UDC 004.35 004.78 004.454

MURAVIOV I.,
TARANIUK V.,
KLYMENKO I.

MAKING AN IOT DEVELOPMENT PLATFORM FROM A SIMPLE MICROCONTROLLER DEMONSTRATION BOARD

The IOT is now in trend. Because of its huge popularity and business interest to it, IOT is only now coming massively to universities worldwide as a separate study. This article provides an IOT solution based on embedded technologies that were not specifically designed for IOT. The core of this solution - board is developed for studying peripherals of a particular general-purpose MCU. But we successfully adapted it to use in IOT and for study of IOT systems.

Keywords: Internet of things, wireless connectivity, Wi-Fi, embedded platform, AT commands.

1 Introduction

Thing in IOT (Internet of things) is a device, that interacts with outer world using physical processes and connectivity with other devices. In other words, thing is an embedded system, connected to the Internet and has a unique way of identification and can be addressed, IOT gives an opportunity to have two forms of communication: human-thing and thing-thing (machine to machine) [1]. Development of IOT today is development of such things, specific hardware architectures those include wireless connectivity, development of connection protocols used for interaction between such things, development of back-end, cloud systems for managing things remotely, and front-end for connecting things to humans. Things can control, as examples of such things can be any remote sensors: gas sensor, remote measuring instruments, such as thermometer, barometer. The examples of things those can be controlled are switchers, relays, lamps.

Of course, the connection between two things can be of any nature, but if a particular thing is declared by manufacturer as part of IOT, it is usually supposed that it supports wireless connectivity. Wireless connectivity is much more flexible, than wired and usually cheaper. But it was confirmed, that huge infrastructures with things, connected to their hubs, those are connected to clouds, clients are efficient if constructed as mutually complementary network systems between wireless and wired communications [2]. The wireless connectivity used in IOT is divided to 2 types of communication protocols: short range protocols and low power wide area networks [3].

Embedded world is huge of different manufacturers of embedded systems, such as single board computers, MCUs (microcontroller units). A typical embedded developer is given with such set, is studying their production, learns programming for this specific production, for example special purpose wireless microcontroller, then develops application – solution to a given problem. The IOT is a little bit different to embedded. There is less documentation available to the hardware and it is more closed. Part of information is usually for security purpose or because of commercial considerations. Sometimes only a communication protocol with it is provided, accordingly, this embedded system is then called a black box, into which signals can be sent and response signals at the output can be received [4]. In this article given solution uses two distinctive parts: embedded system, which is provided with full available documentation how it works, which is connected to another embedded system, which is used as black box, but it gives an ability to connect solution to the IOT. The solution used in this article is built with most popular technologies and platforms. It also shows that it is not so difficult to connect already developed embedded solution to the internet even if it wasn’t intended for this initially.

2 Related work

Today we have many of different development boards from different manufacturers those push their production to the market. Usually each development kit comes with demonstration solution, code base, documentation for very specific solution. There are numbers of examples of embedded systems with host controller and slave device, for example wireless connectivity module with single board computer [5] or MCU [6] as host. There exist some IOT development boards, based on general-purpose microcontrollers as hosts [7].

Any MCU with enough resources can be used as host that controls wireless connectivity module. The most popular choise of general–purpose MCUs those work well as hosts are ARM (Ashton Raggatt McDougall) architecture industry-standard core based STM32 MCU family [6] [8] [9].

Because of its popularity this family supports wide variety of RTOSes(real-time operating systems), a particular one is open source and one of most popular on different architecuters. It is FreeRTOS, that is used in variety of different MCU projects, there exists port of FreeRTOS, that supports Amazon cloud services, it is Amazon FreeRTOS, that is also used in a variety of IOT projects [9]. In this article no cloud services were used, so the mainstream FreeRTOS was used as host MCU RTOS.

3 The purpose and objectives of the study

The purpose of this study is to find out a way to simplify the connection of an arbitrary embedded system to IOT. This article shows one of ways of studying the world of IOT using just one MCU based embedded system and how to make their conception and code more portable.

The developed software solution can be easily ported to any other MCU-based sensor.

There exist devices with sensors and display inside. The ported solution to any particular of such devices can give remote web interface to this particular device and remote control of its sensors and display via web interface via local network.

The solution may be used for remote notification boards with displays, for example in railway stations for train arrival notifications via web interface.

4 Materials and methods of research

A Global Logic starter kit was used as a device with embedded symbolic display and sensors, such as temperature sensor, accelerometer etc. It is connected with STM32F407-DISC1 Discovery board with STM32F407VGT6 microcontroller inside. This was an embedded solution with no software and hardware examples provided for connecting it to the IOT.

In this solution we used Wi-Fi networking technology, as it can be used in applications with varying levels of power consumption and signal range and now is one of the most popular [10] networking technologies in IOT.

To implement Wi-Fi connectivity, one of the most popular transceivers was used, it is ESP8266 wireless module [11]. we used its factory firmware, that supports opening TCP/IP (Transmission Control Protocol/Internet Protocol) connection.

With the standard firmware ESP8266 and ESP32 modules communicate via AT (Attention) commands. AT commands are one of simple interfaces used for communication with host and slave devices. ESP8266 uses UART (Universal Asynchronous Receiver Transmitter) to communicate with its host. Any MCU with UART support can interface ESP8266 connection [12]. This command set is also known as Hayes command set, that was originally developed for smart modems and today is still used in various modems [4] [13]. AT command set was chosen to communicate with ESP8266 because it is easy to parse expressions of this language and generate them.

The code base for solution was written in the C programming language that is de-facto standard for programming embedded systems.

5 Design hardware of IOT solution

So, we take a particular device with microcontroller or microcomputer inside and place support for AT commands for a particular module. So, the Global Logic Starter Kit, based on STM32F407-DISC1 Discovery [14] board was used as sensors board with the main MCU which was connected by the UART interface to ESP8266 Wi-Fi module.

The MCU (microcontroller unit) - STM32F407VGT6 is connected with ESP8266 module via UART interface. Schematic for connection between starter kit and ESP8266 you can see in figure 1. The MCU with its initialized pins PD8, PD9 for UART number 3 is connected to default UART pins of ESP8266. One wire of UART bus is for transfer data from MCU to slave and another for reverse direction. MCU can send AT commands as string of characters via UART and receive responses via interrupts from UART.

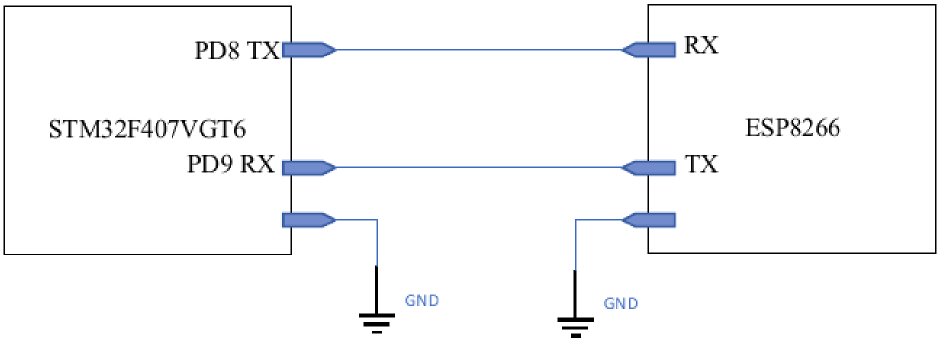


Figure 1. The connection diagram between MCU and ESP8266

6 Design software of the IOT solution

The task, that interfaces host and slave device is made as FreeRTOS thread. This thread uses the library that can generate AT commands, send them, parse responses from ESP8266, that was also written by us. After startup main task initializes ESP8266 using AT command set, connects to the Wi-Fi given in settings of firmware and starts server mode. After that it waits for connection from client. If connected client is a web browser, it sends HTTP requests, in particular to receive the web page, they are received by the interrupt handler of MCU from UART of ESP8266 and given via a message passing interface from interrupt to the ESP8266 interfacing thread in MCU. This thread gets identifier of connected client and sends it a generated web page. The library that has an ability to statically generate HTML (hypertext markup language) web pages on MCU with given parameters and settings in header files is also written by us. After generating it places web page as character string in statically defined RAM (random access memory) buffer. After that thread sends this string to ESP8266. The web browser can request to edit input form if it is provided in web page and user added or modified data in this form or pressed a button given in this page. It sends HTTP request with identification of that form and the value result. We added the ability to retrieve this result and save the value to the variable associated with the input form.

The FreeRTOS port for STM32 is built upon HAL (hardware abstraction layer) and CMSIS (cortex microcontroller software interface standard). These layers are also used for creating drivers for STM32F4 Discovery board and Global Logic Starter Kit peripherals. Drivers for symbolic display, ADC (analog to digital converter) thermometer, accelerometer were created and connected with the built web server via its configuration. The given schematic for software solution is shown on figure 2. Drivers collect data to a given structure, which contains also state machine of the web server. Then data is copied to a formed web page – text on RAM buffer. And another task configures ESP8266 for initializing server and connecting to the internet. If ESP8266 is successfully connected, task, handling HTTP and any other requests from ESP8266, may receive connection from the other device. When it comes true, connection is established, and the web page is updated by new values and transferred to the client.

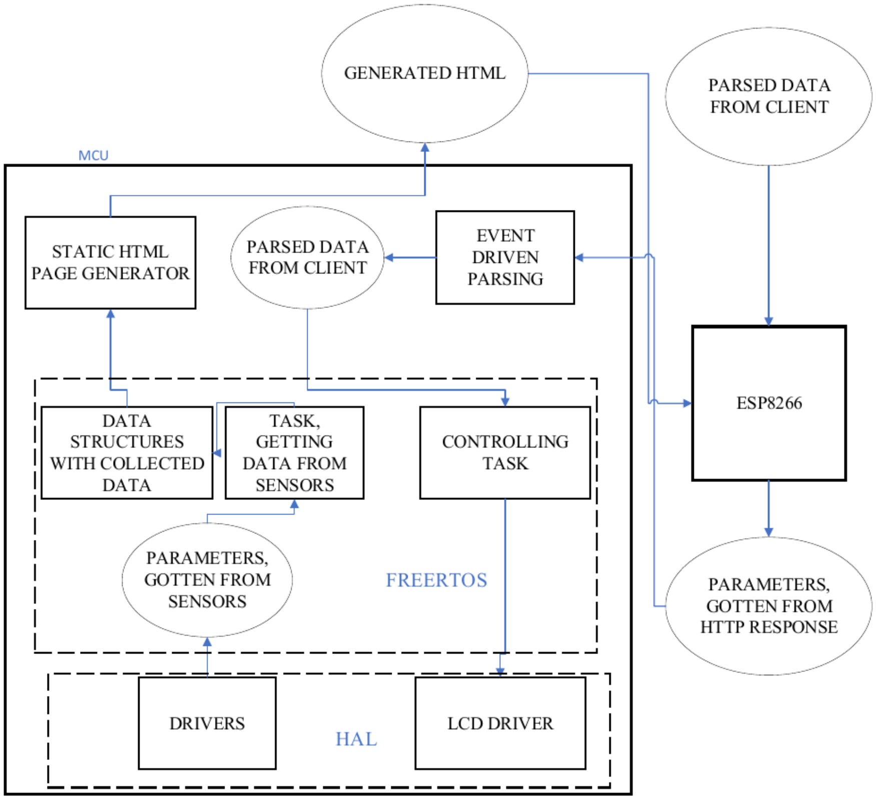

GENERATED HTML
PARSED DATA FROM CLIENT
MCU
STATIC HTML PAGE GENERATOR
PARSED DATA FROM CLIENT
EVENT DRIVEN PARSING
ESP8266
DATA STRUCTURES WITH COLLECTED DATA
TASK, GETTING DATA FROM SENSORS
CONTROLLING TASK
PARAMETERS, GOTTEN FROM SENSORS
FREERTOS
PARAMETERS, GOTTEN FROM HTTP RESPONSE
DRIVERS
HAL
LCD DRIVER

Figure 2. Schematic for software solution

7 Research results and future work

Research and experiments with given embedded system showed that it is possible to study IOT with standard embedded systems and general-purpose development boards. The platform for IOT creating was developed with software library for building firmware. The software solution can be easily ported to another platform.

The structure of connection of developed solution is provided in figure 3.

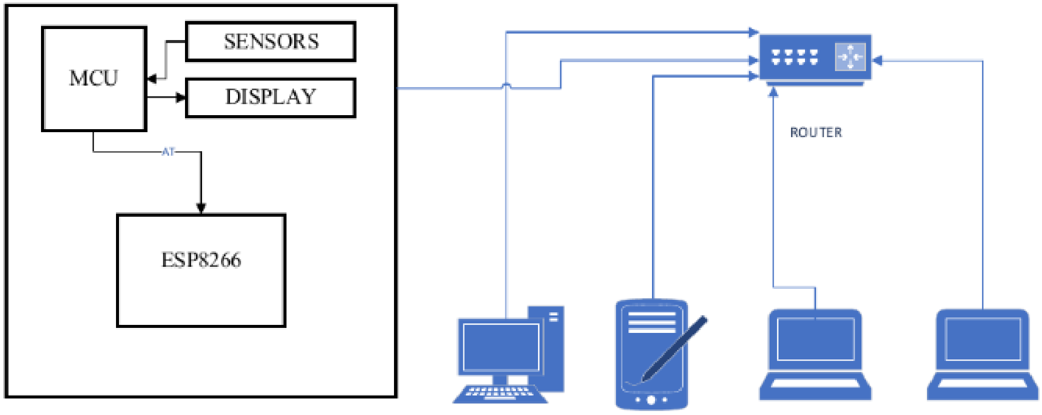


Figure 3. The structure of connection of developed solution

This solution does not have a cloud, it is local system. And the host MCU is server because it handles HTTP requests from any other device in local network.

The MCU is host and it sends AT commands, waits for answers and for other notifications from ESP8266. MCU via ESP8266 is automatically connected to the working network with specified SSID (Service Set Identifier) and password in the settings applied in firmware, and becomes a web server. Now other hosts from network can connect to this server and get statically created web page with real-time measures from sensors. The photo of working solution is provided in figure 4.

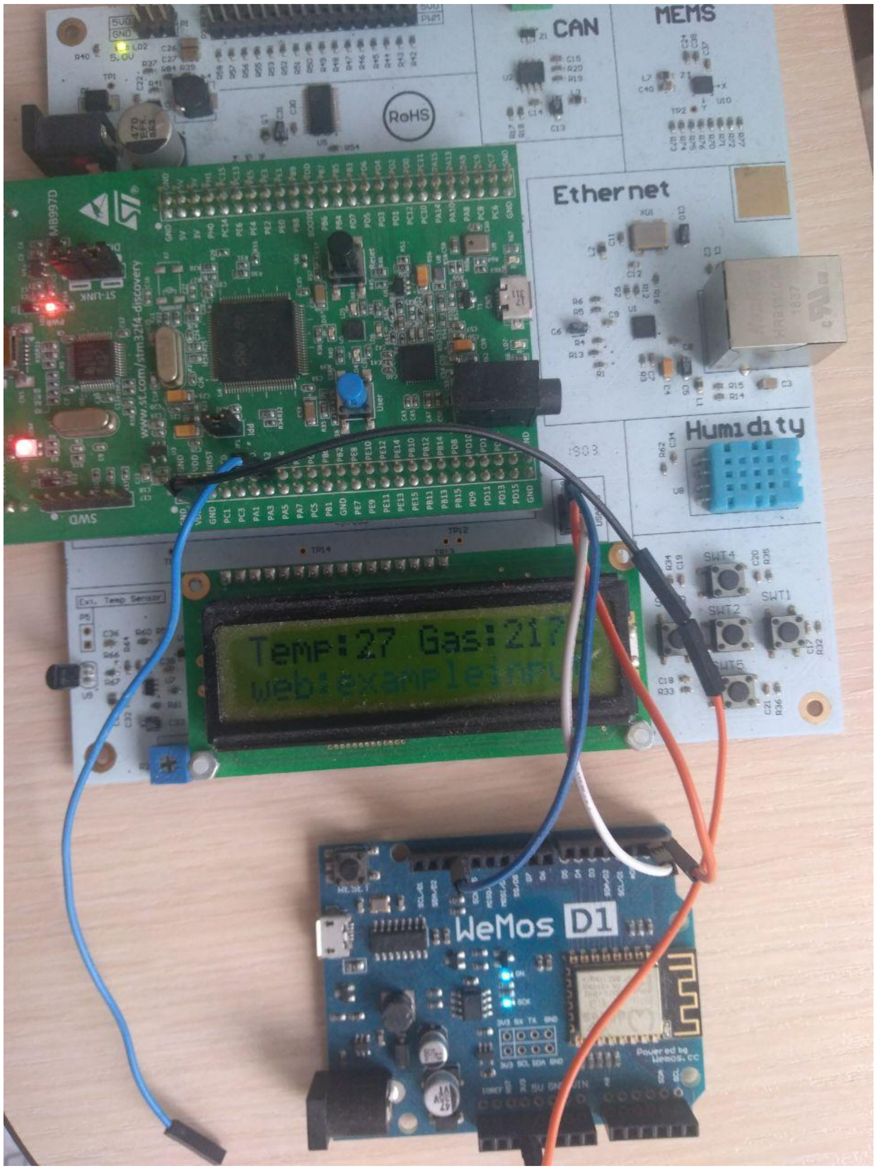
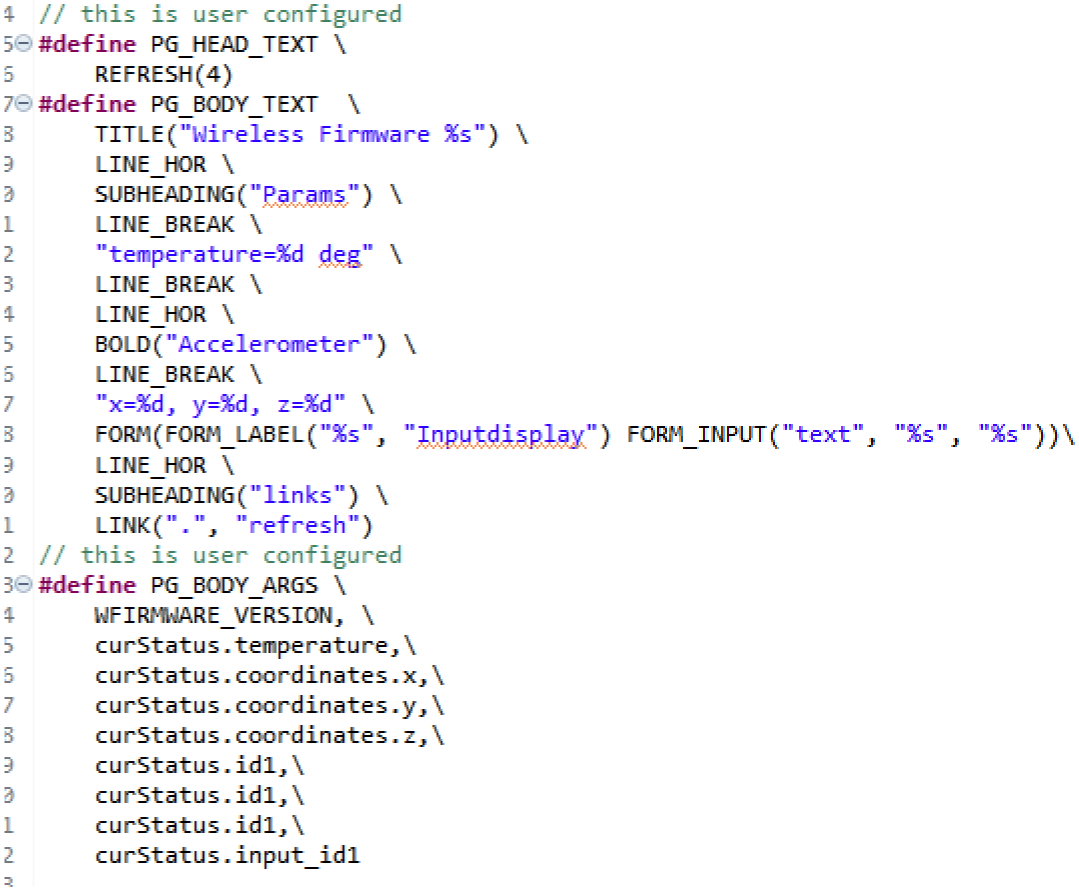


Figure 4. photo of the working solution

The web user interface of embedded system is developed in firmware of MCU. Any mobile or desktop device with the web browser can be connected to the solution provided and get the web page. The software solution gives an ability to give a web interface for sensors and displays. Information from peripherals drivers can be configured to be output to web page, the input field on web page can be configured to be output to display of embedded system.

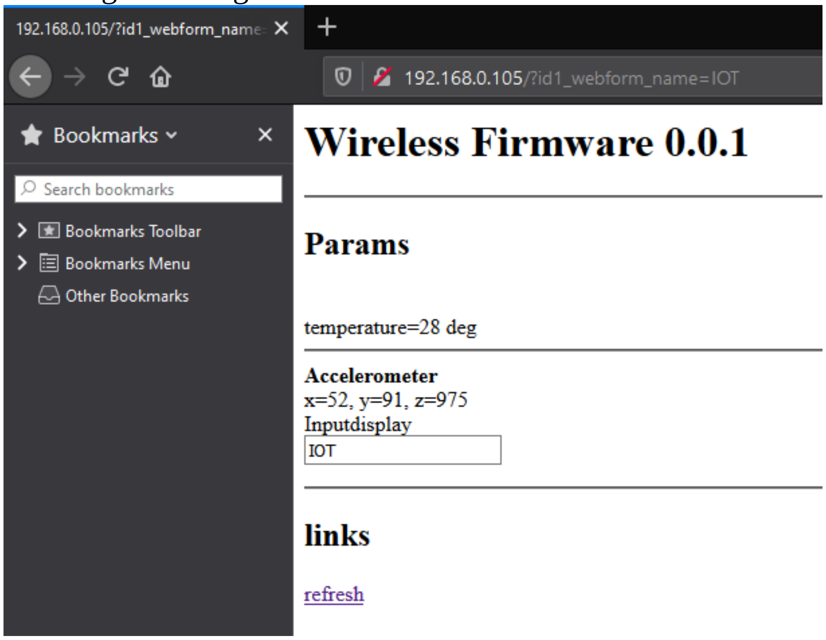
The example of statically generating web page configuration is given in figure 5.



```
// this is user configured
#define PG_HEAD_TEXT \
    REFRESH(4)
#define PG_BODY_TEXT  \
    TITLE("Wireless Firmware %s") \
    LINE_HOR \
    SUBHEADING("Params") \
    LINE_BREAK \
    "temperature=%d deg" \
    LINE_BREAK \
    LINE_HOR \
    BOLD("Accelerometer") \
    LINE_BREAK \
    "x=%d, y=%d, z=%d" \
    FORM(FORM_LABEL("%s", "Inputdisplay") FORM_INPUT("text", "%s", "%s"))\
    LINE_HOR \
    SUBHEADING("links") \
    LINK(".", "refresh")
// this is user configured
#define PG_BODY_ARGS \
    WFIRMWARE_VERSION, \
    curStatus.temperature,\
    curStatus.coordinates.x,\
    curStatus.coordinates.y,\
    curStatus.coordinates.z,\
    curStatus.id1,\
    curStatus.id1,\
    curStatus.id1,\
    curStatus.input_id1
```

Figure 5. Structure of web page

And how it looks like is given in figure 6.



Wireless Firmware 0.0.1

Params

temperature=28 deg

Accelerometer
x=52, y=91, z=975
Inputdisplay
IOT

links

refresh

Figure 6. Web page

The page is configured to refresh automatically in 4 seconds, it gives temperature, accelerometer axis output in real time. And it is possible to input text to display using "Inputdisplay" form, the result of input text "IOT" is given in figure 7.

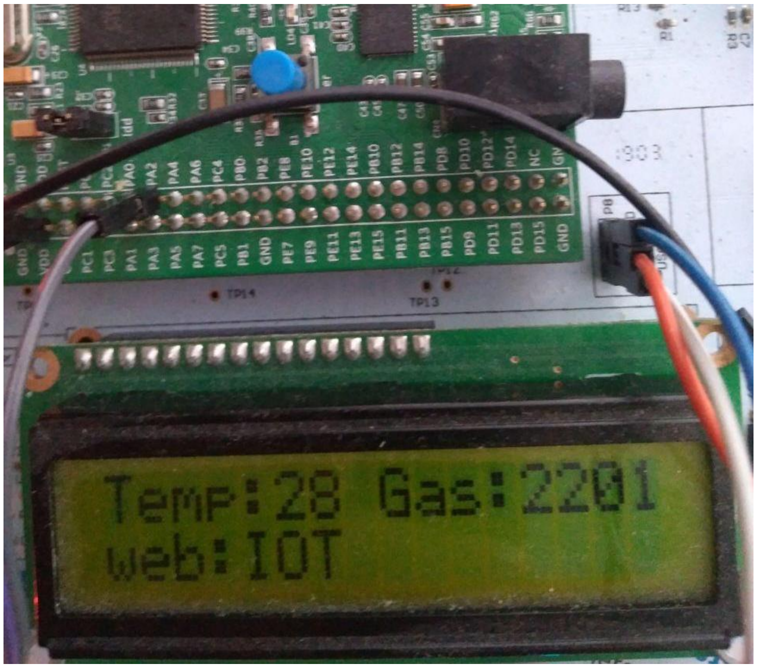


Figure 1. The result of inputted text to web page "IOT"

This software solution makes a web server from the given development board for local networks. IOT without cloud technologies is used with small number of interconnected devices, smart home solutions, those usually come with cloud technologies, are the opposite to this software solution. Also, cloud can benefit from IOT by increasing its scope to deal with existent world things in a more distributed and lively manner, and for bringing new services in a large number for real life scenario [15]. But the idea of local cloudless web server is alive because it has its own benefits, like it is cheaper and more reliable, and standalone solution.

The future improvement of solution assumes code portability and flexibility improvements and support of another MCU families, support of other wireless modules those support another networking technologies.

6 Conclusions

Existing special-purpose MCU-based IOT solutions are restricted to their specific hardware architecture and usually can't be expanded to another use cases using only their own peripherals. Unlike them embedded systems have much more use cases because of their more universal architectures. The ESP8266 has a little number of peripherals itself.

The constructed solution is Wi-Fi supported board, that has a huge number of peripherals, so it has huge amount of capabilities, those are unavailable on most of other IOT boards. All these peripherals can be controlled by web interface provided. This solution can be used for studying IOT and for developing any particular IOT device for a particular use case. We expanded ESP8266s capabilities using general-purpose MCU board as host computer, that can control and can be controlled via ESP8266 from external devices. For this purpose, we used ESP8266s factory firmware with AT commands, that is usually used for verifying correct functionality of ESP8266.

The embedded board for studying is adapted for use in IOT. The adaptation process showed that it is possible and not so difficult to use general-purpose microcontrollers for IOT and it is possible to modify existing embedded platforms and embedded system to use in IOT and to study IOT. It also showed that the embedded world and IOT world are interconnected. Any embedded platform can be used to study IOT.

The software part of solution shows that it is possible using standard interfaces, such as AT commands, to create portable software solutions and make them work in other platforms with minimal changes of code.

The use of RTOS showed that it makes code much better structured than without it. It makes code more flexible and gives ability to divide software solution into different tasks and modules.

The web server creation and creation of web page using general-purpose MCU has showed that it is possible to use general-purpose MCUs to construct modern user interface inside thing and it is not necessary to have a cloud for a small IOT system that can provide user interface itself. It is not necessary to have a cloud to study IOT. And if thing provides a user interface itself, such IOT can`t be scalable enough but it fits well local usage, when only one thing or a few ones needed and is much cheaper, is less resource intensive and is much easier to set up.

The creation of statically generating web page library showed, that user interface for thing can be highly configurable and output or input any information provided.

The creation of hardware solution that can be controlled because it has display and web interface to fill it, and can control, because it has sensors and interfaces to collect data from them in real time, showed that this platform is well adapted to study IOT, because it shows different aspects of creating requests and responses.

References

1. L. Tan, "Future Internet: The Internet of Things," in 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE) , Shanghai, 2010.
2. M. K. K. Y. Satoshi Kawaguchi, "Mutually Complementary Network System between Wired and Wireless Communication for Home Appliances," The 13th IEEE International Symposium on Consumer Electronics (ISCE2009), 2009.
3. M. A. K. A. M. A. Shadi Al-Sarawi, "Internet of Things (IoT) Communication Protocols : Review," in 2017 8th International Conference on Information Technology (ICIT) , Penang, 2017.
4. SIMCom, "SIM800c," 16 10 2017. [Online]. Available: https://simcom.ee/documents/SIM800C/SIM800C_Hardware_Design_V1.05.pdf. [Accessed 8 August 2020].
5. K. G. R. Johari, "IOT based Electrical Device Surveillance and Control System," pp. 2-3, 2019.
6. J. Hu, S. Zeng and Z. Zhang, "The Design of Wireless Data Acquisition System," in 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, Guilin, 2012.
7. Microchip Technology Inc, "AVR-IoT WG Development Board User Guide," 2020. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/AVR-IoT-WG-Development-Board-User-Guide-DS50002809C.pdf. [Accessed 25 July 2020].
8. B. Yang, "Design and Implementation of Intelligent Home," in 4th International Conference on Information Science and Control Engineering, Guilin, 2017.
9. N. Nikolov and O. Nakov, "Research of Communication between STM32L475 and Private Cloud realized by using Amazon FreeRTOS and MQTT," in 27-th National Conference with International Participation "TELECOM 2019", Sofia, 2019.
10. "The Role of WiFi in IoT," Maravedis, 31 March 2020. [Online]. Available: https://www.iotforall.com/wifi-role-iot/. [Accessed 15 July 2020].
11. Espressif Inc, "ESP8266," 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Accessed 8 August 2020].

12. S. S. Pushkar Singh, "Arduino-Based Smart Irrigation Using Water Flow Sensor, Soil Moisture Sensor, Temperature Sensor and ESP8266 WiFi Module," in IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Noida, 2016.
13. Telegesis (UK) Ltd, "ETRX2 ZigBee Modules AT-Command Dictionary," June 2009. [Online]. Available: http://wless.ru/files/ZigBee/ETRX2/TG-ETRX-R302-Commands.pdf. [Accessed 25 July 2020].
14. STMicroelectronics, "Discovery kit with STM32F407VG MCU," May 2017. [Online]. Available: https://www.st.com/resource/en/user_manual/dm00039084-discovery-kit-with-stm32f407vg-mcu-stmicroelectronics.pdf. [Accessed 25 July 2020].
15. N. K. Walia, "An IOT by Information Retrieval approach:Smart Lights controlled using WiFi," in 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016.