UNITECH 2017 GABROVO

INTERNATIONAL SCIENTIFIC CONFERENCE
17-18 November 2017, GABROVO

A SMART HOME APPLICATION BASED ON IoT

Aleksandar Č. Žorić
Faculty of Technical Sciences
University of Priština, Serbia

Čedomir A. Žorić
Bitgear Wireless Design Services
Belgrade, Serbia

Siniša S. Ilić
Faculty of Technical Sciences
University of Priština, Serbia

Bojan Jovanović
The Technical School-Uroševac,
Leposavić, Serbia

Abstract

This paper describes the development of one of numerous IoT applications based on ESP8266 SoC and IC2Cloud server. This application is intended to monitoring and control the indoor temperature with email alert via WiFi network. It is described the hardware design of the application, mobile application for the android platform as well as the device cloud development like the distributable virtual machine. The application designed in that way provides the possibility of developing its own user-friendly android interface to interaction with the hardware. Aditional advantage of such approach of the designing is arbitrary number of devices on the cloud server without the cloud fee.

Using the described design procedure, it is possible to developing more sophisticated applications for home automation.

Keywords: ESP8266 SoC, MIT application inventor, WiFi network, android, device builder, IoT, IC2Cloud.

INTRODUCTION

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The Internet of Things World Forum (IoTWF) predicts that the worldwide installed base of IoT endpoints will grow from 14.9 billion at the end of 2016 to more than 82 billion in 2025. At this rate, the IoT may soon be as indispensable as the Internet itself [1]. Also, broadband Internet is become more widely available, the cost of connecting is decreasing, more devices are being created with Wi-Fi capabilities and sensors built into them, technology costs are going down and smart phone penetration is unmeasurable. All of these things are creating a perfect base for the IoT implementation in the companies, smart buildings, smart cities, transportation, healthcare etc [2]. The inter-networking of physical devices (smart devices or connected devices) and other items embedded with electronics, software, sensors and actuators should enable these objects to collect and exchange data across existing network infrastructure [1], [4].

The designing a simple and complete solution of IoT application to remote monitoring and control the temperature with automatic alert is presented in this article.

DEVICE HARDWARE

The designed hardware is based on the well known System on Chip ESP8266 WiFi module [3] with integrated TCP/IP stack, suitable for adding WiFi functionality to an existing microcontroller. With the complete and self-contained Wi-Fi networking capabilities, ESP8266 can perform either as a standalone application or as the slave to a host MCU.

The SoC is a 32-bit processor (enhanced version of Tensilica's L106 Diamond series), running at 80 MHz, with integrated SRAM, some ROM and WiFi radio.

The compact design minimizes the PCB size and requires minimal external circuitries.

It can be interfaced with external sensors and other devices through the GPIOs.

The ESP8266-01 8-pins very low cost development board with a limited number of I/O lines has been used in this article. This board contains ESP8266 SoC, few capacitors, crystal, an external quick SPI Flash memory and a PCB trace antenna. An integrated UART module has been used to programming the ESP8266-01 board. Owing to such choice of the main component (ESP8266-01), the minimum number of the external components has been implemented to the final IoT design.

Figure 1 shows a simple schematic of the IoT hardware and the component arrangement, used in this paper.

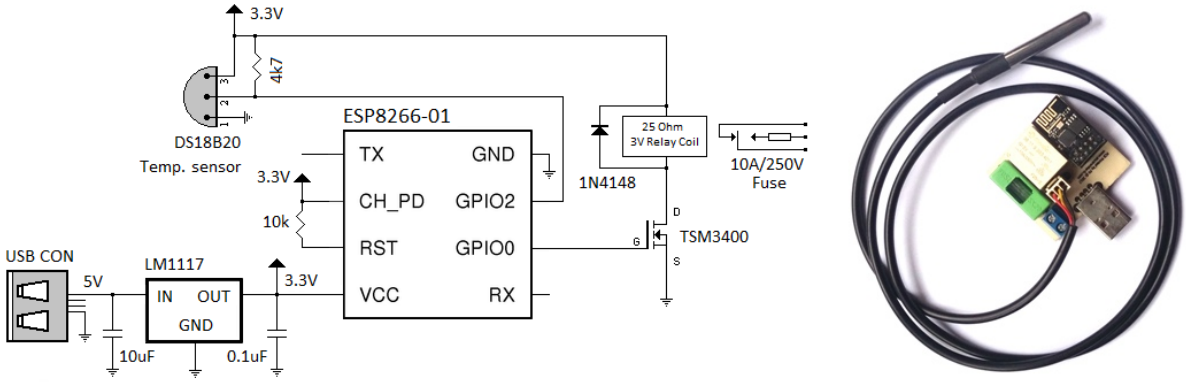

Fig. 1. *IoT hardware design.*

The IoT hardware operates at 3.3V. Average current consumption of the ESP8266 is about 80 mA, while the relay coil current amounts about 130 mA [3]. Fixed voltage regulator is designed to provide up to 1 A output current and to operate down to 1 V input to output differential.

As can be seen in Figure 1, a Dallas's one wire temperature sensor DS18B20 [7] was connected to the GPIO2 pin of the ESP8266-01 board, while for the relay coil control was used NMOS transistor.

Arduino IDE C/C++ compiler has been used to programming the ESP8266-01 board. The Arduino environment lets a user using the familiar Arduino functions and libraries and run them directly on the ESP8266 without an external microcontroller.

ESP8266 Arduino core comes with libraries to communicate over WiFi using TCP and UDP, set up HTTP, mDNS, SSDP, and DNS servers, do OTA updates, use a file system in flash memory, work with SD cards, SPI, One-Wire and IIC peripherals.

DEVICE CLOUD

IC2Cloud is an IoT open platform reducing the complexity of IoT development [5]. The main goal is to move the hardware intelligence into the cloud and minimize the tasks executed by the hardware. Figure 2 shows the main logic of IC2Cloud IoT platform.

As can be seen in the Figure 2, the hardware connected to the IC2Cloud sends information like temperature, pressure, humidity and other data and it can also receive information from the platform.

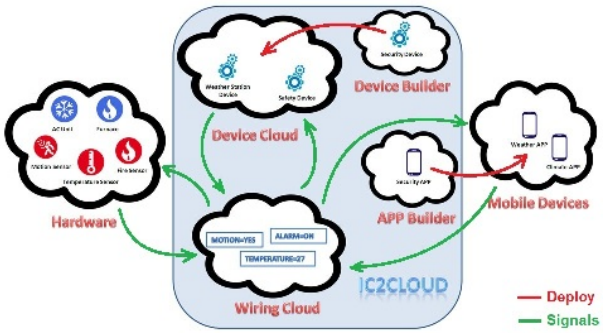

Fig. 2. *Main components of the IC2Cloud IoT system.*

The "*Wiring Cloud*" is the binding between all of other components of the platform. All of signals exchanged in the platform are handled by this component.

"*Device Builder*" is the development tool that allows user to create the virtual device, while the "*Device Cloud*", as a distribute virtual machine, runs the devices created by the "*Device Builder*" tool. Therefore, the "*Device Builder*" and "*Device Cloud*" are software components running in the cloud that has the role to execute a certain logic specific to the overall desired functionality.

IC2Cloud platform provides an extensive visual language that gives the ability to run any logic in the cloud while the hardware simply acts as sensor or an actuator.

"*APP Builder*" is the tool to create the mobile application for monitor and control of the system on a smart phone or a tablet.

Created device cloud logic is shown in the Figures 3.a and 3.b.

Fig. 3.a. *Initial logic of the "Device Cloud".*

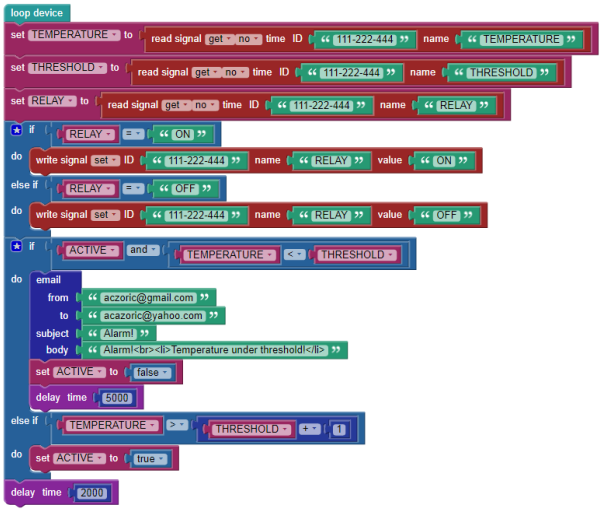

Fig. 3.b. *Main loop of the "Device Cloud".*

As can be seen in Figure 3.a and 3.b, the cloud logic is similar to a puzzle like language using blocks that is very easy to learn. Using drag and drop functions the user can build a device quickly. This type of language insulates the user from the complexities of a classic language like C or Java, but it does not lose from the power of a classic programming language [5].

The cloud logic is fairly simple. After initialization of the signals, Figure 3.a, the device cloud will read the temperature, threshold and state of the relay. It will send an email if the temperature goes under the threshold. The threshold signal is set using the mobile application. Once that is done the device cloud can be started. To test and run the device cloud, the "*Signal Manager*" and "*Device Manager*" tools are available.

MOBILE APPLICATION

The mobile application was created using App Inventor platform from MIT [6]. App Inventor consists of the "*Designer*" and "*Blocks Editor*".

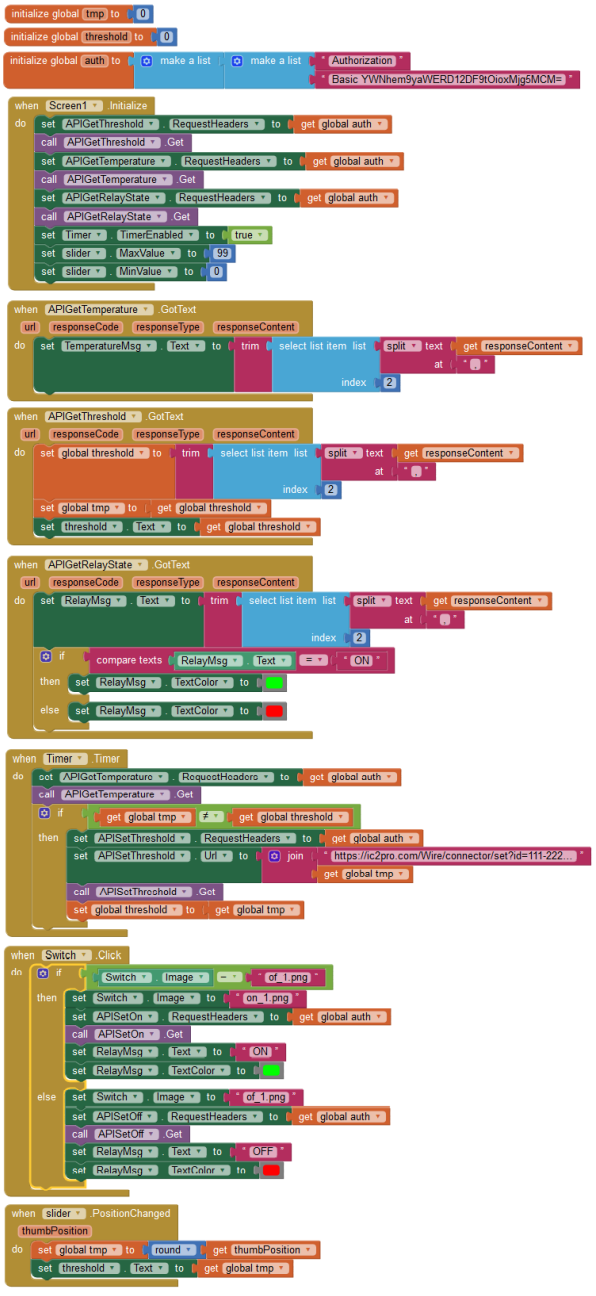

Fig. 4. *Program structure of the mobile application.*

Using the "*Designer*", an user can arranging on- and off-screen components from the component's palette, drag components from the palette to the viewer and change the components properties (color, size, behavior).

The block diagram was created by finding the blocks in the component-specific drawers, by dragging them to the "*Blocks viewer*" or

"*Blocks Editor*" and by mutual connecting. Consequently, by dragging blocks from the drawers to the "*Blocks viewer*", the appropriate relationships and behaviors were created. In total this represents the program.

Figure 4 presents the created block diagram of the mobile application. As can be seen in the Figure 4, the structure of the program is similar to the program structure of the "*Device Cloud*" creating by "*Device Builder*" (puzzle) on the IC2Cloud IoT platform.

After initialization the screen and some global variables, see Figure 4, three APIs were created for temperature, threshold and relay state reading. When periodical Timer event occurs, the API set the temperature threshold, below whom the application send email alert. When clicking to the switch, see Figures 4 and 5, API set on or off, depending on the switch state before clicking.

Fig. 5. *Screenshot of the mobile application*

The temperature threshold can be changed by using the slider. When slider position changes, the temperature threshold text sets to the thumb position.

Screen review of the described mobile application is shown in Figure 5. Besides the actual temperature indicator, an user can control a relay switch as well as the slider to setting temperature threshold.

CONCLUSION

Remote control and monitoring of things from anywhere has become possible today owing to the global Internet network.

A step by step process of designing the simple Internet of Things application with the ESP8266 WiFi module was the topic of this article. Except the ESP8266 firmware, the application hardware and the software support are fully explained. The mobile application can control any consumer as heater and notify the user if the temperature dropped below the chosen threshold.

Further improvements of this solution relate to the addition of more different digital sensors, more switches for controlling a larger number of consumers as well as multiple decision thresholds.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia within the project III 47016.

REFERENCE

[1] Mung Chiang & Tao Zhang: '*Fog and IoT: An Overview of Research Opportunities*', IEEE Internet of Things Journal, Vol. 3, Issue 6, 2016, pp. 854-864.

[2] A. Zanella, N. Bui, A. Castelani, L. Vangelista, M. Zorzi: '*Internet of Things for Smart Cities*', IEEE Internet of Things Journal, Vol. 1, Issue 1, 2014, pp. 22-32.

[3] *www.espressif.com*, ESP8266EX datasheet, ver 5.4, 2017.

[4] S. Kamenov: '*Ethernet System Module for Monitoring Climate Parameters*', Proc. of Int. Sci. Conference UNITECH-2016 – Gabrovo, Bulgaria - 2016. - Vol. I. - P. 287–290.

[5] *www.ic2cloud.com/documentation*

[6] *http://appinventor.mit.edu/explore/library.html*

[7] *https://datasheets.maximintegrated.com,* DS18B20-one wire temperature sensor datasheet.