

# ОДНОПЛАТНІ КОМП'ЮТЕРИ RASPBERRY PI PICO W

## RASPBERRY PI PICO SINGLE BOARD COMPUTERS

У статті наведена коротка інформація про модуль Raspberry Pi Pico W призначений для використання у вбудованих системах Інтернету речей. Наведені основні характеристики модуля та наведені рекомендації з його програмування.

V. Makarenko

*Abstract – This article provides brief information about the Raspberry Pi Pico W module designed for use in embedded systems of the Internet of things. The main characteristics of the module are given and recommendations for its programming are given.*

V. Makarenko

Одноплатні комп'ютери все частіше використовуються в промислових проектах і в Інтернеті речей [1]. Як показують звіти дистриб'юторів, половина опитаних професійних інженерів використовують одноплатні комп'ютери (SBC, Single-Board Computer) в промислових виробках і пристроях Інтернету речей (IoT). Найпопулярнішою платою є Raspberry Pi, якій віддають перевагу 44% професійних користувачів. Друге місце зайняв Arduino (28 відсотків), а на третьому місці з 6 відсотками знаходиться плата Beagleboard, підтримувана Texas Instruments.

SBC використовуються на всіх етапах розробки і виробництва продукції, причому, як і слід було очікувати, 23% респондентів використовують їх для підтвердження концепції, а 35% – для створення прототипів. Хоча плати також використовувалися в дрібносерійному виробництві, опитування показало, що 22% встановлюють недорогі SBC у виробничому обладнанні; при цьому близько 20% такої продукції випускається в обсягах від 5 тис. або більше на рік, а 20% використовується для розробки випробувального обладнання та тестування.

Глобальне опитування проводилося з березня по травень 2021 року і зібрало майже 1500 відповідей від професійних інженерів, розробників і виробників, які працюють над рішеннями в області SBC. Три чверті респондентів були професійними користувачами, а лише чверть-любителями. Питання формулювалися таким чином, щоб зрозуміти, як популярні SBC від деяких провідних світових виробників використовуються в професійних продуктах і проектах.

Близько 24% професіоналів створюють власні плати для використання з SBC, демонструючи переваги стандартної обчислювальної платформи з користувацькими інтерфейсами вводу-виводу у багатьох додатках.

Однак лише 20% інженерів зараз використо-

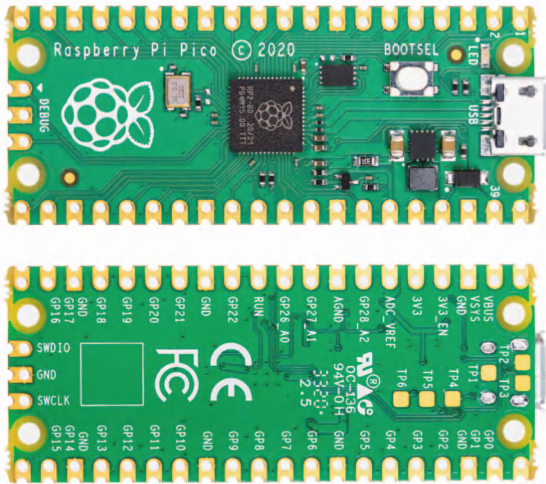
вують штучний інтелект (AI) та машинне навчання у своїх додатках на SBC, що може бути пов'язано з обмеженою продуктивністю плат. Найбільш поширеними побажаннями розробників до виробників SBC є підвищення продуктивності при вирішенні завдань AI і збільшення обсягу пам'яті.

Найпопулярнішим аксесуаром на сьогоднішній день, безумовно, є сенсорні екрани, проте також користуються попитом камери і комплекти для додаткового живлення від акумуляторів або сонячних батарей.

Використовувати плати, адаптовані до конкретного додатка, з більшою ймовірністю будуть професійні користувачі, ніж любителі. "Це дослідження чітко демонструє широке поширення SBC серед інженерів, які створюють комерційні, IoT та промислові програми. Успіх Raspberry Pi та Arduino зумовлений потужним поєднанням високої продуктивності, універсальності, функцій та аксесуарів, низької вартості та великої активної спільноти користувачів, що робить ці платформи дуже привабливими для інженерів, які прагнуть швидко розробляти та впроваджувати у виробництво широкий спектр додатків. Використання SBC в прототипах надає інженерам можливості значного зниження витрат і набагато більш швидкого виведення нових продуктів на ринок", – сказав Ромен Соро (Romain Soreau), керівник відділу одноплатних обчислювальних пристроїв компанії Farnell.

На сьогоднішній день Farnell продала понад 15 мільйонів одиниць і пропонує повний асортимент одноплатних комп'ютерів Raspberry Pi, включаючи випущений у 2020 році Raspberry Pi Pico на базі процесора RP2040 (рис. 1), а також аксесуари, включаючи чохла, блоки живлення, кабелі мікро-HDMI та камери високої чіткості Raspberry Pi.

Компанія Raspberry Pi активізувала розробку своїх чіпів, випустивши чотириядерний мікроконтролер з двома ядрами ARM Cortex-M33 і двома



**Рис. 1. Одноплатний комп'ютер Raspberry Pi Pico**

власними ядрами RISC-V. Планується до випуску плата Raspberry Pi Pico 2, що використовує мікроконтролер RISC-V RP2350, а безпроводову версію Pico 2 W з модемом від Infineon Technologies.

Процесор RP2350 вдвічі більший за розміром, має більш високу тактову частоту ядра (150 МГц), вдвічі більший обсяг пам'яті (520 кбайт SRAM в десяти банках) і нові функції безпеки в порівнянні з RP2040, випущеним більше трьох років тому, з двома ядрами M0+. Розмір матриці становить 5.3 мм<sup>2</sup> проти 2 мм<sup>2</sup> у RP2040, але найкомпактніша версія RP2350A буде всього на десять центів дорожче і буде коштувати 0.80 долара за 3400 одиниць або 1.1 долара за одиницю продукції.

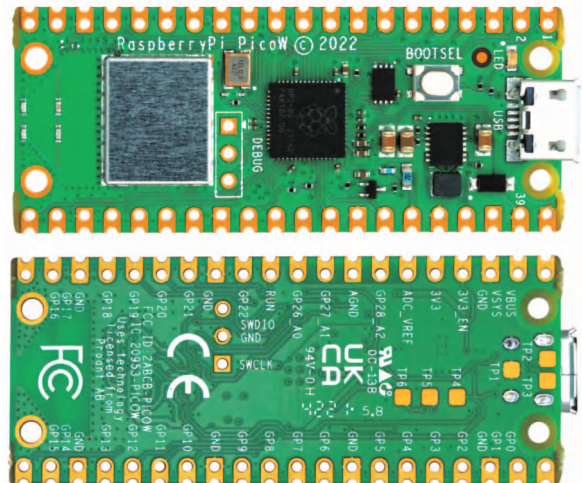
Крім ядер M33, в RP2350 додані два ядра RISC-V, які можуть бути виділені під час завантаження. Завантажувальний ПЗП автоматично визначає архітектуру для двійкового файлу другого рівня та перезавантажує чіп у відповідний режим.

На відміну від одного варіанту корпусу QFN56 розміром 7×7 мм для RP2040, RP2350 випускається у корпусі QFN60 розміром 7×7 мм (RP2350A) з роздільною здатністю 30 точок на дюйм або корпусі QFN80 розміром 10×10 мм (RP2350B) з роздільною здатністю 48 точок на дюйм, а також варіанти з 2 МБ вбудованою захищеною flash-пам'яттю QSPI (RP2354A і RP2354B).

Pico 2 і RP2350 підтримуються оновленою версією Pico SDK, а також новими образами MicroPython і CircuitPython. Метою проекту Trusted Firmware є створення RP2350 як еталонної апаратної платформи для випуску довгострокової підтримки Trusted Firmware-M 2.1.0, а TF-M надає еталонну реалізацію для сертифікованих чіпів Arm V8-

M, надаючи розробникам простий спосіб захистити пристрої від поширених атак.

Розглянемо основні особливості Raspberry Pi Pico W [2] з вбудованим модулем Wi-Fi (рис. 2). На рис. 3 наведено розташування виводів плати Raspberry Pi Pico W [3].



**Рис. 2. Одноплатний комп'ютер Raspberry Pi Pico W**

Основні характеристики Raspberry Pi Pico W:

- 32-розрядний двоядерний процесор Rp2040 Cortex-M0 + з частотою 133 МГц
- 2 МБ флеш-пам'яті Q-SPI
- 264 КБ SRAM-пам'яті
- 26 GPIO (сумісність з +3.3 В)
- вбудований 12-розрядний АЦП
- прискорені вбудовані бібліотеки обчислень з плаваючою комою
- вбудований односмуговий чіп Infineon CYW43439 (безпроводовий інтерфейс 2,4 ГГц (802.11 b/g/n) та Bluetooth® 5.4
- послідовний порт налагодження (SWD)
- порт Micro-USB (USB 1.1) для живлення (+5 В) і даних
- 2 × UART, 2 × I2C, 2 × інтерфейс шини SPI
- 16 каналів ШІМ
- 1× таймер (з 4 будильниками), 1× лічильник реального часу
- бортовий датчик температури
- вбудований світлодіод на GPIO, керований модулем 43439
- зазубрений модуль, що дозволяє припаювати безпосередньо до плат
- 8 програмованих кінцевих автоматів вводу-виводу (PIO)
- програмування на MicroPython, C, C++
- швидке програмування через USB

- номінальна напруга VBUS +5 В ±10 %
- діапазон напруги VSYS від +1.8 В до +5.5 В
- діапазон робочих температур від -20 до +85 °С.

Вбудовану 2 Мбайт Q-SPI-флеш-пам'ять можна перепрограмувати або за допомогою послідовного порту налагодження, або за допомогою спеціального USB-порту.

Найпростіший спосіб перепрограмувати флеш-пам'ять Pico – це використовувати USB. Для цього потрібно вимкнути живлення плати і потім, утримуючи кнопку BOOTSEL (біла кнопка на рис. 2) включити живлення плати. Після цього Pico буде відображатися як USB-накопичувач. Перетягування спеціального файлу '.uf2' на диск призведе до запису цього файлу на флешку та перезапуску Pico. Завантажувальний код USB зберігається в ПЗП (ROM) RP2040, тому його неможливо випадково перезаписати.

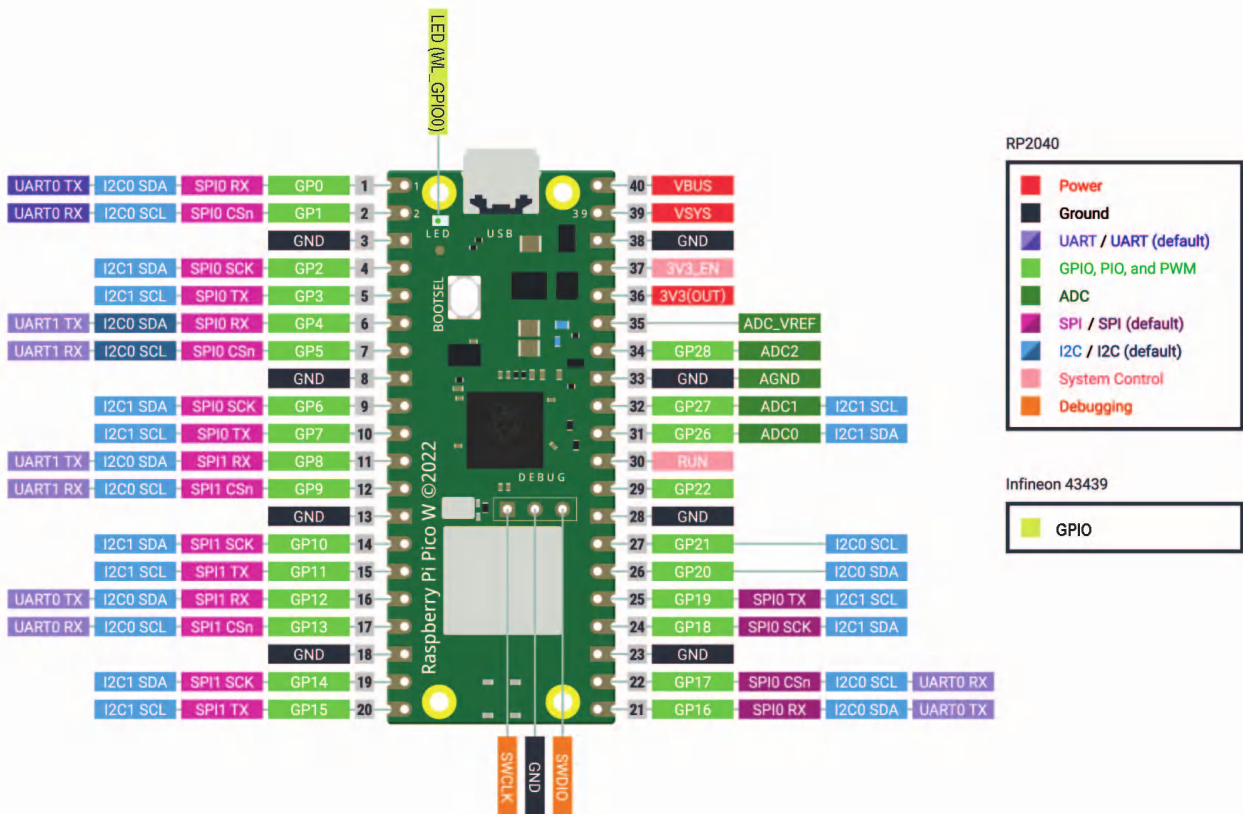
Плата сумісна з макетною платою (тобто відстань між контактами 0,1 дюйма), і після пайки контактів плату можна підключити до макетної плати для простого підключення до контактів GPIO за допомогою перемичок.

На одному краю плати знаходиться порт мікро-

USB для подачі живлення на плату і для програмування плати. Поруч з USB-портом знаходиться вбудований спеціальний світлодіод, який можна використовувати під час розробки програм. Поруч із цим світлодіодом знаходиться кнопка BOOTSEL (рис. 2), яка використовується під час програмування мікроконтролера. Поруч з чіпом процесора є 3 отвори, до яких можна зробити зовнішні підключення. Вони використовуються для налагодження програм за допомогою Serial Wire Debug (SWD). На іншому краю плати знаходиться однодіапазонний модуль Wi-Fi 2,4 ГГц (802.11n). Поруч з модулем Wi-Fi розташована бортова антена.

На рис. 2 наведено вид ззаду апаратного модуля Pico. На рис. 2 і 3 всі контакти GPIO позначені буквами і цифрами:

- GND – загальний джерела живлення (цифровий загальний)
- AGND – загальний джерела живлення (аналоговий загальний)
- 3V3 – живлення +3,3 В (вихід)
- GP0 – GP22 – цифрові виводи GPIO
- GP26\_A0 – GP28\_A2 – аналогові входи
- ADC\_VREF – опорна напруга АЦП



- TP1 – TP6 – контрольні точки
  - SWDIO, GND, SWCLK – інтерфейс налагодження
  - RUN – вивід RUN за замовчуванням, низький рівень для скидання
    - 3V3\_EN – за замовчуванням включає живлення +3,3 В (живлення можна відключити, підключивши цей контакт до загального)
    - VSYS – вхідна напруга системи (від 1.8 В до 5.5 В), для створення живлення +3.3 В для плати
    - VBUS – вхідна напруга micro-USB (+5 В).
- Деякі контакти GPIO використовуються для внутрішніх функцій плати:
- GP29 (input) – використовується в режимі АЦП (ADC3) для вимірювання VSYS / 3
  - GP24 (input) – вхід якщо рівень VBUS sense високий, якщо VBUS присутній, інакше низький
  - GP23 (output) – управляє виводом енергозбереження SMPS.

Вбудований імпульсний DC/DC-перетворювач +3.3 В використовується для живлення RP2040 в діапазоні вхідної напруги від 1.8 В до +5.5 В. Наприклад, 3 лужні батареї типу AA можуть використовуватися для забезпечення +4,5 В для живлення Pico.

Піко може житися кількома способами. Найпростіший спосіб-використовувати USB-порт комп'ютера або адаптер живлення + 5 В. Треба подати живлення на вхід VSYS (рис. 4) через діод Шотткі. Напруга на вході VSYS буде дорівнювати напрузі VBUS за вирахуванням падіння напруги на діоді Шотткі (близько 0.7 В). Виводи VBUS і VSYS можуть бути замкнені, якщо плата живиться від зовнішнього USB-порту +5 В. Напруга VSYS подається на SMPS через DC/DC-перетворювач RT6150, який формує фіксовану напругу +3.3 В для

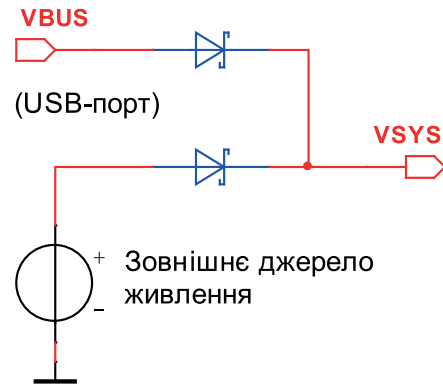


Рис. 4. Підключення зовнішнього джерела живлення

живлення процесора та інших частин плати. Напруга VSYS ділиться на три і доступна через аналоговий вхідний порт GPIO29 (ADC3), який легко контролювати. GPIO24 перевіряє наявність напруги VBUS і має високий логічний рівень, якщо напруга VBUS присутня.

На рис. 5 наведена принципова схема кола живлення модуля.

На рис. 6 наведена спрощена структура апаратного модуля Pico. Треба звернути увагу, що контакти GPIO безпосередньо підключені від мікросхеми процесора до роз'єму GPIO. GPIO26-28 можна використовувати або як цифровий GPIO, або як вхід АЦП. Входи АЦП GPIO26-29 мають зворотні діоди на 3 В, тому вхідна напруга не повинна перевищувати 3.6 В.

### Програмування Raspberry Pi Pico W

Хоча Raspberry Pi Pico за замовчуванням налаштований для використання з потужними і популярними мовами C / C++, багатьом новачкам простіше використовувати MicroPython-версію

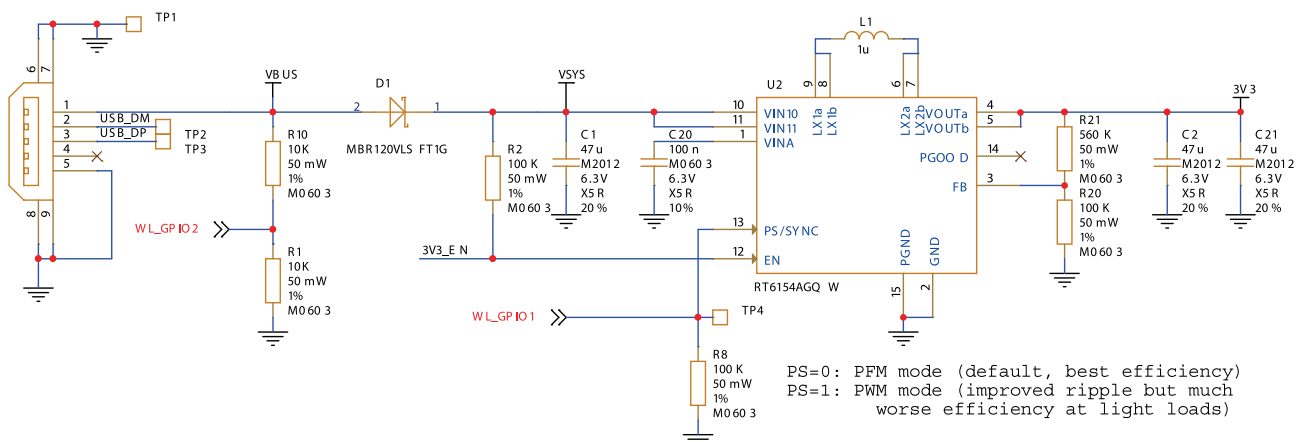
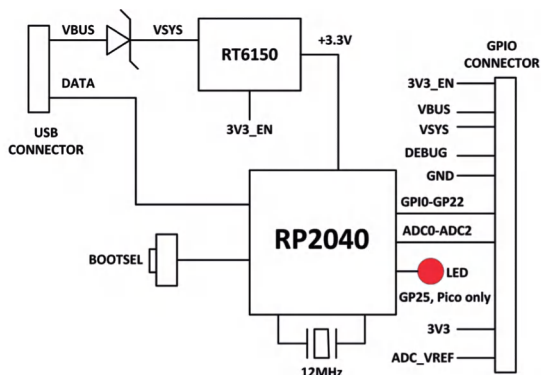


Рис. 5. Принципова схема кола живлення модуля Raspberry Pi Pico



**Рис. 6. Спрощена структура апаратного модуля Raspberry Pi Pico**

мови програмування Python, розроблену спеціально для мікроконтролерів.

Перед використанням плати потрібно встановити MicroPython на Raspberry Pi Pico W. Після встановлення MicroPython залишається на Pico, поки він не буде перезаписаний чимось іншим [4, 5]. Для встановлення MicroPython потрібне підключення до Інтернету, і це потрібно лише один раз. Це можна зробити або за допомогою Raspberry Pi (наприклад, Raspberry Pi 4), або за допомогою ПК. Завантажити MicroPython можна за посиланням [6].

Для встановлення MicroPython потрібно:

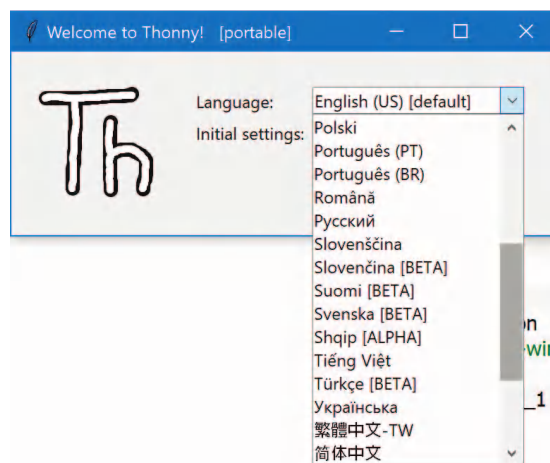
- натиснути і утримувати кнопку BOOTSEL на платі Pico
- підключити Pico до порту USB ПК за допомогою кабелю мікро-USB, утримуючи кнопку натиснутою
- зачекати кілька секунд і відпустити кнопку BOOTSEL (ви повинні побачити, що Pico відображається як знімний диск з ім'ям RPI-RP2)
- перетягнути завантажений файл MicroPython UF2 на диск RPI-RP2 (Raspberry Pi Pico перезавантажиться, і після цього можна використовувати MicroPython на Pico.

Після вимикання програма залишається у флеш-пам'яті. У джерелах [4...6] рекомендується встановити текстовий редактор Thonny з вбудованим інтерпретатором Python на ПК [7]. Завантажити останню портативну версію редактора можна за посиланням [8].

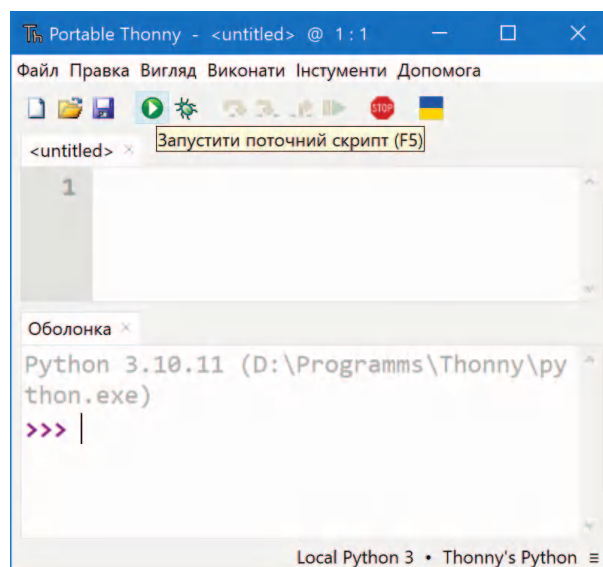
Після розпакування і запуску програми виводиться вікно (рис. 7) у якому можна обрати мову інтерфейсу.

Після вибору мови відкривається робоче вікно програми (рис. 8).

Для прикладу розглянемо просту програму обчислення середнього значення двох чисел (Average



**Рис. 7. Вибір мови інтерфейсу при першому запуску редактора Thonny**

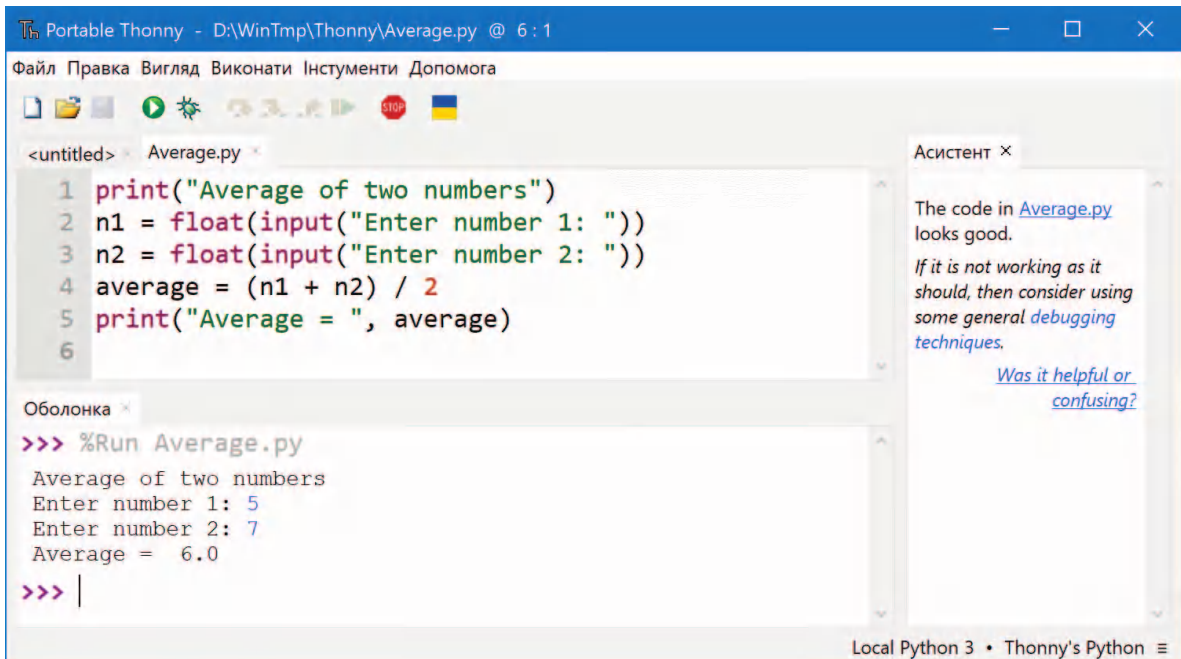


**Рис. 8. Робоче вікно редактора Thonny**

ge), що вводяться з клавіатури [10]. Результат виконання наведено на рис. 9.

**Приклад 2:** середнє значення 10 чисел, прочитаних з клавіатури У цьому прикладі з клавіатури зчитуються 10 чисел і відображається їх середнє значення. Мета цього прикладу-показати, як можна побудувати цикл у Python.

Програма називається Average 10 [10], а лістинг програми і приклад запуску програми показані на рис. 10. У цій програмі будується цикл, який виконується від 0 до 9 (тобто 10 разів). У середині цього циклу числа зчитуються з клавіатури, складаються один з одним і зберігаються в змінній sum. Потім обчислюється середнє значення і відображається шляхом ділення sum на 10. Зверніть увагу, що новий рядок не друкується після операторів друку,



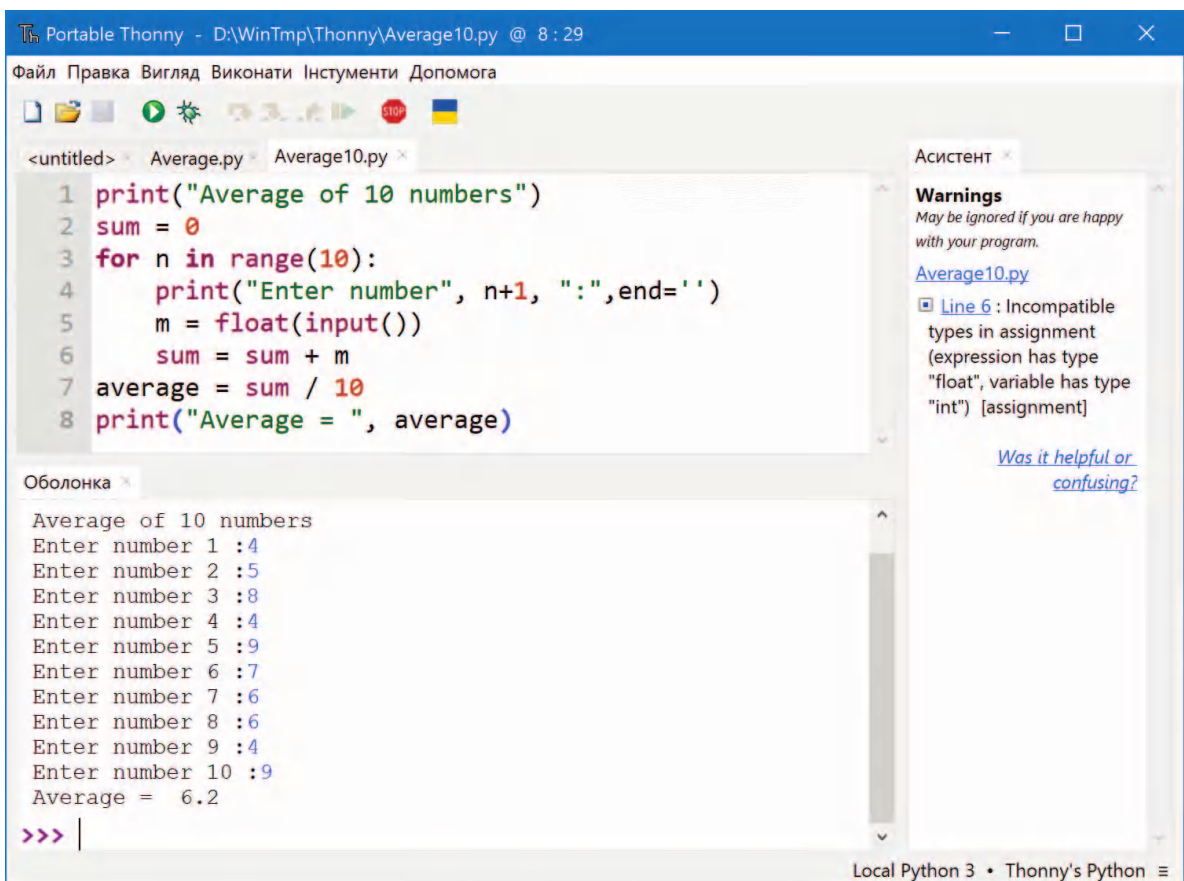
```
Portable Thonny - D:\WinTmp\Thonny\Average.py @ 6 : 1
Файл Правка Вигляд Виконати Інструменти Допомога
<untitled> Average.py
1 print("Average of two numbers")
2 n1 = float(input("Enter number 1: "))
3 n2 = float(input("Enter number 2: "))
4 average = (n1 + n2) / 2
5 print("Average = ", average)
6

Оболонка
>>> %Run Average.py
Average of two numbers
Enter number 1: 5
Enter number 2: 7
Average = 6.0
>>> |

Асистент
The code in Average.py looks good.
If it is not working as it should, then consider using some general debugging techniques.
Was it helpful or confusing?

Local Python 3 • Thonny's Python
```

Рис. 9. Результат виконання програми Average



```
Portable Thonny - D:\WinTmp\Thonny\Average10.py @ 8 : 29
Файл Правка Вигляд Виконати Інструменти Допомога
<untitled> Average.py Average10.py
1 print("Average of 10 numbers")
2 sum = 0
3 for n in range(10):
4     print("Enter number", n+1, ":", end='')
5     m = float(input())
6     sum = sum + m
7 average = sum / 10
8 print("Average = ", average)

Оболонка
Average of 10 numbers
Enter number 1 :4
Enter number 2 :5
Enter number 3 :8
Enter number 4 :4
Enter number 5 :9
Enter number 6 :7
Enter number 7 :6
Enter number 8 :6
Enter number 9 :4
Enter number 10 :9
Average = 6.2
>>> |

Асистент
Warnings
May be ignored if you are happy with your program.
Average10.py
Line 6 : Incompatible types in assignment (expression has type "float", variable has type "int") [assignment]
Was it helpful or confusing?

Local Python 3 • Thonny's Python
```

Рис. 9. Результат виконання програми Average

оскільки всередині оператора друку використовується опція `end=""`.

В [10, 11] наведена велика кількість прикладів

для першого знайомства з MicroPython.

Розглянемо ще один приклад що демонструє як зовнішній світлодіод можна підключити до Pico.

Схема підключення світлодіода наведена на рис. 11. На початку програми GP1 налаштований як ви-

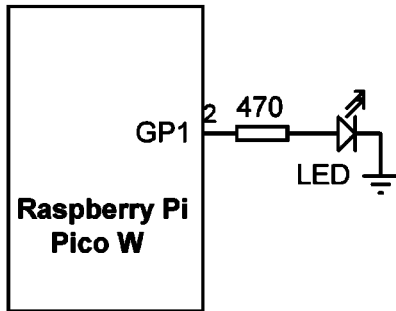


Рис. 11. Схема підключення світлодіода до Raspberry Pi Pico

хід. Потім формується цикл `while`, який виконується до тих пір, поки користувач не зупинить його. Всередині цього циклу світлодіод включається і вимикається з затримкою в одну секунду (рис. 12).

При відсутності зв'язку з Raspberry Pi Pico у вікні Асистент виводиться повідомлення про похибку. Тому при програмуванні потрібно мати підключе-

ний модуль для перевірки програми. Звісно, у рамках однієї статті розглянути всі аспекти роботи з модулем Raspberry Pi Pico неможливо. Більш детально з програмуванням на MicroPython з великою кількістю прикладів можна ознайомитись у [10, 11].

## ЛІТЕРАТУРА

1. Nick Flaherty Raspberry Pi launches its first RISC-V multicore chip. Technology News, August 9, 2024. / <https://www.eenewseurope.com/en/raspberry-pi-launches-its-first-risc-v-multicore-chip/>
2. RASPBERRY-PI RASPBERRY PI PICO W / <https://uk.farnell.com/raspberry-pi/raspberry-pi-pico-w/raspberry-pi-board-arm-cortex/dp/3996082>
3. Raspberry Pi Pico W Datasheet / [https://www.farnell.com/datasheets/3759470.pdf?\\_gl=1\\*1vnsnie\\*\\_gcl\\_au\\*Mzc0OTk4Mzg1LjE3MjQxNDQyNTA](https://www.farnell.com/datasheets/3759470.pdf?_gl=1*1vnsnie*_gcl_au*Mzc0OTk4Mzg1LjE3MjQxNDQyNTA).
4. Getting started with pico / <https://datasheets.raspberrypi.com/pico/getting-started-with-pico.pdf>

The screenshot shows the Thonny IDE interface. The main editor window displays the following Python code:

```

11 #-----
12 from machine import Pin
13 import utime
14
15 LED = Pin(1, Pin.OUT) # LED at GP1
16
17 while True: # DO FOREVER
18     LED.value(1) # LED ON
19     utime.sleep(1) # Wait 1 second
20     LED.value(0) # LED OFF
21     utime.sleep(1) # Wait 1 second

```

The console window shows the following error message:

```

>>> %Run ExtFlash.py
Traceback (most recent call last):
  File "D:\WinTmp\Thonny\ExtFlash.py", line 12, in <module>
    from machine import Pin
ModuleNotFoundError: No module named 'machine'
>>>

```

The Assistant window on the right displays the error details:

```

Асистент
ModuleNotFoundError:
No module named
'machine'
cp_back.py, line 353
No specific suggestions for
this error (yet).
[+] Let Thonny developers
know
[+] Search the web
Was it helpful or
confusing?
General advice on dealing
with errors.

```

Рис. 12. Лістинг програми та результат її виконання на ПЕ без підключеного модуля Raspberry Pi Pico

5. Beginner's Guide for Raspberry Pi Pico / [https://files.seeedstudio.com/wiki/Grove\\_Shield\\_for\\_Pi\\_Pico\\_V1.0/Beginner%27s-Guide-for-Raspberry-Pi-Pico.pdf](https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Beginner%27s-Guide-for-Raspberry-Pi-Pico.pdf)

6. Raspberry\_Pi\_Pico\_Advanced / [https://www.elecrow.com/download/product/RPK13250K/Raspberry\\_Pi\\_Pico\\_Advanced\\_Kit\\_tutorial\\_V1.0.pdf%3Fsrsltid%3DAfmBOoo3AfumIWhk-QYzJq73Um852U5Gm0PXvfkiZriQg8vBjhtSEBnF&ved=2ahUKEwid0\\_zYpIOIAxVYRvEDHeaqB-IQF-noECBIAQ&usg=AOvVaw3XYDASG48IRTz3PA2Eu-qky](https://www.elecrow.com/download/product/RPK13250K/Raspberry_Pi_Pico_Advanced_Kit_tutorial_V1.0.pdf%3Fsrsltid%3DAfmBOoo3AfumIWhk-QYzJq73Um852U5Gm0PXvfkiZriQg8vBjhtSEBnF&ved=2ahUKEwid0_zYpIOIAxVYRvEDHeaqB-IQF-noECBIAQ&usg=AOvVaw3XYDASG48IRTz3PA2Eu-qky)

7. MicroPython /

<https://www.raspberrypi.com/documentation/micro-controllers/micropython.html>

8. Thonny Python IDE for beginners / <https://thonny.org/>

9. <https://github.com/thonny/thonny/releases/tag/v4.1.4>

10. Dogan Ibrahim Raspberry Pi Pico W. Program, build, and master 60+ projects with the Wireless RP2040. – Elektor International Media B.V., 2022. – 310 p.

11. Gьnter Spanner's MicroPython for Microcontrollers. – Elektor International Media B.V., 2021. – 215 p.



# ДРУКОВАНІ ПЛАТИ

## ▪ проектування

Проектування друкованих плат в НВФ VD MAIS – це:

- багаторічний досвід конструювання і знання специфіки:
  - . розробки електронних пристроїв
  - . технології виготовлення друкованих плат
  - . різних CAD-CAM систем проектування (P-CAD, Mentor Graphics, CAM-350 та ін.)
- повна реалізація можливостей виготовлення друкованих плат і контрактного виробництва VD MAIS
- система менеджменту якості фірми сертифікована на відповідність вимогам стандартів ISO 9001:2015, ISO 14001:2015, IATF 16949:2016 і ISO 13485:2016

[ Від ремесла до мистецтва ]

**VD MAIS**  
ДИСТРИБ'ЮЦІЯ +  
КОНТРАКТНЕ  
ВИРОБНИЦТВО

**VD MAIS**  
PCB Professionally

тел.: (0-44) 220-0101, info@vdmiais.kiev.ua, www.vdmiais.kiev.ua