International Journal of Internet, Broadcasting and Communication Vol.12 No.2 77-82 (2020)
http://dx.doi.org/10.7236/IJIBC.2020.12.2.77

IJIBC 20-2-11

IoT Device Testing for Efficient IoT Device Framework

Dong-Hwan Gong

Assistant Professor, Department of Electronic Software, Hansei University, Gunpo, Korea
armyvision@hansei.ac.kr

Abstract

IoT devices frequently require input resources to communicate with various sensors or IoT platforms. IoT device wastes a lot of time as idle time or waiting time to check the data of the input resource and use the input resource. In addition, IoT devices use various input resources. We compares and analyzes input idle time and input waiting time generated from hardware serial input resource, software serial input resource, digital port input resource, and analog port input resource using Arduino widely used as IoT device. In order to design the IoT device framework, it is necessary to understand the characteristics of input resources and to design them to minimize unnecessary input idle time and input waiting time. The analog input wait time has a much larger input wait time than the digital input wait time, so it must be designed to receive analog information periodically at the appropriate timing. The characteristics of the input resources analyzed in this way help to design an efficient IoT device.

Keywords: *Use IoT, IoT Device, Arduino, IoT Device Framework, Input Idle Time, Input Waiting Time*

1. Introduction

IoT devices require frequent external input / output resources to receive various sensor information in real time or to communicate with another IoT device or IoT platform[5]. Periodic input idle time and input wait time are required for the IoT device to input / output with various sensors or to communicate with the IoT platform[5]. If the IoT device I / O framework is not configured properly, data processing problems or I / O side effects may occur, resulting in device inefficiencies or ineffective performances. Therefore, it is important to understand the input / output characteristics of IoT devices in order to efficiently configure essential sensor and network technologies with limited resources and power of IoT devices[1-3].

In this paper, we intend to measure the latency of input resources and analyze the characteristics of input resources to construct an efficient IoT device framework using Arduino and common libraries widely used as IoT devices. Arduino is widely used as an IoT device as open source hardware and provides an integrated development environment for rapid development and testing. There are various types of Arduino devices, and this paper uses Arduino Uno R3. The input resources used for testing are tested using hardware serial input resources, software serial input resources, digital port input resources, and analog port input resources. For

Manuscript Received: March. 10, 2020 / Revised: March. 14, 2020 / Accepted: March. 20, 2020
Corresponding Author: armyvision@hansei.ac.kr
Tel: +82-31-450-5340, Fax: +82-31-450-5172
Assistant Professor, Department of Electronic Software, Hansei University, Gunpo, Korea

input idle time and wait time test, first, measure the total execution time of Arduino when there is no input / output. Second, measure the idle time of available() function input of hardware serial and the idle time of available() function input of software serial. In order to receive data such as user input or Bluetooth, the third, digital GPIO input wait time is measured for each PWM port and non-PWM port in serial. Third, the analog input waiting time is measured for each port. Fourth, the IoT device framework is configured to have a better IoT input latency based on the measured input latency.

2. IoT device and test environment configuration

IoT devices are devices that have basic communication functions and are generally responsible for receiving various information through sensors[4]. It communicates with the IoT platform using the Internet communication network, and exchanges key information and control information[6-7]. IoT devices receive and process information and store information based on the ability to store information, and exchange it with other IoT devices or IoT platform through a connected communication network[4].

The IoT device Arduino used in this paper provides a digital input port and an analog input port to receive information from various sensors. Arduino provides an integrated development environment for program development that runs on various OSs, enabling rapid development and testing. In addition, many companies have developed Arduino compatible shields that have specific functions and are developed as open source, so it is possible to expand the functions to independent modules simply by combining with each device module. With hardware serial and software serial, you can make Arduino module that can communicate with Bluetooth function shield, Wi-Fi function shield, and Ethernet function shield by simply combining. Arduino has developed a variety of libraries based on the most widely used programming language C ++ language and is widely used for IoT device testing[8-10].

The Arduino Uno R3 used in the test environment provides 6 analog inputs from A0 to A5 and 14 digital GPIOs from 0 to 13, and can use one hardware serial and multiple software serials (using digital GPIO) using the Arduino common library.

Table 1 shows the test environment as a table. Arduino Uno basically provides hardware serial ports with digital ports 0 and 1.

Table 1. Test environment

Test environment veriable	Test environment value
IoT Device	ARDUINO UNO Rev3
H/W Serial Port	GPIO 0(Rx), 1(Tx)
Unit of Measure	Times per second
S/W Serial Port	No1 : 10(Rx), 11(Tx) No2 : 7(Rx), 8(Tx)
Digital Port	2, 3, 4, 9
Analog Port	A0, A1, A2, A3

The software serial uses the Arduino common library, Software Serial library. The first software serial was tested using digital ports 10 and 11 as Rx and Tx respectively, and the second software serial was tested using digital ports 7 and 8 as Rx and Tx respectively. Arduino Uno software serials can be installed in multiple ways, but two software serials and one hardware serial were used for comparison. Ports for digital input waiting time

use digital ports 2, 3, 4, and 5, and ports for analog input waiting time use analog ports A0, A1, A2, and A3. The input waiting time is measured by the number of times the loop() function is executed per second. Up to three software serials are compared and measured for serial input idle time along with hardware serials. Digital input waiting time is measured using up to four ports, and analog input waiting time is measured using up to four ports.

3. Input idle time and input waiting time test

Arduino devices frequently use serial resources to communicate with other external devices and use the available() function to determine whether data has arrived. Since we don't know when the data will arrive, a basic idle time occurs while using the available() function, and it is important to understand these characteristics. Figure 1 shows the idle time consumed when the available() function of hardware serial and software serial is called. As a result of 10 measurements, the loop count axis is the number of executions of the loop() function. Software serial using Arduino's common library is about 5% faster than hardware serial. The first line is the result of measurement using one software serial, and the gray line is the result of superimposing two software serials. When using two software serials overlapping a 27% performance decrease occurs than the result of using one software serial for Arduino.

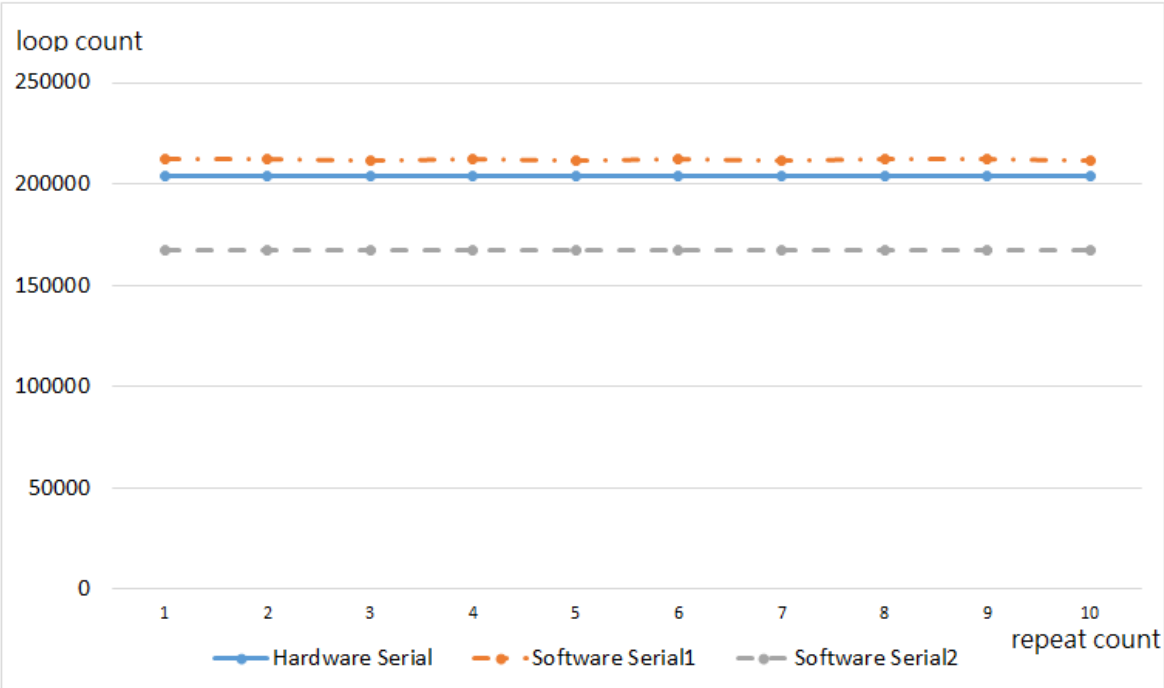

Figure 1. Hardware serial and software serial input idle time comparison

IoT devices use many analog ports and digital ports to receive various sensor data in real time, and the Arduino Uno R3 provides 6 analog inputs and 14 digital inputs. Figure 2 shows the result of measuring the input waiting time for four digital inputs using the digitalRead() function. If you overlap two or more digital inputs than one digital input waiting time, you can see that the input waiting time is more than double delayed.

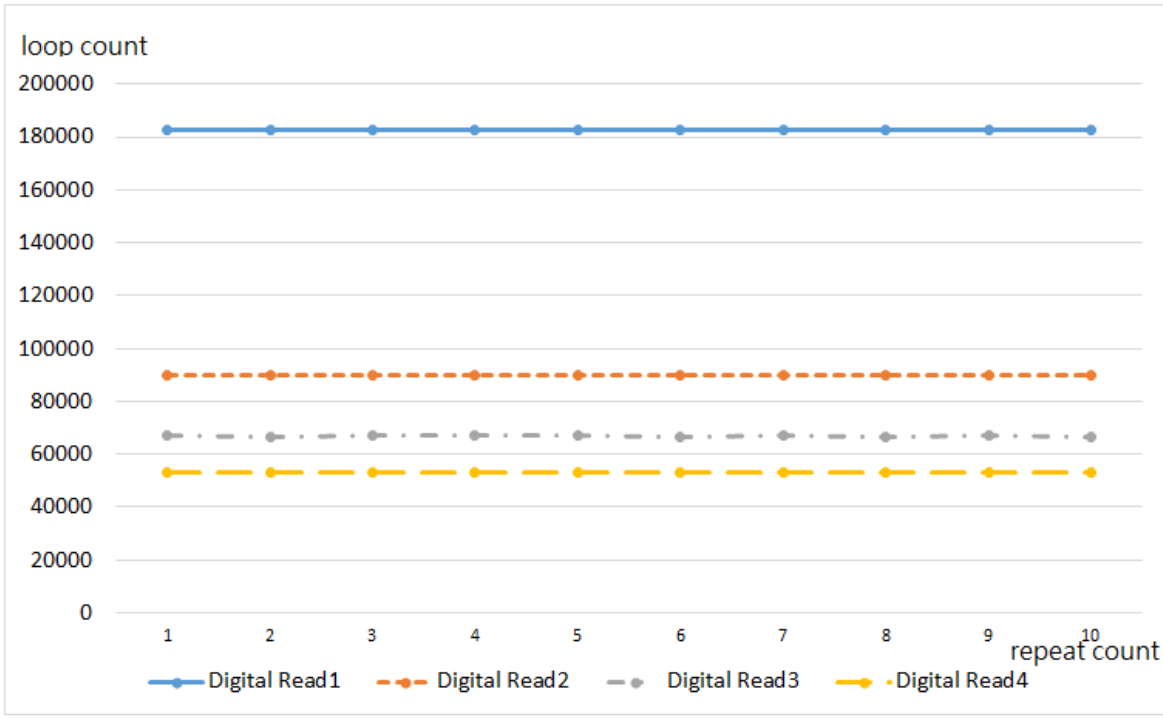

Figure 2. Comparison digital input waiting time and overlapping input waiting time

Figure 3 shows the result of measuring the input waiting time for four analog inputs using the analogRead() function. If you overlap two or more analog inputs than one analog input wait time, you can see that the input wait time is more than doubled, similar to the digital input time. In addition, it can be seen that the analog input standby time is about 20 times slower than the digital input standby time.

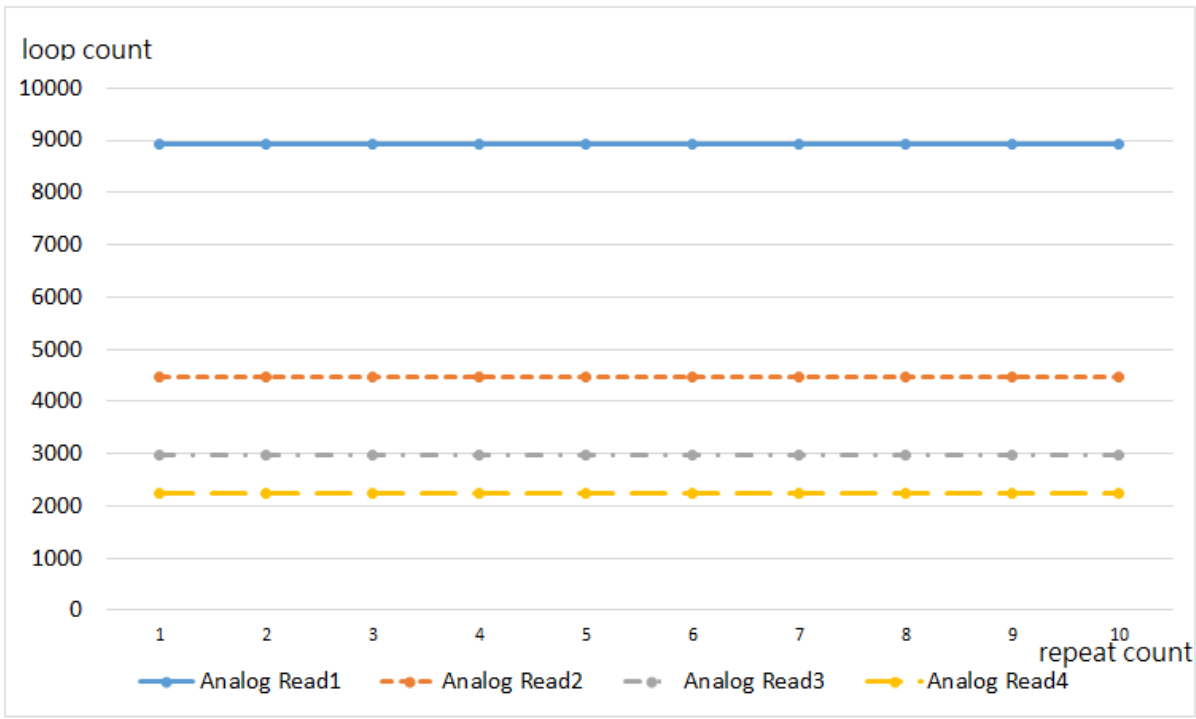

Figure 3. Comparison analog input waiting time and overlapped input waiting time

4. Results

According to the test results, the hardware serial input idle time is slower than the software serial input idle time. The analog input wait time has a much larger input wait time than the digital input wait time, so it must be designed to receive analog information periodically at the appropriate timing.

Figure 4 shows the execution time of the loop() function when there is no input, the input idle time consumed when the available() function of hardware serial and software serial is called, and the analog / digital input wait time. Additionally, the time spent on digital output was also added as the fifth line. You can see that the input idle time of the software serial available() function takes the least time and the analog serial input wait time takes the longest.

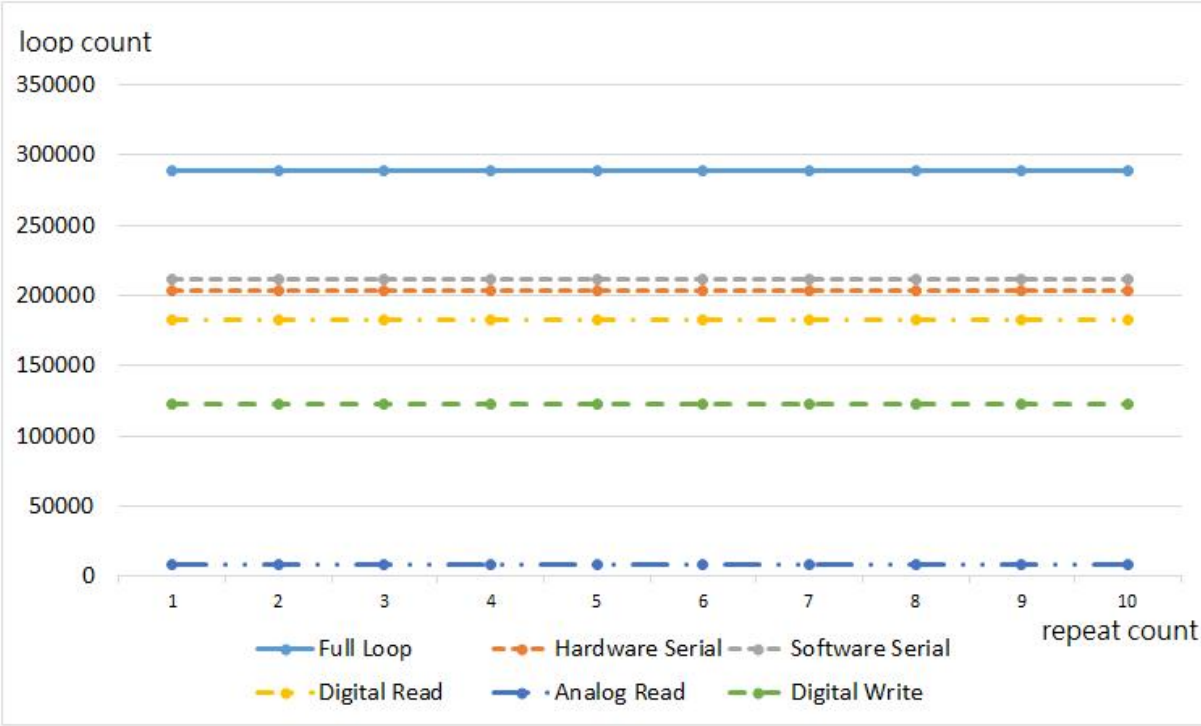

Figure 4. Comparison of input idle time and input waiting time

5. Conclusion

We measured and tested the various input idle times and input wait times that can occur on IoT devices using Arduino Uno. Based on this, it was possible to confirm the characteristics of the input resource of the IoT device. IoT devices use a variety of input resources. Therefore, it is important to use input resources efficiently. In general, IoT devices are composed of limited resources and limited power, so to design an efficient IoT device framework, it is necessary to understand the characteristics of input resources and to minimize unnecessary input idle time and input waiting time. The analog input wait time has a much larger input wait time than the digital input wait time, so it must be designed to receive analog information periodically at the appropriate timing. In addition, digital input waiting time or serial input idle time is also a waste of time when multiple tasks overlap, so it must be designed as an IoT device capable of real-time processing suitable for each input resource.

Acknowledgement

This work was supported by Hansei University Research Fund of 2019.

References

[1] Woo Young Chang, Yong Cheol Lee, Jeong Jin Kang, "Implementation of IoT Sensors Network Using Mobius Platform", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 17, No. 2, pp.211-218, Apr 2017.
DOI: https://doi.org/10.7236/JIIBC.2017.17.2.211

[2] Mi Hwa Song, "A Study on IoT based Real-Time Plants Growth Monitoring for Smart Garden", International Journal of Internet, Broadcasting and Communication(IJIBC), Vol.12, No.1, pp. 130-136, 2020.
DOI: http://dx.doi.org/10.7236/IJIBC.2020.12.1.130

[3] Moonsung Yoo, "Development of experimental water level measuring device using an Arduino and an ultrasonic sensor", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 18, No. 4, pp.143-147, Aug 2018.
DOI: https://doi.org/10.7236/JIIBC.2018.18.4.143

[4] Mahdi H. Miraz, Maaruf Ali, Peter S. Excell, Rich Picking, "A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)", 2015 Internet Technologies & Applications (ITA), p219-224, 2015.

[5] S. Tarkoma and H. Ailisto, "The internet of things program: the Finnish perspective," IEEE Commun. Mag., vol. 51, no. 3, pp. 10-11, Mar. 2013.

[6] Mendez, Diego M., Ioannis Papapanagiotou, and Baijian Yang. "Internet of things: Survey on security and privacy. " arXiv preprint arXiv:1707.01879 (2017).

[7] G. Broll, et al., "Perci: pervasive service interaction with the Internet of things," IEEE Internet Computing, vol. 13, pp. 74-81, Dec. 2009.

[8] IoT Devices, https://www.arm.com/glossary/iot-devices

[9] Arduino, https://www.arduino.cc/

[10] Arduino, https://en.wikipedia.org/wiki/Arduino