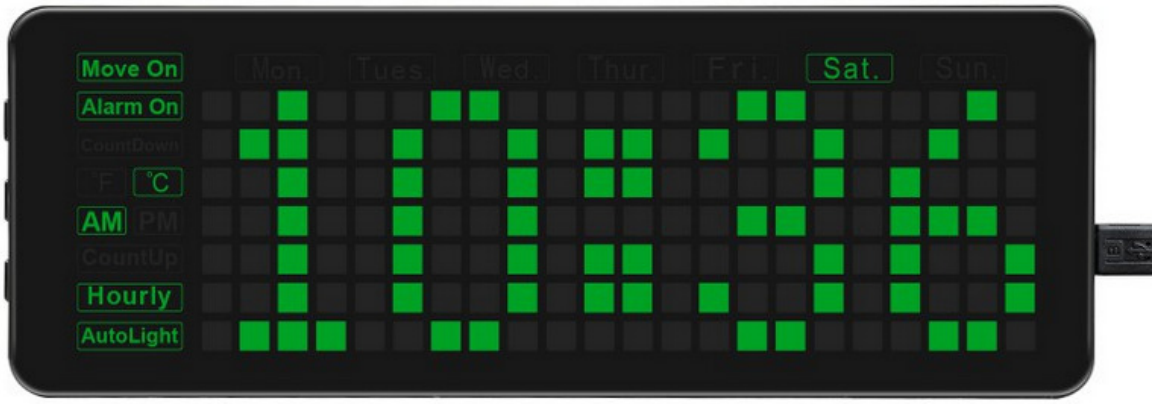
# Pico Green Clock




**Firmware Version 5.00**
**User Guide**
**Updated July 26th, 2022**

**IMPORTANT :**
This User guide is about firmware Version 5.00 from Andre St. Louys. Versions 2.00 and up are based on the original Waveshare's Version 1.00 and add more features to the clock. If you're using the original firmware from Waveshare, many features described in this manual do not apply.

# Pico Green Clock User Guide

## Index

# Pico Green Clock User Guide

Do you use this version of the firmware?
Do you like it?
What are the features you appreciate the most?
Did you find some bugs?
Did you improve the firmware?
Did you add some more features?

Let me know, I'm curious!

Andre St. Louys
(Quebec, Canada)
astlouys@gmail.com

## About Version 5.00

Version 5.00 is mostly a consolidation version, fine tuning many things, correcting a few bugs (and of course, adding a few more features!). You can refer to the revision history at the end of this guide to get details about changes in this version.

**Special note about DHT22 device**

Please read the special warning about DHT22 in Appendix C

# Introduction

## *Waveshare Pico Green Clock*

As mentioned on the cover page, this User guide is about Firmware Version 5.00 from Andre St. Louys. Firmware Versions 2.00 and up are based on original Waveshare's firmware Version 1.00 and add more features to the clock and tons of comments in the source code which is in C-Language.

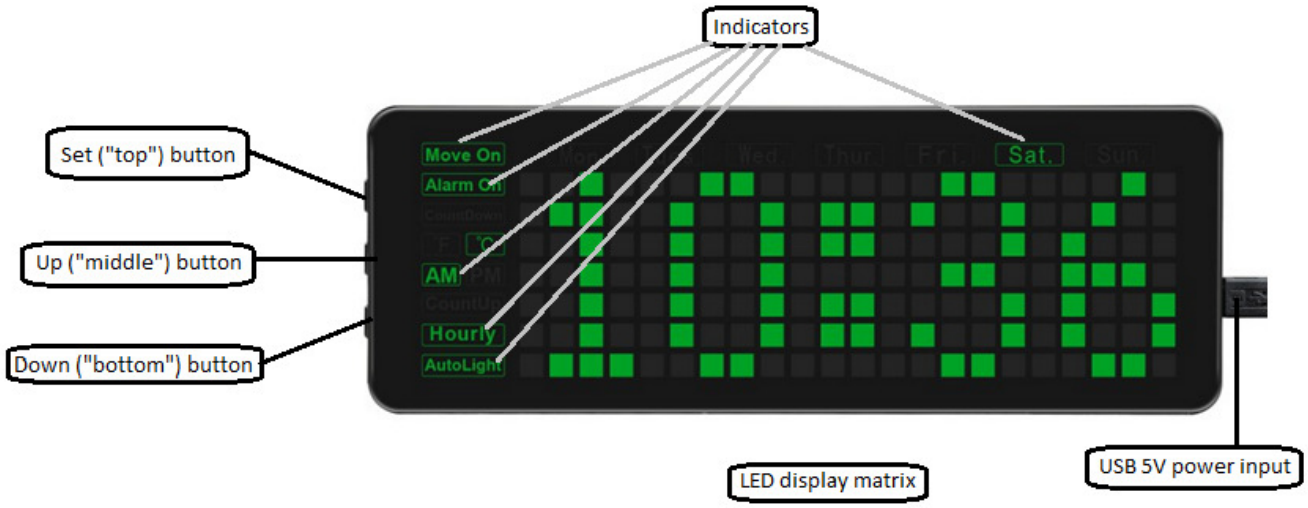
The Waveshare's Green Clock uses a Raspberry Pi Pico microcontroller to control most of its functions / features (IMPORTANT: the microcontroller itself, Raspberry Pi Pico, is not included with the clock. However, it is not very expensive – more or less 15 US$). The clock display is made of a matrix of 22 X 7 green LEDs, along with many indicators (for Day-of-week, AM / PM, Alarm On / Off, etc.). See figure below for more details.

There is also a real-time clock integrated circuit ("RTC IC") with a battery back-up, allowing the clock to keep the correct time / date and alarm settings in case of a power failure.

Some of the options / features / parameters of the clock are configurable at "run-time" (when the clock is powered on), while others are configurable only at "compile-time" (before compiling the executable firmware), as indicated in the text that follows. Options that are configurable at compile-time mean that a change must be done in the source code and the firmware needs to be rebuilt and re-flashed ("re-programmed") on the Pico microcontroller for the option to take effect. On the other side, an option that is configurable at run-time means that the user can configure it once the clock is powered on and running.

Keep in mind the name given to the three clock buttons on the left of the clock. They will be used very often throughout this guide to show how to configure the clock.

# Pico Green Clock User Guide




The button at the top is called the “Set” button since it is usually used to scan through the many settings of the clock.

The button in the middle is called the “Up” button (not to be mixed up with the “top” or “Set” button) because it is usually used to increase the value of current setting.

The button at the bottom is called “Down” button because it is usually used to decrease the value of current setting.

## *Clock power-up sequence*

The clock is powered by a USB (5 volts) charger / adaptor / power supply with a mini-USB connector.

When power is applied to the clock, many tests are performed. You may want to keep an eye on the different devices to make sure everything works fine.

Here is a summary of the clock power up sequence. [Square brackets] at the beginning of a line means that this step of the power-up sequence will execute only if the user has enabled the conditional compile for this device / option.

1) Turn On (for a few seconds) the two white LEDs (“night light”) near the clock buttons on the left side of the clock.

2) Sound a few “beeps” with the integrated active buzzer to test it.

3) Performs a clock display pixel test. Each column of pixels is turned On in sequence.

4) Perform indicators test. Most indicators have two back-LEDs. Each one is turned On in sequence.

5) Blink the two white LEDs (night light) a few times.

6) Pixels twinkling animation for a few seconds (just for fun).

7) [Passive buzzer]: If a passive buzzer has been installed by user, it will make random sounds during pixel twinkling.

8) If display scrolling has been configured "On" by default at compile time, the "Move On" indicator is turned On accordingly on the clock display.

9) The appropriate temperature unit indicator is turned On ("°C" or "°F"), depending on the default temperature unit that has been configured at compile time.

10) If hourly chime has been configured "On" or "Day" at compile time, the "Hourly" indicator will be turned On accordingly.

11) Display Green clock firmware version number.

12) [BME280]: If the conditional compile is set for an installed BME280 and its initialization returns an error code, an error message will scroll on the clock display. Other than this error message, algorithms have been tuned so that the clock will perform normally even if a BME280 is not installed.

13) [BME280]: If a BME280 has been installed by user, its "DeviceID" will be scrolled on clock display. The device ID for a "real BME280" is 0x60.

14) [BME280]: If a BME280 has been installed by user, its "Unique ID" will be scrolled on clock display (see the "Unique ID" as a "serial number"). The Unique ID is an hexadecimal number with a format of: XXXX XXXX.

15) [DHT22]: If the conditional compile is set for an installed DHT22 and an error occurs while trying to read the DHT22, an error message will scroll on the clock display. Other than this error message, algorithms have been tuned so that the clock will perform normally even if a DHT22 is not installed.

16) [Passive buzzer]: If a passive buzzer has been installed by user, it will play a few jingles to test it while continuing to scroll information on the clock display.

17) Display Pico microcontroller "Unique ID" in hexadecimal with a format of: XXXX XXXX XXXX XXXX (see the "Unique ID" as a "serial number" for the Pico installed in the Green Clock). As you can guess by its name, this number is

unique to the Pico that is installed in the clock. If you would replace it by another Pico, the number would be different.

18) Scroll current "Daylight saving time" mode / status, according to clock parameter setting and current time and date. Three different statuses are possible. "No support for daylight saving time" means that the clock completely ignores DST. It's like if DST does not exist for the clock. "Daylight saving time inactive" means that the clock takes care of the DST, but given the current date and time, daylight saving time is not active. Finally, "Daylight saving time is active" means that the clock takes care of DST, and given the current date and time, daylight saving time is active. Note: DST support is based on North American standard, which is the same for most northern hemisphere countries.

19) Display power supply voltage value (voltage supplied by USB power supply). It should be around 5.0 volts, but is often a little lower (4.85 or so).

20) If sound has been cut off at compile time, a warning will scroll on clock display. There is an option in the code to completely turn off all sounds generated by the clock. I used this feature during the development phase, when some coding sessions went on late during the night, while my wife was sleeping. If "sound cut-off" is scrolled on the clock display at power up, it means that this compile time option is active. No need to try finding a bug with alarm, chime, or anything else!

21) Display Pico microcontroller internal temperature (should be similar to ambient temperature, subject to the precision of the temperature sensors used).

22) Turn on the appropriate day-of-week indicator according to the real time clock setting.

23) Display time.

# Clock features

Here is a brief description of clock features. More details are given later in the guide where each feature is described in its specific setup section.

## *Alarms (run-time)*

There are two alarms available on the clock. They can be independently set at a specific time and day-of-week, and they will sound an alarm when the programmed alarm parameters are reached. Note: Alarm sound sequence may be different for alarm1 and alarm2.

## *Auto-brightness (run-time)*

The auto-brightness feature allows the clock LED matrix display to be brighter when the ambient light is bright and to dim the clock display when the ambient light is darker (see section on auto-brightness setting later in this guide).

The ambient light detection / sensor takes advantage of one of the analog-to-digital converter integrated in the Raspberry Pi Pico.

## *Beep types (compile-time)*

The piezo (buzzer) provided in the clock is an “active” piezo. That is, it has its own integrated oscillator, so it is not possible to change the frequency of the sounds that are produced for the different usages (calendar events, keyclick, hourly chime, alarms, timers, etc…). However, a feature has been implemented in the clock firmware since Version 2.00 allowing the making of a different number of “beeps” of different duration. This allows distinguishing between a calendar event, hourly chime, button keyclick, etc…

Beginning with version 3.00 this feature has been improved to another level by adding the concept of a “sound pack” (a primary group of sounds) and a repeat count for this “sound pack”. You will find more information about this later in this guide.

Firmware Version 4.00 and up also supports a “passive piezo” (that must be installed by user), allowing to change the sound frequency (the clock can then play jingles).

## *Calendar events (compile-time)*

The user can define short strings of text (40 characters maximum) that will scroll on the clock display at specific dates. For example, "John's Birthday" would scroll on the display on April 14th, if this date (14-APR) has been defined in the source code as a "Calendar event" with the text associated. Up to 50 such calendar events may be programmed and more than one may be programmed for a specific day. However, it must be understood that it takes some time to scroll the strings on the clock display. If there are too many events defined for the same day, there is a risk that the audience will miss part of them.

The calendar events will scroll on the screen during all day at the date defined in the clock configuration at compile time (the firmware must be rebuilt and re-flashed in the Pico). The text will scroll at xxh14 and xxh44 of each hour all day long (that is, every half hour), and a few beeps will also sound when the text begins scrolling. Those warning beeps are subjects to the same rules as the hourly chime. That is, if hourly chime is Off, the warning beeps will not sound (although the text will scroll, no matter the hourly chime setting). If hourly chime is On, warning beeps will sound during the whole day (that is, during the 24 hours defined by the date of the calendar event). Finally, if hourly chime is intermittent ("OI" for On, Intermittent in the settings), the warning beeps will sound during daytime, as defined between "chime time on" and "chime time off" (set by default from 9h00 to 21h00 – run-time parameters). You will find more information about this later in this guide.

A few of these calendar events have been programmed by default to show the user how to program others if desired. For example, "Happy New Year" and "Merry Christmas" are two such events that are programmed by default. Also, every first day of each month, an event called "Calendar event *Month* 1st" will scroll on clock display (where "*Month*" is replaced by the actual month name). This is to easily provide a demonstration of the feature without having to remember a specific date.

## *Count-down timer (run-time)*

A count-down timer can be programmed at run time and set to a specific start value. Once the count-down reaches zero, an alarm will sound (without respect to "chime time on" and "chime time off").

## *Count-up timer (run-time)*

A count-up timer may be started on the clock at run time. When started, the count-up timer will count from 00m00s and up, until stopped by user (no alarm sound is associated with this timer).

## *Date scrolling (run-time)*

The user may configure the clock to scroll the date, the ambient temperature, external temperature, relative humidity and barometric pressure (if user installed a BME280), Pico internal temperature and power supply voltage on the display at a predefined frequency (by default, every 5 minutes, which is a compile time parameter) (see section "Date scrolling" on clock setup later in the guide).

Beginning with firmware Version 2.00, the date, temperature and voltage are displayed as in "Friday July 6 2022  21.25°c  4.96 Volts" (the temperature and power supply voltage are shown after the date). The letters are made of 5 X 7 character matrix (variable-width).

Version 3.00 adds the suffix to the day of month (in the previous example: "…July 6th …". The support for outside temperature, relative humidity and barometric pressure reading is also an option available when user adds a BME280 sensor (not supplied with the Green Clock itself). See section on BME280 support later in the guide.

## *Daylight saving time (run-time)*

The firmware automatically supports the Daylight Saving Time for most northern hemisphere countries. This feature can be turned off, or other types of time change (for other countries) can be easily implemented in the source code as compile-time / run-time options.

When the clock is powered up, the current status of the daylight saving time will be quickly scrolled on the display (see section "Clock power-up sequence" above).

"North America" (defined as "1" in the clock settings) is the DST algorithm currently supported. It also corresponds to most northern hemisphere countries DST scheme. The second Sunday of March at 2h00 in the morning, the time will automatically go forward by one hour. In a similar way, the time will go back by one hour at 2h00 in the morning on the first Sunday of November.

## *Daytime hourly chime (run-time)*

This feature allows the "hourly chime" and "calendar events" to sound a "warning sound" only during the hours configured. This can prevent those alarms to sound during the night, when people are sleeping.

There is also the support for "nighttime workers" where the sounds are heard during the night. See section on how to setup chime time on and chime time off later in the guide.

### *Digits*

The digits are built with the LED matrix of the clock. There are 4 X 7 characters and also 5 X 7 characters, making the digits more elegant ("better-shaped") than the usual 7-segments displays that we use to see with other clocks on the market.

### *Double dots blinking*

There are two "dots" in the middle of the clock, between hour and minutes (see picture above in the Introduction). Note that since firmware Version 2.00 "slim" dots are used, as opposed to the "thick" dots what can be seen on the picture.

When the minute changes (say, for example, from 7h18m59s to 7h19m00s), the top dot begins to blink once a second, from 00 sec to 19 sec. Then, the bottom dot will blink from 20 sec to 39 sec. Finally, both dots will blink together from 40 sec to 59 sec. This allows someone to quickly evaluate (relatively) how "deep" we are in the current minute, and if the minute change will happen soon.

### *Hour display mode (run-time)*

The hour can be displayed in 24-hours format (00h00 to 23h59), or in 12-hours format (12h00 AM to 11h59 AM, then 12h00 PM to 11h59 PM). This can be set at run time (see section on clock setup later in the guide).

### *Hourly chime (run-time)*

Every hour, at xxh00, an hourly chime will sound to indicate that the hour just changed. This feature can be configured: On, Off, or Daytime. If configured for "Daytime" ("OI" for "On, Intermittent" in clock settings), the hourly chime will sound only during

predefined hours (set between 9h00 and 21h00 by default) (see section on clock setup later in the guide).

NOTE: If the hourly chime is Off, the “Hourly” indicator on the display will be turned Off. If the hourly chime is On, the “Hourly” indicator on the display will be turned On with two LED backlights. If the hourly chime is set to “Day” (“OI”), only the left LED behind the “Hourly” indicator will be turned On. It is not easy to see the difference between full-2-LEDs On and only left-1-LED On and it may take some practice for the user to see the difference.

### *Keyclick sound (run-time)*

A “keyclick” sound may be turned On or Off to give some feedback when the user pushes a clock button. If this option is turned On, a quick “click” will be heard each time a clock button is pressed to give the user a positive feedback that the button press has been processed by the clock (see section on clock setup later in this guide).

### *Language (run-time)*

Two languages have been implemented so far in the clock firmware for date display: French and English. Provisions have been made for other languages, so that they could be easily implemented (as long as they use the usual English-like character set). English is the default language set for firmware releases. (The “default” setting is a compile-time parameter, but language can be set / changed as a run-time parameter).

### *Night light (run-time)*

Since the two white LEDs on the side of the clock (near the buttons) were not used, I thought it would be a good idea to use them as a “night light”.

Obviously, do not expect the LEDs to provide enough light for reading or replacing your 800 lumens room light. But since they were not used anyway…

Also note that given the way the electric circuit is done, those two white LEDs will follow the green matrix LEDs brightness. So, if the clock brightness is set to automatic and ambient light is dark, the two white LEDs will be dimmed, along with the green matrix clock display.

## *Power supply voltage display (run-time)*

If “Display scroll” is enabled (see section “Display scroll” on clock setup later in the guide), then the actual power supply voltage value will scroll on the display after the date and temperature. It should be around 5 volts.

The power supply voltage reading takes advantage of one of the analog-to-digital converter integrated in the Raspberry Pi Pico.

## *Temperature display (run-time)*

The ambient temperature may be scrolled on the display, along with the date and the power supply voltage. The temperature may be shown in Celsius or Fahrenheit. This is a run time option (see section on clock setup – “Display scroll” - later in the guide).

If user installed a DHT22, it can be used to read ambient temperature and relative humidity and those two values will be scrolled on clock display.

If user installed a BME280, it can be used to read outside temperature, relative humidity and barometric pressure and those three values will be scrolled on clock display.

By default, DHT22 is used to read ambient temperature and BME280 is used to read outside temperature.

If a DHT22 is not installed, it will have no impact on normal clock behavior, but the ambient temperature and relative humidity will not be scrolled. “Pico temperature” could be used as an approximation of ambient temperature.

If a BME280 is not installed, it will have no impact on normal clock behavior, but the outside temperature, relative humidity and barometric pressure will not be scrolled.

For statistic purposes, total number of errors while reading DHT22 and BME280 are reported while scrolling the date. For example, “(5/8734)” means that there has been 5 reading errors out of 8734 readings of the DHT22. Same applies for BME280, but it is indicated between square brackets instead of parenthesis. For example: “[2/12434]”. If no DHT22 and / or BME280 has been installed, it is normal to see that both numbers are the same “(534/534)”, since every reading trial will end up in an error.

## *Options*

The following are options that are not supplied with the clock but for which support has been added in the firmware. If you provide the device required, you can turn on the conditional compile in the code and get support for it.

### DHT22

DHT22 is a temperature and humidity sensor. I added support for it in the code. I personally installed such a sensor outside to get rid of those remote temperature RF sensors which require battery change when temperature reaches -40 Celsius (yes… I live in Quebec!). However, when trying to install the device outdoors with a 3-meters cable (10 feet), I got about 80% reading errors. I strongly recommend not using a DHT22 other than locally, with a few centimeters cable. Personally, I kept the DHT22 for ambient temperature reading since it also gives ambient relative humidity.

### BME280

BME280 is a temperature, humidity and barometric pressure sensor. I also added support for it in the code. In fact, it shares the same I2C line protocol as the real time clock IC.

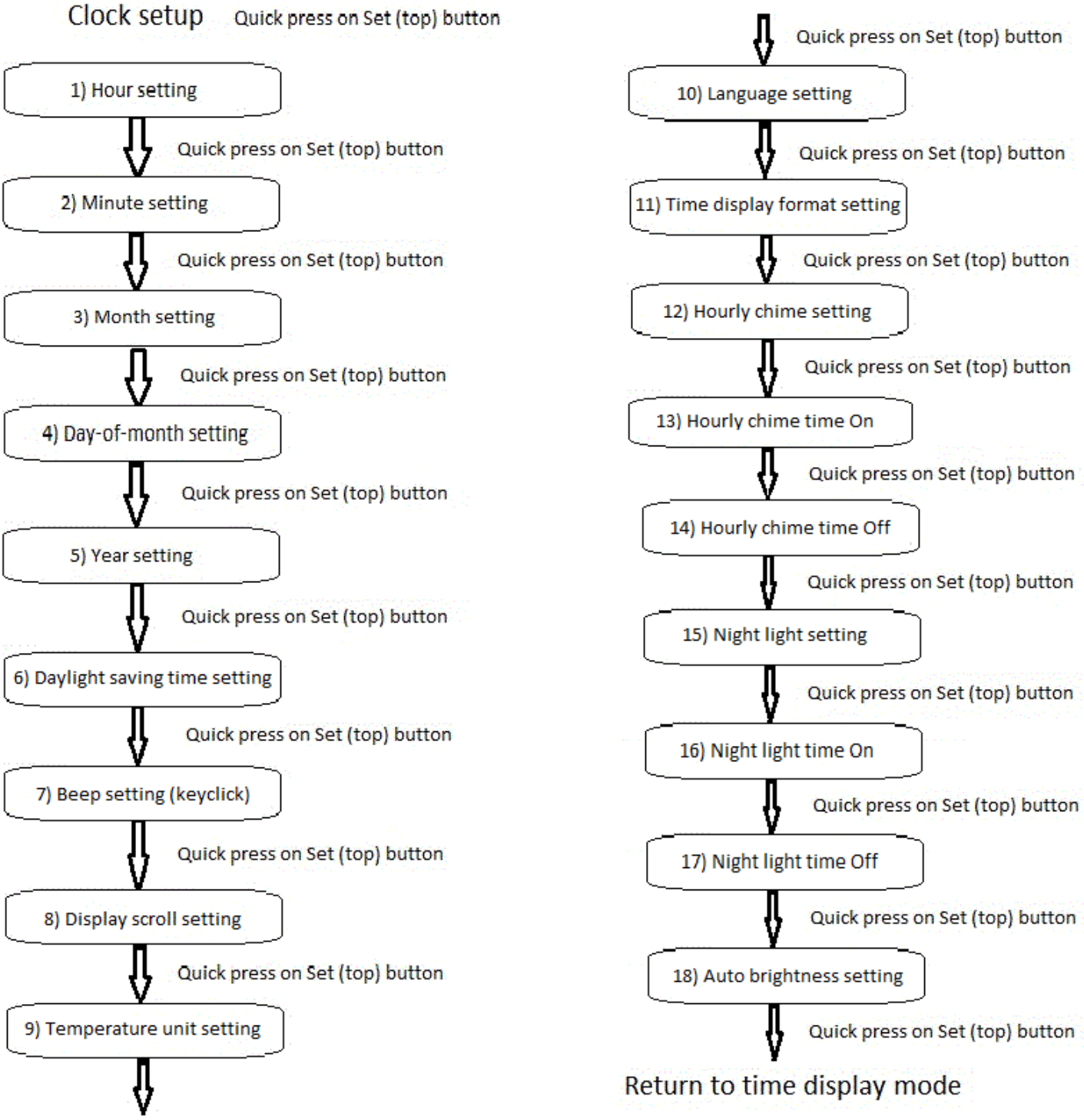
### Infra-red sensor

I also added an infrared sensor to receive commands from a remote control. Since one of my clocks will be installed in a relatively high location, remote control becomes very handy.

### Passive buzzer / piezo

As opposed to the buzzer that comes with the clock, a passive buzzer does not have an integrated oscillator. It is then possible to drive the piezo with an external signal. By changing the frequency of this signal, we can change the audio frequency generated and play “tunes” / “jingles”.

## Clock setup




Refer to the paragraph – in the following section - with the number indicated in the box above to get more details about each step of the clock setup.

## *About clock buttons*

Remember the name given to the three clock buttons:

- The Top button is called “Set” button since it is primarily used to proceed with settings.
- The Middle button is called “Up” button since it is primarily used to increase values (like hour setting, minute setting, etc).
- The Bottom button is called “Down” button since it is primarily used to decrease values (like hour setting, minute setting, etc).

When not otherwise mentioned, a “button press” means a “quick” button press, which is less than 300 milliseconds (shorter than one third of a second). In some cases, the clock will handle a “long press” (longer than one third of a second) by performing a different behavior than for a quick press. So, keep in mind the difference between a “quick press” and a “long press”.

## *Entering clock setup mode*

To enter clock setup mode, make a quick press on the “Set” (top) button while the clock is in the usual “Time display” mode.

Each time you press on the “Set” (top) button, you navigate through the different clock settings, as illustrated on the diagram above.

To exit clock setup mode, you can:

- Wait for a timeout (20 seconds without pressing a button on the clock – see next section).
- Press the “Set” (top) button to scan all settings and then, press it once more when reaching the last setup step to return to the time display mode.
- Make a “long press” (longer that one third of a second) on the “Down” (bottom) button to exit clock setup mode at any time.

## *Clock setup timeout*

If you leave the clock unattended for more than 20 seconds (changing the value of “20 seconds” is a compile-time option), it will return to the time display mode. Any change that has been made so far during this clock setup session will be saved in the clock settings.

### *1) Hour setting*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Hour setting” step.

Current hour will blink on the clock display. Press the “Up” (middle) button to increase the hour value, or press the “Down” (bottom) button to decrease the hour value.

Depending on the current “Hour display format” (12-hour or 24-hour format), hour setting will go from 1 to 12 along with the AM / PM indicator, or from 00 to 23, without AM / PM indicator.

Take note that when you complete the clock setup, the integrated real-time clock integrated circuit (“RTC IC”) will be programmed with the new time and date and the seconds will start from zero. This IC is backed-up with a battery and will keep the accurate time in case of a power failure.

### *2) Minute setting*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Minute setting” step.

Current minute will blink on the clock display. Press the “Up” (middle) button to increase the minute value, or press the “Down” (bottom) button to decrease the minute value.

Take note that when you complete the clock setup, the integrated real-time clock integrated circuit (“RTC IC”) will be programmed with the new time and date and the seconds will start from zero.

### *3) Month setting*

NOTE: Given the difference in the date format between French and English, the sequence for setting the month and the day-of-month is reversed when the clock language is set to French.

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Month setting” step.

Current month will blink on the clock display. Press the “Up” (middle) button to increase the month value, or press the “Down” (bottom) button to decrease the month value.

The day-of-month may be automatically changed / updated in some occasion. For example, if day-of-month is currently set to 31, and you change the month from March to February, the day-of-month will automatically change from 31 to 28 (or 29 on a leap year) to comply with the upmost value for day-of-month in February.

### *4) Day-of-month setting*

NOTE: Given the difference in the date format between French and English, the sequence for setting the month and the day-of-month is reversed when the clock language is set to French.

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Day-of-month” setting step.

Current day-of-month will blink on the clock display. Press the “Up” (middle) button to increase the day-of-month value, or press the “Down” (bottom) button to decrease the day-of-month value.

The day-of-month will not allow you to go higher than the upmost value for any given month. So, you may have to adjust the month before adjusting the day-of-month. For example, if you want to change the date from February 14$^{th}$ to March 31$^{st}$, day-of-month will not allow you to go higher than 28 (or 29) as long as the month is set to February.

### *5) Year setting*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Year setting” step.

Current year (last two digits) will blink on the clock display. Press the “Up” (middle) button to increase the year value, or press the “Down” (bottom) button to decrease the year value.

Even if only the last two digits of the year are blinking, the year will go from 2000 down to 1999 and / or from 2099 up to 2100 as we would expect if we decrease or increase year value. However, if there is a power outage, when the power goes back on, the high part of the year (first two digits) will revert to “20”. (Which shouldn’t a problem, except if you plan to install the Pico Green Clock in the next Electric De Lorean, to go back in 1845 or forward in 2187!).

### *6) Daylight Saving Time (DST)*

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Daylight saving time" setting step.

"ST" (for "saving time") will be displayed on the clock, along with a number representing current setting. The value can be changed by pressing either on the "Up" (middle) or "Down" (bottom) button.

Value "0" means that the clock completely ignores DST (it's like if DST does not exist).

Value "1" means that the clock support DST, following the North America standard (which is also compliant to most northern hemisphere countries). The second Sunday of March at 2h00 in the morning, the time will automatically go forward by one hour. In a similar way, the time will go back by one hour at 2h00 in the morning on the first Sunday of November. If ever there is a power failure between 2h00 and 3h00 in the morning the night the DST changes, the clock may be wrong in guessing DST setting. Re-starting the clock a few hours later will fix the problem.

### *7) Keyclick setting*

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Keyclick setting" step.

"BP" (for "beep") will be displayed on the clock, along with the value "ON" or "OF" (for OFF) blinking. The value can be changed by pressing either on the "Up" (middle) or "Down" (bottom) button.

This setting controls the "keyclick sound" produced when the user presses a clock button. ON allows the keyclick to be heard each time the user presses a button (to give a positive feedback) whereas OFF makes no sound when a button is pressed.

### *8) Display scroll setting*

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Display scroll setting" step.

"DS" (for "Display scroll") will be displayed on the clock, along with the value "ON" or "OF" (for OFF) blinking. The value can be changed by pressing either on the "Up" (middle) or "Down" (bottom) button.

This setting controls if the clock will scroll the date, temperature and power supply voltage on the display every 5 minutes. (The setting can be "ON" or "OFF". Changing the 5 minutes period is a compile-time option).

NOTE: Beginning with Version 4.00, the display will no longer scroll every xxx seconds as was the case with previous versions. Instead, the display will scroll every "integer number of times the scroll period" as defined in the source code. For example, when set to 5 minutes period, scrolling will occur at xh00, xh05, xh10, xh15, etc… this allows the user to know in how long the next scroll will occur. In previous versions, scroll would have occurred every 300 seconds (5 minutes X 60 seconds each). Depending the exact time when the clock has been powered up, this 300 seconds count would have not necessarily be at xh00, xh05, etc… The scrolling time is now more deterministic.

NOTE: Beginning with Version 5.00, scrolling will begin 5 seconds after hour change. So, for example, scrolling will begin at 4h00m05s instead of 4h00m00s like in previous versions. This way, when the hourly chime sounds (at 4h00m00s in our example), it gives five seconds to look at the clock display to see what is the time, before beginning to scroll the date and other information. This is true only at hour change. So, when time changes from 4h04m59s to 4h05m00s, scrolling will begin without delay (and obviously, there is no "hourly chime" at 4h05m00s.

### *9) Temperature unit*

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Temperature unit setting" step.

NOTE: Starting with firmware version 4.00, temperature unit setting has been added to the main clock setup list of parameters.

"TU" (for "Temperature unit") will be displayed on the clock, along with current setting ("°C" or "°F") blinking. The setting can be toggled from one to the other by pressing either on the "Up" (middle) or "Down" (bottom) button. The indicator on clock display will follow selected setting, between Celsius and Fahrenheit.

### *10) Language*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Language setting” step.

“LG” (for “Language”) will be displayed on the clock, along with current setting. The setting can be changed by pressing either on the “Up” (middle) or “Down” (bottom) button.

Languages supported for now are:
“E” – for English
“F” – for French

More languages should be easily added in the code. If ever we reach a point where the firmware becomes too big to fit in Pico’s flash memory, this is an option that could be transferred to compile-time instead of run-time. That being said, we are still far from the 2MB flash of the Pico.

Language setting is mostly useful when scrolling the date on the display where day-of-week and month are displayed in the selected language. But there are other messages that change depending on language setting.

### *11) Hour display format setting*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Time display format setting” step.

“HD” (for “Hour display”) will be displayed on the clock, along with the value “12” (for 12-hour format) or “24” (for 24-hour format) blinking. The value can be changed by pressing either on the “Up” (middle) or “Down” (bottom) button.

If 12-hour format is selected, indicator AM or PM will show-up on the clock to complete the time information.

### *12) Hourly chime setting*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Hourly chime setting” step.

“HC” (for “Hourly chime”) will be displayed on the clock, along with current setting “ON”, “OF” (for OFF) or “OI” (for ON, Intermittent) blinking. The value can be changed by pressing one or more times on the “Up” (middle) or “Down” (bottom) button.

Hourly chime is a sound made of a few beeps that will be produced each time the hour changes from xxh59 to xxh00. The ON and OFF configurations are self-explanatory, but the OI ("On, Intermittent") means that the sound will be heard only during daytime (so that people in the house can sleep without being disturbed by these sounds every hour during the night). The settings (On, Off, OI) are run-time options, along with starting and ending time (see next two settings).

There is an indicator on the clock display that will light up when hourly chime is ON. Behind this indicator, there are two LEDs that will turn on. When the option is set to "OI" (daytime), then only the left LED will turn on. It is not easy to make the difference between one LED On or two LEDs On and it may take some time for the user to see the difference between both settings from the indicator.

Firmware versions 3.00 and up add support for "nighttime workers". We assume that it is possible, in some cases, that we want the hourly chime to sound during the night but be silent during the day. So, if the Chime time on is later than the Chime time off (which is opposed to what we would normally expect), we assume that we want the sound to be heard *after* the Chime time on and *before* the Chime time off.

NOTE: In case of power outage, Hourly chime setting will revert to compile-time setting.

NOTE: The Hourly chime setting (On, Off, OI) also applies to the sounds generated during calendar events.

NOTE: It you have installed a passive buzzer, you will also hear a jingle (close encounter of the 3rd kind) after the usual hourly chime.

### *13) Chime time on*

NOTE: Even if you selected "Hourly chime On" or "Hourly chime Off" in the previous setup, you still have to go through the "Chime time On" and "Chime time Off" settings (even if they are useless if Hourly chime setting is not "OI"). This makes the setup procedure always the same, no matter what are the clock parameter settings.

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Chime time on setting" step.

"ON" (for "Chime time on") will be displayed on the clock along with the current setting blinking. The value can be changed by pressing the "Up" (middle) button to increase it, or the "Down" (bottom) button to decrease it. The default value for chime time on is 9h00 in the morning.

The chime time on is the time at which the hourly chime will begin to sound during the day. See also the chime time off in next setting. Note that the sound for the calendar events also complies with the chime time settings.

NOTE: The hourly chime setting must be set to "Daytime" (that is: "OI" for "On, Intermittent" on the clock display) for the Chime time on setting to take effect.

NOTE: In case of a power failure, Chime time on and Chime time off will revert to their default compile-time values (respectively 9h00 and 21h00 – 9h00 PM).

NOTE: The discussion in this paragraph applies for a "normal" daytime setting. The behavior described must be adapted "mutatis mutandis" if the Chime time on is later than Chime time off, what has been called "nighttime worker" setting.

### *14) Chime time off*

NOTE: Even if you selected "Hourly chime On" or "Hourly chime Off" in the previous setup, you still have to go through the "Chime time On" and "Chime time Off" settings (even if they are useless if Hourly chime setting is not "OI"). This makes the setup procedure always the same, no matter what are the clock parameter settings.

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Chime time off setting" step.

"OF" (for "Chime time off") will be displayed on the clock along with the current setting blinking. The value can be changed by pressing the "Up" (middle) button to increase it, or the "Down" (bottom) button to decrease it. The default value for chime time off is 21h00 (9h00 PM).

The chime time off is the last time at which the hourly chime will sound during the day. See also the chime time on in previous setting. Note that the sound for the calendar events also complies with the chime time settings.

So, for example, if you set the chime time on at 8h00 in the morning and the chime time off at 23h00 (11h00 PM), the hourly chime will sound every day (each time the hour changes from xxh59 to xxh00) from 8h00 in the morning up to (and including) 23h00 (11h00 PM). The warning sound associated with the calendar events will also comply with the same on and off times. By default, the on and off times are set to 9h00 in the morning up to (and including) 21h00 (9h00 PM) in the evening. (Obviously, this assumes that "Hourly chime setting" is set to ("OI" – On, Intermittent).

NOTE: The hourly chime setting must be set to "Daytime" (that is: "OI" for "On, Intermittent" on the clock display) for the Chime time off setting to take effect.

NOTE: In case of a power failure, Chime time on and Chime time off will revert to their default compile-time values (respectively 9h00 and 21h00 – 9h00 PM).

NOTE: The discussion in this paragraph applies for a "normal" daytime setting. The behavior described must be adapted "mutatis mutandis" if the Chime time on is later than Chime time off, what has been called "nighttime worker" setting.

### *15) Night light setting*

While referring to the Clock setup diagram above, press the "Set" (top) button until you reach the "Night light setting" step.

"NL" (for "Night light") will be displayed on the clock, along with current setting blinking: "ON", "OF" (for OFF), "OI" (for ON, Intermittent) or "AU" (for "Auto"). The setting can be changed by pressing one or more times on the "Up" (middle) or "Down" (bottom) button.

The night light uses the two white LEDs located inside the clock, near the push buttons. You can watch them during the power up sequence of the clock since they will be turned on for a while and also blink a few times to indicate that the clock is starting. Since those LEDs were not used, I thought it would be a good idea to use them as a night light, even if the light level is quite low. For sure, they will not replace your 800 lumens room light!

In some ways, the setting of the night light is similar to the setting of the hourly chime. The ON and OFF configurations are self-explanatory, and the OI ("On, Intermittent") means that the night light will be ON only during nighttime. As opposed to the hourly chime settings, however, the night light allows the LEDs to be turned on only during the night. There is no provision to turn on LEDs during the day for "daytime workers" since I thought it would have not been logical to do so.

There is also another setting available for the night light: "AU" for "Auto". If set to automatic, the LEDs will turn on when the ambient light falls under a certain level and becomes dark enough. They will automatically turn back off when ambient light level is bright enough. There is a "twilight zone" between bright and dark during which the night light status does not change. This is to prevent continuous night light blinking between On and Off when we are in the light brightness area close to the toggle area.

The night light setting (ON, OF, OI, AU) are run-time options, along with starting and ending time (see next two settings).

NOTE: In case of power outage, Night light setting will revert to compile-time setting ("AU" - automatic).

NOTE: When “Auto-brightness” setting is enabled, keep in mind that the white LEDs will follow the clock display brightness. That is, if the clock LEDs matrix display is dimmed, the white LEDs will also be dimmed. This is the way the clock electronics work.

NOTE: Every few seconds, the firmware checks the status of the night light settings. So, if you change the clock time, the white LEDs will eventually comply with the night light settings in the next few seconds…

### *16) Night light time on*

NOTE: Even if you selected “Night light On”, “Night light Off” or “Night light Auto” in the previous setup, you still have to go through the “Night light time On” and “Night light time Off” settings (even if they are useless if Night light setting is not “OI”). This makes the setup procedure always the same, no matter what are the clock parameter settings.

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Night light time on” setting step.

“ON” (for “Night light time on”) will be displayed on the clock along with the current setting blinking. The value can be changed by pressing the “Up” (middle) button to increase it, or the “Down” (bottom) button to decrease it. The default value for night light time on is 21h00 in the evening.

The night light time on is the time at which the night light will turn on in the evening. See also the night light time off in next setting.

NOTE: The night light setting must be set to “Nighttime” (that is: “OI” for “On, Intermittent” on the clock display) for the Night light time on setting to take effect.

NOTE: In case of a power failure, Night light time on and Night light time off will revert to their default compile-time values (respectively 21h00 and 8h00). However, “Night light setting” defaults to “Auto” (which doesn’t care about time on and time off).

NOTE: The two white LEDs are subject to the “Auto brightness” setting and will follow the clock display brightness. That is, if the main clock display is dimmed, the two white LEDs will be dimmed too. This is the way the clock electronics work.

### *17) Night light time off*

NOTE: Even if you selected “Night light On”, “Night light Off” or “Night light Auto” in the previous setup, you still have to go through the “Night light time On” and “Night light time Off” settings (even if they are useless if Hourly chime setting is not “OI”). This makes the setup procedure always the same, no matter what are the clock parameter settings.

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Night light time off” setting step.

“OF” (for “Night light time off”) will be displayed on the clock along with the current setting blinking. The value can be changed by pressing the “Up” (middle) button to increase it, or the “Down” (bottom) button to decrease it. The default value for night light time off is 8h00 in the morning.

The night light time off is the time at which the white LEDs will turn off in the morning. See also the night light time on in previous setting.

So, for example, if you set the night light time on at 23h00 in the evening and the night light time off at 6h00 in the morning, the night light will turn on every night at 23h00 and will turn off every morning at 6h00.

By default, the on and off times are set to 21h00 in the evening and 8h00 in the morning. (Obviously, this assumes that “Night light setting” is set to (“OI” – On, Intermittent).

NOTE: The night light setting must be set to “Nighttime” (that is: “OI” for “On, Intermittent” on the clock display) for the Night light time off setting to take effect.

NOTE: In case of a power failure, Night light time on and Night light time off will revert to their default compile-time values (respectively 21h00 and 8h00). However, “Night light setting” defaults to “Auto” (which doesn’t care about time on and time off).

NOTE: The two white LEDs are subject to the “Auto brightness” setting and will follow the clock display brightness. That is, if the main clock display is dimmed, the two white LEDs will be dimmed too. This is the way the clock electronics work.

### *18) Auto brightness setting*

While referring to the Clock setup diagram above, press the “Set” (top) button until you reach the “Auto brightness setting” step.

“DIM” will be displayed on the clock. There is no “visible setting” beside “DIM”, as opposed to most other settings that we went through so far. Instead, user must take a look at the auto-light indicator in the bottom left of the clock.

The setting can be changed by pressing on the “Up” (middle) or “Down” (bottom) button. If the “Auto light” indicator is on, it means that the clock display will be bright if the ambient light level is high and will be dimmed if the ambient light level is low.

In firmware version 5.00, the algorithm behind the auto brightness function has been changed. The clock LEDs matrix brightness will now change more slowly than what it used to do before (some “inertia” has been added to the ambient light change). I didn’t like the display that was often changing intensity while some object was passing in front of the clock or for a similar reason. Brightness change has now a greater latency, so don’t be surprised if you turn off the light in the night. It will take many seconds (up to one minute) for the clock display to slowly dim, down to its stabilized level (and the same applies if you turn on the light in a previously dark room).

NOTE: The two white LEDs (“night light”) follow the same rule as the clock display. So, if auto brightness is enabled, the two white LEDs will also be dimmed if ambient light level is low.

# Alarm setup

## *About alarm setup*

There are two independent alarms that can be set on the Waveshare Green Clock (Alarm 1 and Alarm 2).

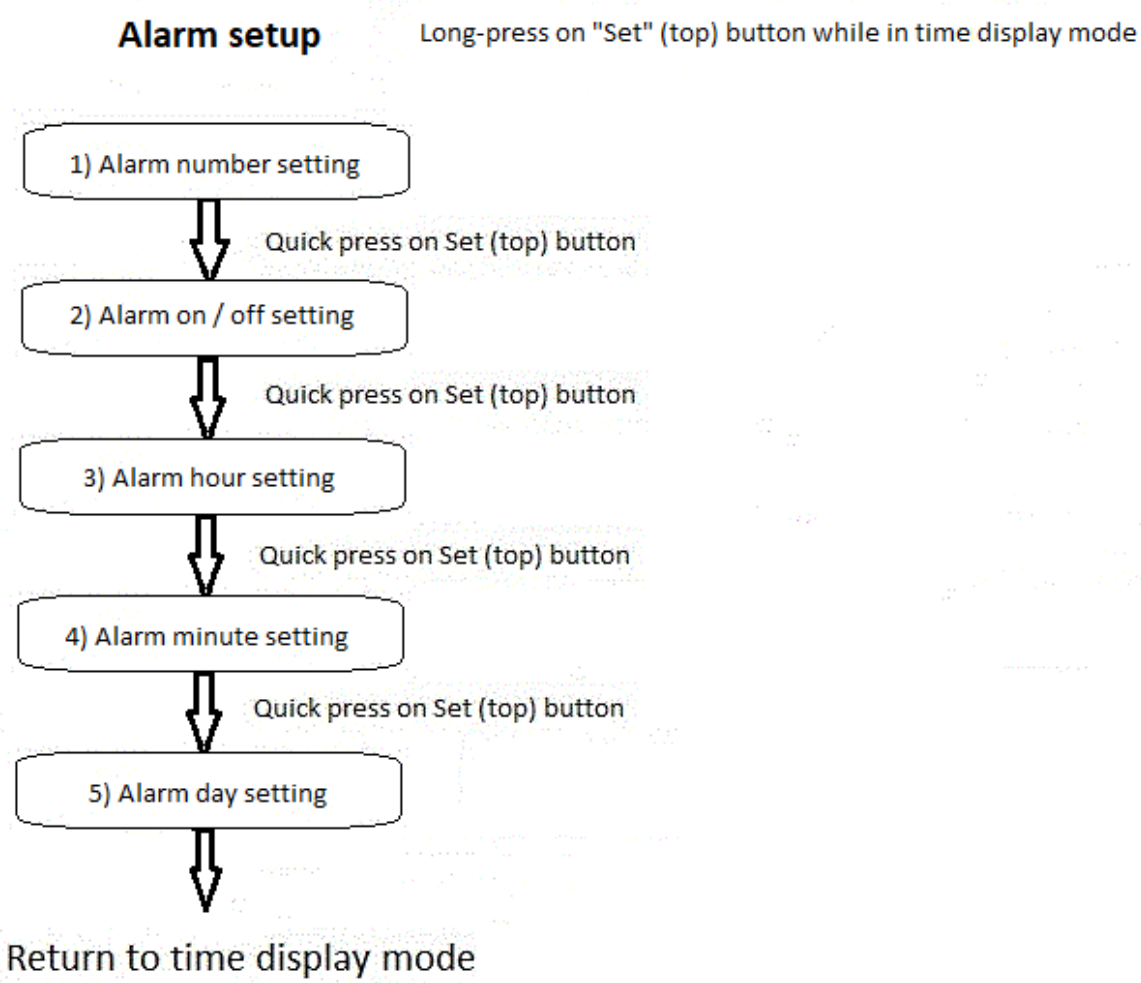
Alarm parameters (Hour, Minute, Day-of-week) are kept in the Real-time-clock integrated circuit (“RTC IC”). So, they will still be remembered by the system in case of restart and / or power failure. However, alarm status (“On” or “Off”) is not battery backed-up. In case of restart / power failure, alarm status for both alarms will be reset to “Off” (that is: Alarms disabled).

If some alarm parameters are set (that is: hour, minute and day-of-week), but the On / Off setting remains to Off, the parameters will not be saved in the RTC IC. In other words, an alarm is saved in the RTC IC only when it is set to “On”.

When an alarm (1 or 2) is on and the configured hour, minute and day-of-week is reached, an alarm sound will be heard. Alarm sound is independently defined in the code (compile-time parameter) and may be different for alarm 1 and alarm 2. Also, alarm sound does not comply with Hourly chime settings. That is, alarm will sound no matter what is the setting for Hourly chime.

NOTE: If “NO_SOUND” is defined in the source code at compile time, absolutely no sound will be heard from the clock (no keyclick, no hourly chime, no alarm, no timer

alarm, …). The message “WARNING: SOUND CUT OFF” will scroll on the clock display during the powered up sequence.




Refer to the paragraph – in the following section - with the number indicated in the box above to get more details about each step of the alarm setup.

## *Entering alarm setup mode*

To enter the alarm setup mode, make a long press (longer than one third of a second) on the “Set” (top) button while the clock is in the usual “Time display” mode.

Then, each time you press on the “Set” (top) button, you go through the different alarm settings, as illustrated on the diagram above.

To exit alarm setup mode, you can:

- Wait for a timeout (20 seconds without pressing a button on the clock).
- Press the “Set” (top) button to go through all alarm settings and then one more time when reaching the last setup step to return to the time display mode.
- Make a “long press” (longer that one third of a second) on the “Down” (bottom) button to exit alarm setup mode at any time.

## *Alarm setup timeout*

If you leave the clock unattended for more than 20 seconds (compile-time option), it will return to the time display mode.

### *1) Alarm number setting*

While referring to the Alarm setup diagram above, press the “Set” (top) button until you reach the “Alarm number setting” step.

Press the “Up” (middle) button or the “Down” (bottom) button to alternate between alarm number 1 and alarm number 2 (as mentioned earlier in the manual, there are two independent alarms available in the Waveshare Green Clock).

The proposed choice will blink on the display until the user presses the “Set” (top) button to select the choice displayed and to proceed with next step.

NOTE: Alarm numbers are 1 and 2 for the clock user, to be more “human-like”. However, when working on the firmware source code, alarm numbers are 0 and 1 respectively.

### *2) Alarm on / off setting*

While referring to the Alarm setup diagram above, press the “Set” (top) button until you reach the “Alarm on / off setting” step.

Once the alarm number (1 or 2) has been selected in the previous step, press the “Up” (middle) button or the “Down” (bottom) button to turn On or Off the selected alarm.

The proposed choice will blink on the display until user presses the “Set” (top) button to select the choice displayed and to proceed with next step.

In a way similar to the Hourly chime “Daytime” setting, when alarm number 1 is on, the left LED in the alarm indicator will turn on. When alarm number 2 is on, the right LED in the alarm indicator will turn on. It may take some practice to be able to see the difference between left LED On, right LED On, or both LEDs On.

### *3) Alarm hour setting*

While referring to the Alarm setup diagram above, press the “Set” (top) button until you reach the “Alarm hour setting” step.

The current hour value for the selected alarm will be displayed and blinks. While current alarm hour blinks on the display, press the “Up” (middle) button to increase the proposed hour, or the “Down” (bottom) button to decrease it. The display will comply with current “Time format setting” to display the alarm in 12-hours or 24-hours display mode.

### *4) Alarm minute setting*

While referring to the Alarm setup diagram above, press the “Set” (top) button until you reach the “Alarm minute setting” step.

The current minute value for the selected alarm will be displayed and blinks. While current alarm minute blinks on the display, press the “Up” (middle) button to increase the proposed minute, or the “Down” (bottom) button to decrease it.

### *5) Alarm day setting*

While referring to the Alarm setup diagram above, press the “Set” (top) button until you reach the “Alarm day setting” step.

The current day-of-week setting for the selected alarm will be displayed and blinks. At this point, you can select the day-of week for the selected alarm. Press the “Up” (middle) button to go to the next day-of-week, or the “Down” (bottom) button to go to the previous day-of-week.

Then make a quick press on the “Set” (top) button to return to the time display mode. If the selected alarm status has been turned On, the alarm indicator on the clock display (left LED and / or right LED) will have been turned on and the alarm parameters will have been saved in the RTC IC. Remember that settings will not been saved if alarm was not set to “On”.

# Timer setup

## *About timer setup*

The Pico Green Clock features a timer that can be set either as a “count-up” timer, or as a “count-down” timer. When used for count-down, you enter a start time (in minutes and seconds) and the timer begins the count-down, toward zero. When it reaches zero, it beeps a ‘Timer alarm”.

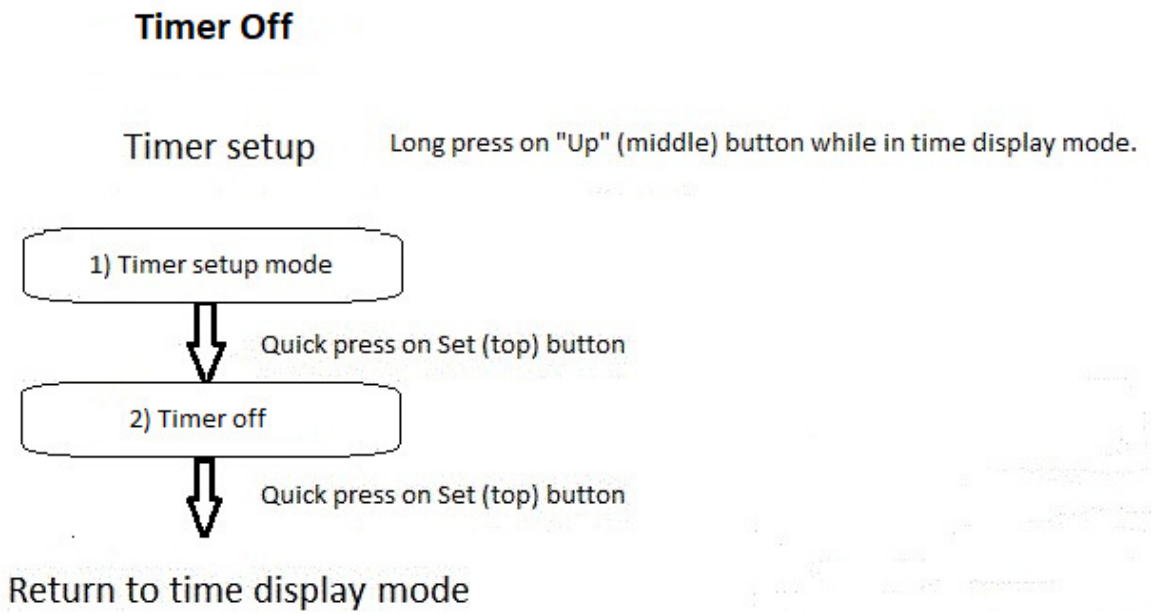
When used for “count-up”, it is different since we logically expect a count-up timer to start from zero. So, when started, the count-up timer begins counting up from zero until it is stopped by the user. There is no alarm sound associated with the count-up timer.

To enter the timer setup mode, make a “long press” on the “Up” (middle) button. A long press means to hold the button for more than one third of a second.

As you can do for Clock setup and Alarm setup, you can make a “long press” (longer that one third of a second) on the “Down” (bottom) button to exit timer setup mode at any time.

NOTE: When used as a “count-down timer”, it will not start if the “Minute” is set, then the “Second” but if the Set (“Top”) button is not pressed to start it before the time-out period.

## *Timer OFF*




Refer to the paragraph – in the following section - with the number indicated in the box above to get more details about each step of the timer off setup.

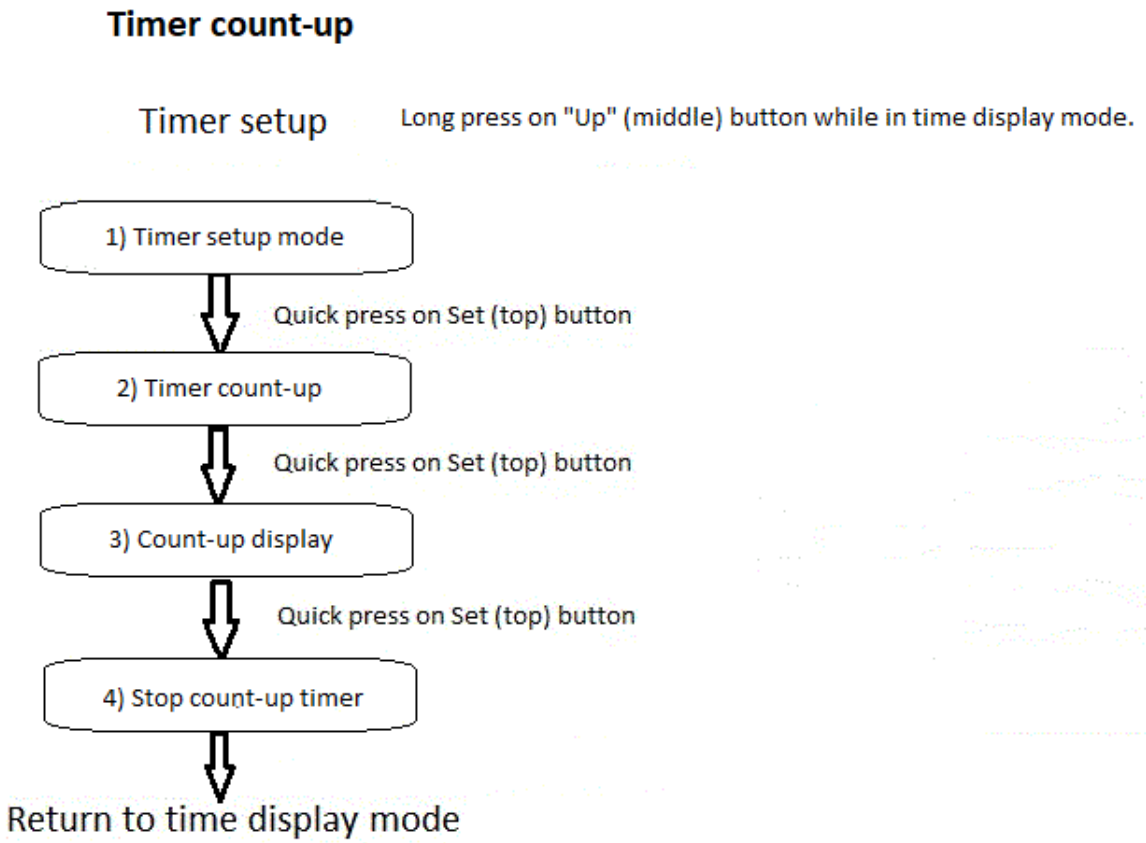
1) To turn off the timer, refer to the diagram above for “Timer off” and proceed as follow. First, make a long press on the “Up” (middle) button to enter the timer setup mode.

   Then, using the “Up” (middle) or “Down” (bottom) button, press until “OF” (for “Off”) blinks on the clock display and press on the Set (top) bottom to select this choice. Choices are “OF” (for off), “DN” (for count-down timer), or “UP” (for count-up timer).

2) When pressing the “Set” (top) button, “OF” will stop blinking on the display to show that you made that selection.

   Press on the “Set” (top) button once again to return to time display mode when the timer has been turned off (or wait for a time-out).

## *Timer Count-up*




Refer to the paragraph – in the following section - with the number indicated in the box above to get more details about each step of the timer count-up setup.

1) To start the timer in "count-up" mode, refer to the diagram above for "Timer count-up" and proceed as follow. First, make a long press on the "Up" (middle) button to enter the timer setup mode.

   Then, using the "Up" (middle) or "Down" (bottom) button, press until "UP" (for count-up timer) blinks on the clock display and press on the "Set" (top) bottom to select this choice. Choices are "OF" (for off), "DN" (for count-down timer), or "UP" (for count-up timer). The corresponding indicator on the clock display will follow the proposed choice (Off / Count-up / Count-down).

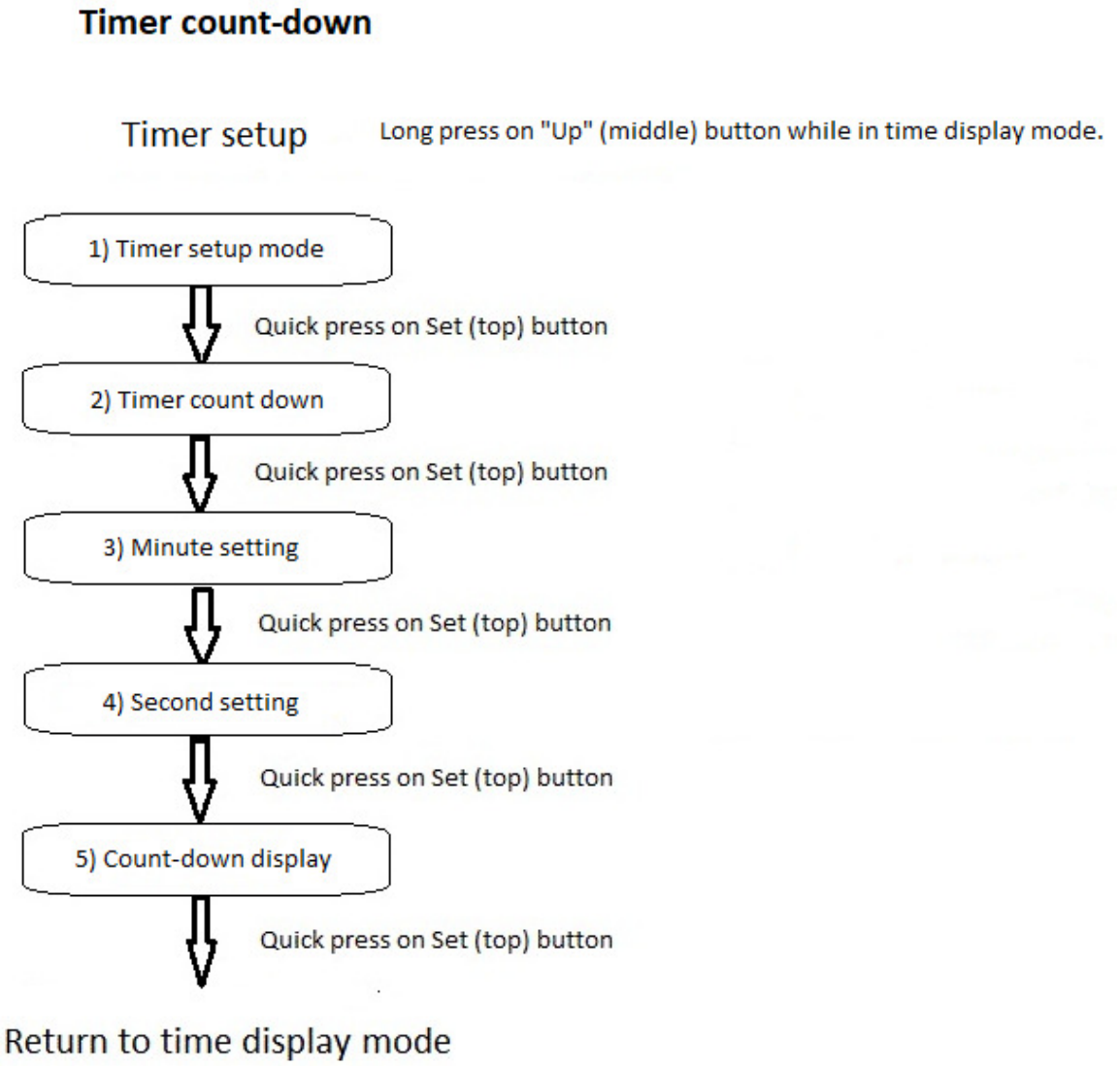
2) When pressing the "Set" (top) button, "UP" will stop blinking on the display to show that you made that selection.

3) Press on the “Set” (top) button once again to see the count-up timer starting to count the time from 00m00s.

4) The count-up timer will count until the user presses on the “Set” (top) button to stop it.

5) At this point, user may press on the “Set” (top) button again to return to the time display mode, or wait for a time-out to return to the time display mode. If the user decides to return to the time display mode, he can come back to the timer setting later to see the count-up timer value.

## *Timer Count-down*




Refer to the paragraph – in the following section - with the number indicated in the box above to get more details about each step of the timer count-down setup.

1) To start the timer in “count-down” mode, refer to the diagram above for “Timer count-down” and proceed as follow. First, make a long press on the “Up” (middle) button to enter the timer setup mode.

   Then, using the “Up” (middle) or “Down” (bottom) button, press until “DN” (for count-down timer) blinks on the clock display and press on the “Set” (top) bottom to select this choice. Choices are “OF” (for off), “DN” (for count-down timer), or

"UP" (for count-up timer). The corresponding indicator on the clock display will follow the proposed choice (Off / Count-up / Count-down).

2) When pressing the "Set" (top) button, "DN" will stop blinking on the display to show that you made that selection.

   Press on the "Set" (top) button once again and the clock will display the count-down starting value with minutes blinking.

3) Press on the "Up" (middle) or "Down" (bottom) button to adjust the minutes of the count-down timer. Press the "Set" (top) button when done and the minutes will stop blinking as the seconds begin to blink.

4) Press on the "Up" (middle) or "Down" (bottom) button to adjust the seconds of the count-down timer. Press the "Set" (top) button when done and the count-down timer will start the count-down until it reaches zero minute and zero second.

5) At this point, you can leave the clock like this to see the count-down actual value at any time. When reaching zero, the "timer alarm" sound will be heard to warn you that the count-down value is over and the clock will automatically return to time display mode.

   Alternately, you can press the "Set" (top) button to return to the time-display mode while the count-down timer continues its work. The count-down timer will proceed to count until zero (as can be seen by the indicator "count-down" light-up on the display). Even if you are in the "time display" mode, an alarm will be heard when the count reaches zero. The "count-down" indicator on the clock display will turn off when the count-down is over. You can come back to the timer mode at any time to see the actual count of the count-down timer.

   It must be noted that the chime time on and the chime time off have no impact on the count-down timer alarm. The alarm will sound even if we are currently outside of the chime defined hours.

# Other settings

## *Quick access to outside temperature*

As mentioned in the clock setup procedure described previously, beginning with firmware version 4.00, the “Temperature unit” setting (toggling between Celsius and Fahrenheit) has been migrated from the “Up” (middle) button to the main list of clock setup parameters.

So, the “Up” (middle) button behavior has been used for other purposes. When pressing this button now, if an outside temperature sensor (BME280) has been installed, the outside temperature will scroll on clock display when pressing the “Up” (middle) button.

NOTE: The night light will also toggle (On or Off), for a few seconds, depending on its current setting.

## *Auto brightness*

As mentioned in the clock setup procedure described previously, beginning with firmware version 4.00, the “Auto brightness” setting (toggling between full brightness and auto brightness has been migrated from the “Down” (bottom) button to the main list of clock setup parameters. However, for a handy quick access, this function has been kept (replicated) on the “Down” (bottom) button. So, you can quickly toggle the setting this way too.

## Appendix A – Access to BOOTSEL button

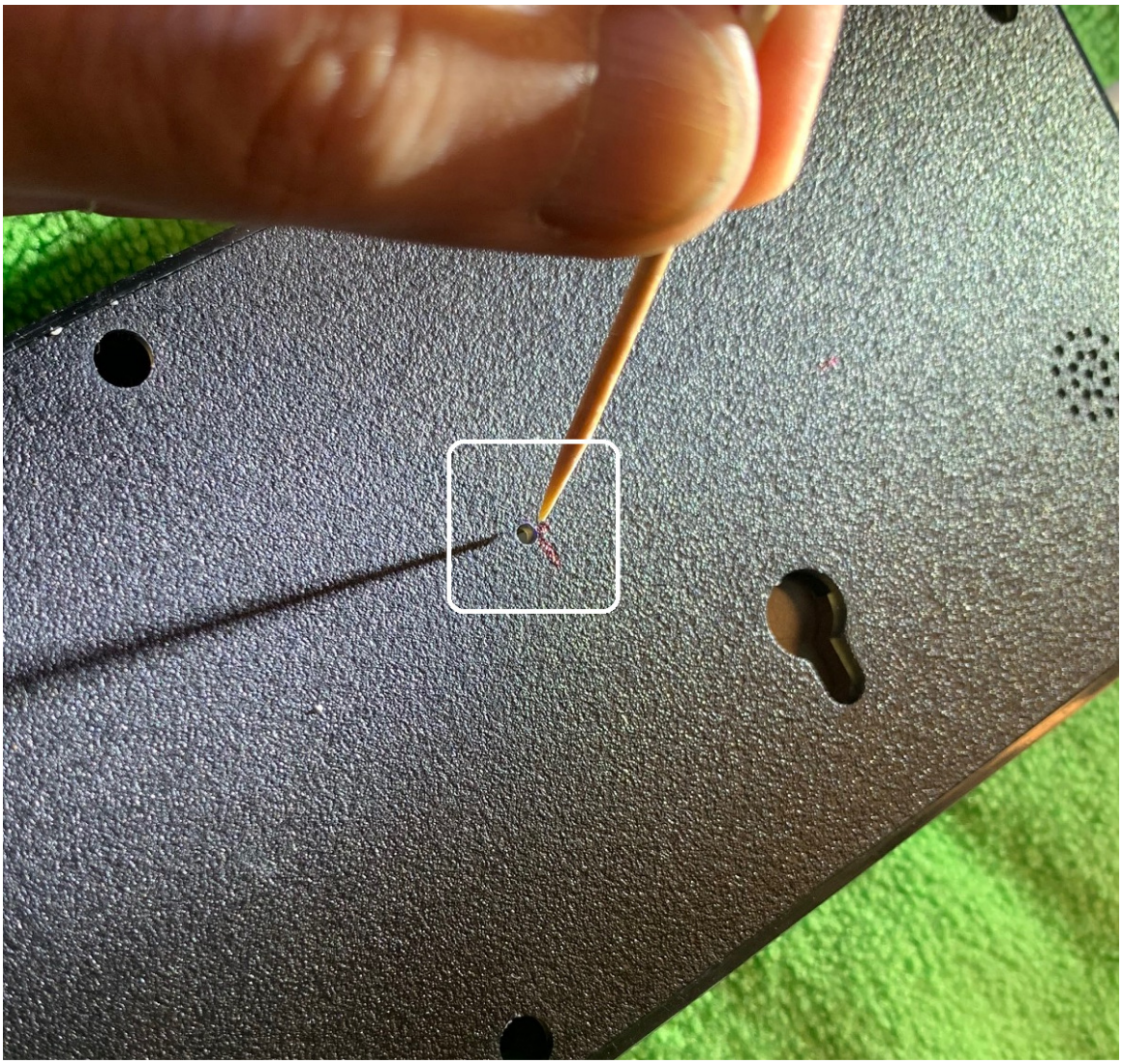
I followed David Ruck suggestion (David is also active on GitHub) and managed an access to the BOOTSEL button on the Pico. This allows re-flashing the Pico without having to open the clock case every time.

See pictures below as a guide. Take note that these pictures show the back cover on top of the clock case. If you take the back cover alone (without the clock case), you have to remove one mm from the values shown (you can see the case border – 1 mm – around the back cover)..


Access to BOOTSEL button – showing narrow side 35 mm from bottom (or 36 mm with case border).


Access to BOOSEL button – showing wide side 82.5 mm from left (or 83.5 mm with case border).

As illustrated on the pictures, I suggest that you use a wooden toothpick to press the BOOTSEL button to prevent possible damage to internal parts that could occur if you use a metal device.



Access to the BOOTSEL Pico button is possible through the hole.

## Appendix B – Prepare cabling for external devices

As mentioned earlier in the guide, so far I installed / added four external devices to the Pico Green clock:

1) An external temperature, humidity and barometric pressure sensor (BME280).
2) An ambient temperature and humidity sensor (DHT22).
3) An infra-red receiver to support commands sent from a remote control.
4) A passive buzzer / piezo allowing changing the sound frequency.

To install these external devices, we first need to make a hole on the side of the clock plastic case so that we can extend a cable outside of the case to connect the devices.

See the picture below. However, you may want to make the hole farther from the bolt stud so that it does not interfere.

Hole in the plastic case for external device cabling.

When the hole has been done, you can reinstall the clock pc board and prepare the cable. It is a good idea to choose a cable with a conductor size that can fit in the extra header connections, beside the Pico. You need to put some solder on the cable tips so that they can be easily inserted in the header slots.

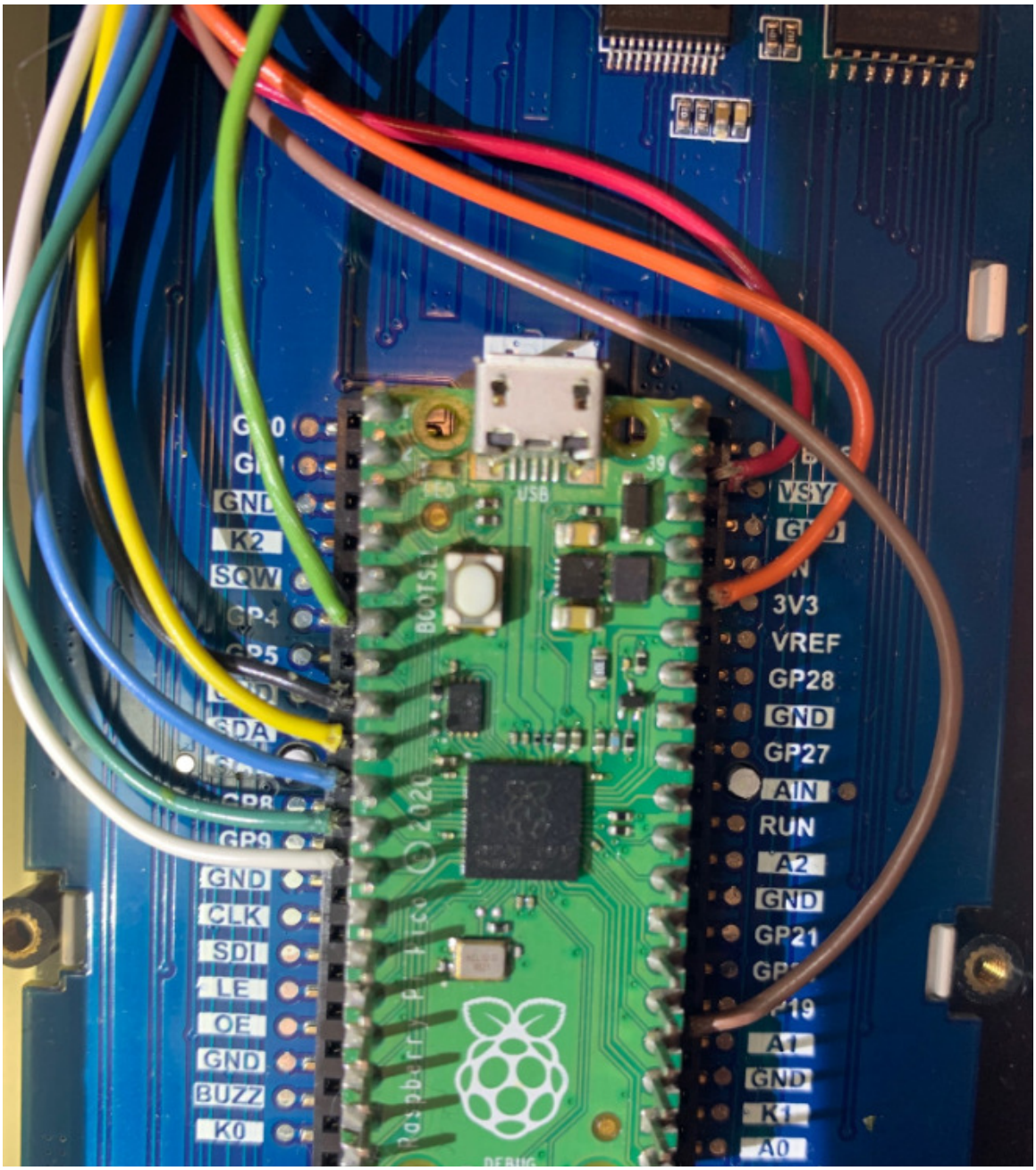
I added a tie wrap on the cable (inside the clock case) to prevent it to be bend outside of the case. Depending on the number of devices you want to add, the number of conductors may vary. I put an 8-wire cable since I already needed those eight, as shown in the table below.

The cabling goes like this:
**NOTE: Be careful… cabling has changed from previous version (Version 4.00)**

| Pico connection | Pico Pin number | Cable Color | Usage |
|---|---|---|---|
| Ground | 8 | Black | Ground for infrared receiver; passive buzzer, BME280, DHT22. |
| VSys + 5 volts | 39 | Red | Passive piezo through a 2N2222 transistor. |
| +3.3 volts | 36 | Orange | BME280, DHT22, Infrared receiver, RS232 interface. |
| GPIO4 | 6 | Stand alone | UART TX pin (external debugging display). |
| GPIO6 | 9 | Yellow | I2C SDA (data line) to BME280. |
| GPIO7 | 10 | Blue | I2C SCL (clock line) to BME280. |
| GPIO8 | 11 | Green | DHT22 |
| GPIO9 | 12 | White | Infrared receiver signal (VS1838b). |
| GPIO19 | 25 | Brown | Passive buzzer signal. |

Note: As you can see, there are 9 cables shown in the table. GPIO4 (Pico's pin 6 – see one of the green cable in the picture below) is a stand-alone temporary cable used to display debug information on an external monitor and is not part of the clock as a final product.

Note: Since the Pico power comes from the pins under the microcontroller (and not from the USB connector), VBus does not provide 5VCC. VSys needs to be used for an external 5 VCC source.




8-wire cable connected to the Pico double header

NOTE: Be careful: there are two green cables. The green cable in GPIO4 is a stand-alone cable (not part of the 8-conductor cable) that is used for RS232 transmission of data to a debug monitor. See Appendix G for more details.

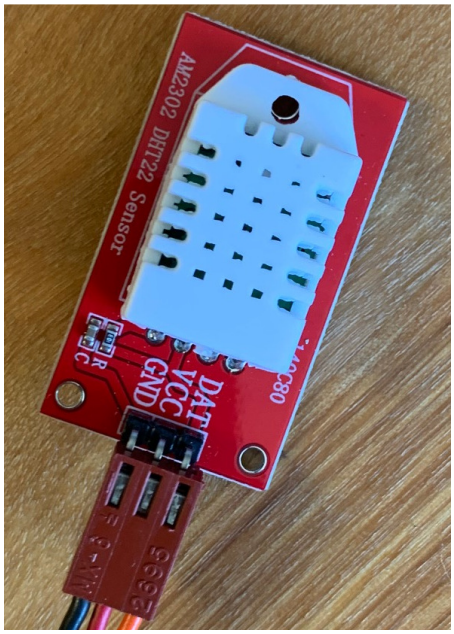
# Appendix C – Adding a DHT22 sensor

## *Special note about DHT22*

In the previous firmware version, a DHT22 device (outside temperature and humidity sensor) was supported by the firmware and also in this guide. When I developed the firmware, the DHT22 worked flawlessly when it was connected to the Green Clock with a 20cm cable. However, when I replaced the cable with another 3 meters long, I got from 75% to 80% errors in the communication between the Clock and the DHT22. Needless to say that such a rate is simply unusable. I tried to find related documentation on the Internet, but found nothing really useful, if not the fact that I was not the first to get similar problems. I tried the external pull-up resistor without success.

I did not remove the code supporting the DHT22 in the firmware. In fact, I continue to use the DHT22 for ambient temperature sensor since it gives me the relative humidity as extra information. However, I installed a BME280 as an outside temperature, relative humidity and barometric pressure sensor.

Starting with firmware version 3.00, support has been added for a DHT22 temperature and humidity sensor. Note that this sensor is not part of the Pico Green Clock and must be bought by user (more or less 10 US dollars).



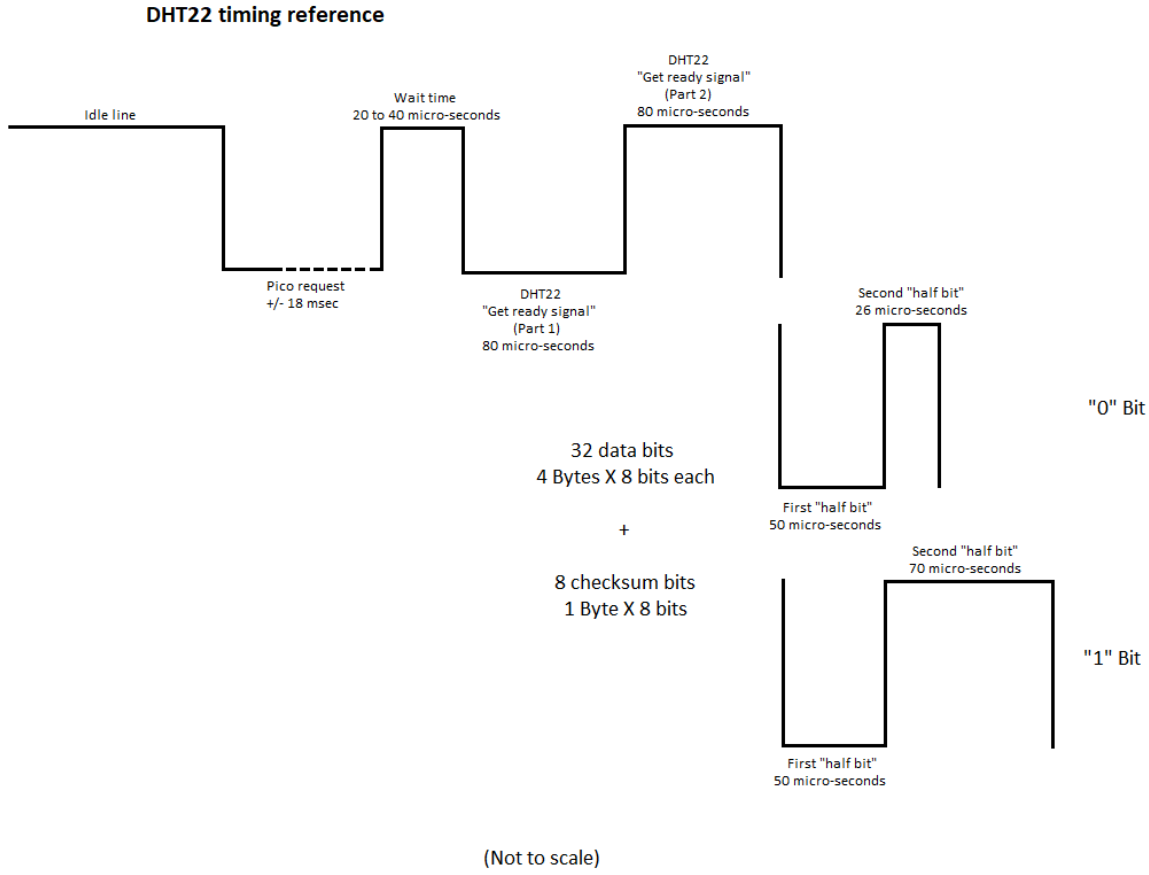

The conditional compile option “DHT_SUPPORT” must be defined in source code to be integrated into the firmware executable. In fact, if you plan to install a DHT22 in a near future, the conditional compile may include the code. I did build my own code to support the DHT22 and a few time-out’s have been integrated, so that the Pico clock will continue working without problem even if a DHT22 is not actually connected. Since the delay of the time-out’s represents more or less 20 milliseconds, and moreover, this delay occurs while the clock is scrolling the date, there is no impact for the user. (Just make sure that the cabling – if there is a cable already installed - is isolated from any outside static and / or electrical source).

DHT22 specifications may be easily found on the Internet. (A timing reference is given below). The timing is similar to the DHT11. However, even if the DHT22 is slightly more expensive, its temperature range is wider and is precision is better.

A three conductor cable is required for 3.3 volts, Ground, and data (I configured the DHT22 on GPIO 8).

When the conditional compile option is turned on in the code, the temperature and humidity will be scrolled on clock display every five minutes (5 minutes is a compile-time parameter), along with the date and other information. Take note that if there is an error in the DHT22 communication, temperature will simply not be part of the data scrolled (a "bad temperature" will not be shown on the clock display).

You must turn on the conditional compile "DHT_SUPPORT" in the source code for the DHT22 to be recognized by the firmware. Also, if the device is enabled in the code and you don't have a DHT22 installed, the clock will work normally without problem. The only impact will be errors reported in parenthesis during date scrolling. Read error from DHT22 are reported in parenthesis during date scrolling without any other indication (for example if you see (35/1537) in the scrolled text, it means that there has been 35 errors during a total of 1537 read cycles from the DHT22. Obviously, if no DHT22 is installed, the number of errors and the total number of read cycles will be the same. This information may be useful if you have communication interference (from a close motor or something else), since you can get an idea of the relative impact of this interference.



# Appendix D – Adding a BME280 sensor

## *Introduction*

A BME280 sensor can read the temperature, relative humidity and barometric pressure. As mentioned in Appendix C, DHT22 seems to have problem when cabling is more than a few inches long and in an actual installation with a cable of about 3 meters, I got 75 to 80% read errors from a DHT22.

I expect the BME280 to be a better candidate with a similar cable length, but at the time of this writing, I'm still using it beside the clock, with a cable about 1-meter long.

Be careful if you buy a BME280. It has been reported by a few Amazon customers that some sellers announce "BME280" devices, but they received a BMP280 instead of a BME280 (BMP280 does not give relative humidity like the BME280. Humidity may be useful for future usage like Humidex factor, etc…). I paid mine about 20US$ (for a single unit) from Waveshare since I wanted to make sure I would receive the right device.

Be aware that not all cables are used in the cabling since de module support either SPI or I2C communication. Cables have been provided for both protocols. For our purposes, only 4 cables are used: VCC (3.3 volts), GND, SDA and SCL are used for an I2C protocol connection.


BME280 module from Waveshare. You can see the device itself which is the little square metallic case in the middle left of the module (left picture).

## *BME280 integration*

The BME280 sensor supports SPI and I2C protocols. I used I2C for implementation in the Green Clock since this protocol is already used by the integrated real-time clock. Since I2C is kind of a “multi-drop” protocol, another device can be added on the same communication lines. It means that we do not use an extra GPIO of the Pico for this purpose.

So, 4 cables are required for the BME280 (see table in Appendix B):

- 3.3 Volts supply (from Pico pin 36)
- Ground (Pico pin 8 – or other ground source)
- SDA data line (GPIO6, Pico pin 9)
- SCL clock line (GPIO7, Pico pin 10)

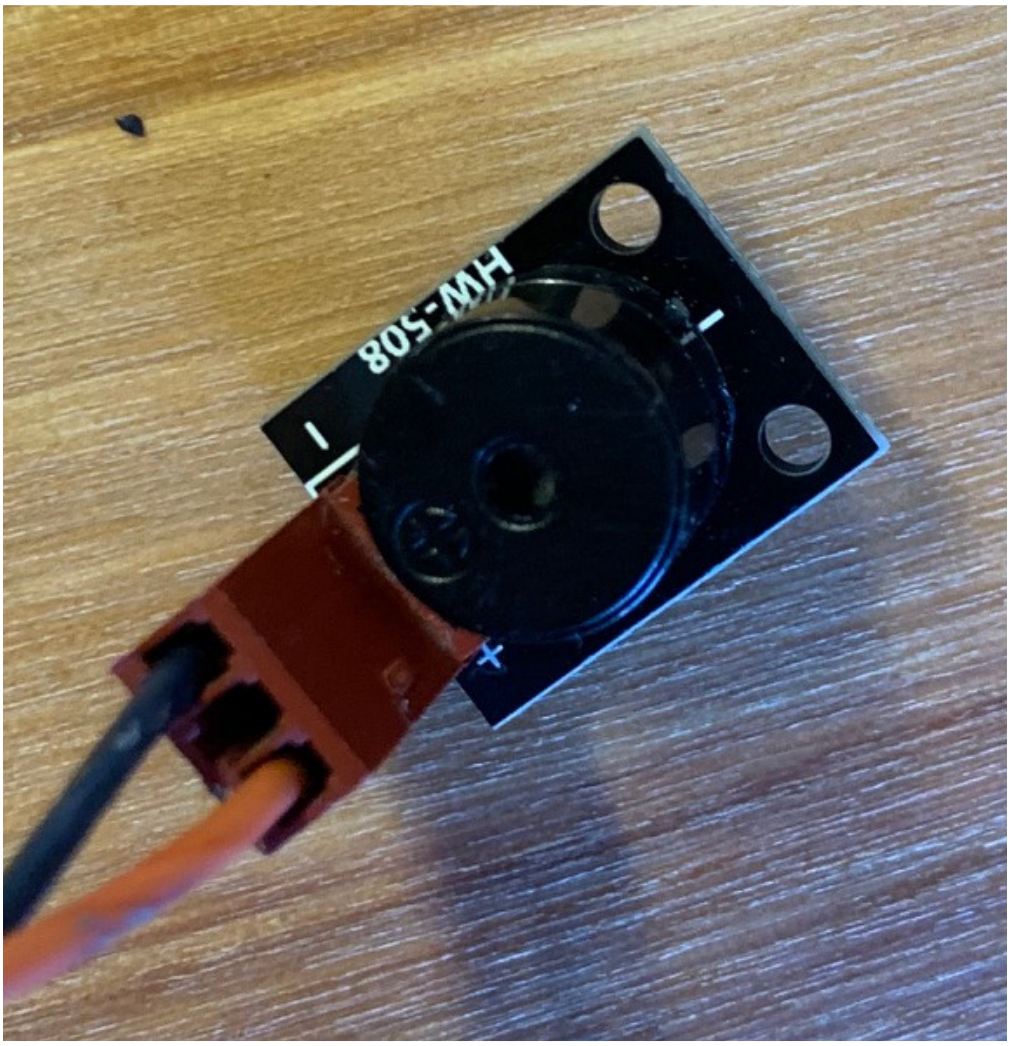
Be careful not to feed the BME280 with 5 volts. Even if the device can work without problem at 5 volts, the Pico logic level is 3.3 volts.

You must turn on the conditional compile “BME280_SUPPORT” in the source code for the BME280 to be recognized by the firmware. Also, if the device is enabled in the code and you don’t have a BME280 installed, the clock will work normally without problem. The only impact will be an error message in the clock power-up sequence (indicating that there has been an error while trying to communicate with the BME280) and the date scrolling will also report errors from BME280. Read error from BME280 are reported in square brackets during date scrolling without any other indication (for example if you see [5/2833] in the scrolled text, it means that there has been 5 errors during a total of 2833 read cycles from the BME280. Obviously, if no BME280 is installed, the number of errors and the total number of read cycles will be the same. This information may be useful if you have communication interference (from a close motor or something else), since you can get an idea of the relative impact of this interference.

# Appendix E – Adding a passive buzzer

## *Introduction*

The Pico Green clock provides an active buzzer. “Active” means that there is an oscillator integrated in the buzzer. This oscillator generates an audio frequency that is always the same, so it is not possible to play tunes or jingles. In version 2.00, I implemented algorithms that allow a specific duration and a “repeat count” for the basic “sound pack” and also another “repeat count” of the basic sound pack (see details in the source code). However, I thought it would be interesting to install a passive buzzer with firmware support to give us more flexibility with sounds. Since there are still some GPIOs available on the Pico, I installed a passive buzzer on GPIO 19.

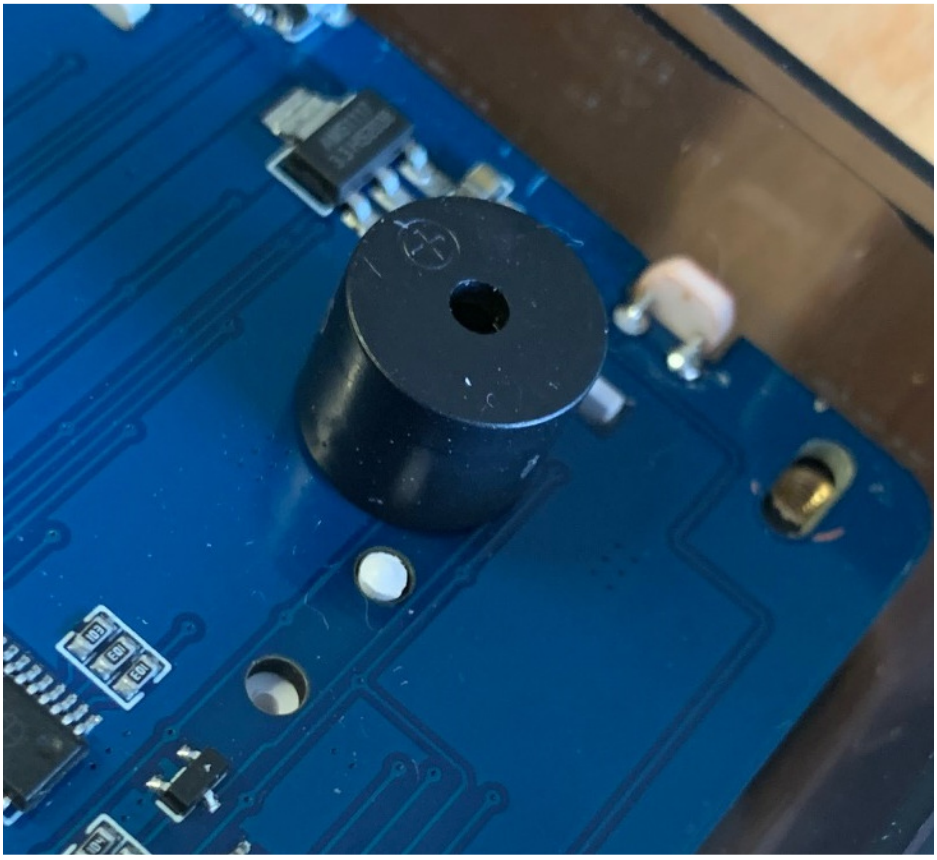



Passive buzzer to be added by user.

As you can see on the picture above, a passive buzzer looks very similar to an active buzzer. If you have a similar part and you want to identify if your buzzer is active or passive, there are two different ways:

1. Connect the buzzer directly to a 3.3 or 5 volts source. An active buzzer will make a sound whereas a passive buzzer will only make a “click”.

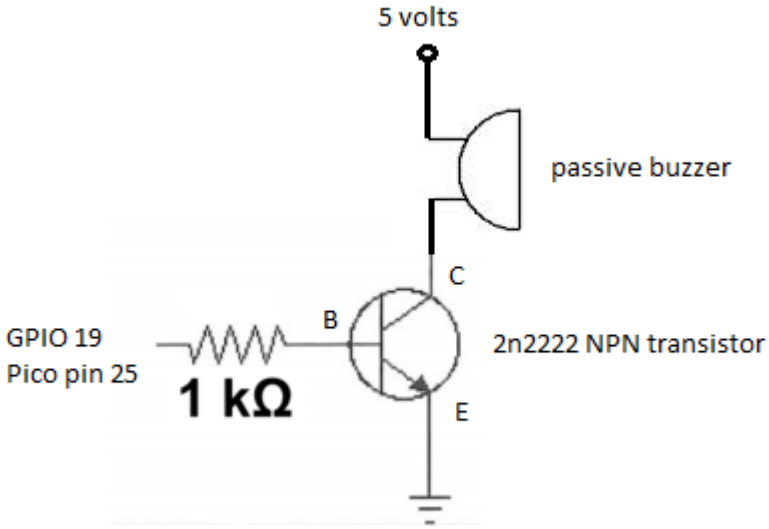
2. Check the buzzer with an ohmmeter. An active buzzer will typically have a resistance of several hundred ohms whereas a passive buzzer will have 8 or 16 ohms.



Active buzzer integrated in the clock (on Green Clock PC board).

As opposed to the other members of the Raspberry Pi family (Raspberry Pi 3, 4, etc…), the Pico is a micro-controller and does not run an operating system like Linux. That being said, the Pico nonetheless works with some form of multi-tasking. The callback functions, being based on timers or sound queue generate interrupts that must be serviced by the Pico. If we would decide to drive the passive buzzer simply by toggling the GPIO On and Off by software to generate the audio frequency, we would probably hear glitches here and there in the sound, when the Pico needs to service an interrupt in the middle of sound generation. For this reason, I used pulse width modulation (“PWM”) to generate sounds on the passive buzzer. This offloads the frequency generation from the Pico software execution to a specific hardware section (in the Pico).

Since the Pico logic level is 3.3 volts, it is useful to add a NPN transistor (2N2222 or the like) as a buffer to drive the passive buzzer with 5 volts and getting a louder sound.





2N2222 NPN Transistor

# Appendix F – Support of a remote control

## *Introduction*

I'm planning to install one of my Green Clocks in a relatively high location in my kitchen. I quickly realized that accessing the clock buttons would become a difficult task. So I thought it would be a good idea to add an infrared remote control to the clock…

The first step is to install an infrared receiver. The VS1838b IR receiver is not expensive and it takes care of filtering out the 38 kHz carrier of most remote controls.


VS1838b infrared receiver

# Pico Green Clock User Guide

I built an infrared protocol analyzer (based on a Pico micro-controller, of course!) and I looked at the commands sent by a Memorex remote control that I use for an "under-the-counter" radio / CD player.


I implemented a decoding function in the Green clock, so that I could act on each command received from the remote control.

The VS1838b infrared receiver must be installed outside the clock case (the colored plastic case would have significantly reduced the range of the remote if the receiver would have been installed inside the case). Like for the DHT22, a 3-wires cable is required for +3.3 volts, Ground, and GPIO 9.

All command decoding has been isolated in a separate include file called "memorex.cpp". This will allow implementation of other remote controls to be also isolated in separate modules without any impact on the main source code module.

When a valid command is received by the clock, a quick "beep" will sound (similar to a keyclick) and the command will be processed. If there is interference and / or if for any reason the command is not recognized by the clock, the infrared stream will simply be ignored and no "beep" will be heard.

## *Implementing another remote control*

Chances are that you won't have a Memorex remote control similar to the one I used. However, if you take a look at the decoding module ("memorex.cpp"), you will see that most parts of the code required to analyze the protocol for another remote control are available. You may want to display the timing values on the clock display for each remote button. The process will take some time but you should be able to come up with your own remote control decoding. Make sure to isolate the protocol related code in an include file similar / compliant to memorex.cpp.

A much more efficient way to analyze another remote control's protocol would be to display protocol information on a remote display instead of scrolling it on the clock display which rapidly becomes tedious since there is relatively lot of data to display. See Appendix G on how to configure an external monitor to display debug information.

## *How to use the remote control*

Once the protocol decoding function has been built and debugged for the Memorex remote control, next step has been to decide how to handle the commands sent from the remote (that is, how to assign a specific function to each remote button). Obviously, this remote control was originally designed to control a radio / CD player. So, the buttons on the remote often have no specific correspondence with the Green clock.

Whenever possible, I tried to implement a "logical" match between the remote control and a specific clock functionality. When not possible, I simply make a correspondence that must be followed without trying to see a logical link between both. Also, a few more functions / features are now possible and have been implemented thanks to the remote control support.

## *Remote control buttons*

For the text below, please refer to the remote control picture above. I tried to present and explain the functions associated with remote buttons in an order that will facilitate the comprehension and that will help the user to remember them. That's why they are not presented from top to bottom on the remote control picture.

### **Power**

The "Power" button on the remote control is not used at all with the clock. This way, as long as the original Memorex radio / CD player remains Off, I can play with all other buttons on the remote to interact with the clock without the risk of the radio becoming crazy by also responding to all button presses. I suggest that you do the same if you implement other remote controls.

If you press the "Power" button on the remote, the clock will beep (since it received a command that has been properly decoded), but will take no specific action.

### CD door

The “CD door” button is located at the top left of the remote, beside the “Power” button and relatively apart from other buttons. I found this button located in a very good “stand alone” position and I decided to assign it to a function that I will use quite often: “Display outside temperature, relative humidity and barometric pressure” (this assumes that you have also installed a BME280 sensor for reading outside weather parameters).

So if you press the “CD door” button, the clock will sound a quick beep to indicate that a valid command has been received, and the outside temperature, relative humidity and barometric pressure will begin scrolling on the clock display.

NOTE: As mentioned in Appendix D about BME280, at the end of text scrolling, you will see numbers in square brackets (like: “[0/2844]”). I added to the text the number of errors in the communication sessions with the BME280 on the total number of exchanges. In the example, there has been no error “0” and the clock requested 2844 times the weather parameters from BME280. This number “2844” depends on how long the clock has been up and running and what you set as the period for date and temperature scrolling. If you have a long cable or if there is a motor or other device near the cable, you may experience more errors on the long run.

### Over

While looking at the remote control, I found that the “Over” and “Mute” buttons were located in symmetrical locations and I needed such an arrangement for two functions that were not available on the clock itself but have been added with the support of the remote control.

So, a new function as been allocated to the “Over” button: “Display calendar events for today”. If there is one or more calendar event programmed for the current day, they will scroll on the clock display.

### Mute

Given the symmetric position of the “Mute” button with the “Over” button, the new function “Display calendar events for current week” has been implemented and allocated to the “Mute” button on the remote control.

If current day-of-the week is not Sunday, the clock will go backward until the nearest Sunday, and starting from that point, will display all calendar events programmed for the week (from Sunday, up to - and including - the next Saturday).

#### Set / memory / clock

**Replicate the "Set" (top) clock button quick press.**

There is a button identified "Set" (written on the button itself) which is also identified by "memory / clock" (written on the remote control body, this time).

It was easy to decide that this button would be used as an equivalent for the "Set" (top) button on the clock. In fact, this button simply replicates the "Set" button on the clock. It will allow the clock to enter clock setup mode if pressed while the clock is in the usual time display mode. It will then allow scanning all clock settings, the same way the "Set" (top) button does it on the clock.

#### Volume up / Volume down

**Replicate the "Up" (middle) and "Down" (bottom) buttons quick press.**

Just above the "Set / memory / clock" button, there is a "Volume up" / "Volume down" double button.

The same way it has been easy to decide on the function to assign to the "Set" button on the remote, the "Volume up" and "Volume down" buttons have been allocated to the same functions the "Up" (middle) and "Down" (bottom) buttons on the clock.

As we would expect, Volume Up and Volume Down will respectively increase and decrease values while we are in setup mode, in a way similar to what the "Up" and "Down" buttons do on the clock.

#### Stop

**Replicate "Down" (bottom) button long press.**

When you learned how to setup the clock, be it for clock settings, alarm settings or timer settings, you've seen that you can get out of setup mode by making a long press on the "Down" (bottom) button. So, I thought there should be a similar button on the remote control in order to get out of any setup mode… and I thought that the "Stop" button would be a good candidate for that purpose.

On top of being an "Exit from setup mode", the "Stop" button will display the average ambient light for the last 60 seconds if the clock is not currently in one of the setup modes. This function has no equivalent on the clock. Theoretically, ambient light values go from 0 to 4096 (12 bits AD converter). However, it is unlikely that you will ever see a

value in those extremes. With a flood light directly on the photocell, the value could be around 4040, whereas when very dark, the value could be around 140.

NOTE: I reversed the values originally given by the function to get ambient light. I thought it would be more intuitive to have a higher value when the light is at a higher level rather than the other way around.

### Repeat / Up

**Replicate "Set" (top) button long press.**

As we've seen earlier, the "Set" button on the remote control has been configured to replicate the "Set" (top) button on the clock.

I thought it could be an easy way to remember the alarm setup button if it is located just beside the clock setup button.

So, pressing the "Repeat / Up" button on the remote control will enter the alarm setup mode. We then use the "Volume up" / "Volume down" buttons to change settings, the "Set" remote button to scan through the different alarm settings and / or the "Stop" remote button to get out of the alarm setting mode (in a way similar to the buttons used on the clock itself).

### Random / Down

**Replicate "Up" (middle) button long press.**

I followed the same logic with the "Random / Down" remote button: The "Set" remote button enters the clock setup, the "Repeat / Up" enters alarm setup, so, the "Random / Down" (located just beside "Repeat / Up") enters the timer setup mode (all three setup buttons are placed on the same remote control row).

Once in the timer setup mode, we use the "Volume up" / "Volume down" to change current setting, the "Set" remote button to scan through the different settings, and the "Stop" button to exit timer setup.

### Band / Tuner

A new function has been allocated to the "Band / Tuner" button. This button is located in the bottom left of the remote control and is easily reached with the thumb, even without looking at the remote. So, when pressing this button, the clock display will roll two dices

and display two random values between 1 and 6. No more need to roll dices with hands! The clock also makes some pixel twinkling animation and sound bubbling while randomizing the numbers.

### CD

Another new function has been added, thanks to the remote control.

The "CD" button on the remote control could be interpreted as "Count Down" (of course, I know that originally, "CD" did not mean count-down on the remote… it's just a way to better remember). We've seen before that the "Random / Down" button has been already configured for the count-down timer. This time, however, we're not talking about the Count-down timer. The CD remote button will make the clock silent for 30 minutes each time the button is pressed.

If you have some visitors at home and there are seven calendar events programmed for this specific date, you may get tired of all sounds coming from the clock (calendar events, hourly chime, etc)… The "CD" remote control button will allow you to add 30 minutes of silence for each press. This silence period will have an effect on hourly chime and calendar events (not for other type of sounds, like alarms, timer count-down, etc…)

Also, if you pressed the button twice a while ago (for a one-hour silence period) and the "silence count-down" is down to 5 minutes, pressing the CD button once more will grow the silence period to 30 + 5 = 35 minutes (adding 30 minutes to the remaining 5 minutes).

If you want to cancel the silence period, use the "Play / Pause" button (see below).

### Time

A new function has been allocated to the "Time" button. When pressing this button on the remote control, the clock display will show for a few seconds the current minute and current second (instead of current hour and current minutes – the seconds will change a few times while the display remains in this function before returning to normal time display mode). While the clock is in this mode, the two middle-dots will stop blinking to indicate that you are in "seconds display mode". Instead of the middle-dots blinking, you will see the seconds changing on the display.

After five seconds, the display will return to the usual time display mode.

### Rewind / Down

The remote button with a “Rewind” symbol on it (also identified with “Dn” written on the remote body) is not used for now.

**Fast forward / Up**

The remote button with a “Fast forward” symbol on it (also identified with “Up” written on the remote body) is not used for now.

**Numeral buttons (1 to 0)**

So far, we’ve not discussed about the remote numeral buttons 1 to 0… Before going forward, let’s go back to the clock setup for a moment…

As we’ve seen previously, to proceed with the clock setup, be it directly with the clock buttons and / or with the remote control buttons that replicate the same clock functions, the procedure is straightforward: we press the “Set” button to enter setup mode, and we then press the “Set” button each time we want to proceed with the next setting in the list.

If you refer to the list of all clock settings in the corresponding section in this guide, you can see that there are a lot of settings in the list. If we want to modify a parameter that is at the end of the list, we have to go through all other parameters until we reach the one of interest. To make this process easier, the numeral buttons have been used.

NOTE: Since button 0 is physically located at the end of the numeral buttons on the remote control, it must be considered the last button (that is, 0 comes *after 9* and not *before 1* in the procedure logic).

Once we are in clock setup mode, we can use the “Set” button to scan through the many settings as we’ve already seen, but we can also use the numeral buttons on the remote control to jump directly to a specific setup section. Here is a quick reference of the sections associated with each numeral:

| Remote control button | Setup section |
|---|---|
| 1 | Time / Date / Year |
| 2 | Daylight saving time |
| 3 | Keyclick |
| 4 | Display scroll |
| 5 | Temperature unit |
| 6 | Language |
| 7 | Time display format |
| 8 | Hourly chime |
| 9 | Night light |

| | |
|---|---|
| 0 | Auto brightness |

For example, once you pressed the “Set” button, the hour will blink since you are in clock setup mode, hour setting. Instead of pressing “Up” or “Down” to change hour setting, you can press the button “7” if you want to jump directly to the “Time display format” setting.

Once in a specific setting section, you can still use the “Set” button to scan the next settings in sequence. As another example, if you press the button “8”, you will be in the hourly chime setting. You can use “Volume up” / “Volume down” buttons to change the setting. Now, if you want to change Hourly time on and Hourly time off (which are the next settings in the regular sequence), you can press the “Set” button to go to the next steps in sequence.

You can go back and forth in the setup section of interest. For example, you’re allowed to return to Time setting (by pressing button “1”) after setting up hourly chime.

**Display**

There is a reason why I presented the numeral buttons in the previous section, before the discussion about the “Display” button.

The “Display” remote control button will enter a “generic display mode”. This can be observed by the clock display stopping the middle dots blinking.

Once in generic display mode, the selection of a specific numeral button will display information about the section selected (See the sections in the table for “Settings” in the previous section above. They also apply for Display).

For example, pressing “Display”, followed by button “4” (“Display scroll” as indicated in the table in the previous section) will scroll on clock display a lot of information about Display scroll (Status On / Off, Time period for scrolling, period of each dot-scroll, etc…)

As opposed to the “Set” button, however, you can select only one display function at a time. If you want to display information about Display scrolling and also about Hourly chime, you have to press first “Display”, then “4” and next “Display” again, then 8.

**Play / Pause**

Finally, the last button that has not been discussed is the “Play / Pause” button. This is the button to remember when your friend hacker comes home. It performs many clock functions / demos and displays a lot of information (Matrix test, Indicators test, White

LEDs test, Pixels twinkling, Active buzzer tests, Passive buzzer tests, Scroll Pico internal temperature, Scroll ambient temperature, Scroll outside temperature (given that you installed a BME280 sensor), Scroll power supply voltage, Scroll ambient light, etc…

As a "side effect", pressing the "Play / Pause" will cancel any silence period currently on-going on the clock. If you still want a silence period after pressing "Play / Pause", simply press the button "CD" (to increment the silence periode by 30 minutes each time) when the clock demo is over.

# Appendix G – Configuring an external monitor

## *Introduction*

If you work on the source code of the Green Clock for significant changes, you will quickly realize that scrolling debug information on the clock display is a tedious task. Not much for the work required in the source code (many functions have been implemented to make the task easy), but for the time it takes to read the information scrolled on the display and making use of it.

This Appendix will show you how to make a connection with an external monitor to display debug information. Of course, such an external monitor with a 80 X 24 screen (or more) is much easier to work (and faster to read) for debugging purposes than looking at a display scrolling with 4 or 5 characters at a time.

## *VT-101 monitor*

If you're much younger than I am, you may have never heard about "VT-101". Back in the old days, when computers were costing hundreds of thousands of dollars, one big central computer was installed (called a "mainframe") and many users were connected to it using a "dumb terminal". This terminal was used only to receive plain data from the computer and send plain data to the computer using a special serial protocol called RS232. In fact, many different model numbers existed, but VT-101 originated from Digital Equipment Corporation and was very popular. If you have such a monitor, you can use it to display debug information.

1) We can program the Raspberry Pi Pico so that specific GPIOs are configured as UART ("Universal Asynchronous Receiver Transmitter"), the IC used to implement serial protocol. You can refer to the source code to see how to properly program the GPIO (make a search for the string "uart1"). You can also take a look

at the function “uart_send()”. The details about serial protocol are beyond the scope of this user guide.

Note: uart0 can also be used instead of uart1 and should allow using “printf” for an easier programming interface.

2) Now that we know how to configure the Pico for serial communication, we’re half done ! In fact, even if the Pico is now able to receive and / or transmit data from / to a serial line, we still have to adjust the voltage levels. The Pico works with 3.3 volts, whereas RS232 works with higher voltage (there has been many protocol revisions over the years, but the signal is usually swapping from -5 volts to + 5 volts. So, we need an RS232 adaptor to convert RS232 levels down to 3.3 volts compatible with the Pico.

   I bought such an interface from Waveshare:

   https://www.waveshare.com/rs232-board.htm


3) You can make things simple by using VCC (3.3 volts), ground and TxIn (which transmit TO the external VT-101 monitor). You should not need to worry about CTS and RTS handshake. Also, as far as firmware Version 5.00 is concerned, there is no need to use the RxOut pin of the interface that could be used to receive characters sent from the VT-101 keyboard to the Pico (except if you implement new functions on the clock that could use it).

   Then, you simply connect the interface module to your VT-101 monitor using a cable with DB9 on both ends (or a DB9 and a DB25).

Be aware that the protocol of the monitor must match the one of the Pico, namely for the speed and the handshake (in Version 5.00 firmware, speed is set at 38400 and no handshake is used). Refer to the monitor's user guide how to set it up.

There is not much to say about sending information to the external monitor. Keep in mind that 38400 bauds (baud is the speed unit for RS232 serial communication) is relatively slow by today's computer standards. If you send to much information from callback or interrupt routines, you may have side effects on the timings that will bring you on more problems instead of solutions !.

## *Computer monitor over RS232*

If you don't have a VT-101 monitor (or another similar product), you can use an old PC to display the debug information.

The PCs sold in the last few years don't have a serial port for RS232 communication. If you have an old PC' however, it may be a good opportunity to give it another life.

1) All you have to do is following the instructions in the section above ("VT-101 type monitor"), but instead of connecting the interface to a VT-101, connect it to the computer DB9 or DB25 that is should have if it was equipped with a serial port.

2) You also need to run a terminal emulator program on the computer. "Procomm" was a popular communication program that could run on DOS. There was also a version to run under Windows. HyperTerminal was also a terminal emulator program included in the first versions of Windows.

   Don't forget that you still have to configure the serial protocol on the communication program as you have to do on the VT-101.

3) Another advantage of using a PC is that most of the communication programs have an option to save all communications in a log file. So, you can save all information sent by the Pico to a file and then take the time to analyze it and modify your program accordingly.

## *Computer monitor over USB*

If you don't have an old computer with a serial port, you still have another solution. It is to use a modern computer to receive information from the Pico via the USB port.

1) It is important to understand that the Pico will still send the information from its internal uart and through an RS232 interface. So, you still have to perform the steps as described above in the section “VT-101 monitor”.

2) Since the Pico sends its data using an RS232 interface and you want to receive it using a USB port, you will need a RS232-to-USB adaptor. Make sure the adaptor you buy provides the device driver required by your OS (correct version of Mac, Windows, Linux, or other). Here are some examples of such an adaptor:


(Note: prices shown above are in Canadian $)

Some adaptors provide an DB9-to-DB25 adaptor. This may be useful if your Raspberry Pi RS232 adaptor has a DB25 instead of a DB9 (a DB9 is shown at point 1 above).

3) Follow the instructions that come with the adaptor to install the device driver. My understanding is that there is a chipset integrated inside one of the connectors (either USB or DB9). Consequently, you may need to plug the adapter before installing the driver. You may also need to reboot the system for the adaptor to work properly. (Note: the chipset is self-powered through USB connection).

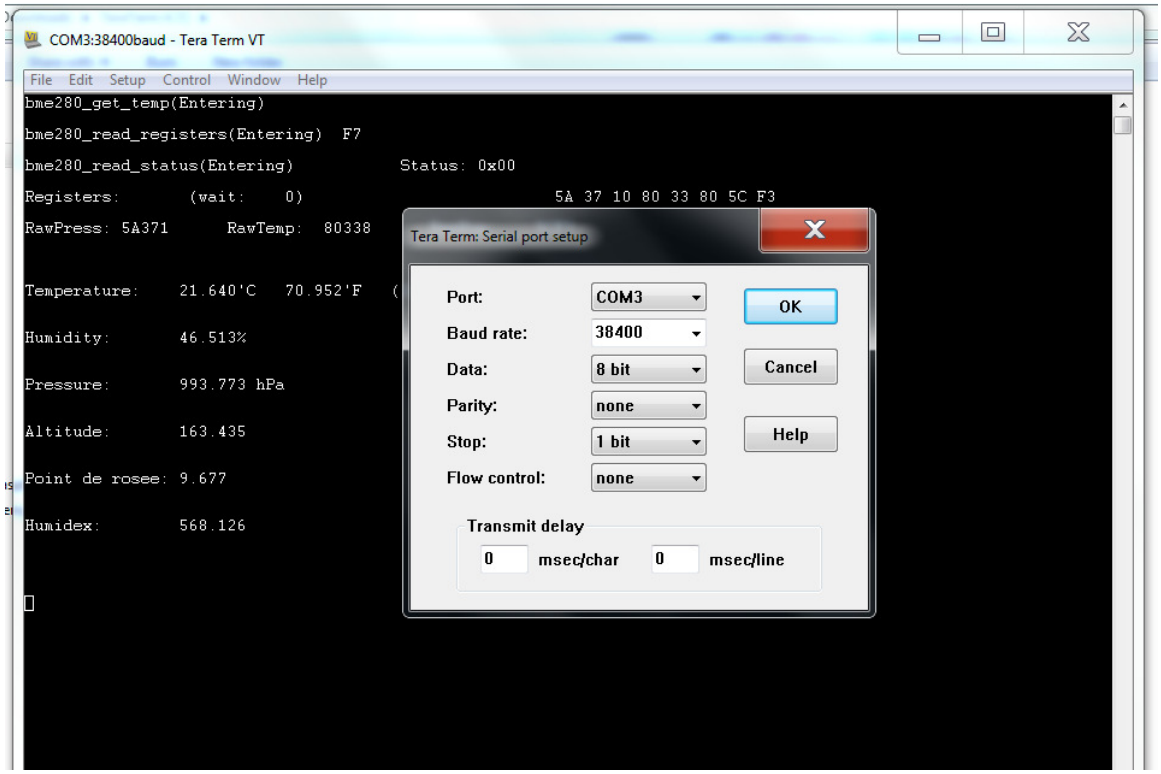
4) The final step is to find a terminal emulator program that will work on your computer. I used TeraTerm, Freeware and Open source. As its author wrote, "it is not a full-fledge terminal emulator", but I found it to be perfect to display debug information sent by a Raspberry Pi Pico.

https://download.cnet.com/Tera-Term/3000-2094_4-75766675.html

5) Start the terminal emulator and go to the "File / New connection" menu, you will see that there is a "Serial" option. For some reason, the device driver of the RS232-to-USB adaptor seems to force a specific name to your serial adaptor. In my case, it was COM3: This seems to be related to the electronics behind the USB port. My deduction is that the first USB port would be named COM1:, the second one COM2: etc… But this is only my guess. Whatever the case, the naming convention should not be a problem.

6) Then, going to the menu "Setup / Serial port", you can configure the protocol as you want (usual setting is "N, 8, 1", but as long as you set the Raspberry Pi the

same way, it will work). As for the speed, I configured 38400 which is relatively fast and will not slow down the Pico too much.




NOTE: Once in a while, it may happen that the terminal emulator will stop display the data coming in. In this case, you may need to close the terminal application and restart it, which is a quick operation. Make sure you have previously saved the selected configuration “Setup / Save setup” so that you don’t need to reconfigure every time. This does not happen frequently (once every two days if you spend many hours a day in front of your screen). If a restart doesn’t work, unplugged the RS232-to-USB adaptor (at the USB end) and plug it back, then stop and restart the terminal program.

You will see that being able to display debug information on a PC screen is really a plus when working on a microcontroller. However, when working with real-time device, interrupt service routines or callback functions, or some other timing-specific stuff, we must be careful on the load we put on the system by sending data through a serial port…

# Appendix H - Firmware

This section gives general information about different subjects related to the source code and will be of interest to those who want to work on the code. It is not intended to explain the code in details, but rather to focus on some specific features of interest. The information is given below, in alphabetic order.

The author of this document is interested in receiving your comments and ideas about features that you have added to the Pico Green Clock (email address given after Index at the beginning of this document).

## *Alarm numbers*

It has been seen in the user guide that two different alarm numbers are available. They have been numbered 1 and 2 to be more "natural" or "human-like" for the clock user. However, in the source code, these alarms are numbered 0 and 1.

## *Alpha characters*

Even if a 5 X 7 character set has been implemented, 4 X 7 character set is still needed. Given the size of the clock display, 5 X 7 characters are too large to be used to display time on the display. So, the 4 X 7 character has been left in the code, and moreover, I tried to add the whole displayable ASCII characters. Result is not always good, but in some situation and / or for some characters, it proved to be useful.

## *Calendar events*

Calendar events are compile time options that may be configured before rebuilding the firmware. Basically, a "calendar event" is made of four elements:

1) A day-of-month (from 1 to 31).
2) A month (from 1 to 12).
3) An optional "jingle" identification.

4) A text (maximum of 40 characters).

When the specific date is reached (say, 15th of August), the text that has been defined in the source code will be scrolled on the clock display during 24 hours (the 15th of August in our example, from 00h00 up to 23h59), twice an hour, at xxh14 and xxh44. Time has been specifically chosen so that it will not interfere with the hourly chime (every xxh00) and / or date display which is by default every 5 minutes (xxh00, xxh05, xxh10, xxh15, etc…).

If an optional passive buzzer has been installed by user and if a specific jingle identification is specified, it will be played when the calendar event is scrolled on clock display. As already mentioned, both passive and active buzzers will sound in accordance with the hourly chime settings.

Up to 50 calendar events may be configured. It must be noted that no validation is done on the day-of-month and month configured in the source code. If invalid values are encoded, the corresponding string will simply never be scrolled.

A special case when the text is “Debug” will trigger a special handling in the “process_scroll()” function. This is very useful when we want to display run-time information, even while inside an ISR.

## *Character bitmap*

Original software Version 1.00 from Waveshare proposed the 4 X 7 character set (a few characters were based on a 5 X 7 bitmap however). It may be observed that the 4 X 7 character bitmap itself is not implemented in the intuitive way we would expect. This is because the lowest bit is on the left side of the LED matrix, while the lowest bit is on the right side of the byte when we manipulate the bit positions.

In order to build on a more intuitive bitmap character set, the 5 X 7 character bitmap uses the function “reverse_bits()” allowing the character bitmap to be based the way we expect it to be. (I didn’t change the way the 4 X 7 character set was handled, tough).

## *Clock “Option section”*

Many options have been grouped at the beginning of the source code. These are the options most likely to be adjusted / changed to user’s taste. The programmer may want to consult this section before modifying elsewhere in the code, and may also want to add any new feature that could be tuned to user’s taste in this section.

## *Coding standard*

Many function names have been changed from the original (Version 1.00) to better represent what they do. Also, even if most modern integrated development environment propose tools to easily navigate through the code, all functions have been sorted in alphabetical order (except “main()”, which is first). There is no “one good standard”, but the one I use has been deployed throughout the code: function name all in lowercase with an underscore to isolate tokens, variable names capitalized, etc…

Variable names have also usually been sorted in alphabetical order in the code.

## *Debug chunks of code*

Many chunks of code that have been used for debugging purposes have been left in the code. I usually enclosed them between comments symbols /*** some code ***/ or “/// some code” to easily recognize them as “debug code”. Also, before making an official release, I always scan all “***” and “///” to clean the code of extra and useless pieces of code. However, since the Pico Green Clock is something that more programmers may want to work on, I left many such chunks of code in the sources.

Since they are commented out, they do not take extra space in the executable.

## *Function scroll_queue_value()*

This function has been implemented to make debugging easier. It can be used even inside ISR to display markers and / or other useful information.

Look for examples in the code.

## *Function scroll_string()*

The “scroll_string()” function is very helpful when implementing and / or debugging sections of the code. It allows displaying status, variables or other useful information on clock display at run-time for those without other debugging tool (serial monitor, USB display, etc…)

## *Options*

As mentioned in the previous sections of this guide, I added four options to the clock so far:

1) A BME280 sensor (for outside temperature, relative humidity and barometric pressure).
2) A DHT22 sensor (for ambient temperature and relative humidity).
3) A passive buzzer (to play jingles).
4) An infrared receiver (to receive remote control commands).

Even if most sections of code related to these options have been made conditional compile, they could remain active all the time without problem.

The BME280 code will go through without significant impact on the timing if no BME280 has been installed. Number of errors and total read cycles [inside square brackets] will continue to be scrolled with date scrolling, tough.

The DHT22 code has time-outs that will make the code behavior without real change even if no DHT22 is installed. The only visual impact will be the error count (appearing in parenthesis) with the date scrolling.

The passive buzzer is feed with a scroll queue working inside an interrupt service routine, so there will be no real impact if there is no passive buzzer connected on the target GPIO.

Finally, if there is no infrared receiver connected, the routine waiting to receive IR commands will simply stand there without receiving anything. Once more, there will be no real impact on clock behavior.

## *Remote control*

A file called “memorex.cpp” contains all infrared commands related processing. If another brand of remote control is implemented, it would be a good idea to create another include file similar to “memorex.cpp”, so that only the include file name has to be changed in the main source file to support this other remote control

## *Test section*

To keep the firmware as small as possible, a conditional compile option has been configured so that test code may or may not be included in the executable. (see "#ifdef TEST_MODE").

Many chunks of test code have been left in the source code to help any programmer adding new features to the clock or proceed with more tests.

Those chunks of test code must be considered as such: test code! It should help you with the implementation of new functions / features, but it must NOT be considered as "debugged" and "fool-proof" code! Use it at your own risk and effort!

# Appendix I – GPIO used in Waveshare Green clock

List of GPIOs used in the Waveshare Green Clock:

| GPIO number | Direction / Usage | Description |
|---|---|---|
| GPIO 0 | | Not used |
| GPIO 1 | | Not used |
| GPIO 2 | (In) | "Set" (Top) button |
| GPIO 3 | (In) | SQW (DS3231 RTC IC) |
| GPIO 4 | (Out) | Pico's UART output to an external VT-101 type monitor |
| GPIO 5 | (In) | Pico's UART input from an external VT-101 type monitor |
| GPIO 6 | (I2C) SDA | Temperature reading from DS3231 / Data reading from BME280 |
| GPIO 7 | (I2C) SCL | Temperature reading from DS3231 / Data reading from BME280 |
| GPIO 8 | (In / Out) | DHT22 (must be added by user) |
| GPIO 9 | (In) | Infrared receiver for remote control (must be added by user) |
| GPIO 10 | (Out) | CLK (LED matrix controller) |
| GPIO 11 | (Out) | SDI (LED matrix controller) |
| GPIO 12 | (Out) | LE ("Latch Enable") for LED matrix controller |
| GPIO 13 | (Out) | OE ("Output Enable") for LED matrix controller |
| GPIO 14 | (Out) | Active buzzer / Piezo (integrated on Green clock pc board) |
| GPIO 15 | (In) | "Down" (Bottom) button |
| GPIO 16 | (Out) | A0 (address line for LED matrix controller) |
| GPIO 17 | (In) | "Up" (Middle) button |
| GPIO 18 | (Out) | A1 (address line for LED matrix controller) |
| GPIO 19 | (Out) | Passive buzzer signal (must be added by user). |
| GPIO 20 | | Not used |
| GPIO 21 | | Not used |
| GPIO 22 | (Out) | A2 (address line for LED matrix controller) |
| GPIO 23 | | Used internally for voltage regulation |
| GPIO 24 | | Used internally for voltage detection |
| GPIO 25 | (Out) | On-board Pico`s LED |
| GPIO 26 | | ADC0 (Ambient light reading) |
| GPIO 27 | | Not used |
| GPIO 28 | | Not used |
| GPIO 29 | | ADC-Vref (Power supply voltage reading) |

# Appendix J – Framebuffer bitmap configuration

Framebuffer[] bitmap configuration

Framebuffer[0]:
Bit 0 = "Move On" indicator - left LED
Bit 1 = "Move On" indicator - right LED
Bit 2 = White LED - top
Bit 3 = "Monday" indicator - left LED
Bit 4 = "Monday" indicator - right LED
Bit 5 = White LED - bottom
Bit 6 = "Tuesday" indicator - left LED
Bit 7 = "Tuesday" indicator - right LED

Framebuffer[1]:
Bit 0 = "Alarm On" indicator - left LED
Bit 1 = "Alarm On" indicator - right LED
Bit 2 = Display matrix 1,1
Bit 3 = Display matrix 1,2
Bit 4 = Display matrix 1,3
Bit 5 = Display matrix 1,4
Bit 6 = Display matrix 1,5
Bit 7 = Display matrix 1,6

Framebuffer[2]:
Bit 0 = "Count Down" indicator - left LED
Bit 1 = "Count Down" indicator - right LED
Bit 2 = Display matrix 2,1
Bit 3 = Display matrix 2,2
Bit 4 = Display matrix 2,3
Bit 5 = Display matrix 2,4
Bit 6 = Display matrix 2,5
Bit 7 = Display matrix 2,6

Framebuffer[3]:
Bit 0 = "Fahrenheit" indicator
Bit 1 = "Celsius" indicator
Bit 2 = Display matrix 3,1
Bit 3 = Display matrix 3,2
Bit 4 = Display matrix 3,3
Bit 5 = Display matrix 3,4
Bit 6 = Display matrix 3,5
Bit 7 = Display matrix 3,6

Framebuffer[4]:
Bit 0 = "AM" indicator
Bit 1 = "PM" indicator
Bit 2 = Display matrix 4,1
Bit 3 = Display matrix 4,2
Bit 4 = Display matrix 4,3
Bit 5 = Display matrix 4,4
Bit 6 = Display matrix 4,5
Bit 7 = Display matrix 4,6

Framebuffer[5]:
Bit 0 = "Count Up" indicator - left LED
Bit 1 = "Count Up" indicator - right LED
Bit 2 = Display matrix 5,1
Bit 3 = Display matrix 5,2
Bit 4 = Display matrix 5,3
Bit 5 = Display matrix 5,4
Bit 6 = Display matrix 5,5
Bit 7 = Display matrix 5,6

Framebuffer[6]:
Bit 0 = "Hour chime" indicator - left LED
Bit 1 = "Hour chime" indicator - right LED
Bit 2 = Display matrix 6,1
Bit 3 = Display matrix 6,2
Bit 4 = Display matrix 6,3
Bit 5 = Display matrix 6,4
Bit 6 = Display matrix 6,5
Bit 7 = Display matrix 6,6

Framebuffer[7]:
Bit 0 = "Auto Light" indicator - left LED
Bit 1 = "Auto Light" indicator - right LED
Bit 2 = Display matrix 7,1
Bit 3 = Display matrix 7,2
Bit 4 = Display matrix 7,3
Bit 5 = Display matrix 7,4
Bit 6 = Display matrix 7,5
Bit 7 = Display matrix 7,6

Framebuffer[8]:
Bit 0 = Not used
Bit 1 = "Wednesday" indicator - left LED
Bit 2 = "Wednesday" indicator - right LED

Framebuffer[9]:
Bit 0 = Display matrix 1,7
Bit 1 = Display matrix 1,8
Bit 2 = Display matrix 1,9

Bit 3 = Not used
Bit 4 = "Thursday" indicator - left LED
Bit 5 = "Thursday" indicator - right LED
Bit 6 = Not used
Bit 7 = "Friday" indicator - left LED

Framebuffer[10]:
Bit 0 = Display matrix 2,7
Bit 1 = Display matrix 2,8
Bit 2 = Display matrix 2,9
Bit 3 = Display matrix 2,10
Bit 4 = Display matrix 2,11
Bit 5 = Display matrix 2,12
Bit 6 = Display matrix 2,13
Bit 7 = Display matrix 2,14

Framebuffer[11]:
Bit 0 = Display matrix 3,7
Bit 1 = Display matrix 3,8
Bit 2 = Display matrix 3,9
Bit 3 = Display matrix 3,10
Bit 4 = Display matrix 3,11
Bit 5 = Display matrix 3,12
Bit 6 = Display matrix 3,13
Bit 7 = Display matrix 3,14

Framebuffer[12]:
Bit 0 = Display matrix 4,7
Bit 1 = Display matrix 4,8
Bit 2 = Display matrix 4,9
Bit 3 = Display matrix 4,10
Bit 4 = Display matrix 4,11
Bit 5 = Display matrix 4,12
Bit 6 = Display matrix 4,13
Bit 7 = Display matrix 4,14

Framebuffer[13]:
Bit 0 = Display matrix 5,7
Bit 1 = Display matrix 5,8
Bit 2 = Display matrix 5,9
Bit 3 = Display matrix 5,10
Bit 4 = Display matrix 5,11
Bit 5 = Display matrix 5,12
Bit 6 = Display matrix 5,13
Bit 7 = Display matrix 5,14

Framebuffer[14]:
Bit 0 = Display matrix 6,7
Bit 1 = Display matrix 6,8
Bit 2 = Display matrix 6,9
Bit 3 = Display matrix 6,10
Bit 4 = Display matrix 6,11
Bit 5 = Display matrix 6,12
Bit 6 = Display matrix 6,13
Bit 7 = Display matrix 6,14

Framebuffer[15]:
Bit 0 = Display matrix 7,7
Bit 1 = Display matrix 7,8
Bit 2 = Display matrix 7,9
Bit 3 = Display matrix 7,10
Bit 4 = Display matrix 7,11
Bit 5 = Display matrix 7,12
Bit 6 = Display matrix 7,13
Bit 7 = Display matrix 7,14

Framebuffer[16]:
Bit 0 = "Friday" indicator - right LED
Bit 1 = Not used
Bit 2 = "Saturday" indicator - left LED
Bit 3 = "Saturday" indicator - right LED
Bit 4 = Not used
Bit 5 = "Sunday" indicator - left LED
Bit 6 = "Sunday" indicator - right LED
Bit 7 = Not used

Framebuffer[17]:
Bit 0 = Display matrix 1,15
Bit 1 = Display matrix 1,16
Bit 2 = Display matrix 1,17
Bit 3 = Display matrix 1,18
Bit 4 = Display matrix 1,19
Bit 5 = Display matrix 1,20
Bit 6 = Display matrix 1,21
Bit 7 = Display matrix 1,22

Framebuffer[18]:
Bit 0 = Display matrix 2,15
Bit 1 = Display matrix 2,16
Bit 2 = Display matrix 2,17
Bit 3 = Display matrix 2,18

Framebuffer[19]:
Bit 0 = Display matrix 3,15
Bit 1 = Display matrix 3,16
Bit 2 = Display matrix 3,17
Bit 3 = Display matrix 3,18

Bit 4 = Display matrix 2,19
Bit 5 = Display matrix 2,20
Bit 6 = Display matrix 2,21
Bit 7 = Display matrix 2,22

Bit 4 = Display matrix 3,19
Bit 5 = Display matrix 3,20
Bit 6 = Display matrix 3,21
Bit 7 = Display matrix 3,22

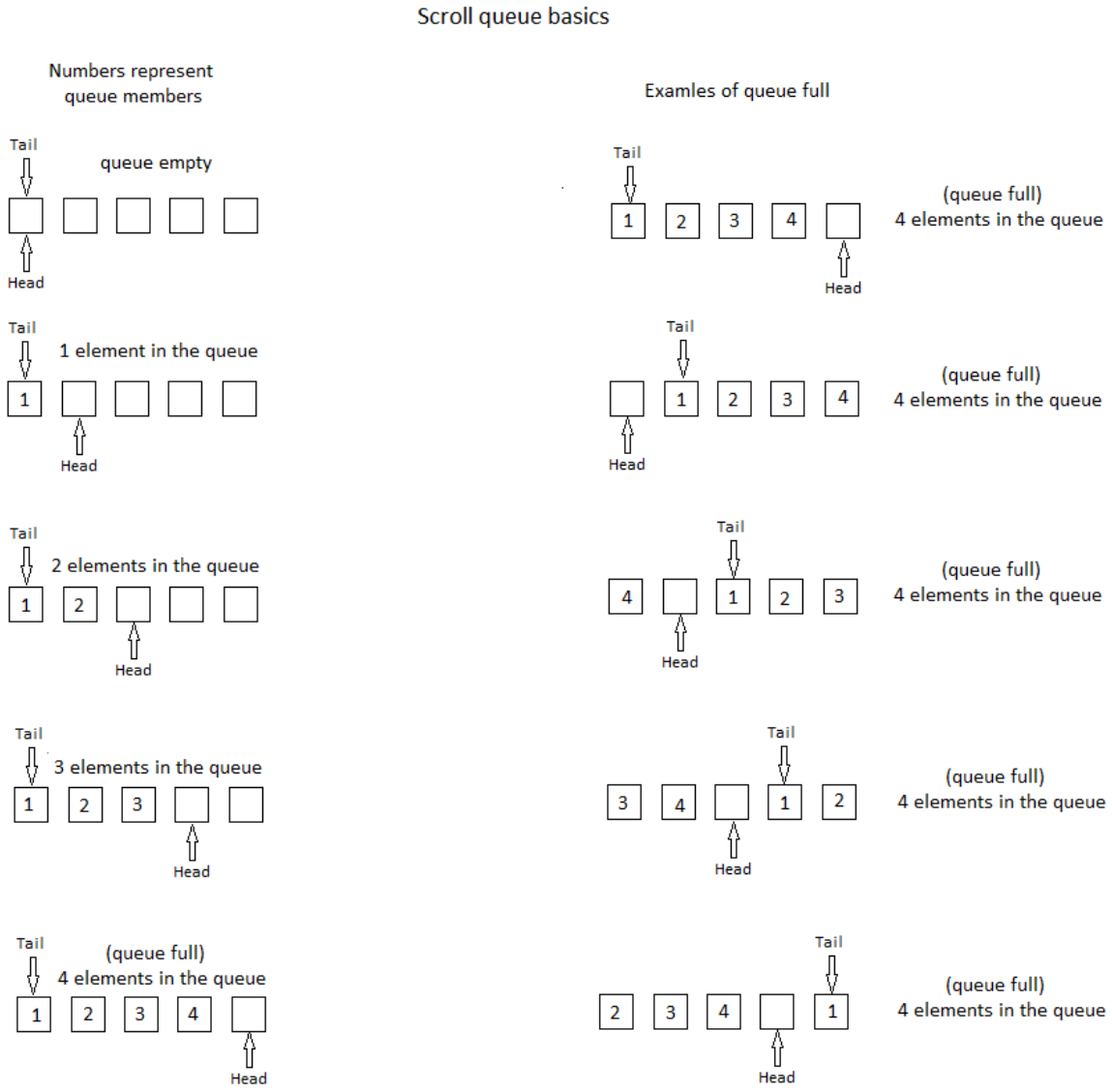
Framebuffer[20]:
Bit 0 = Display matrix 4,15
Bit 1 = Display matrix 4,16
Bit 2 = Display matrix 4,17
Bit 3 = Display matrix 4,18
Bit 4 = Display matrix 4,19
Bit 5 = Display matrix 4,20
Bit 6 = Display matrix 4,21
Bit 7 = Display matrix 4,22

Framebuffer[21]:
Bit 0 = Display matrix 5,15
Bit 1 = Display matrix 5,16
Bit 2 = Display matrix 5,17
Bit 3 = Display matrix 5,18
Bit 4 = Display matrix 5,19
Bit 5 = Display matrix 5,20
Bit 6 = Display matrix 5,21
Bit 7 = Display matrix 5,22

Framebuffer[22]:
Bit 0 = Display matrix 6,15
Bit 1 = Display matrix 6,16
Bit 2 = Display matrix 6,17
Bit 3 = Display matrix 6,18
Bit 4 = Display matrix 6,19
Bit 5 = Display matrix 6,20
Bit 6 = Display matrix 6,21
Bit 7 = Display matrix 6,22

Framebuffer[23]:
Bit 0 = Display matrix 7,15
Bit 1 = Display matrix 7,16
Bit 2 = Display matrix 7,17
Bit 3 = Display matrix 7,18
Bit 4 = Display matrix 7,19
Bit 5 = Display matrix 7,20
Bit 6 = Display matrix 7,21
Bit 7 = Display matrix 7,22

# Appendix K – Scroll queue basics




Note: Also applies similarly to “Sound queue”.

# Appendix L – Revision history

Revision history.

- Initial release from Waveshare

10-FEB-2022 2.00

- Global code reformatting and rework.
- Fix a bug allowing "DayOfMonth" to be set to 0.
- Fix a bug allowing "DayOfMonth" to go higher than the maximum of 28, 29, 30 or 31 (depending of the month).
- Fix a bug and make sure the "count-down" indicator is turned off when count-down timer reaches 00m00s.
- Fix a bug when the clock is set for 12-hours time display and is displaying 00h00 AM instead of 12h00 AM.
- Add a "test section" to put many chunks of code for testing and debugging purposes.
- Add "FRENCH" option so that the date can be displayed in the corresponding format (compile-time option).
(It will be easy to add other languages if the programmer makes a search for "ENGLISH" and / of "FRENCH" in the source code...
...and assuming the characters used are standard ASCII characters, similar to English).
- Implement 5 X 7 character set with variable character width.
- Implement a reverse_bits() function allowing the bitmap of the 5 X 7 characters to be more intuitively defined.
- Add a generic "scroll_string()" function and a fill_display_buffer_5X7() function to easily handle 5 X 7 characters.
- Change the name of many functions to make them more representative of what these functions do.
- Make the scroll_string() function so that it can accept a string longer than what can be handled by the framebuffer. The function will wait until the framebuffer get some free space to transfer next chunk of the string.
- Add a specific section at the top of the source code to select many default clock options at compile time.
- Implement different tone types with different number of tones and duration for different events.
(Note: Tone frequency can't be changed since the oscillator is integrated into the clock piezo).
- Clock display will blink the two center dots according to the number of seconds passed since the last minute change.
- Add automatic handling of "Daylight Saving Time" for most northern hemisphere countries.
(Provision has been made to add different algorithms of daylight saving time for other areas of the world).
- Add a new option for hourly chime. It may be On, Off, or "Day" (that is: OI for "On, Intermittent"). If set to "Day", chime will sound only during day hours, as defined between CHIME_TIME_ON and CHIME_TIME_OFF that can be set during clock setup.
(They are set to 9h00 and 21h00 by default).
- When Hourly chime is set to “Day”, on ly the left LED will be On in the “Hourly chime” indicator, so that user knows what is the setting.
- Add an option in the clock setup to configure the Chime Time On and Chime Time Off.
- Add handling of "Calendar events" that can be added by user at compile time. During the specified day of the year,
the string defined by the user will scroll on the screen twice an hour, at xxh05 and xxh35. Moreover, a special sound will be heard when the message start scrolling during daytime as defined between CHIME_TIME_ON and CHIME_TIME_OFF.

24-MAR-2022 3.00 - More code rework and optimization.
- When powering up the clock, replace DST = 0xFF (hex value) by "DST = On" (and replace "DST = 0x00" by "DST = Off"). Will display “None” if daylight saving time has been set to Off (no support).
- Add logic for 2 different "repeat counts" for sounds, to add versatility and more possibilities for different sound codes.
- Day-of-month will change automatically if it becomes out-of-bound while setting up current month (for example, day-of-month will change automatically from 31 to 28 while we change month from 3 (March) to 2 (February).
- Add suffix to day-of-month when scrolling the date in English (will now display "...March 31st...").
- Setting Chime time on and Chime time off now comply with 12 or 24-hours time format current setting.
- Fix a bug allowing Hourly Chime to sound one hour later than Chime time off.
- Fix a bug allowing Hourly Chime to sound sometimes while doing clock setup.
- Setting alarms now comply with 12 or 24-hours time format current setting.
- To help knowing which alarm is On (0 or 1), when Alarm 0 is On, Left Alarm LED is On in the alarm indicator and when Alarm 1 is On, Right Alarm LED is On in the indicator.
NOTE: On power up, both alarms are set to OFF (this is also true in case of power failure), since there is no backed-up RAM available in the RTC IC to save such variables in case of power failure.
- Change the logic so that each alarm (0 and 1) is now checked individually in the RTC IC.
- Add the logic for each alarm (0 and 1) to have a distinct (different) alarm sound.
- When setting up alarms, current alarm parameters saved in RTC IC are now proposed to user as default values.
NOTE: As mentioned above, alarm status (On or Off) is lost on power failure (and will restart as Off), but other alarm parameters (Hour, Minute, Day-of-week) are kept in RTC IC with battery back-up.
- Now blink the day-of-week while setting up day-of-week in alarm setup.
- Language selection is now a run-time option. English and French are the available languages for now.
- Hourly chime: add logic for "nighttime workers". If Chime time on is greater than Chime time off (as opposed to what we would normally expect), we assume that we want sounds to be heard during nighttime and not heard during daytime. Hourly chime will then sounds after Chime time on (in the evening) and before Chime time off (in the morning).
- Add support for DHT22 (humidity and temperature sensor - compile time option). To implement this option, it must be "#define" in the code, and a 3-wire cable must be installed between GPIO 8 VSYS and GND, and a DHT22 can be installed outdoor.
- Add Daylight saving time as run-time parameter. User can now select 0 (no support for DST) or 1 (North-America-like DST support).
- Add changes already made by David Ruck to fine tune auto brightness.
- When time display format is H12 (12-hours format), do not print a leading 0 to Hour, to comply with the standard.

19-APR-2022 4.00 - Some code optimization.
- Major cleanup in the "setup flags" to make it more straightforward to add options to the clock setup.
- Transferred "Temperature unit" setup to the list of clock parameters setup.
- Transferred "Auto-brightness" setup to the list of clock parameters setup.
(NOTE: A copy of the "auto-brightness" function code has been left on the "bottom button quick press" for faster access if required).
- Implement new "Night light" feature: use the two white LEDs inside the clock as a night light. It can be

turned On, Off, OI (“On, Intermittent” - see Hourly chime for explanation about "OI" setting), or AU (“Auto”).
- Since temperature unit setting is now transferred in the general clock setup, "Up" (middle) button quick press will now scroll on clock display the outside temperature and humidity if user has installed a DHT22. White LEDs will also blink on clock side.
- To be more deterministic, instead of scrolling the date every xxx seconds, scrolling will now occurs when reaching an integer number of times the period specified. So, if we scroll the date every 5 minutes, scrolling will occur at xh00, xh05, xh10, xh15 and so on. This way, we know how long we have to wait until next scroll.
- Add support for a remote control (see User guide for details). Infrared sensor must be bought by user to enable this option.
- Time checking of calendar events has been changed from xxh05 and xxh35 to xxh14 and xxh44. Those values are unlikely to be a multiple integer value of the time period to display date and then, those two actions (date display and calendar events) will not interfere with one another.
- Implement many functions that are accessible only with remote control (see User's guide for details):
- Emulate the three clock buttons on the remote control so that we can achieve a similar behavior remotely.
- Fix a problem in reading VSYS (power supply voltage).
- Add some pixel twinkling when software starts (just for fun...)
- Add column-by-column LED matrix test to make it easy to see if all LEDs work fine. Idem for all clock display indicators.
- Add firmware support for an optional passive piezo so that sound frequency may be changed / modulated by software. PWM has been selected to drive the passive piezo. If we would use a completely "software" oscillator, we would hear hiccups in the sound when the other ISR / callback functions would run (this processing would interfere with the regularity of the sound). Similar to the scroll queue, a sound queue (circular buffer) has been implemented to handle the sounds of the passive piezo.
- Add the structure member "Jingle" to the calendar Event structure. If it is not null, it identifies a jingle (quick music) that will play when the calendar event is scrolled on the display. (For example, we could ask to play "Jingle bell", along with the message "Merry Christmas" for December 25th. (NOTE: Passive piezo must have been installed by user to support this option).
- Add Pico internal temperature display (compliant to the temperature unit - C or F - that has been selected for other temperatures).
- Add programmable "silence period" to temporarily turn off hourly chime and calendar event sounds (available with remote control).
- Add "dice rolling" to randomly display 2 dice values (from 1 to 6) on display (avalailable with remote control).
- Modify get_ads1015() function so that light value returned is intuitive: higher number means more light. Auto-brightness and automatic night-light functions have been modified accordingly.

27-JUL-2022 5.00
- Uncomment matrix_test() and pixel_twinkling() in power-up sequence that were left mistakenly commented-out in release 4.00.
- Add scrolling of firmware version number during power-up sequence and in display function.
- Scroll Pico's "unique number" ("serial number") during power-up sequence and in display function.
- When hour changes (from xh59 to xh00), date scrolling will now begin only 5 seconds later to let the time to look at the clock display to see current time when we hear the hourly chime. This is true only from xh59 to xh00 and not for other integer number of times the specified period (for example from xh04 to yh05 if date scrolling period is set to 5 minutes).
- Command "Display auto-bright" (with remote control generic display): fix the logic for French message.

- Modify the logic behind "automatic night light". There is now a "twilight" period during which the night light status does not change. This is to prevent a period of "On/Off" toggling when light level is just around the value to change night light status.
- Modify the logic behind "clock auto-light" (or automatic brightness), so that clock display does not dim at all when ambient light intensity is high enough.
- Also, slow down the frequency for reading ambient light (from one second to five seconds). Auto-brightness Will now take a few seconds to react when changed with remote control.
- Adjust / change the ambient light levels for clock display dim intensities.
- Since the auto-brightness of the clock display has been fine-tuned, the default configuration when the clock is powered on is now auto-brightness ON.
- Add UART configuration to allow sending data to a VT101-type monitor connected to Pico's uart1.
>>> NOTE: be careful to adjust voltage logic levels before connecting an external monitor.
NOTE: see user guide on how to receive and display data from a Pico to a PC screen through USB port.
- Improve / optimize the logic of the sound circular buffer and sound related functions for passive buzzer.
- Add support for BME280 sensor (temperature, humidity and barometric pressure). The BME280 uses the same I2C channel as the DS3231 real-time clock.
- Add calculation of approximative altitude based on barometric pressure. After a few tries, it has been commented out since it changes with pressure variations and it appears to be not very accurate.
- If a BME280 has been installed, scroll "device ID" and device "unique ID" on clock power-up.
- Rework algorithm of LED matrix automatic brightness and add more brightness levels.