МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. І. СІКОРСЬКОГО»

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ

КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ АПАРАТУРИ

Звіт з розрахункової роботи з дисципліни

«Електронно-обчислювальні пристрої та системи»

На тему:

«Моніторинг серверу для збирання програм»

Виконав:

студент гр. ДК-42

Тяпко М.П.

Перевірив:

Ходнєв Т.А.

Київ – 2017

**Зміст**

Вступ 2

Постановка завдання 3

Вирішення поставленого завдання 4

Висновок 5

Література 6

**Вступ**

Сучасне покоління комп’ютерів не ставить такі суворі обмеження на кількість задіяних програмою ресурсів, як це було раніше. А отже програмісти мають змогу розробляти великі програми, а великі програми з гарною архітектурою, частіше за все, розбиті на велику кількість файлів к десятками тисяч рядків код. Навіть на сучасних комп’ютерах процес збирання програми з її вихідного коду може тривати достатньо тривалий час, і забирати при цьому всі ресурси комп’ютера, що робить некомфортним його використання для людини в цей час.

Для вирішення цієї проблеми виділяють окремий комп’ютер, задача якого збирати програми з вихідного коду. Однак готового рішення, як контролювати чи зібралася програма успішно, чи виникла якась помилка, немає, а отже всім треба витрачати час на вирішення і цієї проблеми також.

**Постановка завдання**

Завданням являється розробка пристрою та його програмного забезпечення для індикації результату збирання програми на спеціальному комп’ютері. В силу того, що цей комп’ютер може знаходитися віддалено, то пристрій повинен мати змогу під’єднуватися до нього за допомогою мережі інтернет бездротовим шляхом. Інтернет зараз дуже поширений, і до нього є доступ майже усюди.

**Вирішення поставленого завдання**

Рішення має буди досить універсальним, щоб його легко можна було переконфігурувати під конкретну задачу. Для цього прилад буде під’єднуватися до мережі інтернет бездротовим шляхом, тому його можна буде розташовувати віддалено від комп’ютера для збирання програм. Передбачається, що комп’ютер також буде під’єднано до мережі інтернет, і його буде настроєно для відповіді на запит про результат збирання програми. Формат обміну даними між пристроєм та комп’ютером обрано також дуже розповсюджений, це HTTP GET[1] запит, а відповідь буде у форматі JSON[2].

На рис. 1 схематично зображено підключення всіх компонентів.

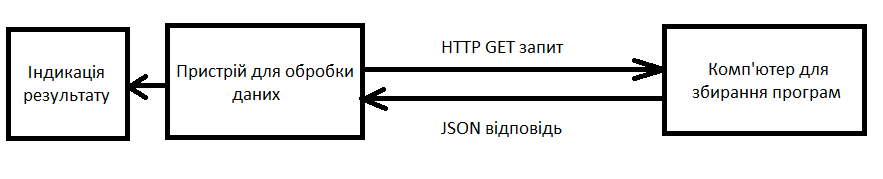


Рис. 1. Схема підключення

За основу пристрою було обрано мікроконтролер «chipKIT Wi-Fire PIC32MZ» від компанії «Digilent»[3]. Вся інформація виводиться на екран модуля «chipKIT Basic I/O Shield»[4] від тієї ж компанії. Для підключення до мережі інтернет та подальшого її використання було обрано бібліотеку «Digilent Embedded IP Stack»[5]. Обробка даних отриманих у форматі JSON відбувається за допомогою бібліотеки «Arduino JSON»[6].

Для перевірки працездатності пристрою було просимульовано роботу комп’ютера за допомогою сервісу «Travis CI»[7]. Він являється безкоштовним сервісом, що збирає програми з коду, який знаходиться на онлайн майданчику для публікації вихідного код програм «GitHub»[8]. Також цей сервіс дозволяє надсилати йому запити про стан збирання програм, і надсилає відповідь у необхідному для створеного пристрою форматі.

Увесь код прошивки доступний за посиланням[9].

**Висновок**

Під час виконання роботи було розроблено прототип IoT пристрою, що здійснює моніторинг результату збирання програм на віддаленому комп’ютері через мережу інтернет, шляхом відсилання HTTP GET запиту та отримання відповіді у форматі JSON. Індикація відбувається шляхом виведення всієї інформації на підключений до пристрою дисплей.

**Література**

1. HTTP GET запит, режим доступу:

https://tools.ietf.org/html/rfc7231#section-4.3.1

1. Формат даних JSON, режим доступу:

https://www.json.org/

1. Плата «chipKIT Wi-Fire PIC32MZ» від компанії «Digilent» , режим доступу:

http://chipkit.net/wpcproduct/chipkit-wi-fire/

1. Модуль «chipKIT Basic I/O Shield» від компанії «Digilent» , режим доступу:

http://chipkit.net/wpcproduct/chipkit-basic-io-shield/

1. Бібліотека «Digilent Embedded IP Stack» , режим доступу:

https://github.com/sergev/deIPcK

1. Бібліотека «Arduino JSON» , режим доступу:

https://github.com/bblanchon/ArduinoJson

1. Сервіс «Travis CI», режим доступу:

https://travis-ci.org/

1. Онлайн майданчик публікації вихідних кодів «GitHub», режим доступу:

https://github.com

1. Опублікований вихідний код прошивки пристрою, режим доступу:

https://github.com/kpi-keoa/TheConnectedMCU\_Labs/tree/master/mtiapko/Build\_Server\_Monitor