
homemate-bridge

Release 0.1.1

Apr 13, 2020

Contents

1	Overview	3
1.1	Supported Devices	3
1.2	Requirements	3
1.3	Installation	4
1.4	Getting the Orvibo ‘PK’ encryption key	4
1.5	Documentation	4
1.6	Development	4
2	Installation	5
3	Usage	7
4	Reference	9
4.1	homemate_bridge	9
5	Contributing	11
5.1	Bug reports	11
5.2	Documentation improvements	11
5.3	Feature requests and feedback	11
5.4	Development	12
6	Authors	13
7	Changelog	15
7.1	0.0.1 (2017-07-11)	15
8	Indices and tables	17
	Python Module Index	19
	Index	21

THIS PROJECT IS NO LONGER BEING ACTIVELY MAINTAINED

CHAPTER 1

Overview

docs	
tests	
package	

Orvibo “Homemate” to MQTT bridge

- Free software: Apache Software License 2.0

1.1 Supported Devices

- Orvibo S20c

If you’ve managed to the bridge working with other devices, please open an issue so I can update the list!

1.2 Requirements

- Python 3.x, on Linux
- The python3-dev, build-essential and libssl-dev packages (for Debian-based systems, or the equivalent)
- One or more supported switches
- HomeAssistant
- An MQTT broker connected to HomeAssistant

1.3 Installation

- Obtain the Orvibo ‘PK’ key (see below) as a *keys.json* file
- Redirect all traffic for `homemate.orvibo.com`, TCP port 10001 to the machine running the bridge. The easiest way to do this is to override the DNS record, but how you can do this will greatly depend on how your network is set up
- *Configure HomeAssistant* <<https://home-assistant.io/docs/mqtt/discovery/>>_ to discover MQTT devices
- Install the bridge:

```
pip install homemate-bridge
```

- Run the bridge:

```
homemate-bridge --keys-file <path/to/key/file> --mqtt-host ...
```

- After ~30 seconds you should see devices connecting to the bridge, and new switch entities in HomeAssistant

1.4 Getting the Orvibo ‘PK’ encryption key

As part of the initial handshake with the server, the switch sends a ‘hello’ packet encrypted with a static key, and expects a packet encrypted with the same key in response that sets a different key for all subsequent packets. This is the ‘PK’ key, and is not included with the source code until I work out if there would be legal issues with doing so. Fortunately, Orvibo hardcode this key in the source code of the ‘Kepler’ Android app.

- Download the *Kepler apk* <<http://www.orvibo.com/software/android/kepler.apk>>_
- Run the *homemate-bridge-seed-keyfile* `-keys-file keys.json <path/to/apk>` script to extract the key and save it. Note that the file will be overwritten if it exists.

1.5 Documentation

<https://homemate-bridge.readthedocs.io/>

1.6 Development

To run the all tests run:

```
tox
```


CHAPTER 2

Installation

At the command line:

```
pip install homemate-bridge
```


CHAPTER 3

Usage

To use homemate-bridge in a project:

```
import homemate_bridge
```


CHAPTER 4

Reference

4.1 homemate_bridge

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

5.1 Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.2 Documentation improvements

homemate-bridge could always use more documentation, whether as part of the official homemate-bridge docs, in docstrings, or even on the web in blog posts, articles, and such.

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/insertjokehere/homemate-bridge/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

5.4 Development

To set up *homemate-bridge* for local development:

1. Fork [homemate-bridge](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/homemate-bridge.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .  
git commit -m "Your detailed description of your changes."  
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

5.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will [run the tests](#) for each change you add in the pull request.

It will be slower though ...

CHAPTER 6

Authors

- Will Hughes - <https://www.willhughes.name>

CHAPTER 7

Changelog

7.1 0.0.1 (2017-07-11)

- First release on PyPI.

CHAPTER 8

Indices and tables

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

h

homemate_bridge, 9

H

homemate_bridge (*module*), [9](#)