

# МОДЕЛЬ СИСТЕМИ НА ОСНОВІ ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ ЕКСПЕРИМЕНТАЛЬНИХ ЯДЕРНИХ ДОСЛІДЖЕНЬ

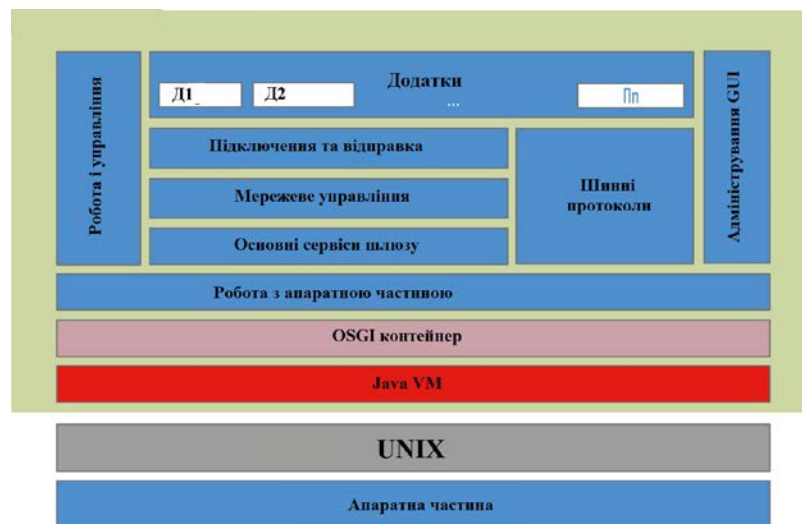
О.М. Семенюк, О. М. Ковальов, О.О. Гаврилов, А.П. Войтер, М.І. Доронін,  
І.О.Мазний

*Інститут ядерних досліджень НАН України, Київ, Україна*

Експериментальні ядерні дослідження здійснюються на технічно складних ядерно-фізичних установках з великою кількістю різноманітних пристроїв і сенсорів, які генерують значні потоки інформації, що має бути своєчасно зібраною, накопиченою та обробленою. Технічною базою таких досліджень є автоматизовані вимірювальні системи з локальною архітектурою.

Близько 10 років тому Інтернет поступово трансформувався з «Інтернету людей» в «Інтернет речей», коли кількість підключених до Інтернету речей (приладів) стало більшим за кількість користувачів Інтернету. На основі інтернету речей було створено Промисловий Інтернет речей (Industrial Internet of Things, IIoT) - систему об'єднаних комп'ютерних мереж і підключених до них промислових (виробничих) об'єктів з вбудованими датчиками і програмним забезпеченням для збору та обміну даними, з можливістю віддаленого контролю і управління в автоматизованому режимі. Є приклади використання Інтернету речей в наукових дослідженнях.

Для напрацювання і оцінки інженерних рішень з використання технологій Інтернету речей в роботі ядерно-фізичних установок інституту та автоматизації експериментальних досліджень створено модель багаторівневої системи, що включає в себе датчики і контролери, встановлені на ядерно-фізичних установках інституту, засоби збору та передачі даних, їх візуалізації, аналітичні інструменти інтерпретації одержуваної інформації та інші компоненти.



Архітектура системи

Архітектура системи базується на різних шарах програмного забезпечення, так що можливо створювати програми для апаратно-абстрагованої платформи, оснащеної всіма, необхідними інструментами розробки і бібліотеками.

Віртуальна машина Java забезпечує апаратну абстракцію на різних апаратних платформах.

Java Native Interface дозволяє коду на Java викликати додатки або бібліотеки, написані на C, C++, і збирати їх.

OSGi Framework забезпечує основу для скорочення часу виходу на ринок і витрат на розробку.

Спеціалізовані бібліотеки інтерфейсів надають розробникам вже протестовані бібліотеки для шин.

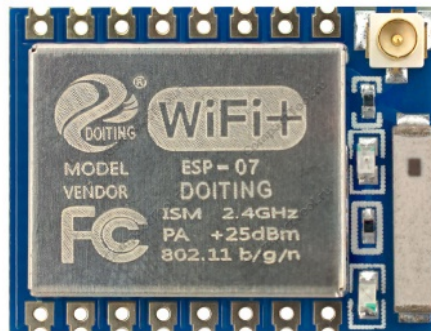
API високого рівня допомагають швидко написати додатки, максимально ефективно використовуючи ці API.

В якості сервера (ОС UNIX), що забезпечує роботу розподіленої системи, використано неліцензійне програмне забезпечення Mosquitto.

Протокол MQTT - Message Queuing Telemetry Transport - протокол для передачі послідовності повідомлень з телеметричними даними, тобто інформації від датчиків до кінцевих користувачів. У червні 2016 р стандарт був визнаний Міжнародною організацією зі стандартизації (ISO). MQTT Version 3.1.1 був зареєстрований технічним комітетом з інформаційних технологій ISO (JTC1) під номером ISO / IEC 20922.

Джерелом даних були модулі мікроконтролера ESP8266, з можливістю виконувати програми з флеш-пам'яті. Контролер підтримує Wi-Fi протоколи 802.11 b / g / n з WEP, WPA, WPA2, обладнаний 14 портами вводу-виводу SPI, I2C, UART, 10-бітним АЦП та має зовнішню пам'ять до 16 МБ.

### Макетні плати на основі модуля ESP-07 ( ESP8266EX)



#### Специфікація

- Напруга живлення: 3,3 В
- Енергоспоживання: 10 мка ... 170 мА
- Флеш-пам'ять: 512 кб
- Процесор: Tensilica L106, 32 біта
- Швидкість процесора: 80 ... 160 МГц
- ОЗУ: 32 кб + 80 кб
- Протоколи WiFi: 802.11 b / g / n
- Діапазон частот: 2.4ГГц-2.5ГГц (2400м-2483.5М)
- Режими WiFi: Клієнт, Програмна точка доступу, Клієнт + Програмна точка доступу (station, softAP, station + softAP)
- Захист: WPA-PSK, WPA2-PSK
- Шифрування: WEP, TKIP, AES
- Мережеві протоколи: IPv4, TCP / UDP / HTTP / FTP
- Висновки загального призначення (вхід / вихід, GPIO): 11
- інтерфейси: UART, HSPI, I2C, I2S, IrDA, GPIO, ШІМ
- ШІМ виходи: 4

- АЦП: 1 введення з роздільною здатністю 1024
- Максимальна кількість підключень ТСП: 5

На макетній платі (Рис.1) знаходиться триколірний світлодіод (виходи GPIO12-GPIO14). Він розміщений на змінній панелі, що дає можливість оперативної заміни на інший пристрій. Таким чином моделюється можливість дистанційного керування включенням / виключенням світлодіодів (інших пристроїв). Дія досить примітивна, що часто зустрічається і є необхідною.

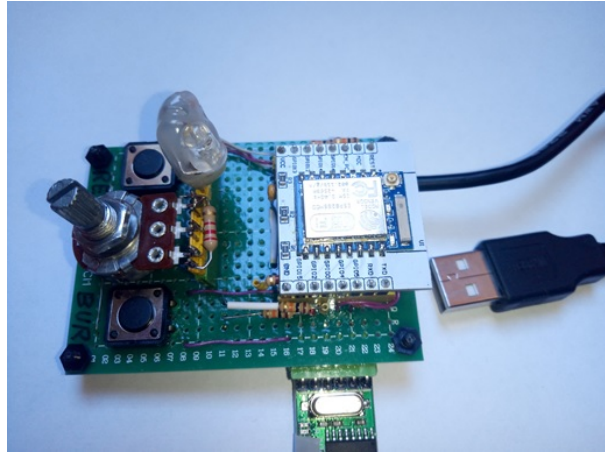


Рис. 1

Більш складна і цікава задача, реалізація ШІМ. Широтно-імпульсна модуляція (ШІМ) - регулювання вихідної напруги здійснюється зміною тривалості імпульсів при постійній частоті проходження. Фізично це виглядає, як зміна яскравості світіння стенового світлодіода. Насправді, при певному схемотехнічному вирішенні, відбуватиметься управління рівнем постійної напруги на виході. Реалізується можливість отримання регульованої напруги керування якимось пристроєм.

Як це реалізується. При імпульсах однієї полярності модулюючий сигнал перетвориться в послідовність однополярних періодично повторюваних широтно-модульованих імпульсів (однотактна ШІМ). Тривалість імпульсів в однотактній ШІМ визначається дискретними значеннями модулюючого сигналу. На рис. 2 приведена діаграма, яка пояснює дію однотактної ШІМ.

Розіб'ємо вісь часу на рівні проміжки, іменовані *періодом*  $T$ . Тепер, наприклад, половину періоду ми будемо замикати ключ  $K$ . Коли ключ замкнутий, до навантаження  $H$  подається напруга від джерела живлення  $U_{ип}$ . Другу частину напівперіоду ключа знаходиться в закритому стані. А споживач залишиться без живлення.

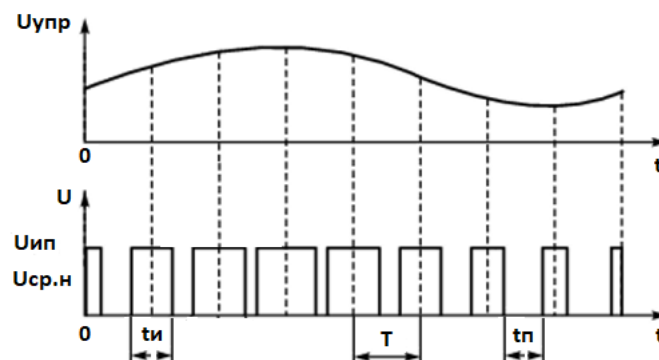


Рис.2

Час, протягом якого ключ замкнутий, називається **часом імпульсу  $t_i$** . А час тривалості розімкнутого ключа називають **часом паузи  $t_n$** . Якщо виміряти напругу на навантаженні, то воно буде дорівнює половині  $U_{un}$ .

Середнє значення напруги на навантаженні можна виразити наступною залежністю:  $U_{ср.н} = U_{unp} = U_{un} t_i / T$ .

Відношення часу імпульсу  $t_i$  до періоду  $T$  називають **коефіцієнтом заповнення  $D$** . А величина, зворотна йому називається **скважністю**:  $S = 1 / D = T / t_i$ .

На практиці зручніше користуватися коефіцієнтом заповнення, який часто висловлюють у відсотках. Коли транзистор повністю відкритий протягом усього часу, то коефіцієнт заповнення  $D$  дорівнює одиниці або 100%.

Якщо  $D = 50\%$ , то це означає, що половину часу за період транзистор знаходиться у відкритому стані, а половину в закритому. В такому випадку форма сигналу називається **меандр**.

Отже, змінюючи коефіцієнт  $D$  від 0 до одиниці або до 100% можна змінювати величину  $U_{unp}$  від 0 до  $U_{un}$ :  $U_{unp} = U_{un} * D$ .

Також на макетній платі (рис.3) реалізована регульована схема подачі постійної напруги (вивід ADC). Потенціометр, з лінійною характеристикою, що знаходиться на змінній панелі забезпечує подачу постійної напруги на вхід АЦП в діапазоні 0 - 1 В. Таке технічне рішення дозволяє використовувати будь-який інший пристрій, що подає постійну напругу в зазначеному діапазоні. ESP8266 має тільки один канал аналого-цифрового перетворювача (ADC0). АЦП 10-бітний, однак, розробник зазначає, що є можливість підвищити розрядність до 16 біт.

При зчитуванні значення з АЦП отримуємо число в діапазоні від 0 до 1023. Якщо отримали 1024 - це означає, що напруга на вході ESP8266 перевищила допустимий максимум (1 Вольт). ADC може працювати в одному з двох режимів. Режим **adc.INIT\_ADC** - вимірює напругу на вході **ADC0**. Режим **adc.INIT\_VDD** - вимірює системну напругу (system voltage).

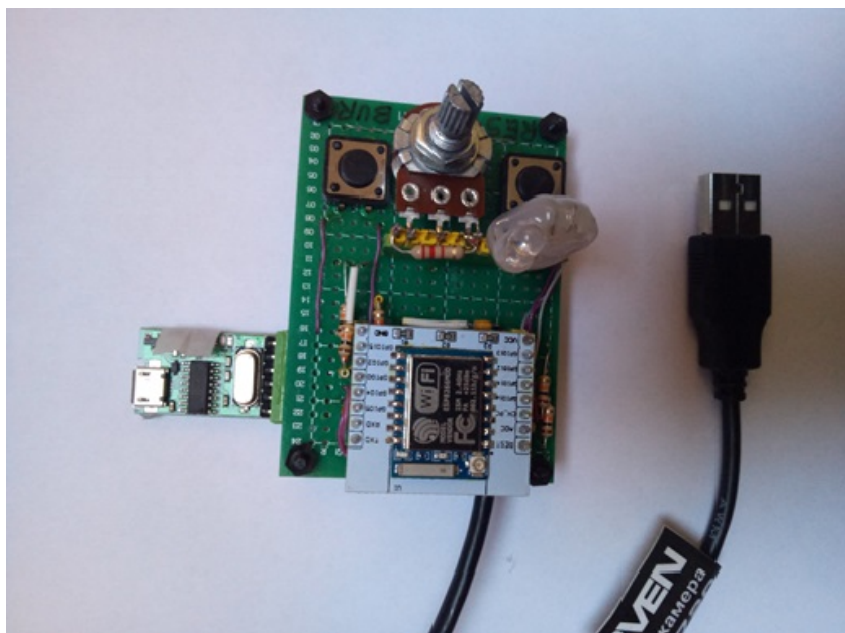


Рис. 3

Модуль можна запрограмувати за допомогою конвертера USB-TTL через контакти для послідовної комунікації (RX і TX). Для програмування ESP8266 можна використовувати не

тільки конвертер USB-TTL, але і плату Arduino. На макетній платі Рис.3 реалізований перший спосіб програмування.

Плати ESP8266 стали дуже популярні через свої якісні характеристики і сумісність з Arduino IDE. ESP вимагає тільки підключення живлення, а збір і відправку даних вона може здійснювати по Wi-Fi. Одним з популярних відкритих протоколів зв'язку для ESP8266 є MQTT.

MQTT або Message Queue Telemetry Transport - це легкий, компактний і відкритий протокол обміну даними створений для передачі даних на віддалених локаціях, де потрібно невеликий розмір коду і є обмеження по пропускну́й здатності каналу. Перераховані вище якості дозволяють застосовувати його в системах M2M (Машинно-Машинна взаємодія) і IoT (Промисловий Інтернет речей). MQTT протокол обміну повідомленнями за шаблоном видавець-підписник (pub / sub). MQTT вимагає наявності сервера - брокера, який об'єднує під'єднані пристрої у мережу. Самі ж пристрої при підключенні підписуються на топіки, в яких надсилаються дані. Брокер відсилає тільки ті топіки пристрою, які йому потрібні.

MQTT є протоколом для потокової передачі даних між пристроями з обмеженою потужністю CPU і / або часом автономної роботи, а також для мереж з дорогою або низькою пропускну́ю здатністю, непередбачуваною стабільністю або високою затримкою. Саме тому MQTT відомий як ідеальний транспорт для IoT. Він побудований на протоколі TCP / IP, але є відгалуження MQTT-SN для роботи по Bluetooth, UDP, ZigBee і в інших мережах IoT, відмінних від TCP / IP.

Система зв'язку, побудована на MQTT, складається з сервера-видавця, сервера-брокера і одного або декількох клієнтів (Рис. 4). Видавець не вимагає яких-небудь налаштувань за кількістю або розташуванню підписників, які отримують повідомлення. Крім того, підписникам не потрібно настрійка на конкретного видавця. В системі може бути кілька брокерів, що поширюють повідомлення.

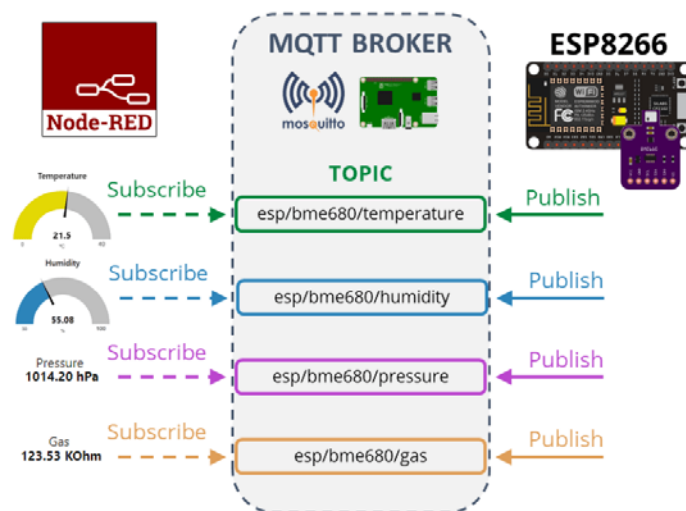


Рис.4

Найбільш відомим сервером-брокером є Eclipse Mosquitto (Рис. 4) - це IoT-платформа, що представляє собою полегшену серверну реалізацію протоколу MQTT для інтернету речей, яка підходить для всіх областей застосування від забезпечених енергією стаціонарних підключених активів до вбудованих і малопотужних пристроїв. Mosquitto кроссплатформний брокер, що дає можливість об'єднати і контролювати роботу безлічі датчиків, підключених через плати ESP8266 як до локальної так і глобальної мережі Internet (після переадресації портів) з будь-якого пристрою - ПК, мобільного пристрою.