

GitHub

This repository Search

[Explore](#) [Features](#) [Enterprise](#) [Blog](#)[Sign up](#)[Sign in](#) **dmand / rpi-ds1302**

★ Star 0

🍴 Fork 0

Python module for using DS1302 RTC unit with Raspberry Pi

14 commits

1 branch

0 releases

1 contributor

&lt;&gt; Code

🔔 Issues 0

🔗 Pull Requests 0

🔊 Pulse

📊 Graphs

HTTPS clone URL









<https://github.com/dmand/>You can clone with [HTTPS](#) or [Subversion](#). Clone in Desktop Download ZIP branch: master **rpi-ds1302 / +**

Fix englando in README



dmand authored on 28 Jan 2014

latest commit f386476814

|   |  |            |
|---|--|------------|
|  <a href="#">.gitignore</a>  | Add gitignore for builds                     | a year ago |
|  <a href="#">LICENSE</a>     | Initial commit                               | a year ago |
|  <a href="#">README.md</a>   | Fix englando in README                       | a year ago |
|  <a href="#">check.sh</a>    | Source files drop-off                        | a year ago |
|  <a href="#">hwclock.py</a>  | Add hwclock script for getting hardware time | a year ago |
|  <a href="#">python.c</a>    | Modify sanity checks, make them mandatory    | a year ago |
|  <a href="#">rpi_time.py</a> | Modify sanity checks, make them mandatory    | a year ago |
|  <a href="#">setup.py</a>    | Oops, this doesn't meant to be here          | a year ago |

 **README.md**

# rpi-ds1302

Python library for using DS1302 RTC unit with Raspberry Pi.

Depends on `wiringPi` library.

TODO:

- split C / CPython / Python code
- way to set up RTC pins from Python code

## Usage

Currently you have to connect RTC unit to the following RPi pins (you can read about RPi pin numbering [here](#), this library uses wiringPi pin numbering):

- CLK pin (also can be named SCLK) goes to RPi pin 4.
- DAT pin (sometimes it is named I/O) goes to RPi pin 5.
- RST pin goes to RPi pin 6.

There are two ways of using RTC unit from Python code.

*Low-level* access procedures are implemented in `ds1302` module.

```
>>> import ds1302
>>> ds1302.get_date()
(2014, 1, 20)
>>> ds1302.get_time()
(15, 56, 1)
>>> ds1302.set_time(14, 14, 14)
>>> ds1302.get_time()
(14, 14, 18)
```

*High-level* access procedures are implemented in `rpi_time` module (this one includes sanity checks for RTC unit and range checks for clock values).

```
>>> import rpi_time
>>> rtc = rpi_time.DS1302()
>>> rtc.get_datetime()
datetime.datetime(2014, 1, 20, 14, 16, 27)
>>> rtc.set_datetime(datetime(2014, 1, 20, 13, 42))
True
>>> rtc.get_datetime()
datetime.datetime(2014, 1, 20, 13, 42, 2)
>>> rtc.reset_clock()
>>> rtc.get_datetime()
datetime.datetime(2001, 1, 1, 0, 0, 5)
```

