

Historia czatu

[ChatGPT](#)

[ChatGPT](#)



[Python](#)



[Pi Code Craft](#)

Jeszcze 1

[Odkryj modele GPT](#)

Projekty

Nowy projekt

Dzisiaj

1.

[Biblioteki do LoRa Python](#)

2.

[Konfiguracja modułu LoRa 868](#)

3.

[Ustawienie statycznego IP RPi](#)

Wczoraj

1.

[Moduły szyfrowania w Pythonie](#)

2.

[Raspberry Pi Matrix Podłączenie](#)

3.

[Problemy z klonowaniem Git](#)

4.

[Podłączenie matrixa RGB RPi](#)

5.

[Budowa płytki PCB](#)

6.

[Charakterystyka zasilacza laboratoryjnego](#)

7.

[Błąd kompilacji FrameCanvas](#)

8.

[Markdown linki](#)

Poprzednie 7 dni

1.

[Poprawa zapisu JSON](#)

2.

[Wczytywanie danych JSON](#)

3.

[Ustawienie linii ciągłej](#)

4.

[Poprawki dla SavePopup](#)

5.

[Kivy Builder Tutorial](#)

6.

[Pomoc w debugowaniu kodu](#)

7.

[Obsługa błędów pliku](#)

8.

[Aktualizacja metryk ze Spinnerem](#)

9.

[Szyfrowanie i operacje XOR](#)

10.

[Biblioteki funkcji skrótu](#)

11.

[Tworzenie venv na RPi](#)

12.

[Używanie plików .kv](#)

13.

[Wprowadzenie do AutoLISP](#)

Poprzednie 30 dni

1.

[Rozdziały z zakładkami PDF](#)

2.

[Błąd vla-object w AutoLISP](#)

3.

[Błąd typu stringp](#)

4.

[Szyfrowanie XMPP wiadomości](#)

Wyświetl planyNieograniczony dostęp, funkcje dla zespołów i wiele więcej

Pi Code Craft

Udostępnij



Powiedziałeś(-aś):

podaj mi jak podłączyć raspberry pi do matrix 64x32 bez bonetu podaj mi piny



To connect a Raspberry Pi to a 64x32 RGB LED matrix without using a HAT or bonnet, you can directly wire it to the GPIO pins of the Raspberry Pi. Here's a list of the pin connections:

Matrix Pins and GPIO Mapping:

1. R1 (Red 1): GPIO 17 (Pin 11)
2. G1 (Green 1): GPIO 18 (Pin 12)

3. B1 (Blue 1): GPIO 22 (Pin 15)
4. R2 (Red 2): GPIO 23 (Pin 16)
5. G2 (Green 2): GPIO 24 (Pin 18)
6. B2 (Blue 2): GPIO 25 (Pin 22)
7. A (Address A): GPIO 5 (Pin 29)
8. B (Address B): GPIO 6 (Pin 31)
9. C (Address C): GPIO 13 (Pin 33)
10. D (Address D): GPIO 19 (Pin 35)

Control Signals:

11. CLK (Clock): GPIO 4 (Pin 7)
12. OE (Output Enable): GPIO 12 (Pin 32)
13. LAT (Latch): GPIO 16 (Pin 36)

Power:

- Connect GND from the matrix to any Raspberry Pi ground pin (e.g., Pin 6).
- Connect the power supply for the LED matrix (5V) directly to the matrix power input. Ensure the power supply can handle the matrix's current requirements, as the Pi itself cannot supply enough current.

Additional Notes:

- Make sure to use a level shifter if the LED matrix requires 5V logic, as the Raspberry Pi GPIO pins operate at 3.3V logic levels.
- Use a library like [rpi-rgb-led-matrix](#) to control the matrix.
- Ensure proper cooling for the Raspberry Pi if running graphics-intensive animations or high refresh rates.

Would you like a detailed Python script or setup guide using the rpi-rgb-led-matrix library?
Let me know!

Powiedział(-aś):

mam kolorowy kabel proszę podaj mi tabelkę

ChatGPT powiedział:



Oto tabela połączeń między Raspberry Pi i 64x32 RGB LED Matrix dla kolorowego kabla:

Kolor kabla	Opis na macierzy LED	Pin GPIO Raspberry Pi	Numer pinu na Pi
Czerwony	R1 (Red 1)	GPIO 17	Pin 11
Zielony	G1 (Green 1)	GPIO 18	Pin 12
Niebieski	B1 (Blue 1)	GPIO 22	Pin 15
Pomarańczowy	R2 (Red 2)	GPIO 23	Pin 16
Żółty	G2 (Green 2)	GPIO 24	Pin 18
Fioletowy	B2 (Blue 2)	GPIO 25	Pin 22
Biały	A (Adres A)	GPIO 5	Pin 29
Szary	B (Adres B)	GPIO 6	Pin 31
Brązowy	C (Adres C)	GPIO 13	Pin 33
Czarny	D (Adres D)	GPIO 19	Pin 35
Różowy	CLK (Zegar)	GPIO 4	Pin 7
Jasnoniebieski	OE (Włączanie Wyjścia)	GPIO 12	Pin 32
Złoty	LAT (Zatrask)	GPIO 16	Pin 36
Czarny (GND)	Masa	GND (Ground)	Pin 6

Uwagi:

1. Zasilanie:

- Podłącz zewnętrzne 5V zasilanie do macierzy LED (zwykle czerwony i czarny kabel do złącza zasilania macierzy).
- Nie podłączaj bezpośrednio zasilania macierzy LED do Raspberry Pi, bo może spalić płytkę!

2. Przetączniki poziomów logicznych:

- Jeśli macierz wymaga sygnałów 5V, użyj konwertera poziomów logicznych między Raspberry Pi (3.3V) a macierzą.

3. Biblioteka:

- Użyj [rpi-rgb-led-matrix](#) do sterowania macierzą.

Jeśli potrzebujesz dodatkowych wyjaśnień, kodu, lub rysunku schematu, daj znać! 😊

Powiedziałeś(-aś):

wcześniej podałeś mi tablkę z rzędami kolorów

ChatGPT powiedział:



Rozumiem! Oto bardziej uproszczona tabela, przypisująca kolory do funkcji dla 64x32 RGB LED Matrix przy użyciu Raspberry Pi, bazująca na kolorowych kablach:

Kolor kabla	Opis na macierzy LED	Pin GPIO Raspberry Pi	Numer pinu na Raspberry Pi
Czerwony	R1 (Red 1)	GPIO 17	Pin 11
Zielony	G1 (Green 1)	GPIO 18	Pin 12
Niebieski	B1 (Blue 1)	GPIO 22	Pin 15
Pomarańczowy	R2 (Red 2)	GPIO 23	Pin 16
Żółty	G2 (Green 2)	GPIO 24	Pin 18
Fioletowy	B2 (Blue 2)	GPIO 25	Pin 22
Biały	A (Adres A)	GPIO 5	Pin 29
Szary	B (Adres B)	GPIO 6	Pin 31
Brązowy	C (Adres C)	GPIO 13	Pin 33
Czarny	D (Adres D)	GPIO 19	Pin 35
Różowy	CLK (Zegar)	GPIO 4	Pin 7
Jasnoniebieski	OE (Włączanie Wyjścia)	GPIO 12	Pin 32
Złoty	LAT (Zatrask)	GPIO 16	Pin 36
Czarny (GND)	Masa	GND (Ground)	Pin 6

Uwagi

1. Jeśli masz problem z dopasowaniem kolorów, warto sprawdzić dokumentację lub oznaczenia kabli w zestawie.
2. Powinieneś podłączyć zasilanie macierzy (5V) do dedykowanego zasilacza, nie przez Raspberry Pi.
3. Dopilnuj, aby GPIO było ustawione w trybie OUTPUT w oprogramowaniu sterującym.

Mam nadzieję, że to rozwiązuje wątpliwości! Jeśli chcesz kod Python do sterowania macierzą, mogę go przygotować. 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/examples-api-use $ cd ..
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master $ cd bindings/
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings $ cd python/
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ python --
version Python 3.11.2 kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-
master/bindings/python $ cd samples/ kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-
master/bindings/python/samples $ python r rotating-block-generator.py runtext.py
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/samples $
python r rotating-block-generator.py runtext.py kopis@raspberrypi:~/Pobrane/rpi-rgb-led-
matrix-master/bindings/python/samples $ python runtext.py
Traceback (most recent call
last): File "/home/kopis/Pobrane/rpi-rgb-led-matrix-
master/bindings/python/samples/runtext.py", line 3, in <module> from samplebase import
SampleBase
File "/home/kopis/Pobrane/rpi-rgb-led-matrix-
master/bindings/python/samples/samplebase.py", line 7, in <module> from rgbmatrix
import RGBMatrix, RGBMatrixOptions
File "/home/kopis/Pobrane/rpi-rgb-led-matrix-
master/bindings/python/rgbmatrix/__init__.py", line 7, in <module>
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/samples $
```

ChatGPT powiedział:



Wygląda na to, że napotkałeś problem przy uruchamianiu skryptu `runtext.py`. Błąd jest związany z biblioteką `rgbmatrix`, która prawdopodobnie nie jest prawidłowo zainstalowana. Postępuj zgodnie z poniższymi krokami, aby rozwiązać problem:

1. Zainstaluj wymagania

Przejdź do katalogu z kodem źródłowym biblioteki i zainstaluj wymagania:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python sudo pip install -r
requirements.txt
```

2. Zbuduj bibliotekę Python

Jeśli to nie zadziała, musisz zbudować i zainstalować moduł Python `rgbmatrix`:

bash

Skopiuj kod

```
sudo python3 setup.py build sudo python3 setup.py install
```

3. Sprawdź instalację

Po instalacji sprawdź, czy moduł `rgbmatrix` jest prawidłowo zainstalowany:

bash

Skopiuj kod

```
python3 -c "import rgbmatrix"
```

Jeśli nie wystąpi błąd, oznacza to, że biblioteka została prawidłowo zainstalowana.

4. Uruchom skrypt ponownie

Przejdź do katalogu samples i spróbuj ponownie uruchomić skrypt:

bash

Skopiuj kod

```
python3 runtext.py
```

5. Rozwiązywanie problemów

Jeśli nadal występują problemy:

- Upewnij się, że używasz kompatybilnej wersji Python (3.11 może być obsługiwany, ale starsze wersje, jak 3.9, są bardziej stabilne dla tej biblioteki).
- Upewnij się, że masz wszystkie zależności systemowe. Możesz je zainstalować za pomocą:

bash

Skopiuj kod

```
sudo apt-get update sudo apt-get install -y build-essential python3-dev python3-pillow
```

Daj znać, czy to rozwiązuje problem, czy potrzebujesz dodatkowej pomocy! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ ls Makefile  
README.md rgbmatrix samples setup.py  
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
```

ChatGPT powiedział:



Wygląda na to, że masz wszystkie potrzebne pliki w katalogu ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python. Aby poprawnie zainstalować bibliotekę Python, postępuj zgodnie z poniższymi krokami:

1. Zainstaluj bibliotekę Python

Upewnij się, że jesteś w katalogu ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python, a następnie uruchom poniższe polecenia, aby zbudować i zainstalować bibliotekę rgbmatrix:

bash

Skopiuj kod

sudo python3 setup.py build sudo python3 setup.py install

2. Sprawdź instalację

Po zakończeniu instalacji upewnij się, że biblioteka jest poprawnie zainstalowana, uruchamiając:

bash

Skopiuj kod

python3 -c "import rgbmatrix; print('rgbmatrix installed successfully')"

Jeśli widzisz komunikat **rgbmatrix installed successfully**, oznacza to, że instalacja się powiodła.

3. Uruchom przykładowy skrypt

Przejdź do katalogu samples:

bash

Skopiuj kod

cd samples

I uruchom skrypt runtext.py:

bash

Skopiuj kod

python3 runtext.py

4. Rozwiązywanie problemów

Jeśli nadal napotykasz błędy:

1. **Sprawdź zależności systemowe:** Upewnij się, że masz wszystkie wymagane pakiety systemowe:

bash

Skopiuj kod

sudo apt-get install -y build-essential python3-dev python3-pillow

2. Sprawdź logi błędów: Jeśli nadal występują problemy, podaj pełny komunikat błędu, aby można było dokładniej zdiagnozować problem.

Daj znać, jak poszło! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ sudo python3 setup.py build cc1plus: fatal error: rgbmatrix/core.cpp: Nie ma takiego pliku ani katalogu compilation terminated. error: command '/usr/bin/aarch64-linux-gnu-gcc' failed with exit code 1 kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
```

ChatGPT powiedział:



Błąd wskazuje, że plik `rgbmatrix/core.cpp` nie został znaleziony. Może to oznaczać, że źródła biblioteki nie zostały poprawnie pobrane lub coś jest nie tak z strukturą katalogów. Postępuj zgodnie z poniższymi krokami, aby rozwiązać problem:

1. Sprawdź kompletność źródeł

Przejdź do katalogu `rgbmatrix` i sprawdź jego zawartość:

```
bash
```

```
Skopiuj kod
```

```
ls rgbmatrix
```

W katalogu `rgbmatrix` powinny znajdować się pliki źródłowe, takie jak `core.cpp`. Jeśli ich brakuje, oznacza to, że biblioteka została niekompletnie pobrana.

2. Pobierz ponownie bibliotekę

Usuń obecny katalog i pobierz bibliotekę ponownie:

```
bash
```

```
Skopiuj kod
```

```
cd ~/Pobrane rm -rf rpi-rgb-led-matrix-master git clone https://github.com/hzeller/rpi-rgb-led-matrix.git rpi-rgb-led-matrix-master
```

3. Spróbuj zbudować bibliotekę ponownie

Po ponownym pobraniu przejdź do katalogu `bindings/python` i uruchom ponownie budowanie:

bash

Skopiuj kod

```
cd rpi-rgb-led-matrix-master/bindings/python sudo python3 setup.py build sudo python3 setup.py install
```

4. Zainstaluj zależności

Jeśli nadal występuje błąd, upewnij się, że masz wszystkie wymagane zależności do kompilacji:

bash

Skopiuj kod

```
sudo apt-get update sudo apt-get install -y build-essential python3-dev python3-pillow python3-setuptools git
```

5. Rozwiązywanie problemów

Jeżeli problem nadal występuje:

- Upewnij się, że używasz kompatybilnej wersji systemu operacyjnego Raspberry Pi OS (np. Raspberry Pi OS Lite lub Desktop w wersji 32- lub 64-bitowej).
 - Sprawdź dokładny komunikat błędu i upewnij się, że masz odpowiednie narzędzia kompilacyjne, takie jak gcc i g++.
-

Spróbuj ponownie pobrać bibliotekę i skompilować. Jeśli problem będzie się powtarzał, podaj wynik polecenia `ls rgbmatrix` oraz pełny komunikat błędu. 😊

Powiedziałeś(-aś):

a jak sprawdzić w jakiej ścieżce to się znajduje rgbmatrix?

ChatGPT powiedział:



Aby znaleźć, gdzie znajduje się folder lub plik `rgbmatrix`, możesz użyć polecenia `find` lub `locate`. Poniżej znajdują się instrukcje:

1. Użycie find

Uruchom poniższe polecenie, aby przeszukać katalog, w którym pracujesz, lub cały system w poszukiwaniu folderu lub pliku `rgbmatrix`:

Przeszukiwanie obecnego katalogu:

bash

Skopiuj kod

```
find . -name "rgbmatrix"
```

Przeszukiwanie całego systemu:

To może zająć trochę czasu, ale przeszuka cały system:

bash

Skopiuj kod

```
sudo find / -name "rgbmatrix"
```

2. Użycie locate

Jeśli masz zainstalowane narzędzie locate, będzie to szybsza opcja. Najpierw zaktualizuj bazę danych:

bash

Skopiuj kod

```
sudo updatedb
```

Następnie wyszukaj rgbmatrix:

bash

Skopiuj kod

```
locate rgbmatrix
```

3. Przykładowe wyniki

Wyniki pokażą pełne ścieżki do folderu lub plików o nazwie rgbmatrix. Na przykład:

bash

Skopiuj kod

```
/home/pi/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix  
/home/pi/rgbmatrix/some_other_file.cpp
```

W ten sposób możesz zidentyfikować, gdzie dokładnie znajduje się brakujący plik lub folder.

4. Co dalej

Po znalezieniu ścieżki:

- Upewnij się, że zawiera pliki źródłowe, takie jak core.cpp. Jeśli folder jest pusty lub go nie ma, prawdopodobnie coś poszło nie tak podczas pobierania biblioteki.
- Jeśli problem nadal występuje, pobierz bibliotekę ponownie zgodnie z wcześniejszymi instrukcjami.

Daj znać, czy udało Ci się znaleźć lokalizację! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ sudo python3
setup.py build cc1plus: fatal error: rgbmatrix/core.cpp: Nie ma takiego pliku ani katalogu
compilation terminated. error: command '/usr/bin/aarch64-linux-gnu-gcc' failed with exit
code 1
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ ls
rgbmatrix core.pxd core.pyx cppinc.pxd graphics.pxd graphics.pyx __init__.py Makefile
__pycache__
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
find . -name "rgbmatrix" ./build/temp.linux-aarch64-cpython-311/rgbmatrix
./build/lib.linux-aarch64-cpython-311/rgbmatrix ./rgbmatrix
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
```

ChatGPT powiedział:



Z Twoich wyników wynika, że w katalogu rgbmatrix brakuje pliku core.cpp, który jest wymagany do kompilacji. Wygląda na to, że kod źródłowy biblioteki rpi-rgb-led-matrix został niekompletnie pobrany lub coś poszło nie tak w procesie klonowania repozytorium.

1. Sprawdź, czy repozytorium zostało prawidłowo pobrane

Aby upewnić się, że masz pełne repozytorium, przejdź do katalogu głównego projektu i uruchom:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master git status
```

Jeśli repozytorium zostało poprawnie pobrane, powinieneś zobaczyć komunikat, że wszystko jest aktualne. W przeciwnym razie spróbuj ponownie pobrać repozytorium.

2. Pobierz repozytorium ponownie

Usuń obecne repozytorium i sklonuj je na nowo:

bash

Skopiuj kod

```
cd ~/Pobrane rm -rf rpi-rgb-led-matrix-master git clone https://github.com/hzeller/rpi-rgb-led-matrix.git rpi-rgb-led-matrix-master
```

3. Sprawdź obecność pliku core.cpp

Po ponownym pobraniu sprawdź, czy plik core.cpp znajduje się w katalogu bindings/python/rgbmatrix. Użyj:

bash

Skopiuj kod

```
ls ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix
```

Powinieneś zobaczyć plik core.cpp. Jeśli go brakuje, oznacza to, że coś jest nie tak z repozytorium, które pobrałeś.

4. Buduj i instaluj ponownie

Jeśli plik core.cpp jest obecny, wróć do katalogu bindings/python i spróbuj zbudować bibliotekę:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python sudo python3 setup.py build  
sudo python3 setup.py install
```

5. Dodatkowe kroki w przypadku problemów

Jeśli problem nadal występuje:

1. Zainstaluj wymagania systemowe:

bash

Skopiuj kod

```
sudo apt-get update sudo apt-get install -y build-essential python3-dev python3-pillow git
```

2. Sprawdź wersję gcc: Upewnij się, że używasz odpowiedniej wersji kompilatora:

bash

Skopiuj kod

```
gcc --version
```

3. Upewnij się, że używasz zgodnej wersji systemu operacyjnego: Biblioteka działa najlepiej na Raspberry Pi OS Lite lub Desktop (64-bit lub 32-bit).

Daj znać, czy udało Ci się znaleźć plik core.cpp i zbudować bibliotekę! 😊

Powiedziałeś(-aś):

```
masz taką ścieżkę: kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-
master/bindings/python $ ls build Makefile README.md rgbmatrix samples setup.py
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ cd ..
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings $ ls c# python
README.md kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings $ cat
README.md Various bindings to other programming languages. Typically these are
wrapping the [C-binding](../include/led-matrix-c.h) that comes with rpi-rgb-led-
matrixkopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings $ a w pliku
Makefile masz to: kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-
master/bindings/python $ cat Makefile # TODO: This contains a lot of {c|p}ython build
boilerplate, this needs cleanup. PYTHON ?= python3 SETUP := setup.py BUILD_ARGS :=
build --build-lib . INST_ARGS := install ifdef DESTDIR INST_ARGS += --root=$(DESTDIR) endif
CLEAN_ARGS := clean --all MANPAGES := $(patsubst %.txt,%, $(wildcard *.txt)) TXTTOMAN :=
a2x -f manpage # Where our library resides. It is split between includes and the binary #
library in lib RGB_LIBDIR=../lib RGB_LIBRARY_NAME=rgbmatrix
RGB_LIBRARY=$(RGB_LIBDIR)/lib$(RGB_LIBRARY_NAME).a ifneq "$(wildcard
debian/changelog)" "" PKGNAME := $(shell dpkg-parsechangelog | sed -n 's/^Source: //p')
VERSION := $(shell dpkg-parsechangelog | sed -n 's/^Version: \[^\-]*\).*\/1/p') UPSDIST :=
$(PKGNAME)-$(VERSION).tar.gz DEBDIST := $(PKGNAME)_$(VERSION).orig.tar.gz endif all:
build build: build-python install: install-python clean: clean-python find ./rgbmatrix -type f -
name \*.so -delete find . -type f -name \*.pyc -delete $(RM) build-* install-* test-*
$(RGB_LIBRARY): FORCE $(MAKE) -C $(RGB_LIBDIR) test: test-python test-python: ifneq
"$(wildcard tests/*.py)" "" nosetests -v -w tests else $(info Test suite is not implemented...)
endif ifneq "$(wildcard debian/control)" "" PYVERS := $(shell pyversions -r -v
debian/control) PYEXEC := $(shell pyversions -d) BUILD_ARGS += --
executable=/usr/bin/$(PYEXEC) INST_ARGS += --no-compile -O0 build-python:
$(PYVERS:%=build-python-%) build-python-%: $(RGB_LIBRARY) $(info * Doing build for
$(PYTHON)$* ...) $(PYTHON)$* $(SETUP) $(BUILD_ARGS) install-python:
$(PYVERS:%=install-python-%) install-python-%: $(info * Doing install for $(PYTHON)$* ...)
$(PYTHON)$* $(SETUP) $(INST_ARGS) clean-python: $(PYVERS:%=clean-python-%) clean-
python-%: $(PYTHON)$* $(SETUP) $(CLEAN_ARGS) else build-python: $(RGB_LIBRARY)
$(MAKE) -C rgbmatrix $(PYTHON) $(SETUP) $(BUILD_ARGS) install-python: $(PYTHON)
$(SETUP) $(INST_ARGS) clean-python: $(PYTHON) $(SETUP) $(CLEAN_ARGS) endif
distclean: clean dist: distclean $(info * Creating ../$(UPSDIST) and ../$(DEBDIST)) @tar --
exclude='.svn' \ --exclude='*.swp' \ --exclude='debian' \ -czvf ../$(UPSDIST) \ -C ../ $(notdir
$(CURDIR)) @cp ../$(UPSDIST) ../$(DEBDIST) @if test -d ../tarballs; then \ mv -v
../$(DEBDIST) ../tarballs; \ fi FORCE: .PHONY: FORCE .PHONY: build install test clean dist
distclean .PHONY: build-python install-python clean-python
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
```

ChatGPT powiedział:



Dzięki za szczegóły! Analizując zawartość katalogów i plik Makefile, wygląda na to, że w celu poprawnego zbudowania i zainstalowania biblioteki rgbmatrix musisz najpierw zbudować główną bibliotekę C w katalogu lib. Prawdopodobnie brak pliku core.cpp wynika z tego, że plik binarny (np. lib/librgbmatrix.a) nie został wygenerowany.

Oto jak to naprawić:

1. Zbuduj bibliotekę główną (librgbmatrix.a)

Przejdź do katalogu głównego projektu i zbuduj bibliotekę C:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master make build-all
```

To polecenie powinno zbudować wszystkie wymagane pliki w katalogu lib.

2. Sprawdź obecność lib/librgbmatrix.a

Po zakończeniu budowania upewnij się, że plik librgbmatrix.a istnieje w katalogu lib:

bash

Skopiuj kod

```
ls lib
```

Powinieneś zobaczyć plik librgbmatrix.a.

3. Zbuduj i zainstaluj bibliotekę Python

Po zbudowaniu biblioteki głównej przejdź do katalogu bindings/python i zbuduj bibliotekę Python:

bash

Skopiuj kod

```
cd bindings/python sudo python3 setup.py build sudo python3 setup.py install
```

4. Uruchom przykładowy skrypt

Po instalacji spróbuj uruchomić przykładowy skrypt:

bash

Skopiuj kod

```
cd samples python3 runtext.py
```

5. Rozwiązywanie problemów

Jeśli nadal napotkasz błędy:

- Sprawdź logi z komendy `make build-all`, aby upewnić się, że nie ma błędów w budowie.
- Zainstaluj zależności, jeśli to nie zostało jeszcze zrobione:

bash

Skopiuj kod

```
sudo apt-get install -y build-essential python3-dev python3-pillow git
```

Podsumowanie

Kluczowym krokiem jest upewnienie się, że biblioteka główna (`librgbmatrix.a`) została poprawnie zbudowana w katalogu `lib`. Dopiero potem możesz przejść do budowy i instalacji wersji Python.

Daj znać, czy to działa, albo podziel się logami z komendy `make build-all` w przypadku problemów! 😊

Powiedziałeś(-aś):

to z informacjami z czatu weźmy skasujemy najpierw wszystko i zbudujemy na nowo bo mi się miesza już

ChatGPT powiedział:



Rozumiem, że można się pogubić, więc zaczniemy od całkowitego usunięcia obecnego projektu i zbudujemy wszystko od nowa. Oto dokładne kroki:

1. Usuń obecne pliki projektu

Przejdź do katalogu, gdzie znajduje się projekt, i usuń cały folder `rpi-rgb-led-matrix-master`:

bash

Skopiuj kod

```
cd ~/Pobrane rm -rf rpi-rgb-led-matrix-master
```

2. Pobierz projekt na nowo

Sklonuj repozytorium:

bash

Skopiuj kod

```
git clone https://github.com/hzeller/rpi-rgb-led-matrix.git rpi-rgb-led-matrix-master
```

Przejdź do katalogu projektu:

bash

Skopiuj kod

```
cd rpi-rgb-led-matrix-master
```

3. Zainstaluj wymagane zależności

Upewnij się, że wszystkie wymagane narzędzia i biblioteki są zainstalowane:

bash

Skopiuj kod

```
sudo apt-get update sudo apt-get install -y build-essential python3-dev python3-pillow git
```

4. Zbuduj główną bibliotekę C

Zbuduj wszystkie pliki wymagane do działania:

bash

Skopiuj kod

```
make build-all
```

Sprawdź, czy plik lib/librgbmatrix.a został poprawnie utworzony:

bash

Skopiuj kod

```
ls lib
```

Powinieneś zobaczyć plik librgbmatrix.a.

5. Zbuduj i zainstaluj bibliotekę Python

Przejdź do katalogu bindings/python i wykonaj budowanie oraz instalację:

bash

Skopiuj kod

```
cd bindings/python sudo python3 setup.py build sudo python3 setup.py install
```

6. Uruchom przykładowy skrypt

Przejdź do katalogu z przykładowymi skryptami i uruchom:

bash

Skopiuj kod

```
cd samples/python3 runtext.py
```

7. Rozwiązywanie problemów

Jeżeli pojawią się błędy:

- Upewnij się, że wszystkie zależności systemowe są zainstalowane.
 - Sprawdź logi z komend `make build-all` i `python3 setup.py build`, aby zrozumieć przyczynę problemu.
-

Daj znać, czy wszystko działa, albo podziel się błędami, które napotkasz. Powodzenia! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master $ make build-all make: *** Brak reguł do wykonania obiektu 'build-all'. Stop. kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master $
```

ChatGPT powiedział:



Wygląda na to, że komenda `make build-all` nie działa w tym projekcie. W takim przypadku musimy zbudować bibliotekę w sposób opisany w dokumentacji projektu. Spróbujmy krok po kroku:

1. Usuń wszystkie wcześniejsze buildy

Jeśli były jakieś częściowe buildy, usuń je, aby uniknąć konfliktów:

bash

Skopiuj kod

make clean

2. Zbuduj główną bibliotekę

Spróbuj zbudować główną bibliotekę w katalogu projektu. Użyj następującego polecenia:

bash

Skopiuj kod

make

To powinno utworzyć plik lib/librgbmatrix.a w katalogu lib.

Sprawdź, czy plik istnieje:

bash

Skopiuj kod

ls lib

Jeśli zobaczysz librgbmatrix.a, oznacza to, że główna biblioteka została poprawnie zbudowana.

3. Budowa biblioteki Python

Jeśli główna biblioteka została zbudowana, przejdź do katalogu bindings/python i wykonaj budowę oraz instalację:

bash

Skopiuj kod

cd bindings/python sudo python3 setup.py build sudo python3 setup.py install

4. Uruchom testowy skrypt

Przejdź do katalogu samples i uruchom testowy skrypt:

bash

Skopiuj kod

cd samples python3 runtext.py

5. Sprawdź błędy

Jeśli pojawi się jakikolwiek błąd, upewnij się, że:

- **Wszystkie zależności są zainstalowane:**

bash

Skopiuj kod

```
sudo apt-get install -y build-essential python3-dev python3-pillow git
```

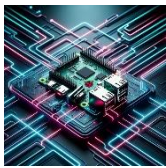
- Podaj szczegółowy komunikat błędu, jeśli problem nadal występuje.

Daj znać, czy udało się pomyślnie zbudować projekt! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ sudo python3
setup.py build cc1plus: fatal error: rgbmatrix/core.cpp: Nie ma takiego pliku ani katalogu
compilation terminated. error: command '/usr/bin/aarch64-linux-gnu-gcc' failed with exit
code 1 kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ sudo
python3 setup.py install /usr/lib/python3/dist-
packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.py
install is deprecated. Use build and pip and other standards-based tools. warnings.warn(
/usr/lib/python3/dist-packages/setuptools/command/easy_install.py:146:
EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip
and other standards-based tools. warnings.warn( cc1plus: fatal error: rgbmatrix/core.cpp:
Nie ma takiego pliku ani katalogu compilation terminated. error: command
'/usr/bin/aarch64-linux-gnu-gcc' failed with exit code 1 kopis@raspberrypi:~/Pobrane/rpi-
rgb-led-matrix-master/bindings/python $
```

ChatGPT powiedział:



Błąd `rgbmatrix/core.cpp`: Nie ma takiego pliku ani katalogu oznacza, że plik `core.cpp`, wymagany do kompilacji, nie został wygenerowany lub brakuje go w repozytorium. Przyczyna tego błędu może leżeć w niewłaściwym ustawieniu ścieżek lub w problemie z budową głównej biblioteki. Oto kroki, które mogą rozwiązać ten problem:

1. Sprawdź, czy brakuje pliku `core.cpp`

W katalogu `bindings/python/rgbmatrix` powinny znajdować się pliki źródłowe C++, takie jak `core.cpp`. Sprawdź ich obecność:

```
bash
```

Skopiuj kod

```
ls ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix
```

Jeśli brakuje `core.cpp`, oznacza to, że repozytorium nie zostało poprawnie sklonowane lub plik nie został wygenerowany.

2. Sklonuj repozytorium na nowo

Aby upewnić się, że repozytorium jest kompletne, usuń obecne repozytorium i sklonuj je ponownie:

bash

Skopiuj kod

```
cd ~/Pobrane rm -rf rpi-rgb-led-matrix-master git clone --recursive  
https://github.com/hzeller/rpi-rgb-led-matrix.git rpi-rgb-led-matrix-master
```

Uwaga: Przełącznik `--recursive` jest kluczowy, ponieważ klonuje również podmoduły, które mogą być wymagane do budowy plików.

3. Buduj główną bibliotekę

Po ponownym sklonowaniu projektu przejdź do katalogu głównego i wykonaj budowę:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master make
```

To powinno wygenerować pliki w katalogu `lib`. Upewnij się, że plik `lib/librgbmatrix.a` istnieje:

bash

Skopiuj kod

```
ls lib
```

4. Budowa biblioteki Python

Jeśli plik `librgbmatrix.a` istnieje, przejdź do katalogu `bindings/python` i spróbuj ponownie zbudować i zainstalować bibliotekę Python:

bash

Skopiuj kod

```
cd bindings/python sudo python3 setup.py build sudo python3 setup.py install
```

5. Uruchom skrypt testowy

Jeśli instalacja przebiegnie pomyślnie, przetestuj przykładowy skrypt:

bash

Skopiuj kod

```
cd samples python3 runtext.py
```

6. Dodatkowe kroki, jeśli błąd się powtarza

Jeśli problem nadal występuje:

1. Zainstaluj zależności systemowe:

bash

Skopiuj kod

```
sudo apt-get update sudo apt-get install -y build-essential python3-dev python3-pillow git
```

2. Zaktualizuj GCC i G++: Upewnij się, że masz najnowsze wersje kompilatorów:

bash

Skopiuj kod

```
gcc --version g++ --version
```

3. Sprawdź dokładny błąd: Podaj wynik komendy, abyśmy mogli dokładniej zdiagnozować problem.

Daj znać, czy to pomaga, lub przekaz więcej informacji o napotkanym błędzie! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python$ ls
~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix core.pxd core.pyx
cppinc.pxd graphics.pxd graphics.pyx __init__.py Makefile
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python$ cat
~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix/Makefile # The *.cpp files
are included in the distribution; this is only needed when # working on the pyx files. # #
Please check in modified *.cpp files with distribution to not require cython # to be installed
on the users' machine. # for python3: make PYTHON=$(which python3) CYTHON=$(which
cython3) CYTHON ?= cython3 all : core.cpp graphics.cpp %.cpp : %.pyx $(CYTHON) --cplus -
o $@ $^ clean: rm -rf core.cpp graphics.cpp kopis@raspberrypi:~/Pobrane/rpi-rgb-led-
matrix-master/bindings/python$
```

ChatGPT powiedział:



Plik Makefile w katalogu bindings/python/rgbmatrix wskazuje, że pliki core.cpp i graphics.cpp powinny zostać wygenerowane z plików core.pyx i graphics.pyx przy użyciu narzędzia cython3. To oznacza, że brakuje wygenerowanych plików .cpp, co jest przyczyną Twojego problemu.

Oto kroki, które pozwolą Ci naprawić ten problem:

1. Zainstaluj cython3

Upewnij się, że cython3 jest zainstalowany, ponieważ jest wymagany do generowania plików .cpp:

bash

Skopiuj kod

```
sudo apt-get update sudo apt-get install -y cython3
```

2. Wygeneruj pliki core.cpp i graphics.cpp

Przejdź do katalogu rgbmatrix i uruchom polecenie make z odpowiednimi zmiennymi:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix make
```

To polecenie powinno wygenerować pliki core.cpp i graphics.cpp.

Sprawdź, czy pliki zostały poprawnie wygenerowane:

bash

Skopiuj kod

ls

Powinieneś zobaczyć teraz pliki core.cpp i graphics.cpp.

3. Zbuduj bibliotekę Python

Po wygenerowaniu plików .cpp, wróć do katalogu bindings/python i spróbuj ponownie zbudować bibliotekę:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python sudo python3 setup.py build  
sudo python3 setup.py install
```

4. Uruchom przykładowy skrypt

Po instalacji spróbuj uruchomić przykładowy skrypt w katalogu samples:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/samples python3 runtext.py
```

5. Dodatkowe kroki, jeśli pojawią się błędy

- Jeśli nadal występuje problem, sprawdź logi błędów z polecenia make.
- Upewnij się, że wszystkie zależności są zainstalowane:

bash

Skopiuj kod

```
sudo apt-get install -y build-essential python3-dev python3-pillow
```

Daj znać, czy udało Ci się wygenerować pliki .cpp i zbudować bibliotekę. Jeśli pojawią się błędy, podaj ich treść, aby mogłem pomóc dalej! 😊

Powiedziałeś(-aś):

```
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix $ ls
core.cpp core.pxd core.pyx cppinc.pxd graphics.cpp graphics.pxd graphics.pyx __init__.py
Makefile
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/rgbmatrix $ cd ..
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ sudo python3 setup.py build/
invalid command name 'build/'
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $ sudo python3
setup.py build
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
sudo python3 setup.py install /usr/lib/python3/dist-
packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.py
install is deprecated. Use build and pip and other standards-based tools.
warnings.warn(
/usr/lib/python3/dist-packages/setuptools/command/easy_install.py:146:
EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip
and other standards-based tools.
warnings.warn(
zip_safe flag not set; analyzing archive
contents...
rgbmatrix.__pycache__.core.cpython-311: module references __file__
rgbmatrix.__pycache__.graphics.cpython-311: module references __file__
kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python $
```

ChatGPT powiedział:



Wygląda na to, że udało Ci się poprawnie zbudować i zainstalować bibliotekę Python rgbmatrix. Komendy setup.py build i setup.py install zakończyły się bez błędów, co wskazuje, że instalacja przebiegła pomyślnie.

Teraz możesz przetestować, czy wszystko działa poprawnie, uruchamiając jeden z przykładowych skryptów.

1. Przejdź do katalogu z przykładami

Przejdź do katalogu samples:

bash

Skopiuj kod

```
cd ~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/samples
```

2. Uruchom przykładowy skrypt

Spróbuj uruchomić przykładowy skrypt runtext.py, aby sprawdzić, czy macierz działa poprawnie:

bash

Skopiuj kod

```
python3 runtext.py
```

3. Rozwiązywanie problemów

Jeśli pojawi się błąd:

- **Upewnij się, że Twoja macierz LED jest poprawnie podłączona do Raspberry Pi.**
- **Sprawdź, czy wszystkie zależności systemowe są zainstalowane:**

bash

Skopiuj kod

```
sudo apt-get install -y build-essential python3-dev python3-pillow git
```

- **Jeśli pojawią się problemy z dostępem do GPIO, spróbuj uruchomić skrypt z uprawnieniami administratora:**

bash

Skopiuj kod

```
sudo python3 runtext.py
```

4. Inne przykładowe skrypty

Jeśli runtext.py działa, możesz spróbować innych skryptów w katalogu samples, takich jak:

bash

Skopiuj kod

```
python3 rotating-block-generator.py
```

Jeśli wszystko działa, gratulacje! Jeśli coś nadal nie działa, podaj szczegółowy komunikat błędu, abyśmy mogli to naprawić. 😊

2/2

Powiedziałeś(-aś):

plik runtext.py się uruchomił i miałem: `kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/samples $ sudo python3 runtext.py` Suggestion: to slightly improve display update, add `isolcpus=3` at the end of `/boot/cmdline.txt` and reboot (see README.md) Press CTRL-C to stop sample witaj ^Z [2]+ Zatrzymano `sudo python3 runtext.py` `kopis@raspberrypi:~/Pobrane/rpi-rgb-led-matrix-master/bindings/python/samples $` ale nixc się nie wyświetliło. Ale tak wygląda plik `runtext.py`: `#!/usr/bin/env python # Display a runtext with double-buffering. from samplebase import SampleBase from rgbmatrix import graphics import time class RunText(SampleBase): def __init__(self, *args, **kwargs): super(RunText, self).__init__(*args, **kwargs) self.parser.add_argument("-t", "--text", help="The text to scroll on the RGB LED panel", default="Hello world!") def run(self): offscreen_canvas = self.matrix.CreateFrameCanvas() font = graphics.Font() font.LoadFont("../fonts/7x13.bdf") textColor = graphics.Color(255, 255, 0) pos = offscreen_canvas.width my_text = self.args.text while True: offscreen_canvas.Clear() len = graphics.DrawText(offscreen_canvas, font, pos, 10, textColor, my_text) pos -= 1 if (pos + len < 0): pos = offscreen_canvas.width time.sleep(0.05) offscreen_canvas = self.matrix.SwapOnVSync(offscreen_canvas) # Main function if __name__ == "__main__": run_text = RunText() if (not run_text.process()): run_text.print_help()`

ChatGPT powiedział:



Skrypt `runtext.py` uruchomił się poprawnie, ale nic się nie wyświetla na macierzy. Oto kroki diagnostyczne i potencjalne rozwiązania:

1. Sprawdź połączenie macierzy

- Upewnij się, że macierz LED jest prawidłowo podłączona do Raspberry Pi. Szczegóły połączeń GPIO podałem wcześniej. Sprawdź także, czy przewody są solidnie przymocowane.
- Upewnij się, że macierz jest zasilana z zewnętrznego źródła 5V, ponieważ Raspberry Pi nie dostarczy wystarczającej mocy.

2. Skonfiguruj izolację procesora

Skrypt sugeruje dodanie `isolcpus=3` do pliku `/boot/cmdline.txt` w celu poprawy aktualizacji wyświetlacza. Możesz to zrobić tak:

1. Edytuj plik /boot/cmdline.txt:

bash

Skopiuj kod

sudo nano /boot/cmdline.txt

2. Dodaj isolcpus=3 na końcu linii.

3. Zapisz plik (CTRL+O, ENTER, CTRL+X) i zrestartuj Raspberry Pi:

bash

Skopiuj kod

sudo reboot

3. Sprawdź czcionkę

Plik runtext.py używa czcionki 7x13.bdf, która powinna znajdować się w katalogu fonts projektu. Sprawdź, czy ten plik istnieje:

bash

Skopiuj kod

ls ~/Pobrane/rpi-rgb-led-matrix-master/fonts/7x13.bdf

Jeśli pliku brakuje, pobierz go:

bash

Skopiuj kod

wget -P ~/Pobrane/rpi-rgb-led-matrix-master/fonts/ <https://github.com/hzeller/rpi-rgb-led-matrix/raw/master/fonts/7x13.bdf>

4. Przetestuj prosty przykład

Aby wykluczyć problem z hardware lub biblioteką, spróbuj uruchomić prosty skrypt testowy:

python

Skopiuj kod

```
from rgbmatrix import RGBMatrix, RGBMatrixOptions
import time
options = RGBMatrixOptions()
options.rows = 32
options.cols = 64
options.chain_length = 1
options.parallel = 1
options.hardware_mapping = 'regular' # Użyj "adafruit-hat" dla bonetu
matrix = RGBMatrix(options=options)
for i in range(10): # Pętla na 10 sekund
    matrix.Fill(255, 0, 0) # Czerwony
    time.sleep(1)
    matrix.Fill(0, 255, 0) # Zielony
    time.sleep(1)
    matrix.Fill(0, 0, 255) # Niebieski
    time.sleep(1)
```

Zapisz to w pliku simple_test.py i uruchom:

bash

Skopiuj kod

```
python3 simple_test.py
```

Macierz powinna wyświetlić czerwony, zielony i niebieski kolor.

5. Uruchom skrypt runtext.py z niestandardowym tekstem

Spróbuj uruchomić runtext.py z własnym tekstem, aby upewnić się, że jest przekazywany poprawnie:

bash

Skopiuj kod

```
sudo python3 runtext.py --text "witaj"
```

6. Rozwiązywanie problemów

- Jeśli nadal nic nie wyświetla, upewnij się, że sterownik GPIO ma dostęp do sprzętu. Spróbuj uruchomić skrypt z uprawnieniami administratora (sudo).
- Jeśli problem się powtarza, sprawdź logi błędów i podziel się szczegółami.

Daj znać, czy udało się coś wyświetlić, lub podaj wynik działania powyższych kroków. 😊

Powiedziałeś(-aś):

a tam jakiś flag sie nie dodaje i wtedy jakiś tekst

ChatGPT powiedział:



Tak, możesz użyć flag w skrypcie runtext.py, aby lepiej skonfigurować działanie programu i dostosować tekst lub inne ustawienia. Skrypt obsługuje już flagę --text, która pozwala ustawić niestandardowy tekst.

Oto jak możesz uruchomić skrypt z flagami i dodatkowymi opcjami:

1. Uruchomienie z własnym tekstem

Flaga --text umożliwia ustawienie własnego tekstu, który będzie przewijał się na macierzy:

bash

Skopiuj kod

```
sudo python3 runtext.py --text "Twój tekst tutaj"
```

Na przykład:

```
bash
```

Skopiuj kod

```
sudo python3 runtext.py --text "Witaj na RGB LED Matrix!"
```

2. Dodanie opcji sterowania jasnością

Jeśli chcesz ustawić jasność macierzy, możesz użyć opcji `--led-brightness`. Domyślnie jasność wynosi 100%, ale możesz zmienić ją na dowolną wartość od 1 do 100.

Przykład:

```
bash
```

Skopiuj kod

```
sudo python3 runtext.py --text "Hello World" --led-brightness=50
```

3. Dodanie opcji rozmiaru macierzy

Jeśli używasz niestandardowego rozmiaru macierzy, ustaw odpowiednie parametry za pomocą opcji takich jak:

- `--led-rows` (liczba wierszy, np. 32)
- `--led-cols` (liczba kolumn, np. 64)
- `--led-chain` (liczba połączonych macierzy w poziomie)

Przykład:

```
bash
```

Skopiuj kod

```
sudo python3 runtext.py --text "RGB Matrix" --led-rows=32 --led-cols=64 --led-chain=1
```

4. Inne flagi do sterowania macierzą

Skrypt obsługuje inne opcje, takie jak:

- `--led-slowdown-gpio`: Dostosowanie opóźnienia GPIO (jeśli obraz migocze, zwiększ tę wartość).
- `--led-gpio-mapping`: Mapowanie GPIO (np. regular, adafruit-hat, adafruit-hat-pwm).

Przykład:

```
bash
```

Skopiuj kod

```
sudo python3 runtext.py --text "Testowanie" --led-slowdown-gpio=2 --led-gpio-mapping=regular
```

5. Pełna lista opcji

Aby zobaczyć wszystkie dostępne flagi, możesz uruchomić skrypt z flagą `--help`:

bash

Skopiuj kod

```
sudo python3 runtext.py --help
```

Przykładowy wynik:

vbnet

Skopiuj kod

```
Usage: runtext.py [options] Options: -h, --help show this help message and exit -t TEXT, --text=TEXT The text to scroll on the RGB LED panel --led-rows=ROWS Number of rows supported by the display --led-cols=COLS Number of columns supported by the display --led-chain=CHAIN Daisy-chained boards --led-brightness=BRIGHTNESS Brightness of the panel (1-100) ...
```

6. Przykład uruchomienia z kilkoma flagami

Oto przykład uruchomienia skryptu z ustawieniem tekstu, jasności, mapowania GPIO i opóźnienia:

bash

Skopiuj kod

```
sudo python3 runtext.py --text "Witaj na LED Matrix!" --led-brightness=75 --led-slowdown-gpio=3 --led-gpio-mapping=regular
```

Jeśli po dodaniu flag nadal nie działa, upewnij się, że wszystkie podane opcje odpowiadają Twojej konfiguracji sprzętowej. Daj znać, czy udało się coś wyświetlić! 😊

Dostępna jest nowa wersja modelu GPT — Czatuj dalej, aby używać starej wersji, lub rozpocznij [nowy czat](#), aby używać najnowszej wersji.

?

ś