NFC enabled Wi-Fi managging system for ESP32 based IoT system

Nikola Mitrović

Department of microelectronics,
Faculty of electronic engineering
Niš, Serbia
nikola.i.mitrovic@elfak.ni.ac.rs

Milan Đorđević

College of Applied Technical Sciences
Niš, Serbia
milan.nebojsa.djordjevic@gmail.com

Sandra Veljković

Department of microelectronics
Faculty of electronic engineering
Niš, Serbia
sandra.veljkovic@elfak.rs

Danijel Danković

Department of microelectronics
Faculty of electronic engineering
Niš, Serbia
danijel.dankovic@elfak.ni.ac.rs

Abstract— This paper presents system that manages Wi-Fi credentials for ESP32 based IoT system. Managing system is based on NFC technology. Paper offers a solution for the problem of frequent changing of Wi-Fi credentials when an IoT system is moved from place to place and needs to connect to different access points. NFC technology is chosen as a medium because it is widely accessible in modern smartphones, and also offers low-cost solutions. Data containing SSID name and password of access point that IoT system should establish connection to are written into NFC tag using smartphone. Using designed system, that data is transferred to ESP32. Speed and form of that data transfer is analyzed. In systems where speed is not of dominant interest, described system can deliver appropriate results.

Keywords - Internet of Things, Wi-Fi, ESP32, NFC.

I. INTRODUCTION

Continuous growth of the Internet of Things field led to development of many devices that are able to perform many specific tasks using many different communication methods. Great versatility of IoT devices, beside many qualities, also constantly brings some new challenges [1, 2]. Depending on the special application of the device and the primary wireless communication method, different applications put pressure on different resources of devices. This paper concerns systems where the primary wireless communication method is through local accessible Wi-Fi access point. One of the specific problems that occurs in the later phases of Wi-Fi enabled products implementation is the method of acquiring Wi-Fi configuration data (SSID name and/or password) for local access point [2-4].

In the most cases, during the testing and debugging phase, Wi-Fi credentials are hardcoded into the code. Development of the device also includes testing of other properties of the device, so most of testing engineers are focused on operation of the device. Still, most of the IoT devices that uses Wi-Fi as a primary wireless data transfer method are designed to have vast area of application, meaning, beside other, that it can be used in various places, using different access points. To fully exploit this capability, it is needed that designed devices have intuitive and straightforward method of inserting SSID name and password for the targeted access point. This problem is usually solved with adding of some input circuit that can enter these credentials. Most basic solution is to connect a small keyboard and a display so that all of the needed data can be entered. Still, this solution impact the size of the devices drastically and demands supply for many components. Solution is improved with adding of touch-screen displays. In that way, keyboard is not necessary anymore. Display is anyway part of almost all devices, so this improves the solution drastically. On the other hand, touch-screen displays demand a lot of memory resources by the driving MCU, robust libraries and are still not suitable for low-cost solutions. Therefore, it is needed to develop a method that is versatile to use, energy efficient and adapted to low-cost IoT solutions [3, 4].

II. METHODOLOGY

Basic use of the Wi-Fi managing system is to ensure that Wi-Fi credentials are dynamic and editable. Dynamic credentials are usually realized using dynamic buffer. During development phase, credentials for the main access point can be loaded to the buffer. If the main access point is accessible, after booting, device attempts to connect to the main access point. If it succeeds, it remains connected for the further operation. However, it is needed to accordingly handle cases when the connection is not successful. Sometimes, connection fail because of temporary spontaneous disruptions. In those cases, with multiple connection attempts, device finally connects and operate further, before it returns to sleep or low energy mode. There are cases when the mainly used access point is simply unavailable (in case of power shutdowns, constant obstacles or simple modem or router malfunctioning). Then, it is still needed to perform measurements and or actions, but instead of sending it, data is stored into the back up memory unit, most often a SD card [5, 6]. Even more often, it occurs that device used on one location using an access point should be mobile, and needed to use in different locations. On every different location, device connects to a different access point. Therefore, it is needed to allow simple process of inserting credentials. Proposed solution focuses on this problem.

Most of the modern smartphones come with NFC (near field communication) module. NFC is a wireless short-range and high frequency communication technology that allows the exchange of data between devices [7]. It is is based on RFID principle. System consists of a tag and a reader. When a tag is put close enough into the reader field, data is transferred using load modulation. Comparison between RFID and NFC technologies are given in table I [8, 9].

TABLE I: COMPARISON OF RFID AND NFC TECHNOLOGIES

	RFID	NFC
Communication type:	Unidirectional	Bidirectional
Range:	< 1 m	< 0.1 m
Operating frequency:	LF – Microwave range	13.56 MHz
Set-up time:	< 0.2 s	< 0.2 s
Avg. consumption:	< 35 mA	< 15 mA
Tag data size:	< 1 kB	< 8 kB

Main difference between the most of commercial RFID systems and NFC systems is that using NFC, unlike RFID, data transfer can be done in both directions, both from a tag to a reader and from a reader to a tag. A lot of NFC tags come with enough memory to take enough data (commercial ranging from 512 B to even 8 kB), and are widely available as flexible tags, gluing tags or even wearable tags. In this paper, Wi-Fi credentials managing system is implemented using NFC data transfer. There were researches concerning using NFC for information transfer [9, 10]. It was concluded that this technology requires minimum human operation and that it offers a lot of possibilities for further implementation.

III. EXPERIMENT

In order to connect to the new Wi-Fi access point, data containing new Wi-Fi credentials is to be transferred to NFC tag. Tag is then put in front of the reader, where the data is read and transferred to the ESP32 device. Data is then parsed and two specific strings (SSID name and password) are extracted. ESP32 attempts to connect to the local Wi-Fi network using extracted data. Block schematic of the experiments is shown in the Fig. 1.

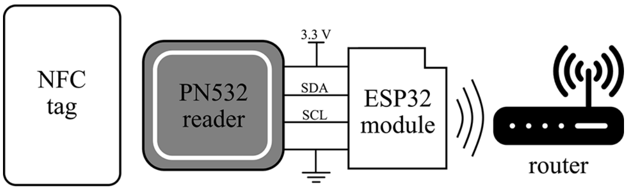

Fig. 1.Block schematic of the setup.

A. Used devices and tools

Tags used in these experiments are widely accessible Mifare Classic tags [11] with a memory size of 1 Kilobyte (kB) that use ISO 14443 standard. This type of tags contains Unique Identifier Number (UID) and is NFC Data Exchange Format (NDEF) enabled. UID for every tag is different, seven bytes long and cannot be changed. Tag being NDEF enabled means that it can receive NDEF messages. Writing data to a tag corresponds to sending appropriate NDEF message to the tag. NDEF content of the tag is formattable and rewriteable. In this paper, only NDEF method for data transfer is used.

For the transfer to be accomplished, it is needed to write data into NFC tag. In described experiments data containing Wi-Fi credentials is loaded into the tag via Android smartphone. Freeware third party application NFC Tools is used [12]. Using this application, it is possible to write different types of data to the tag (simple strings, active mobile phone number, Wi-Fi credentials, links, and other). Data is written with selecting specific options and putting tag close to the smartphone.

Data is read from the tag using NFC reader. For these experiments, PN532 reader is used [13]. It is a NFC reader with square embedded antenna that can read multiple types of the NFC tags (including Mifare Classic tags) in the 13.56 MHz range. It is based on 8051 MCU and can communicate with other devices using SPI, I2C or UART protocols. This reader is selected because it can read both UID number of tags and NDEF messages.

Main part of the setup is ESP32 board with WROOM2 chip, manufactured by the company Esspresif [14]. This device is selected because it contains Wi-Fi module and therefore can connect to Wi-Fi networks. The module support 802.11 b/g/n protocol in the standard Wi-Fi frequency range (2.4 GHz - 2.5 GHz). Besides Wi-Fi capabilities, chip contains peripheral units for many serial interfaces (I2C, SPI, UART), as well as RTC and Bluetooth modules. These properties make this chip suitable for many IoT applications using different methods [15, 16].

In these experiments, PN532 and ESP32 are connected over two lines, using I2C interface. After reading NDEF messages, raw data is sent from the PN532 to the ESP32. Data is then processed and the connection is attempted. Software for the data receiving, data parsing and Wi-Fi connecting is manually coded and flashed to ESP32.

B.Experiment setup and results

Experiments are divided into three rounds. In the first round, Wi-Fi credentials are written on the NFC tag using NFC Tools. Tag is then put in front of a reader in order to start data transmission. When that is done, an external timer is started. Wi-Fi credentials in the form of string data are transferred from the tag to the reader and then from the reader to ESP32 over I2C. ESP32 receives the string, parses the data and tries to connect to the access point with the given credentials. When the connection is successful, external timer is stopped. In this way, it is possible to measure time between the putting of a tag in front of the reader and the successful connection. Figure 2 shows the results from ten consecutive connection attempts.

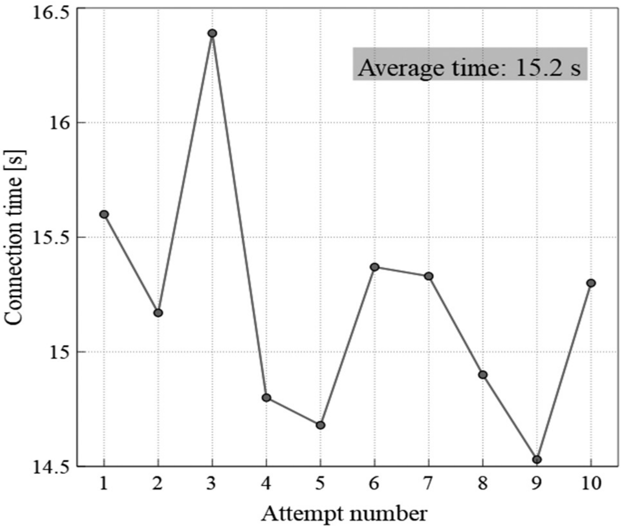

Fig. 2. Connection time for ten connection attempts with received credentials..

After every attempt, devices were reset, so that process can be started from the beginning. During all ten attempts, distance between the ESP32 and the router access point is kept constant. As can be seen from the Fig. 2, time between putting tag in front of a reader to establishing connection is ranging from 14.53 seconds to 16.39 seconds. Average time for connection is 15.2 seconds. These results are in line with the similar research [10]. It is worth mentioning that the length of the credentials was 20 characters in total (10 for SSID name and 10 for password). Still in order to further analyze needed time, it is necessary to divide measured time into time for NFC data transmission, time for data processing and time for Wi-Fi connecting. In order to tackle this complicated problem, second round of experiments has to be done.

When the Wi-Fi credentials are hardcoded, and no external components are connected to the ESP32, average connection time is 5.5 seconds. It means that the processes of tag reading, data sending and processing consume around 10 seconds. To investigate time needed for every of the mentioned actions, logic analyzer probes are connected to the SDA and SCL pins of the PN532 reader, with the goal to determine starting and ending point of the data transmission between the PN532 and ESP32. For this purpose, USB logic analyzer LHT00SU1 is used. Second round of ten connection attempts are made again. Table II shows the measured results.

TABLE II: PERIOD OF DATA TRANSFER STEPS

Attempt	Time [s]				
	Reading	Data sending	Data parsing	Connecting	Total
1	6.39	2.056	0.820	5.73	15.03
2	7.35	2.036	0.780	5.16	15.33
3	7.52	2.030	0.790	5.39	15.73
4	6.91	2.052	0.780	5.47	15.68
5	7.08	2.039	0.780	5.70	15.60
6	6.65	2.041	0.780	5.35	14.83
7	7.57	2.055	0.830	5.72	16.17
8	6.96	2.039	0.800	5.49	15.29
9	8.03	2.047	0.790	5.44	16.31
10	6.72	2.050	0.780	5.12	14.67

As can be seen, longest process is reading of the data from the tag by the PN532 reader, ranging from 6.39 s to 8.03 s. Time for data transmission is guided by the used I2C protocol, so no noticeable difference between these times is expected. Same can be said for the time used by ESP32 for parsing the data. Also, as expected, after the Wi-Fi credentials are extracted, around 5.5 seconds is needed to establish Wi-Fi connection. Different times for reading data from the tag can be explained by the fact that the tag was not always put in the exact same position in front of the reader. Tag was always put in the reading range, but slight time deviations are caused with this difference.

Third round of experiments has the same setup as the previous ones. In this round, for every connection attempt, settings of the router (SSID name and password) are changed. In every change, length of these data is increased, starting from 10 characters (5 for SSID name and 5 for password) to 30 characters (15 for SSID name and 15 for password). This means that after every attempt, NFC tag should be formatted and then rewritten with the new Wi-Fi credentials. Again, timer is started when tag is put in front of the reader, and stopped if and when the connection is successful. This approach is used in order to see whether the different size of Wi-Fi credentials impacts total connection time by critical amount. Measured results are given in the Fig. 3.

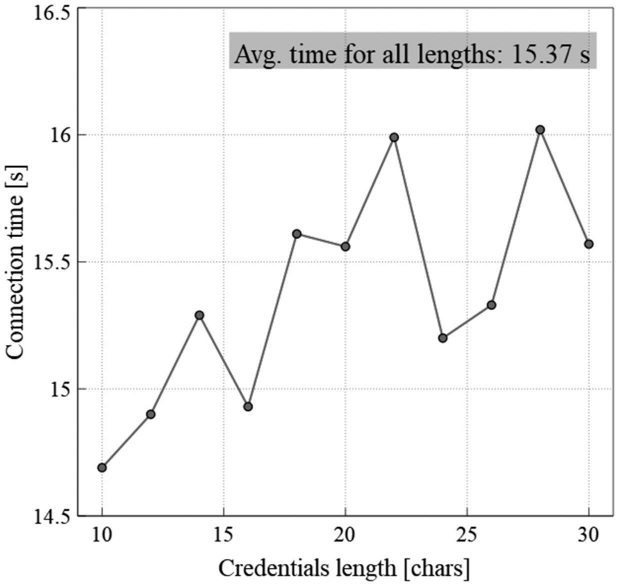

Fig. 3. Connection time for connection attempts with received credentials of different length.

As can be seen from the Fig. 3, different size of the credentials does not impact total connection time drastically. Main reason for this is in the principle of NDEF messages that used type of tag/reader uses to transfer data. Regardless of amount of data written to the tag, total memory of tag is always read by the PN532 reader. Content of total memory is being sent to the ESP32, and that is the reason why specific data parsing is needed. After credentials are extracted, ESP32 establishes Wi-Fi connection regardless of their length.

IV. FUTURE WORKS

One of the improvement points and the future steps of the proposed solution is to embed the tag into the PN532 reader, and to connect additional devices for proximity measuring to the reader. When the NFC writing smartphone is put in front of the tag embedded into the PN532, proximity circuit activates. It turns off the PN532 using additional circuitry, and enables NFC data transfer between the smartphone and the tag. When the writing into the tag is finished, and the smartphone is removed, proximity circuit deactivates and turns on the PN532 reader, that can now read data from the embedded tag. Using this improved solution, user of the system would not need to have a tag with the system.

Also, other improvement point lies in adding some small visual indicators (LEDs) or even small display that will inform the user about ongoing communication and active steps. Since plastic obstacles do not disrupt the NFC, and since the PN532 antenna is embedded into the PCB, it is possible to provide custom encapsulation of the proposed solution that can be adapted to the working environment. It is also worth mentioning that not only Wi-Fi credentials can be transferred using NFC. With the use of described tools, it is possible to send custom strings concerning many options in the generic IoT systems.

V. CONCLUSION

Simple Wi-Fi management solution for ESP32 based IoT systems that is based on NFC is presented in this paper. The problem of changing Wi-Fi credentials that ESP32 system tries to connect to, with the help of additional free tools can be solved using NFC tag data transfer. Connection does not establish instantly, it is needed averagely more than 15 seconds, but in the application where connection speed is not mandatory, proposed solution can provide reliable results. Described system does demand adding NFC reader to the IoT system, but it compensates with greater versatility and higher connectivity with other devices.

ACKNOWLEDGMENT

This work was supported by the Republic of Serbia under the Grant No. 451-03-9/2021-14/200102.

REFERENCES

[1] D. Evans, The Internet of Things: How the Next Evolution of the Internet Is Changing Everything, Cisco Reports, 2011

[2] L. Farhan, R. Kharel, O. Kaiwartya, M. Quiroz-Castellanos, A. Alissa and M. Abdulsalam, "A Concise Review on Internet of Things (IoT) -Problems, Challenges and Opportunities," In proc. of 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2018, pp. 1-6.

[3] J. Ding, T.-R. Li and X.-L. Chen, "The Application of Wifi Technology in Smart Home", J. Phys. Conf. Series, vol. 106, no. 1, 2018.

[4] M. Babiuch, and J. Postulka, "Smart Home Monitoring System Using ESP32 Microcontrollers", in Internet of Things. London, United Kingdom: IntechOpen, 2020

[5] M. Đorđević, B. Jovičić, S. Marković, V. Paunović and D. Danković, "A Smart Data Logger System based on Sensor and Internet of Things technology as part of the smart faculty", J. Ambient Intell. Smart Environ., vol. 12, no. 4, pp. 359-373, July 2020.

[6] D. Danković and M. Đorđević, "A Review of Real Time Smart System Developed at University of Niš", Facta Universitatis: Electronics and Engineering, vol. 33, no. 4, pp. 669-686, Dec. 2020.

[7] T. Igoe, D. Coleman, and B. Jepson, Beginning NFC, O'Reilly 2014.

[8] J. J. Echevarria, J. Ruiz-de-Garibay, J. Legarda, M. Álvarez, A. Ayerbe, J. I. Vazquez, "WebTag: Web Browsing into Sensor Tags over NFC", Sensors, vol. 12, pp. 8675-8690, 2012.

[9] M. H. A. Wahab, N. F. M. Suhaimi, M. F. M. Mohsin, A. Mustapha, N. A. Samsudin, R. Ambar, "NFC-based Data Retrieval Device", J. Phys. Conf. Ser., vol. 1019, no. 1, p. 012084, 2018.

[10] A. Widiyanto and M. Nuryanto, "Prototype of NFC Reader as a Attendance Sign at The Presence System", J. Phys. Conf. Ser., vol. 1196, p. 012042, 2019.

[11] Mifare Classic 1k tag datasheet, NXP semiconductors. Available at: https://www.nxp.com/docs/en/data-sheet/MF1S50YYX_V1.pdf

[12] NFC Tools Android App. Available on Google Play platform.

[13] PN532 NFC reader datasheet, NXP semiconductors. Available at: https://www.nxp.com/docs/en/nxp/data-sheets/PN532_C1.pdf

[14] ESP32-WROOM2 datasheet, Espressif. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[15] N. Mitrović, M. Đorđević, S. Veljković, D. Danković, "Testing the efficiency of Wi-Fi data transmission in ESP-based IoT systems", in Proc. Int. Conf. E-Business Technologies (EBT), 2021, pp. 110-112.

[16] N. Mitrović, M. Đorđević, S. Veljković, D. Danković, "Implementation of WebSockets in ESP32 based IoT Systems", in Proc. 15th Int. Conf. Adv. Technol. Syst. Services in Telecommunications (TELSIKS), 2021, pp. 261-264.