardianSensorAlarmSilence` (class in decocare.commands), 22

`GuardianSensorDemoGraphTimeout` (class in decocare.commands), 22

`GuardianSensorRateChangeAlerts` (class in decocare.commands), 22

`GuardianSensorSettings` (class in decocare.commands), 22

H

`hack51` (class in decocare.history), 31

`hack52` (class in decocare.history), 31

`hack53` (class in decocare.history), 31

`hack55` (class in decocare.history), 31

`hack56` (class in decocare.history), 31

`hack7d` (class in decocare.history), 31

`hack82` (class in decocare.history), 31

`head` (decocare.stick.TransmitPacket attribute), 7

`head_length` (decocare.history.IanA8 attribute), 27

`head_length` (decocare.history.Model522ResultTotals attribute), 28

`head_length` (decocare.history.MResultTotals attribute), 28

`head_length` (decocare.history.NoDelivery attribute), 29

`head_length` (decocare.history.SensorAlert attribute), 30

`hexbytes()` (in module decocare.lib), 34

`hexdump` (decocare.records.tests.TestSaraBolus attribute), 36

`hexdump()` (decocare.commands.BaseCommand method), 8

`hexdump()` (in module decocare.lib), 34

`HighByte()` (in module decocare.lib), 33

`HistoryPage` (class in decocare.history), 27

I

`Ian50` (class in decocare.history), 27

`Ian54` (class in decocare.history), 27

`Ian69` (class in decocare.history), 27

`IanA8` (class in decocare.history), 27

`iface_key` (decocare.stick.ProductInfo attribute), 4

`init()` (decocare.session.Session method), 8

`int_dump()` (in module decocare.lib), 34

`INTERFACE_IDX` (decocare.stick.InterfaceStats attribute), 3

`interface_stats()` (decocare.stick.Stick method), 5

`InterfaceStats` (class in decocare.stick), 3

`ISO()` (decocare.commands.FilterHistory class method), 18

`item_size` (decocare.commands.ReadCarbRatios attribute), 20

`item_size` (decocare.commands.ReadCarbRatios512 attribute), 20

J

`JournalEntryMealMarker` (class in decocare.history), 27

`JournalEntryOtherMarker` (class in decocare.history), 28

K

`KeypadPush` (class in decocare.commands), 8

L

`label` (decocare.stick.InterfaceStats attribute), 4

`label` (decocare.stick.ProductInfo attribute), 4

`label` (decocare.stick.StickCommand attribute), 6

`Link` (class in decocare.link), 3

`link` (decocare.stick.Stick attribute), 5

`LinkStatus` (class in decocare.stick), 4

- log_format (decocare.download.Downloader attribute), 25
- log_format (decocare.download.PageDownloader attribute), 25
- log_name() (decocare.commands.ManualCommand method), 15
- log_name() (decocare.commands.PumpCommand method), 9
- log_name() (decocare.commands.ReadHistoryData method), 12
- lookup (decocare.lib.CRC16CCITT attribute), 33
- lookup (decocare.lib.CRC8 attribute), 33
- LowBattery (class in decocare.history), 28
- LowByte() (in module decocare.lib), 33
- LowReservoir (class in decocare.history), 28
- ## M
- makeByte() (in module decocare.lib), 34
- ManualCommand (class in decocare.commands), 15
- maxRecords (decocare.commands.BaseCommand attribute), 8
- maxRecords (decocare.commands.KeypadPush attribute), 9
- maxRecords (decocare.commands.PowerControl attribute), 9
- maxRecords (decocare.commands.PumpCommand attribute), 9
- maxRecords (decocare.commands.ReadBasalTemp attribute), 10
- maxRecords (decocare.commands.ReadBasalTemp508 attribute), 17
- maxRecords (decocare.commands.ReadBatteryStatus attribute), 10
- maxRecords (decocare.commands.ReadContrast attribute), 10
- maxRecords (decocare.commands.ReadCurGlucosePageNumber attribute), 15
- maxRecords (decocare.commands.ReadCurPageNumber attribute), 10
- maxRecords (decocare.commands.ReadDetailTrace attribute), 19
- maxRecords (decocare.commands.ReadErrorStatus attribute), 11
- maxRecords (decocare.commands.ReadFirmwareVersion attribute), 11
- maxRecords (decocare.commands.ReadHistoryData attribute), 12
- maxRecords (decocare.commands.ReadISIGHistory attribute), 18
- maxRecords (decocare.commands.ReadNewTraceAlarm attribute), 19
- maxRecords (decocare.commands.ReadOldTraceAlarm attribute), 19
- maxRecords (decocare.commands.ReadPumpID attribute), 12
- maxRecords (decocare.commands.ReadPumpModel attribute), 12
- maxRecords (decocare.commands.ReadPumpState attribute), 13
- maxRecords (decocare.commands.ReadPumpStatus attribute), 13
- maxRecords (decocare.commands.ReadPumpTrace attribute), 19
- maxRecords (decocare.commands.ReadRadioCtrlACL attribute), 13
- maxRecords (decocare.commands.ReadRemainingInsulin attribute), 14
- maxRecords (decocare.commands.ReadRTC attribute), 13
- maxRecords (decocare.commands.ReadSettings attribute), 14
- maxRecords (decocare.commands.ReadTodayTotals508 attribute), 17
- maxRecords (decocare.commands.ReadTotalsToday attribute), 14
- maxRecords (decocare.commands.SetAutoOff attribute), 15
- maxRecords (decocare.commands.SetEnabledEasyBolus attribute), 15
- maxRecords (decocare.commands.SetSuspend attribute), 14
- millis() (decocare.lib.Timer method), 34
- Model511_Experiment_OP119 (class in decocare.commands), 23
- Model511_Experiment_OP120 (class in decocare.commands), 23
- Model511_Experiment_OP121 (class in decocare.commands), 23
- Model511_Experiment_OP122 (class in decocare.commands), 23
- Model511_Experiment_OP123 (class in decocare.commands), 23
- Model511_Experiment_OP124 (class in decocare.commands), 23
- Model511_Experiment_OP125 (class in decocare.commands), 23
- Model511_Experiment_OP126 (class in decocare.commands), 23
- Model511_Experiment_OP127 (class in decocare.commands), 23
- Model511_Experiment_OP128 (class in decocare.commands), 23
- Model511_Experiment_OP129 (class in decocare.commands), 24
- Model511_Experiment_OP130 (class in decocare.commands), 24
- Model511_Experiment_OP165 (class in decocare.commands), 24

care.commands), 19
 Model511_ExperimentOP125 (class in decocare.commands), 19
 Model511_ExperimentOP126 (class in decocare.commands), 19
 Model512Experiment_OP150 (class in decocare.commands), 21
 Model522ResultTotals (class in decocare.history), 28
 MResultTotals (class in decocare.history), 28

N

NewTimeSet (class in decocare.history), 29
 NoDelivery (class in decocare.history), 29
 num_items (decocare.commands.ReadCarbRatios attribute), 20
 num_items (decocare.commands.ReadCarbRatios512 attribute), 20

O

old6c (class in decocare.history), 31
 old_download_packet() (decocare.stick.Stick method), 5
 OldBolusWizardChange (class in decocare.history), 29
 opcode (decocare.history.BasalProfileStart attribute), 25
 opcode (decocare.history.Battery attribute), 25
 opcode (decocare.history.BGReceived attribute), 25
 opcode (decocare.history.BigBolusWizardChange attribute), 25
 opcode (decocare.history.ChangeAlarmNotifyMode attribute), 26
 opcode (decocare.history.ChangeAudioBolus attribute), 26
 opcode (decocare.history.ChangeBasalProfile_new_profile attribute), 26
 opcode (decocare.history.ChangeBasalProfile_old_profile attribute), 26
 opcode (decocare.history.ChangeBGReminderEnable attribute), 26
 opcode (decocare.history.ChangeBolusWizardSetup attribute), 26
 opcode (decocare.history.ChangeCaptureEventEnable attribute), 26
 opcode (decocare.history.ChangeParadigmLinkID attribute), 26
 opcode (decocare.history.ChangeRemoteID attribute), 26
 opcode (decocare.history.ChangeTempBasalType attribute), 27
 opcode (decocare.history.ChangeTime attribute), 27
 opcode (decocare.history.ChangeTimeDisplay attribute), 27
 opcode (decocare.history.ClearAlarm attribute), 27
 opcode (decocare.history.EnableDisableRemote attribute), 27
 opcode (decocare.history.hack51 attribute), 31
 opcode (decocare.history.hack52 attribute), 31

opcode (decocare.history.hack53 attribute), 31
 opcode (decocare.history.hack55 attribute), 31
 opcode (decocare.history.hack56 attribute), 31
 opcode (decocare.history.hack7d attribute), 31
 opcode (decocare.history.hack82 attribute), 31
 opcode (decocare.history.Ian50 attribute), 27
 opcode (decocare.history.Ian54 attribute), 27
 opcode (decocare.history.Ian69 attribute), 27
 opcode (decocare.history.IanA8 attribute), 27
 opcode (decocare.history.JournalEntryMealMarker attribute), 28
 opcode (decocare.history.JournalEntryOtherMarker attribute), 28
 opcode (decocare.history.LowBattery attribute), 28
 opcode (decocare.history.LowReservoir attribute), 28
 opcode (decocare.history.Model522ResultTotals attribute), 29
 opcode (decocare.history.MResultTotals attribute), 28
 opcode (decocare.history.NewTimeSet attribute), 29
 opcode (decocare.history.NoDelivery attribute), 29
 opcode (decocare.history.old6c attribute), 31
 opcode (decocare.history.OldBolusWizardChange attribute), 29
 opcode (decocare.history.PumpResume attribute), 29
 opcode (decocare.history.PumpSuspend attribute), 29
 opcode (decocare.history.questionable22 attribute), 32
 opcode (decocare.history.questionable23 attribute), 32
 opcode (decocare.history.questionable24 attribute), 32
 opcode (decocare.history.questionable2c attribute), 32
 opcode (decocare.history.questionable3b attribute), 32
 opcode (decocare.history.questionable5e attribute), 32
 opcode (decocare.history.questionable61 attribute), 32
 opcode (decocare.history.questionable65 attribute), 32
 opcode (decocare.history.questionable66 attribute), 32
 opcode (decocare.history.questionable6f attribute), 32
 opcode (decocare.history.questionable7c attribute), 32
 opcode (decocare.history.Rewind attribute), 29
 opcode (decocare.history.Sara6E attribute), 29
 opcode (decocare.history.SelectBasalProfile attribute), 29
 opcode (decocare.history.SensorAlert attribute), 30
 opcode (decocare.history.SetAutoOff attribute), 30
 opcode (decocare.history.SetBolusWizardEnabled attribute), 30
 opcode (decocare.history.SettingSomething57 attribute), 30
 opcode (decocare.history.TempBasal attribute), 30
 opcode (decocare.history.TempBasalDuration attribute), 30
 open() (decocare.link.Link method), 3
 open() (decocare.stick.Stick method), 6

P

page (decocare.commands.ReadHistoryData attribute), 12

- PagedData (class in decocare.history), 29
- PageDownloader (class in decocare.download), 25
- pages (decocare.commands.ReadCurPageNumber attribute), 10
- params (decocare.commands.BaseCommand attribute), 8
- params (decocare.commands.Bolus attribute), 16
- params (decocare.commands.KeypadPush attribute), 9
- params (decocare.commands.PowerControl attribute), 9
- params (decocare.commands.PowerControlOff attribute), 9
- params (decocare.commands.PumpCommand attribute), 9
- params (decocare.commands.PumpDisableDetailTrace attribute), 23
- params (decocare.commands.PumpEnableDetailTrace attribute), 23
- params (decocare.commands.PumpResume attribute), 10
- params (decocare.commands.PumpSuspend attribute), 10
- params (decocare.commands.Read128KMem attribute), 17
- params (decocare.commands.Read256KMem attribute), 17
- params (decocare.commands.ReadAlarms attribute), 16
- params (decocare.commands.ReadBasalTemp attribute), 10
- params (decocare.commands.ReadBasalTemp508 attribute), 17
- params (decocare.commands.ReadBatteryStatus attribute), 10
- params (decocare.commands.ReadBolusHistory attribute), 16
- params (decocare.commands.ReadContrast attribute), 10
- params (decocare.commands.ReadCurGlucosePageNumber attribute), 15
- params (decocare.commands.ReadCurPageNumber attribute), 10
- params (decocare.commands.ReadDailyTotals attribute), 16
- params (decocare.commands.ReadErrorStatus attribute), 11
- params (decocare.commands.ReadErrorStatus508 attribute), 16
- params (decocare.commands.ReadFirmwareVersion attribute), 11
- params (decocare.commands.ReadGlucoseHistory attribute), 11
- params (decocare.commands.ReadHistoryData attribute), 12
- params (decocare.commands.ReadISIGHistory attribute), 18
- params (decocare.commands.ReadPrimeBoluses attribute), 16
- params (decocare.commands.ReadProfileSets attribute), 17
- params (decocare.commands.ReadPumpID attribute), 12
- params (decocare.commands.ReadPumpModel attribute), 12
- params (decocare.commands.ReadPumpState attribute), 13
- params (decocare.commands.ReadPumpStatus attribute), 13
- params (decocare.commands.ReadRadioCtrlACL attribute), 13
- params (decocare.commands.ReadRemainingInsulin attribute), 14
- params (decocare.commands.ReadRemoteControlID attribute), 17
- params (decocare.commands.ReadRTC attribute), 13
- params (decocare.commands.ReadSensorSettings attribute), 18
- params (decocare.commands.ReadSettings attribute), 14
- params (decocare.commands.ReadTodayTotals508 attribute), 17
- params (decocare.commands.ReadTotalsToday attribute), 14
- params (decocare.commands.ReadUserEvents attribute), 17
- params (decocare.commands.SelectBasalProfileA attribute), 24
- params (decocare.commands.SelectBasalProfileB attribute), 24
- params (decocare.commands.SelectBasalProfileSTD attribute), 24
- params (decocare.commands.SetSuspend attribute), 14
- params (decocare.commands.TempBasal attribute), 15
- params (decocare.commands.TempBasalPercent attribute), 15
- parse (class in decocare.lib), 34
- parse() (decocare.stick.InterfaceStats method), 4
- parse() (decocare.stick.LinkStatus method), 4
- parse() (decocare.stick.ProductInfo method), 4
- parse() (decocare.stick.ReadRadio method), 5
- parse() (decocare.stick.SignalStrength method), 5
- parse() (decocare.stick.StickCommand method), 6
- parse() (decocare.stick.TransmitPacket method), 7
- parse_midnight() (in module decocare.history), 31
- parse_record() (in module decocare.history), 31
- parse_time() (decocare.history.Model522ResultTotals method), 29
- parse_time() (decocare.history.MResultTotals method), 28
- poll_size() (decocare.stick.Stick method), 6
- port (decocare.link.Link attribute), 3
- power_control() (decocare.session.Pump method), 7
- PowerControl (class in decocare.commands), 9
- PowerControlOff (class in decocare.commands), 9
- process() (decocare.stick.Stick method), 6
- product_info() (decocare.stick.Stick method), 6

ProductInfo (class in decocare.stick), 4
 Program() (decocare.commands.TempBasal class method), 15
 Pump (class in decocare.session), 7
 PumpCommand (class in decocare.commands), 9
 PumpDisableDetailTrace (class in decocare.commands), 23
 PumpEnableDetailTrace (class in decocare.commands), 22
 PumpExperiment_OP69 (class in decocare.commands), 24
 PumpExperiment_OP70 (class in decocare.commands), 24
 PumpExperiment_OP71 (class in decocare.commands), 24
 PumpExperiment_OP72 (class in decocare.commands), 24
 PumpExperiment_OP73 (class in decocare.commands), 24
 PumpExperiment_OP75 (class in decocare.commands), 24
 PumpResume (class in decocare.commands), 9
 PumpResume (class in decocare.history), 29
 PumpSuspend (class in decocare.commands), 10
 PumpSuspend (class in decocare.history), 29
 PumpTraceSelect (class in decocare.commands), 22
 PushACT() (in module decocare.commands), 14
 PushDOWN() (in module decocare.commands), 14
 PushEASY() (in module decocare.commands), 14
 PushESC() (in module decocare.commands), 14
 PushUP() (in module decocare.commands), 14

Q

query() (decocare.session.Pump method), 7
 query() (decocare.stick.Stick method), 6
 questionable22 (class in decocare.history), 32
 questionable23 (class in decocare.history), 32
 questionable24 (class in decocare.history), 32
 questionable2c (class in decocare.history), 32
 questionable3b (class in decocare.history), 32
 questionable5e (class in decocare.history), 32
 questionable61 (class in decocare.history), 32
 questionable65 (class in decocare.history), 32
 questionable66 (class in decocare.history), 32
 questionable6f (class in decocare.history), 32
 questionable7c (class in decocare.history), 32

R

radio_stats() (decocare.stick.Stick method), 6
 RadioStats (class in decocare.stick), 4
 read() (decocare.link.Link method), 3
 Read128KMem (class in decocare.commands), 17
 Read256KMem (class in decocare.commands), 17

read_current() (decocare.download.PageDownloader method), 25
 read_history_0() (decocare.session.Pump method), 7
 read_model() (decocare.session.Pump method), 7
 read_status() (decocare.stick.Stick method), 6
 ReadAlarms (class in decocare.commands), 16
 ReadBasalTemp (class in decocare.commands), 10
 ReadBasalTemp508 (class in decocare.commands), 17
 ReadBatteryStatus (class in decocare.commands), 10
 ReadBGAlarmCLocks (class in decocare.commands), 20
 ReadBGAlarmEnable (class in decocare.commands), 21
 ReadBGReminderEnable (class in decocare.commands), 21
 ReadBGTargets (class in decocare.commands), 20
 ReadBGTargets515 (class in decocare.commands), 20
 ReadBGUnits (class in decocare.commands), 20
 ReadBolusHistory (class in decocare.commands), 16
 ReadBolusReminderEnable (class in decocare.commands), 22
 ReadBolusReminders (class in decocare.commands), 22
 ReadBolusWizardSetupStatus (class in decocare.commands), 20
 ReadCalibrationFactor (class in decocare.commands), 22
 ReadCarbRatios (class in decocare.commands), 20
 ReadCarbRatios512 (class in decocare.commands), 20
 ReadCarbUnits (class in decocare.commands), 20
 ReadContrast (class in decocare.commands), 10
 ReadCurGlucosePageNumber (class in decocare.commands), 15
 ReadCurPageNumber (class in decocare.commands), 10
 ReadDailyTotals (class in decocare.commands), 16
 ReadDetailTrace (class in decocare.commands), 19
 ReadErrorStatus (class in decocare.commands), 11
 ReadErrorStatus508 (class in decocare.commands), 16
 ReadFactoryParameters (class in decocare.commands), 22
 ReadFirmwareVersion (class in decocare.commands), 11
 ReadGlucoseHistory (class in decocare.commands), 11
 ReadHistoryData (class in decocare.commands), 11
 ReadInsulinSensitivities (class in decocare.commands), 20
 ReadISIGHistory (class in decocare.commands), 18
 ReadLanguage (class in decocare.commands), 19
 readline() (decocare.link.Link method), 3
 readlines() (decocare.link.Link method), 3
 ReadLogicLinkIDS (class in decocare.commands), 21
 ReadNewTraceAlarm (class in decocare.commands), 19
 ReadOldTraceAlarm (class in decocare.commands), 19
 ReadOtherDevicesIDS (class in decocare.commands), 22
 ReadPrimeBoluses (class in decocare.commands), 16
 ReadProfile_A512 (class in decocare.commands), 21
 ReadProfile_B512 (class in decocare.commands), 21
 ReadProfile_STD512 (class in decocare.commands), 21
 ReadProfiles511_A (class in decocare.commands), 19

- ReadProfiles511_B (class in decocare.commands), 19
 - ReadProfiles511_STD (class in decocare.commands), 18
 - ReadProfileSets (class in decocare.commands), 16
 - ReadPumpID (class in decocare.commands), 12
 - ReadPumpModel (class in decocare.commands), 12
 - ReadPumpState (class in decocare.commands), 13
 - ReadPumpStatus (class in decocare.commands), 13
 - ReadPumpTrace (class in decocare.commands), 19
 - ReadRadio (class in decocare.stick), 4
 - ReadRadioCtrlACL (class in decocare.commands), 13
 - ReadRemainingInsulin (class in decocare.commands), 13
 - ReadRemoteControlID (class in decocare.commands), 17
 - ReadReservoirWarning (class in decocare.commands), 21
 - ReadRTC (class in decocare.commands), 13
 - ReadSavedSettingsDate (class in decocare.commands), 22
 - ReadSensorHistoryData (class in decocare.commands), 18
 - ReadSensorSettings (class in decocare.commands), 17
 - ReadSettings (class in decocare.commands), 14
 - ReadSettings511 (class in decocare.commands), 19
 - ReadSettings512 (class in decocare.commands), 21
 - ReadTodayTotals508 (class in decocare.commands), 17
 - ReadTotalsToday (class in decocare.commands), 14
 - ReadUserEvents (class in decocare.commands), 17
 - ReadVCNTRHistory (class in decocare.commands), 22
 - reasons (decocare.stick.LinkStatus attribute), 4
 - record_error() (decocare.stick.LinkStatus method), 4
 - resp_1 (decocare.commands.ReadInsulinSensitivities attribute), 20
 - respond() (decocare.commands.BaseCommand method), 8
 - respond() (decocare.commands.ReadCurPageNumber method), 10
 - respond() (decocare.commands.ReadHistoryData method), 12
 - respond() (decocare.stick.ReadRadio method), 5
 - respond() (decocare.stick.StickCommand method), 6
 - respond() (decocare.stick.TransmitPacket method), 7
 - responded (decocare.commands.BaseCommand attribute), 8
 - retries (decocare.commands.BaseCommand attribute), 8
 - retries (decocare.commands.KeypadPush attribute), 9
 - retries (decocare.commands.PowerControl attribute), 9
 - retries (decocare.commands.PumpCommand attribute), 9
 - retries (decocare.commands.ReadBasalTemp attribute), 10
 - retries (decocare.commands.ReadBasalTemp508 attribute), 17
 - retries (decocare.commands.ReadBatteryStatus attribute), 10
 - retries (decocare.commands.ReadContrast attribute), 10
 - retries (decocare.commands.ReadCurGlucosePageNumber attribute), 15
 - retries (decocare.commands.ReadCurPageNumber attribute), 11
 - retries (decocare.commands.ReadErrorStatus attribute), 11
 - retries (decocare.commands.ReadFirmwareVersion attribute), 11
 - retries (decocare.commands.ReadHistoryData attribute), 12
 - retries (decocare.commands.ReadPumpID attribute), 12
 - retries (decocare.commands.ReadPumpModel attribute), 13
 - retries (decocare.commands.ReadPumpState attribute), 13
 - retries (decocare.commands.ReadPumpStatus attribute), 13
 - retries (decocare.commands.ReadRadioCtrlACL attribute), 13
 - retries (decocare.commands.ReadRemainingInsulin attribute), 14
 - retries (decocare.commands.ReadRTC attribute), 13
 - retries (decocare.commands.ReadSensorSettings attribute), 18
 - retries (decocare.commands.ReadSettings attribute), 14
 - retries (decocare.commands.ReadTodayTotals508 attribute), 17
 - retries (decocare.commands.ReadTotalsToday attribute), 14
 - retries (decocare.commands.SetSuspend attribute), 14
 - retries (decocare.commands.TempBasal attribute), 15
 - retries (decocare.commands.TempBasalPercent attribute), 16
 - Rewind (class in decocare.history), 29
 - rf_table (decocare.stick.ProductInfo attribute), 4
- ## S
- Sara6E (class in decocare.history), 29
 - save() (decocare.commands.PumpCommand method), 9
 - SelectBasalProfile (class in decocare.commands), 24
 - SelectBasalProfile (class in decocare.history), 29
 - SelectBasalProfileA (class in decocare.commands), 24
 - SelectBasalProfileB (class in decocare.commands), 24
 - SelectBasalProfileSTD (class in decocare.commands), 24
 - send_force_read() (decocare.stick.Stick method), 6
 - SensorAlert (class in decocare.history), 29
 - serial (decocare.commands.PumpCommand attribute), 9
 - Session (class in decocare.session), 8
 - set_reason() (decocare.stick.LinkStatus method), 4
 - SetAutoOff (class in decocare.commands), 15
 - SetAutoOff (class in decocare.history), 30
 - SetBasalType (class in decocare.commands), 15
 - SetBolusWizardEnabled (class in decocare.history), 30
 - SetEnabledEasyBolus (class in decocare.commands), 15
 - setModel() (decocare.session.Pump method), 7
 - SetSuspend (class in decocare.commands), 14

SettingSomething57 (class in decocare.history), 30
signal_strength() (decocare.stick.Stick method), 6
SignalStrength (class in decocare.stick), 5
size (decocare.stick.ReadRadio attribute), 5
size (decocare.stick.StickCommand attribute), 7
Stick (class in decocare.stick), 5
StickCommand (class in decocare.stick), 6
StickError, 35
suggest() (in module decocare.history), 32
SW_VER (decocare.stick.ProductInfo attribute), 4

T

temp_basal_type() (decocare.commands.ReadSettings method), 14
TempBasal (class in decocare.commands), 15
TempBasal (class in decocare.history), 30
TempBasalDuration (class in decocare.history), 30
TempBasalPercent (class in decocare.commands), 15
test_cal_bg() (decocare.records.tests.TestSaraBolus class method), 36
TestSaraBolus (class in decocare.records.tests), 35
timeout (decocare.commands.BaseCommand attribute), 8
Timer (class in decocare.lib), 33
transfer() (decocare.session.Session method), 8
transmit_packet() (decocare.stick.Stick method), 6
TransmitPacket (class in decocare.stick), 7

U

unmask_m_midnight() (in module decocare.history), 32
unsolved_bolus_wizard() (in module decocare.records.tests), 36
UP() (decocare.commands.KeypadPush class method), 8
usb_stats() (decocare.stick.Stick method), 6
UsbStats (class in decocare.stick), 7

W

write() (decocare.link.Link method), 3
WriteGlucoseHistoryTimestamp (class in decocare.commands), 19