Development of Generic Software System Transparent to Any Flexible and Scalable Sensor Array

Ryota Nakamura^{a,*}, Akira Yamawaki^a

^a Kyushu Institute of Technology, Kitakyushu, 804-8550 Japan

*Corresponding Author: yama@ecs.kyutech.ac.jp

Abstract

The sensor arrays receive more attention to make advanced technologies in the future. In conventional sensor arrays, the greater the number of sensors, the greater the number of cables to the local controller, which increases the scale of the device. In addition, the increase in the number of cables is a factor in the complexity of the sensor network, leading to an increase in the difficulty of constructing sensor arrays. To solve this problem, we have proposed an organization method in which controllers and sensors are connected by a daisy chain. This makes it possible to realize a sensor array with a controller and a fixed number of cables, independent of the number of sensors. Furthermore, the proposed sensor array is flexible and scalable enough to be easily customized by the user according to the application. However, this sensor array has an issue with the low versatility of the software of each sensor module. In order to switch the sensors controlled by a sensor module, the software of the sensor module also needs to be switched. In this study, we proposed and developed a new software system for arbitrary sensor arrays in order to improve the versatility of the software. As a preliminary evaluation of the sensor array equipped with the developed software, the relationship between the number of modules and the transfer speed from the modules to the controller was investigated.

Keywords: sensor array, flexible, scalable.

1. Introduction

Sensor arrays are expected to have a wide range of applications in industry, society, and daily life. In order for sensor arrays to become widespread in the future, it will be important to reduce the difficulty of constructing them. Based on these current conditions, we aim to realize a new

method of constructing sensor arrays.

In conventional sensor arrays, the larger the number of sensors, the larger the number of cables to the controller, which controls the entire sensor array, and the larger the scale of the device. In addition, the increase in the number of cables is a factor that increases the complexity of the sensor network, leading to an increase in the difficulty of constructing sensor arrays.

To solve these problems, a conceptual configuration of ultrasonic sensor array with scalability and flexibility to arbitrary objects has been proposed (4). As shown in Fig. 1, an arbitrary number of ultrasonic sensor modules that can adhere to the object to be measured are connected in a daisy chain. Since each module is connected by a flexible cable, the entire system can be adapted to any object of various shapes and sizes. Each module is controlled by a host computer. However, the proposed sensor array has a drawback in the software of the sensor modules. The disadvantage is that only one sensor can be controlled for one software. Therefore, when the type of sensor in the sensor module is changed, the software also needs to be changed. Changing the software of a sensor module every time increases the cost of maintenance and management.

Therefore, in this contribution, we propose a software system for sensor arrays that improves the versatility of the software of sensor modules. In the proposed system, we aim to improve the versatility by hiding the software of sensor modules. With this system, the type of sensor can be changed without changing the software of the sensor module.

In this contribution, we first describe the configuration of the sensor array in the proposed system. Here, we describe the sensor modules that make up the sensor array and the controller that controls them. Next, we describe the outline of operation in the proposed system. The communication between the controller and each sensor module is done by

serial communication. Therefore, when a large number of sensor modules are connected, delay is a concern. Therefore, in this study, as a preliminary evaluation, we examined the relationship between the number of modules and the transfer speed from the modules to the controller.

2. Sensor Array

The schematic of the proposed sensor array is shown in Fig. 2.1. Each sensor module and controller are connected in a daisy-chain fashion. The following sections describe the configuration of the sensor modules and controllers.

2.1 Sensor Module Structure

The structure of the proposed sensor module is shown in Figure 2.2. The proposed sensor module consists of three parts: the CPU, the digital circuit, and the sensor circuit. the CPU controls the sensor circuit and its interaction with the neighboring modules through the digital circuit. The digital circuit is implemented with an interface circuit for communication. The sensor circuit performs the actual measurement. The measurement data is sent to the CPU through the digital circuit. In the proposed system, a monitor

Fig. 2.1. Sensor array

Fig. 2.2. Single module

program is also coded. The monitor program controls the digital circuit and communicates with the sensor circuit according to the data sent from the controller.

2.2 Controller Structure

The structure of the proposed controller is shown in Fig. 2.3. The controller consists of three parts: the CPU, the digital circuit, and the database. the CPU sends commands to the modules to control the sensor array. The CPU sends commands to the modules to control the sensor array, and also obtains from the database the control data for the sensors that operate the monitor program of the sensor modules, and sends them to the sensor modules. In the database, the data necessary to control the sensors used in the sensor array is

Fig. 2.3. Controller

stored.

3. Execution of Proposed Sensor Array

Fig. 3 shows the execution of the proposed sensor array. We explain the operation of the proposed sensor array when the two modules are connected.

Sensor array synchronization in Fig. 3(a): The first step is to synchronize the sensor array. To synchronize the sensor array, a synchronization signal is sent from the controller to the module. The module that receives the synchronization signal sends the same signal to the next module. The synchronization signal then runs once around the sensor array. The synchronization signal also counts up the counter and counts the number of modules as it circles the sensor array.

Selecting the software for the module in Fig. 3(b): When the synchronization of the sensor array is completed, the sensor to be used in the sensor array is selected. The input for selection is done through the user interface of the

controller. For example, if sensor A is used in the sensor circuit of the sensor module, select sensor A via the user interface of the controller.

Sending and receiving commands in Fig. 3(c): When the sensor to be used is selected, the controller sends a command to the module. There are three commands: `command_num`, `recv_data_num`, and `sensor_command`. `command_num` indicates the data capacity of `sensor_command`. `sensor_command` data capacity is variable, so it must be indicated by `command_num`. Since the data capacity of `sensor_command` is variable, it needs to be indicated by `command_num`. `recv_data_num` indicates the data capacity of measurement data. Since the data capacity

to be measured and acquired is different for each type of sensor, it needs to be indicated by `recv_data_num`. `sensor_command` is a command used to execute the monitor program of the sensor module. one command can communicate with the sensor circuit once. This realizes communication with the sensor circuit with multiple commands. The module that receives these commands sends the same command to the next module.

Measurement by module in Fig. 3(d): The module sends the received `sensor_command` to the CPU, which uses the `sensor_command` to execute the monitor program. The CPU uses the `sensor_command` to execute the monitor program, which controls the digital circuit, communicates

Fig. 3. Execution snapshot of proposed sensor array

with the sensor circuit, and performs the measurement. When the monitor program is executed, it controls the digital circuit to communicate with the sensor circuit to perform measurement, and then receives measurement data from the sensor circuit to complete the measurement.

Module sending module data in Fig. 3(e): When module 1 completes the measurement, it sends its own measurement data to module 2. When module 2 receives the measurement data, it adds its own measurement data to the received measurement data and sends it to the controller. In this way, the controller can receive the measurement data of the entire sensor array.

4. Experiment and Discussion

The developed sensor module and controller are shown in Fig. 4.1. The digital circuitry and CPU of the sensor module uses FPGA. The sensor is a BME280 that can measure temperature, humidity, and air pressure. The controller is a Raspberry pi. And MySQL is used as the database.

Fig. 4.1. Designed sensor module and controller

4.1 Operation Verification

Figure 4.2 shows the results of the operation verification of the developed sensor array. The selection of the sensor is done by the CUI of the controller. To measure temperature,

humidity, and barometric pressure, enter "1". To measure only temperature, enter "2". To measure only humidity, enter "3". Input "4" to measure the air pressure. (a) shows the result of selecting the temperature sensor in the controller's CUI. (b) shows the result of selecting the humidity sensor in the controller's CUI. From the figure, we can see that the sensor used in the sensor array is dynamically switched by selection in the controller. In this experiment, we also verified the operation by connecting a maximum of four sensor modules.

Figure 4.3 shows the measurement result table of the database implemented in the controller. The figure shows that the measurement results of the sensor array can be managed in the database.

```
----- mesurment_section -----
Please enter the type of sensor !!
1 : BME280
2 : BME280 TMP
3 : BME280 HUM
4 : BME280 PRES
2
----- module 1 -----
temp : 25.75 °C
----- module 2 -----
temp : 25.82 °C
```

(a) Temperature sensor

```
Please enter the type of sensor !!
1 : BME280
2 : BME280 TMP
3 : BME280 HUM
4 : BME280 PRES
3
----- module 1 -----
hum : 42.03 %
----- module 2 -----
hum : 42.89 %
```

(b) Humidity sensor

Fig. 4.2. Results of operation verification

MariaDB [homestead]> select * from sensor_data;						
	id	sensor_id	module_number	sensor_type	sensor_data	created_at
1	1	1	0	0	17.78	2020-12-09 09:12:3
2	1	1	0	10	1024.02	2020-12-09 09:12:3
3	1	1	0	20	39.33	2020-12-09 09:12:3
4	1	1	0	0	17.87	2020-12-09 09:12:5
5	1	1	0	10	1023.96	2020-12-09 09:12:5
6	1	1	0	20	39.34	2020-12-09 09:12:5
7	1	1	0	0	18.40	2020-12-09 09:24:3
8	1	1	0	10	1023.81	2020-12-09 09:24:3
9	1	1	0	20	42.70	2020-12-09 09:24:3
10	1	1	1	0	19.07	2020-12-09 09:24:3
11	1	1	1	10	1024.02	2020-12-09 09:24:3
12	1	1	1	20	38.36	2020-12-09 09:24:3
13	1	1	0	0	18.46	2020-12-09 09:25:0
14	1	1	0	10	1023.78	2020-12-09 09:25:0

Fig. 4.3. Measurement data table

Fig. 4.4. Gun chart of operating time measurement

4.2 Measurement of the operating time of the module

In the designed module, we measured the time taken for each operation. Fig.4.4. shows a gun chart that shows each operating time of module in a sensor array consisting of two modules with a baud rate of 100[Kbps]. The yellow part is the time it takes to send the command_num and the green part is the time it takes to send the recv_data_num, blue part is the time it takes to send the sensor_command. These times are 1.4[ms]. The orange and light green parts are time it takes to send the data processed by each module. It takes 0.8[ms] to send one module data. The red part is the time that the module measures by driving the sensor. The time to measure takes 15.1[ms] Transmission to the next module is not possible until this measurement is finished.

In the chart, the time taken to communicate between modules increases sequentially when the command turns back. Therefore, as the number of modules connected increases, the operating time of the sensor array increases sequentially.

4.3 Evaluation of the operation time in the sensor array

Fig.4.5. shows the relationship between operating time and number of modules in the sensor array. The more connected modules, the more time is needed to send the command and collect module data throughout the sensor array. The increase is sequential, which is caused by the sequential increase in the communication time for the measurement data. Also, the higher the baud rate between modules, the shorter the communication time for the whole sensor array. For example, in the case of the sensor array of 250 connected modules, at a baud rate of 50[Mbps] between

modules, the operating time is 0.09[s], whereas at a baud rate of 1[Mbps] between modules, the operating time is 3.79[s]. As a large percentage of the sensor array operation time is accounted for by the communication time between modules, the effect of the communication speed between modules has a significant impact on the operation time.

Fig. 4.5. Relationship between the number of modules and operating time in the sensor array

5. Conclusions

In this study, we proposed and developed a new software system for arbitrary sensor arrays in order to improve the versatility of sensor array software. As a preliminary evaluation of the sensor array with the developed software, the relationship between the number of modules and the transfer rate from the modules to the controller was investigated. As a result of the operation verification, it was possible to dynamically switch the software of the sensor modules by selecting the sensors to be used in the sensor array controller. Since the transfer rate between modules has

a significant effect on the operation time of the sensor array, it is necessary to set the communication rate between modules according to the scale of the sensor array device.

As future work, we will propose and develop a method for dynamically switching the hardware part of the sensor module, such as the digital circuit and sensor circuit.

[Click here](#) and insert your conclusions text.

References

- (1) Anshuman Bhuyan, Jung Woo Choe, Byung Chul Lee, et al (2013) Integrated Circuits for Volumetric Ultrasound Imaging With 2-D CMUT Arrays, IEEE Transactions on Biomedical Circuits and Systems, Volume: 7, Issue: 6, pp.796 -804, 2013
- (2) Yuichi Morita, Kono Sota, and Akira Yamawaki: “Proposal of an Ultrasonic Sensor Array with Flexible and Scalable Organization”, Proceedings of the Twenty-Third International Symposium on Artificial Life and Robotics 2018, pp. 381-384, 2018
- (3) Yuichi Morita, Akira Yamawaki (2018), Design of Daisy-chain Connection for Flexible and Scalable Ultrasonic Sensor Array, Proceedings of the 6th IIAE International Conference on Intelligent Systems and Image Processing 2018, pp 132-136, 2018