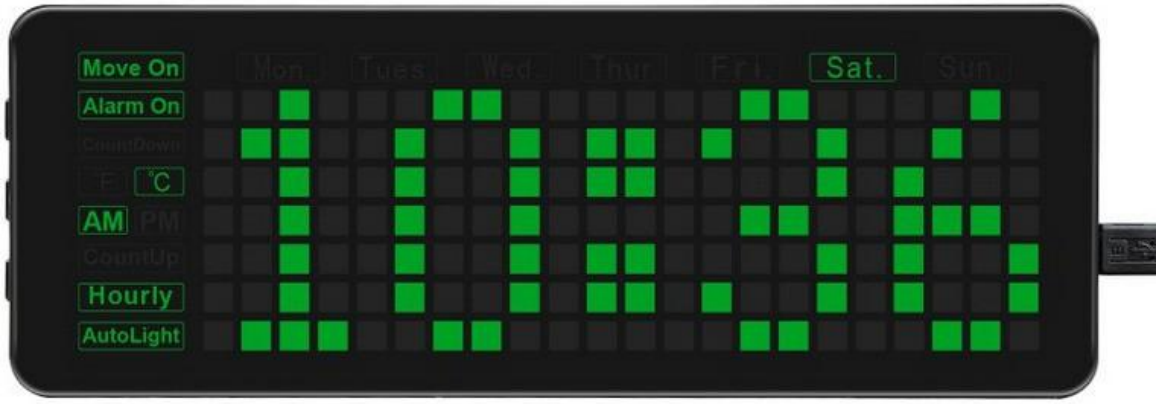
Pico Green Clock



Firmware Version 10.10 User Guide
Updated January 3rd, 2025

IMPORTANT :
This User Guide is about Pico Green Clock firmware Version 9.02 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.
Versions 10.00 onwards have additional features not supported by Andre St. Louys.

Join our Pico-Green-Clock discussion group on:
https://github.com/astlouys/Pico-Green-Clock/discussions

Pico Green Clock User Guide

Index

Pico Green Clock User Guide

Pico Green Clock User Guide

Do you want to share your experience with the Pico Green Clock and help other users ?
Join our discussion group on:
https://github.com/astlouys/Pico-Green-Clock/discussions

Among the subjects of interest:
Which version of the firmware do you use?
Do you like it?
What are the features you appreciate the most?
Did you find some bugs (what are they)?
Did you improve the firmware (how)?
Did you add some more features (which)?
Let us know!

If you want to send me a personal email (as long as it is something constructive),
here is my email address:

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

About Version 9.02

The main features / changes in Firmware Version 9.02 are:

1) Fix a problem when initializing CYW43 with the new Pico-W SDK library.
2) Delayed date scroll at half-hour light chime so that user has enough time to take a look at the time before the date begins scrolling.
3) Modify the "middle-dots" blinking algorithm (see later in the User Guide for details). Thanks to Frank Seidel for his suggestion and coding.
4) Add basic German language support. Thanks to Frank Seidel for the translation.
5) Add Czech language support. Thanks to KaeroDot for the excellent work!
6) Increase string length of Calendar Events up to 50 characters.
7) Fix dates encoded in CalendarEventsGeneric.cpp (Calendar Events examples).
8) Many other cleanup and optimizations.

As usual, you can refer to the revision history in the source code and / or at the end of this guide to get details about other changes among different Firmware versions.

Notes from previous Firmware version:

NOTE: If you are a developer and you plan to work on the source code (particularly on NTP), you may want to review the rules of engagements requested from public NTP servers. The Green Clock complies with those rules and it is important that you remain compliant if you modify the code.

NOTE: For those already using a previous version of the Firmware: If you installed a Pico-W with NTP option turned on (which is the default), you will realize that there is a small delay (more or less 5 seconds) before you see the time display on the clock. This delay is used to establish a NTP connection and update the time to make sure time display is correct from the beginning (this assumes that DST country and Timezone have been properly initialized, tough…) Moreover, if the credentials saved in flash memory are wrong, the clock will retry ten times before giving up (you can watch the Pico-W`s LED under the back plastic case to count the retries. In this case, it may take up to one minute for the clock to come up with something on the display.

***** IMPORTANT *****

You can refer to Appendix L on how to setup Network Time Protocol ("NTP") on your Green Clock.

You must understand that when you configure your clock to support NTP, you will actually save your credentials to the Pico-W's non-volatile memory (that is, your "SSID" - or network name - and password) for accessing your Wi-Fi network. If someone else has physical access to your Green Clock and / or Pico-W microcontroller, they will be able to retrieve those credentials and access your Wi-Fi.

Pico Green Clock User Guide

If ever you need to lend your clock (or your Pico-W microcontroller) to someone else and feel more comfortable to erase your credentials before, you may proceed in two different ways:

1) Use the Pico-Flash-Utility.uf2 in one of my repositories to completely erase your Pico-W`s flash memory.
2) Refer to Appendix L and save two “dummy” strings as your SSID and Password to overwrite the original ones (make them long enough to completely overwrite the old ones).

About Version 10.00

The programming of the clock is quite complex. The big advantage of this clock design is that the time is synchronized to an NTP server and that the configuration is restored on power on. I wanted a clock that will still wake you up if you have a power interruption in the night. The number of different alarm combinations was a good feature. This is of course personal and everyone's needs are different. The controls are very complex and I wanted to make it simpler, so I set about building a web based control interface for the Pico-W that augments the one provided in this guide. Version 10.00 adds this capability and extends this with the following feature changes:

1) Add extra dark dimming to the display. The clock display is quite large so a lot of light is emitted even when at the dimmest display.
2) Add a hostname to the unit so that multiple Pico-W based Pico Green Clocks can be present on the same WiFi network.
3) Each Pico-W light sensor gives a different lower reading for a dark room. It was therefore impossible to come up with a common lower limit in the code. The code is then able to support measuring the light sensor values, storing maximum and minimum values and providing different step points based on a variable minimum value. This is only configurable on the web page.

In developing this code, the following enhancements were added:

1) Added the ability to immediately silence an active alarm by pressing any of the clock's physical buttons rather than only the "set" button.
2) Added the ability to mute all alarms for a 4 hour period by pressing any two keys and releasing them together. This is intended for times when you wake up a few minutes early and want to temporarily disable the alarms. The mute period is fixed in the code.
3) Added the ability to swap the time and alarm set button press duration modes mentioned in the section on Entering Clock Setup Mode and section on Entering Alarm Setup Mode. This is because with the NTP synchronization, the clock rarely needs setting but the alarms will change. This is only configurable on the web page.
4) Added the ability to beep the Piezo or the internal buzzer and added different jingles that can be configured per alarm. This extends the previous alarm number setting the beep sounds to provide more flexibility. This is only configurable on the web page. The cascading alarm sound effect has been removed and now only the first triggered alarm will sound.
5) Added the ability to set different alarm messages. These are then easier to set on the web page.
6) Set the dimming button to be able to step between 5 manual fixed levels in addition to the automatic mode.
7) Ability to change the WiFi network SSID and password. Since this needs the web page to change, The Pico-W must already be connected to a WiFi network and accessible. This is only useful when moving between WiFi networks.
8) The web page makes the configuration and control of the hourly chime mode and night light easier to follow and set up.

There were some other minor changes to clocks operation when running in 12hr mode.

About Version 10.10

The version 10.10 code added the ability to change the pattern used for the hour / minute separator ':' character. It can be set to be static, flashing every second, following the original 4 character pattern or a new 5 segment bar graph. This is configurable on the web page only.

Introduction

Waveshare Pico Green Clock

As mentioned on the cover page, this User Guide is about Firmware Version 10.10. This si an extended design based on Firmware Version 9.02 from Andre St. Louys. Firmware Versions 2.00 and up are developed by Andre based on the original Waveshare firmware Version 1.00 and add more features to the clock and tons of comments in the source code which is in C Language.

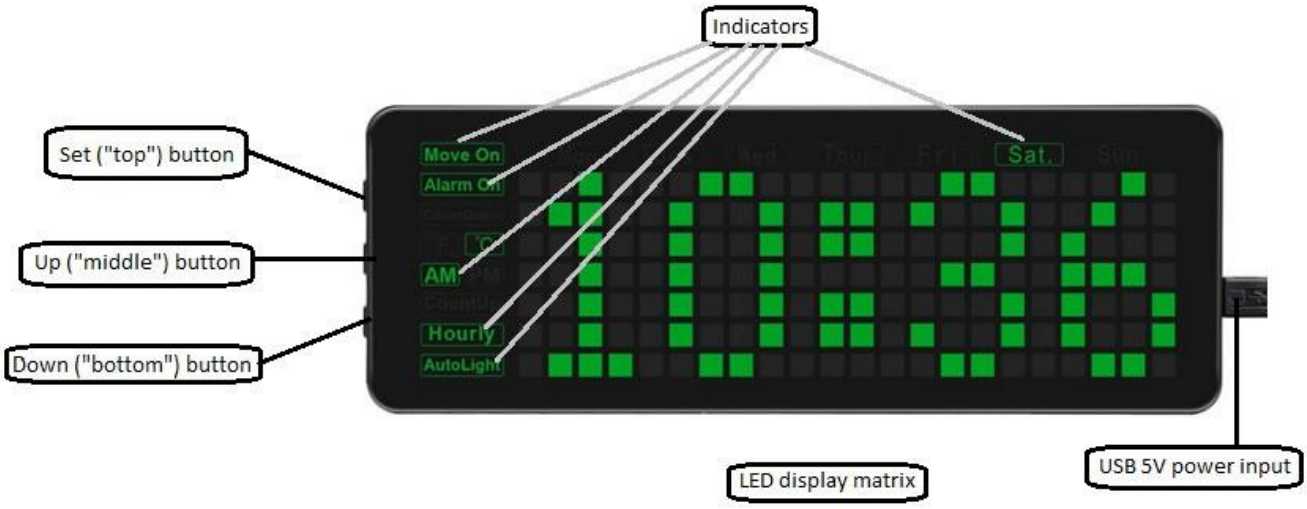
The Waveshare's Green Clock uses a Raspberry Pi Pico (or Pico-W) microcontroller to control most of its functions / features (IMPORTANT: the microcontroller itself, Raspberry Pi Pico (or Pico-W), is not included with the clock. However, it is not very expensive (more or less 15 US$ at the time of this writing). 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 and date in case of a power failure. Version 6.00 (or higher) allows for most clock configuration parameters to be saved to Pico's flash memory. In case of a reboot and / or power failure, those parameters are read back from flash memory so that there is no need to go through the clock configuration after each power on.

Beginning with Firmware Version 9.00, the Green Clock may synchronize its time with an Internet based time reference (which is known as "Network Time Protocol" or NTP for short). Even if the real-time integrated circuit in the Green Clock keeps track of the time with a very small drift over a long period of time, NTP allows to resynchronize the clock often enough to keep this drift to a minimum. The downside is that you will need to reprogram your SSID (network name) and Password to make sure your clock keeps its Internet access (refer to Appendix L). If not, it will continue to keep the time from its internal real-time integrated circuit (with battery backup) without resync's from the Internet.

Many 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 source code to get 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") in the Pico microcontroller's non-volatile memory 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.



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 Configuration Saved to Flash Memory

If you upgraded from a previous Firmware version (older that Version 6.00), you may have to go through the clock settings cycle once, in order to properly adjust the new available settings that would have been set with default values. Once this is done, flash configuration will be read back at each power up, restoring the configuration you previously made.

If ever something goes wrong with your configuration saved in Pico`s flash, you may use the Pico-Flash-Utility.uf2 in one of my repositories to completely erase the Pico`s flash. When restarting the Green Clock after that, a default configuration will be automatically created with defaults values.

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.

Pico Green Clock User Guide

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 be executed 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 is configured “On”, 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 temperature unit that is configured.
10) If hourly chime setting is configured “On” or “Day”, the “Hourly” indicator will be turned On accordingly.
11) Scroll 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, BME280 algorithms have been tuned so that the clock will perform normally even if no BME280 has been installed, even if the conditional compile is turned On.
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 (consider the “Unique ID” as a “serial number”). The Unique ID is a 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, DHT22 algorithms have been tuned so that the clock will perform normally even if no DHT22 has been installed, even if the conditional compile is turned On.
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 microcontroller type (Pico or Pico W).

18) Display Pico microcontroller's "Unique ID" in hexadecimal with a format of: XXXX-XXXX-XXXX-XXXX (consider 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 your clock. If you would replace it with another Pico, the Unique ID would be different.

19) Scroll current "Daylight Saving Time" / "Summer Time" mode / status, according to clock configuration and current time and date. The first information is the country setting for Summer Time (see in Appendix L at the end of this User Guide). Then, 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: The clock manages automatically the change between active and inactive, based on current time and date and in compliance with the clock DST setting (refer to the clock setup later in this guide).

20) Display power supply voltage value (voltage supplied by USB power supply). It should be around 5.0 volts, but it may be a little higher or lower. (Note: this option / function needs review. Ignore its value for now).

21) If sound has been cut off at compile time, a warning will scroll on clock display. There is an option in the source code to completely turn off all sounds generated by the clock. I used this feature during the development phase / cycles, 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 to explain how come no sound comes out of the clock! NOTE: ***ABSOLUTELY NO SOUND*** will be generated by the clock if this option is set at compile time.

22) Display Pico microcontroller internal temperature (it is usually 1 or 2 degrees higher than ambient temperature).

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

24) Display time.

Clock Features

Here is a brief description of clock features (in alphabetical order). More details are given later in the guide where most features are described in their specific section.

Alarms (Run-Time)

Beginning with Version 6.00, there are now nine (9) alarms available on the clock. They can be independently set at a specific time (hour and minute) and with a selection of any number of days-of-week (new with Version 6.00), and they will sound an alarm when the programmed alarm parameters are reached, and a specific text will scroll on clock display to show which alarm is ringing (this text can be customized by user at compile time, see later in this User Guide). If a passive buzzer has been installed, the sound pitch will be slightly different for each alarm number and the number of "beeps" will correspond to the alarm number being heard. The alarm will sound during a one-hour period, or until the user presses the "Set" (top) button, whichever happens first.

NOTE: When user presses the "Set" (top) button, all alarms will shut off (in case more than one alarm is ringing). If a count-down timer alarm is also currently ringing, it will shut off as well.

NOTE: From Firmware 10.00, a playing alarm will silence immediately when any button is pressed. This allows for a sleepy arm to correctly silence a playing alarm so as not to disturb other people. A temporary disable has been added to allow alarms to be skipped if they are not required. A 4 hour mute period can be activated by pressing any two buttons and releasing them together.

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). Take note that there is now a hysteresis with this feature so that the brightness will change smoothly with changing light conditions according to the average ambient light for the last 120 seconds period (and not change back and forth very quickly as soon as something runs in front of the clock as was the case before).

The ambient light detection / sensor takes advantage of one of the analog-to-digital converter integrated in the Raspberry Pi Pico. Also, beginning with Version 8.00, the clock display brightness is controlled by a pulse width modulation (PWM) signal from the Pico microcontroller. This offloads the software to take care of this function, and it also adds literally thousands of different brightness levels (with the previous Firmware versions, only 5 or 8 brightness levels were available). Keep in mind that the clock display may be ***very*** dim if the ambient light is dark. Also remember that if you turn On the light, the clock display will gradually increase its brightness, due to the implementation of the hysteresis (up to two minutes may be required to come up to full clock brightness… the same happens when you turn Off the light)

Beep Types (Compile-Time)

The piezo (buzzer) provided in the Green 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 even more 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 User Guide.

Version 6.00 brings the feature one step further by implementing an active buzzer queue (circular buffer) to optimize the code and make it much easier to handle the clock sound activity for those who want to play in the source code.

Firmware Version 4.00 and up also supports a "passive piezo" (that must be bought and 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 (50 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 (see file "CalendarEventsGeneric.cpp"). Up to 50 such calendar events may be programmed and more than one may be programmed for a specific day.

The Calendar Events will scroll on the screen during all day at the date defined in the clock source code at compile time (the firmware must be rebuilt and re-flashed in Pico's memory). 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 period 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" in the clock configuration (set by default from 9h00 to 21h00). You will find more information about this, later in this guide.

If a passive buzzer has been installed by user, a specific jingle may be selected in the CalendarEventsGeneric.cpp file and the specified jingle will be played while the Calendar Event description is scrolled on the clock display.

A few of these Calendar Events have been programmed by default to show the user how to program others if desired. For example, "Merry Christmas" and "Happy New Year" 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*" will be 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 (in minutes and seconds). Once the count-down timer reaches zero, an alarm will sound (without respect to "Chime Time On" and "Chime Time Off"). The count-down timer alarm will sound every few seconds during a 30-minutes period, or until user presses on the "Set" (top) button, whichever happens first.

NOTE: When user presses the "Set" (top) button, the count-down timer alarm will shut off. If one or more "clock alarms" were also ringing at the same time, they will all shut off simultaneously as well.

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 (Compile-Time for What to Scroll and Run-Time for On/Off)

The user may configure the clock to scroll a lot of information if so desired: current date, ambient temperature, external temperature, relative humidity, atmospheric pressure (if user installed a BME280 or a DHT22), Pico's internal temperature, power supply voltage Wi-Fi credentials, ambient light level, system idle monitor, etc… The information will scroll 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 this User Guide).

Beginning with Firmware Version 3.00, the date and temperature are displayed as in "Friday July 6th 2022 21.25°C" (the temperature is shown after the date). The letters are made of 5 X 7 character matrix, variable-width.

The support for outside temperature, relative humidity and atmospheric pressure reading is available when user adds a BME280 sensor (not supplied with the Green Clock itself). See section on BME280 and / or DHT22 support later in this User Guide.

Daylight Saving Time (Run-Time)

NOTE: The concept of Daylight Saving Time (DST) is also called in some countries: Summer Time / Winter Time and / or Spring Forward / Fall Back

The firmware automatically supports Daylight Saving Time for most (if not all) countries of the world. When properly configured, this feature will automatically increase or decrease the time when reaching the Summer Time or Winter Time trigger date and time (usually by one hour, but for some countries, the time shift is 30 minutes).

The time change will occur one second after crossing the xxh00 hour change. This is to prevent the case where there is a small difference between the "clock time" and the "real- time integrated circuit time". In this particular case, the time change (+60 minutes or -60 minutes) would have happened twice in a row. It is assumed that this one-second precision is good enough at the time of time change.

The clock will also automatically adjust the Daylight Saving Time status in the clock configuration when user changes the clock settings to modify time and / or date.

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

The Daylight Saving Time feature can be turned Off if user so desires (refer to the “DST section” of Clock Setup later in this guide).

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 sounds to be heard during the night, when people are sleeping.

There is also support for “nighttime workers” where the sounds are heard only during the night. See section on how to setup Chime Time On and Chime Time Off later in this User Guide.

Digits

The digits are built using 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 most 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 that 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 up to 14 sec. Then, the bottom dot will blink from 15 sec up to 29 sec. After that, both dots will blink alternately (one after the other)

from 30 sec. up to 44 sec. Finally both dots will blink together from 45 sec up to 59 sec. This allows someone to quickly evaluate (relatively) how “deep” we are in the current minute, and if the next minute change will occur soon.

Flash Memory

Beginning with firmware Version 6.00, the clock configuration is saved to flash (Pico’s non-volatile memory). During the power-up sequence, the configuration is read from flash memory and restored to the clock. If this is the first time that the new Firmware is run (Version 6.00 or a more recent version), the clock will detect a “wrong” flash configuration (in fact, wrong checksum since at this point, there is no configuration at all in flash memory) and will generate and save a default configuration.

Once this is done, every five (5) minutes thereafter, the clock will compare the checksum of its active configuration with the one of the flash configuration. If user has made some change to the active clock configuration, it will be detected by the clock (checksums won't match) and the new active configuration will be saved to flash just before the next date scrolling on the clock display. Updating the flash configuration just before date scrolling prevent a "glitch" on the clock display since it is mandatory to disable interrupts to program the flash. Doing so stops the scanning of the LED matrix for a few milliseconds, which is not elegant. By disabling interrupts just before date scrolling, it is possible to blank the display for a few milliseconds, program the flash, and start scrolling the date just after. This process allows for a smooth behavior of clock display. Take note that no active wear-leveling algorithm has been implemented in the Firmware for flash writes. Most probably, writing a new configuration here and there during the life span of the clock will not be a problem for the +/- 100,000 flash write cycles estimated during the life of the Pico, but user must be aware that if flash writes are to occur many times a second, current algorithm is not appropriate. That being said, as a "passive wear leveling", the configuration is saved to flash only when the clock is in the normal "time display" mode, and just before date scrolling. This means that if user remains in "setup mode" for a while and change many configuration parameters, chances are that only one flash write will occur, once user exit setup mode and go back to time display mode.

The following configuration parameters are saved to flash memory:

Field	Length	Content
Software version number	6 Bytes	Will eventually make updates easier if ever the configuration changes in the future.
Current year centile	1 byte	First year two digits ("20"). Not very useful since it could have been hard-coded, but not a big problem to simply add it to the flash configuration.
Language	1 byte	Czech, English, French and German (basic translation only for German) are the supported languages for now. NOTE: If the text / message is not available for the target language, the English message will be used instead.
Daylight Saving Time Country Code	1 byte	Select the appropriate algorithm for User's country. `Not supported' may also be selected. Once properly setup, the clock will determine on power-up if Summer Time must be active or inactive, and will make automatic adjustment to clock time and UTC time at the appropriate moment of time change in spring and fall. It will also adjust the Daylight Saving Time if / when user changes the clock time or date.
Temperature unit	1 byte	Celcius or Fahrenheit
Time display mode	1 byte	24-hours or 12-hours mode
Chime mode	1 byte	On / Off / Day ("OI")
Chime Time On	1 byte	9h00 by default
Chime Time Off	1 byte	21h00 by default
Night light mode	1 byte	On / Off / Night ("OI") / Auto
Night light time On	1 byte	21h00 by default

Field	Length	Content
Night light time Off	1 byte	8h00 by default
Auto brightness	1 byte	On / Off
Beep (keyclick)	1 byte	Audible feedback when pressing clock buttons
Scroll enable	1 byte	Scroll the date every 5 minutes.
FlagSummerTime	1 byte	Flag indicating current status of Daylight Saving Time (active or inactive). This is kept in flash configuration, but it is not adjustable by user.
Timezone	1 byte	Difference in time between “local time” and “UTC time” (Coordinated Universal Time)
Dimmer Min Level	2 bytes	The lower value that the light sensor reads to set the display to its lowest illumination level – set the PWM control to its lowest duty cycle
Short Set Key	1 byte	Flag to set the short “Set” top key to enter the Time / Date or Alarm setting mode
Separator Mode	1 byte	Numerical code for the hour / minute ‘:’ separator mode, 0 & 3 = default pattern, 1 = static, 2 = 1s flash, 4 = bar graph
Reserved	44 bytes	Reserved for future use
Alarms 1 to 9	9 x 48 bytes	Alarms parameters: Status (On / Off) Alarm hour Alarm minute, Alarm second (not adjustable by user), minute, hour, Bitmask of active alarm days (any combination of days of week) Alarm jingle / beep pattern, 2 reserved byes, Displayed text
Hostname	40 bytes	Clock network hostname
SSID	40 bytes	“Service Set Identifier” also called “Network Name”.
Password	70 bytes	Network password to get Wi-Fi access for Network Time Protocol synchronization
Reserved	48 bytes	Reserved for future use
CRC16	2 bytes	Cyclic redundancy check of the configuration data.

Half Hour Light Chime (Run-Time)

Every half hour, at xxh30m00s, a half hour “light Chime” may sound to indicate that we reached the half hour. This option follows the same rules as the hourly Chime settings (see “Hourly Chime” later in this User Guide).

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 this User Guide).

Hourly Chime (Run-Time)

Every hour, at xxh00m00s, an hourly Chime may sound to indicate that the hour just changed. This feature can be configured: On, Off, or Daytime ("OI"). If configured for "Daytime" ("OI" for "On, Intermittent" in Clock Settings), the hourly Chime will sound only during predefined hours in clock configuration (set between 9h00 and 21h00 by default) (see section on Clock Setup later in this User 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 1-left-LED On and it may take some practice for the user to see the difference.

Idle Monitor

It may be interesting to know what the current load is, on the microcontroller (Pico). Since firmware Version 6.00, a "System idle monitor" has been implemented. It gives an "average number of system loops performed by the firmware every second, based on the analysis / history of the last 60 seconds". That means the highest the number, the less busy is the system.

System idle monitor can be displayed on clock display with the remote control (remote must be installed and configured by user). It can also be added to the clock scrolling data by a quick change to the source code.

For those interested in going deeper in this analysis, you may want to connect an external terminal emulator to the Pico's USB port and enable the DEBUG_IDLE_MONITOR option.

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)

The following languages have been implemented so far in the clock Firmware for date display and many other messages: Czech, English, French and German (only basic translation for German). 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 setting for Firmware releases and it can be changed in clock configuration saved to flash memory.

NOTE: When some text / message is not available for the target language, English message will be used instead.

Night Light (Run-Time)

Since the two white LEDs on the side of the clock (near the three 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…

Moreover, be aware that given the way the electronic circuit is built, those two white LEDs will follow the green matrix display LEDs brightness. So, if the clock brightness is low, the two white LEDs will be dimmed as well.

Network Time Protocol

Network Time Protocol ("NTP") allows the Green Clock to synchronize its time with a time reference server over the Internet. A "Pico-W" microcontroller is required for NTP. It is 99% compatible with the Pico, but it features a Wi-Fi interface which is required for NTP. (The only compatibility difference is for the Pico-W LED which doesn't turn On or Off the same way as for the Pico).

Appendix L will give you all information required to configure NTP in the Green Clock. You will need to have access to a Wi-Fi network and know the correct credentials to access it ("Network name" and "Password").

Power Supply Voltage Display (Run-Time)

NOTE: This section must be ignored for now. Code must be reviewed…

You may scroll the power supply voltage on clock display with the remote control (remote must be installed by user). It can also be added to the clock scrolling data every five (5) minutes by a quick change to the source code.

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 will be scrolled on the clock display along with the date (and with other information if you change the source code accordingly). The temperature may be shown in Celsius or Fahrenheit. This is a run time option (see section on Clock Setup – "Display Scroll" - later in this User Guide).

If user installed a BME280, it can be used to read outside temperature, relative humidity and barometric pressure and those three values may also be scrolled on clock display. Simply add the instruction "scroll_queue(TAG_BME280);" in the periodic data scrolling, then rebuild and re-flash the Firmware (search for the string "PERIODIC SCROLLING CONFIGURATION" to find the point in the code where you can change the data that will scroll every five minutes)..

The text in the preceding paragraph also applies if user installed a DHT22 temperature and humidity sensor. The instruction to be used for a DHT22 is “scroll_queue(TAG_DHT22);”

If a DHT22 and / or a BME280 has been installed by user, total number of errors while reading the device(s) DHT22 and / or BME280 are reported while scrolling the date (for statistic purposes). For example, “(2/1758)” means that there has been 2 reading errors out of 1758 readings cycles of the DHT22. Same applies for BME280, but it is indicated between square brackets instead of parenthesis. For example: “[2/12434]”.

Timezone (Run-Time)

The support for the Timezone (difference between local time and UTC time) has been added since Firmware Version 7.00. User must set this parameter once, when configuring the clock, but the clock will then maintain it automatically (since this time difference changes when hour changes from Summer Time to Winter Time and back).

Keeping track of the Timezone value allows correct algorithm behavior for those countries that use UTC time as a reference for changing from Summer Time to Winter Time (and back). This parameter is also be required if you want to take advantage of NTP (“Network Time Protocol”) to synchronize the Green Clock with an Internet time reference.

Options

The following are options that are not provided with the clock but for which support has been added in the Firmware. If you supply 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. Support for it has been added 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!). I had problems with the initial use of the DHT22. Timing seems very critical and doesn’t get along well with all callback and other interrupt service routines. To solve the problem, I moved the code to support the DHT22 to Pico’s second core (core 1). Since that time, I got almost 100% successful read cycles. While running on core 1, DHT22 communications are not corrupted with callbacks and / or other interrupts. I also fine-tuned the algorithm on my way.

BME280

BME280 is a temperature, humidity and atmospheric pressure sensor. Support has also been added for it in the code. In fact, it shares the same I2C line protocol as the real time clock IC (DS3231). So far, I never had a single reading error out of many thousands read cycles.

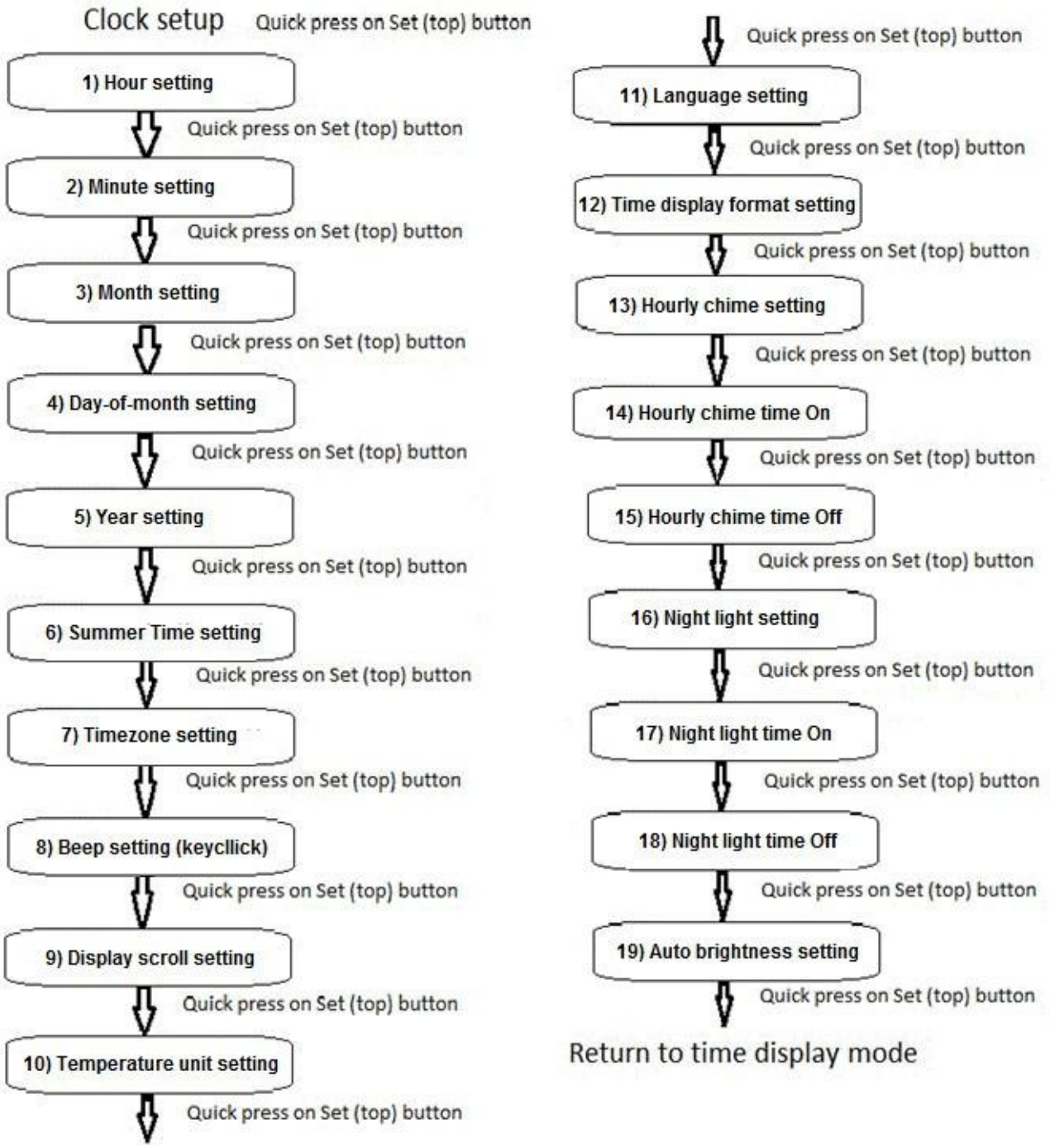
Infrared Sensor

Support for an infrared sensor to receive commands from a remote control has also been added in the code. Since one of my Green Clocks will be installed in a relatively high location on a wall, remote control becomes very handy since access to the clock's buttons will be difficult (Note: if you use a different remote model than mine, you'll have to make the protocol analysis in order to properly integrate it in the code. You may want to take a look to the Pico-Remote-Analyzer utility in one of my repositories on GitHub.)

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 passive piezo with an external signal. By changing the frequency of this signal, the audio frequency generated will change and it becomes possible to 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 scan through all settings.
- The Middle button is called “Up” button since it is primarily used to increase values (for example: increase hour value, increase minute value, etc).
- The Bottom button is called “Down” button since it is primarily used to decrease values (for example: decrease hour value, decrease minute value, 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). Often, 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.

If the Firmware version 10.00 web page has the ‘Short “Set” key sets:’ mode set to ‘Date and Time’, then only a short press of the “set” key will enter the clock setup mode (and a longer press will be needed to enter the alarm setup 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 may:

- Wait for a timeout (20 seconds without pressing a button on the clock – see next paragraph).
- Quick press the “Set” (top) button many times to scan through 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 to the clock settings and also to the flash configuration during the next date scrolling (if changes made are part of the flash configuration).

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 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. This IC is backed-up with a battery and will keep the accurate time in case of a power failure.

NOTE: When setting the hour value, if you cross the point when Daylight Saving Time (DST) changes from Summer Time to Winter Time (or back), the clock may automatically change hour value when you get out of setting mode (assuming you configured the clock for automatic handling of DST).

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. Also note that if you changed the setting of the minutes, the seconds will automatically 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.

NOTE: 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.

NOTE: When setting the month value, if you cross the point when Daylight Saving Time (DST) changes from Summer Time to Winter Time (or back), the clock may automatically change hour value when you get out of setting mode (assuming you configured the clock for automatic handling of DST).

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.

NOTE: 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 14th to March 31st, day-of- month will not allow you to go higher than 28 (or 29) as long as the month is set to February.

NOTE: When setting the day-of-month value, if you cross the point when Daylight Saving Time (DST) changes from Summer Time to Winter Time (or back), the clock may automatically change hour value when you get out of setting mode (assuming you configured the clock for automatic handling of DST).

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.

NOTE: 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 you would expect if you decrease or increase year value. Also, since Firmware Version 6.00, those first two digits are saved to flash memory. So, in case of power outage, when the power goes back On, the high part of the year (first two digits) will keep the value assigned. This will help if you plan to install a Pico Green Clock in the next Electric De Lorean, either to go back in 1845 or to go forward in 2187 - smile.

NOTE: When setting the year value, if you cross the point when Daylight Saving Time (DST) changes from Summer Time to Winter Time (or back), the clock may automatically change hour value when you get out of setting mode (assuming you configured the clock for automatic handling of DST).

6) Daylight Saving Time (DST)

NOTE: Daylight Saving Time (DST) / Normal Time are also called, in some countries: Summer Time / Winter Time and / or Spring Forward / Fall Back.

While referring to the Clock Setup diagram above, press the “Set” (top) button until you reach the “Daylight Saving Time Setting” step.

"ST" (for "Summer 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). So if your country does not use Daylight Saving Time or if you want to manually handle DST, simply configure "0" as the "ST" value.

Refer to Appendix L at the end of this User Guide to find the setting corresponding to your country. (You may want to write this setting here for future reference: ____).

A value other than zero means that the clock supports DST, following the specifications of the selected country as shown in Appendix L.

Once configured, the clock will automatically determine if DST must be active or inactive, depending on current setting (country) and current date and time. It will also change the time automatically as described in Appendix L.

Some countries require the Timezone to be properly set for DST to be correctly handled (this is the case for European Union, in particular). So, make sure to setup the Timezone correctly (see next section).

NOTE: If you change the DST setting, it may happen that the clock automatically changes the time. It will be the case if the current status of the DST kept in flash memory is different than the one you just configured.

NOTE: I would like to hear from users from Chile and Lebanon to know if my understanding of DST setting is correct and if the clock properly handles DST in those countries.

NOTE: Keep in mind that if you prefer to manually adjust DST you're only two (2) "button press" away from a one-hour time change:

- To increase by one hour: One quick press on the "Set" (top) button, then one quick press on the "Up" (middle) button.
- To decrease by one hour: One quick press on the "Set" (top) button, then one quick press on the "Down" (bottom) button.

7) Timezone Setting

While referring to the Clock Setup diagram above, press the "Set" (top) button until you reach the "Timezone Setting" step.

"TZ" (for "Timezone") 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. It goes from 0 to 14 and then -12 down to 0

Detailed explanation of Timezone is beyond the scope of this guide but those interested may easily find information on Wikipedia.

Basically, you must enter the difference between UTC and your local time. Again, for those not familiar with this, it can easily be found on the Internet.

Be aware of something important: if your country supports Daylight Saving Time, your Timezone will change depending on the period of the year. When you cross the Summer Time change point, you change your local time (usually by increasing by one hour), but the Universal Coordinated Time ("UTC") reference "does not change". So the Timezone will change (by a value of +1). The same happens when crossing Winter Time. The Timezone will change (by a value of -1).

So, for example, if you adjust Timezone to be "-5" while you are in "Winter Time", it means, for example that when it is 18h00 UTC, it is 13h00 at your local time. Your local time is "-5" hours from UTC Time (18h00 – 5 = 13). However, during Summer Time, when your local time has changed to add one hour, Timezone will become "-4" (-4 hours), since one hour has been added to your local time while UTC time reference has not changed.

Once the user has setup the clock, the Timezone will automatically be taken care of by the clock. However, if ever there is a need to set the clock, this concept must be understood.

NOTE: As mentioned above, Timezone is required by the clock DST algorithm to properly handle automatic DST support for some countries. Timezone is also required for NTP support ("Network Time Protocol"). NTP allows the Green Clock to automatically synchronize its time over a Wi-Fi (wireless) connection with an Internet time reference server to prevent a drift in time over long period of time (although the real-time clock integrated circuit in the clock seems to do a great job at keeping the time).

8) 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" (beep) 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.

9) 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 / or other information that may have been configured by user in the source code) every five (5) minutes. (The setting can be "On" or "Off". Changing the five (5) minutes period is a compile-time option).

NOTE: Beginning with Firmware 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: Current scrolling cycle will be skipped if the clock is not in "Time Display" mode (for example, if user is setting up the clock - or an alarm, or a timer – when scrolling time is reached).

NOTE: Beginning with Firmware 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).

10) 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 instead of being changed with a press on the middle button.

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

11) 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 may be changed by pressing either on the "Up" (middle) or "Down" (bottom) button until the correct target language shows up on the right of clock display.

Languages supported for now are:

"CZ" – Czech "EN" – English "FR" – French

"GE" – German (basic translation only)

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 also other messages that change depending on Language Setting.

12) 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 may 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.

13) 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 may 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 configuration settings 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 intuitively expect), we assume that we want the sound to be heard *after* the Chime Time On and *before* the Chime Time Off.

NOTE: The Hourly Chime Setting (On, Off, OI) also applies to the sounds generated during Calendar Events.

NOTE: If the user has installed a passive buzzer, a jingle will also be heard (close encounter of the 3rd kind or another one configured by user) after the usual hourly chime.

NOTE: If, in some special occasion, your life cycle is “out-of-usual-life” (and you need to sleep during the afternoon), see the “Silent Period” (“CD” button) on the remote control for a way to easily adapt to the situation.

14) Chime Time On

NOTE: Even if you selected “Hourly Chime On” or “Hourly Chime Off” in the setup section 13 above, 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 selected clock parameters.

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 sounds for the Calendar Events also comply with the Chime Time Settings rules.

NOTE: The Hourly Chime Setting must be set to “Daytime” (that is: “OI” for “On, Intermittent” on the clock display while in setup mode) for the Chime Time On Setting to take effect.

NOTE: The discussion in this section 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) Chime Time Off

NOTE: Even if you selected “Hourly Chime On” or “Hourly Chime Off” in setup section 13 above, 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 selected clock parameters.

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 section. Note that the sound for the Calendar Events also complies with the Chime Time Settings (that means that Calendar Events will also sound at 21h14 and 21h44 – in our example – and also the half-hour light chime at 21h30).

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 sounds associated with the Calendar Events will also comply with the same On and Off times. (Note: For Calendar Events, Chime Time Off represents the last hour period during which Calendar Events will sound. So, if set to 23h00, Calendar Events will sound at 23h14 and 23h44 since xxh14m and xxh44m are the minutes when Calendar Events are verified). The same applies to the half hour chime. It will sound for the last time at 23h30 in the last example.

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 while in setup mode) for the Chime Time Off Setting to take effect.

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.

16) 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 "Automatic"). 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 clock push buttons. You can watch the LEDs during the power up sequence of the clock since they will be turned On for a few seconds 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, in fact, very 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 the nighttime period specified. 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 "Automatic". 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 between On and Off. As opposed to the clock display brightness, however, there is no hysteresis for the Night Light logic algorithm and they will react without delay to a change in ambient light (except if the change occurs in the "twilight zone".

The Night Light Setting (ON, OF, OI, AU) are run-time options, along with starting and ending time (see next two settings).

NOTE: When “Auto-brightness” setting is enabled, keep in mind that the white LEDs (Night Light) will follow the clock display brightness. That is, if the clock LED 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 and Night Light Setting is set to “OI”, the white LEDs will eventually comply with the Night Light Settings in the next few seconds…

17) Night Light Time On

NOTE: Even if you selected “Night Light On”, “Night Light Off” or “Night Light Auto” in the Night Light Setting section 16 above, 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 selected clock parameters.

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 section.

NOTE: The Night Light Setting must be set to “Nighttime” (that is: “OI” for “On, Intermittent” on the clock display while in setup mode) for the Night Light Time On setting to take effect.

NOTE: The Night Light may also be used as a discrete and silent warning in cases where an alarm is overkill. By setting the “Night Light Setting” to “OI” and adjusting the “Night Light Time On” to a specific time, you may then simply take a look at the Night Light to see if you reached (or passed) the “target time”.

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) Night Light Time Off

NOTE: Even if you selected “Night Light On”, “Night Light Off” or “Night Light Automatic” in the Night Light Setting section 16 above, 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 selected clock parameters.

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 the previous section above.

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 while in setup mode) for the Night Light Time Off setting to take effect.

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.

19) 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. As opposed to most other settings that we went through so far, there is no "visible setting" blinking beside "DIM" on the clock display. Instead, user must take a look at the "AutoLight" indicator at the bottom left of the clock display.

The setting can be changed by pressing on the "Up" (middle) or "Down" (bottom) button. If the "AutoLight" indicator is On, it means that the clock display brightness will change automatically, depending on ambient light. It will be bright if the ambient light level is high and it will be dimmed if the ambient light level is low.

Since Firmware Version 5.00, the algorithm behind the Auto Brightness function has been changed to add a hysteresis. The clock LEDs matrix brightness will now change more slowly than what it used to do before. The display was often changing intensity while some object was moving quickly in front of the clock or for a similar reason. Brightness change has now a greater latency, so don't be surprised when you turn Off the room light in the night. It will take many seconds (up to two minutes) 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).

Since Firmware Version 8.00, the clock display brightness is controlled by a Pico's pulse width modulation (PWM) signal. This allows for a much smoother handling of the clock brightness. Whereas there were 5 to 8 different levels of brightness before, there are now a few thousands levels of brightness!! When ambient light level changes, the clock automatically adjusts its display brightness every five seconds (assuming the Auto Brightness option is On). A hysteresis prevents rapid changes of display brightness when an object briefly moves near the clock sensor).

NOTE: As explained, when set to “Auto Brightness”, the clock display will be more or less bright depending of the ambient light intensity. It is interesting to know that when switching to “manual mode” (as opposed to Auto Brightness mode), the clock display will remain with the current intensity. So, if you want the display to remain very bright in manual mode, either change the value from Auto Brightness to manual mode during the day, when the ambient light is very high (or alternatively, use a flashlight to beam the light sensor, just above the USB connector on the right side of the clock) and then change the setting from Auto Brightness to manual mode when the display has reached a bright status. In a similar way, if you want the display to remain very dim in manual mode, either change from Auto Brightness to manual mode during the night, when the ambient light is very low (or alternatively, use your hands or something else to cover the light sensor, just above the USB connector on the right side of the clock) and then change the setting from Auto Brightness to manual mode when the display has reached a dim status.

NOTE: When the brightness reaches its lowest value, be aware that the clock display is **_very_** dim. The algorithm has been done this way so that the clock display does not light up the bedroom too much during the night if it is installed there.

NOTE: Clock Auto Brightness setting can be configured in clock settings as described above. There is also a shortcut allowing to quickly toggle between Auto Brightness and manual mode: simply make a quick press on the “Down” (bottom) button. You will see the “AutoLight” indicator (at the bottom left of the clock display) changing its status to reflect the change.

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.

NOTE: If user installed a remote control, you may use the “Stop” button on the remote control to display the ambient light value, the ambient light hysteresis value, and also the current clock display brightness (in %, from 0% - very dim up to 100% - its maximum intensity).

Alarm Setup

About Alarm Setup

NOTE: The way alarms are handled with Firmware version 6.00 (or greater) is different than with previous Firmware versions. User should read carefully this section to understand all changes that have been made.

There are now 9 independent alarms available in the Pico Green Clock. This has been made possible since the clock can now save all alarm parameters to Pico's flash memory. The alarms will not sound during a power failure, but they will be restored unchanged when the power goes back On (including their status "On" or "Off").

Also, alarms can now be configured for any number of days-of-week. For example, you can have one alarm for Monday to Friday, and a different one (different wake-up time) for the week-end, Saturday and Sunday, etc.

When the alarm time is reached, the alarm will ring during a one hour period, or until user presses the "Set" (top) button, whichever happens first. When pressing the "Set" button, the alarm will be silent, but will remain "On", until the next time alarm parameters are reached, in which case it will ring again. The alarm will sound its "beeps" every few seconds. The number of "beeps" corresponds to the alarm number which has been triggered. If more than one alarm has been triggered, two (or more) "trains of beeps" will be heard, one for each alarm number (and with a different number of beeps, corresponding to each alarm number).

Moreover, if user installed a passive buzzer, the pitch of every alarm will be different. That is, an alarm with a higher number will have a higher pitch (higher sound frequency).

Since there are now 9 alarms available, it is no more possible to turn On the left and / or right LED of the alarm indicator to show if a specific alarm is On or Off as was possible with previous Firmware versions, when only two (2) alarms were available. Instead, when at least one alarm is On, the alarm indicator is turned On. User must check alarm settings to determine which alarms are On and which alarms are Off.

Given the limited amount of buttons available on the clock and the limited width of the display (number of characters), some tricks have been done for Alarm Settings. User must follow the instructions below, some of them not being as intuitive as for other Clock Settings.

HINT: Since the number of "beeps" corresponds to the alarm number, if it is usually difficult for you to wake up, it may be a good idea to use a "high" alarm number (alarm number 7 or 9, for example). When alarm time is reached, 7 or 9 beeps (depending on alarm number) will ring instead of 1 (for alarm number 1, for example). On the other hand, if you wake up easily, you may prefer to use an alarm with a low number since it is less aggressive.

HINT: If you installed a passive buzzer in your Green Clock, you will realize that the alarm pitch (sound frequency) is lower for alarms with a small number and higher for alarms with a high number. This is something you may want to explore, along with the number of "beeps" (see previous paragraph) to find the optimal alarm for your tastes (and needs).

HINT: Another hint for those who find it difficult to get out of the bed… You may want to program two or more alarms at the same time (with 9 alarms available, this is something now possible!). This way, more beeps will sound and this may be helpful (feature removed in Firmware 10.00). Also, since a press on the "Set" (top) button shuts off all alarms simultaneously, it is not a problem. Another alternative is to program a second alarm one minute later than the first one and a third alarm two minutes later. If ever the first alarm didn't wake you up, chances are the second will (or the third). The downside of this hint is that if you wake up with the first alarm, you will have to come back one minute later to shut off the second alarm by pressing again the "Set" button when the second alarm begins ringing.

HINT: When an alarm is triggered, a text will scroll on clock display indicating "Alarm X" (where "X" represents the alarm number being triggered). You may personalize the text by changing it in the source code before re-building the Firmware. Make a search for the string "(ALARM TEXT)" – without the quotes – to easily find the area in the code to make the change. There is one such Alarm Text for each of the 9 alarms. This is a compile-time feature and can't be changed at run time.

NOTE: Alarm beeps use a circular buffer ("sound queue") to keep track of the sounds to ring and let the clock performs its other duties. As mentioned, an alarm will repeat its "beeps" every few seconds. However, if many alarms are triggered at the same time, it may happen that there are so many beeps to sound that it takes more than those few seconds to beep them all. If ever the case, many beeps may be pending in the circular buffer and it may take some time for the alarm to stop ringing when user presses the "Set" (top) button (the buffer will have to empty first). Since this is a situation that is unlikely to happen in usual life, I didn't spend much time trying to find a solution (other than making the beeps shorter depending on how many alarms are simultaneously ringing). Keep that in mind if ever you want to try / test it.

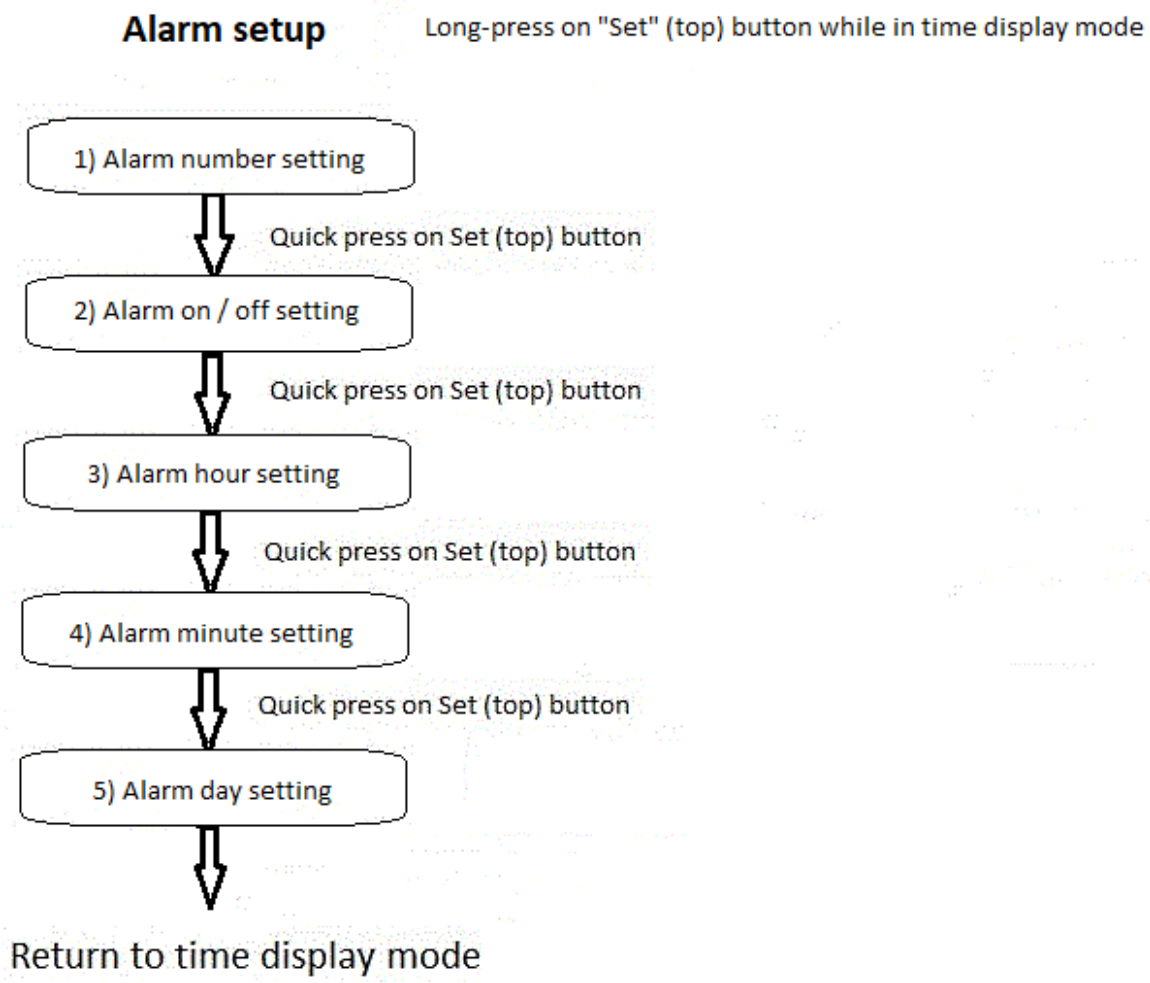
NOTE: Firmware version 10.00 clears the sound queue immediately so the effect mentioned above does not occur. Also, this firmware removes the cascaded alarm operation so only one alarm sounds at any time.

NOTE: If more than one alarm is triggered, as indicated above, number of beeps will indicate which alarm is ringing, and there could be more than one. However, when user presses the "Set" (top) button to make the alarm silent, all alarms currently active (beeping) will be shut off simultaneously. There is no provision to shut off only one of the alarms currently ringing.

NOTE: If a count-down timer alarm is ringing at the same time as an alarm, it will also shut off if / when user presses the "Set" (top) button (see section about the timers in this User Guide for more details). Firmware version 10.00 allows for any button to be pressed to shut off an active alarm.

NOTE: Alarm sound does not comply with Hourly Chime Settings. That is, alarm will sound no matter what the current setting for the 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 clock display during the power up sequence. Make sure you didn't disable this message with an eventual change to the QUICK_START option.



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 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.

If the Firmware version 10.00 web page has the ‘Short “Set” key sets:’ mode set to ‘Alarms’, then only a short press of the “set” key will enter the alarm setup mode (and a longer press will be needed to enter the clock setup mode).

Then, each time you press 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 alarm 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 (except while setting up days-of- week… see below in section 5).

When an alarm number is selected, current alarm values (On or Off, Hour and Minute) will be presented to user as starting parameters.

Alarm Setup Timeout

As indicated in the previous section, if you leave the clock unattended for more than 20 seconds (compile-time option), it will return to the normal 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 to increase alarm number or the “Down” (bottom) button to decrease alarm number. As previously indicated in the user guide, there are now nine (9) independent alarms available in the Pico-Green-Clock).

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.

NOTE: For those who want to work on the source code, alarms are numbered 1 to 9 for the clock user, to be more “human-like”. However, when working on the Firmware source code, alarms are numbered 0 to 8.

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 for the alarm number you want to configure.

Once the alarm number 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 (On or Off) will blink on the display until user presses the “Set” (top) button to select the choice displayed and to proceed with next step.

NOTE: Since Firmware Version 6.00 (or greater), there are now 9 alarms available. So, it is no more possible to turn On the left and / or right LED of the alarm indicator on the clock to show which alarm is On or Off. Instead, when at least one alarm is On, the alarm indicator is turned On. User must check Alarm Settings to determine which alarms are On or Off.

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.

NOTE: Even if it is not configurable by user, there is a "second" parameter in the source code. This value has been set to 29 seconds to be outside of the peak periods (00, 14, or 44 seconds). Even if the System Idle Monitor seems to indicate that the load on the Pico is not that heavy, It has been decided to take a value relatively free of load. Trial and error will show if seconds can be initialized to "0" later on. This also means that the clock display will change the minute and it will take 29 more seconds before the alarm begins to sound. Don't be surprised if minute changes on clock display and if the alarm does not ring immediately.

5) Alarm Days Setting

This is the tricky part…

While referring to the Alarm Setup diagram above, press the "Set" (top) button until you reach the "Alarm Days Setting" step.

Each day-of-week indicator may be in any of those three different statuses:

- If a day-of-week indicator is turned "On", it means that this day is selected ("active") for this Alarm Number.
- If a day-of-week indicator is turned "Off", it means that this day is not selected (it is "inactive") for this Alarm Number.
- The one day-of-week indicator that is blinking is the current "target day-of-week" for which we may decide if we want to turn it On (make it active and add it to the days selected) or turn it Off (make it inactive and remove it from the days selected).

IMPORTANT: <u>At this point in time, there is no way to know if the blinking day-of-week is selected or not for the current Alarm Number</u>. We only know that it is a "proposed" (is a "target") day-of-week since it is blinking..

User has now the following choices:

1) Make a quick press on the “Up” (middle) button to move the target to the next day-of-week. While doing so, user can see if the day-of-week that was blinking before is selected for this Alarm Number (that is, if it is now On or Off). By pressing the “Up” button, the target day-of-week now blinking has changed for the next day-of-week in sequence. However, if user observed carefully, he knows if the new target day-of-week was previously On or Off (before pressing the “Up” button).
2) Make a quick press on the “Down” (bottom) button to move the target to the previous day-of week. While doing so, user can see if the day-of-week that was blinking before is selected for this Alarm Number (that is, if it is now On or Off). By pressing the “Down” button, the target day-of-week now blinking has changed for the previous day-of-week in sequence. However, if user observed carefully, he knows if the new target day-of-week was previously On or Off (before pressing the “Down” button).
3) Make a long press on the “Up” (middle) button to add the target day-of-week (currently blinking) to the current alarm configuration (no matter if it is already active or not). The day-of-week will continue to blink until you change the target day-of-week. There is no way now to confirm if this day-of-week has been added or not (but it *must* have been added). Make a quick press on the “Up” (middle) or the “Down” (bottom) button to change the “target” day number to confirm that the previously blinking day-of-week has been added as expected (it should now be turned On).
4) Make a long press on the “Down” (bottom) button to remove the target day-of- week (currently blinking) from the current alarm configuration (no matter if it is already inactive or not). The day-of-week will continue to blink until you change the target day-of-week. There is no way now to confirm if this day-of-week has been removed or not (but it *must* have been removed). Make a quick press on the “Up” (middle) or the “Down” (bottom) button to change the “target” day number to confirm that the previously blinking day-of-week has been removed as expected (it should now be turned Off).

The most straightforward way to proceed is to select either option 3 or 4 above (depending if the target (blinking) day-of-week must be included in (or excluded from) the Alarm Setting), and then select option 1 or 2 above to confirm that everything is compliant to what we want.

As mentioned previously in this section (and as you can realize), a long press on the “Down” (bottom) button can’t be used to exit from Alarm Setup mode while setting up days-of-week. When your selection of days-of-week is done, make a quick press on the “Set” (top) button to exit from Alarm Setup mode and return to the Time Display mode. If one or more Alarms have been turned On, the Alarm Indicator on the clock display will be turned On.

Alarm configuration will be saved to Pico`s flash at the next data scroll sequence (within the next five minutes if you left the default values) after leaving the Alarm Setup Mode.

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, user enters a start time (in minutes and seconds) and the timer begins the Count-Down, toward zero. When counter reaches zero, it beeps a ‘Timer Alarm”.

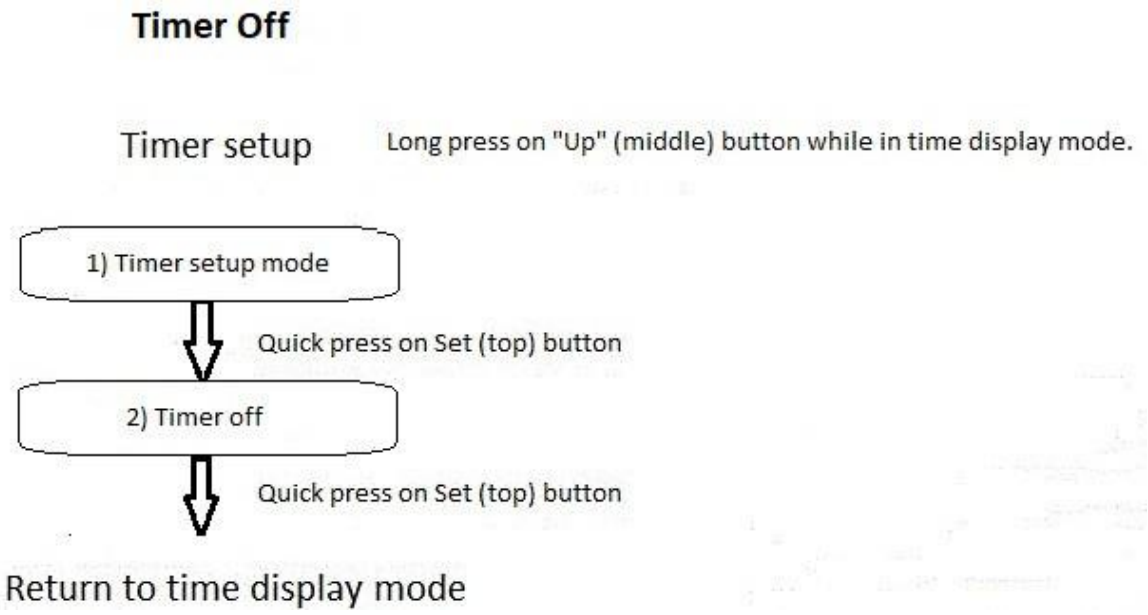
When used for “Count-Up”, it is different since we logically expect a Count-Up Timer to start from zero. So, when started, the Count-Up Timer begins counting up from zero until it is stopped by the user. There is no alarm sound associated with the Count-Up Timer.

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 the timer before the time-out period (that is, within the next 20 seconds).

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.

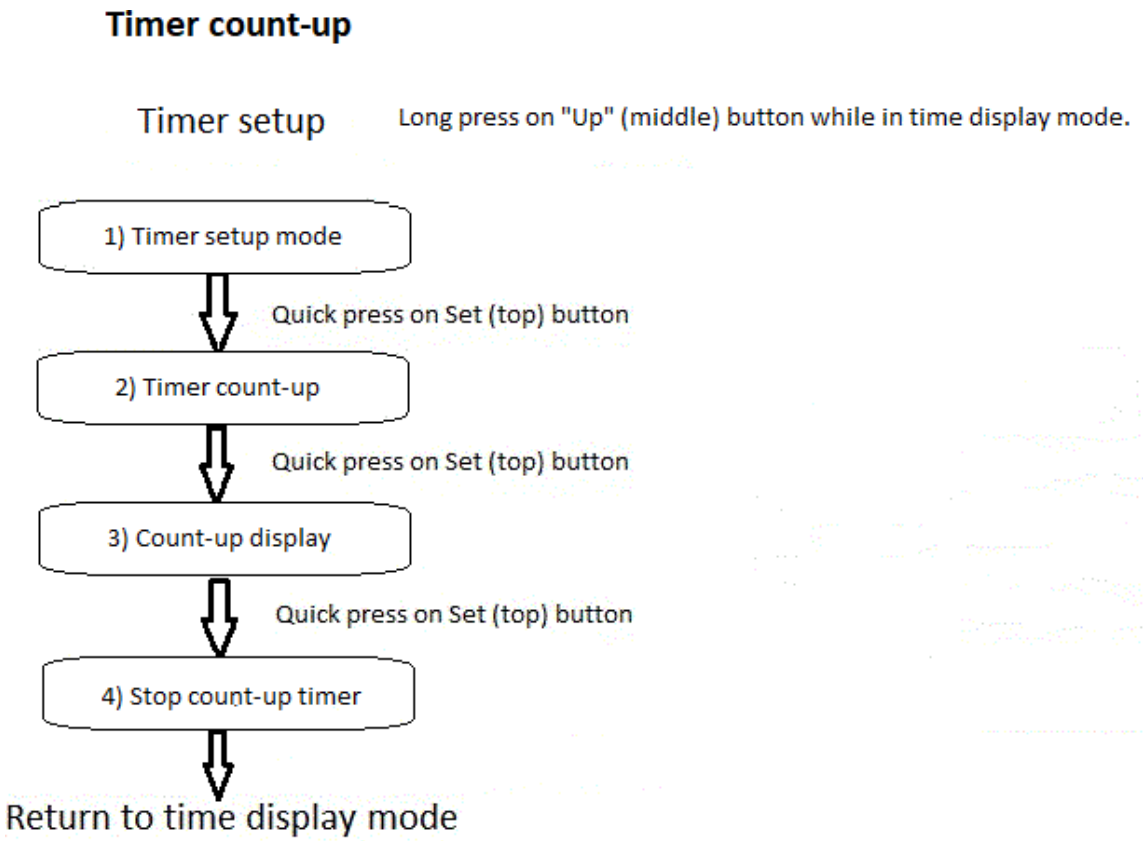
1) To turn Off the timer, refer to the diagram above for “Timer Off” and proceed as follow. First, make a long press on the “Up” (middle) button (longer than one third of a second) to enter the Timer Setup mode.

   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 20 seconds 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 (longer than one third of a second) to enter the Timer Setup mode.

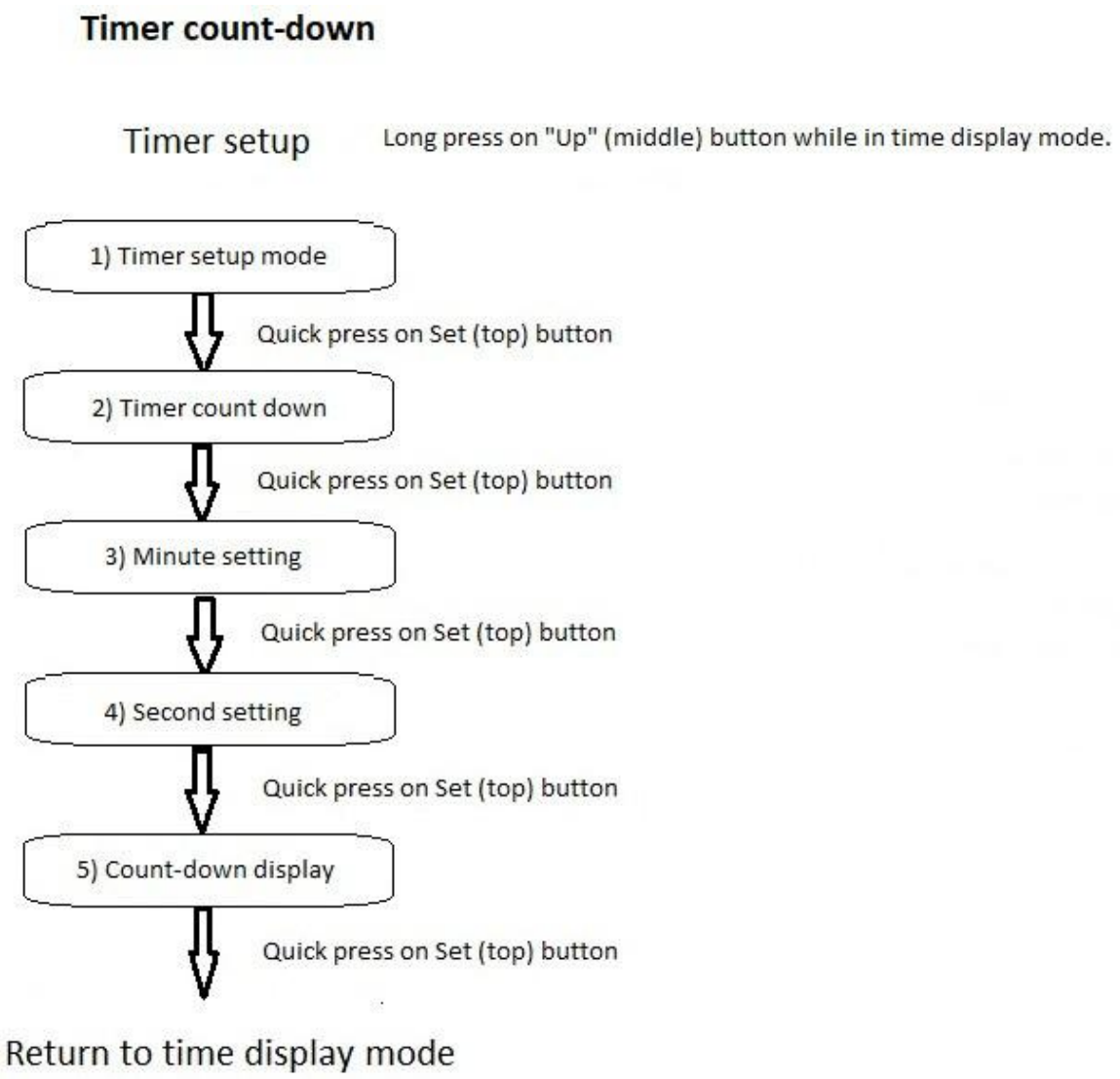
   Then, using the “Up” (middle) or “Down” (bottom) button, press until “UP” (for Count-Up Timer) blinks on the clock display and press on the “Set” (top) bottom to select this choice. Choices are “OF” (for Off), “DN” (for Count-Down Timer), or “UP” (for Count-Up Timer). The corresponding indicator on the 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. Take note that the clock will remain in the Timer Up Display mode until it is stopped by user.
5) When the Count-Up Timer has been stopped, 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 (that is, after 20 seconds).

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 (that is, after adjusting the minutes and the seconds), 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 in the background. The Count-Down Timer will proceed to count until zero (as can be seen by the “Count-Down Indicator” light-up on the clock display). Even if you are in the “Time Display” mode, the Timer 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.

NOTE: The count-down timer alarm will sound every few seconds for a 30-minutes period or until user presses the “Set” (top) button, whichever happens first.

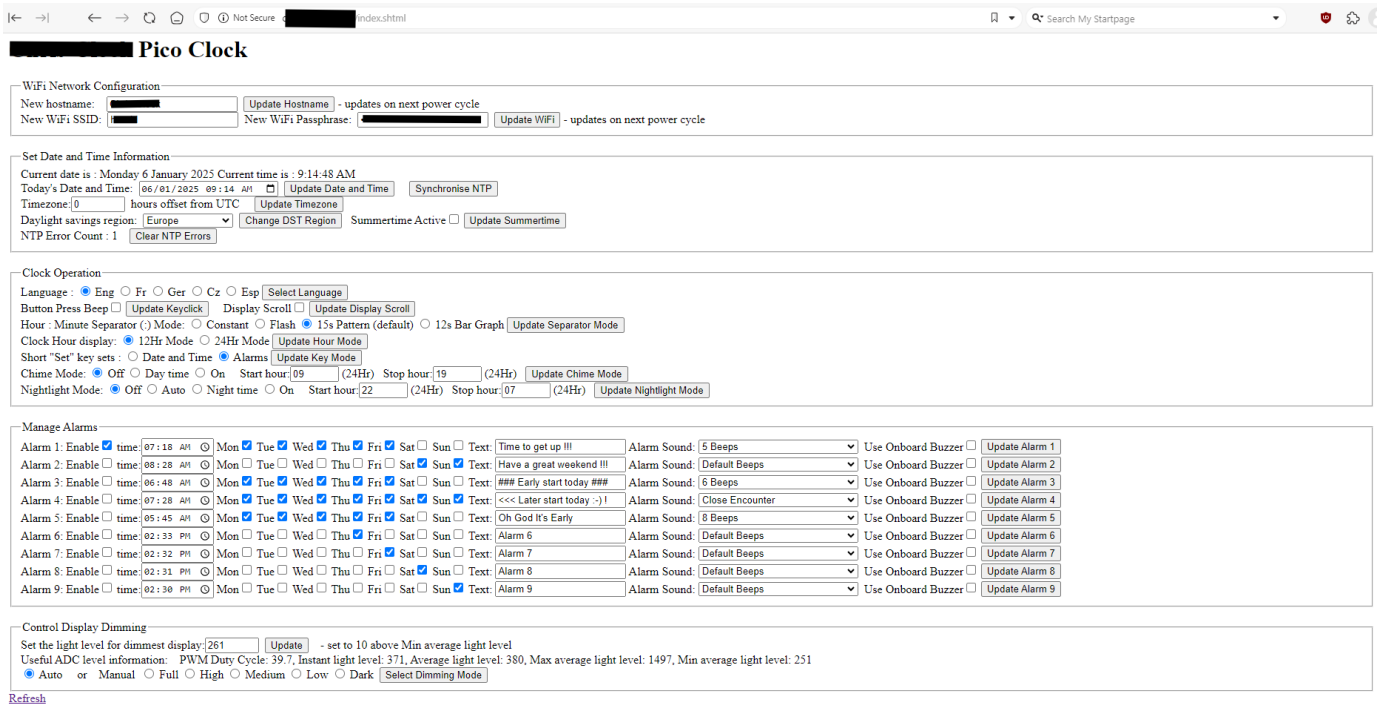
NOTE: If there are “clock alarms” ringing at the same time as the count-down timer alarm when user presses the “Set” (top) button to shut off the count-down timer alarm, be aware that all “clock alarms” currently ringing will be shut off at the same time.

NOTE: 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 Hourly Chime defined hours.

Web Page Configuration

From version 10.00, a web page is present for the Pico-W build versions. This is available at the web address http://<address>/index.shtml where the <address> is either the Pico-W unit's IP address or configured hostname.

The web page looks like:



When the web page is first entered, it is populated with the current clock configuration.

It is split into five main sections. Clicking on the Update buttons applies the configuration changes. There is an update button for each item, and will only update that item Hence, if multiple alarms are changed, clicking on the Update Alarm button will only update that alarm and all other configuration changes are discarded. The web page will automatically update with the new configuration.

From Firmware version 10.01, the Pico-W based units will print out a message every 5 minutes (at a 2 minute 18 seconds offset) on the serial console to say whether the clock is not connected to a WiFi network or the assigned IP address along with the assigned hostname. This is a debugging method to find a lost Pico-W when all other methods fail (such as finding the unit within the network DHCP server's assigned address table).

WiFi Network Configuration

This allows setting of the WiFi network parameters and hostname for the clock. The information is stored in flash memory as soon as a configuration change is applied. The network configuration is read when the clock powers up, so any changes take effect when the clock is power cycled. This allows the clock to be configured or any errors in the WiFi network to be corrected without having to restart the web connection.

Set Date and Time Information

This allows the clock's date and time to be configured from a drop down calendar or typed in directly. The clock's current date and time is displayed to help with this. The clock can be manually synchronized to the NTP server with the 'Synchronise NTP' button.

The Timezone is configured as an offset (between 0 and 23 hours) from the base UTC time zone (London). The daylight savings region is populated with a drop down list of the main entries in the table in Appendix N. The tickbox configures whether summer time adjustment is active or not.

The clock keeps a running count of the number of times an NTP synchronization has failed. This counts to a very large number, but the count can be cleared using the 'Clear NTP Errors' button.

Clock Operation

This section configures the clock's operating modes. It repeats the settings available in the Clock Setup section, so either method can be used to configure the clock – whichever is the most convenient at the time. A change to the clock configuration is applied immediately and also stored into flash memory when the next update event is scheduled (every 5 minutes). If the clock has power removed before the changes are stored, then the clock will power up with an old configuration.

The clock's displayed language only sets the clock display itself, not the web page. The idea is that the web page can use an online translation tool which will be more effective and convenient than changing the web page coded into the clock.

The configuration modes should be self-explanatory based on the Clock Setup section.

Firmware version 10.10 adds the control for the hour / minute ":" separator character flashing modes. These can be one of

1) Static constant ':' narrow character.
2) Flashing ':' narrow character with a 1 second update.
3) Default pattern. This is the ':' flashing every second and changing between upper '·' dot, lower '.' dot, alternating upper '·' then lower '.' dots and ':' character. Each pattern shown for 15 seconds (¼ minute).
4) 5 segment bar graph which steps every 12 seconds (⅕ minute) starting with one dot filled and increasing the dots as the minute progresses.

This is a requested feature to reduce the clock distraction when the pattern changes.

The chime mode 'Day time' mode corresponds to the 'OI' setting and will then sound between the configured start and end hours. The Nightlight 'Auto' mode corresponds to the 'AU' setting and will come on as the light level reduces. The 'Night time' mode corresponds to the 'OI' setting and will come on between the configured start and end hours.

The configuration of the up or down counter via a web page didn't seem to make sense, so these controls have not been added.

Manage Alarms

Here the different alarms can be configured and adjusted. Each alarm has its own update button, so each alarm should be adjusted separately. The alarm can be enabled and disabled easily using the tick box. The time is set to a specific hour and minute of the day (and follows daylight savings time adjustment) and is actually triggered on the clock at a fixed 29 seconds past the minute offset from within the code.

There are check boxes for each alarm day that is easier to configure and see than on the clock itself. The scrolled text that is displayed when the alarm triggers can be configured to any text string. The text is limited to 40 characters by the web page entry box.

The alarm sound is set by a drop down menu and can be changed between the default value (the alarm number beeps), between 1 and 9 beeps and one of 7 different jingle tunes. The alarm will play using the external piezo buzzer if fitted, otherwise it will use the onboard buzzer. The buzzer use can be forced per alarm if this provides a better (or louder) alarm signal.

Control Display Dimming

The display dimming is controlled by a light sensor and an ADC reading. This can vary quite a lot on different units and so requires a certain degree of calibration per unit. It's hard to add this to the button controlled set up menu, so has only been added to the web page.

The web page displays the current light level and a historic maximum and minimum light levels (these values are not retained across power cycles). It also gives the current display PWM duty cycle %. This information can be used to set the programmed value for the code to use for the lowest light level. This value is stored within the non-volatile flash memory so once set it should not need to be updated. It is best to set this when the clock is in its night time environment. A good value to enter is around 10 above the lowest light level reading. This programmed value is used for the manual stepping levels too, but its value is less significant than the auto mode.

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, the outside temperature will scroll on clock display (if an outside temperature sensor (BME280 or DHT22) has been installed and properly configured in the source code.

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 (or greater), the “Auto Brightness Setting” (toggling between steady (or “manual” 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.

For Firmware 10.00 the number of brightness steps has been extended to cycle through 5 fixed levels in addition to the auto mode. These steps are applied immediately, so the code no longer freezes the current clock light level.

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.

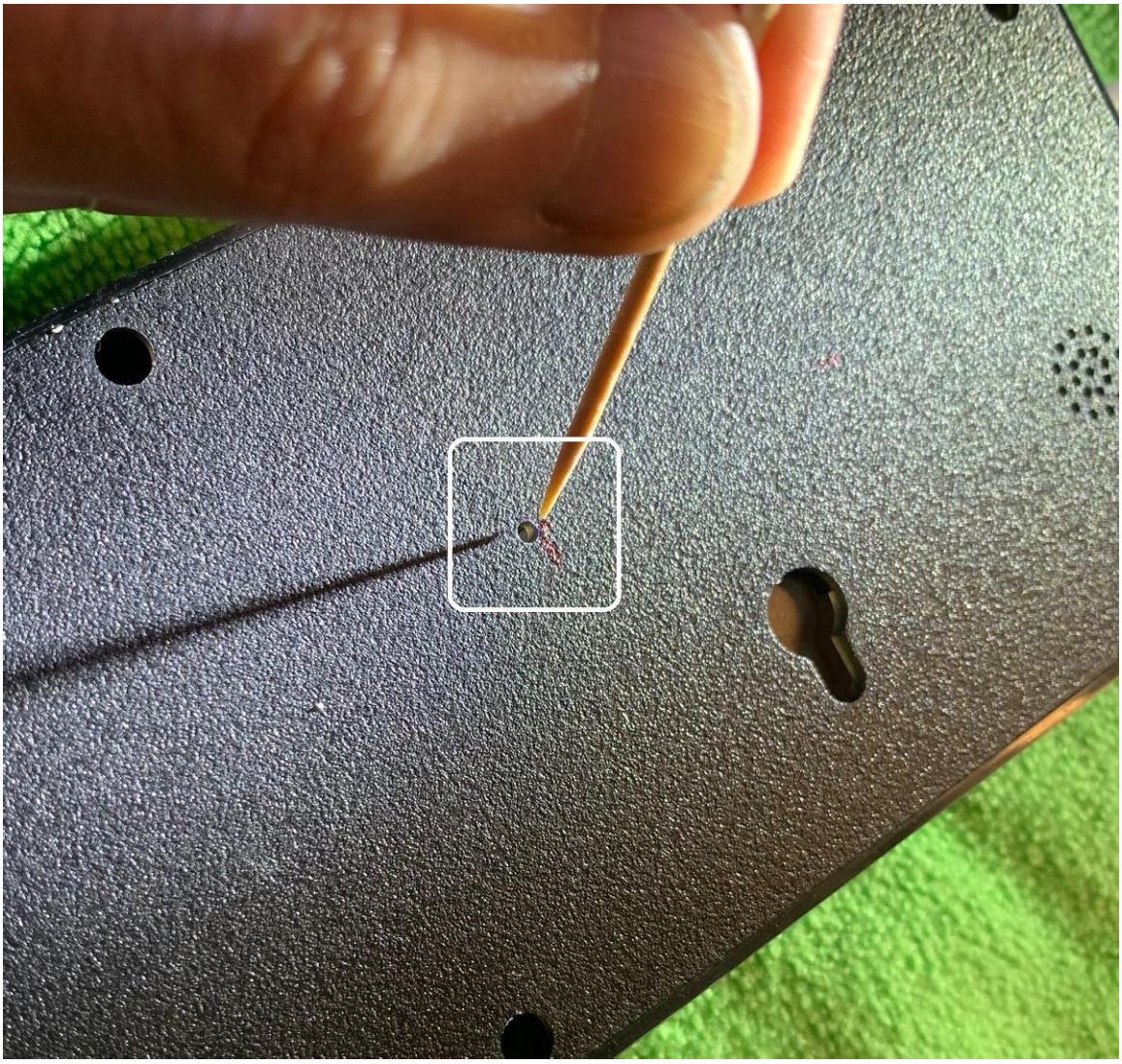
See pictures below as a guide. Take note that these pictures show the back cover on top of the clock case. If you take the back cover alone (without the clock case), you have to remove one mm from the values shown (you can see the case border – 1 mm – around the back cover)..

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).

Location of the BOOTSEL button is the same for the Pico W as it is for the Pico.

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 this User Guide, so far I installed / added four external devices to the Pico Green Clock:

1) An external temperature, humidity and atmospheric pressure sensor (BME280).
2) An ambient temperature and humidity sensor (DHT22).
3) An infrared receiver to support commands sent from a remote control (VS1838b).
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 pulled 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:

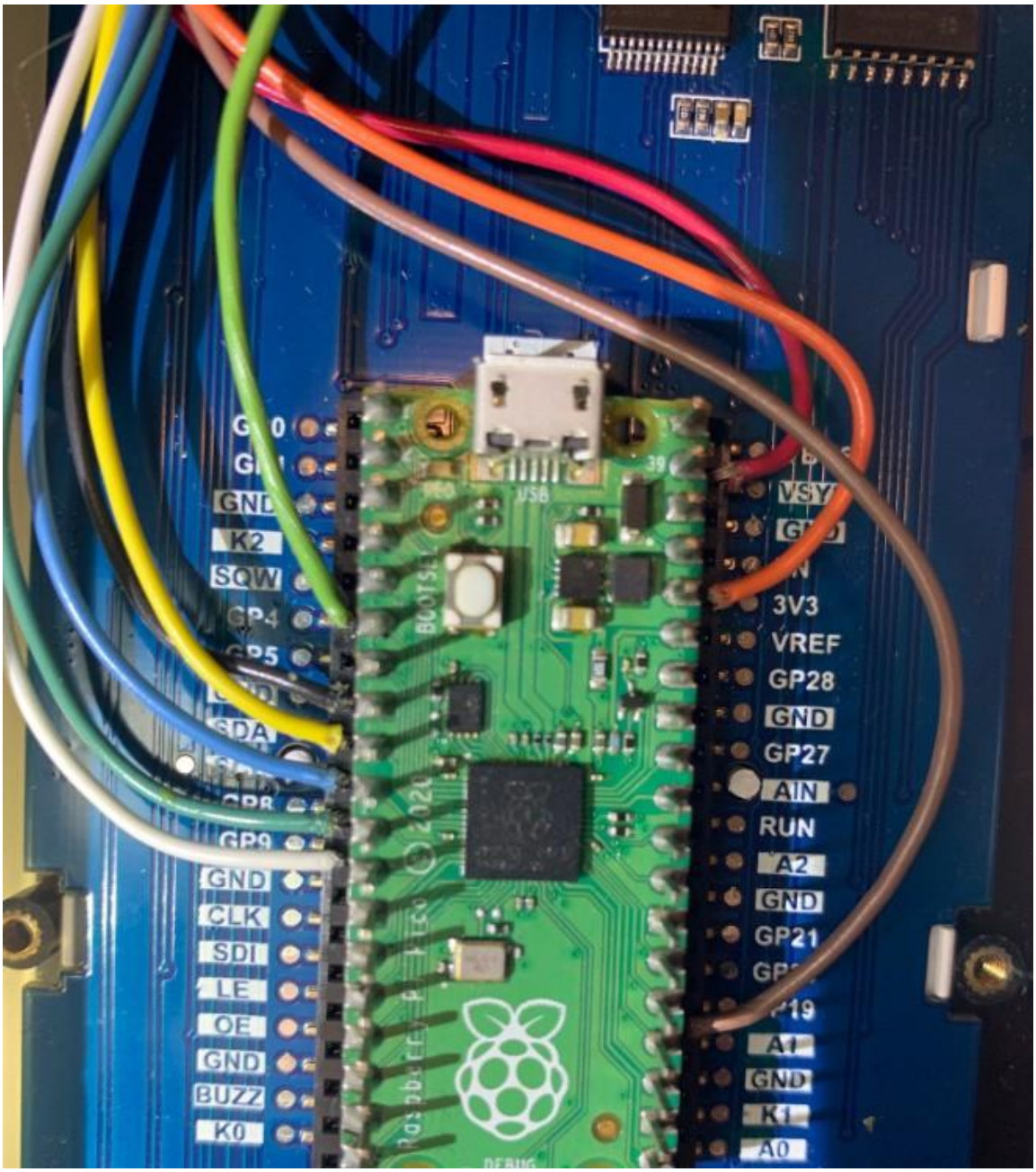
Pico Connection	Pico Pin Number	Cable Color	DB9 Female Pin Number	Useage
Ground	8	Black	5	Ground for infrared receiver; passive buzzer, BME280, DHT22.
VSys +5 volts	39	Red	3	Passive piezo through a 2N2222 transistor.
+3.3 volts	36	Orange	1	BME280, DHT22, Infrared receiver.
GPIO6	9	Yellow	4	I2C SDA (data line) to BME280.
GPIO7	10	Blue	8	I2C SCL (clock line) to BME280.
GPIO8	11	Green	7	DHT22 data line
GPIO9	12	White	6	Infrared receiver signal (VS1838b).
GPIO19	25	Brown	2	Passive buzzer signal.

Note: As you can see, there are 8 cables shown in the table.

NOTE: Since the Pico power comes from the pins under the microcontroller in the Pico Green Clock (and not from the USB connector), VBus does not provide 5Vcc. VSys needs to be used to feed external 5 Vcc.

NOTE: GPIO4 (Pico's pin 6) was used in the first Firmware versions as a UART output to display debug information on an external monitor. A simpler connection is now used, USB-to-USB where the USB port of the Pico is directly connected to a PC's USB port (configured as a USB CDC serial). Refer to the specific section in this User Guide for instructions on how to do it.

NOTE: There is a problem with the Pico W`s USB CDC connection. After Cyw43 initialization, USB CDC communication randomly stops. If you use a Pico W with Network Time Protocol and you want to display debug information on an external monitor, you may want to review the Appendix G and use a UART connection. Another solution is to disable NTP in the code for the time of your tests (so that there is no problem with USB CDC) and restore it after. Obviously, this doesn't help if your needs are about Wi-Fi information.



8-wire cable connected to the Pico double header

NOTE: The green cable connected to pin 6 of the Pico (GPIO4) was used for UART1 communication. UART communication (if used) is now moved to UART0 (GPIO0 and GPIO1). See NOTE before the picture above.

Appendix C – Adding a DHT22 sensor

Special note about DHT22

In the early Firmware versions, a DHT22 device ("outside temperature and humidity sensor") was supported and described in this User Guide. While I developed the Firmware, the DHT22 worked flawlessly when it was connected to the Green Clock with a 20cm cable. However, when this cable was replaced by another one, 3 meters long, I got from 75% to 80% communication errors between the Clock and the DHT22. Needless to say that with such an error rate, the device 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 solution" without success.

I then tried to improve the supporting code by changing the "polling" algorithm to an interrupt-driven routine… without success. I was not sure at this point if the DHT22 had a very poor timing, if the signal was degraded too much in the cable, or if the Pico was too busy with the callback functions and other interrupt service routines to properly handle the DHT22…

While working on firmware Version 7.00, I moved the DHT22 support code from Pico's core 0 to Pico's core 1 and used the polling algorithm. There has been a significant improvement in the communication (less than 1% errors) although the cable used was relatively short (more or less 30 cm). I believe that all callbacks and other interrupts on the microcontroller were causing problems with communication timings. After fine- tuning the code again, I have now 0 errors for a cumulative count of more than 10,000 DHT22 read cycles. I'm not sure at this point if a longer cable would have a significant impact and I'm particularly curious to hear from users that may try using a DHT22 for external temperature reading with a 3 to 5 meters cable…

(For info: I installed a BME280 as an outside temperature, relative humidity and atmospheric pressure sensor. This new device works flawlessly and after a few thousands reading cycles, I got 0 read errors.)

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 its precision is better.

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

NOTE: Be careful about what is written on the small PC board on the picture! The DHT22 pin-out (on the sensor itself) is (from left to right on the picture): 1) VCC 3.3 Volts 2) Data In / Out 3) Not used 4) Ground.

The conditional compile option “DHT_SUPPORT” must be defined in the 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 already include the code. I built my own driver code to support the DHT22 and timeout’s are used, so that the Pico Green Clock will continue working without problem even if a DHT22 device is not actually connected. Since the delay of the timeout’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).

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 Green 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 (2/1530) in the scrolled text, it means that there has been 2 errors on a total of 1530 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.

DHT22 white paper

DHT22 communication occurs on a single line, but it is not the “1-wire” communication standard.

When the line is idle, a pull-up resistor should keep the line at a logic HIGH level.

The CPU originate the communication by sending a low signal for 18 msec. This is considered a "start" signal. Then, the CPU let the line returns to a HIGH level (by the pull-up resistor) and "listen" for a response from the DHT22 (which should take from 20 to 40 usec (micro-seconds).

Once the CPU has sent the "start" signal and return to a HIGH level, the DHT will initiate the answer by sending a LOW level for about 80 usec. Then, DHT22 returns the line to HIGH level for another 80 usec.

DHT11 then sends 40 bits of data. Each bit starts with a 50 usec LOW level, followed by 26 usec HIGH level to send a "zero", or a 70 usec HIGH to send a "one".

When done, the line returns to idle state (HIGH level imposed by the pull-up resistor).

a) MSB is sent first.

b) 5 bytes of data are sent.

c) For the DHT11, bytes 2 and 4 are always "zero" (different for DHT22).

The 5 bytes sent by the DHT11 are as follow:

1) Relative Humidity integral data in % (Integer Part).

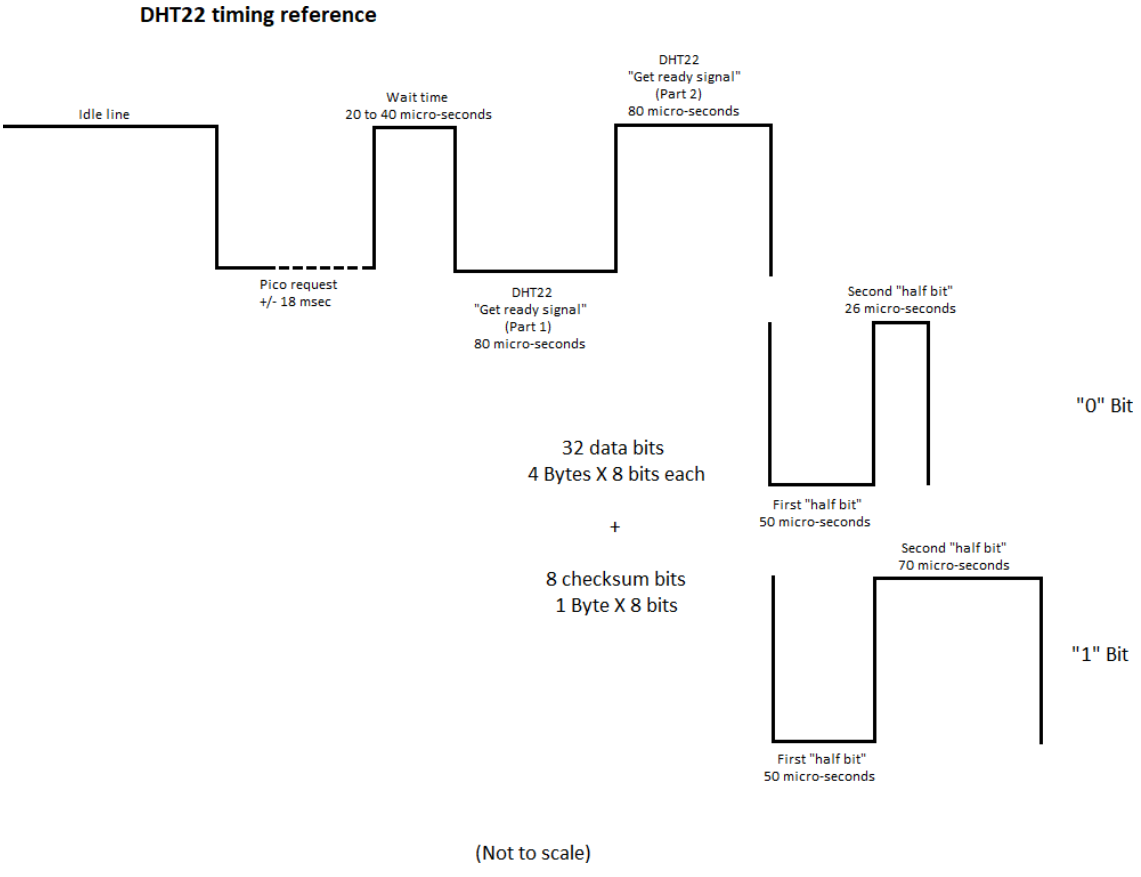
2) Byte equals "zero" for DHT11. Relative Humidity, decimal data in % (Fractional part - for DHT22).

3) Temperature integral in degree Celsius (Integer Part).

4) Byte equals "zero" for DHT11. Temperature, decimal data in % (Fractional part - for DHT22).

5) Checksum (Last 8 bits of {1st Byte + 2nd Byte + 3rd Byte+ 4th Byte}).

NOTE: The probing interval must not be less than 1 second for DHT11 and not less than 2 seconds in order to get valid data.



Appendix D – Adding a BME280 sensor

Introduction

A BME280 sensor can read the temperature, relative humidity and atmospheric pressure. I expected the BME280 to be a good candidate for temperature reading and in fact, after many thousands read cycles, I got 0 errors while reading the BME280 with a 3-meters cable.

Be careful if you buy a BME280. It has been reported by a few Amazon customers that some sellers announce "BME280" devices, but what they received, in fact, was 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 my BME280 about 20US$ (for a single unit) from Waveshare since I wanted to make sure I would receive the right device...which was the case.

In reference to the product on the picture below, be aware that not all cables are used since the module support either SPI or I2C communication. Cables have been provided for both protocols. For our purposes, only 4 cables are used for an I2C protocol connection: Vcc (3.3 volts), GND, SDA (data) and SCL (clock).

BME280 module from Waveshare. You can see the BME280 device itself which is the little square metallic case at the 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, as long as its device address is different than other addresses already used. It means that we do not use an extra Pico's GPIO for this device.

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

- 3.3 Volts supply (Pico pin 36) (green tag block #1)
- Ground (Pico pin 8 – or other ground pin) (green tag block #2)
- SDA data line (GPIO6, Pico pin 9) (green tag block #3)
- SCL clock line (GPIO7, Pico pin 10) (green tag block #4)

Be careful not to feed the BME280 with 5 volts. Even if the device itself 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 Green Clock will work normally without problem. The only impact will be an error message during the clock power-up sequence (indicating that there has been an error while trying to initialize the BME280) and the date and temperature scrolling will also report errors from BME280. Read errors from BME280 are reported in square brackets during date and temperature scrolling without any other indication (for example if you see [3/28338] in the scrolled text, it means that there has been 3 errors on a total of 28338 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.

NOTE: Before it is ready to be used, the BME280 must be initialized by a special software function which is executed only once during the Green Clock power-up sequence. So, if ever you connect / install the BME280 while the clock is already powered On, the BME280 will not work as expected and the clock will continue to report errors. You need to make a power Off / power On cycle with the BME280 connected so that the initialization routine is properly executed.

Appendix E – Adding a Passive Buzzer

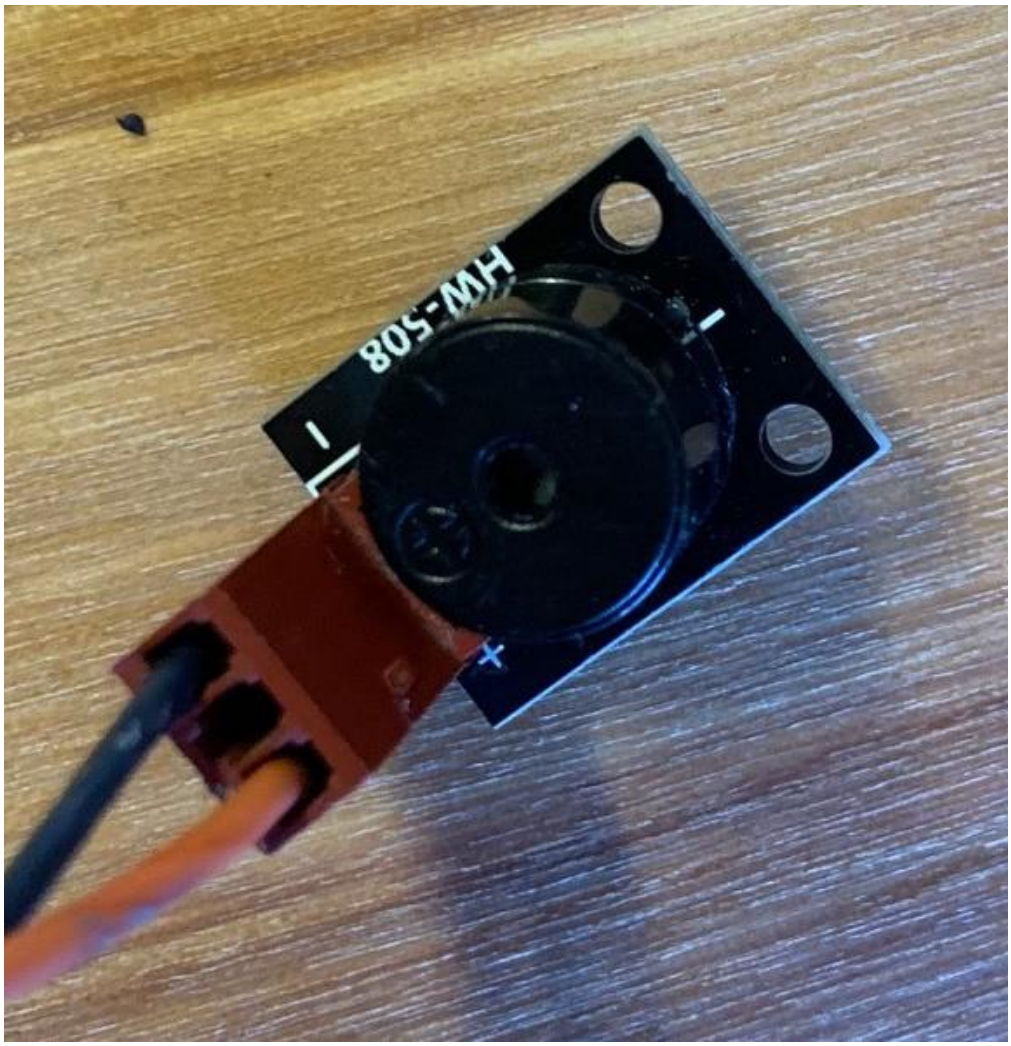
Introduction

The Pico Green Clock provides an active buzzer on its pc board. “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 (or greater), 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). In Firmware Version 6.00 (or greater) a sound queue has been added to optimize and simplify support for this active buzzer.

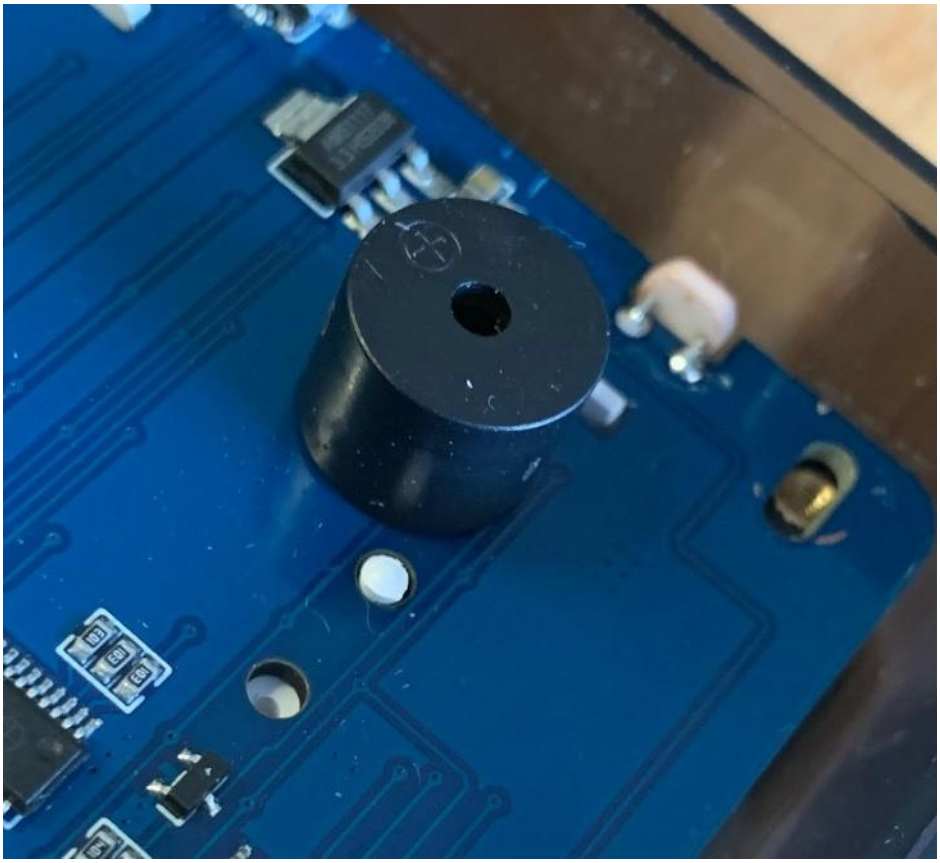
However, I thought it would be interesting to install a passive buzzer with Firmware support to give more flexibility with sounds. Since there are still some GPIOs available on the Pico, I installed a passive buzzer on GPIO 19.

As you can see on the picture below, 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 quick “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 a low resistance value around 8 or 16 ohms.



Passive buzzer to be added by user.

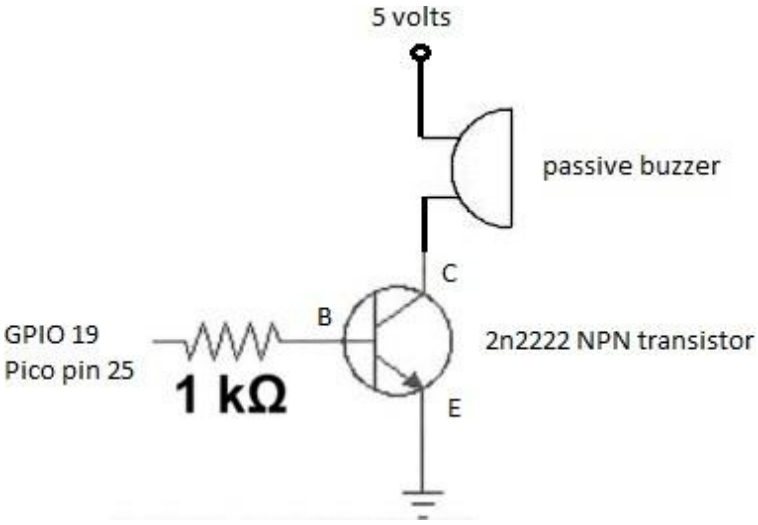


Active buzzer integrated in the Green Clock (on Pico Green Clock pc board).

As opposed to the other members of the Raspberry Pi family (Raspberry Pi 3, 4, etc…), the Pico is a microcontroller 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 queues 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 frequencies, we would most 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, 2N3904 or similar) as a buffer to drive the passive buzzer with 5 volts and getting a louder sound.

Pico Green Clock User Guide



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 on a wall, in the 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

I built an infrared protocol analyzer (based on a Pico microcontroller, 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.

NOTE: You may download the Pico-Remote-Analyzer from one of my repository on GitHub.

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 (data input).

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 / files 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 on the clock buttons) 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 use. However, if you take a look at the Pico-Remote-Analyzer in one of my repository, you will get all instructions required to analyze the protocol for another remote control and implement it to the Pico-Green-Clock.

Make sure to isolate the protocol related code in an include file similar / compliant to memorex.cpp.

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 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. Below is an example on how to assign functions to remote control buttons.

1) 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 as seen on the remote control picture.

2) 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 Green Clock without the risk of the radio becoming crazy by also responding to all remote button presses. I suggest that you do the same if you implement another remote control.

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.

3) 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 atmospheric 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 atmospheric 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 scrolled data the number of errors in the communication sessions with the BME280 on the total number of read cycles. In the example above, 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 the compile-time parameter who determines the frequency for the date and temperature scrolling. If you have a long cable or if there is a motor or other similar device near the cable, you may experience more errors on the long run. Take note that both values (number of errors and number of read cycles) will be reset to zero at each power-up.

4) 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.

5) 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-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 current week (from Sunday, up to - and including - the next Saturday).

6) 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.

7) 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 modes (clock, alarm or timer), in a way similar to what the “Up” and “Down” buttons do on the clock.

8) Stop

Replicate “Down” (bottom) button long press.

When you learned the different setup modes, 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 ambient light information 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 4095 (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. This function will also scroll the ambient light hysteresis (average ambient light value for the last two minutes) and current clock display brightness (from 0% to 100%).

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.

9) 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).

10) 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.

11) 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 (if a passive buzzer has been installed) while randomizing the numbers.

12) 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 also 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 add a 30 minutes silence period to the clock 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 with each button 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).

13) 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 minute – the seconds will change a few times while the display remains in this mode 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.

14) Rewind / Down

The remote button with a “Rewind” symbol on it (also identified with “Dn” written on the remote body) is used to display the System Idle Monitor value. As mentioned in the Green Clock features previously in this User Guide, the System Idle Monitor gives the average number of system loops performed by the software during one second. This average is based on system activity during the last 60 seconds. So, the more processing the system must perform during the interrupt / callback routines and the more processing the system must perform during its endless system loop, the less system loops will be performed per second.

At the time of this writing (Firmware Version 8.00), the average System idle monitor is around 850,000 loops per second (migrating display brightness to PWM allowed a significant increase of idle time).

15) 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.

16) 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 when using the remote control.

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).

When using the remote control, 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 as “short cuts”, 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 / Network Time Protocol
2	Daylight Saving Time
3	Keyclick / Reminders
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 the “Set” button many times to skip to the desired setting, you can press the button “7”, for example, if you want to jump directly to the “Time Display Format” setting (see the quick reference above).

Once in a specific setting section, you can still use the “Set” button to skip to the next setting 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 Chime Time On and hourly Chime 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.

17) Display

There is a reason why I presented the numeral buttons in the previous section, before the discussion about the “Display” button. In some way, the Display function is slightly similar to the Set function seen in the previous section.

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 quick reference 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…)

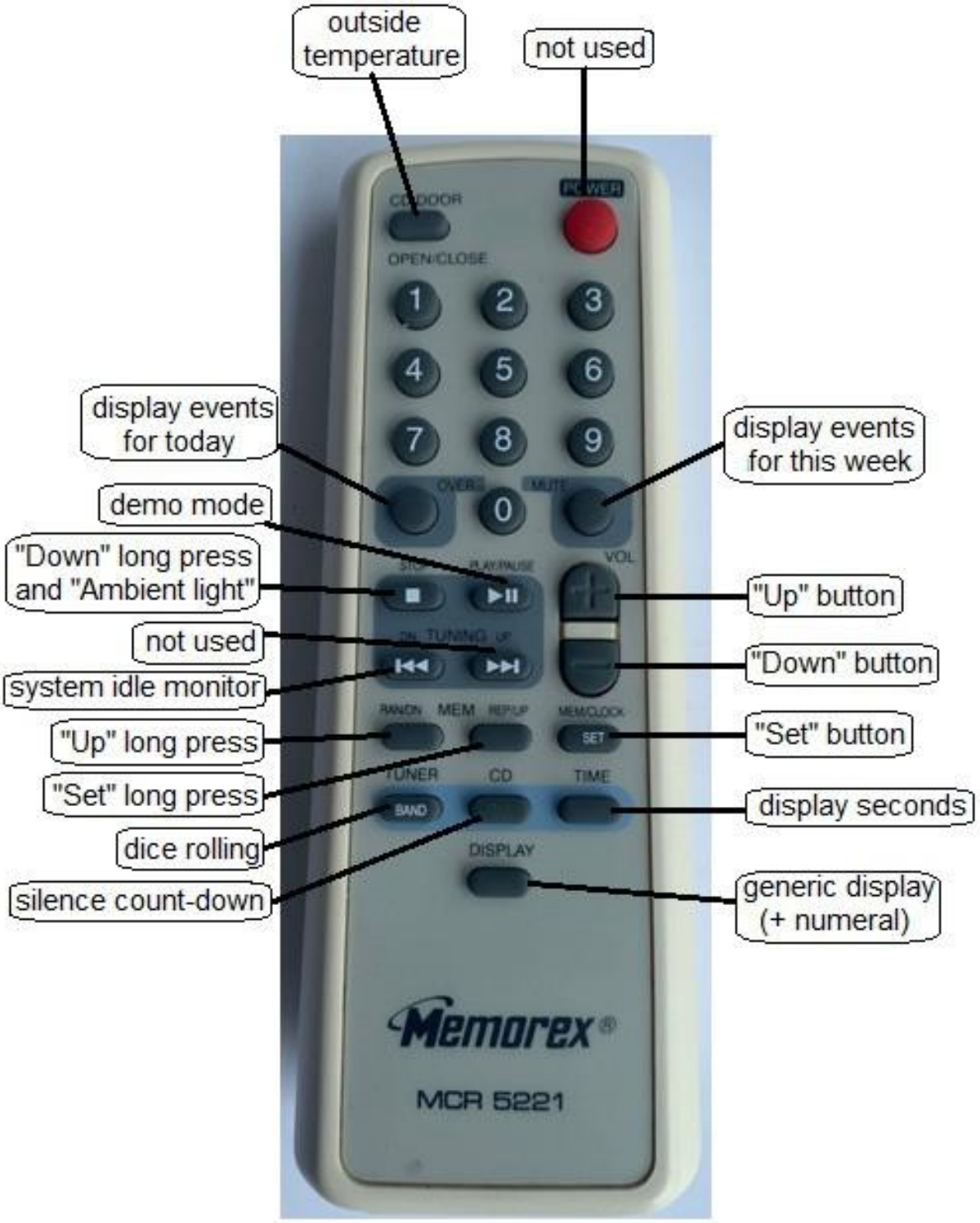
As opposed to the “Set” button, however, you can select only one display function at a time. If you want to display information about “Display Scroll” and also about “Hourly Chime”, you have to press first “Display”, then “4” (“Display Scroll”) and then “Display” again, followed by 8 (“Hourly Chime”).

18) Play / Pause

Finally, the last button that has not been discussed so far is the “Play / Pause” button. This is the button to remember when your friend hacker comes home. It performs many clock functions / demos, and also 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 value, 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 period by 30 minutes each time) when the clock demo is over.

Pico Green Clock User Guide



Appendix G – Configuring an External Monitor

Introduction

If you work on the source code of the Pico 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 terminal emulator to display debug information. Not only such an external terminal emulator with a 80 X 24 screen (or more) is much easier to work with, but it is also much faster to read for debugging purposes than looking at a display scrolling only 4 or 5 characters at a time.

I recommend that you read the whole section before taking any action. The appendix goes from the most complicated setup to the easiest.

VT-101 Terminal Over RS232 Line

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". Without going into details, such a terminal was used to receive plain data from the computer and send back plain data to the computer using a communication protocol called RS232. In fact, many different terminal models existed, but VT-101 originated from Digital Equipment Corporation (DEC) and was very popular. If you have such a monitor, you can use it to display debug information on its screen.

1) We can program the Raspberry Pi Pico so that two (2) specific GPIOs are configured as UART ("Universal Asynchronous Receiver Transmitter"), the IC used to implement the RS232 serial protocol. You can refer to the source code to see how to properly program the GPIOs (make a search for the string "uart"). You can also take a look at the function "uart_send()". The details about serial protocol are beyond the scope of this User Guide.

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 transmits TO the external VT-101 monitor on the version of the interface that I received). You should not need to worry about CTS and RTS handshake, as long as the Pico's UART and the terminal are both programmed to ignored handshake. Also, for now, 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 if your terminal has a DB25 connector).

Be aware that the protocol of the monitor must match the one of the Pico, namely for the speed and the handshake. Refer to the terminal's ("VT-101") user guide for how to set it up.

There is not much to say about sending information to the external monitor. Keep in mind that serial communication speed is relatively slow by today's computer standards. If you send too much information from callback or interrupt routines, you may have side effects on the timings that will trigger more problems than solutions!

Computer screen over RS232

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

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

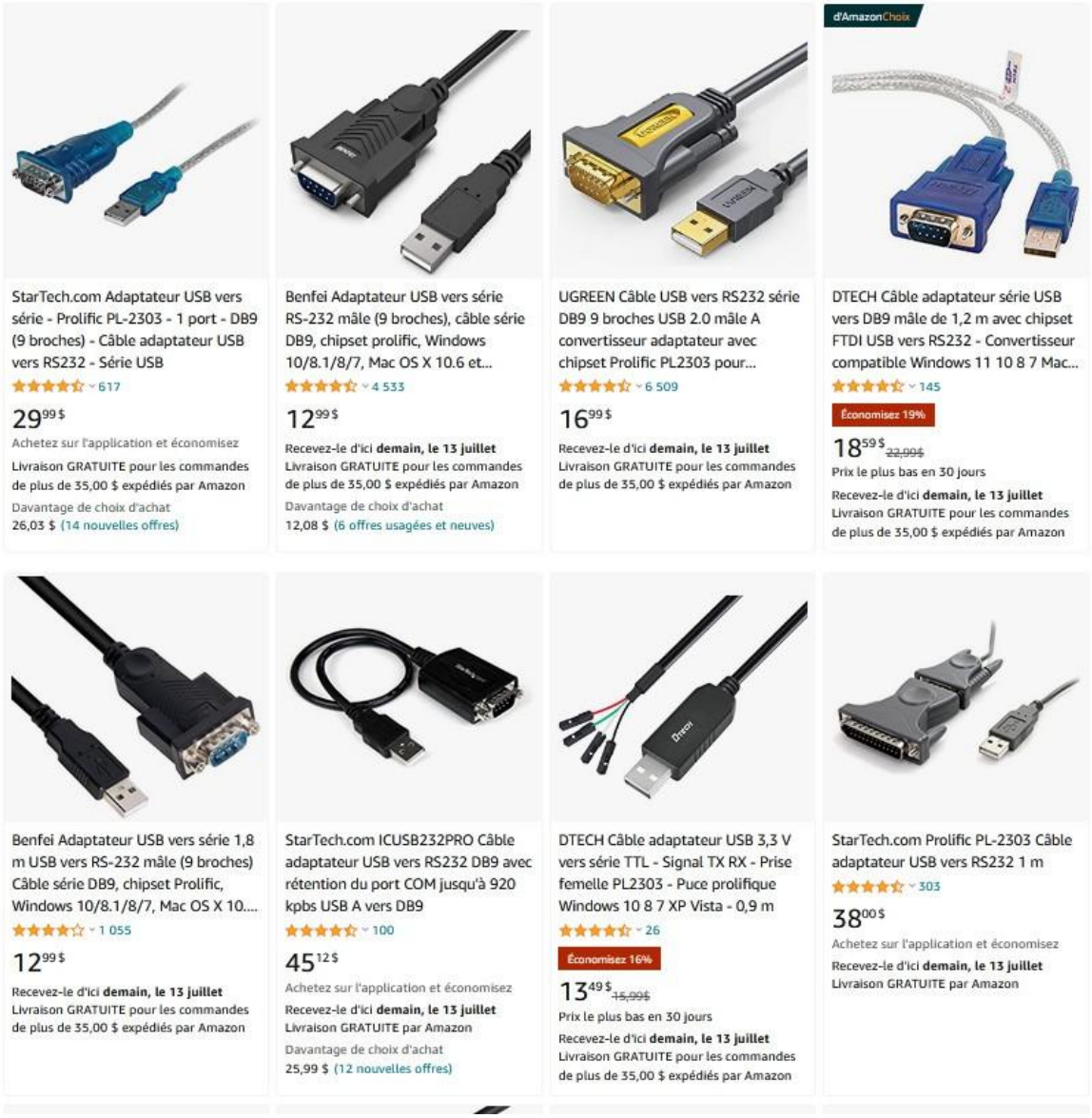
1) All you have to do is following the instructions in the section above ("VT-101 terminal over RS232 line"), but instead of connecting the Waveshare interface to a VT-101, connect it to the computer`s DB9 or DB25 (it should have one if it was equipped with a serial port).
2) You also need to run a terminal 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. TeraTerm is a "new generation" terminal emulation software who works fine.

   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 (Green Clock's Firmware is set by default to 921,600 – N – 8 – 1 but if you use an old PC, you may have to slow it down). Also remember that both ends (the Pico and the terminal emulator program on the PC) must have the same RS232 configuration.
3) Another advantage of using a PC is that most of the communication programs have an option to save all communications to a log file. So, you can save all information sent by the Pico to a file and then take the time to analyze the file content 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 terminal over RS232 line".
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 $, as of mid-2022)

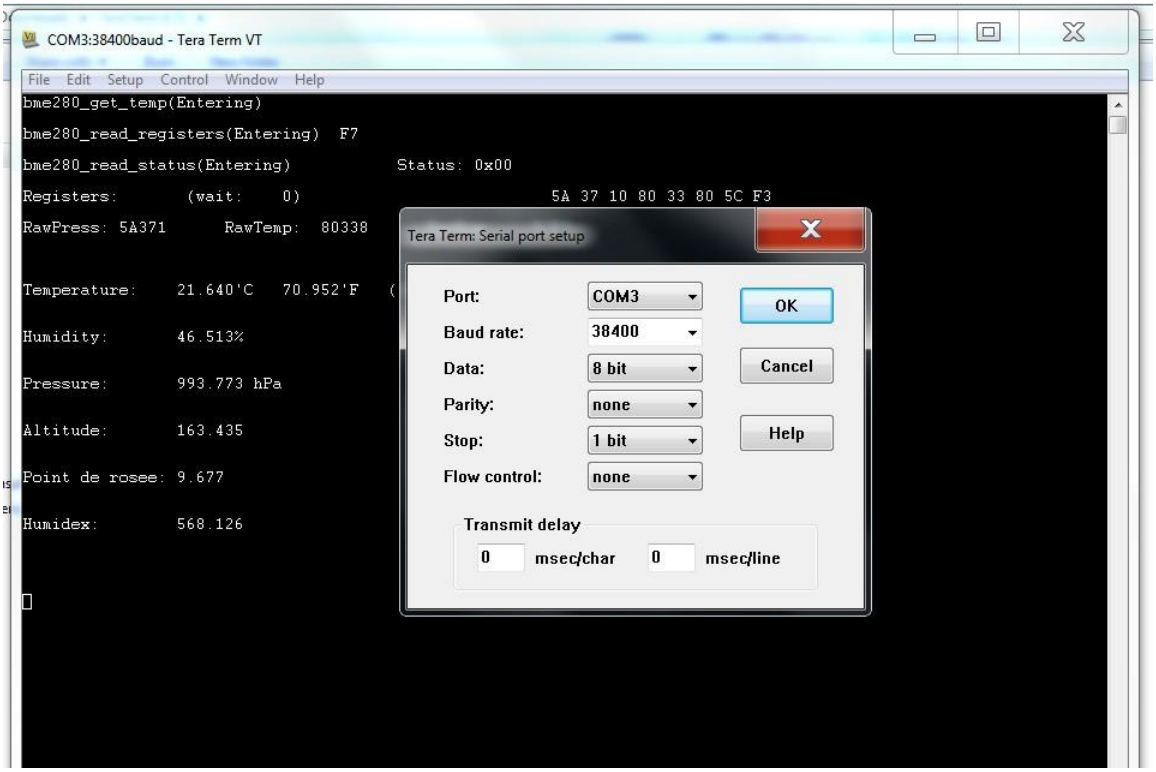
Some adaptors provide a DB9-to-DB25 adaptor. This may be useful if your Raspberry Pi RS232 adaptor has a DB25 connector instead of a DB9 connector. A

DB9 is shown on the grey adapter below. The other, wider end is a DB25 (we actually don't see it on the picture).

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 the USB connection).
4) The final step is to find a terminal emulator program that will work on your computer. I use the popular 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 and a serial port number has been assigned to the serial-to-USB adaptor. Note: on some older Windows versions, you may need to install a special CDC driver for the adaptor to be recognized as a COMx (serial port) device.
6) Then, going to the menu “Setup / Serial port”, you can configure the protocol as you want (921,600 – N – 8 – 1 is the default setting for the Pico-Green-Clock Firmware), but as long as you set the Raspberry Pi the same way, it will work).



NOTE: Once in a while, it may happen that the terminal emulator stops 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 it every time. 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.

Pico to computer, USB-to-USB

The easiest way to communicate between the Pico and your computer is probably with a USB-to-USB connection. In your Pico Green Clock make file (CmakeLists.txt), add the following lines:

Pico_enable_stdio_uart (Pico-Clock-Green 0)

Pico_enable_stdio_usb (Pico-Clock-Green 1)

The first line will stop sending the debug information to the Pico’s UART and the second line will make it sent it through the USB port instead (you can also leave both to 1 if you want).

You can refer to the previous paragraphs to see how to setup a terminal emulator program on your PC and use it with a USB port.

TeraTerm will give an error if you try to connect to the USB port when the Green Clock is not connected… and when the Green Clock is connected, it may be already too late to log the critical information that you want to analyze. For this reason, I added a loop at the beginning of the source code that beeps until the USB CDC communication has been established (or until user presses the “Set” (top) button). While the system waits in the loop, user can start the terminal emulator program. Once the Green Clock is connected, the terminal program will recognize the USB connection and will be ready to receive data as the Pico gets out of the loop and go on with the Firmware. Take a look at the source code for more information. When the firmware is built as a “Release” version, this loop is disabled so that the clock starts without any special user action.

You will see that being able to display debug information on a PC screen is really a plus when working on a microcontroller. However, remember that when working with real- time device, interrupt service routines or callback functions, or some other timing- sensitive stuff, we must be careful about where in the code we put the debug display in order not to interfere with critical Firmware timings…

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 is given after Index at the beginning of this document).

Alarm numbers

Beginning with Firmware Version 6.00, since it is now possible to save the clock configuration to flash memory, nine (9) independent alarms have been implemented. They are saved to flash, so that in case of a power failure, all alarms parameters will be restored to the active clock configuration on power-up.

As was the case when only two (2) alarms were available, alarm numbers have been numbered 1 to 9 to be more “natural” or “human-like” for the clock user. However, in the source code, these alarms are numbered 0 to 8.

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 (which is relatively important for a clock!!). So, the 4 X 7 character set has been left in the code. Moreover, I tried to add the whole displayable ASCII characters in the 4 X 7 character set. 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) A 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 specific jingle identification is given, it will be played when the Calendar Event is scrolled on clock display. As already mentioned, both passive and active buzzers will be heard 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 already 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 (and also variable 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).

Debug Chunks of Code

Since the Pico Green Clock is meant to be a learning tool, many chunks of code that have been used for debugging / learning / analyzing purposes have been left in the code. You may want to take a look at the “DebugBitMask += DEBUG_XXX” section at the beginning of the main() function. By removing the comment symbol from a line, you enable execution of debugging code for the specified functions / algorithm. Be aware that enabling debug sections may have an impact on time critical sections of the code and / or on specific timings. It is assumed that if you use this “feature”, you know what you do.

You will sometimes see code enclosed between comments symbols /*** some code ***/ or "/// some code". Those chunks of code are usually parts that still need rework / optimization or "code that must be reviewed later".

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 if you don't have an external monitor. 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 atmospheric pressure).
2) A DHT22 sensor (for ambient temperature and relative humidity).
3) A passive buzzer (to play jingles).
4) An infrared receiver VS1838b (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 behaves 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 sound queue working inside a callback checking if sounds are pending, 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 interrupts will simply stand there without receiving anything. Once more, there will be no real impact on clock behavior.

Pico W Users, Be Aware

If you want to synchronize the Green Clock using Network Time Protocol ("NTP"), you must use the Pico W, the Pico's version integrating a Wi-Fi communication integrated circuit. You must know that there is a difference in the way to turn On the Pico's LED on the Pico and on the Pico W (you can find information about this on the Internet).

However, an even more important point to be aware of for those who want to work on the source code and use an external terminal emulator to display info is that there seems to have some strange behavior with the Pico W when the CYW43 IC ("Wi-Fi" IC) is being initialized. For some reason, and randomly, USB CDC communication gets cut. I haven't found a work around to this problem so far (and would be happy to know if you did find one!). For this reason, you may want to disable Wi-Fi to do your tests (assuming you want to debug something else than Wi-Fi!) and re-enable Wi-Fi after (you can simply comment out #define PICO_W in the code). Another way to proceed is to use Pico's UART instead of USB CDC and use a logic level conversion to 3.3 volts as shown in Appendix G with the Waveshare interface module.

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

For those wanting to implement another remote control, you may want to take a look to the Pico-Remote-Analyzer in one of my repositories on GitHub.

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 programmers 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 risks and efforts! Main code section will often be modified / optimized and I don't always take care of updating the test code every time to comply with all changes. So, sections of tests that may have worked before may not work anymore after code change. In any case, they may give you ideas / clues for some tests you want to do.

Appendix I – GPIO Used in Waveshare Green Clock

List of GPIOs used in the Waveshare Green Clock:

GPIO number	Direction / Usage	Description
GPIO 0	(Out)	Pico's UART output to an external terminal emulator program.
GPIO 1	(In)	Pico's UART input from an external terminal emulator program.
GPIO 2	(In)	"Set" (Top) button GPIO.
GPIO 3	(In)	SQW (DS3231 RTC IC's square wave – not used).
GPIO 4		Not Used.
GPIO 5		Not Used.
GPIO 6	(I2C) SDA	I2C Data line to read DS3231 real-time IC and also to read BME280.
GPIO 7	(I2C) SCL	I2C Clock line for DS3231 real-time IC and also BME280.
GPIO 8	(In / Out)	DHT22 data line (must be added by user)
GPIO 9	(In)	VS1838b infrared receiver for remote control (must be added by user)
GPIO 10	(Out)	"CLK" Clock line for LED matrix controller.
GPIO 11	(Out)	SDI line for LED matrix controller
GPIO 12	(Out)	"LE" (Latch Enable) for LED matrix controller
GPIO 13	(Out)	"OE" (Output Enable) PWM control of clock brightness
GPIO 14	(Out)	Active buzzer integrated on Green Clock pc board.
GPIO 15	(In)	"Down" (Bottom) button GPIO
GPIO 16	(Out)	"A0" Address line 0 for LED matrix controller.
GPIO 17	(In)	"Up" (Middle) button GPIO.
GPIO 18	(Out)	"A1" Address line 1 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 2 for LED matrix controller
GPIO 23		Used internally for voltage regulation.
GPIO 24		Used internally for voltage regulation.
GPIO 25	(Out)	On-board Pico`s LED (different on Pico W)
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
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[9]:
Bit 0 = Display matrix 1,7
Bit 1 = Display matrix 1,8
Bit 2 = Display matrix 1,9
Bit 3 = Display matrix 1,10
Bit 4 = Display matrix 1,11
Bit 5 = Display matrix 1,12
Bit 6 = Display matrix 1,13
Bit 7 = Display matrix 1,14

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

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

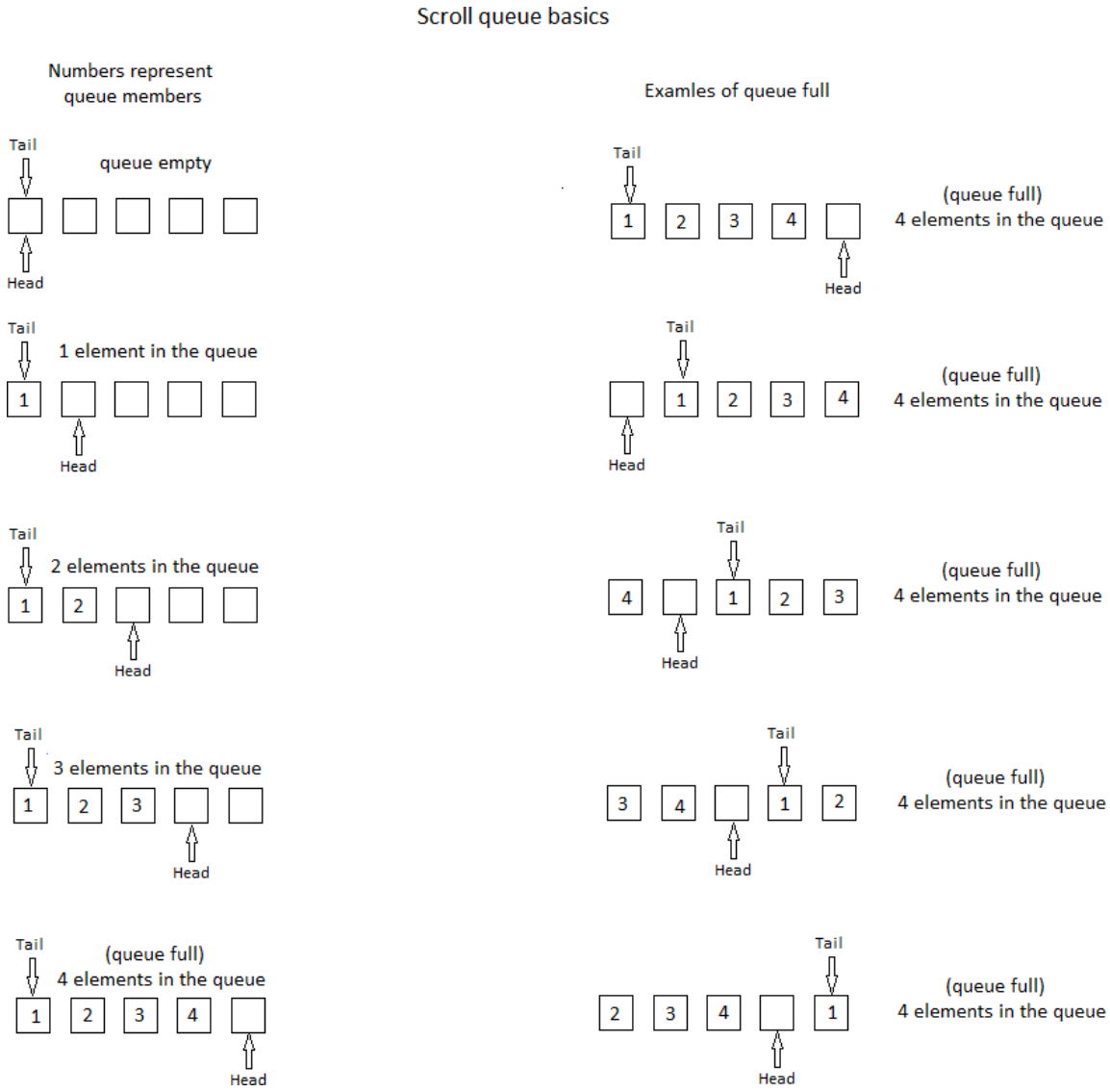
Framebuffer[20]:
Bit 0 = Display matrix 4,15
Bit 1 = Display matrix 4,16
Bit 2 = Display matrix 4,17
Bit 3 = Display matrix 4,18
Bit 4 = Display matrix 4,19
Bit 5 = Display matrix 4,20
Bit 6 = Display matrix 4,21
Bit 7 = Display matrix 4,22

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 other circular buffers ("queues") implemented in the code.

Appendix L – Network Time Protocol

Introduction

Explaining in details what is “Network Time Protocol” (NTP) is beyond the scope of this User Guide. In short, we can simply say that it is a way for the Green Clock to synchronize its date and time (and re-adjust the time kept by its real-time integrated circuit) with a trusted time-reference server over the Internet.

Since this Internet server has no clue where you live (and consequently, in which timezone you are), the parameters “Daylight Saving Time” (or “DST Country”) and “Timezone” must be properly setup on the Green Clock for NTP to work as expected.

Also, be aware that setting up NTP on the Green Clock is a little tedious. Make sure you have some free time ahead to complete the task (more or less 20 minutes?) That being said, your setup will be saved to Pico’s non-volatile memory, so you shouldn’t have to repeat this setup except if you change your Wi-Fi access credentials.

A Pico W is required if you want to use NTP. The “plain” Pico does not contain the Wi- Fi interface circuitry as does the Pico W and which is required for accessing the Internet. However, you can use the Green Clock with a Pico without problem as many users have done so far with the previous Firmware versions. The real-time clock IC in the Green Clock has shown that it is able to keep a very good time precision over long period of time. Moreover, the Pico-Green-Clock repository contains two versions of the Firmware: one for the Pico and one for the Pico W. The repository also contains two versions of the CMakeLists.txt. As you may guess, one is for the Pico and the other one is for the Pico W (see details below: “Two Firmware versions”).

IMPORTANT:

As already mentioned at the beginning of this User Guide, keep in mind that your Wi-Fi credentials (network name and password) will be saved in Pico`s non-volatile memory, in a location that is unlikely to be overwritten even if you upload another program / firmware to your Pico. If ever you give away your Pico to someone else, this person could relatively easily retrieve your credentials from your Pico. For this reason, if you are to dispose of your Pico, you may want to use the Pico-Flash-Utility in one of my repositories to wipe (erase) the whole Pico’s flash memory space.

NOTE: If you are a developer and you plan to work on the source code (particularly on NTP), you may want to review the rules of engagements requested from public NTP servers. The Green Clock complies with those rules and it is important that you remain compliant if you modify the code. I will not assume any responsibility for changes that you may make to the source code.

Two Firmware Versions

The same source code supports both the Pico and the Pico W for NTP support. All you have to do is comment (or uncomment) the #define PICO_W in the source code to enable Pico W (and NTP) support. (Take note that the #define NTP_ENABLE has not been implemented yet and it is the #define PICO_W that controls the NTP support code for now.

So, the developers will work with the same version of source code, with or without NTP support. However, the CMakeLists.txt file is different if you build for the Pico or for the Pico W. The make file must determine the need to include the libraries for the Pico W (Wi-Fi support). Take note that the default "CMakeLists.txt" file in the Pico-Green-Clock repository is for the "plain" Pico. The same "plain" Pico CMakeLists.txt" file is also backed-up as "CMakeLists.txt.Pico". Unsurprisingly, the "CMakeLists.txt" file for the Pico W has been named "CMakeLists.txt.Pico-W". You may copy either of those versions (for Pico or Pico W) over the "CMakeLists.txt" as you see fit for your needs.

The same concept applies to the ".uf2" executables. The default "Pico-Clock-Green.uf2" is the "plain" Pico version, which is also backed-up as "Pico-Clock-Green.uf2.Pico". As for the Pico W executable version, it is named "Pico-Clock-Green.uf2.Pico-W". Copy either the Pico or Pico W version over the ".uf2" file as you see fit to upload to your Pico or Pico W.

Take note that the "plain" Pico Firmware version will work without problem on a Pico W. If ever you play with the code and you want to use the on-board Pico LED, however, you'll have problems since the Pico W requires some specific cyw43 support libraries to turn On / Off the on-board LED. Other than that, I haven't seen any glitch so far to run Pico code on Pico-W.

As for Pico W code, be aware of the problem with USB CDC communication that seems to be randomly interrupted once the cyw43 is being initialized. As mentioned elsewhere in this User Guide, I will really appreciate if someone finds a workaround about that (other than using a Pico's UART instead of USB CDC, which is what I do for now).

Following what is said above, take note, however that the Pico W Firmware version will not work on a "plain" Pico, even if you are ready to accept that NTP would not work. Simply said, the Wi-Fi libraries will crash the Firmware if you run on a "plain" Pico. For those who are curious, you may want to take a look at the difference in code size between the Pico version and the Pico W version. The Wi-Fi libraries do take a lot of space. Developers already began to ask for more flash memory space in the next Pico version to make it for the extra space required for Wi-Fi library support (and we can expect that BlueTooth support will simply make things even worse!)

For Those With a Development Environment

If you have a setup that allows you to easily rebuild the Green Clock Firmware, the easiest way to configure hostname, SSID and password for NTP support is to go in the source code and set the #define for SSID and password at the beginning of the code (search for "MyNetworkName" and "MyPassword"). You must do that the first time you run Firmware Version 9.00 (or 10.00), before the system updates from a previous Firmware version. If you already ran Firmware 9.00 (or 10.00) without setting the two variables, you will need to search for the string "One-time" in the source code and put the two variable settings there. Remember that SSID and Password begins as the 5th character for both variables, on top of the different footprints(".:.:.:.:.:.:" and ".;.;.;.;.;.;.;")

You need to make sure that the CMakeLists.txt is the version for the Pico W (it is backed-up in CMakeLists.txt.Pico-W in the Pico-Green-Clock repository). Then, simply rebuild the code and upload it to the Pico W. The SSID and password will be updated in flash the first time you run Version 9.00 (or 10.00) of the Firmware (you must wait for the first date scroll for the flash to be updated). You can then delete the strings in the source code.

Pico Green Clock User Guide

NOTE: Firmware version 10.01 onwards supports the ability to add the WiFi credentials to a file called “WiFiCredentials.cpp” which can be included in the build file set but not committed to the GIT repository. This then keeps this information local to the build machine. The format for the contents of the file is the same as the code within the Pico-Green-Clock.c file:

```
// /*** One-time FlashConfig writes may be inserted below... ***/
// // NOTE: If you already ran Firmware Version 10.0x, network hostname, SSID and password will not be updated just by replacing the
// //   #define HOSTNAME, #define NETWORK_NAME and #define NETWORK_PASSWORD at the beginning of the source code, instead, to set hostname,
// //       network SSID and password in the code instead of setting them up with the clock, uncomment the six lines below and replace
// //       MyNetworkName and MyNetworkPassword with the proper strings while keeping the surrounding "double-quotes".
// sprintf(FlashConfig.Hostname, ".@.@.@.@.@.@.@.@.@.@.@.@.@.@.@.@.@.@.");               // write specific footprint to flash memory.
// sprintf(FlashConfig.SSID,         ".;.;.;.;.;.;.;.;.;.;.;.;.;.;.;.;.;.;.;.");
// sprintf(FlashConfig.Password,     ".:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.:.");
// sprintf(&FlashConfig.Hostname[4], PICO_HOSTNAME);
// sprintf(&FlashConfig.SSID[4],     NETWORK_NAME);
// sprintf(&FlashConfig.Password[4], NETWORK_PASSWORD);
```

Replace the PICO_HOSTNAME, NETWORK_NAME and NETWORK_PASSWORD text with the required values.

For Those Without a Development Environment

The Green Clock is – before everything else - …a clock! So, I wanted to make sure everyone would be able to configure it for NTP access, even those without technology background and / or development tools. The solution I found is to enter the hostname, SSID (“network name”) and password directly from the Green Clock itself. It will take some time, but with a little patience, you should go through the process easily. See below…

Procedure to Setup NTP on the Green Clock

Make sure you carefully follow all steps below in the order they are listed for the setup to work as expected.

Be aware that there is no way to make a correction if you enter a bad character. You’ll have to start over from scratch! Be careful and proceed slowly. Let’s start!

1) Make sure the microcontroller you use in your Green Clock is a Pico W, which has a Wi-Fi IC on it. The “plain Pico” will not allow implementing NTP. See the picture below to confirm that your microcontroller is a Pico W. The two pictures on the left show both sides of a “plain” Pico, whereas the two pictures to the right show both sides of a Pico W. The easiest way to recognize which one is the Pico W, is certainly the relatively large metallic case on top of it containing the Wi-Fi interface circuitry.

Both sides of a “plain” Pico		Both sides of Pico W	
			

2) Make sure you have access to a Wi-Fi network for Internet access. You may want to test Internet access with a cell phone or another device to make sure you have the exact spelling of the SSID and password. Write down on paper (in BIG letters) the SSID (“network name”) and password for the Wi-Fi access. You will use it later.

3) Download the files (or clone the repository) from the GitHub repository to your working environment.

4) The default CMakeLists.txt in the repository is the one for the “plain” Pico. In case you want to work with the source code later (and rebuild the code), put in place the right CMakeList.txt file:

    cp CMakeLists.txt.Pico-W CMakeLists.txt

5) Do the same with the Firmware executable:

    cp Pico-Clock-Green.uf2.Pico-W Pico-Clock-Green.uf2

6) Transfer the executable to the Pico W installed in the Green Clock (it is assumed that you know how to proceed).

7) The Firmware should start and you should see the time after a few seconds (more or less 40 seconds). Note: if you used previous versions of the Firmware before, you may realize that it takes many more seconds for the Firmware to start. The Firmware with NTP will make many retries to get Wi-Fi access. Since your credentials are not setup yet, those retries take some time (you can see the Pico`s LED through the back plastic case as those retries are performed). When the credentials are properly configured, startup time will be shorter, but still a little longer than with previous Firmware versions as the Green Clock synchronizes its clock with NTP on power-up.

8) Once the clock is up and running, let it runs in "time display mode" until the next date scroll on the display (should be no more than 5 minutes, but there is no problem if you let it run for a longer time). This will allow for the creation of the default clock configuration in flash memory. Note: Even if you already have a previous configuration in flash, it must be updated with more data than before. Your previous parameters will be properly restored, but this step is still required to generate the default extra parameters for version 9.00

9) When at least one date scroll has been done, remove the power from the clock.

10) While pressing the "Up" (middle) button, apply the power to the clock. Note: You will need to keep your finger on the "Up" button for nearly one minute. After many seconds, the clock will sound a few beeps. After hearing *at least 5 beeps*, you can release the "Up" (middle) button. Check the clock display. You will see the credentials (hostname, SSID and password) currently saved in the Pico`s non-volatile memory (probably garbage for now). Then, you should see "Enter HOST:" scrolling on the display. The clock has entered the "set-and-save credentials" for Wi-Fi access and is now waiting for you to enter the HOST (network hostname) followed by WiFi network SSID and then the WiFi password. Take note that if / when your credentials are properly setup, you will see them during this step. However, Hostname, SSID and Password begin at the 5th characters of the string. Don't be surprised if you see other characters (.;.; or .:.:.) before your own SSID and password.

11) At this point, you may want to become familiar with the clock display. Don't worry, there is no timeout and you will remain with the same status for all the time required… The left of the display is blank for now and at the right, you see a capital [A] inside square brackets, which is a proposed ("target") character that you can choose.

12) You need to enter the network name letter by letter (be careful, both the network name and password are case sensitive. So, a capital [A] and a lowercase [a] can't be interchanged). You need to proceed as follow (it is better if you read all this section before proceeding):

- You can press on the "Set" (top) button to select the capital A as the first letter of the desired hostname.
- You can press on the "Up" (middle) button to skip the capital A. The clock will propose the next character (capital [B]) as a potential target character.
- You can press on the "Down" (bottom) button to skip the capital A. The clock will propose the previous letter [@] as a potential target character.

- You can press the “Up” (middle) or “Down” (bottom) button as many times as required until you reach the target character that you need. When you reached the right character, press the “Set” (top) button to select it. When you do that, the last character that you selected will show up at the left of the clock display. This is to remind you of the last character selected since the process is relatively long.
- When reaching the end of all displayable characters, the display will roll back to the beginning and the sequence will start over again.
- Instead of pressing the “Up” (middle) or “Down” (bottom) buttons many times, you can keep you finger on the button. After a few seconds, the characters will change more quickly. After another few seconds, the change will be even faster, so that you don’t have to wait too long. If ever you go further than desired, simply release the “Up” or “Down” button and use the other one to go back (or ahead) and reach the character needed.
- When you are done entering all characters of the network name, press the “Set” (top) button for at least 3 full seconds (it’s OK if you keep pressing longer). This “very long press” concludes entering the hostname. If you check the clock display, you should now see scrolling “Enter SSID:”.
- Proceed the same way to enter the SSID (network name) as you did for the hostname. Again, when done entering all characters, make a “very long press” (at least 3 seconds on the “Set” button to complete the password saving to flash.
- When you are done entering all characters of the network name, press the “Set” (top) button for at least 3 full seconds (it’s OK if you keep pressing longer). This “very long press” concludes entering the SSID. If you check the clock display, you should now see scrolling “Enter password:”.
- Proceed the same way to enter the password (WEP pass phrase) as you did for the SSID. Again, when done entering all characters, make a “very long press” (at least 3 seconds on the “Set” button to complete the password saving to flash.

NOTE: There is no way to make a correction if you enter the wrong character (you will have to start again from scratch). Proceed slowly. It is recommended to mark each character on the sheet of paper you prepared earlier as you enter them in the clock.

NOTE: If you make a mistake while entering the ***password***, you’ll have to start the “set- and-save credentials” setup again from the beginning. However, when asked to enter the ***HOST*** or ***SSID***, proceed with the “very long press” on the “Set” button without entering any character. The hostname and SSID entered previously will remain valid.

NOTE: If you make a mistake while entering the ***SSID*** but the password entered is OK, start the setup again to enter the correct SSID. When asked to enter the ***password***, simply proceed with the “very long press” on the “Set” button without entering any character. The password entered previously will remain valid.

13) If you look at the back of your clock now, you should see that the Pico's on board LED is steady On. This is because last time the clock tried to get Wi-Fi access (during power-up, before you entered the credentials), it failed. So you now need to turn Off / turn On the clock to get Wi-Fi access and re-sync the clock with NTP. At this point, I suggest that you change the clock time (setting a wrong time), on purpose, to see the NTP re-sync after the next power-up. You should also see how long it takes for the clock to complete its power-up cycle. As mentioned previously, it is slightly longer than with previous Firmware versions.

NOTE: Don't be surprised if your clock requires 2 or 3 retries to get a successful Wi-Fi connection during the power-up (initialization) sequence. When it is time to re-sync the clock through NTP, however, the first try is usually successful.

NOTE: When the data scrolls on clock display, you may see a message at the end of the scroll cycle indicating "NTP Errors: 3". If you see many NTP errors over time, it may indicate a problem with your Wi-Fi access. However, a small number 3 or 4 errors over a month or so, is normal. Here is why: when synchronizing the clock with NTP, there is a small possibility of a problem if the configuration is done near a minute change. There could be a "race condition" between the clock real-time clock and the Firmware executing is date and time setting. For this reason, if the NTP update is performed after xx:xx:58 seconds, the algorithm simply skip this setting and wait for the next NTP cycle. In this case, the skipped cycle has been added in the number of NTP errors since the clock has not been updated.

That's it! Enjoy your update and if you appreciate, put a star in my repository ;-)

Appendix M – Reminders

NOTE: Reminders of type 1 were implemented in Firmware Version 10.00 but a system crash occurred and I'll need to re-code most of it. I left the text here as a future reference but be aware that it is not functional for now.

Introduction

Have you ever been asked by your neighbor to go to his apartment and add water to his green plants every Thursday for the next two months while he will be traveling outside of the country for his work ?

Did your mother ever told you that she would like to have dinner with you and your sister every three weeks, on Friday nights?

If one of these situations (or another one similar to those examples) already happened, the Green Clock may help reminding you of your "appointments", "duties", "promises" (or whatever else you may call them) by becoming your "electronic Post-it" to make sure you don't forget what you have to do.

You must know that this feature is a "compile-time option". That means you need to adjust the "Reminder" parameters in the source code, than rebuild the Firmware and upload the new version to your clock.

Basically, what you do is: you configure a few parameters: time, duration, repeat, etc (we'll see details below) and when time comes, the text that you configured will scroll on clock display, along with a few beeps to grab your attention.

Reminders of Type 1

Reminders of type 1 are those for which the elapsed time is exactly the same between each occurrence of the target event. The two examples above are good examples. Your neighbor wants you to go every Thursday (every week), so there will be exactly 7 days between each action. The same applies for your mother: 3 weeks (or 21 days) will be the exact elapsed time between two consecutive actions.

Basically, Reminders of type 1 apply to repetitive actions based on minutes / hours / days / weeks. Actions based on months do not apply. For example, "do something every 5th of each month". Elapsed time between January 5th and February 5th is different than elapsed time between February 5th and March 5th (because of the different number of days in January and February). This last example is an example of Reminder of type 2 which is not currently supported by the Green Clock. It would be the same thing for an action that will occur "every 3rd Wednesday" of each month. Those examples will be covered with Reminders of type 2 if / when implemented in the code.

Reminders of Type 2

As mentioned in the text about Reminders of type 1 above, Reminders of type 2 are for those actions that are repeatable, but with a delay that is not the same between each occurrence. They are usually based on months / season, etc. For example, this needs to be done "every 12th day of each month" or this must be done "every two months, on the second Monday of the month". Since the number of days is different from one month to another, the elapsed time from one action to the other is different and the algorithm to cover those cases must also be different. As of this writing (Firmware Version 10.0), Reminders of type 2 are not supported yet.

NOTE: You may think "*eh, an action based on year should also be considered as a Reminder of type 2*". Yes, you're right, but this kind of event may simply be covered by a "Calendar Event" ;-)

Configuring Reminders of Type 1

For this example, let`s use the following scenario (don't spend time to analyze the likelihood of the scenario… it`s simply used as a guide to make things simple).

So, your neighbor leaves on Sunday, September 4th, 2023 and will be out of the country for about 8 weeks. He'll be back on Monday, October 31st, 2023.

He wants you to go to his apartment every Thursday to add water to his green plants (and you decide that a good time for you to go there, is in the afternoon, somewhere between 13h00 and 18h00.

Those are the parameters that we'll use to configure our first Reminder. Once we go through the example, you will understand how to proceed and you'll be able to configure other specific cases on your own.

Refer to the table below while reading the text that follows to see how to enter the parameters in the source code. The table below that contains the example is similar to those used in the source code. Edit the file "ReminderGeneric.cpp" and configure your own Reminders before rebuilding the Firmware.

Up to 50 Reminders may be configured. There are less than 50 pre-configured Reminders in the source code, but you may copy-paste up to 50. Be careful to make a clean copy- paste including the comma, braces, parenthesis, as the system expects them.

Here is an explanation of what represents each line of the table.

1) The first line in the table is the start of the period. In our case, the period starts when the neighbor leaves the country, that is, September 4th, 2023. Since we need to complete all fields for the time and date, let's say that the period begins at 00h00m00s on Sunday 04-SEP-2023 (the very beginning of the day). The order in which this information must be entered on the first data line is indicated on top of the table: We must enter the Hour first, then the Minutes, seconds, DayOfMonth, Month, Year, DayOfWeek, DayOfYear and summer time flag. In our case, the parameters are: 0, 0, 0, 04, 09, 2023, 1, 0, 0. Note: Sunday is 1 −> Saturday is 7. Also, you can leave the DayOfYear and SummerTimeFlag as 0, the system will calculate and update them (as long as you properly adjusted the "DST country" and "Timezone" parameters on the clock, as described in the clock setup in this User Guide). For those familiar with computer language, those parameters are very similar to a "structure tm" of time description. However, some items have been changed to be more "human-like", like the Year, Month, DayOfYear, etc.

2) The second line of the table will be taken care of by the system. Basically, it will contain the date information you entered on the first line once translated in "Unix Time". That is: the number of seconds between 01-JAN-1970 and the date entered on the first line. This is a good moment to tell you that most of the Reminder algorithm is based on "elapsed seconds".

3) The third line of the table corresponds to the end of the period. Since the neighbor comes back on Monday, October 31st, 2023, this is what we will enter as the end of the period. In a way similar to what we did for the start period, we will enter all fields and assume that the end period corresponds to 23h59m59s (at the very end of the day). Following the same order as in paragraph 1 above (and as indicated at the top of the table) we enter: 23, 59, 59, 31, 10, 2023, 2, 0, 0. As mentioned before, the system will take care of inserting the correct day-of-year and summer time flag.

4) In a way similar to paragraph 2 above, the system will take care of the 4th line in the table by converting the "end of the period" you entered on line 3 to "Unix Time".

5) The fifth line of the table must be filled with the parameters in the same order as for lines 1 and 3 (see paragraphs 1 and 3 above). Fifth line contains the date and time of the first Reminder alarm. In 1 and 3 above, we defined the global period of time during which the system will consider our Reminder to be active (outside of this period, system will simply ignore – skip – this Reminder). Now, we must tell the system at which time we want the clock to ring a "beep" and scroll a reminder message on the clock display. We said previously that every Thursday, between 13h00 and 18h00 would be a good time to go at the neighbor's house. So, let's ask the clock to ring a reminder message beginning at 13h00. The first Thursday after September 4th 2023 is Thursday September 8th 2023. Since we preferably don't want the reminder message to scroll at the same time as the date scroll, let's make the exact time 13h01m36s. So, the data on 5th line will read: 13, 01, 36, 8, 9, 2023, 5, 0, 0 (We will continue to use "13h00" in the example to make things simple even if it is 13h01).

6) You probably won't be surprised if I tell you that system will take care of the 6th line, the same way as for lines 2 and 4 (again to convert the time parameters of line 5 to "Unix Time" on line 6).

7) Line number 7 contains the interval for each ring. When the clock reached the target time (defined in step 5 above), it will sound a few beeps. If you look at the clock at this time, you will see a message scrolling to remind you that you must go to your neighbor house (more on this message later). But if you're not near the clock at this exact moment, you may miss this beep and this message. So let's add an interval of time after which the beeps and the message will be back again. So, let`s define this interval to be 15 minutes, to make sure we don't miss it. As mentioned before, we work mostly with elapsed seconds in the Reminder algorithm, so you can convert the 15 minutes in seconds using a calculator, or more simply, you can enter in the code: 15 * 60l (15 minutes X 60 seconds) Note: For those wondering, the trailing "small L" after the 60 has to do with the size of the variable, a technical detail.

8) Line 8 is the Reminder duration. In our case, let's ask the clock remind us from 13h00 up to 18h00, in case we're out for lunch and come back home only at 17h00. In this particular case, we still want the clock to remind us after 17h00. And if we hear the first beeps earlier and don't want to get beeps for the next five hours, we can simply shut-off the Reminder alarms by pressing on the "Set" (top) button. Since the duration is from 13h00 to 18h00, we will enter 5 * 60 * 60l (5 hours X 60 minutes X 60 seconds). (Don't forget the trailing small "L" as in the example in the table below). The Reminder will ring every 15 minutes, from 13h00 up to 18h00, or until user presses on the "Set" (top) button, whichever happens first.

9) Line nine corresponds to the "Repeat step". Remember in the introduction section of this chapter, it was mentioned that Reminders of type 1 require that the elapsed time between actions be exactly the same. So, in our case, the next "action" will be next Thursday, we want to get another reminder at 13h00. And as mentioned, most of the algorithm is based on "elapsed seconds", so the "Repeat step" corresponds to the number of seconds between Thursday Sep 8th, 2023 at 13h01h36 and Thursday 15th, 2023 at 13h01h36. You may take a calculator to find it, or more easily, you can simply enter: 7 * 24 * 60 * 60l (7 days X 24 hours X 60 minutes X 60 seconds). The system will calculate the total number of seconds for you (again, add the small trailing "L" to the last number).

10) Lines 10 and 11 are reserved for system usage.

11) Line 12 is where you enter the message to scroll on clock display when the Reminder will ring. I suggest that you enter a few dummy characters before the string so that you don't miss the first characters if you take a few seconds before looking at the clock. The string may be up to 50 characters long. For example, we could enter: "Take care of John's plants!".

That's it ! So, basically, what we did is this:

- We defined a global period of time (Start period and End period) during which the system will consider this Reminder as "active".
- We then decide the moment (the first time) at which the Reminder will trigger (ring and scroll a message on clock display).
- We indicate the interval of time at which the Reminder will beep. In our case, we decided that the Reminder will beep every fifteen minutes.
- We also specified to total duration of this Reminder's occurrence (from 13h00 to 18h00).

- Finally, we determined the "Repeat step". Once we shut-off the current Reminder's occurrence with the "Set" button (or once it has exhausted its ringing duration – 5 hours), when is the next time we want it to trigger again (in our case, next Thursday at 13h00, specified as an elapsed number of seconds). As with parameters on lines 7 and 8, parameter on line 9 must be specified in number of seconds.

NOTE: When you press the "Set" (top) button to shut-off a triggered Reminder, the clock will also shut-off all other triggered Reminders, triggered clock alarms and the count- down timer if some of them are ringing at the same time. The same applies the other way around: if you shut-off the count-down time (or a clock alarm), you will shut-off all triggered Reminders at the same time.

NOTE: When you enter parameters in the table, you may enter a single "0" (zero) if you want to indicate zero minute or 0 hour, etc… However, if you want to enter 5 seconds, you can't enter "05" since the system interprets values beginning with "0" as an octal value (except if it is a single 0).

NOTE: If you want to temporarily disable one specific Reminder, you don't have to erase it completely. Simply make the ***year*** of the start period (first line of the specific Reminder number) as "0". When year of the start period is "0", the system simply skips it (consider it inactive).

NOTE: If you power-up the clock and a specific Reminder is "active" (that is, today is inside Start period – End period)… If the line 5 (first Reminder ring time) is in the past, the system will add the "Repeat step" until it finds a date in the future. For example, in our case, if we restart the clock on Sep 30th and line number 5 still contains the date and time we entered for the first ring (Thursday Sep 8th), since this date (Sep 8th) is in the past when you start the clock on Sep 30th, the system will add "Repeat step" (we entered 7 days – in number of seconds) a few times, until reaching Thursday October 6th and will make it its next Reminder ring time.

NOTE: Keep in mind the right terminology: an ***active*** Reminder is a Reminder for which the exact current moment is inside the period defined by the Start period and the End period. A ***triggered*** Reminder is a Reminder that is active and for which we reached the moment to ring and to scroll its descriptive message. Even if a Reminder is currently *triggered*, it may be silent between the *interval* that has been defined to ring and scroll its descriptive message.

NOTE: For those unfamiliar with C language, the line at the top of the table (beginning with /* and ending with */ is a comment and is not processed by the compiler. In a way similar, when the symbols // appear on a line, whatever is written after those symbols is considered a comment and is not processed by the compiler.

Example of the Reminder described in the example above:

```
/* --- Reminder 0 --- struct tm: Second, Minute, Hour, Day-of-month, Month, Year, Day-of-week, Day-of-year, Summer time flag */
  {
    0,  0,  0,  4,  9, 2023, 1, 0, 0,  // global start period: valid period is from Sunday 04-SEP-2023 at 0:00:00
    01,                                // ...system will convert the start period above and feed this variable with start period in Unix time.
    59, 59, 23, 31, 10, 2023, 0, 0, 0,  // global end period: valid period ends Monday 31-OCT-2023 at 23:59:59
    01,                                // ...system will convert the end period above and feed this variable with end period in Unix time.
    36,  1, 13,  8,  9, 2023, 5, 0, 0,  // next Reminder will start ringing on Thursday 8-SEP-2023 at 13:01:36
    01,                                // ...system will convert next reminder above and feed this variable with next reminder in Unix time.
    15 * 60l,                    // Reminder will ring every "this number of seconds" interval. For example, every 15 minutes is: 15 * 60l,
     5 * 60 * 60l,               // Reminder will remain active - will ring intermittently - for "this number of seconds" duration.
     7 * 24 * 60 * 60l,          // Reminder will repeat itself after this number of seconds later. For example, 1 week later is: 7 * 24 * 60 * 60l,
    01,                          // ...system will keep track of elapsed time for Reminder current occurrence and stop it after "Reminder duration".
     0,                          // ...system will cumulate elapsed seconds until "Ringing interval", when it is time to ring the Reminder.
    "- - - - - - - Take care of John's green plants"    // message to scroll when this Reminder is triggered.
  },
```

Appendix N – Daylight Saving Time

While in "Clock Setup" mode, when it is time to setup the Daylight Saving Time mode ("ST" for "Saving Time" – or "Summer Time" - showing up on clock display), refer to this table to determine what must be the setting to be used for your country.

NOTE: In all cases, the hour shift is 60 minutes (back or forward), except for Australia Lord Howe Islands which is 30 minutes.

NOTE: I would appreciate receiving any feedback from users in different countries of the world to confirm if the Pico Green Clock properly supports Daylight Saving Time / Summer Time for their country.

NOTE: For DST setting "5" (see table below), it is important that UTC setting be also properly setup on the clock for automatic DST support.

DST Setting	Country / World Area	DST Start Time	DST End Time
0	Any country – No DST support at all. It's like if Summer Time doesn't exist for the clock	= = =	= = =
1	Australia	1st Sunday October, 2h00	1st Sunday April, 3h00
2	Australia Lord Howe Island (30 minutes shift)	1st Sunday October, 2h00	1st Sunday April, 2h00
3	Chile	1st Saturday September, 24h00	1st Saturday April, 24h00
4	Cuba	2nd Sunday March, 0h00	1st Sunday November, 1h00
5	European Union Akrotin and Dhekelia Albania Andorra Bosnia and Herzegovina Faroe Islands Gibraltar Greenland (except Danmarkshavn and Thule Air Base) Guernsey Isle of Man Jersey Kosovo	Last Sunday March, 1h00 *UTC Time*	Last Sunday October 1h00 *UTC Time*
6	Israel	Friday before last Sunday March, 2h00	Last Sunday October, 2h00
7	Lebanon	Last Sunday March, 0h00	Last Sunday October, 0h00
8	Moldovia	Last Sunday March, 2h00	Last Sunday October, 3h00

DST Setting	Country / World Area	DST Start Time	DST End Time
9	New Zealand	Last Sunday September, 2h00	1st Sunday April, 2h00
10	Bahamas Bermuda Canada (except Yukon and Saskatchewan) Greeland (Thule Air Base) Haiti Nunavut Ontario Quebec U. S. A	2nd Sunday March, 2h00	1st Sunday November, 2h00
11	Palestine	Saturday before last Sunday March, 2h00	Saturday before last Sunday October, 2h00
12	Paraguay	1st Sunday October, 0h00	4th Sunday March, 0h00

Appendix O – Revision History

Revision history.

Version 1.00	- Initial release from Waveshare
Version 2.00 10-FEB-2022	- 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”, only 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.
Version 3.00 24-MAR-2022	- 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.
Version 4.00	- Some code optimization. - Major cleanup in the "setup flags" to make it more straightforward to add options to the

19-APR-2022	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 (available 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.

Version 5.00 27-JUL-2022	- 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 approximate 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.
Version 6.00 15-OCT-2022	- Add Pico's flash memory read, write and erase functions. - Add CRC16 function to calculate the cyclic redundancy checksum of an array of characters (using the 0x1021 polynom). - Add algorithms to save Green Clock configuration to flash memory and restore it when required. - Improve the "degree" symbol for Celsius and Fahrenheit. Thanks to Eric Escolano for proposing this improvement. - Fix a few "compile bugs" when some features are turned Off (remote control, DHT22, BME280, etc...). Thanks to "pjbroad" and "maxromanovsky" on GitHub for bringing this to my attention. - Add a "QUICK_START" compile option to easily allow skipping parts of the power-up sequence if so desired.

	- Change debug output from UART1 to UART0 for easier transfer to a PC through USB CDC interface. - Add a system idle monitor to get an idea of the load on the microcontroller. - Add a sound queue for the active buzzer (the one integrated with the Pico Green Clock). - Adapt the sound callback function to support both the passive buzzer and the active buzzer. - Implement 9 alarms since we can now save them in flash. Alarm algorithm has been completely reworked - Optimize sound algorithm throughout the code by using the sound active queue. - Tried to improve the DHT22 algorithm by replacing the “polling” algorithm by an interrupt-driven function… without success. I’m Still getting more or less 75% read errors. Maybe the Pico is too busy with the callback functions to be able to properly handle the DHT22. When I have some extra time, I may take a look with a logic analyzer to see what happens. Whatever the case, - I have 100% communication success (no error at all) with the BME280 device that I use (see User guide for more info).
Version 7.00 16-NOV-2022	- Given all timing problems while trying to read datastream from DHT22, offload that function to Pico's second core (core 1). - Implement two queues for inter-core communications since the SDK documentation recommends not using the fifos for this purpose if at all possible. - Implement a new generic Daylight Saving Time algorithm covering most countries of the world. - Add Universal Time Coordinate to the clock setup and to the flash configuration. It will be required when implementing time update / synchronization via NTP and it also allows to properly handle Daylight Saving Time / Summer Time for those countries who change the time based on UTC. - Add current DST status to flash memory to optimize DST algorithm. - Change algorithm to find DayOfWeek that was sometimes returning an out-of-sync value.
Version 8.00 10-JAN-2023	- Add "snowflakes" pixel animation (now showing up with the “Play/Pause” remote button).. - Optimize top indicators update function. - Change clock display brightness configuration and algorithm so that it is now driven by a Pico's PWM signal. - Add features and optimize sound queues handling. - Add a command queue to prevent overrun while adding more and more processing in callback functions. - Prevent a quick press on "set" (top) button to enter setup mode if data is currently scrolling on clock display. - Rework / improve / optimize Daylight Saving Time / Summer Time algorithm. - Make some cleanup and optimization on the analog-to-digital related functions. - Debug hourly chime sometimes sounding at xxh01. - Add automatic detection of USB CDC while in "DEVELOPER_VERSION". - Add automatic detection of microcontroller (Pico or Pico W). - Add a half-hour "light chime" (2 quick beeps). - Add an option where hourly chime sounds a number of beeps being equal to current hour (2 o'clock = 2 beeps / 5 o'clock = 5 beeps, etc...) - Cleanup and optimizations in PWM handling and integration of PWM for passive buzzer and clock display brightness. - Modify checking of clock's flash configuration to prevent a glitch on clock display while disabling interrupts.

	- Cleanup code for easier configuration and handling of temperature reading from either Pico, DS3231, DHT22 and / or BME280. - Some general code cleanup and optimization.
Version 9.00 08-FEB-2023	- Change count-down timer alarm algorithm. Will now ring until reset by user or when reaching maximum ringing duration. - Implement NTP (Network Time Protocol). Green Clock will now resync its time periodically from Internet. (Requires Pico W). - Add a function to "set-and'save" Wi-Fi credentials to flash. - Modify CHIME_HOUR_COUNT algorithm, following Ewan Harrow's suggestion and code implementation. Will now ring slower and simulate those "ding-dong" clocks (compile-time option). - Cleanup and many optimizations and cosmetic changes. - Now include two CMaleLists.txt files for Pico and Pico W - Now include two ".uf2" executable files for Pico and Pico W. - Debug some specific cases of Daylight Saving Time. Fix a bug with days-of-week handling in clock alarms. - Add display of NTP-related data available with remote control. - Add capability to simply address any clock pixel with row and column parameters (function set_pixel()).
Version 9.02 01-JUN-2023	- Change "middle dots blinking" algorithm. Thanks to Frank Seidel for his suggestion and work on this feature. - Add basic German language support. Thanks to Frank Seidel for his work and translation on this feature ! (Note: English is used if / when German translation is not available). - Add functions to easily support and convert tm_time, human_time, unix_time. - Fix a problem when initializing CYW43 with new SDK library. - Increase string length for Calendar Events up to 50 characters. - Begin implementation of "Reminders", giving even more flexibility than Calendar Events (to be completed). - Fix dates encoded in CalendarEventsGeneric.cpp (example Calendar Events). - Add Czech language support. Thanks to KaeroDot for the excellent work and translation on this feature !
Version 10.00 29-JAN-2024	- Add hostname to the NTP and networking initialisation. Add to flash structure too. - Add web page to control the clock settings. This can change many things. Accessed as http://hostname or IP/index.shtml - Reduce the display minimum dim level by altering the display PWM frequency. - Add a step dimmer on the lower button to switch between auto dimming and 5 levels of manual modes. - Add a web control to set a local level for the minimum light level for maximum dimming. This can vary between clocks. Value stored in flash. - Add a web control to swap the operation of the set key from short press to set clock to short press to set the alarms. - Fix triggering of afternoon alarms when in 12hr display mode. - Release on GitHub.
Version 10.01 25-FEB-2024	- Correct the flash_config structure in the Pico-W NTP Client code. Now matches the one in the main code. - Add more bytes to the alarm structure. This may cause the 10.01 first install to wipe out all

	flash alarm and WiFi data. - Correct the light ADC initialisation after a firmware 10.00 release voltage reading fix caused more problems than it solves. - Add in a new feature to temporarily disable all alarms for a 4 hour window by pressing any two buttons together. This doesn't affect any settings. - Add in the ability for all of the alarms to drive jingles, a different set of beeps or use the onboard buzzer if the passive Piezo one is fitted. - Add in support for local reminder, event and WiFi config files that can be pulled in when built without appearing in the git sources. - Tweak daylight savings region web control drop down so that it's now populated correctly rather than having a print out of the region and drop down to change. - Add periodic print out of the Pico-W WiFi connection status and IP address to the UART or USB serial console (will work with putty, etc.). Serial baud rate is 921600.
Version 10.10 26-DEC-2024	- Add a new feature to control the hour / minute separator flashing mode to add 3 more modes.