

THERMOplate Command Reference

This list can also be accessed by typing `THERMO.getHelp()` from the Python command line:

Definitions:

- addr:** Address - THERMOplates have jumpers on the board that allow their address to be set to a value between 0 and 7.
- channel:** Each THERMOplate has twelve individually addressable inputs numbered 1 through 12. Channels 1 through 8 are Type K thermocouple inputs while channels 9 through 12 accept the DS18B20 digital temperature sensor.
- scale:** the THERMOplate can return measurements in three different temperature scales: Celcius, Farenheit, and Kelvin. At power up, the board will default to Celcius. If a function requires a scale setting then the argument is a simple character enclosed in single quotation marks: 'c', 'f', or 'k'. Note that the scale argument is optional for the `getTEMP()` and `getCOLD()` functions and will default to the global setting if not included as an argument.
- value:** many functions will return a value. For example, `getTEMP()` returns a the temperature value of the selected channel as a decimal number. Other functions may return a string of characters or a single integer Value.
- freq:** to minimize power line noise, the THERMOplate averages multiple readings during power line cycles. In North America, the line frequency is 60Hz (the default value) while in Europe and other parts of the world the frequency is 50Hz.
- Type:** the THERMOplate can measure temperature using both K type and J type thermocouples. Valid arguments are 'k' and 'j'. Be sure and include the quotation marks.

Temperature Read Functions

`getTEMP(addr,channel,scale*)` - returns the measured temperature of the specified channel. If the optional scale argument is included, it will override the global scale specified by the `setSCALE` function.

`setSCALE(scale)` - sets the temperature scale for all of the THERMOplates attached to the Raspberry Pi. This is a global value that applies to all temperature measurements unless a different scale setting is passed to the `getTEMP` or `getCOLD` functions.

`getSCALE(scale)` - returns a single character which indicates the current temperature scale being reported by all of the THERMOplates.

`setType(addr,channel,type)` - sets the thermocouple type for a specific channel. Since This only applies to thermocouple inputs, an error message will be generated if the channel argument is not between 1 and 8.

getCOLD(addr,scale*) - returns the temperature of the cold junction on the addressed THERMOplate. If the optional scale argument is included, it will override the global scale specified by the setSCALE function. See the documentation for the purpose of this temperature.

setLINEFREQ(addr,freq) - instructs the THERMOplate to collect samples over 1/60 of a second or over 1/50 of a second. The default value is 60 for North America. It should be set to 50 if you are in the UK or Europe. Note that this function only needs to be called once at the beginning of your program.

setSMOOTH(addr) - instructs the THERMOplate to apply a smoothing filter to each thermocouple channel. This feature is on by default.

clrSMOOTH(addr) - instructs the THERMOplate to pass the raw data from each thermocouple channel.

Interrupt Functions

setINTchannel(addr, channel) - directs the THERMOplate to pull the INT line (pin 15 on the Raspberry Pi) low when a new temperature value is available for the selected channel

intEnable(addr) - enables the INT feature on the addressed THERMOplate.

intDisable(addr) - disables the INT feature on the addressed THERMOplate.

getINTflags(addr) - returns an 8-bit value which indicates if a thermocouple or a digital sensor generated the INT. The bits in this register are:

NA|NA|NA|NA|NA|NA|DS18B20|Thermocouple|

After this register is read from the THERMOplate, all the bits are cleared.

LED Control Functions

setLED(addr) - turn on the LED

clrLED(addr) - turn off the LED

toggleLED(addr) - if LED is on, turn off. If LED is off, turn on.

getLED(addr) - returns current state of LED - 1 is ON and 0 is OFF.

System Level Functions:

getID(addr) - return Pi-Plate descriptor string

getFWrev(addr) - return FW revision

getHWrev(addr) - return HW revision

getVersion() - returns revision of python module

getADDR(addr) - return address of pi-plate. Used for polling available boards at power up.

RESET(addr) - set THERMOplate to power on state. Returns temperature scale to Celcius.