Near Field Communication as Sensor to Cloud Service Interface

Tore Leikanger, Christian Schuss and Juha Häkkinen
Circuits and Systems (CAS)
University of Oulu
Oulu, Finland
E-mail: firstname.lastname@oulu.fi

***Abstract*—In this paper, we present a new approach to interrogate sensors using Near Field Communication (NFC). This approach utilizes NFC data exchange format (NDEF) to report the sensor data directly to a cloud service through a smartphone. The new advantage in this approach is that no smartphone application is needed to move the data from the sensor to the cloud. We also discuss advantages and disadvantages of this approach compared to earlier NFC sensor interrogation solutions.**

Keywords—Near Field Communication, NFC, FPGA, Smartphone, Sensor interrogation, Cloud service, Internet of Things, IoT

I. Introduction

Near Field Communication (NFC) has traditionally been used for identification purposes, for example as key cards [5]. In recent years, new uses for NFC have been explored, for example the use of NFC as a sensor interface [1]. In our earlier research, we have looked into how NFC can be used to directly interrogate digital sensors using a smartphone [1].

A common problem with earlier NFC enabled sensor systems is the need of a specific smartphone application, e.g. as we previously did [1]. Even though a specific smartphone application makes the design of the NFC interrogation system easier, this smartphone application dependency is inconvenient for the end user.

In this paper, we are presenting a new way of uploading sensor data achieved over NFC directly to an online cloud. This new approach removes the need to install a smartphone application. We are also looking into to advantages and disadvantages with the approach presented in this paper compared to earlier approaches.

To verify our approach, we used our previously developed NFC system as a starting point [1]. The NFC system does not feature an energy harvesting circuit, and is thus powered through a Universal Serial Bus (USB) [1]. We developed our NFC tag further to interrogate a specific sensor, and to report the results to the smartphone. To avoid the smartphone application dependency, we used the data in the NFC data exchange format (NDEF) described in the NFC Forum standard [2].

Google used the same approach when they designed their Eddystone Bluetooth Low-Energy (BLE) beacon standard [3]. The difference between Eddystone and the system described in this paper, however, that our approach is based on NFC, while Eddystone is based on BLE. The advantage of BLE is that BLE works over a longer range [4]. The advantage of NFC over BLE, however, is that NFC works without the need of batteries [5].

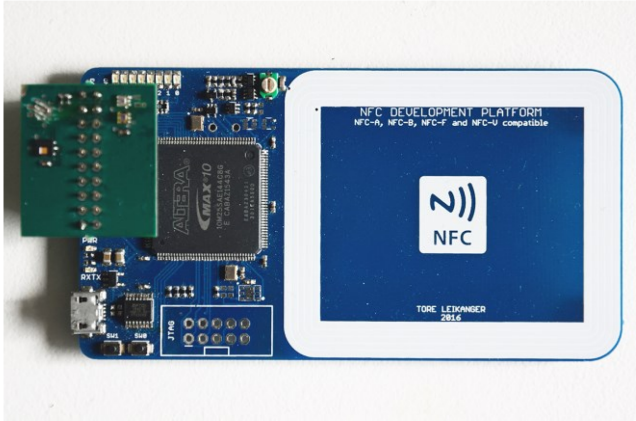


Fig. 1. Picture of the NFC development system

This paper is divided into three sections. Section II describes our approach, and section III shows how this system works. Section IV then discusses advantages and disadvantages of this system compared to earlier solutions.

II. Approach

Because our goal was that, our sensor interrogation system works without the need of a smartphone application, the NFC device needed to interrogate the sensor itself before the NDEF message was read. In order to do this task, we used our previously developed NFC development system seen in fig. 1, where we had complete control over the NFC tag functionality [1]. This development system is based on an analog front end (AFE) designed with discrete components, and a field-programmable gate-array (FPGA).

To avoid the need of a specific smartphone application, the sensor data were embedded in an NDEF message. The NDEF format is the widely used standard defined by NFC Forum, and is supported by smartphones featuring NFC Forum compatible functionality [2]. In addition, the NDEF format enabled us to refer to a webpage and embed the sensor data in the webpage address. Due to this feature of the NDEF standard, the sensor

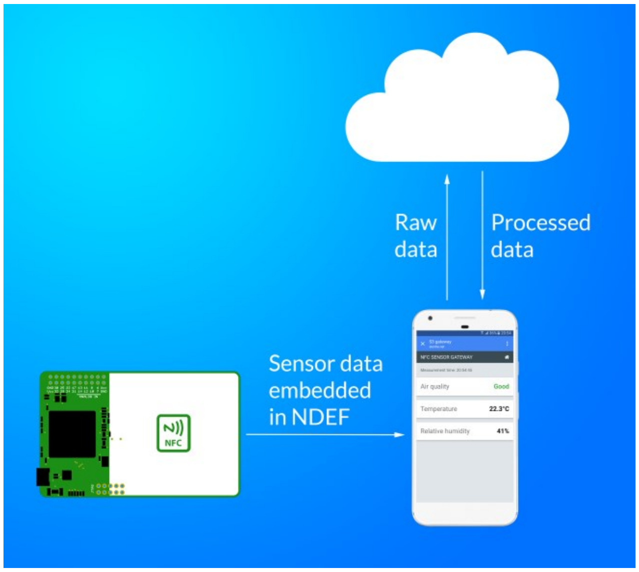
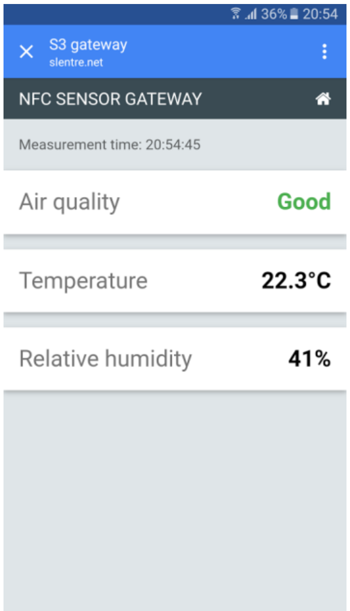


Fig. 2. The data flow in our proposed approach

data were uploaded automatically to a cloud service for processing.

We programmed the FPGA to interrogate the subjected sensor when an NFC reader (smartphone) initiated a communication channel with the developed NFC tag. Initiating a communication channel, means that the smartphone is brought close to the antenna on our NFC system. The sensor data were thus read while the NFC reader and the NFC tag were going through the anti-collision states described in the NFC Forum standard [5]. The data were thus read and embedded in the NDEF message by the time the NFC tag was ready.

The NDEF message contained, in addition to the sensor data, a link to a webpage working as a gateway to the cloud service.



S3 gateway
slentre.net

NFC SENSOR GATEWAY

Measurement time: 20:54:45

Air quality	Good
Temperature	22.3°C
Relative humidity	41%

Fig. 3. The user interface of the web page as seen after reading the NFC sensor tag

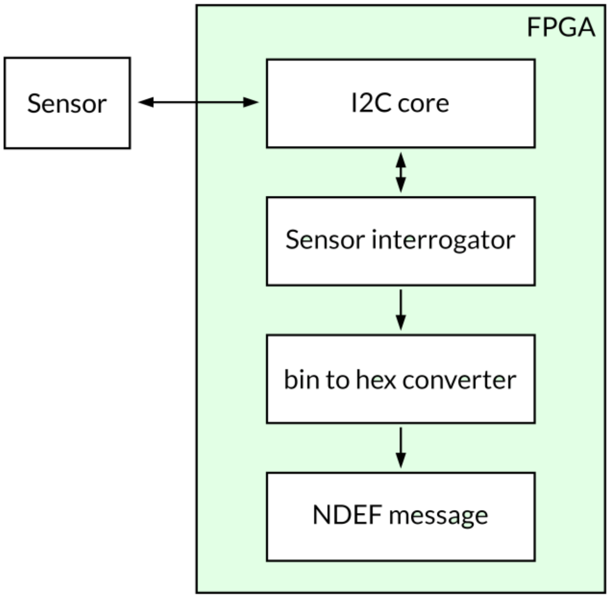


Fig. 4. The data flow in the FPGA from the sensor to the NDEF message

The raw sensor data was forwarded to the webpage through the arguments in the webpage address. This method enabled us to embed all raw data in the web page address. The data was then processed in the cloud service, and returned to the smartphone through the web page. A data flow diagram describing this can be seen in fig. 2.

III. Functionality

Verification of the functionality of our system was done using the Sensirion SHT31-D relative humidity and temperature sensor [7]. In addition, we set up a webpage working as the sensor data gateway. The only functionality of this webpage was to process the received sensor data, and to report the result back to the user. Fig. 3 shows a screenshot of this webpage as seen from an Android smartphone.

The raw data of the sensor was embedded in the NDEF message as part of the webpage address. The address was on the form

http://slentre.net/s3/sgw/#62A0&513C,

where 62A0 and 513C are the raw temperature and humidity data as read from the sensor, respectively. The raw data were sent as hexadecimal values to shorten the web address length.

The digital design of the NFC core was based upon our previous research [1], where we modified the inter-integrated circuit (I2C) core to work without being controlled by a smartphone. The new addition to the digital design was the sensor interrogator module, controlling the I2C core, and the binary to hexadecimal converter. The binary to hexadecimal converter module also inserted the values into the NDEF message. Fig. 4 shows the sensor to NDEF message data flow.

The cloud service received the raw data through arguments in the webpage address, meaning that the data was encoded as mentioned above. By using JavaScript, we were able to calculate the temperature and humidity values based on the raw data, using the equations given in the SHT31-D datasheet [7]. These values were then returned to the user through the webpage front end of the cloud service, as shown in Fig. 3.

IV. Discussion

This approach makes reading a sensor using a smartphone easier. However, the system design is harder for the designer compared to our earlier solution [1]. Reading digital sensors through NFC is also possible using commercially available NFC integrated circuits (IC) by e.g. NXP, Texas Instruments and ST Microelectronics. However, these depends on a microcontroller (MCU) to operate [8], [9], [10]. In addition, these commercially available solutions depends on a smartphone application to operate correctly [8], [9], [10], which again makes the user experience less convenient, compared to the approach presented in this paper.

This proposed approach, however, is not as flexible as earlier solutions. Our previously proposed system gives the smartphone application full control over the I2C bus, meaning that the NFC circuit can be used to read any I2C enabled sensor without reprogramming [1]. Further, commercially available NFC ICs depend on an MCU to interrogate the sensors, and thus requires programming. MCU based systems are however flexible because MCUs can be reprogrammed to fit any sensor system [8], [9], [10].

However, the approach presented in this paper can be made more flexible, if the sensor interrogator has a lookup table, where all I2C commands are predefined, and reprogrammable using e.g. a smartphone. This lookup table can be placed at certain addresses in the electrically erasable programmable read-only memory (EEPROM) found in most NFC ICs, for example.

Another improvement of our approach is to include an analog-to-digital converter (ADC), as was done by A. Zaher et. al. [11]. By including an ADC and applying our approach, also resistive and capacitive analog sensors can be read such that the results are uploaded directly to a cloud service without needing a smartphone application. Also the functionality of the ADC can be programmed using a lookup table as proposed above.

V. Conclusion

In this paper, we have presented a new way of reading a digital I2C enabled sensor, focusing on the usability for the end user rather than the flexibility of the system. We showed that an NFC enabled sensor system is possible to read without a specific smartphone application. Further, our proposed approach use a cloud service to convert raw data to human readable data, meaning that sensor data is available for other cloud service based applications as well.

Further, we briefly looked into advantages and disadvantages with the proposed approach compared to earlier approaches. We found that earlier approaches are more flexible. However, we also found that the approach proposed in this paper is easier to use, because no smartphone application is needed.

In addition, we also discussed future development of the proposed approach. Most importantly, the described approach can be made more flexible. Flexibility can be improved for example by embedding a lookup table in the EEPROM embedded in the NFC IC, where all I2C commands are found in the correct order. By including a lookup table like this, reprogramming of the developed system is performed using an NFC reader, for example a smartphone. Further, the system can be improved by including an ADC to read resistive and capacitive sensors.

References

[1] T. Leikanger, J. Häkkinen and C. Schuss, "Interfacing external sensors with Android smartphones through near field communication," Measurement Science and Technology 28.4 (2017): 044006
doi: 10.1088/1361-6501/aa57da

[2] “NFC Data Exchange Format (NDEF)”, NFC Forum Technical Specification, Rev. 1.0, 2006

[3] “Google Beacon Platform: Platform Overview“, https://developers.google.com/beacons/overview, Accessed 11.05.2017

[4] “Discover Bluetooth”, https://www.bluetooth.com/what-is-bluetooth-technology/discover-bluetooth, Accessed 12.05.2017

[5] K. Finkenzeller, “RFID handbook : fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication, third edition”, Wiley, 2010.

[6] “Type 2 Tag Operation", NFC Forum Technical Specification, Rev. 1.2, 2014

[7] “Datasheet SHT3x-DIS. Humidity and Temperature Sensor”, technical report, version 4, 2017.

[8] “NTAG I2C - Energy harvesting NFC Forum Type 2 Tag with field detection pin and I2C interface datasheet NT3H1101/NT3H1201", technical report, 2015.

[9] “RF430FRL15xH NFC ISO 15693 Sensor Transponder”, technical report, Rev. C, 2014

[10] “Dynamic NFC/RFID tag IC with 4-Kbit EEPROM, NFC Forum Type 4 Tag and I²C interface”, technical report, Rev. 15, 2017

[11] A. Zaher et al, "Integrated electronic system for implantable sensory NFC tag," 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, 2015, pp. 7119-7122.
doi: 10.1109/EMBC.2015.7320033