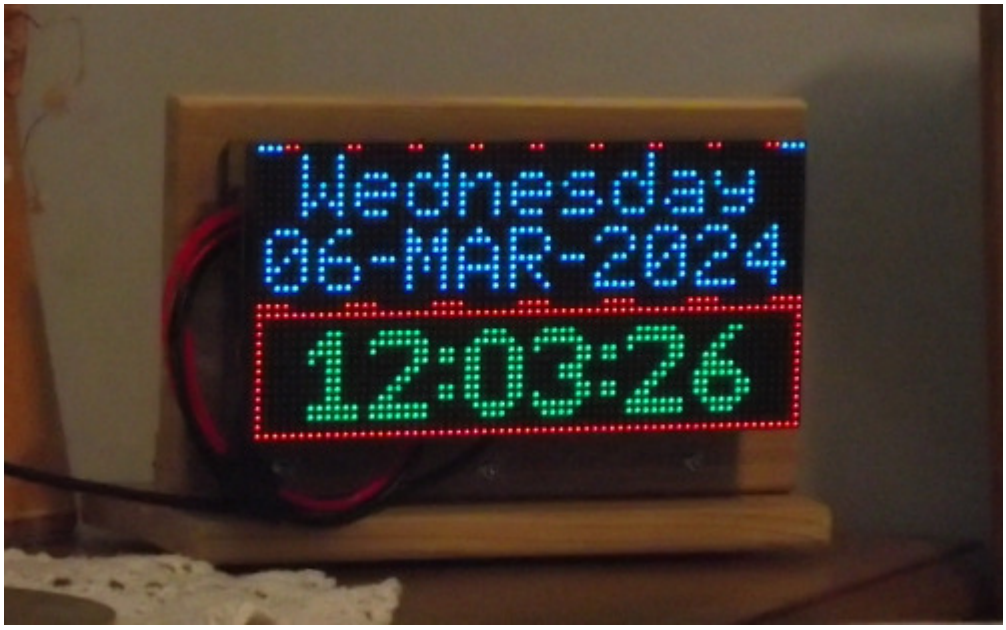
Pico RGB Matrix



Firmware Version 2.01
User Guide
Updated April 25th, 2024

IMPORTANT :
This User Guide is about Pico RGB Matrix firmware
Version 2.01 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 device. If you're using the original firmware from Waveshare, many features described in this manual do not apply.

Join our Pico-RGB-Matrix discussion group on:
https://github.com/astlouys/Pico-RGB-Matrix/discussions

Pico RGB Matrix User Guide

Index

How to use this User Guide

This User Guide has been written with a « chronological » order in mind. Basically, first section should be read first, then second section, and so on and so forth.

That being said, here is another approach if you want to get faster results. It is always possible to come back later and read what you skipped.

At the minimum, you should read the section « Using the RGB Matrix remote control » to understand how the remote control is used to access different features of the device.

You should also read the section « Using an external terminal emulator » to understand how to connect an external PC running a terminal emulator program to the USB port of the RGB Matrix. A terminal will provide an easier and faster way to access all features of your RGB Matrix. In the future, more features will be added to the remote control support but nonetheless, there will always be some options that simply can't reasonably be used through the LED matrix display (many actions available from the terminal menu display too much information to be used with the LED display).

Finally, when you want to do something specific with your RGB Matrix, look at the section « Understanding the RGB Matrix features ». This section will explain how to use / setup / configure a specific feature of the device, but it assumes that you are already familiar with using the remote control and terminal emulator and will focus only on the specific feature you want to use.

Do you want to share your experience with the Pico RGB Matrix device and help other users? Join our discussion group on:

https://github.com/astlouys/Pico-RGB-Matrix/discussions

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

What's new with Firmware Version 2.01

Not much!

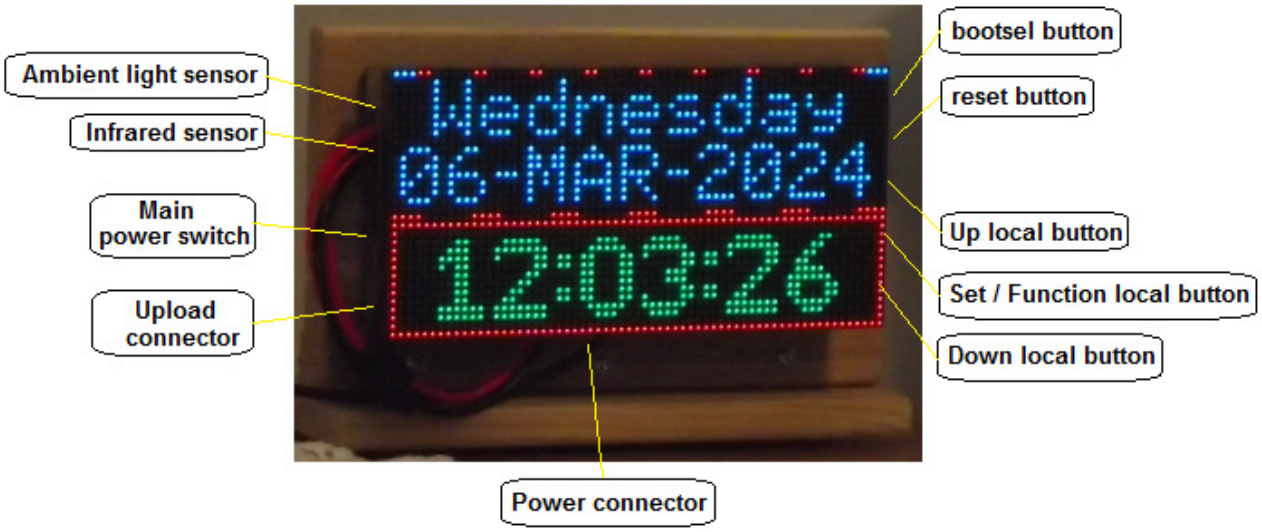
However, Waveshare produced a new version of the LED Matrix board, and also a new version of the control board. The Firmware who worked fine on the old versions of the hardware didn't work at all on the new version because of a very small change in the timing of the matrix scan.

I want to thank Waveshare support team and engineering team for their help on this one, since at the time the problem happened, I didn't have a new version of the boards on hands.

Version 2.01 also improve the bitmap of digits « 6 » and « 9 » in the 8x10 character set, and add some optimization in the NTP algorithm.

Enjoy!

Pico-RGB-Matrix reference pictures



Front view

Pico RGB Matrix User Guide

Rear view

NOTE: Be aware that the names given to some of the switches above do not correspond to the names given by Waveshare. The names have been assigned to represent the behavior of the switches and to be easily identified while using the RGB Matrix.

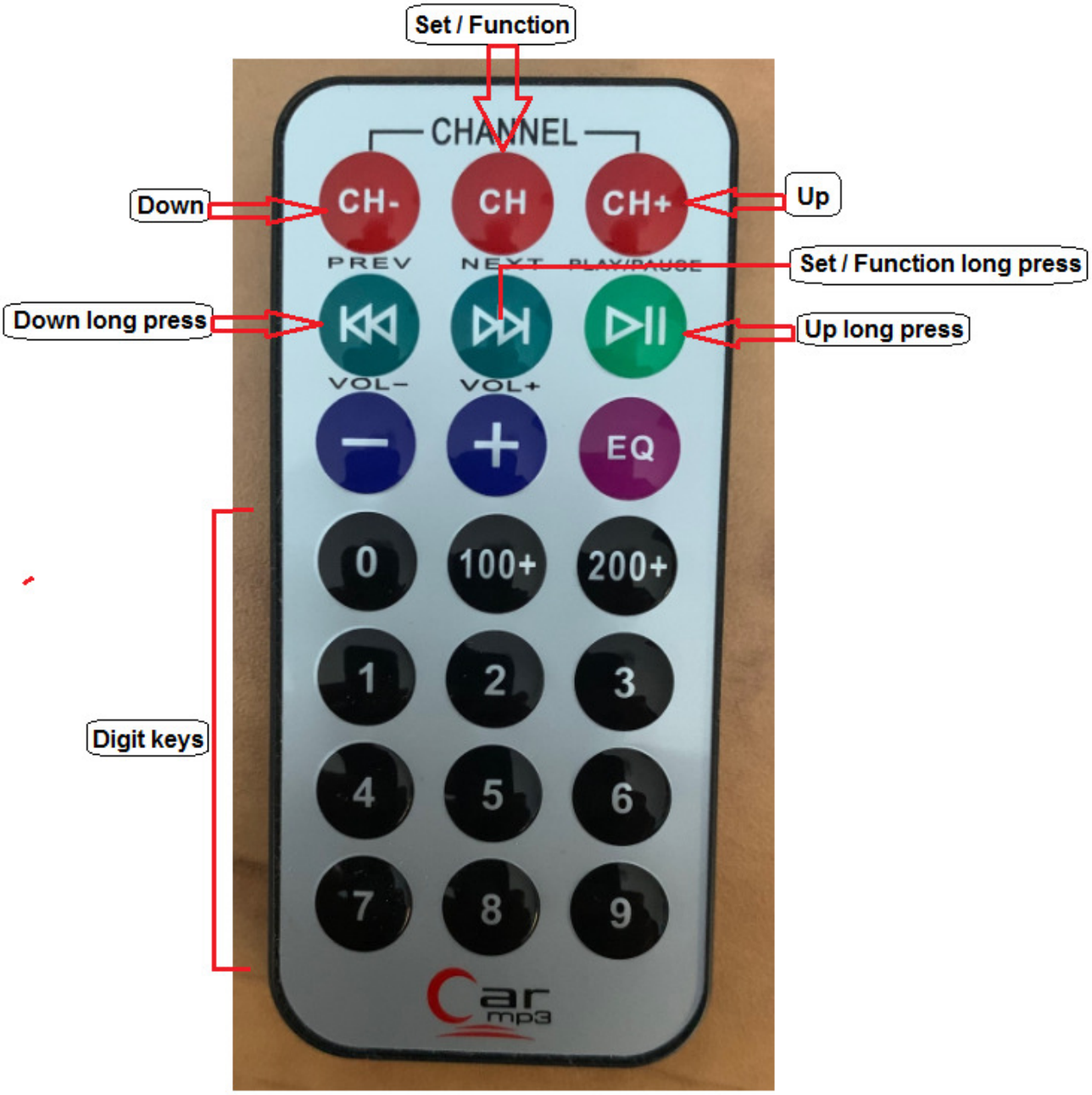
The first button at the top of the device is called the « Bootsel » button since it is used to « flash » (to « upload ») a new Firmware version to the Raspberry Pi Pico. Be aware that this button is « hardwired » (electrically connected) to the Pico and the Firmware can't change its behavior. See the section on « How to upload new code to the PicoW » for more details. IMPORTANT: This section will also give you more information on a specific « glitch » with the Raspberry Pi PicoW and the workaround.

The second button from the top is the « Reset » button. It is simply used to reboot (« restart ») the Firmware on the Raspberry Pi Pico. As for the « Bootsel » button, Be aware that the « Reset » button is « hardwired » (electrically connected) to the Pico and the Firmware can't change its behavior.

The third button from the top (the middle button on the side of the RGB Matrix, in fact) is called the « Up local button » since it is usually used to increase the value of current setting (it is the « top » of the three buttons used by the Firmware).

The fourth button from the top is called the « Set / Function local button » because it is usually used to scan through the many settings of the device (it is in the « middle » of the three buttons used by the Firmware).

The button at the bottom is called « Down local button » since it is usually used to decrease the value of current setting (it is at the bottom of the three buttons used by the Firmware).



Remote control unit

The remote control picture above shows the button names used with the Firmware and throughout the guide.

The three red buttons at the top reproduce the « Down », « Set / Function » and « Up » local buttons on the RGB Matrix. In a way similar, the three green buttons on the remote control reproduce a « long-press » on the « Down », « Set / Function » and « Up » local button on the RGB Matrix. (A long-press means pressing the local button for more than one third of a second).

Introduction

Waveshare Pico RGB Matrix

As mentioned on the cover page, this User Guide is about Firmware Version 2.00 by Andre St. Louys. This Firmware is based on the original Waveshare's Firmware Version 1.00 and adds more features to the device and tons of comments in the source code which is in C Language.

As its name implies, the Waveshare's Pico-RGB-Matrix uses a Raspberry Pi Pico (or PicoW) microcontroller to control most of its functions / features (IMPORTANT: the microcontroller itself - Raspberry Pi Pico - is not included with the device. However, it is not very expensive – more or less 15 US$ at the time of this writing). The LED display is a matrix of 32 rows by 64 columns (for a total of 2048 positions or « pixels ») of tri-color LEDs. Each of these pixels can be individually turned On or Off in any combination of the three basic colors: red, green, blue, red + green (yellow), red + blue (magenta), green + blue (cyan), red + green + blue (white). The intensity of the LEDs can also be changed, but take note that the intensity can't be modulated among the LEDs and the pixel's brightness will be the same for the whole matrix.

There is also a real-time clock integrated circuit (« RTC IC » – DS3231) with a battery back-up, allowing the device to keep the correct time and date in case of a reboot and / or power failure. This real-time clock IC has shown a great stability and very small time drift over long periods of time. Nonetheless, if the user decides to use a PicoW (instead of a Pico) as microcontroller, he will benefit from the Wi-Fi capability and NTP (« Network Time Protocol ») that is part of Firmware Version 2.0, so that the RGB Matrix will periodically resynchronize itself from a time reference source over the Internet for an even better accuracy.

Many options and features may also be customized and this configuration is saved to flash (non volatile) memory, so that it will be read back every time the device is powered On, preventing the user to reconfigure the device each time the device is rebooted or in case of a power failure.

For those already familiar with the « Pico-Green-Clock » (another product from Waveshare), and the Firmware that I made available on my GitHub repository, you will realize that many features implemented in the Green Clock have also been ported to the RGB Matrix,

The Pico-RGB-Matrix comes with a small remote control that can be used to remotely control the device. More details are given in specific sections of this User Guide.

To get the most out of the RGB Matrix, users may use a personal computer with a terminal emulator program. Many options to better fine-tune the device are currently available only via a terminal emulator program connected to it. Once properly configured, however, the details of the configuration are saved to flash and the PC is no more required.

This User Guide was written with all user profiles in mind. Whether you simply want a color alarm clock or you want a platform to develop some functionality you should find the sections required to get the most out of your device in this guide. So, don't be intimidated with the technical parts and simply

skip the parts of the document that does not apply to you. It will still be time later to come back and learn more if you so desire…

Now that you have an overview of what is the Pico-RGB-Matrix, you may want to take a look at the next section: « Firmware highlight » to have a better idea of all the features available on the Pico-RGB-Matrix !

Firmware highlights

Here is a list of the main features of this Firmware:
(note that some planned options / features are still to be completed as indicated below).

64-bits framebuffer

Handling of the RGB Matrix framebuffer has been ported from 8-bits (in Waveshare's original Firmware) to 64-bits for easier, faster and optimized support.

5x7 character set bitmap with variable width

A 5x7 character set bitmap with variable width has been implemented.

8x10 character set bitmap with variable width

A 8x10 character set bitmap with variable width has been implemented. Thanks to Pasquale D'Antini (« oldmaker » on GitHub) for his collaboration for the numerals bitmaps. (Note: some characters remain to be defined as of Firmware Version 2.00).

Active buzzer handling by callback

Active buzzer sound handling is now performed by a callback for a much easier support and for independent handling from the main system thread.

Alarms

Support nine (9) independent alarms. Many parameters can be configured by the user for each alarm:

- Target hour
- Target minute
- Target days-of-week (as many as desired)
- Number of « Beeps » for each ring.

- Duration of each « Beep » (in msec)
- Repeat period of each « Ring ».
- Total ring duration if user does not shut off the alarm.
- Message to scroll when the alarm rings.
- Number of times to scroll the message on the display.

NOTE: See « Alarm » section in the User Guide for more information.

Alarm indicators

Show which alarms are On and which alarms are Off on the usual « Date and Time » display mode.

NOTE: See « Alarm » section in the user guide for more information.

Alarm target days-of-week indicators

Those indicators show which days-of-week that currently have active alarms and those that don't have. (Available when the device is in the usual « Date and Time » display mode).

NOTE: See « Alarm » section in the User Guide for more information.

Ambient light sensing logic

Adapt the logic of ambient light sensing so that a higher value means « more ambient light », to make it more intuitive (as opposed to the original Waveshare's Firmware).

Ambient temperature display

Provide ambient temperature display.

Auto-brightness control

Option to automatically change the LED display brightness based on current ambient light.

Auto-detection of a terminal emulator connection

Auto-detection if there is a connection with an external terminal emulator. Firmware will automatically jump into a terminal menu. Provide a delay during the power-up sequence so that users have time to start a terminal emulator program to interact with the Pico-RGB-Matrix.

Auto-detection of microcontroller type

Provide automatic detection of the microcontroller installed in the device (Pico or PicoW), along with microcontroller « Unique ID » (« serial number »).

Auto-scroll option

Customizable « Auto-scroll » option: allows automatic scrolling of information on RGB Matrix at programmable time intervals. Information to be scrolled can be selected by user among a predefined list of choices (see details in the User Guide). Five (5) independent Auto-scrolls may be configured.

Automatic support of Daylight Saving Time

Allow automatic support of daylight saving time for most – if not all – countries in the world, along with different time zones (to be completed).

Brightness hysteresis

Automatic brightness control is supervised by a hysteresis to prevent quick changes in the LED matrix brightness in case of a quick and temporary visual interference with the ambient light sensor.

Brightness range control

Allow adjustment of the auto-brightness range (« Low-limit » and « High-limit »).

Bootsel mode timed window

Add a delay (« timed window ») during the power-up sequence so that developers can put the Pico in « Bootsel mode » while the LED matrix is completely blank (to prevent LEDs over-bright).

Calendar events

Add support for 64 different « Calendar events » allowing the user to program birthdays and / or other dates of special meaning. An alarm will periodically ring and a message will periodically scroll during the target day when the date is reached.

Circular buffer for sound handling

Sound handling uses a circular buffer for quick and easy handling.

Count-down timer

Provide a count-down timer with an alarm (to be completed).

Count-up timer

Provide a count-up timer (to be completed).

Debug engine

Provide a debugging engine and algorithm to easily turn On debugging for specific sections of code displaying data through an external terminal emulator program. This can be done at compile time and / or from an external PC running a terminal emulator program.

Device customization

Provide a mechanism in the source code to allow for easy RGB Matrix customization / personalization for different rooms / users / usages.

Device integrity check

Provide a « device integrity check » function to sequentially test every LED of the RGB Matrix.

Device total up-time

Optionally scroll the cumulative device « Total Up time » since last power-up.

Golden age option

Provide a « Golden Age » option to help senior people having problems in their day-to-day lives.

Half-hour chime

An option allows to ring a « light chime » when reaching the half-hour.

Horizontal scrolling

Horizontal text scrolling feature: to scroll text on any line of the LED display.

Hourly chime

An option allows to ring a chime when hour changes.

Hourly chime On-time and Off-time

Provide a way to shut off the hourly chime (and half-hour light chime) during a configurable period of time so that there is no sound during the night.

Infrared burst pilot indicator

A « pilot indicator » on the RGB Matrix shows when an infrared burst is received, decoded and recognized.

Language support

Currently support English and basic French support. Chinese language has been discarded from original Waveshare's Firmware (sorry, I don't understand Chinese).

Layout of the RGB Matrix display

The layout of the RGB Matrix display has been changed from the original Waveshare's version. See front cover of this User Guide for a picture or the « normal » Date and Time display mode.

Main system loop status indicator

LED display provides indicators to show that the main system « forever loop » is up and running (still active).

Network Time Protocol

Support for Network Time Protocol (« NTP ») to periodically resynchronize the RGB Matrix with an Internet time reference source. This requires the use of a PicoW.

Night light

RGB Matrix can be configured to become a « night-light » when the ambient light is dark, or a « timed service light » to provide a light source for a few minutes upon request from the remote control.

Other languages support

As mentioned above, English and basic French are supported. Source code infrastructure has been planned to easily add more languages in the future. (I suggest some coordination before adding other languages since the coming versions of the Firmware could make significant changes to the structure of the text).

Pico's core 1

Pico's core 1 is used to relief core 0 from some time-critical tasks.

Pico's temperature display

Provide microcontroller's temperature display.

PWM Brightness control

LED display brightness is now controlled by a Pulse Width Modulation (« PWM ») signal, allowing a much better control over the RGB Matrix display brightness and free up the microcontroller from managing this task. This allows for a very large range of brightness levels. NOTE: At the lowest level, the RGB Matrix is VERY dim.

Remote control support

Support for the remote control included with the Pico-RGB-Matrix.

RGB_matrix_xxx() family of functions

Many RGB_matrix_xxx() family of high-level functions have been implemented for an easy-to-use interface.

Service light

A « timed service light » may be turned On with remote control allowing to temporarily increase the RGB Matrix brightness when more light is required in a dark environment.

Silence period

A « silence period » can be triggered by the remote control. It will shut off the hourly and half-hour chime for a specific period of time, along with the ringing of Calendar events (text scrolling will be executed as usual, tough).

Software reset

Allow the device to be set in « Bootsel » mode by firmware to overcome a problem due to a physical difference between Pico and PicoW (see section about uploading a new Firmware version in User Guide).

Sound shut off

Provide a compile time option to completely shut off sound from the device.

Source code for Raspberry Pi Pico and PicoW

Keep the Raspberry Pi Pico (or PicoW) source code (discard EPS32 microcontroller support from original Waveshare's version).

Temperature unit configuration

Temperature unit is configurable (Celsius or Fahrenheit).

Terminal support

Support an external PC running a terminal emulator program. Allows a much easier interaction with the RGB Matrix for control, configuration, information display.

Test zone source code

Provide a « Test zone » in source code with many chunks of test code (for developers).

Time display format

Allow time display in 12-hour or 24-hour format (to be completed).

Time functions

For developers, specific functions have been implemented to support « human time », « tm time » and « Unix time », to better support DS3231, Network Time Protocol and other inherent functions.

User Guide

Provide a User Guide to help users getting the most out of this amazing device.

Variable names and function names

Rename many variable names and function names from Waveshare Firmware so that they are more representatives.

Watchdog supervision

Provide a watchdog mechanism to recover from a Firmware crash for users who want to modify / work on the Firmware.

Watchdog status indicator

Provide an indicator on the RGB Matrix showing the watchdog timing status and « how far we are » from an eventual restart.

Wi-Fi health indicator

Provide LED display indicators on the RGB Matrix to report Wi-Fi network health and / or NTP status.

WIN_xxx() family of functions

Many win_xxx() family of high-level windowing functions have been implemented for an easy-to-use basic windowing system interface.

This project provides Firmware for the « Pico-RGB-Matrix » product from Waveshare (The device is available directly from Waveshare website, or from Amazon). It is based on a Raspberry Pi Pico microcontroller.

Waveshare provides a basic Firmware (Version 1.00) for the Pico-RGB-Matrix. Current project provides many enhancements and functionalities to the device. You may refer to the User Guide of the most recent Firmware Version (included in my GitHub repository) to see the detailed list of changes / enhancements.

But that's enough reading for now. Let's put together all the parts and proceed to the next section for assembling and cabling the Pico-RGB-Matrix!

Pico-RGB-Matrix assembling and cabling

Before going further, we need to put together all the parts of the Pico RGB Matrix since it comes unassembled. This is an easy task and this section will guide you through the different steps…

Parts description

First of all, you may want to make sure you have all parts readily available:

Pico-RGB-Matrix parts.

1) Main shipping box.
2) RGB Matrix back cover.
3) Optional RGB Matrix front filter.
4) Double-sided transparent stickers.
5) RGB Matrix LED display.
6) RGB Matrix control board.
7) Remote control unit.
8) Optional power supply connector.
9) Optional magnetic holders.

10) Optional expansion ribbon cable.
11) Main power connector cable.
12) Back cover spacers.
13) CR1220 button-type battery (not shown on the picture).

NOTES :

- The Raspberry Pi Pico (or PicoW) itself is not included in the package and you must buy it separately. It is available from Amazon, but you may want to check if you have a local dealer for Raspberry Pi products since there could be a significant price difference if you do have one.
- If you buy a Raspberry Pi PicoW (instead of a « plain » Pico, you will take advantage of synchronizing the clock with NTP (« Network Time Protocol »). However, there is a little « glitch » associated with the PicoW… Read the section about « Uploading a new Firmware Version » in this User Guide to get more details.
- The double-sided transparent stickers (part number 4 on the picture) may be used if you don't like the crispiness of the display LEDs and if you want to put the optional front filter on top of the Matrix LED display. Make sure you have long minutes available to remove the red plastic cover!!
- The optional power supply connector (part number 8 on the picture) could be used if you plan to use a power supply with a standard « power connector ».
- The optional magnetic holders (part number 9 on the picture) could be used if you plan to install the RGB Matrix on a metallic surface. The magnetic holders can be screwed to the back cover spacers to hold the RGB Matrix.
- The expansion ribbon cable (part number 10 on the picture) could be used if you want to add another « hub-75 compatible » RGB board beside (or under) the RGB Matrix. I didn't try it, but it seems that another « hub-75 compatible » RGB board may be added, You'll be on your own to adapt the software for now since I don't have such a board on hands. Also, be aware that if you add a « hub-75 » board, it must not be another « Pico-RGB-Matrix ». The « Pico-RGB-Matrix » provides a control board that is not required if you plan to extend. All you need is a « dumb hub-75 » board. I suggest you contact the Waveshare customer support to guide you.

Assembling

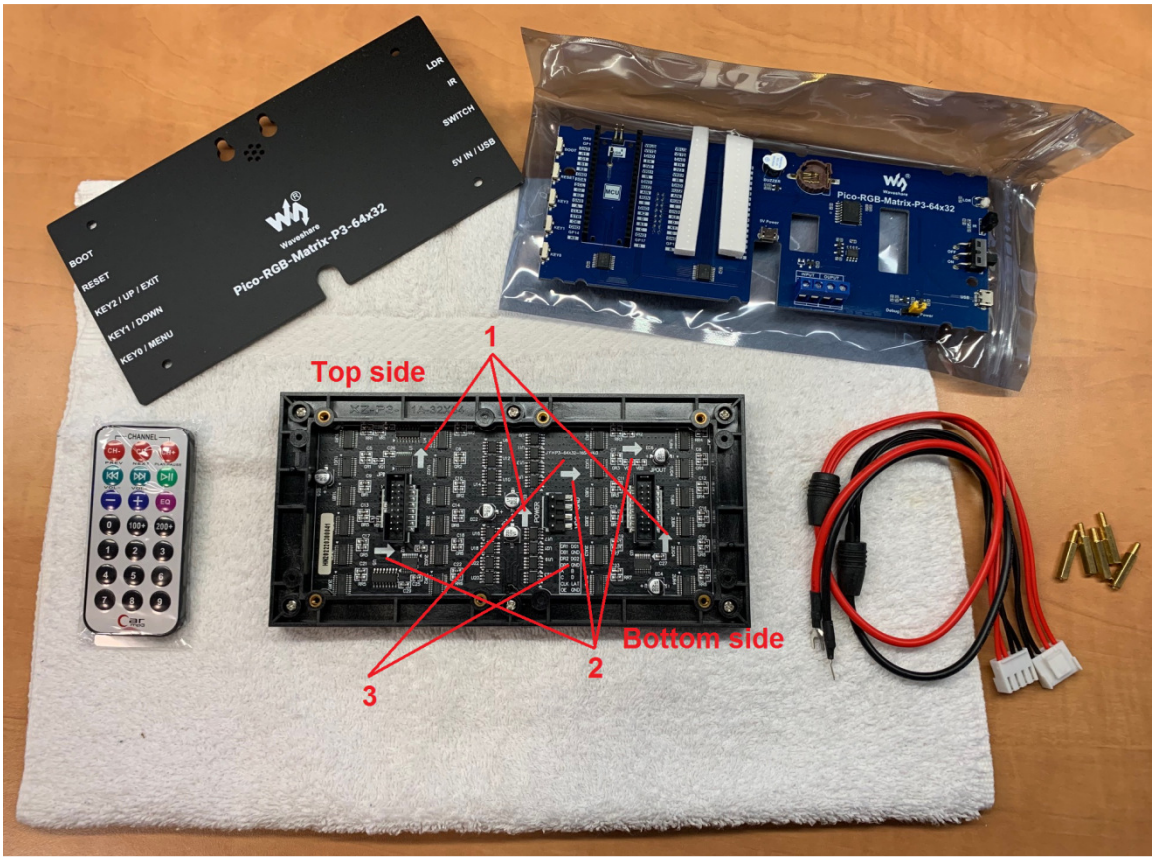
NOTE: The pictures below refer to the old models of the RGB Matrix LED display and the RGB Matrix control board. The instructions remain the same, beside the fact that the pictures may not correspond to the modules that you received if you have the new models.

1) Put aside the parts that you're not planning to use in the short term: Maybe the front filter, ribbon cable, magnetic holders, power supply connector, double sided stickers? And also all the anti-static bags. You may want to put them all in the original RGB Matrix shipping box in case you want to use some of them in the future.

Unused parts are stored in the original shipping box.

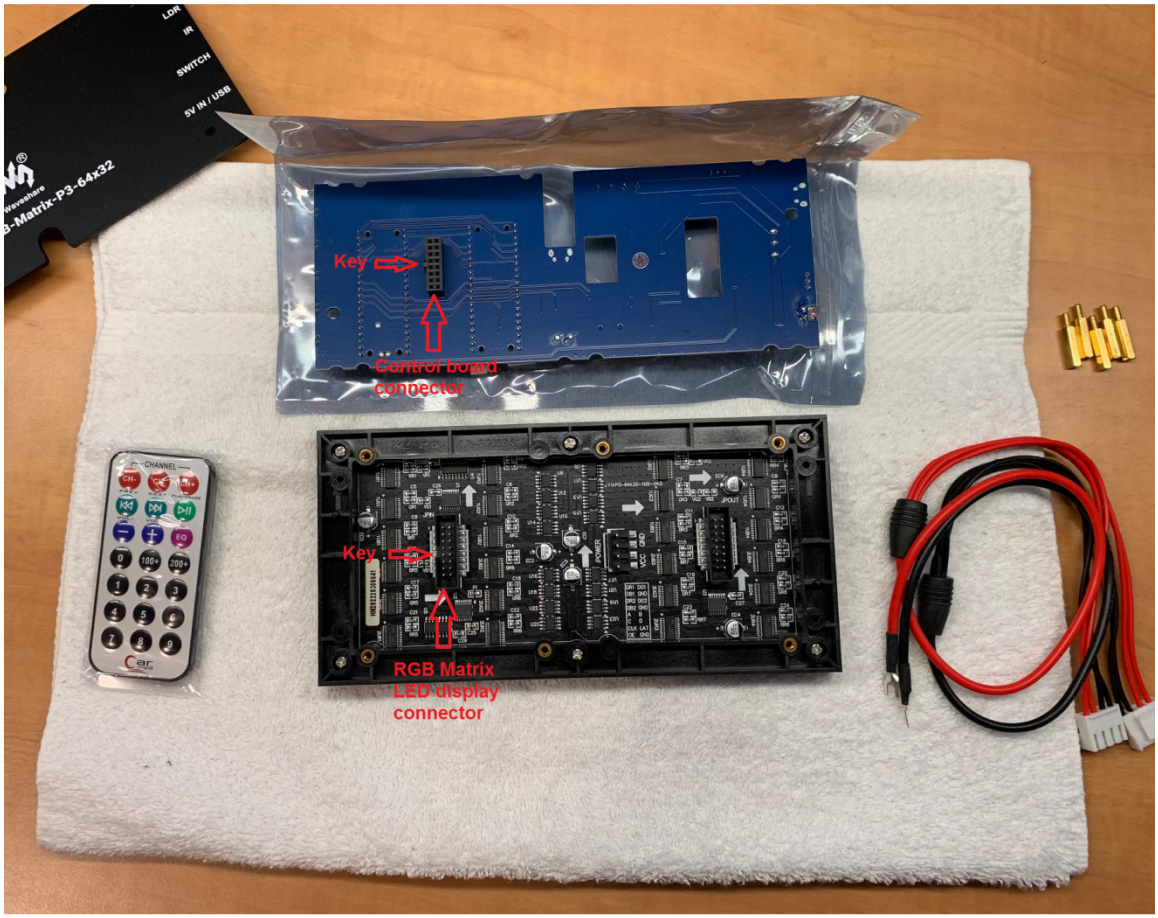
2) Put a soft cloth on your working area and put the RGB Matrix LED display (part number 5 on the « Parts description » picture above) upside down on it. Make sure all the arrows on the PC board point toward the right (see arrows number « 2 » on the picture below) or away from you (see arrows number « 1 » on the picture below). Also make sure that the text is not upside down (see text on number « 3 » on the picture below. This way, the top of the LED display is the side away from you and the bottom of the display is the side near you.

NOTE: Be careful when you manipulate the RGB Matrix. The « plastic screen » filling up the spaces between all LEDs is particularly fragile, in particular on the device borders. You should hold the device by the main plastic borders located below the plastic screen.



RGB Matrix LED board upside down.

3) Next step is to insert the control board on top of the RGB matrix LED board. On the picture below, the control board is shown upside down. The control board connector must be inserted on the RGB Matrix LED board left connector. There is a « key » on each connector, so that there is only one way they can be connected to each other. Gently and carefully insert the control board on top of the RGB Matrix LED board.



Control board (shown upside down) must be inserted on top of the RGB Matrix LED board.

4) You can now insert the battery in its holder as shown on the picture below. The battery must be inserted with the text on top so that you can read it when it is inserted in its holder. First, slide one side of the battery under the metal lip on the « straight » side of the battery holder, and then press on the other side of the battery until you hear a « click » indicating it is locked in place.

   The battery is used to provide power to the real-time clock integrated circuit (DS3231) in case of a power failure. My experience with the Pico-Green-Clock (also available from Waveshare) has shown that the DS3231 is very stable and provides a very good precision over long periods of time. This, along with the fact that the RGB Matrix can benefit from Network Time Protocol makes it an excellent product.

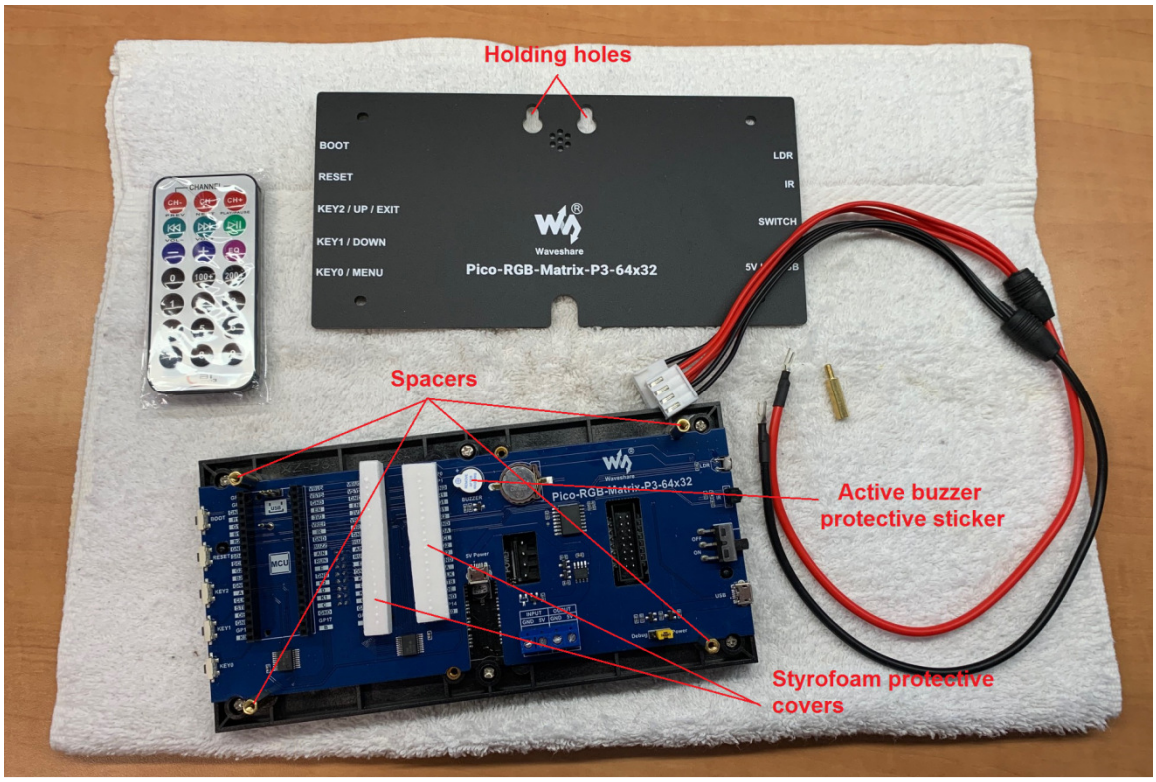
Battery to be inserted in its holder.

5) At this point you should have the equivalent of the picture below. The control board has been inserted on top of the RGB Matrix LED board with their respective connector matched together and the battery has been put in it holder.

   The picture below shows four (4) spacers that have been screwed in place in the four corners of the RGB Matrix LED board. If you plan to use the magnetic holders, you may screw the fifth spacer in the top or bottom middle position, but it is not mandatory. If you're not using the magnetic holders, you will probably use the two « holding holes » to harness the RGB Matrix on the wall or another platform, as shown on the picture below. In this case, it is better not to use the 5th spacer.

   It is also a good time to remove the active buzzer protective sticker as shown on the picture below. Personally, I'm leaving it because this particular device unit will go in a bedroom and it is better to keep the sound to a lower level.

   You can also see on the picture below the foam protective covers on top of the free socket, beside the Raspberry Pi Pico socket. We will keep these foam covers for now, but we may come back later to extend the ecosystem of our Pico-RGB-Matrix based products.



Ready to proceed to cabling.

Cabling

1) At this point in time, it is a good moment to talk about RGB Matrix power requirements. If you refer to Appendix B, you will have an idea of what are the power requirements for the RGB Matrix under a few different scenarios (different brightness levels, different display contents, etc). The goal here is not to give an exact figure since there are so many different variables (different brightness levels, different number of LEDs turned On and different colors used), but rather to make it clear that in some conditions, current requirement is not far from 1 amp.!!

   We could connect the RGB Matrix with a USB cable between the Pico on the RGB Matrix and a PC, flash the Firmware and let it run. In such a scenario, the LED matrix will take its power from the PC USB port and will work fine. However, as we go and turn On more LEDs and / or increase the brightness, the PC USB port may have problems to supply this current. USB specifications indicate that a USB port 2.0 can provide a maximum of 500 mA, whereas USB 3.0 is 900 mA. Even if USB 3.0 is rated for a little more than the maximum required (and even more for the latest USB standards), it is nonetheless a good idea to provide a robust source of current if you plan to leave the RGB Matrix powered On most of the time. It is a good opportunity to go in your basement (or in your attic) and look for this old PC that you were using years ago that is now obsolete and covered with dust. A PC power supply can typically provide many amps on its 5 volts DC (the one beside me indicates 22 amps!! – see picture below). If you don't like the noise of its fan, you can always buy a Raspberry Pi power supply. Some of them can provide 3 amps (5 amps for the new Raspberry Pi 5) or more without fan. However, the most recent Raspberry Pi use USB type-C connectors whereas the RGB Matrix uses a standard micro USB connector. An adaptor will be required if you go this way.

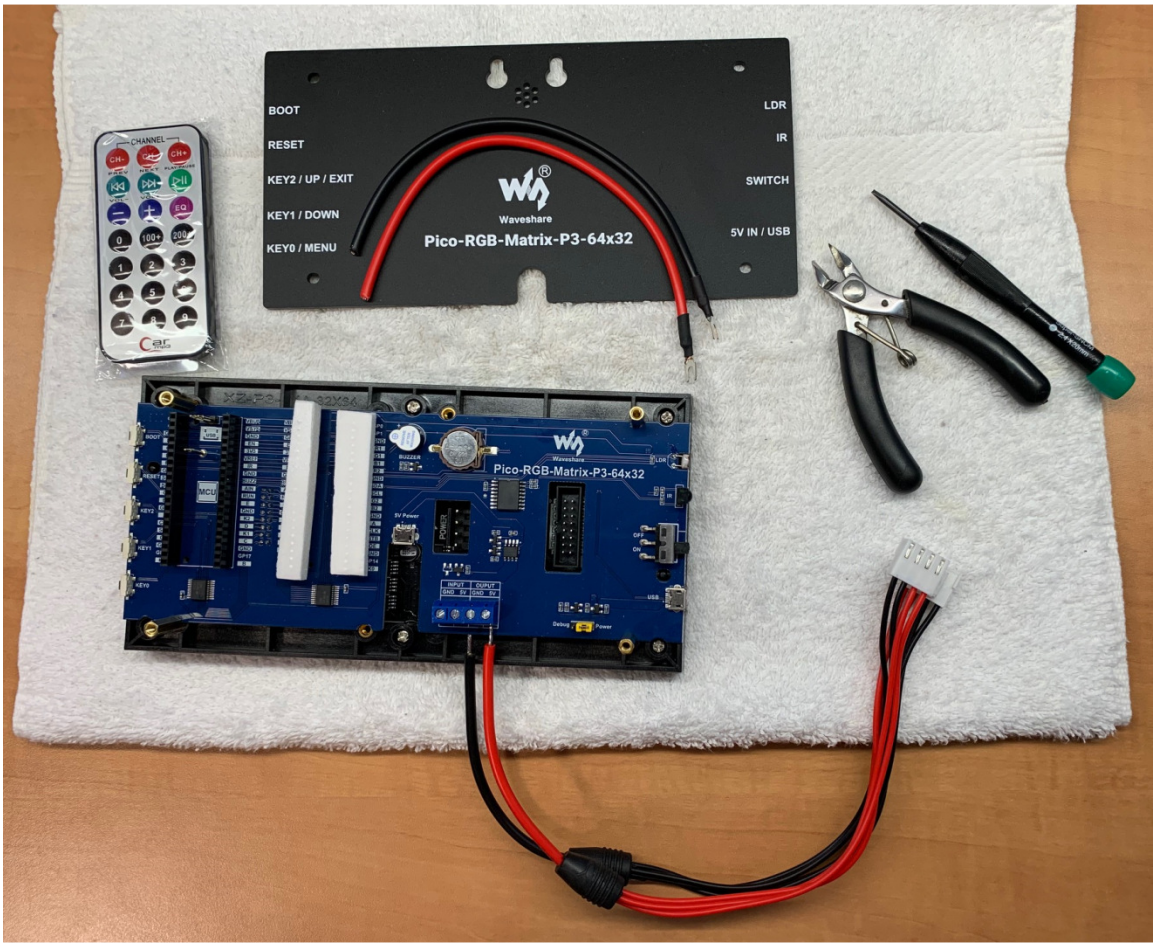
   Next steps will show how to connect the power cables to the RGB Matrix.


ANCER
SWITCHING POWER SUPPLY
MODEL: PP-220U
INPUT: 115/230V ~ 60Hz/50Hz
5.8A/3.2A
OUTPUT: 220W
+5V +12V -5V -12V
MAX. (A) 22A 8.5A 0.5A 0.5A
LEVEL 3
Achtung!
Dieses Netzteil ist wartungsfrei! Im Inneren des Netzteils treten gefährliche Spannungen auf. Das Öffnen ist lebensgefährlich. Im Servicefall wenden Sie sich bitte an Ihren Händler.
verlust der Garantie beim Öffnen.
LR91879
CAUTION! HAZARDOUS AREA
Under no circumstance is this to case be opened. This machine is not user servicable and dangerous high voltages inside this case. In event of difficulty, please notify your dealer for prompt service.
E147055
NOV 1993 S/N P 2204808
CPC

2) Cut the cables provided with the RGB Matrix as shown on the picture below. You may want to leave the black cable one or two mm longer than the red cable so that both cables can be kept together easily.

   Prepare both cables by leaving about ¼ of an inch (about 7 mm) of copper.

   If you're equipped to do so, you may put a drop of solder on both cables to prevent copper strands to stay apart.

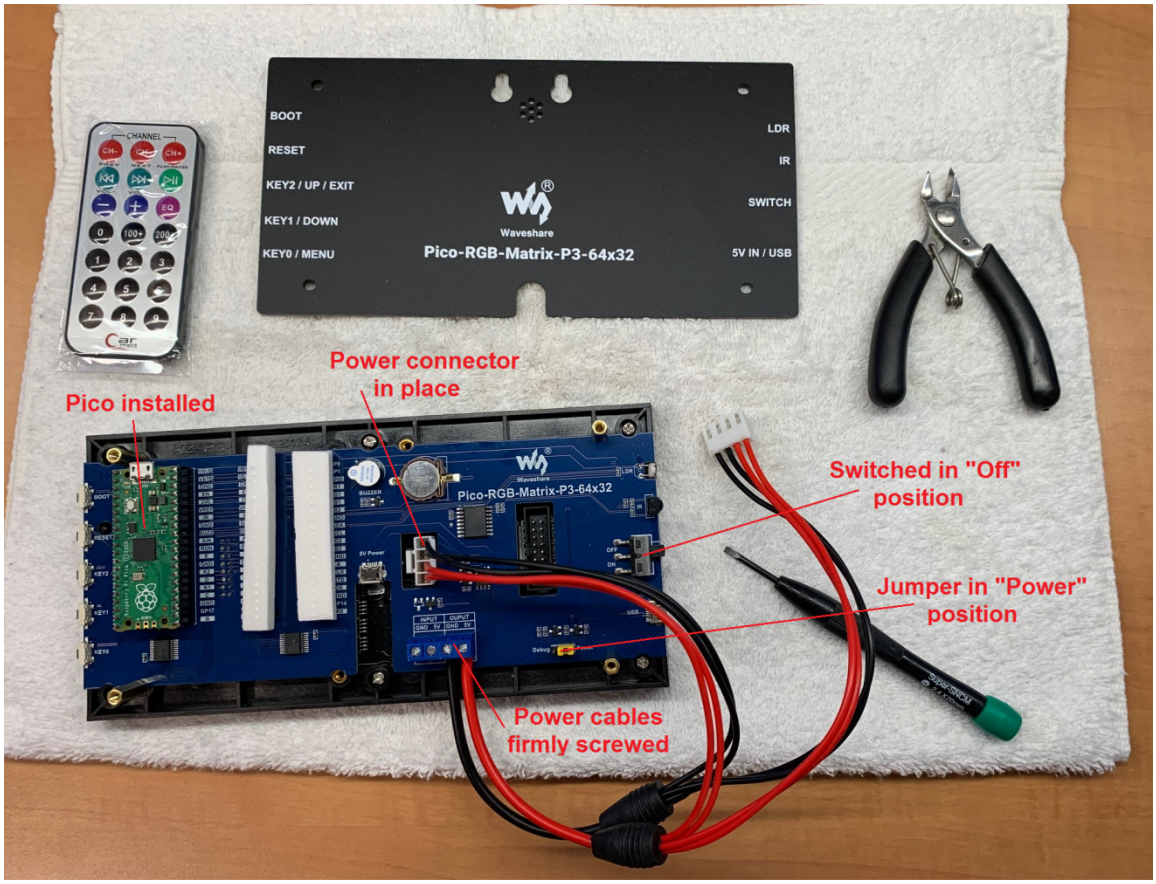


Preparation of power cable.

3) Insert the power cable that you just prepared in the right blue socket labelled « output » as shown on the picture below. Make sure the black cable is in the « GND » position and the red cable in the « 5V » position. Strongly screw both cables in place, then pull on them to make sure they are firmly hold in place by the screws. You will need a very small screwdriver to do this.

   Then, insert one of the other ends of the cable in the LED board, going through the control board. There is only one side the connector can be connected. The little pin on the side of the cable connector should be located on the plastic side of the connector on the LED board (see on the picture below).

   NOTE: The extra cable connector is a provision in case you plan to add an extra « hub75 » board, as discussed previously in the paragraph « Parts description ».



Power cable connection

4) For this step, refer to the picture in the previous step. Make sure the jumper (yellow on the picture) is installed between the middle and the « Power » position (and not between the middle and the « Debug » position).

   Make sure the power switch is in the « Off » position.

   It is now time to install the Raspberry Pi Pico in its socket. Before doing so, look carefully at the space between the two header connectors of the Pico. You can see three metal pins with a spring-loaded end. We will come back to discuss these pins in the next section: « How to upload a new Firmware Version », so it is a good idea that you keep in mind an « image » of those pins.

   As mentioned previously, the Pico is not supplied with the RGB Matrix and must been bought separately. Make sure the Pico's pins are properly aligned before pressing gently on it to insert it in its socket.

   Let's go on to the next section to understand all the details about uploading a new Firmware on the Pico...

How to upload a new Firmware Version

Introduction

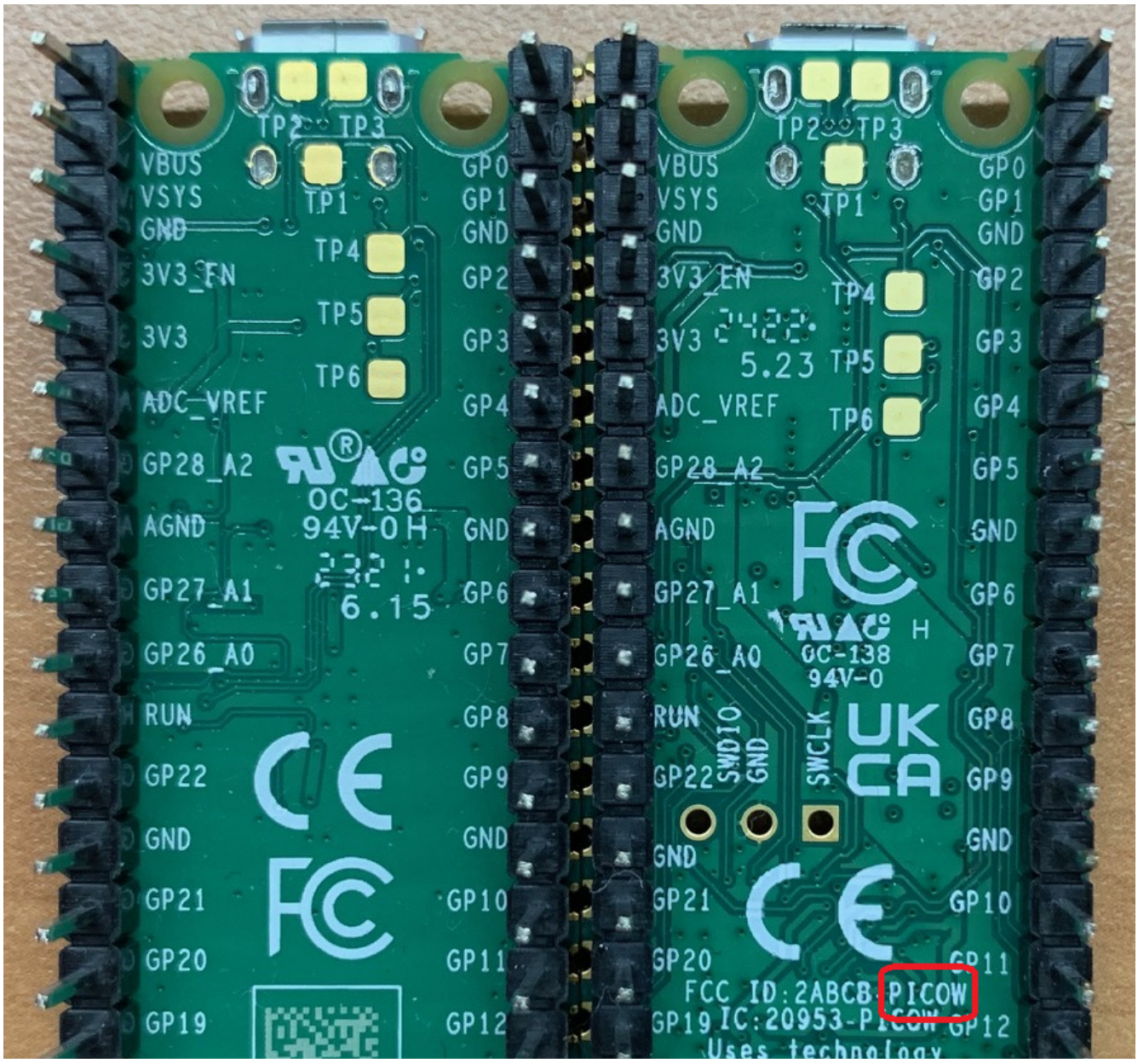
Make sure that you upload the Firmware for your particular microcontroller. If you use a Pico, use the Firmware for the Pico. You will not be able to take advantage of the Network Time Protocol and / or other future options requiring a Wi-Fi access.

If you use a PicoW, use the Firmware for the PicoW to benefit from all options requiring Wi-Fi (as of Firmware Version 2.00 there is only the Network Time Protocol option).

Difference between the Pico and the PicoW

If you just completed the « Assembling and cabling » of the RGB Matrix, it is a good time to take a look at a little glitch on the RGB Matrix.

Look carefully at the picture below. On the left side, you see a Raspberry Pi Pico, and on the right side, a Raspberry Pi PicoW. Do you see some differences?



In fact, the pad identified « TP6 » on the PicoW is slightly lower than the same pad number on the Pico. If you looked at the electronic board before inserting the Pico in its socket on the RGB Matrix, you've seen the three metal pins that have been put in place by Waveshare in the design. One of these pins (« TP6 ») is used to set the Pico in « Bootsel » mode (ready to upload a new firmware). As of this writing (March 2024), all the RGB Matrix that I bought were designed for the Pico. So, if ever you use a PicoW (which is better since you can take advantage of NTP and eventually future options that will require a Wi-Fi connection), the small difference in the pad « TP6 » position is such that the « Bootsel » button on the RGB Matrix device will not do what it is supposed to do and it will not put the Pico in its « Bootsel » mode.

It you do have access to the PicoW itself (that is, if the RGB Matrix back cover has not been installed yet), you may proceed as indicated below under the section for the Raspberry Pi Pico, except that instead of using the « Bootsel » button on the side of the RGB Matrix, you need to use the one directly

on the PicoW itself. Note that this problem is only for the « Bootsel » button. The reset button on the RGB Matrix works fine as it should for both the Pico and the PicoW.

If you do not have access to the PicoW itself (that is, if the back cover has been installed already), see below the section « Bootsel with remote control ».

On an RGB Matrix with a Raspberry Pi Pico

To upload a new Firmware Version, connect the RGB Matrix to the computer where is located the « .uf2 » executable file. Then, proceed as follow:

Press on the RGB Matrix « Reset » button and release it. As you will read in the next section (« RGB Matrix power-up sequence »), when you press the « Reset » button, you do have a short period of time (more or less two seconds) during which the RGB Matrix has been initialized and is blank (assuming that you already have a version of this Firmware flashed in your Pico).

During this two seconds period, press on the « Bootsel » button, and while keeping your finger on the « Bootsel » button, press again on the « Reset » button and release it.

Now release the « Bootsel » button and the Pico should be seen by your computer as a USB drive. Now, drag-and-drop the « .uf2 » Firmware file to the USB drive and the RGB Matrix Firmware will be downloaded and will start automatically after a few seconds.

You may need to practice a few times to become familiar with this procedure to blank the LED matrix before uploading a new firmware and prevent over bright LEDs.

On an RGB Matrix with a Raspberry Pi PicoW, first time with this Firmware

As mentioned above, if the back cover is not installed on your RGB Matrix, you can proceed as described in the section for the Pico above, but using the PicoW's « Bootsel » button (the button located on the PicoW itself) instead of the « Bootsel » button located on the side of the RGB Matrix.

If the back cover is installed and you're installing this Firmware for the first time, you're out of luck and you'll have to remove the back cover to have access to the PicoW's « Bootsel » button.

« Bootsel » with the remote control

If you are already running a previous Firmware Version (from A. St-Louys, not the original Waveshare's version), you can use the remote control and execute « Function 111 ». This function is named « Bootsel » and it is in the « Setup » category of functions. (Check in the User Guide if the number has changed in newer Firmware releases, or simply scan all « Setup » functions until you find « Bootsel »). Follow the instructions and this function will toggle the PicoW in the upload mode by software without the need to remove the back cover to have access to the PicoW's « Bootsel » button.

It is assumed that you know how to use the remote control. If you don't, you will need to read the specific section in the User Guide to understand what a « function » is and how to execute it.

You have now downloaded (« flashed ») the Firmware to the Pico microcontroller, so you are ready to see your RGB Matrix in action. Let's go on and learn everything we need to know about the power-up sequence.

RGB Matrix power-up sequence

You put together all the parts of the Pico-RGB-Matrix, you wired a powerful enough external power supply and you flashed the Firmware in the Pico. You're now ready to turn the power On. This section will tell you the important things that you need to know about the power-up sequence.

Step 1 – Turning power On

You may refer to the reference pictures previously in this guide to locate the position of the RGB Matrix power switch and buttons. Take note that, as opposed to the usual standard on most public devices, the RGB Matrix power switch is in the On position when it is pushed downward and it is in the Off position when it is pushed upward.

Once power is applied to the device, many internal initializations and tests are performed (but they only take a fraction of a second). The first important thing to remember is that as soon as the device is properly initialized, the LED matrix is positively and electrically blanked to make sure that all LEDs are in the Off position. Then, a delay of more or less two (2) seconds has been inserted on purpose in the startup sequence.

If you plan to upload new firmware often (maybe because you are a developer), this delay allows you to put the RGB Matrix in « Bootsel » mode without having some LEDs over bright while the matrix scan is stopped. When LEDs are left in such an over bright condition, we can expect their expected life span to be shortened significantly. So, remember that when power is applied, you have a short period of time to put the device in « Bootsel » mode (obviously, when you read « when power is applied » that means this 2-seconds period also appears as soon as you press the « Reset » button, since the reset is equivalent to applying the power to start a new power up sequence).

Step 2 – Starting terminal emulator software

After the 2-seconds delay, the device will show the message « Start emulator now ». As previously explained, a PC running a terminal emulator program is extremely useful to display information and proceed with the RGB Matrix configuration.

If you start the emulator program too soon, the RGB Matrix will not have been properly initialized yet and the emulator will not detect any device. If you start it too late, the RGB Matrix Firmware will assume that no emulator program has been started and it will run in its « standalone mode » without communication with an external PC.

So, the message « Start emulator now » indicates when it is time to start the program. You do have a few seconds to do so (more or less 10 seconds). If you get stuck in the emulator configuration and setup, you will miss the time window and you will have to turn Off both the RGB Matrix and terminal emulator and then: 1- Restart the RGB Matrix and 2- Restart the terminal emulator - to make sure the

connection handshake occurs during the right period of time (read next steps for more information about this).

Step 3 – Displaying date and time on terminal

If the Firmware detects a terminal emulator connected to CDC USB, it will display (on the terminal) the date and time it has retrieved from the DS3231 real-time IC. This may prevent problems in case funny results are found in the DS3231 and RGB matrix tries to display, say, month number 125 (depending on care spent on « parameter validation », this could cause a firmware crash).

Step 4 – Displaying date and time on LED display

The RGB Matrix will then « explode » the « Date window » at the top of the RGB Matrix, then « explode » the « Time window » at the bottom of the LEDs matrix. The « Time window » at the bottom of the display will keep its border visible, whereas the « Date window » at the top of the display will see its border vanishes after it is opened. Then, the date and time will be displayed for the first time in their respective windows. This information (date and time) is retrieved from the real-time clock IC (DS3231) integrated in the RGB Matrix. Of course, if this is the first time you turn On the power, both the time and date will be wrong and need to be adjusted. Otherwise, the battery backup should have maintained the correct time even if the RGB Matrix has been turned Off.

Step 5 – Blinking dark blue LEDs

Once the date and time have been displayed on LED display, you can see three dark blue LEDs blinking at the top left and at the top right of the LEDs matrix. Referring to step 2 above, about CDC USB connection of the terminal emulator program, those blinking dark blue LEDs indicate that RGB Matrix is still trying to connect with an external terminal. If a connection is successful, the Firmware jumps to a specific « terminal menu » from where user can navigate through many different menus.

Step 6 – Stopping dark blue blinking LEDs

More or less 10 seconds after the date and time show up on the display for the first time, the dark blue blinking LEDs will stop blinking (remain On) and later, become cyan (say, « light blue »). As soon as they stop blinking (and / or when they become cyan), it is too late to continue trying to establish a CDC USB connection with a terminal emulator program. As mentioned previously, if this happens and you still want to connect, you'll have to start over from the beginning. Otherwise, let the device continue and it will enter its normal clock behavior.

Step 7 – Connecting to Wi-Fi network

At this point, there are two (3) possible scenarios:

1) If you're using a Pico (as opposed to a PicoW): The RGB-Matrix will not try to connect to your Wi-Fi network and this step will be almost instantaneous.

2) If you're using a PicoW but this is the first time you turn On the RGB-Matrix: It will try to connect to an external time reference source over the Internet to synchronize its time (what is called « Network Time Protocol » or « NTP » for short). Obviously, if this is the first time you use the device, the credentials (network name and password) will not be initialized with the right credentials, **so this step will be relatively long (more or less 130 seconds for many unsuccessful retries)** and will end-up without a Wi-Fi connection (the RGB Matrix will nonetheless have a normal behavior, except for anything that has something to do with the Wi-Fi connection).

   Given what is mentioned above, you must be patient during the Wi-Fi initialization procedure. You can take a look at the PicoW's LED (the LED that is on the PicoW microcontroller itself that you can see even if the back cover is installed). While the device tries to establish a Wi-Fi connection, it will slowly blink the retry number (« 1 blink », pause, then « 2 blinks », pause, then « 3 blinks », pause, etc.). After a maximum number of unsuccessful retries (20 retries, in Firmware Version 2.00), the PicoW's LED will end-up by making 30 fast-blinks to indicate a Wi-Fi connection failure.

   When the RGB-Matrix gives up trying to connect to Wi-Fi after a failure, you will see that the two « double-dots separators » (between hours / minutes and between minutes / seconds) will be red, indicating a problem with your Wi-Fi connection. This statement remains true all the time (not only during the power up sequence). If, one day, you take a look at the RGB Matrix and you realize that the « double-dots separators » turned red, you should check either your Wi-Fi network, or the credentials that have been saved in the RGB Matrix.

3) If you use a PicoW but you did properly setup the network name and password before, this step should be relatively fast (more or less 5 seconds). As mentioned above, you can follow the Wi-Fi connection procedure by checking the PicoW's LED indicating what's going on. The LED slowly blinks each time it makes a retry and then, once the Wi-Fi connection is successful, it makes five (5) fast-blinks to indicate it. Be aware that it is normal to have 2 or 3 retries before establishing a successful connection (even more if the Wi-Fi signal is weak).

   When the procedure is over, you can see the outcome on the LED matrix. Take a look at the two « double-dots separators » » (between hours / minutes and between minutes / seconds). If the double-dots are green, this is an indication that the Wi-Fi connection has been successful.

NOTE: Refer to the section « Setting up Network credentials » in the User Guide to know how to proceed to enter the SSID (network name) and password to the Pico-RGB-Matrix.

Step 8 – Turning On Alarm and Target days alarm indicators

When done with the eventual Wi-Fi connection, the Alarm indicators (red or green dots at the top of the LED matrix) and also the Target days alarm indicators (small red or green lines between the date and the time box border) will be displayed. Those items are explained in the « Alarm » section of the User Guide.

Step 9 – Displaying main terminal menu

If you made a connection between the RGB Matrix and a terminal emulator, the Firmware will automatically bring you to the main terminal menu from which you can navigate through many different menus and this, before starting the main system loop. The first clock reading will be displayed, but at this point, it won't be refreshed on the LED matrix.

This has been done so that it is possible to insert specific functions in the source code that we want to execute before starting the main system loop. If you check the top box border of the « Time window », you will see yellow dots slowly appearing, meaning that the watchdog is counting (read « Watchdog section » in this User Guide for more details.

If you don't have any reason to proceed otherwise, you should press « ESC », followed by « Enter » to exit the terminal menu, turn Off the watchdog (until next time the main system loop is stopped) and then start the automatic refresh of the time display.

Step 10 – Scrolling the Firmware Version number

The RGB Matrix will now scroll the Firmware Version number on the LED matrix. It is now in its « cruise speed », all steps of the power up sequence have been completed.

If you did established a connection with an external terminal emulator program, there is more things to be aware of. You should press <ESC>, followed by <Enter> now on the terminal emulator and read the section about the watch dog.

NOTE: At this point, more information may scroll on the display. For example, if the Firmware has been built with the option to disable all sounds, a warning will scroll on the display.

RGB Matrix local buttons

There are five (5) small buttons on the right side of the RGB Matrix (if you look at it from the front of the LED matrix).

From top-down, they are called:

- « Bootsel » button – Used to toggle the Pico in upload-ready mode.
- « Reset » button – To restart the Firmware or to toggle the Pico in upload mode with the « Bootsel » button.
- « Up » button – Used by the Firmware, usually to increase a setup value (that explains its name « Up »).
- « Set / Function » - button (also called « Set » for short) Used to enter in « Function » mode and also used to execute the function being displayed.
- « Down » button – Used by the Firmware, usually to decrease a setup value (that explains its name « Down »).

It has been mentioned previously that the first two (2) buttons (« Bootsel » and « Reset ») are directly connected to specific connections of the Pico and consequently, they can't be used by the Firmware to execute specific actions other than those they are connected to. For this reason, when we use the terminology « Local buttons » in this User Guide, we mean the three other buttons (« Up », « Set » and « Down »).

The action taken by the Firmware when there is a press on these buttons is different if it is a « quick press » (less than one-third of a second), or if it is a « long-press » (more than one-third of a second). For this reason, you will read indications about a press on the « Set » button (meaning a quick press if not otherwise specified), or a « Set long-press » (meaning pressing the « Set » button for more than one-third of a second.

You will not find anywhere else in this document specific indications about the local buttons. You should simply remember that the local buttons simply mimic the six equivalent buttons on the remote control unit. (Since the duration of the press on the remote control does not change the infrared code being sent, there are three (3) remote control buttons corresponding to the « quick press » of the three (3) local buttons and there are three (3) other remote control buttons corresponding to the « long press » of the three (3) local buttons. The names of the remote control buttons clearly indicate what you must do on the local buttons to replicate the same behavior.

Using the RGB Matrix remote control

Introduction

Waveshare had the good idea to include a remote control unit with the RGB Matrix device. So, every user owns the same remote control unit. Adding functions in the Firmware to support this remote control would then benefit all users.

That being said, if you take a look at the remote control, you will quickly realize that it is really a basic unit. There are only 21 buttons and they are pretty simple. After thinking and analyzing how it could be possible to get the most out of it, I came out with the following implementation plan.

1) Every possible « action » to be done with the help of the remote control will be called a « function ».

2) The functions will be identified with a 3-digits code called « ID number ». This will allow for 999 different « actions » (ID number « 000 » is not used and it corresponds to an « undefined function »).

3) To make things easier, the functions will be classified in three (3) different groups or categories:

    a. Functions related to some form of « Setup » will have a function ID between 001 and 199. For example, « Setup » functions will be used to set the time, the date, to change the Time On or Time Off of the hourly chime, etc. Basically, this category of functions allows us to « change » or « modify » something on the RGB Matrix.

    b. Functions to get information back from the device will have a function ID between 200 and 399. Examples of function in this category are: display current settings for daylight saving time or time zone, display calendar events for this week, etc. We can « display » a setting, but we do not change it with the « Info » category of functions. At this point, it is useful to say that it is possible to have two different functions with the same name. For example, a function named « DST » with an ID number between 001 and 199 (« Setup » category) would allow us to change the settings of the daylight saving time, while another function named « DST » with a function ID number between 200 and 399 (« Info » category) would allow us to consult (display) the current DST-related settings.

    c. Other functions will have a number between 400 and 599. We will call them « Tools » functions. We can think of the « Tools » category of functions as functions allowing the RGB Matrix to perform duties other than a clock. In some way, those functions change the nature of the RGB Matrix. For example, in this category, we have the count-down and count-up timers, device integrity check, dice rolling, side-light, etc (most of these tools are not implemented yet in Firmware Version 2.00).

Infrared pilot on the LED matrix

During the development cycle, I implemented some kind of « pilot » on the LED matrix so that I could see when the device was receiving an infrared signal. Even when the Firmware development cycle for the infrared remote has been completed, I thought it would be a good idea to leave this pilot active.

When you press a key on the remote control unit, if you check the middle of the RGB Matrix, you will see LEDs changing color as the infrared stream goes through. The colors have the following signification: first color is blue and means that the infrared sensor has « seen » an infrared burst coming in. The color will then change, either to green, indicating that the infrared burst has been successfully decoded to be a recognized remote control button, either to red, meaning that trying to decode the infrared stream was unsuccessful. The infrared data stream goes very fast and there are some delays integrated in the logic of the pilot so that the colors are shown long enough that we can see what is going on. The LEDs return to their usual colors after only a few seconds.

This also means that if you use your television or radio remote control unit and you press a key toward the RGB Matrix, you will see the pilot LEDs changing colors. Chances are that the infrared burst of your television will simply not be recognized and the pilot LEDs will change from blue to red on the RGB Matrix, indicating that they have been « seen » but not decoded successfully (since your television most probably does not use the same infrared protocol than the RGB Matrix).

Description of the remote control unit

Refer to the remote control unit reference picture at the beginning of the guide to follow the instructions below.

The most important buttons are probably those on the first two rows of the remote control unit. They are called (from left to right): « Down », « Set / Function » and « Up » for the first row, and « Down long-press », « Set long-press » and « Up long-press » for the second row.

The first row reproduces the behavior of the local buttons on the RGB matrix, while the second row reproduces a « long-press » on the local buttons on the RGB Matrix. So, care has been taken to keep the same names between the remote control unit buttons and the RGB Matrix local buttons.

The « Digits » buttons on the remote control are also used (we will see how later).

For now, the other buttons on the remote control unit are not used and are reserved for later use.

Executing a specific function

We already explained above how « Functions » are used to define actions that we can have on the RGB Matrix. So, how can we execute a specific function?

To access a function ID, the user simply presses the « Set / Function » button on the remote control unit (« Set » will be used from now on but it means the « Set / Function » button). The infrared sensor is located on the left side of the RGB Matrix, more or less in line with the middle of the date line (refer to the reference pictures at the beginning of the Guide if you want more precision). You may want to hold the remote control unit toward this direction when you send a command. Also, make sure there is no object between the remote control and the infrared sensor, otherwise, the remote control would simply not be recognized by the RGB Matrix.

When the « Set / Function » button is pressed, the LED matrix will open a special « Function » window that is used in « Function » mode. By default, function ID 200 is suggested (category « Info »). You will see « F-200 » on the second line of the top window, and the first line of the top window will scroll the suggested function name « Firmware Version ». We know that functions ID between 200 and 399 are « information » functions. So, we can expect the Function ID 200 to display the Firmware Version number (which is exactly what the functions does)

There are a few different actions that you can take at this point:

- Let the system idle for more or less 30 seconds, until it reaches the limit of a timeout period, after which it will simply return to the usual date and time display mode.

- Press the « Up » button on the remote control unit to jump to the next valid function ID (jumping to the next function ID doesn't always mean +1, depending on the functions that have been defined in the Firmware). When reaching the last valid function ID, the system jumps back to the beginning (Function ID 001).

- Press the « Down » button on the remote control unit to jump to the previous valid function ID (jumping to the previous function ID doesn't always mean -1, depending on the functions that have been defined in the Firmware). When reaching the first defined function ID (Function ID 001), the system jumps to the last defined Function ID.

- Press the « Set » button once more to enter (« execute ») the function ID currently being scrolled / suggested. You don't have to wait until current scrolling is complete. As soon as you press the « Set » button, the scrolling will be truncated and the function will be executed.

- Press the digit buttons on the remote control to sequentially enter the function ID and then access a specific function ID, possibly faster than by scanning every function with the « Up » and / or « Down » buttons.

  It is important to understand how entering Function ID with digit numbers works, since it is used for other algorithms in the RGB Matrix. Suppose that the function ID « F-200 » is displayed on the LED matrix. What you want is, say, function ID « 341 ». You first press on digit « 3 » (corresponding to the first digit in « 341 »). This digit « 3 » will appear at the right of the current proposed function ID, while the previous digits will be shifted left. So, « F-200 » becomes « F-003 » since the previous « 2 » is shifted left and disappear, the two « 0 » in the previous « F-200 » are also shifted left, and finally, the « 3 » that you just pressed appears on the right of the function ID (« F-003 »). The same way, you will then press the second digit of the function ID that you want to reach (corresponding to digit « 4 » in our example for function

« 341 »). In a way similar to what is described above, the current proposed function ID « F-003 » will be shifted left by one position, while the digit you just pressed (« 4 ») will appear at the right « F-034 » will be displayed. Finally, you enter the final digit « 1 » in our example for function « 341 ». Without surprise, the current « F-034 » will be shifted left by one position and the last digit (« 1 ») will be added to the right. The display will show the function ID that we want « F-341 » and the function ID will scroll on first line of the display. (NOTE: We used « 341 » as an example, but this is an invalid function number for now. Don't be surprised to see « Invalid function number » on the display if you actually followed the example above.

You can press as many digit buttons as you want. The same process will repeat indefinitely (current digits shifted left by one position and new digit added to the right). So, if you made a mistake at some point and entered the wrong digit, you can simply continue entering the correct digits and you will end up with the desired function ID.

You will notice that every time you enter a new digit, an « intermediate » function ID is temporarily created. In our example, when we press « 3 », which is the first digit of the function that we want to access, we've seen that the display will temporary indicate « 003 ». Depending if this function ID (« 003 ») is valid or not, the corresponding function name will be scrolled on the display.

NOTES:

- In this Firmware Version 2.00, many functions are planned but not implemented yet. Most programmed functions are in the « Info » category for now.
- As mentioned on my GitHub repository, I encourage you to report « bugs », but needless to report missing features, this is a « work-in-progress ». For example, most « Setup » functions with the remote control unit remain to be implemented in the Firmware. However, you can use an external terminal emulator to fine tune the RGB Matrix as you want. (Data validation also needs to be added in many situations).
- Function IDs may change in future releases.

Planned functions are listed below. There is also an indication so that you can see if the function is currently available in current version or if it is planned for a future release.

Functions of category « Setup » (Function IDs 001 to 199)

Function ID 001 – Time set

When you enter the « Time set » function, the display shows « Please set hour » and the current hour setting will blink. Press the « Up » or « Down » buttons to increment or decrement the current hour setting, or enter the hour setting using the remote control digit buttons. When done, press the « Set » button to switch to « Minute » setting. Proceed the same way to set minutes. Press the « Set » button again to set seconds.

You can toggle as many times as you want between « Hours », « Minutes », « Seconds » setting.

When you're done with time setting, press the « Set long-press » button to save the new time setting.

Function ID 002 – Date set (to be implemented)

Function ID 003 – Alarms settings (to be implemented)

Function ID 004 – Network settings (to be implemented)

Function ID 005 – Brightness settings (to be implemented)

Function ID 006 – Calendar events settings (to be implemented)

Function ID 007 – Hourly and half-hour chimes settings (to be implemented)

Function ID 008 – Daylight Saving Time (DST) settings

First, use the « Up », « Down » or digit buttons to set the « Country code » for the Daylight Saving Time parameters used in your country (refer to Addendum C in this User Guide). Then press the « Set » button to enter the « Time zone » setting. Use the « Up » or « Down » buttons to set the current difference between you local time and UTC time.

You can toggle as many times as you want between « Country code » and « Time zone ».

When you're done, press the « Set long-press » button to save the new Daylight Saving Time settings.

Function ID 009 – Golden age settings

In Quebec, we call « Golden age » the period of the life when we stop working to take benefit of retirement (I think the term « senior » is more common in English). Unfortunately, it is also sometimes the period where cognitive problems begin to show up. A friend of mine asked me if the RGB Matrix could be used to help in such a case. The « Golden age » option has been designed for this purpose. Basically, it is an On / Off options. When On, the following behaviors of the RGB Matrix will change:

- During the day, the display will be yellow (color of the Sun).
- During the night, the display will be dark blue (color of the night).

- The date (2nd line on the RGB Matrix) will alternate every 5 seconds between today's date and the period of the day (« Morning », « Afternoon », « Evening », « Night »).

It is also possible to configure the « Hour » when we want to change the display from « Afternoon » to « Evening », from « Evening » to « Night », etc... (Only « Hour » may be set for that purpose. Not « Minutes ». Also, a terminal emulator program is required to change this setup).

My friend also used the flexibility of the alarms on the RGB Matrix to setup specific alarms for her father so that he knows when it is time to go to the dining room for the meals in his senior residence.

Function ID 111 – Bootsel mode

If you read the section about uploading a new firmware version, you already know that there is a problem with the PicoW that prevents it from falling to the Bootsel mode with the RGB Matrix « Bootsel » button.

This function is there to help you. Press the « Set » button to enter the function select mode, then, use the « Up », « Down » or digit buttons to go to Function ID 111 (« Bootsel »). Press « Set » to execute the function, then « Set long-press » to confirm and the PicoW will automatically enter its Bootsel mode.

Functions of category « Information » (Function IDs 200 to 399)

Function ID 200 – Firmware Version number

This function will scroll the current Firmware Version number of the RGB Matrix.

Function ID 201 – Type of microcontroller

This function will scroll the type of microcontroller installed in your RGB Matrix (Pico or PicoW).

It will also scroll the 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 or PicoW installed in your RGB Matrix). As you can guess by its name, this number is unique to the Pico that is installed in your device. If you would replace it with another Pico, the Unique ID would be different.

Function ID 202 – Temperature

This function will display the internal temperature of the Pico (or PicoW). It will also display the ambient temperature, based on the DS3231 internal circuit (real-time clock IC), since this IC uses electronic circuitry to compensate for time drift due to external temperature changes.

Function ID 203 – Brightness current settings

This function will scroll current settings related to the LED matrix display brightness.

Function ID 204 – Current alarm settings

This function will display the current settings of active alarms.

Function ID 205 – Calendar events for today

This function will display all Calendar events defined for today.

Function ID 206 – Calendar events for current week

This function will display all Calendar events defined for current week (week beginning on the most recent Sunday).

Function ID 207 – Calendar events for current month

This function will display all Calendar events defined for current month.

Function ID 208 – All Calendar events defined in the system

This function will display all Calendar events defined in the system.

Function ID 209 – Daylight Saving Time current settings

This function will display the settings related to the Daylight Saving Time (Country code setting and current Time zone). You may refer to the Appendix C in the User Guide for more information about these settings.

NOTE: The Time zone is currently used to properly handle the Network Time Protocol received from Internet. The DST Country code is not used in Firmware Version 2.00. Currently, when there is a change from Summer time to Winter time, the Time zone must be changed manually on the RGB Matrix.

Function ID 210 – Network credentials

This function will display the network credentials saved in Pico's flash (non volatile) memory.

Function ID 211 – Network information

This function will display network-related useful information (except network credentials).

Function ID 212 – System idle time monitor (to be implemented)

Function ID 213 – Silence period (to be implemented)

Function ID 214 – First free heap memory pointer

This function will display the next free heap memory pointer in Pico`s memory. This is basically to help developers checking if there is a heap memory leak in the code.

Function ID 215 – Active Auto-scrolls

This function will display all Auto-scrolls that have been defined in the system.

Function ID 216 – Total system up time

This function will display the total time the RGB Matrix has been up and running since the last reboot.

Functions of category « Tools » (Function IDs 400 to 599)

Function ID 401 – Count-up time (to be implemented)

Function ID 402 – Count-down timer (to be implementd)

Using an external terminal emulator

Using a personal computer (« PC »), running a terminal emulator program (for example « TeraTerm ») is the best way to reach the full potential of the RGB Matrix. Moreover, in current Firmware Version, many features are not available / reachable with the remote control unit and / or the local buttons. Even after full Firmware deployment, many features available from terminal menus display too much information to be used through the LED matrix. So, it is a good idea to get familiar with the terminal menus, if only to have an idea of what features are available in case you want to use some of them later.

Refer to Appendix A if you need more information on how to setup a terminal emulator program to use it with the RGB Matrix.

Once you install the terminal emulator program on your PC, you may refer to the « Power up sequence » section of the User Guide to understand the timing constraints to observed when connecting and starting the terminal emulator program.

When the RGB Matrix recognizes the terminal emulator connected to the USB port, it will automatically jump to a « Generic terminal menu ». From there, you can access submenus. If you already read the section about remote control functions, you won't be surprised to read that there are three main submenus: « Setup » submenu, « Info submenu » and « Tools » submenus.

There are also more options available from the main terminal menu and they are not available from the remote control since they wouldn't make sense. For example, displaying the complete configuration written in flash would scroll information forever on the LED display. Same thing if we decided to display a significant part of memory content...

As you will see in the « Watchdog section », while you navigate through the different terminal menus, the main system loop is on hold since you jump from this loop inside the terminal menus and submenus. That means that while you navigate through the different menus, the watchdog cumulate the time the endless loop is inactive and it will restart the system if it becomes more than five minutes. This is usually enough to perform many settings and display information. So, you must check the yellow LEDs appearing in the top border of the Time window (in the middle of the matrix). They will converge toward the center of the matrix. When they meet together, the Firmware will restart.

To prevent a restart, press « ESC », followed by « Enter » as many time as required in order to get out of the menus. As soon as you do that, the watchdog will be satisfied and it will reset the display to the usual red LEDs. If you want to return to the terminal menus, simply press « Enter » again. The main loop will detect that you pressed « Enter » and it will jump again to the main terminal menu.

Depending on your Windows version, if ever the watchdog restarts the RGB Matrix, you may lose the connection with terminal emulator. I've seen version of Windows where the connection is automatically restored whereas with other systems, the connection is lost and it is required to restart both the RGB Matrix and TeraTerm with the correct timing to re-establish the communication.

Understanding the RGB Matrix features

Alarms

The RGB Matrix features nine (9) independent alarms. Each alarm can be programmed for a specific target time (that is: hours and minutes) and also for any number of target days. For example, you can set alarm number 1 for a particular wake-up time for, say, Monday, Wednesday and Friday, and you can program alarm number 2 for another particular wakeup time for Tuesday and Thursday.

The following items may be programmed for each alarm, after you selected the alarm number (from 1 to 9):

- Alarm status (On / Off).
- Target hour.
- Target minute.
- Target days-of week (any number).
- Message (text to scroll when alarm rings, up to 40 characters).
- Number of scrolls (number of timer the message will scroll on the display when alarm rings).
- Number of beeps (number of beeps when the alarm rings).
- Beep duration (in msec, duration of each « beep » when alarm rings).
- Repeat period (in sec – alarm will ring every so many seconds) .
- Ring duration (total duration of repeat period during which alarm will ring).

Alarm indicators

When you setup alarms, you are asked if you want to display the alarm status on the LED matrix. If you do (and you should), you will see on top of the LED matrix 9 little indicators, each one made of 2 colored dots. Each indicator corresponds to alarm status, from alarm number 1 on the left, up to alarm number 9 on the right. If the indicator is red, it means that the alarm status of its corresponding alarm number is Off. If the indicator is green, it means that the alarm status of its corresponding alarm number is On.

Days-of-week indicators

When you setup alarms, you are asked if you want to display the target days-of-week status on the LED matrix. If you do (and you should), you will see under that date (just above the time window's top border) 7 indicators, each one made of 4 colored dots. Each indicator corresponds to a specific day-of-week, from Sunday (on the left) up to Saturday (on the right). If a specific day-of-week indicator is red, it means that there is currently no alarm On for this specific day-of-week. If a specific day-of-week indicator is green, it means that there is currently at least one alarm On for this day-of-week. If more than one alarm is On, you can see which alarm is set for which day, by requesting « Alarm info », either from the « Info » terminal menu or from the corresponding « Alarm » function (in the « Info » category).

NOTE: Alarm sound does not comply with Hourly Chime Settings. That is, alarm will sound no matter what is the setting for Hourly Chime.

NOTE: If “NO_SOUND” is defined in the source code at compile time, ***absolutely no sound will be heard from the RGB Matrix*** (no Keyclick, no Hourly Chime, no Alarm, …). The message “WARNING: ALL SOUNDS DISABLED” will scroll on the LED display during the power up sequence.

Auto-brightness

As opposed to the original Waveshare original Firmware, Firmware Version 2.00 and up controls the brightness of the LEDs with a pulse-width-modulation signal (« PWM ») from the Pico. This PWM signal is generated by a specific electronic part of the Pico, so that the « microcontroller » part of the Pico does not have to take care of it (in other words, the software does not have to care about all the timings required to provide the display brightness). This allows for a very large number of different brightness levels.

The auto-brightness feature allows the LED matrix display to be brighter when the ambient light is bright and to dim the LED display when the ambient light is darker. Take note that there is 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 (the LED display will not change back and forth very quickly as soon as something runs in front of the device's sensor).

The ambient light sensor takes advantage of one of the analog-to-digital converter integrated in the Raspberry Pi Pico. Since the precision of the Pico's analog-to-digital converter is 12 bites, the ambient light value will be in the range of 0 (very dark) up to 4096 (2 ^ 12 – very bright). Keep in mind that the LED display may become ***very*** dim if the ambient light is dark (it also depends on the auto-brightness settings). Remember that when you turn On the light, the LED display will gradually increase its brightness, due to 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).

There are four different settings available for the RGB Matrix brightness. The first setting is to determine if you want the LED matrix to be always with the same brightness or if you want it to vary with the ambient light (being brighter when the ambient light is bright and vice-versa). This setting is called the « Auto-brightness » and is simply On or Off.

The second parameter is used when the Auto-brightness mode is On. It is the « Low » parameter and it allows you to determine « how low » you want the LED display to go when the ambient light is dark. For example, if the device is installed in a bedroom, you may want to set it to the lowest possible value (« 1 »), so that there is not much light in the bedroom during the night. For some other purposes, for example in the kitchen, you may want to set the lowest value a little higher (say « 3 » or « 4 »), so that the RGB Matrix will become a night light if someone has to go in the kitchen during the night.

Keep in mind that when the RGB Matrix is in its lowest brightness, if you look at it with « normal » ambient light, you may think that it is turned Off since its display becomes very dim.

Without surprise, the third setting is also used when the Auto-brightness mode is On. It is the « High » parameter. It allows you to determine « how bright » the LED matrix will go when the ambient light becomes bright.

Finally, the fourth parameter is used if / when you turn Off the Auto-brightness mode. It is called « Steady » and it determines the LED display brightness to use when the Auto-brightness is turned Off.

Auto-scrolls

It is possible to configure up to five (5) Auto-scrolls on the RGB Matrix. Basically, an Auto-scroll if a list of information data that you want to scroll on the RGB Matrix at specific intervals.

When you setup an Auto-scroll, you must first select which one you want to use. There are 5 independent Auto-scrolls available, you would usually begin with number 1, then 2, etc…

You then enter the time interval (in minutes) at which the Auto-scroll will become active. For example, if you set the time interval to ten (10) minutes, the data will scroll on the matrix at HOUR:00, HOUR:10, HOUR:20, etc…

Finally you select from a predefined list of items which ones you want to scroll on the display. Be aware that more items may be added in the future. You can define up to 32 items to be scrolled each time the time interval is reached.

NOTES:

- Be aware that if you put so many items to scroll each time (and a too short interval), items will be added at the end of the current scroll queue if it is not already done scrolling the previous one. You may definitely expect strange behavior in such a situation.
- If the RGB Matrix is not in the usual « Time and Date » display mode, the Auto-scroll will be postponed until the Firmware returns in the « Time and Date » display and will then be executed.
- As mentioned above, the Auto-scroll will not execute, say « every 10 minutes » but rather at HOUR:00, HOUR:20, etc. So, if a 10-minutes Auto-scroll has been postponed to, say 10:12, the next one will resume at 10:20 (not at 10:22 as you would expect for a 10-minutes period). Auto-scrolls have been programmed this way to be deterministic. If you configure one for 10-minutes interval, you know that it will start at HOUR:10, HOUR:20, etc...
- Without going through technical details, be aware that Auto-scrolls are triggered by a callback functions but executed in the main Firmware thread context. So, there will be a quick delay (a few seconds) before Auto-scroll execution begins.

Calendar events

A good example for a « Calendar event » is a birthday. You can program in the RGB Matrix up to 64 dates that have a special meaning for you. When the specified date is reached, the device will sound a few beeps twice an hour and at the same time, will scroll on the screen the text that you entered while configuring the Calendar event (up to 40 characters long).
(NOTE: Sometimes, the single word « Event » is used in the source code and / or in the Firmware and / or in this User Guide. It is simply for short and means « Calendar event »).

For example, « John's Birthday » would scroll on the LED display on April 14th, if this date (14-APR) has been defined in the Calendar events with the text associated. Up to 64 such calendar events may be programmed and more than one may be programmed for a specific day. The text will scroll twice an hour all day long, 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 and half-hour 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, the warning beeps will sound during daytime, as defined between "Chime Time On" and "Chime Time Off" in the configuration (set by default from 9h00 to 21h00). See also section about « Hourly chime » in this User Guide.

There are provisions to play a specific jingle if a passive buzzer has been installed by user. However, this part is not implemented yet.

A few of these Calendar events have been programmed by default to show the user how to program more of them 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.

If you want temporarily put a Calendar event on hold, simply set its day number to 0. No validation is done on the date. So, if the day number is set to 0, the target date will never be reached and this event will never be scrolled by the system. In a similar way, if an event is configured for 29-FEB, it will be scrolled only every leap year.

Configuration settings

As of Firmware Version 2.00, most configuration settings must be done with the help of a PC running a terminal emulator program. You should read the section « Using an external terminal emulator program » in this User Guide to know how to proceed to use it.

Once you have access to the different terminal menus and submenus, you shouldn't have problems to navigate through the menus and find what you are looking for.

Keep an eye to the yellow dots on the LED matrix while you navigate the menus so that you won't be surprised by the watchdog.

You may also take a look at the « Functions » section that shows what are the settings available with the remote control unit (they are very few in Version 2.00, but as future versions are released, more settings will become available from the remote control).

Count-down timer

There are plans to include a count-down timer in the RGB-Matrix. However, as of Firmware Version 2.00, the count-down timer implementation is not completed yet.

Count-up timer

There are plans to include a count-up timer in the RGB-Matrix. However, as of Firmware Version 2.00, the count-up timer implementation is not completed yet.

Daylight Saving 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

It is planned that the Firmware will automatically supports Daylight Saving Time for most (if not all) countries of the world. This has not been fully implemented in Firmware Version 2.00. However, it is required that you properly setup the Time zone for the Network Time Protocol « NTP » option to work fine.

NTP will receive the current UTC (« Universal Time Coordinates ») from the Internet. As its name implies, this time value is universal (is the same for everyone, no matter where you are on the planet). In order to properly interpret this time and adjust the RGB Matrix with the local time, it is required to know in which time zone you are located. So, if you use a PicoW and the NTP option is turned On, it is mandatory to setup your local Time zone. Otherwise, your clock will indicate the wrong time.

Also, in Firmware Version 2.00, automatic handling of Daylight Saving Time has not been implemented yet. That means, if we go through a time change without this option being implemented in the Firmware, it will be required to manually change the Time zone twice a year. It is useful to understand that the only thing done by the automatic DST handling is to change the Time zone. The clock takes care of the current time zone when updating its display.

Device integrity check

The « Device integrity check » is an option accessible either from the remote control (in the « Tools » category of function), or from the terminal menus.

This option will sequentially scroll all rows, then all columns of the LED matrix and this, using all color combinations. When launched from the remote control, the functions executes unattended, whereas it will require the user to press « Enter » between each scan if it is launched from the terminal menu.

Digits

The usual character set used to display data on the LED matrix is a « 5x7, variable width » character set. This make the display more elegant (« better-shaped ») than the usual 7-segments displays that we see with other clocks available on the market.

Another « 8x10, variable width » character set has been partially implemented (not complete as of Firmware Version 2.00). This one is used to display bigger digits for the Time display. Even if it is defined as « variable width », I realized that using this feature and changing the space between digits in the time display while it is updated every second is very annoying. So the potential of variable-width is not used for the time display (in fact, only the numeric digit « 1 » could benefit from this feature.

Another 4x7 character set could be made available. It is not used for now and while I used it for other projects, it is sometimes ugly with some character. It is there in case it could be useful in some occasion. (Bitmap will probably need to be reversed to be used with the pixel display engines that I implemented since I wanted a more intuitive way to draw the characters than what was available in the original firmware version).

Display colors

It is relatively easy to change display colors if you want to do so. Simply check the « win_init() » function and change the colors (border and inside) of the specific window you want.

However, while designing the Firmware, I did test many color combinations before coming up with the final setting. Here are a few things to have in mind if you want to play with colors:

- Check the RGB Matrix power requirements in Appendix B. You will see that LED colors have a significant impact on power dissipation. Blue is the color requiring the least power, followed by green, and then red. If you plan to leave your RGB Matrix powered up most of the time (it is the usual behavior for a clock!), you may want to use a color combination using low power dissipation.
- Even if blue is the color requiring the least current, I realized that in very dark conditions (that is, when the RGB matrix brightness is at its lowest level), blue is hard to read, either for the date, either for the time, while the other part of the matrix, if another color is used will be easily readable. That's why blue is not used in the default setting.
- The display matrix is made of tri-color LEDs. Each LED contains a red, green and blue light source. You can use any combination of the basic colors to produce a total of 7 colors (let's say that « black » - no color is not a « color » for the current discussion. When trying different combinations, I've seen that some colors are not pleasant because you kind of see the two colors combinations instead of seeing only one resulting color. For example, yellow (green and red) and magenta (red and blue) are bad. As opposed to that, cyan (green and blue) is more homogeneous since we don't see the blue and the green as separate colors when we look at it. That's why, in fact, cyan has been used as a default color for the Date window. Nonetheless, the other combined colors have been used for windows that are used less often (Function window, and others).

Double dots separators

There are two « double-dots » in the Time window to separate hours and minutes and also to separate minutes and seconds. Their color is usually green. If you use the « Golden age » option, their color will change from yellow (color if the sun during the day) to dark blue (color of the dark sky during the night).

If ever there is a problem with your Wi-Fi connection, these double-dots will become red to highlight this problem. If this happens, you should check the status of your Wi-Fi network and / or the credentials that are currently saves in your RGB Matrix flash memory.

Firmware endless loop

For those familiar with real-time firmware and / or embedded systems, you probably know already that the firmware is usually an endless loop reading switches and / or other devices and acting on some other devices, while hardware and software interrupts may also be active.

The RGB Matrix is no exception: there is an endless loop in charge of different duties, while callback functions (« Software interrupts ») are in charge of other time-sensitive duties.

One thing that may not be intuitive with the RGB Matrix is that, even if the « endless loop » main thread crashes, the RGB Matrix will continue to display the date and time as usual, since the date and time are updated by the independent real-time IC (DS3231), while the LED display is refreshed by callback functions.

During the development cycle, it happened a few times that the main thread crashed. I decided to integrate a way to know if the main thread is still active and I thought it could be a good idea to let this feature active in the released Firmware.

So, while the clock is running in its usual mode (date and time display mode), you will see three blinking cyan pixels at the top left and at the top right of the display. As long as these pixels are blinking, it is an indication that the Firmware endless loop works fine. If you see that the pixels stop blinking (being either On or Off), it is an indication that the endless loop is stopped for some reason.

Be aware that it is normal for the endless loop to stop looping when you navigate through the terminal menus (with the external terminal emulator program).

Read also the section about « Watchdog ».

Flash memory

Most configuration settings of the RGB Matrix are saved in Pico's flash (« non volatile ») memory. During the power-up sequence, the configuration is read back from flash memory and restored to the RGB Matrix. If this is the first time that the new Firmware is run, 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 device configuration.

Every once in a while, the Firmware will compare the checksum of its current active configuration with the configuration saved in flash. If the user has made some change to the active configuration, it will be detected by the Firmware (checksums won't match) and the new active configuration will be saved to flash. (NOTE: In some occasion, the change in the active configuration may originate from the Firmware itself).

Since updating the Pico's flash requires to disable interrupts (which temporarily stops LEDs scanning), to prevent unpleasant behavior on the LED display, if / when a new configuration needs to be flashed, a warning message will appear on the display « Flashing 1 » (or « Flashing 2 », depending which flash page needs to be updated). Then the display will be blanked for a second and returned to normal.

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 device 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 RGB Matrix configuration is saved to flash only a few seconds after the Firmware is back in the main system loop (that is in the « Date and Time » display mode). This means that if user navigates through and changes many configuration parameters while navigating through the terminal menus, chances are that only one flash write will occur, once user exit all terminal menus and go back to time display mode.

Heap memory pointer

While developing the Firmware, I wanted to check the free memory heap pointer to make sure there was no memory leak for a specific algorithm. I left some functions in the code that allow checking the free memory pointer for a typical chunk size of memory. We expect this value to remain the same for a given Firmware Version, indicating there is no memory leak. Typical memory addresses are given below:

What kind of variables	Address sample
Global variable pointer	0x200061E4
Local variable pointer	0x20041F52
Heap variable pointer	0x20041F54

Hour display mode

It is planned that the hour display mode can be set in 24-hours format and / or 12-hours format. However, as of Firmware Version 2.00, the 12-hours format is not implemented yet.

Hourly chime and Half-hour light chime

The « Hourly chime » is an option that can be turned On, Off, or Intermittent. When turned On, the system will sound a few beeps every time the hour changes (from HOUR:59 to HOUR:00).

The « Half-hour » light chime is also an option that can be set the same way as the Hourly chime. It is called « light » because the there are fewer beeps for the Half-hour light chime than for the Hourly chime.

The « On » and « Off » settings are self-explanatory. However, the « Intermittent » option (also called « Day » setting) allows to set « Time On » and « Time Off » to specify a time at which the chimes will beginning in the morning and a time at which the chimes will stop in the evening. This will prevent sounds to be heard during the night, when people are sleeping (Time On and Time Off are set respectively to 9h00 and 21h00 by default).

Be aware that the « Ringing » (sounds) for the Calendar events follow the same rules that the hourly chimes (On, Off, or Day). In the case of Calendar events, however, even if the sound is Off (because it is out of the Time On and Time Off period of the Hourly chime, the message that has been configured with the Calendar will scroll on the LED display.

Idle time monitor

Some parts in the source code have already been planned for a system « Idle time monitor ». This can be seen as the opposite of a « System load monitor ». The highest number of the « Idle monitor », the least busy is the system.

As of Firmware Version 2.00, the « System idle monitor » is not completely implemented and will be in a future release.

Infrared audio feedback

An audio feedback sound may be turned On or Off in the configuration to give some feedback when the Firmware receives an infrared data stream that is successfully decoded.

This may be used, along with the LED matrix infrared « pilot » to give a feedback of what is going on. It must be understood that for the audio feedback to be heard, the infrared data stream must not only be received by the Firmware, but it must also be recognized as a valid remote control button. So, do not try to make it sound with your television remote control! It is most likely using a protocol that won't be recognized by the RGB Matrix (however, you will see the pilot changing color on the LED matrix since it will « see » an infrared data burst coming in).

Infrared pilot

When the infrared sensor of the RGB Matrix receives an infrared burst, it will automatically indicate it on the LED matrix. If you look at the center of the matrix, you will see a few LEDs changing colors. First, these LEDs will be turned On in blue color to show that an infrared data burst has been detected. Then, as the Firmware tries to decode this infrared burst, the LEDs will change from blue to green (to indicate that a remote control button has successfully been recognized and decoded), or from blue to red (to indicate that the infrared data burst has not been recognized as a valid remote control button).

It must be noted that an infrared data burst is really quick. The timings of the LEDs that you can see on the matrix have been slowed down so that you can have enough time to see the blue and green (or red) colors.

Languages

Two languages have been implemented so far in the RGB Matrix Firmware: English and basic French.

Provisions have been made to easily add new languages by creating more « lang-xxx.h » translation files.

IMPORTANT: Since this is the first Firmware Version that I make public, there must be a lot of changes, modifications to the code before it becomes relatively stable. For this reason, I strongly suggest that for now, you simply copy the « lang-english.h » file to your new language name and simply translate the current content without adding new strings. As the Firmware evolves, more parts of the Firmware will be added to the language files. If you decide to add more stuff in your language file, you will need to support it as new Firmware versions are released.

For now, when a string does not exist (is not defined) in the language file, English is used instead.

Local buttons audio feedback

An audio feedback sound may be turned On or Off in the configuration to give some feedback when the User presses a local button. If this option is turned On, a quick « sound » will be heard each time a local button is pressed to give the user a positive feedback that the button press has been properly received by the Firmware.

As already mentioned previously, the « local buttons » are defined as the three buttons at the bottom of the RGB Matrix. The two top buttons (« Bootsel » and « Reset ») are electrically connected to some Pico's parts and they are not recognized by the Firmware. So, the audio feedback (if turned On), will only be heard for the three (lowest) local buttons recognized by the Firmware.

Network Time Protocol

Network Time Protocol (« NTP » for short) allows the RGB Matrix to synchronize itself with a time reference server over the Internet. A « PicoW » 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 PicoW's LED which doesn't turn On or Off the same way as with the Pico's).

The first time you turn On the Pico-RGB-Matrix with the NTP version of the Firmware, it will take some time (up to 2 minutes) before the clock starts. This is because the Firmware will try many times to connect with the Wi-Fi. Since this is the first power-on (network name and password have not been setup yet), many unsuccessful retries will be done (up to 20 in Firmware Version 2.00). You can take a look at the PicoW's LED to see the sequence of the retries.

Once you setup the network name and password with the terminal emulator program, get out of the terminal menus and wait (more or less 30 seconds) until you see the « Flashing 1 » message on the display indicating that the Firmware is saving the network credentials to Pico's flash memory.

Reminders

The « Reminders » can be seen as a more flexible types of « Calendar events ». ***Their implementation is not completed yet in Firmware Version 2.00,*** but nonetheless I included Appendix I that gives a good idea of what it is.

Service light

When you are in a dark environment and some more light is required, the « Service light » feature of the RGB Matrix may be used.

If you press the « Up Long-press » button on the remote control (while the RGB Matrix is the usual « Date and Time » display mode), the RGB Matrix brightness will be increased to its maximum for a period of one minute.

You can press more than once to increase the brightness period by one minute each time, up to a maximum of 5 minutes.

The LED matrix brightness will automatically return to its original brightness setting (either automatic or steady) when the period is over.

Temperature display

The Firmware can read the temperature of the Raspberry Pi Pico. There are options to display this temperature from a terminal menu and / or to scroll it in an Auto-scroll.

The same applies with the ambient temperature that is read from the DS3231 real-time clock. To give a good time precision and prevent time drift, the DS3231 integrates electronic circuitry to compensate for the impact of ambient temperature in its time keeping. For this reason, it is possible to extract the ambient temperature from the IC. It gives a good idea of the ambient temperature.

It is planned that the temperature can be displayed either in Fahrenheit or in Celsius. However, as of Firmware Version 2.00, Fahrenheit conversion is not implemented.

Future firmware versions may also provide « How-to » options on how to install temperature sensors like DHT-22 and / or BME-280. For those interested in adding such a device, you may want to take a look at the « Pico-Green-Clock » User Guide and Firmware in my repository since those two sensors have been implemented as options in this device.

It is also possible to build an ecosystem with Pico microcontrollers so that one device equipped with a BME280 sensor shares the information through an MQTT broker. I would like to eventually develop such an ecosystem in the future…

Time setting

PicoW

If you use a PicoW and you do have access to a Wi-Fi network you should never have to set the time. Once you properly configured the network name (« SSID ») and password, the RGB Matrix will periodically synchronize itself with a time reference server over the Internet.

However, you must setup the Daylight Saving Time country code and Time zone. The Time zone is used by the RGB Matrix to properly convert the UCT (« Universal Coordinated Time ») received from NTP to your local time. If you do not setup the Time zone, the device will continually update the time with UCT time, which may not be what you want.

As of Firmware Version 2.00, Daylight Saving Time support has not been completely implemented. That means, for now, you will need to adjust the Time zone twice a year (« Summer time » and « Winter time ») depending when you change from one to the other in your country. As I did in the Pico-Green-Clock (in another of my repositories), I plan to implement an automatic Time zone handling in a future release. If you properly identify and set your country code (see Appendix C), the change will occur automatically when the Firmware has been updated to handle it.

Pico

If you use a Pico, I strongly suggest that you update to a PicoW if you have access to a Wi-Fi network. As mentioned above, PicoW would allow you to synchronize the RGB Matrix to a time server over the internet. It will also provide access to eventual new options in the future (like MQTT and / or others).

In the meantime, if you continue use the Pico, it is possible to setup the time with the terminal menus and also with the remote control. (Date setting must be done with the terminal menus as of Firmware Version 2.00, tough).

Even if the power is turned Off for a while, the battery attached to the control board should keep the correct time for a relatively long period of time.

Time format

There is an option planned in the configuration to toggle the time format between 12-hours format and 24-hour format. As of Firmware Version 2.00, however, this option has not been completely implemented and the time format is 24-hours.

Timezone

The Time zone mus be properly setup by the user to handle the Network Time Protocol option. Please refer to the section « Daylight Saving Time » above for more details.

When Daylight Saving Time will be fully implemented, the Time zone change (between simmer time and winter time) will be handled automatically by Firmware).

Watchdog

If you read the section about « Firmware endless loop » above, you've seen that the main system thread gave sometimes problems during code development. Another feature that has been added to the code is a software watchdog making sure that the main endless loop is active.

If the watchdog detects that the main system loop is stopped, it will cumulate the time and if it passes more than five (5) minutes, it will automatically reset (restart) the Firmware. While the watchdog cumulates the time the main thread is stopped, it will give a visual indication on the LED matrix. It will gradually change the red top border of the Time window to yellow. So, you will gradually see the top box border changing from red to yellow on the left and on the right. If / when the yellow dots meet together at the middle of the display, the device will restart.

This also means that if you navigate through the terminal menus for more than five minutes, the Firmware will restart. Five minutes is usually long enough to perform what we need to do. Also, the user simply needs to return from the terminal menus for the watchdog to reset and restart a new 5-minutes count.

In summary, when you use the terminal menus, take a look at the time box border. When you see the yellow dots approaching the middle of the matrix, get out of the menus (press « ESC », then « Enter » as many times as required to get out of all menus).

Appendix A – Configuring an external monitor

Introduction

If you want to get the most out of your Pico-RGB-Matrix, using a PC running a terminal emulator program is the way to go. It will allow you to quickly display a lot of information available on the RGB Matrix and / or to go through a lot of configurations settings. A PC keyboard and a PC monitor will be much faster than the RGB Matrix LED display and remote control.

This Appendix will show you how to make a connection between the RGB Matrix and an external terminal emulator to display device information, perform setup operations and even to help you if you decide to work on the source code to add new features to your device. Not only such an external terminal emulator is much easier to work with, but it is also much faster to read information and / or for debugging purposes than looking at a small LED display scrolling a few characters at a time.

Even if we focus on the RGB Matrix, remember that this type of connection is possible for all your projects using a Raspberry Pi Pico.

We will begin with the easier and potentially faster way to establish a connection. Then, we will see other techniques that could be a little more complicated and / or slower, but they may be useful in some specific situations.

WARNING: As a general comment, you must understand that sending data through a serial port (even if it is a USB port configured at 921,600 bauds) is relatively slow for a microcontroller by today's standards. You will quickly have timing problems and / or undesirable side effects if you add too many « display » instructions, moreover if you do it inside a callback with a quick repeat period or inside an interrupt service routine.

Pico to computer, USB-to-USB

1) The easiest way to communicate between the Pico and your computer is definitely with a USB-to-USB connection. Make sure the following lines appear in your Pico RGB-Matrix « make » file (CMakeLists.txt):

   pico_enable_stdio_uart(Pico-RGB-Matrix 0)
   pico_enable_stdio_usb(Pico-RGB-Matrix 1)

   If not, add them. The first line will stop sending the debug information to the Pico's integrated UART and the second line will make it sent through the Pico's USB port (you can also leave both equal to « 1 » if you want).

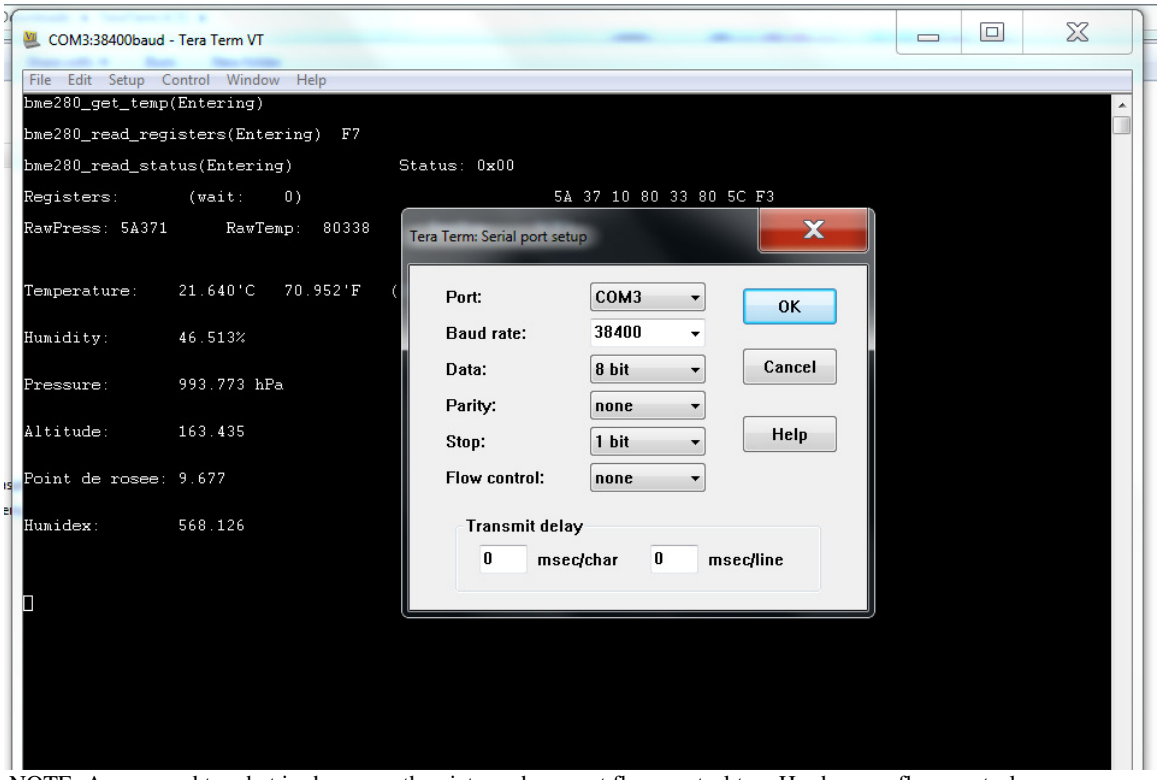
2) Now, you need to find a terminal emulator program that is compatible with your computer. Personally, I use the popular TeraTerm, Freeware and Open source. As its author wrote, « it is not a full-fledge terminal emulator », but I found it to be the perfect tool to display debug information sent by the Raspberry Pi Pico. You may consider making a donation to the author if you use it and find it useful.

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

3) 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 USB connection (if it has not already been assigned, it will when you start the Pico-RGB-Matrix).

   Note: on some older Windows versions, you may need to install a special USB driver so that it is recognized as a COMx (serial) port. This is outside the scope of this User Guide, but you may search the Internet with the keyword « Zadig ».

4) 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-RGB-Matrix Firmware), but as long as you set the Raspberry Pi with the same protocol values, it should be OK). Set « Flow control » to « Hardware ».



NOTE: As opposed to what is shown on the picture above, set flow control to « Hardware » flow control

5) TeraTerm will give an error if you try to connect to the USB port when the RGB Matrix is not connected (because it does not « see » a valid working serial port on a USB port). And when the

RGB Matrix is up and running, it may be already too late to establish because the RGB Matrix did not recognized any device connected to its USB port. (The most recent PCs have a behavior that makes them easier to work with that older PCs).

To prevent this « chicken-and-egg » type of situation, I added a loop at the beginning of the source code that should give you enough time to start the terminal emulator program. So, as soon as you see « Start emulator program » on the LED matrix, start TeraTerm.

Read the section « Power up sequence » to understand how long is the time window you have to establish the connection (you have more or less 15 seconds, so you may need to restart the RGB Matrix a few times on your first tries, until you configure – and save the right configuration – your terminal emulator program).

6) As soon as the USB CDC communication has been established, the RGB Matrix will detect it and automatically jumps to a terminal menu from which you may navigate through many other submenus.

7) You will want to turn On the « logging » option on your terminal emulator program so that all information displayed on the terminal is simultaneously logged to a file. It is often interesting to refer to this log file later. If not required, it is easy to simply erase the file when it is no more required.

8) WARNING: You must be aware of this: When you are navigating through the different terminal menus, the main system « endless loop » will be « on hold », since this is that thread who jumps through the terminal menus. If you stay too long on those menus, the watchdog will do what it was programmed to do and it will restart the Firmware. As of version 2.00, the watchdog will bite after 5 minutes which should give you more than enough time to perform a lot of actions in the menus. Watch the yellow LEDs on the display and read the section about the watchdog timer to better understand what is going on behind the scene. Returning from terminal menus (« ESC », followed by « Enter » until there is no more menu displayed) will reset the watchdog (all yellow LEDs on the display will return to red).

Pico's UART to PC's USB

A USB-to-USB connection as described in the previous section above is certainly the easiest way to exchange data between a computer and the Raspberry Pi Pico. If for some reason, you can't use the Pico's USB port to do so, here is another way to proceed to establish a connection between a computer and a Pico. We still use a PC with a USB port, but this time, we use one of the Pico's internal UART instead of its USB port.

1) The Raspberry Pi Pico has two internal UART (« Universal Asynchronous Receiver Transmitter »): the integrated circuit used to implement the RS232 serial protocol. The details about serial protocol are beyond the scope of this User Guide.

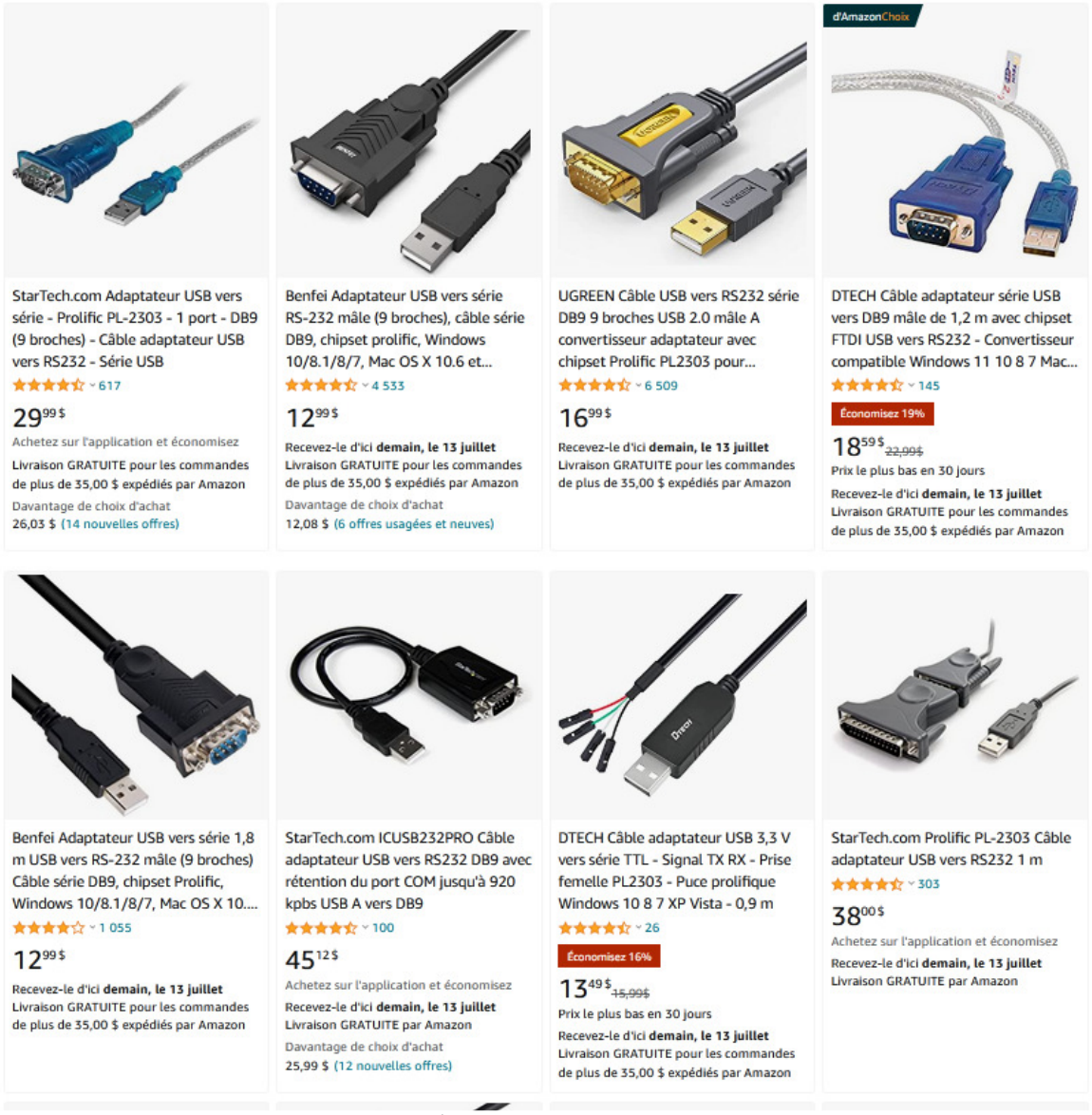
2) However, even if the Pico is able to receive and / or transmit data from / to a serial line, we still have to adjust the voltage levels. The Pico works with 3.3 volts, whereas RS232 works with higher voltage (there has been many protocol revisions over the years, but the signal is usually swapping from -5 volts to + 5 volts. So, we need an RS232 adaptor to convert RS232 levels from: between 0 volt to 3.3 volts on the Pico side, to: between -5 volts to +5 volts on the PC side. Some time ago, I bought the following adaptor from Waveshare:

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

3) It is easy to make the connection between the Pico and the adaptor using the following signals:
   - Vcc (3.3 volts)
   - Ground
   - TxIn connected to Pico's UART Out (TxIn transmits TO the PC)
   - RxOut connected to Pico's UART In (RxOut receives FROM the PC)

   TxIn and RxOut seems reversed, but this is the way it worked on the adaptor that I received.

4) So far, we converted the Pico's voltage levels, but we still have an RS232 communication line, whereas we want to receive data through the PC's USB port. We need a RS232-to-USB adaptor to convert the RS232 data coming out from the adaptor to CDC USB serial on the PC side. Below are some examples of such an adaptor taken from Amazon web pages, as of mid 2022. Make sure the adaptor you buy provides the device driver (software) required by your OS (correct version of Mac, Windows, Linux, or other).



(Note: prices shown above are in Canadian $, as of mid-2022)

5) 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).

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

NOTE: Based on my own experience, once in a while, it may happen that the terminal emulator stops displaying 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's UART to PC's serial port

If you don't have a computer on which you can run TeraTerm but you do have an old system with too much dust in your basement, you can use this old PC to make your connection with the Pico.

This old PC may not have a USB port (if it is very old), so we will use the PC's RS232 serial port to establish the communication.

1) Follow the instructions in the previous section on how to connect an RS232 adaptor on the Pico's side.

2) Once this is done, all what remains to be done is to connect the PC's serial port to the RS232 Pico's adaptor. (No need for a USB adaptor since we will connect to the PC's serial port).

3) 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 the first versions of Windows. HyperTerminal was another terminal emulator program included in the early Windows versions. TeraTerm is a new generation terminal emulation software who works fine, but it may not work on your old machine.

   Don’t forget that you still have to configure the serial protocol on the communication program. (Pico-RGB-Matrix’s Firmware is set by default to 921,600 – N – 8 – 1 since it is configured for a CDC USB connection, but if you use an old PC, you will 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 parameters.

4) Even the old terminal emulator programs had an option to save information to a log file. You will want to turn logging On to be able to refer to the logged information later if required.

Pico's UART to a VT-101 terminal

So far, we used a terminal emulator program running on a PC to receive the data from the Pico. This terminal emulator program may be replaced, of course, by a « real » terminal.

In fact, many different terminal brands and models existed, but the « VT-101 » originated from Digital Equipment Corporation (« DEC ») and was very popular. If you have such a monitor, you can use it to display RGB Matrix information on its screen.

Follow the instructions from the previous section « Pico's UART to PC's serial port » and simply connect the cable to your terminal instead of the PC's serial port. Again, you must make sure that the terminal serial protocol matches the one on the Pico's side.

Appendix B - RGB Matrix power requirement

IMPORTANT

You must be aware that the Pico-RGB-Matrix requires a relatively powerful current source to work properly. Moreover, the current required changes depending of the intensity of the LED matrix, the number of LEDs that are simultaneously turned On at any time, and the color of the LEDs that are turned On (the LEDs of different colors needs more or less current).

For example, if the LED matrix is fully illuminated (2048 LEDs) at full intensity, the current required will be similar to the following figures (for a « normal » brightness):

Color	Matrix status	Current
White (red + green + blue)	Fully illuminated	730 to 740 mAmp
Yellow (red + green)	Fully illuminated	680 to 710 mAmp
Magenta (red + blue)	Fully illuminated	630 to 680 mAmp
Cyan (green + blue)	Fully illuminated	480 to 490 mAmp
Red LEDs	Fully illuminated	550 to 560 mAmp
Green LEDs	Fully illuminated	315 to 330 mAmp
Blue LEDs	Fully illuminated	200 to 230 mAmp

Current required for a fully illuminated matrix (2048 LEDs turned On) at « normal » brightness.

When the RGB-Matrix is waiting to be uploaded (all LEDs turned Off), or when the program is running with the RGB electronics but without any LED turned on, the current is around 70 to 75 mA.

When the RGB matrix is running with the date and time displayed (time in green and date in blue), the current is around 100 to 120 mA.

Considering these figures, it is important to provide a proper source of current to the device to prevent blowing up your PC USB port. You must use a power supply able to provide at least 2 or 3 amps. It is time to go to the basement and remove the power supply of this old PC that you didn't use for the last five years. A typical PC power supply is able to provide 15 to 25 Amps !! Make sure you connect to +5 volts since many PC power supplies also provide -5, +12, and -12 volts.

Appendix C – Daylight Saving Time setting

To properly setup the « Daylight Saving Time » (« DST »), refer to the table below and select the Country Code (first column) corresponding to your country and / or the parameters for DST that are used in your country. The number in the first column is the one that you must entered when asked for the « Country code ».

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-RGB-Matrix properly supports Daylight Saving Time / Summer Time for their country when it is fully implemented / supported in the code.

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 device.	= = =	= = =
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 Danmark shavn 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	Moldova	Last Sunday March, 2h00	Last Sunday October, 3h00
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 D - RGB Matrix remote control timing

The RGB Matrix remote control unit protocol is similar to the NEC remote control protocol. However, the timings are slightly different.

The « Start signal » pulse distance is around 13650 usec, whereas the bit 0 pulse distance is around 1140 usec and the bit 1 pulse distance is around 2280 usec (see table below).

Using the Pico-Remote-Analyzer Firmware Version 2.00, you can see on the next page an example of the timing obtained when pressing the « Channel- » button on the remote control.

Still using the Pico-Remote-Analyzer Firmware Version 2.00, the decoding function for the NEC protocol has been adjusted to match the timings of the RGB Matrix remote control and the resulting command codes are shown in the page following the timing page.

= =

Pico-Remote-Analyzer - Firmware version 2.00
Microcontroller is a PicoW
Pico's Unique ID: E661-4103-E79D-5723
Brand under analysis: Car MP3
Remote control model number: RGB Matrix remote control
Step count: 71

= =

Raw timing display
Button: <Channel->

Step number	Logic level	Duration (in usec)	Pulse Distance	Step number	Logic level	Duration (in usec)	Pulse Distance
1	low	9207		51	low	629	
2	high	4434	13641	52	high	514	1143
3	low	622		53	low	601	
4	high	486	1108	54	high	1648	2249
5	low	656		55	low	628	
6	high	486	1142	56	high	514	1142
7	low	658		57	low	603	
8	high	484	1142	58	high	1648	2251
9	low	632		59	low	655	
10	high	484	1116	60	high	1594	2249
11	low	657		61	low	657	
12	high	486	1143	62	high	1619	2276
13	low	657		63	low	630	
14	high	484	1141	64	high	485	1115
15	low	605		65	low	686	
16	high	511	1116	66	high	1591	2277
17	low	657		67	low	658	
18	high	487	1144	68	high	39883	40541
19	low	631		69	low	9210	
20	high	1619	2250	70	high	2176	11386
21	low	655		71	low	631	
22	high	1622	2277				
23	low	658					
24	high	1592	2250				
25	low	629					
26	high	1621	2250				
27	low	656					
28	high	1621	2277				
29	low	604					
30	high	1646	2250				
31	low	631					
32	high	1621	2252				
33	low	632					
34	high	1645	2277				
35	low	631					
36	high	1620	2251				
37	low	631					
38	high	511	1142				
39	low	630					
40	high	1621	2251				
41	low	630					
42	high	512	1142				
43	low	603					
44	high	513	1116				
45	low	657					
46	high	487	1144				
47	low	660					
48	high	1592	2252				
49	low	656					
50	high	486	1142				

Timings of the RGB Matrix remote control unit

```
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
                    Pico-Remote-Analyzer - Firmware version 2.00
                         Microcontroller is a PicoW
                     Pico's Unique ID: E661-4103-E79D-5723
              Brand under analysis: Car MP3 - RGB Matrix remote control
            Remote control model number: Remote control model not identified
                                 Step count: 71
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

                  Number of remote control buttons decoded so far: 21

                 Remote control                         Infrared command
                   button name                               decoded

                 [  1] Channel-                              0x00FFA25D
                 [  2] Channel                               0x00FF629D
                 [  3] Channel+                              0x00FFE21D
                 [  4] Previous                              0x00FF22DD
                 [  5] Next                                  0x00FF02FD
                 [  6] Play/Pause                            0x00FFC23D
                 [  7] Volume-                               0x00FFE01F
                 [  8] Volume+                               0x00FFA857
                 [  9] EQ                                    0x00FF906F
                 [ 10] 0                                     0x00FF6897
                 [ 11] 100+                                  0x00FF9867
                 [ 12] 200+                                  0x00FFB04F
                 [ 13] 1                                     0x00FF30CF
                 [ 14] 2                                     0x00FF18E7
                 [ 15] 3                                     0x00FF7A85
                 [ 16] 4                                     0x00FF10EF
                 [ 17] 5                                     0x00FF38C7
                 [ 18] 6                                     0x00FF5AA5
                 [ 19] 7                                     0x00FF42BD
                 [ 20] 8                                     0x00FF4AB5
                 [ 21] 9                                     0x00FF52AD
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

Decoded commands of the RGB Matrix remote control unit

Appendix E – For developers

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-RGB-Matrix (email address is given after Index at the beginning of this document).

Alarm numbers

There are nine (9) independent alarms available in the RGB Matrix. They are numbered from 1 to 9 to be more « human-like » (and to keep the alarm designation only one-digit wide). However, in the code, they are numbered from 0 to 8.

Alarm configuration is saved to Pico's flash memory so that in case of a power failure, all alarms parameters will be restored to the active device configuration on power-up.

Coding standard

Many function names (and also variable names) have been changed from the original Waveshare's Version 1.00 to better represent what they are / what they do. Also, even if most modern integrated development environment propose tools to easily navigate through the code, functions have been sorted in alphabetical order (except « main », which comes first).

Debug chunks of code

Since the Pico-RGB-Matrix 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 a significant impact on time critical sections of the code and / or on specific timings (read « the Firmware may crash »). It is assumed that if you use some of those « debug » options, you know what you do.

Debug flags may also be toggled On or Off while the Firmware is running, from the main terminal menu.

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 and / or optimization or « code that must be reviewed later » or « code that can be uncommented for debugging purposes ».

Pico-RGB-Matrix based ecosystem

If should be relatively easy to add temperature sensors to the RGB Matrix. For those interested to do so, you may take a look at the source code and / or User Guide of the Pico-Green-Clock in one of my repositories. However, I thought it could be a good idea to develop an ecosystem based on the Pico-RGB-Matrix. A centralized MQTT broker could be easily implemented using a Raspberry Pi (why not using the new Raspberry Pi 5 to do so?!)

Then, it should be easy to add MQTT publishers for outside temperature, telephone caller ID, and even other Pico-based units to control Zigbee lighting devices, Zigbee blinds, heater controls, etc…

RGB Matrix « Option section »

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

Test section

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

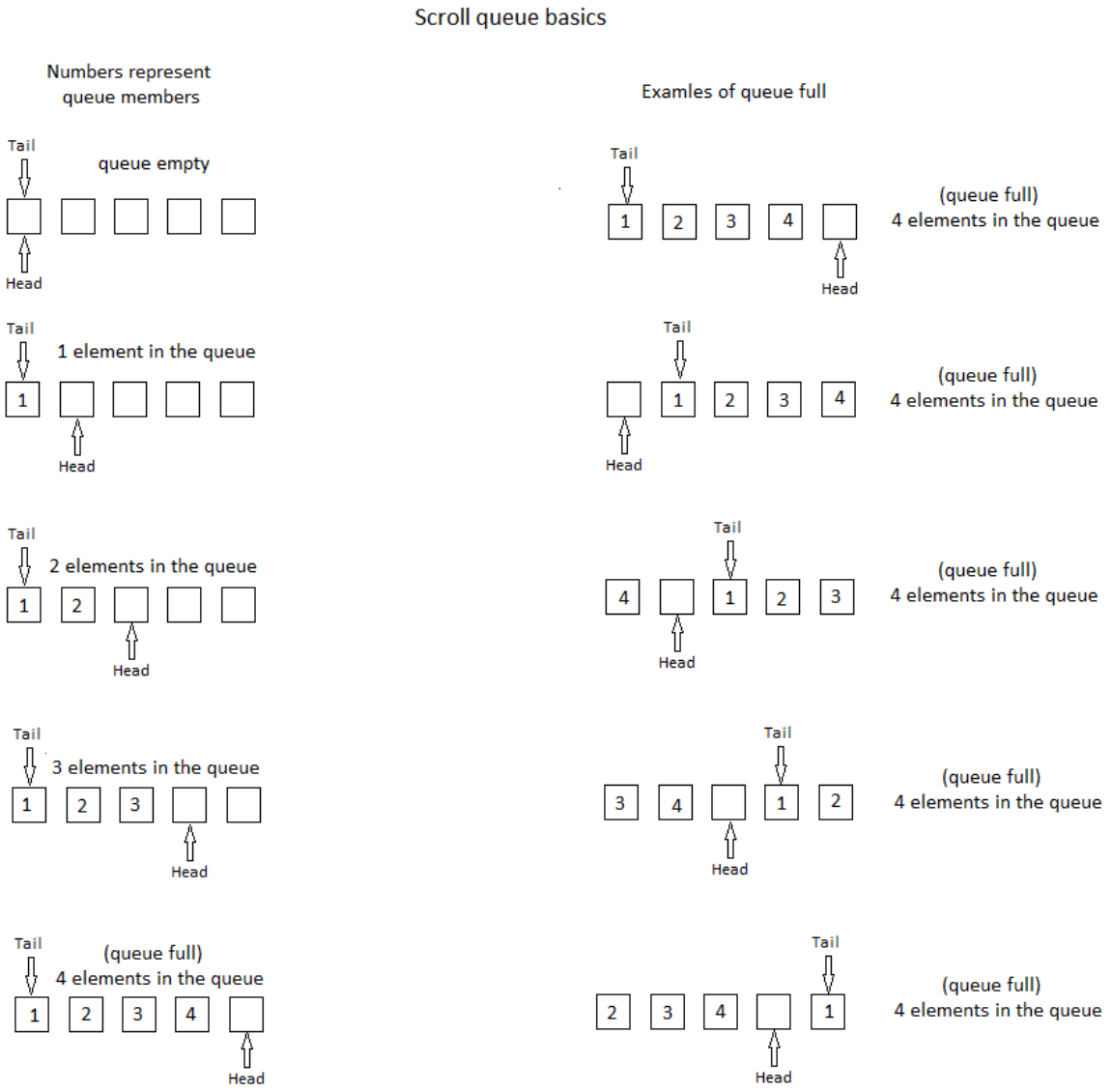
Those chunks of test code must be considered as such: test code! It may help you with the implementation of new functions / features, but it must NOT be considered as « debugged » and / or « fool-proof » code! Use it at your own risks and efforts! The main tested sections of code will often be modified / 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. Keep that in mind, even when you ask to execute a specific test number from within the terminal menu.

Appendix F – GPIO used in Waveshare's Pico-RGB-Matrix

List of GPIOs used in the Waveshare's Pico-RGB-Matrix:

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	(Out)	Red top matrix half data byte.
GPIO 3	(Out)	Green top matrix half data byte.
GPIO 4	(Out)	Blue top matrix half data byte.
GPIO 5	(Out)	Red bottom matrix half data byte.
GPIO 6	(I2C) SDA	I2C Data line to read DS3231 real-time IC.
GPIO 7	(I2C) SCL	I2C Clock line for DS3231 real-time IC.
GPIO 8	(Out)	Green bottom matrix half data byte.
GPIO 9	(Out)	Blue bottom matrix half data byte.
GPIO 10	(Out)	« A » scan bit line select.
GPIO 11	(Out)	Clock.
GPIO 12	(Out)	« Latch » (also called « Strobe »).
GPIO 13	(Out)	« Output Enable » (active low).
GPIO 14	- - -	Not used.
GPIO 15	(In)	Local button « Up » GPIO.
GPIO 16	(Out)	« B » scan bit line select.
GPIO 17	- - -	Not used.
GPIO 18	(Out)	« C » scan bit line select.
GPIO 19	(In)	Local button « Down » GPIO.
GPIO 20	(Out)	« D » scan bit line select.
GPIO 21	(In)	Local button « Set » GPIO.
GPIO 22	(Out)	« E » scan bit line select (Hub-75 expansion).
GPIO 23	- - -	Not used.
GPIO 24	- - -	Not used.
GPIO 25	(Out)	On-board Pico`s LED (different on PicoW).
GPIO 26	(In)	ADC0 (Ambient light reading with photoresistor).
GPIO 27	(Out)	Active buzzer GPIO
GPIO 28	(In)	Infrared receive sensor.
GPIO 29	(In)	ADC-Vref (Power supply voltage reading).
GPIO 30	- - -	Not used

Appendix G – Sound queue basics



Note: Also applies similarly to other circular buffers (“queues”) implemented in the code.

Appendix H – Network Time Protocol white paper

Introduction

Explaining in details what is the: « 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 RGB Matrix 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 time zone you are), the parameters « Daylight Saving Time » (or « DST Country ») and « Time zone » must be properly setup in the RGB Matrix configuration for NTP to work as expected.

Once you entered the network name and password, your setup will be saved to PicoW's non-volatile memory, so you shouldn't have to repeat this setup except if you change your Wi-Fi access credentials.

A PicoW is required if you want to use NTP. The « plain » Pico does not contain the Wi-Fi interface circuitry as does the PicoW, which is required for accessing the Internet. However, you can use the RGB Matrix with a Pico without problem with the correct Firmware versions. The real-time clock IC in the RGB Matrix has shown that it is able to keep a very good time precision over long period of time.

IMPORTANT:

Keep in mind that your Wi-Fi credentials (network name and password) will be saved in PicoW`s non-volatile memory, in a location that is unlikely to be overwritten even if you upload another program / firmware to your PicoW. If ever you give away your PicoW to someone else, this person could easily retrieve your credentials from your PicoW. For this reason, if you are to dispose of your PicoW, you may want to use the Pico-Flash-Utility in one of my repositories to wipe (erase) the whole PicoW'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 RGB Matrix 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 PicoW for NTP support. All you have to do is comment (or uncomment) the #define NTP_SUPPORT in the source code to enable PicoW (and NTP) support.

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 PicoW. The make file must determine the need to include the libraries for the PicoW (Wi-Fi support). Take note that the default "CMakeLists.txt" file in the Pico-RGB-Matrix repository is for the PicoW. The same PicoW

« CMakeLists.txt » file is also backed-up as « CMakeLists.txt.PicoW ». Unsurprisingly, the « CMakeLists.txt » file for the Pico has been named « CMakeLists.txt.PicoW ». You may copy either of those versions (for Pico or PicoW) over the « CMakeLists.txt » as you see fit for your needs.

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

Take note that the « plain » Pico Firmware version will work without problem on a PicoW. If ever you play with the code and you want to use the on-board Pico LED, however, you'll have problems since the PicoW 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 PicoW. (However, see section about « How to upload a new Firmware Version » about physical differences between both devices.

Following what is said above, take note, however that the PicoW 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 the PicoW Firmware 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 PicoW 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 versions to make it for the extra space required for Wi-Fi library support.

Appendix I – Reminders (NOT IMPLEMENTED YET)

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 RGB Matrix 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.

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 the LED matrix 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 event occurrence. 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 and / or hours and / or days and / or weeks. Actions based on months, seasons, or years 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 planned for the RGB Matrix. It would be the same thing for an action that will occur « every 3rd Wednesday » of each month. Those examples will eventually be covered with Reminders of type 2 if / when implemented in the code. NOTE: For a repetitive action that occurs each year, the « Calendar events » is exactly what you need!!

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 and / or 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.

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 Wednesday, September 4th, 2024 and will be out of the country for about 8 weeks. He'll be back on Wednesday, October 30st, 2024.

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.

Up to 40 Reminders may be configured. Here is an explanation of what represents each parameter.

1) The first parameter is the start of the global period. In our case, the period starts when the neighbor leaves the country, that is, September 4th, 2024, say at 00h00m00s (the very beginning of the day).

2) The second parameter corresponds to the end of the global period. Since the neighbor comes back on October 31st, 2024, this is what we will enter as the end of the period, say at 23h59m59s (the very end of the day)..

3) The third parameter is the date and time of the first Reminder alarm (or « ring »). In paragraphs 1 and 2 above, we defined the global period of time during which the system will consider our Reminder to be active (outside of this global period, the system will simply ignore – skip – this Reminder). Now, we must tell the system at which time we want the device to ring a « beep » and scroll a reminder message on the LED 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 2024 is Thursday 5th 2024. Since we preferably don't want the reminder message to scroll at the same time as another Auto-scroll, we should select a time that will not interfere with a defined Auto-scroll. Let's say a good time for the third parameter could be: 13h01m36s.

4) Fourth parameter contains the interval for each ring. When the device reaches the target time (defined in step 3 above), it will make a « few beeps ». If you look at the display 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 device at this exact moment, you may miss these few beeps and this message. So let's add an interval of time after which the beeps and the message will repeat again. Let`s define this interval to be 15 minutes, to make sure we don't miss it. As mentioned before, we work mostly with seconds in the Reminder algorithm, so you

must convert the 15 minutes in seconds using a calculator (15 minutes X 60 seconds = 900 seconds).

5) Next parameter is the Reminder duration. In our case, let's the device reminds 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 device to remind us after we came back at 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. Since the duration is from 13h00 to 18h00, we will enter 18000 (5 hours X 60 minutes X 60 seconds). The Reminder will ring every 15 minutes, from 13h00 up to 18h00, or until user shuts it Off, whichever happens first.

6) Next parameter 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 5th, 2024 at 13h01h36 and Thursday Sep 12th, 2024 at 13h01h36. You may take a calculator to find it: (7 days X 24 hours X 60 minutes X 60 seconds = 604800).

7) Finally, you enter the message to scroll on the LED 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 display. The string may be up to 50 characters long. For example, you 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 to be « active ».
- We then decide the moment (the first time) at which the Reminder will be triggered (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 (15) minutes (900 seconds).
- 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 (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 the other parameters it must be specified as a number of seconds.

NOTE: Keep in mind the right terminology: an ***active*** Reminder is a Reminder for which the current moment is inside the global 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.