This library is built specifically for Arduinos connecting to the MCP7940M RTC chip. (Code is currently untested).

begin() begins I2C communications.

timeSet(**second**, **minute**, **hour**, **wkday**, **date**, **month**, **year**) sets the RTC to the time described in the inputs. **wkday** is a number 1-7 arbitrarily defined by the user (i.e. Monday does not have to be 1). **year** must be input as just the last two digits (e.g. 2018 is input as 18).

stopClock() shuts off the RTC oscillator, regardless of whether it is an external clock input or crystal. It returns a 0 if an external clock was in use, a 1 if an external crystal was in use, and a -1 if neither were yet set.

clockSelect(**external\_clock**) is used to select the RTC clock source. **external\_clock** == true selects an external clock source. **external\_clock** == false (default) selects an external crystal.

alarmEnable(**alarm0**, **set\_alarm**, **alarm\_low**, **alarm\_mask\_setting)** enables or disables RTC alarms (configuration must be set to general purpose/alarm mode first). **alarm0** == true (default) selects alarm 0. **alarm0** == false selects alarm 1. **set\_alarm** == true (default) enables the selected alarm. **set\_alarm** == false disables the selected alarm. **alarm\_low** == true (default) sets output pin low in alarm state. **alarm\_low** == false sets output pin high in alarm state. **alarm\_mask\_setting** configuration is defined in the table below:

|  |  |
| --- | --- |
| **alarm\_mask\_setting** | Alarm mask |
| 0 | Seconds match |
| 1 (default) | Minutes match |
| 2 | Hours match |
| 3 | Day of week match |
| 4 | Date match |
| 5 | All above and month match |

alarmAdjust(**alarm0**, **second**, **minute**, **hour**, **wkday**, **date**, **month**) sets the alarm time. **alarm0** == true (default) defines the settings for alarm 0. If false, it defines the settings for alarm 1. **second** is an integer from 0 to 59. **minute** is an integer from 0 to 59. **hour** is an integer from 0 to 59. **wkday** is an integer from 1 to 7 (note: this is an arbitrarily defined field, see timeSet description). **date** is an integer from 0 to 31, depending on the month. Setting nonreal dates (e.g. February 30th) results in undefined behavior. Leap years \*should\* be ok. **month** is an integer from 1 to 12.

configure(**Mode**, **out\_pin**, **trim**, **sqwave\_set**) configures the RTC. **Mode** is an integer. **Mode** == 1 (default) sets the RTC to general purpose/alarm mode\*. **Mode** != 1 sets the RTC to square wave clock output mode. **out\_pin** == true (default) sets the multipurpose pin to high (only used in general purpose mode). If false, multipurpose pin is set to low. **trim** == false (default) turns off coarse trimming of square wave output (see trimConfig for more details). If true, coarse trimming is enabled. **sqwave\_set** configuration is defined in the table below:

|  |  |
| --- | --- |
| **sqwave\_set** | Squarewave setting |
| 0 (default) | 1 Hz |
| 1 | 4.096 Hz |
| 2 | 8.192 Hz |
| 3 | 32.768 Hz |

trimConfig(**cycles**, **add**) adjusts the RTC digital trimming of the oscillator. Note **trim** in the configure method is NOT a prerequisite. **cycles** is an even integer from 0 (default) to 254. This is the number of cycles to be added or subtracted. **add** == true (default) adds the cycles. If false, subtracts the clock cycles. Functionality is dependent on the **trim** parameter in the configure method. If **trim** == false, the adding or subtracting of clock cycles occurs every minute. If **trim** == true, the adding or subtracting of clock cycles occurs 128 times per second. Because of the drastic affect the digital trimming can have when **trim** == true, it is not recommended for normal operation.

Recommended alarm setup operation

1. begin
2. configure
3. trimConfig
4. clockSelect
5. timeSet
6. alarmAdjust
7. alarmEnable