
LMT8x

Release 1.0.0

Charlee Li

Apr 03, 2021

CONTENTS

1	Overview	1
1.1	Installation	1
1.2	Usage	2
1.3	Documentation	2
1.4	Development	2
2	Installation	3
3	Usage	5
4	Reference	7
4.1	lmt8x	7
5	Contributing	9
5.1	Bug reports	9
5.2	Documentation improvements	9
5.3	Feature requests and feedback	9
5.4	Development	10
6	Authors	11
7	Changelog	13
7.1	1.0.0 (2021-04-02)	13
8	Indices and tables	15
	Python Module Index	17
	Index	19

OVERVIEW

docs	
tests	
package	

Transfer tables for LMT8x Temperature Sensors.

LMT8x (LMT84, LMT85, LMT86, LMT87) are a series of [analog temperature sensors](#) made by Texas Instrument. These sensors can provide -50°C ~ 150°C with $\pm 0.4^{\circ}\text{C}$ accuracy. Although the output voltage is nearly linear to the temperature, it does have a slight umbrella parabolic shape. Therefore transfer tables are required to convert the voltage to the temperature.

Datasheets:

- [LMT84](#)
- [LMT85](#)
- [LMT86](#)
- [LMT87](#)

This package provides tranfer functions for LMT84, LMT85, LMT86, and LMT87 based on their transfer tables. These funcitons basically do a binary search through the transfer tables and return the match. If no match found, linear interpolation will be used to generate an approximate value.

1.1 Installation

```
pip install lmt8x
```

You can also install the in-development version with:

```
pip install https://github.com/charlee/python-lmt8x/archive/master.zip
```

1.2 Usage

lmt8x package provides functions *lmt84_v2t*, *lmt85_v2t*, *lmt86_v2t*, and *lmt87_v2t* for converting voltage to temperature.

- input voltage must be in mV.
- output temperature is in celsius.

```
from lmt8x import lmt87_v2t      # or `lmt86_v2t`, `lmt85_v2t`, `lmt84_v2t`

# read voltage from sensors.
# v = read_sensor()

# convert v to temperature.
# the parameter must be in mV. Return value is in celsius.
temp = lmt87_v2t(v * 1000)

print('Temperature is %s C.' % temp)
```

1.3 Documentation

<https://python-lmt8x.readthedocs.io/>

1.4 Development

To run all the tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Win- dows	set PYTEST_ADDOPTS=--cov-append tox
Other	PYTEST_ADDOPTS=--cov-append tox

INSTALLATION

At the command line:

```
pip install lmt8x
```


USAGE

lmt8x package provides functions *lmt84_v2t*, *lmt85_v2t*, *lmt86_v2t*, and *lmt87_v2t* for converting voltage to temperature.

- input voltage must be in mV.
- output temperature is in Celsius.

```
from lmt8x import lmt87_v2t      # or `lmt86_v2t`, `lmt85_v2t`, `lmt84_v2t`

# read voltage from sensors.
# v = read_sensor()

# convert v to temperature.
# the parameter must be in mV. Return value is in celsius.
temp = lmt87_v2t(v * 1000)

print('Temperature is %s C.' % temp)
```


REFERENCE

4.1 lmt8x

`lmt8x.lmt84_v2t(v)`

Transfer function for LMT84.

Parameters `v(float)` – Voltage in mV.

Returns Temperature in Celsius.

Return type float

`lmt8x.lmt85_v2t(v)`

Transfer function for LMT85.

Parameters `v(float)` – Voltage in mV.

Returns Temperature in Celsius.

Return type float

`lmt8x.lmt86_v2t(v)`

Transfer function for LMT86.

Parameters `v(float)` – Voltage in mV.

Returns Temperature in Celsius.

Return type float

`lmt8x.lmt87_v2t(v)`

Transfer function for LMT87.

Parameters `v(float)` – Voltage in mV.

Returns Temperature in Celsius.

Return type float

CONTRIBUTING

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

LMT8x could always use more documentation, whether as part of the official LMT8x 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/charlee/python-lmt8x/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 *python-lmt8x* for local development:

1. Fork [python-lmt8x](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:YOURGITHUBNAME/python-lmt8x.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 and docs builder 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 -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p auto
```

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
SIX

AUTHORS

- Charlee Li - <https://charlee.li>

CHANGELOG

7.1 1.0.0 (2021-04-02)

- First release on PyPI.

INDICES AND TABLES

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

PYTHON MODULE INDEX

I

lmt8x, 7

INDEX

L

`lmt84_v2t()` (*in module `lmt8x`*), 7

`lmt85_v2t()` (*in module `lmt8x`*), 7

`lmt86_v2t()` (*in module `lmt8x`*), 7

`lmt87_v2t()` (*in module `lmt8x`*), 7

`lmt8x`

 module, 7

M

module

`lmt8x`, 7