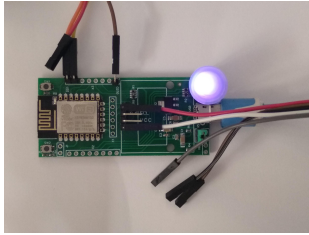


GCemu20_V1 емулятор лічильника радіоактивних частинок Гейгера



Опис

Емулятор детектора іонізуючого випромінювання GCemu20_V1 - готовий до використання пристрій, розроблений IoT-devices LLC, який виконує повну емуляцію сенсора радіації типу лічильника Гейгера з імпульсним виходом та трубкою СБМ-20, такого, як-от GGreg20_V3.

Призначення

Емулятор детектора радіоактивних частинок - програмно-апаратний електронний модуль, призначений для емуляції лічильника рівня іонізуючого випромінювання. З цією метою емулятор включає в себе вихід підрахунку імпульсів на головний контролер. Arduino, ESP8266, ESP32, Raspberry Pi, STM32 та інші можуть використовуватися як хост-контролер.

Симульований рівень радіації та режим роботи емулятора відображається світловими сигналами на вбудованому RGB-світлодіоді.

GCemu20_V1 – недорогий та корисний прилад для:

- безпечного та системного навчання,
- модульного тестування і розробки нових пристроїв,
- виявлення та налагодження несправностей у системах, що вже працюють, тощо.

Цей модуль корисний під час навчання, тестування, та побудови вимірювальних приладів з визначення потужності іонізуючого випромінювання як в приміщенні, так і зовні, як в ручному / кишеньковому, так і в стаціонарному дизайні.

Єдине, що потрібно, щоб почати використовувати модуль емулятора, це будь-який мікроконтролер, який може підраховувати кількість вхідних імпульсів за одиницю часу на своєму GPIO, а також живлення через micro USB.

Специфікації

1. Розміри модуля - 30 x 65 x 10 мм. За основу взято модуль [ESP12.OLED](#) без дисплея.
2. Генератор справжніх випадкових чисел (True Random Number Generator) - вбудований TRNG ESP8266.
3. Потужність імітованої радіації: 5 режимів від 0 до 1.5 мкЗв/год.
4. Живлення: від AC/DC 5V адаптера (не постачається) через micro USB.
5. На платі модуля передбачено отвори 2.54мм під пайку для підключення консолі через UART. Консоль може застосовуватися під час експлуатації емулятора чи прошивки вбудованого у GCemu20_V1 контролера ESP8266.
6. Струм споживання - аналогічний до модуля ESP12.OLED_V1 та складає до 80 мА з включеним інтерфейсом WiFi (WiFi і АЦП вимагається TRNG).
7. Імпульсний вихід GCemu20_V1 сумісний з рівнями логічного сигналу 3V3 ACTIVE-LOW.
8. Тривалість імпульса на виході - близько 10 мкс, аналогічно GGreg20.
9. Програмне забезпечення: прошивка NodeMCU / Lua. Код емулятора стартує автоматично після подачі живлення у Режимі 1 (імітація нормального рівня радіації оточуючого середовища [18-35] CPM).

Для чого і кому потрібен емулятор лічильника Гейгера

Головна ідея будь-якого емулятора в галузі DIY-електроніки полягає у тому, щоб під час розробки IoT-пристроїв, проведення експериментів чи навчання, замість реального модуля тимчасово, на певних етапах, використовувати віртуальний заміник, який дозволяє відтворювати роботу та характеристики реального пристрою з високою точністю. Емулятор має спростити та прискорити розробку, а також додати зручності на початкових етапах запланованого проекту, або під час виконання модульних тестів.

Переваги емулятора лічильника Гейгера

- Відсутність високої напруги на модулі
- Спрощений навчальний процес
- Нижча вартість порівняно зі справжнім лічильником Гейгера
- Реальне джерело радіації не потрібне
- Ресурс трубки Гейгера-Мюллера не виснажується, адже її немає
- Дані налагодження доступні через UART-консоль

Користувачі емулятора

- Радіоаматори у царині IoT- та DIY- мікроелектроніки
- Викладачі та студенти технологічних університетів
- Дослідницькі команди та інститути
- Батьки та діти, що опановують нові технології самостійно вдома
- Розробники та тестувальники стаціонарних та/чи кишенькових дозиметрів
- Випробувальні лабораторії з контролю якості та/чи захисту прав споживачів

Обмеження емулятора

Серед обмежень даного емулятора лічильника Гейгера, які наразі відомі, є кількість пам'яті контролера ESP8266, в якій в циклі створюється необхідна кількість одноразових таймерів з випадковим часом спрацювання. Кожен таймер по суті є функцією, яка займає певну оперативну пам'ять.

Таймери що спрацювали, одразу ж вивільняють пам'ять. Виконання програмного коду нагадує пружину, що у циклі раз на хвилину різко стискається і поволі розтискається в межах наявної пам'яті контролера.

Таким чином, максимально можлива кількість випадкових подій, що генерується на виході емулятора, напряму залежить від кількості вільної оперативної пам'яті та швидкодії контролера.

Експериментально встановлено, що ESP8266 з прошивкою NodeMCU та мовою Lua здатен впевнено генерувати близько 500 - 550 подій на хвилину, або близько 1,5 мкЗв на годину. Це більш ніж достатня кількість імпульсів на хвилину для емулятора та рівнів радіації, які він начебто реєструє.

Як працює емулятор

Вбудоване програмне забезпечення в циклі створює в межах хвилини певну кількість одноразових таймерів. Кожен таймер, коли спрацьовує, ініціює на GPIO імпульсного виходу емулятора логічний рівень "1" (логіка ACTIVE-LOW).

Тривалість логічного рівня "1" є близькою до 10 мікросекунд і є аналогічною до імпульсів на справжньому модулі лічильника Гейгера GGreg20_V3. Відмінність полягає лише у тому, що GGreg20_V3 може також підтримувати логіку 5V, натомість GCemu20_V1 підтримує лише 3V3-логіку.

Кількість імпульсів на виході емулятора та відповідних їм випадкових таймерів, в межах хвилини, задається випадково та має встановлений діапазон, що відповідає поточному режиму роботи емулятора. Емулятор може працювати в одному з п'яти режимів імітації потужності радіації.

Режими потужності радіації, яку імітує модуль GCemu20_V1, підбрано таким чином, щоб охопити весь спектр задач, в яких може бути доцільно використовувати емулятор:

Режим 0. Немає імпульсів (імітація помилки сенсора);

Режим 1. Природне фонове випромінювання (за замовчуванням після Power-On Reset);

Режим 2. Припустимий рівень випромінювання;

Режим 3. Підвищений рівень випромінювання;

Режим 4. небезпечний рівень випромінювання.

Після подачі живлення на емуляторі за замовчуванням встановлюється Режим 1.

Щоб змінити режим роботи емулятора, потрібно натиснути кнопку Flash / D3 (SW1 на платі модуля).

Режими обираються по черзі натисканням вбудованої кнопки Flash:

Режим 1 -> Режим 2 -> Режим 3 -> Режим 4 -> Режим 0 -> **Режим 1** ...

Для кожного режиму потужності на вбудованому RGB-світлодіоді призначено свій колір так, щоб користувач бачив не лише вихідні імпульси від емулятора до хост-контролера, а також міг розрізняти поточний режим роботи:

Режим роботи	Еквівалент потужності випромінювання	Імпульси на хвилину (CPM)	Колір спалахів	R	G	B
Режим 0	0 мкЗв/год	0	немає спалахів black	0	0	0
Режим 1	0.1 - 0.2 мкЗв/год	18 - 35	cyan	0	1	1
Режим 2	0.2 - 0.3 мкЗв/год	36 - 52	green	0	1	0
Режим 3	0.3 - 0.6 мкЗв/год	53 - 105	red	1	0	0

Режим 4	0.6 мкЗв/год - 1.5 мкЗв/год	106 - 264	magenta	1	0	1
---------	-----------------------------	-----------	---------	---	---	---

Реальний модуль GGreg20_V3 оснащено трубкою Гейгера радянського виробництва СБМ-20. Ця трубка має наступний коефіцієнт перерахунку [імпульсів на хвилину, CPM] у [мікрозіврти на годину] для джерела Цезію-137:

$$\text{мкЗв на годину} = \text{CPM} * 0.0057$$

Щоб розрахувати для діапазонів радіації вказану у таблиці вилку кількості імпульсів на годину, які мав би генерувати емулятор, працюючи у певному режимі, виконаємо обернену операцію:

$$\text{CPM} = \text{мкЗв на годину} / 0.0057$$

За зібраними IoT-devices LLC статистичними даними, трубка СБМ-20 є найпопулярнішою серед DIY проектів, саме тому в емуляторі зроблено прив'язку програмних характеристик емулятора GCemu20_V3 до цієї трубки.

Для забезпечення справжньої випадковості імпульсів на виході емулятора, застосовується апаратний генератор справжніх випадкових чисел контролера ESP8266 (True Random Number Generator, TRNG).

Щоб підсилити ефект випадковості, кожен щохвилинний цикл роботи емулятора генерує не стає число імпульсів, а знову ж таки, випадково обирає кількість вихідних імпульсів з діапазону значень, вказаного для кожного режиму у таблиці вище.

Наприклад, Режим 1 буде генерувати за хвилину від 18 до 35 імпульсів і ця цифра буде щоразу змінюватися випадковим чином.

Відтак, на виході GCemu20_V1 має випадкову кількість випадково розподілених у часі (в межах кожної хвилини роботи) імпульсів.

Продукт GCemu20_V1 має вбудований, готовий до використання програмний код. Щоб почати працювати з емулятором необхідно лише подати живлення через порт micro USB та підключити імпульсний вихід до хост-контролера, який обробляє імпульси та розраховує рівень імітованої радіації.

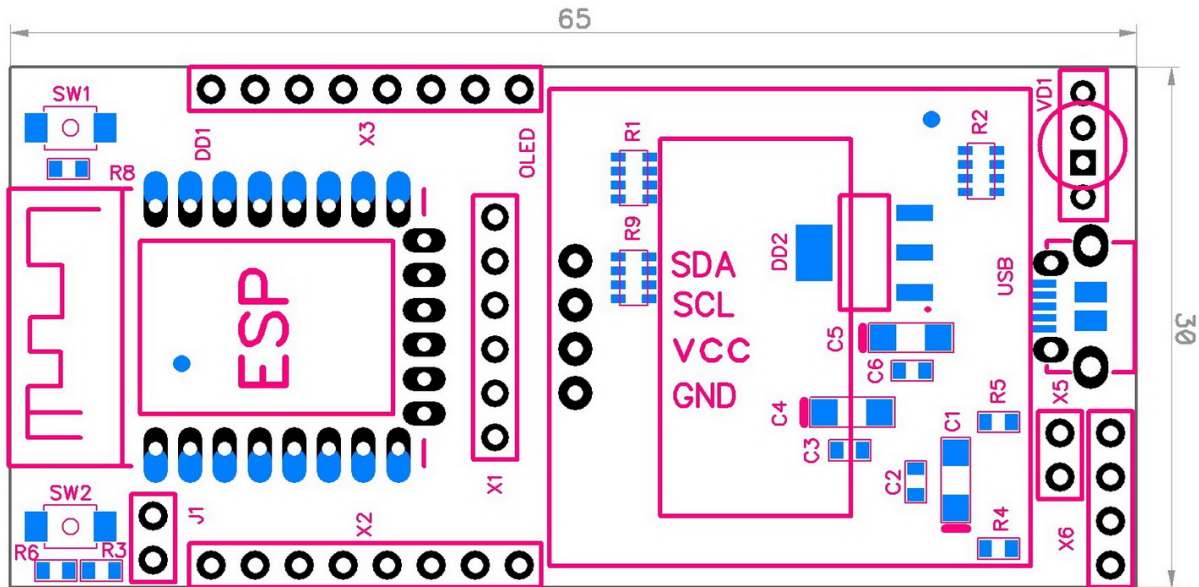
Придбавши цей продукт, користувач не має потреби щось програмувати чи прошивати самостійно, про це вже подбали у IoT-devices LLC.

Але за потреби користувач може прошити модуль іншою прошивкою і використовувати модуль на свій розсуд для інших задач через інтерфейс UART, призначеними для ESP8266 засобами. Наразі відомо, що для модулів на базі ESP8266 існує безліч IoT-платформ, як-от ESP-IDF, Arduino, NodeMCU, MicroPython, ESPHome, Tasmota та багато інших.

Розміри модуля, призначення портів вводу/виводу

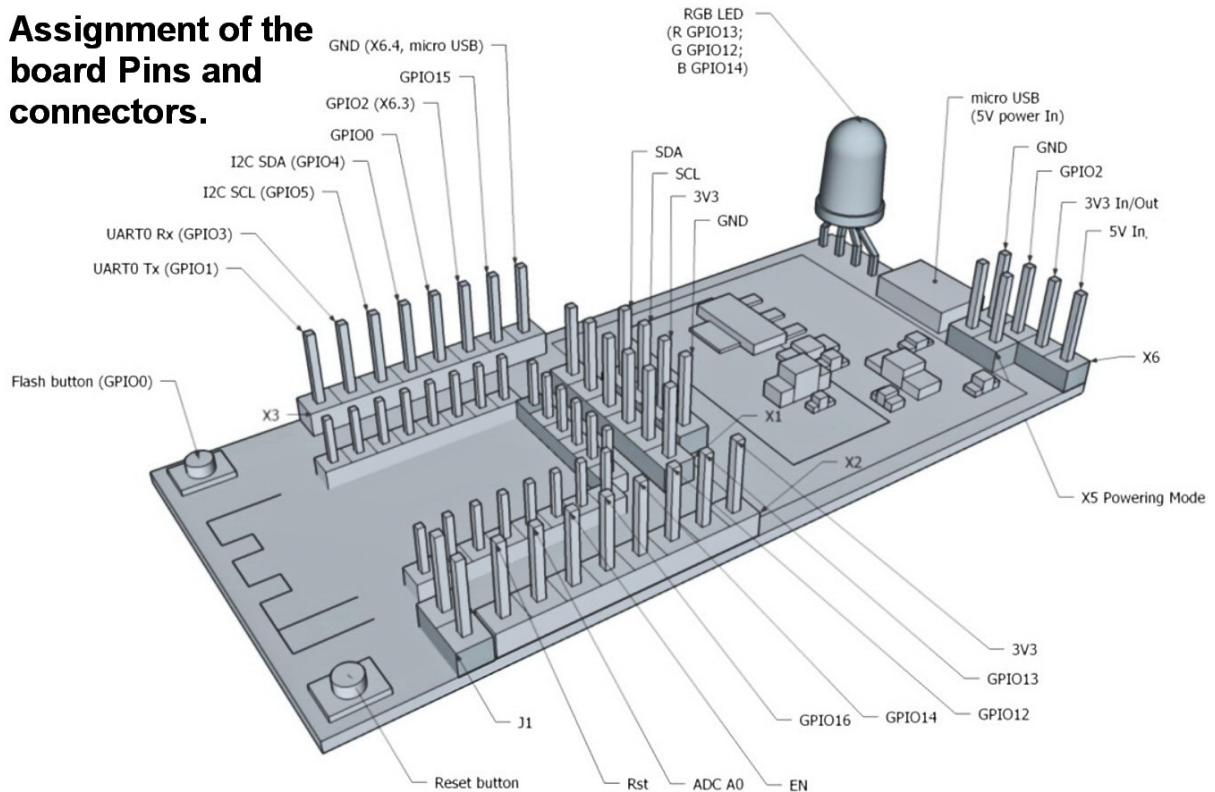
У якості апаратної платформи для GCemu20_V1 застосовується універсальний модуль контролера ESP12.OLED виробництва IoT-devices LLC, у виконанні без дисплея. Тому призначення портів на платі модуля відповідає документації на модуль ESP12.OLED та на контролер ESP8266, вбудований до цього продукту.

Друківана плата модуля ESP12.OLED_V1:



Призначення всіх портів вводу-виводу модуля ESP12.OLED_V1, як апаратної платформи, на якій побудовано продукт емулятора лічильника Гейгера GCemu20_V1:

Assignment of the board Pins and connectors.



Втім, для роботи у ролі емулятора GCemu20_V1 застосовується лише частина наявних на контролері ESP12.OLED_V1 портів вводу-виводу.

Зокрема в емуляторі GCemu20_V1 задіяно наступні порти контролера:

- порт живлення micro USB 5V;
- імпульсний вихід емулятора лічильника Гейгера (GPIO4 / D5, на платі позначено як SDA);
- інтерфейс (опціональний) UART для підключення консолі розробника:
 - UART0 Rx / GPIO3;
 - UART0 Tx / GPIO1;
- джампер X5 вибору режиму живлення (має бути встановлений під час живлення від micro USB);
- перемикач J2 для вибору режиму імпульсного виводу (налаштування користувача для перемикання випадкової кількості / фіксованої кількості імпульсів при поточному режимі емуляції);
- вбудований RGB-світлодіод:
 - GPIO14 / D5 - Blue;
 - GPIO12 / D6 - Green;
 - GPIO13 / D7 - Red;
- кнопка Reset (на платі позначено як SW2);
- кнопка Flash (GPIO0 / D3, на платі позначено як SW1);

На наступному рисунку показано які саме це порти:

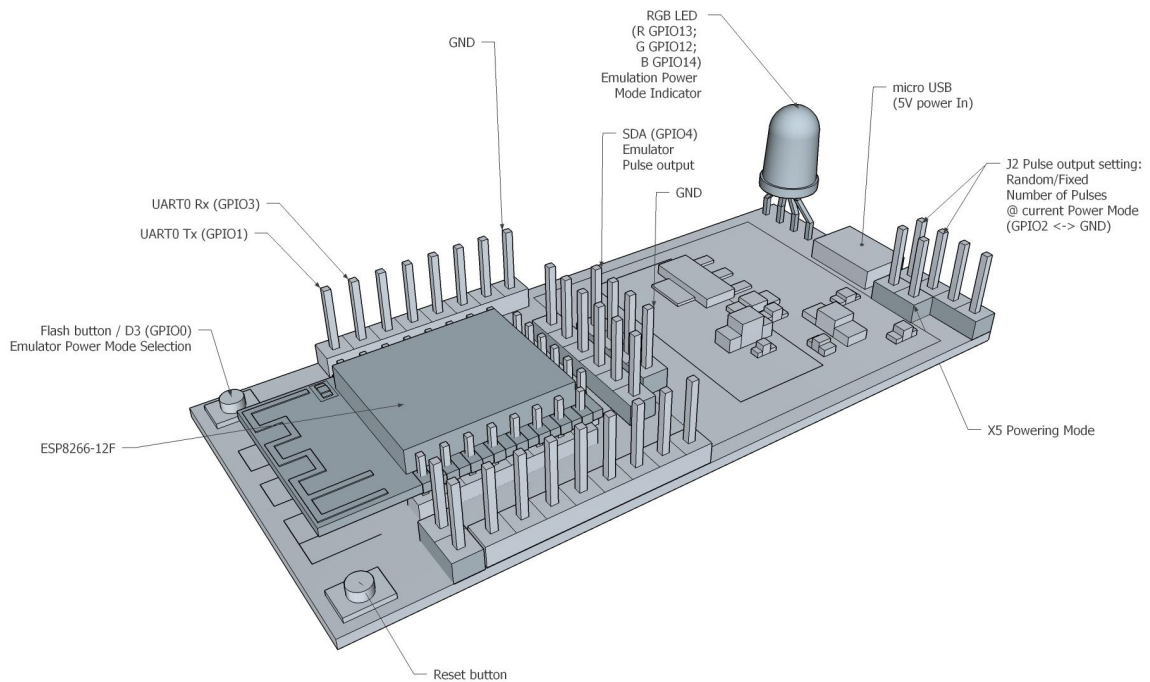
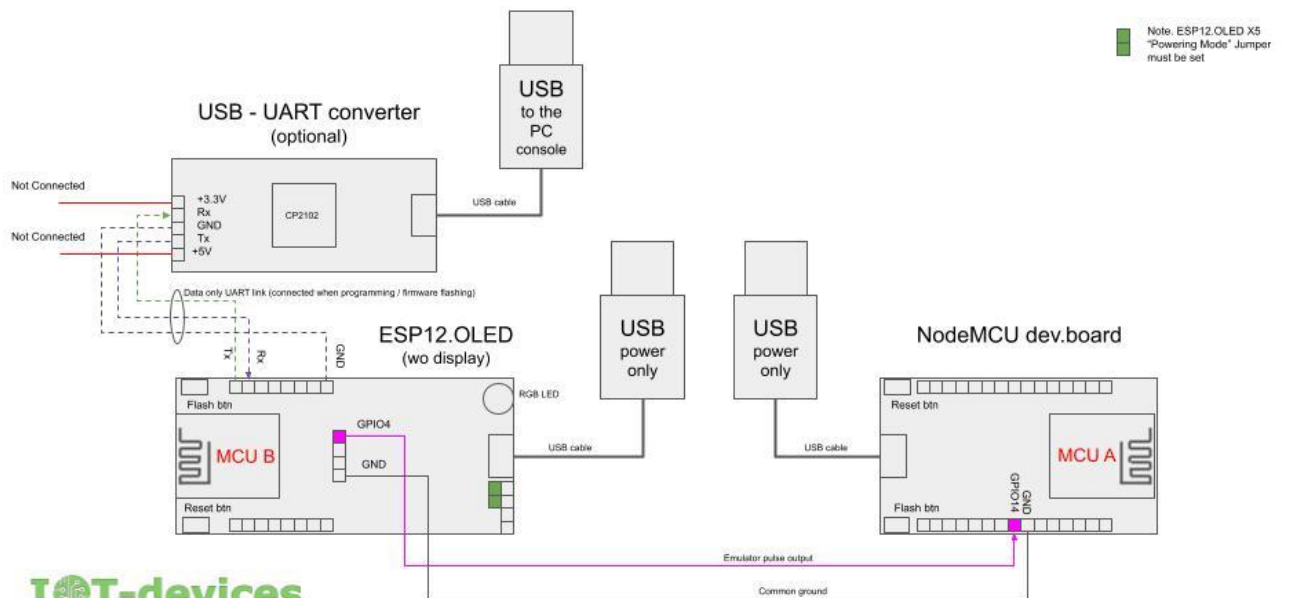


Рис. Порти вводу-виводу ESP12.OLED_V1 (без дисплея), які задіяно у емуляторі GCemu20_V1

Схема підключення модуля емулятора (MCU_B) ESP12.OLED_V1 до головного контролера (MCU_A) NodeMCU може бути така:

DIY Geiger Counter Emulator

GGreg20_V3 emulator using ESP12.OLED module



IoT-devices

Copyright 2022 IoT-devices LLC, Kyiv, Ukraine

File version 1.1; Last updated: 2022-12-09

Також на цій схемі показано опціональне підключення консолі розробника / програматора через перетворювач UART - USB.

У якості довідкового матеріалу за нумерацією портів рекомендуємо наступні матеріали:

- Стандарт планування та застосування пінів розроблений alterstrategy.lab:
<https://alterstrategy.com/recommended-pin-use-standard/>
- Документація на прошивку NodeMCU:
<https://nodemcu.readthedocs.io/en/latest/modules/gpio/>
- Документація на модуль ESP12.OLED_V1 на сайті:
<https://iot-devices.com.ua/en/product/esp12oled-universal-esp8266-mcuboard-oled-en/>
та на Tindie:
<https://www.tindie.com/products/iotdev/esp12oled-universal-esp8266096oled-mcu-board/>

Розміри продукту GCemu20_V1, що побудовано на апаратній платформі ESP12.OLED такі:

- X: 65 мм;
- Y: 30 мм;
- Z: 12 мм.

Для порівняння, справжній модуль лічильника Гейгера GGreg20_V3 з трубкою СБМ-20 має наступні розміри:

X: 126 x Y: 30 x Z: 12 мм.

Порівняння емулятора GCemu20_V1 та справжнього GGreg20_V3

	GCemu20_V1 (емулятор лічильника Гейгера)	GGreg20_V3 (справжній лічильник Гейгера)
Тип трубки Гейгера-Мюллера	немає, не передбачається; імітується програмно	СБМ-20 beta, gamma
Ресурс трубки Гейгера-Мюллера	необмежений	не менше $2 \cdot 10^{10}$ подій за строк служби
Максимальний теоретичний рівень радіації, що детектується	імітує 5 різних режимів. Від 0 до 1.5 $\mu\text{Sv} / \text{h}$	обмежується трубкою СБМ-20 на рівні $315\,780\text{ CPM} \cdot 0.0057 = 1799.95\, \mu\text{Sv} / \text{h}$ для джерела Cs-137
Імітація помилки сенсора	так, Режим 0 - немає імпульсів (0 CPM)	не передбачається
Вивід оперативних даних налагодження у консоль	так, через опціональний інтерфейс UART	не передбачається
Споживання, мА	близько 80 мА (TRNG потребує увімк WiFi) при 5В	18 мА при 5В чи 30 мА при 3.7В від Li Ion

Вмикання та вимірювання

!

Цей модуль лише емулятор, який імітує сенсор радіації і не є справжнім лічильником Гейгера та не детектує реальний рівень радіації.

Якщо вас цікавить дійсний сенсор радіації, рекомендуємо ознайомитися з іншим продуктом - [GGreg20_V3](#), який є справжнім сенсором радіації з імпульсним виходом та трубкою Гейгера-Мюллера СБМ-20 у якості детектора beta- і gamma- випромінювання.

Втім, модуль емулятора GCemu20_V1 є готовим до використання. Перед відправкою модулі GCemu20_V1 проходять програмування і тестування на відповідність до декларованих технічних характеристик. Виконання клієнтом будь-яких операцій з перепрограмування є можливим, але може пошкодити модуль або внести технічні невідповідності у його роботу. Тож користувач виконує будь-які зміни на свій ризик та власним коштом. Модулі які зазнали таких змін не підпадають під дію Політики повернення та/чи заміни.

Включення GCemu20_V1 рекомендується виконувати наступним чином:

Увага. Завжди перед (пере)запуском пристрою знімайте джампер J2 (GPIO2 <-> GND). GCemu20_V1 не буде стартувати, якщо не зняти джампер J2. Така особливість роботи пов'язана з тим, що GPIO2 який задіяно під джампер J2 приймає участь у завантаженні контролера ESP8266 і переводить його у інший режим роботи.

У той же час, джампер нормально працює після завантаження пристрою - його можна знімати чи встановлювати без обмежень.

Крок 1. Підключіть вхідну потужність від джерела живлення.

Крок 2. Увімкніть джерело живлення. Не більше ніж через 10 секунд ви побачите світлові сигнали на RGB-світлодіоді, які імітують потрапляння **вдаваних** частинок у детектор модуля. На імпульсному GPIO-виході модуля будуть з'являтися випадкові логічні одиниці тривалістю 10 мікросекунд кожна, за логікою 3V3 ACTIVE-LOW відповідно до встановленого режиму роботи.

Крок 3. Обрати необхідний режим роботи емулятора кнопкою Flash / D3 (на платі ESP12.OLED_V1 позначено як SW1).

Комплекти постачання продукту

GCemu20_V1 basic

1. Модуль ESP12.OLED (без дисплея) --- 1 шт
2. Готове до використання, вбудоване ПЗ емулятора --- 1 шт
3. Комплект штирів 2.54мм під пайку для самостійного встановлення --- 1 шт

Посилання

Сайт виробника	https://iot-devices.com.ua
Магазин для замовлень	https://iot-devices.com.ua/shop/
Магазин Tindie	https://www.tindie.com/stores/iotdev/
Facebook	https://www.facebook.com/loT-devices-114746816966582
Twitter	https://twitter.com/iotdevicescomua
YouTube	https://www.youtube.com/channel/UCHpPOVVlbdtYtvLUDt1NZw
Email	info@iot-devices.com.ua

Історія внесення змін

2021/05/13 - початковий документ.

2022/12/12 - опис продукту GCemu20_V1 - емулятора лічильника Гейгера.

2023/01/22 - оновлено та доопрацьовано україномовну версію документу

Manufacturer message

Шановний Читач! Дякуємо, що цікавитеся нашими продуктами. Сподіваємося, що вам сподобається і цей наш пристрій. IoT-devices LLC народилася дякуючи підтримці наших Клієнтів та завдяки нашому досвіду і закоханості в Електроніку.

Розроблено дизайн та виготовлено компанією IoT-devices зі свободою і мудрістю в Україні у 2022 році. Всі права застережено.