Special issue on International Workshop on Networked Sensing System

Trans. of the Society of Instrument and Control Engineers
Vol.E-S-1, No.1, 2/10 (2006)

Realization of Bluetooth-equipped Device for Wireless Sensor Network[†]

Tatsuya HARADA *, Takaharu NAGAI *,
Taketoshi MORI ** and Tomomasa SATO *

In this paper, we realized a portable Bluetooth-equipped device for the wireless sensor network. This device is designed to be able to connect various sensors and control external devices. Bluetooth has so nice features such as low power consumption, sufficient data throughput and high resistance to noise that we utilize Bluetooth as a basis of the device for the wireless sensor network. We implement not only the ad hoc network to realize a small network but also the scatternet to construct a large scale network by connecting and switching the small network. We demonstrate applications using the device for wearable sensors, environmental information sensors and controllers of home electric appliances. These demonstrations prove that the realized device has sufficient performance for the wireless sensor network in the daily life.

Key Words: Scatternet, Ad hoc, Wearable sensor, Home electric appliance, Environmental sensor

1. Introduction

Computers become more compact and more powerful according to the Moore's low. At one time, many people shared one host computer. After that, the era of the personal computer that one person had his or her own computer came. Recently, the audiovisual goods and the home electric appliances include high power microprocessors and network functions. In the near future, because of the more price reduction and the more downsizing of computers, the ubiquitous computing that the networked huge amount of computers are distributed in the daily life will come before long.

In ubiquitous computing, many sensors and actuators are thought to be distributed in daily life for supporting humans by measuring and accumulating their behavior and environmental information. Under this situation, it is necessary to construct the sensor networks for measuring sensor information effectively and controlling the many actuators concertedly. Moreover since this sensor network will be spread over the daily life environment, it is also necessary for these sensors to be connected wirelessly in order that anyone can install them easily.

Smart Dust[1)] is the famous device for the wireless sensor network that is millimeter scale sensing. Mote was developed by improving Smart Dust utilizing commercially offered parts[2)]. Tiny OS[3)] that realized the effective event processing under small amount of memory and low computing power was implemented to Mote. After the completion of Smart Dust project in 2001, Mote was commercialized by Crossbow technology as MICA Mote. PicoRadio[4)] that consists of low power and low cost PicoNodes aims to support the assembly of ad hoc wireless sensor network. Medusa[5)] is also node device for wireless sensor network that has similar features as MICA nodes. μAMP Project[6)] develops sensor nodes emphasis on energy efficiency. Kawahara et al. also developed the wireless sensor network devices U^3 for the daily life[7)].

In our research, with the daily life use in mind, we develop a small device which can construct the wireless sensor network and implement the networking protocol that can realize both a small ad hoc network and a large network by connecting small networks with each other. Related researches mainly focus on power efficiency. Although power efficiency is very important for the daily life use, the affinity to many kinds of devices is also important. In addition, monitoring the human motion is a necessary technique for the human centered computing. Since the amount of human motion data is very large, it is necessary to keep sufficient data throughput for transmitting

† Presented at INSS2004 (2004.6)
* Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo Japan
** Graduate School of Interdisciplinary Information Studies, The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo Japan
(Received November 1, 2004)
(Revised April 11, 2005)

the motion data. Considering the power consumption, the data throughput and the affinity, we select Bluetooth as a basis of the device for the wireless sensor network. Many researches utilize the frequency band such as 900 [MHz] or 300 [MHz]. Since the available frequency is different among each country in these frequency bands, it is difficult to deal with wireless communication parts uniformly. However Bluetooth can be used all over the world, because it uses 2.4 [GHz].

Apart from researches about developing the wireless sensor devices, managing methods for the large network are also actively studied. Bulusu[8] discussed about the connection between the sensor network and the internet. Our research target does not include the construction of such large network.

We also demonstrate applications using the realized device for wearable sensors, environmental information sensors and controllers of home electric appliances. By these demonstrations, we will verify the effectiveness of the realized device for the wireless sensor network in the daily life.

Fig. 1 shows the realized device for the wireless sensor network next to a wristwatch for scale.

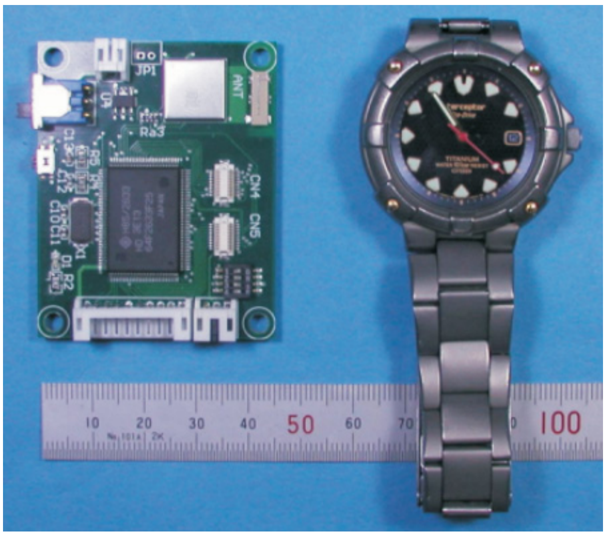

Fig. 1 Overview of realized Bluetooth-equipped device for wireless sensor network (left).

2. Bluetooth-equipped Device for Wireless Sensor Network

2.1 Necessary Function

For the introduction of the wireless sensor network to the daily life, it is necessary to avoid the interference with other wireless devices. As for data throughput, its necessary performance depends on the application. We decide the necessary performance on the assumption of the human motion monitoring. When the human motion is monitored as 30 degree of freedom, 10 [bits] resolution and 100 [Hz] frequency, 30 [kbps] data throughput is needed. It takes up to 300 [kbps] for the throughput, if multiple humans are monitored. Node devices should be so small as to wear them and install them to everywhere. In case of going out, the battery should have a life of at least 8 hours.

2.2 Bluetooth

Bluetooth uses frequency-hopping spread spectrum that switches the frequency every 625 [μsec] and uses Industrial Scientific and Medical band, which is free to use in many countries. It has high tolerance for noise. These features are suitable for daily use. Its throughput is 723.2 [kbps] at an asymmetric link and is 433.9 [kbps] at a symmetric link. This is also sufficient performance for the daily life applications.

Comparing with IEEE 802.11b (Wireless LAN), throughput of Bluetooth is low. However Bluetooth has superior performance in the power consumption and the cost. Ordinary Li-ion battery for cell phones is about 3.7 [V] 600 [mAh]. Since IEEE 802.11b needs 200 [mA], a battery has a life of only 3 hours. On the other hand, Bluetooth needs up to 30 [mA], the battery has a life of 20 hours.

Bluetooth does not need the access points or base stations. Bluetooth supports ad hoc network and Scatternet which realizes multihop networks. We can realize a large scale network by connecting small networks based on Bluetooth.

2.3 Microprocessor

A microprocessor must have many channels for analog inputs and the serial interfaces for sensor inputs and controlling the external devices. The microprocessor should process both a network and sensor measurement. We adopt H8S/2633 (Renesas) as the microprocessor which has sufficient internal memories (ROM 256 [kbytes], RAM 16 [kbytes]), 16 analog input channels with 10 [bits] resolution, many serial interfaces and many interrupts. 8 [Mbit] SRAM is connected to the microprocessor in case of handling a large amount of data.

2.4 Protocol Stack and Implementation

Host Controller Interface (HCI) is a standard for the communication and the control defined by Bluetooth SIG. Since the radio control and the process for the physical layer of Bluetooth are processed by the Bluetooth module, we only implement the Host Controller Interface and the application such as the logical communication process and sensor data process. We utilize MBM02 (Hitachi Maxell) as the Bluetooth module which is based on Bluetooth 1.1.

The Bluetooth module and the microprocessor are connected through UART serial communication. To control the Bluetooth module, it is necessary for the microprocessor to handle all the data without the loss of data and the delay. Moreover the Microprocessor should measure and accumulate sensor data. Coping with these tasks, we used transmitting and receiving interrupt and 1 [kbyte] ring buffers without specific operating systems.

2.5 Realization of Device

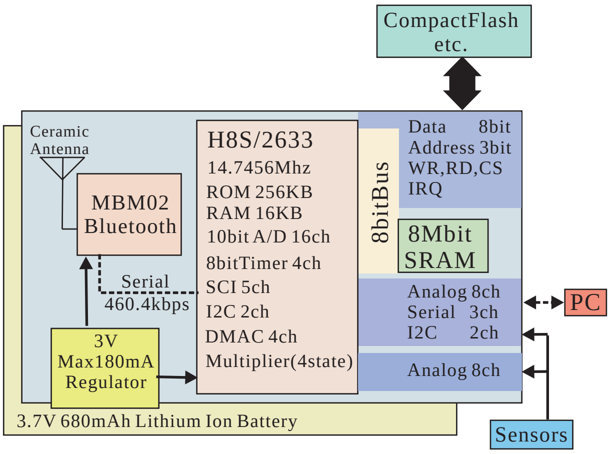

Fig. 2 Configuration of the realized device. The Bluetooth module and the microprocessor are connected UART. The microprocessor is selected so that many sensors, actuators and external storages can be connected to the realized device.

Fig. 2 shows the configuration of the realized device. The microprocessor and the Bluetooth module are connected by 460.8 [kbps] serial communication. All the devices are driven by 3 [V] and can be driven by only the Li-ion battery. To cope with many kind of sensors, the reference voltage of the analog input can be changed independent of the 3 [V] power supply. Maximum current consumption is up to 90 [mA] (H8S/2633 30 [mA], SRAM 25 [mA], MBM02 30 [mA] and others 5 [mA]). The realize device has 16 analog input channels, 3 serial communication channels, two I_2C channels and the 8 [bit] CPU bus which enables connection with an external storage such as a compact flash. **Fig. 3** shows the realized device. This device has 4 layers. Its size is 45 × 56 [mm]. Its weight is 14 [g] excluding a battery weight.

3. Networking

In order to construct a large ad hoc network, we implement a one-to-one connection, an autonomous connection and a scatternet to the wireless sensor network devices.

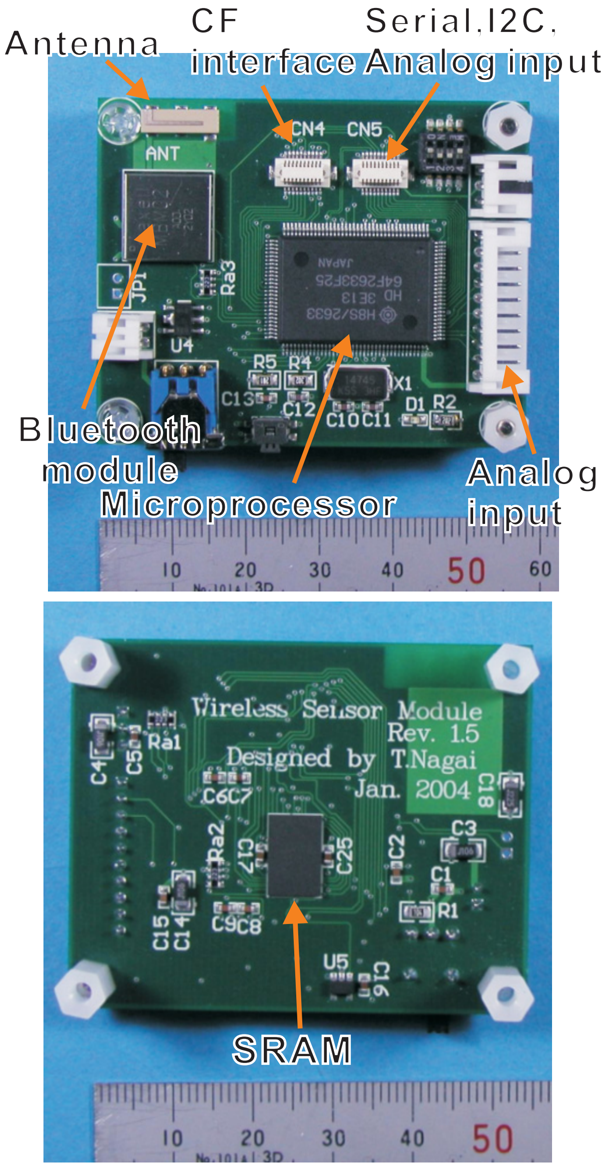

Fig. 3 Top (top figure) and bottom (bottom figure) layer of the realized device. The Bluetooth module (MBM02) and the microprocessor (H8S/2633) are mounted on the top layer. The 8 Mbit SRAM is mounted on the bottom layer.

3.1 One to One Connection

At first, the one-to-one connection was constructed as the basis of the wireless sensor network. In order to build the connection, the basic process is 1) obtaining the target's information by inquiring, 2) paging the target and 3) establishing the connection.

In this section, we consider only two sensor devices A and B. Firstly, both A and B devices are initialized by executing "Reset Command" to enter the standby mode. Secondly, the device B enables the inquiry and page scan by executing the "Write Scan Enable Command". The device A is inquiring targets for some length of time by

"Inquiry Command". The device B responds to the Inquiry Command from the device A automatically. "Inquiry Result Event" occurs in the device A that receives the response of the device B. The device A can obtain the Bluetooth address and the displacement of the clock of the device B as the parameters of "Command Complete Event". Thirdly, the device A tries to connect the device B by executing "Create Connection Command" with obtained information from the device B. "Connection Request Event" occurs in the device B that receives paging from the device A. The device B can obtain the Bluetooth address and the class of device of the device A at this time. In order to accept the connection, the device B executes "Accept Connection Request Command". Finally, both A and B devices complete the connection with "Connection Complete Event" and can obtain the connection handle. After above process, they are ready to communicate.

3. 2 Autonomous Connection

We implement the autonomous connection. The device searches other devices by inquiring at fixed intervals. If there are responses from other devices, the device autonomously tries to connect the responded devices. **Fig. 4** shows a frame format of the autonomous connection. The inquiring device will be a master. The responded device will be a slave.

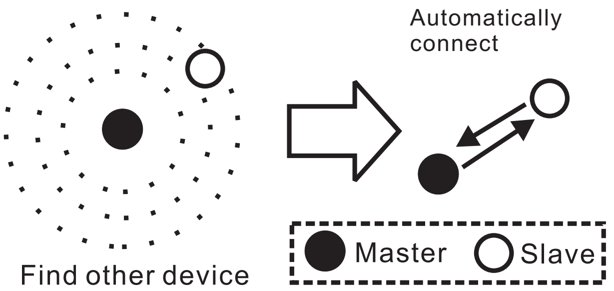

Fig. 4 Autonomous establishment of wireless network. The device searches the specific devices by inquiring at fixed intervals. If the device finds the specific device, the device autonomously tries to connect the responded devices. The inquiring device will be a master. The responded device will be a slave.

Each device set the class as "Class of Device" for each role of the device by using "Write Class of Device Command". Each device also enables inquiry and page scan with "Write Scan Enable Command". The devices utilize "Event Filter" to find and connect the specific device automatically. After these settings, the device executes "Periodic Inquiry command" to find the specific device actually. If the specific device is found, the device executes "Exit Periodic Inquiry Command" to stop inquiring. Finally, the device executes "Create Connection Command" to connect the specific device. Through above process the autonomous connection can be established.

3. 3 Scatternet

The network consists of one master and more than one slave in Bluetooth. Although the master controls the communication frequency, the number of slave devices is limited up to 7 devices. This small network is called a piconet. In order to construct a larger network, the piconets should be connected with each other (**Fig. 5**). The large network that consists of the piconets is called scatternet. We realize the scatternet by switching the master device to the slave in another piconet temporary. Based on the scatternet, the huge network can be constructed by relaying piconets.

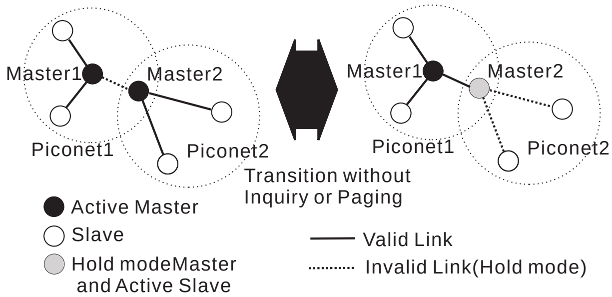

Fig. 5 Scatternet with hold mode. The scatternet is realized by switching the master device to the slave in another piconet temporary.

We implemented scatternet by using "Hold mode". Under the hold mode, the device can switch between the master and the slave. In this section, we consider three devices A, B and C.

Firstly, there is a piconet which includes the master device A and the slave device B. Both devices A and B execute "Write Link Policy Setting Command" to allow themselves to change the hold mode. The master device A executes "Hold Mode Command" and sets the period of the hold mode. After the "Mode Change Event", the device A is transferred to the hold mode. In the hold mode, the master device A executes "Write Scan Enable Command" to enable the inquiry and page scan.

Secondly, the third device C requests the connection to the device A. "Connection Request Event" occurs in the device A. After the event, the device A executes "Accept Connection Request Command" to allow the connection from the device C. "Connection Complete Event" occurs in the device C. After that, the second piconet is constructed which includes the master device C and the slave device A.

Thirdly, in the second piconet, the master and the slave devices execute "Write Link Policy Command" to allow themselves to change the hold mode. The device A in the second piconet executes "Hold Mode Command" and sets the period of the hold mode. "Mode Change Event" occurs in the devices A and C. The device A in the second piconet changes into the hold mode. The period of the hold mode of the device A in the first piconet finishes. "Mode Change Event" occurs in the Devices A and B.

Finally, the device A executes "Hold Mode Command" and sets the period of the hold mode before the end of the period of the hold mode in the second piconet.

Moreover the slave device in the second piconet can become the hold mode master of the third piconet. This connection can be beaded repeatedly. By using this procedure, a huge network can be realized in **Fig. 6**. However the routing in this huge network is the big problem. There are many papers to attack the routing problem to find the optimal path. This paper does not mention the routing problem in this paper, since this is our on-going work.

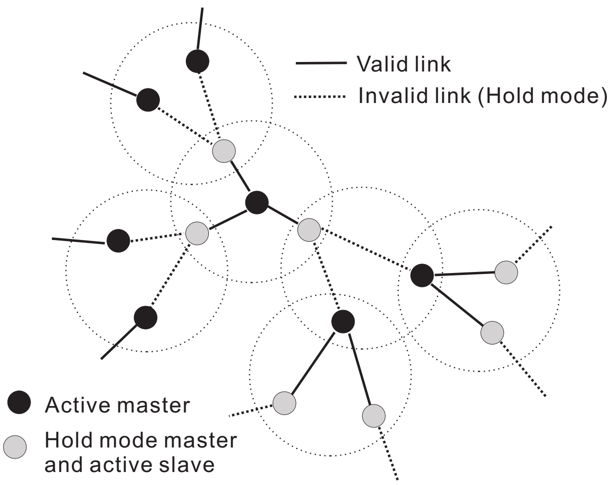

Fig. 6 Huge sensor network. The huge network can be realized by connecting piconets with hold mode.

3.4 Time Synchronization

In order to gather sensor data from many kinds of devices, it is necessary to synchronize the time of sensor data. The time synchronization procedure is explained as follows:

(1) One device becomes the master to be synchronized.

(2) Other devices are connected as the slaves.

(3) The master device broadcasts packets for the time synchronization.

(4) The entire device which accepts the broadcasted packet starts the timer.

Since all the device has the same hardware, the processing time for receiving the packet is assumed same. The time of flight of the radio wave is ignored because the length of the communication in Bluetooth is about 10 m.

4. Performance

4.1 Environment and Condition

We conducted all the experiments under the usual office environment. There were active IEEE802.11b wireless LAN access points. During the experiments, there were no communication errors and disconnections. However we did not measure the interference between the wireless LAN and the Bluetooth. We tested two different communication distances (1.5 [m] and 4.0 [mm]), then we obtained the same results. By changing some Bluetooth devices, we confirmed that there were no individual differences.

4.2 Time to Find Target Device

The time to start measuring the sensor data from the start of the power supply is very important for usability. The time to obtain the information of the target device from the reset of the device is evaluated. The evaluation method is explained as follows:

(1) The device B is set as the inquiry scan enable mode with "Write Scan Enable Command".

(2) The device A starts the timer after execution of "Reset Command".

(3) The device A executes "Inquiry Command".

(4) The device A stops the timer immediately when it obtains the information of the device B through the "Inquiry Result Event".

This procedure is repeated for about 70 times. The timer is counted by using the 8192 diving of the clock (14.7456 MHz) of the microprocessor.

The response time is changed with respect to each 15 or 16 measurement. The inquiring device checks 16 different frequencies 256 times every 10 [msec]. After that (after 2.56 [sec]), the inquiring device changed the inquiring hopping sequence. On the other hand, the inquired device changes the frequency every 1.28 [sec]. The alternate of the finding time is thought to be caused by the combination of the frequency patterns. Although there is variety of the response time, the device can find the target device in 5 seconds.

4.3 Time to Establish Communication

We evaluate the communication establishment time from the inquiry to the start of the data transmission. This measurement time includes the request of connection and the connection establishment. We evaluate 75 trials continuously. **Fig. 7** shows the experimental result. The communication establishment time is about one second or three seconds. The alternate of the establishment time is also thought to be caused by the combination of the frequency patterns.

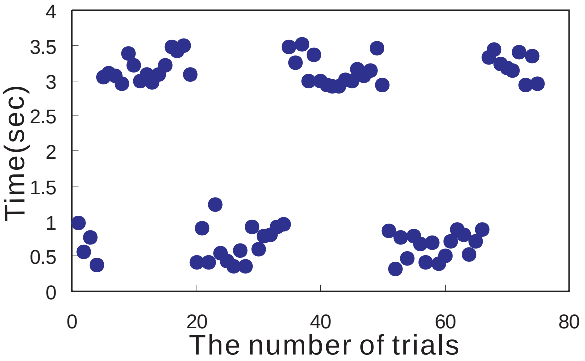

Fig. 7 The time required to establish a connection. The connection establishment time is changed with respect to each 15 or 16 measurement. The alternate of the establishment time is thought to be caused by the combination of the frequency patterns. The communication establishment time is within three seconds.

4.4 Data Throughput

We evaluate the data throughput by measuring the communication time from the start of data reception to the end of reception. The transmitted data consists of 10 data packets which size is 384 [bytes] (the maximum packet size supported by the Bluetooth chip). Since the receive buffer overflows when the more than 12 packets are successively transmitted, we decide 10 data packets. The mean throughput is 320.85 [kbps] and the standard deviation is 29.09 by 5 trials. Because the throughput between the microprocessor and the Bluetooth module is 460.8 [kbps], this result is appropriate. This throughput is sufficient performance for the assumed application. In order to obtain the maximum data throughput 723.2 [kbps], it is necessary to speed up the communication between the microprocessor and the Bluetooth module.

4.5 Communication Delay

We evaluate the communication delay by measuring the time from the start of transmission of one packet to the end of reception. Firstly, two devices B and C connect the master device A as the slaves. Secondly, the device A broadcasts a timer start command to the device B and C in order to synchronize the timer of B and C. Thirdly both B and C start the timer and disconnect A. After that, B connects C and stops its timer when B starts to transmit the packet. C stops its timer when C finishes the packet reception. Finally, the communication delay of one packet is evaluated by the difference between the timer counter of B and C.

The communication delay is measured by changing the size of the payload. The evaluated payload sizes are 1, 10, 20, 30, 100, 200, 300 and 384 [bytes]. The means of the delay in 5 trials are plotted in the **Fig. 8**.

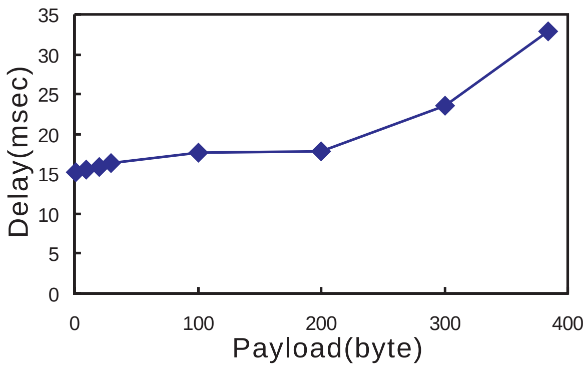

Fig. 8 Relation between the delay and payload. The delay of sending packets is within 35 milli second under 384 [bytes].

In the wireless communication, the number of time slots can be selected among 1, 3 or 5. The maximum payload is set to 27, 183 and 339 [bytes] for each time slot. When the payload size is changed from 20 to 30, or 100 to 200 [bytes], the communication delay is predicted to be changed significantly. The results show that this significant change is not observed. The delay of the process in the Bluetooth chip is thought to have more influence than the delay of the wireless communication.

4.6 Battery Life

We evaluate the battery life of the realized device by using the Li-Ion battery (Voltage: 3.7 [V], capacity: 680 [mAh], size: $33 \times 50 \times 4.7$ [mm] and weight: 16 [g]). We evaluate the battery life by measuring the duration time in the continuous one-to-one data transmission. When the receiver is driven by the battery, the battery life is 9 hours and 20 minutes. On the other hand, when the transmitter is driven by the battery, the battery life is 9 hours and 10 minutes. Generally, in the wireless communication, the transmitter needs more power than the receiver. Since in Bluetooth, the receiver returns the report of no-error to the transmitter, the results of the battery life is thought to have no difference between the transmitter and the receiver.

5. Application

By using the realized devices and the ad hoc network functions, we construct the application that has automatically exchanging name cards, controlling the heater when returning from the cold outside and turning on and off the desk light depend on the brightness of the user's hand.

Fig. 9 shows the configuration of the application system.

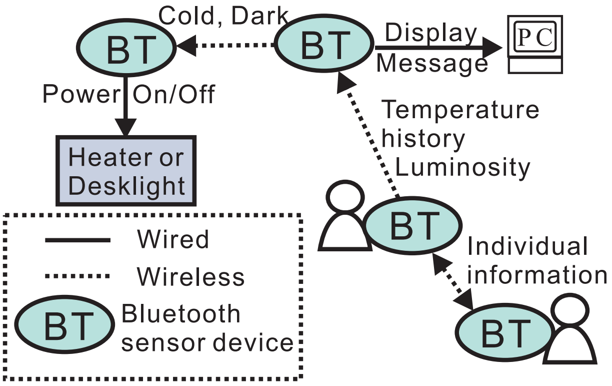

Fig. 9 Configuration of application. This system consists of wearable devices, PC connection devices and electric appliance control devices.

The system consists of the following three kinds of additional sensor devices which are connected to the realized wireless sensor network device.

5.1 Wearable Device

This device is attached to the wrist and has a temperature sensor, illuminometer and their amplification circuits which are connected to the analog input of the realized device. This device searches another wearable device and tries to connect the found wearable device. If the connection is established, the name card information is exchanged. When this device is requested by the PC connection devices, it connects the PC connection device and transmits the accumulated name card information, the history of the temperature and the intensity of illumination. The detailed procedure is explained as follows.

This device has name card information. This device sets the "Class of Device" as the wearable device with "Write Class of Device Command". This device measures the temperature every 30 seconds. The device enables the inquiry and page scan with "Write Scan Enable Command". This device searches the devices at fixed intervals with "Periodic Inquiry Command".

If it finds the wearable device with "Inquiry Result Event" and "Class of Device", the device tries to connect it by using "Create Connection Command". After the completion of the connection, the device transmits the name card information to the other wearable device and receives the other's name card information. If the device finishes the transmission with "Number of Completed Packets Event" and the reception, the device returns to the status before the connection with "Disconnect Command". When "Connect Request Event" occurs in the device and "Class of Device" is the wearable device, the device accepts the connection with "Accept Connection Request Command". After the "Connection Complete Event", the name card information is exchanged.

If the connection request device is detected as the PC connection device with "Class of Device", the device accepts the connection with "Accept Connection Request Command". After the completion of the connection, the device transmits the accumulated temperature data and the name card information. This device also measures the intensity of illumination every 5 seconds, this intensity information is also transmitted to the PC connection device.

5.2 PC Connection Device

This device is connected to the PC through the serial communication interface. This device can display the information via the connected PC. This device searches electric appliance control devices and tries to connect them on finding them. If this device finds the wearable devices, it tries to connect the wearable devices and displays the received information as the name card information. Moreover if it detects that the human wearing the device returns from a cold outside, this device tries to control the device that is connected to the heater. By the same token, the device detects the darkness, it tries to transmit turning on command to the device that is connected to the desk light. The detailed procedure is explained as follows.

The PC connection device sets "Class of Device" as the PC connection device with "Write Class of Device Command". This device starts to search the devices at fixed intervals with "Periodic Inquiry Command". If the detected device is the electric appliance control device, this device tries to connect it with "Create Connection Command".

If the detected device is the wearable device with "Class of Device", the device connects it and sends the message to PC display as "Hi, [Owner's name of the wearable device].". After "Connection Complete Event", the accumulated temperature data is received from the wearable device. According to the temperature data, the device decides whether the device transmits the control packets to the electric appliance control device of the air conditioner. The device also sends the message to the PC display according to the temperature as "It is very cold. So I will turn on the heater.". After the temperature data, the wearable device transmits the name card information. The device sends the name card information to the PC display. The device also decides whether it sends the control packet to the electric appliance control device of the light according to the intensity of illumination from the wearable device. After sending the control packets, the device sends the message to the PC display as "It is too dark. I will turn on the light.".

5.3 Electric Appliance Control Device

This device is connected to the electric appliances and controlled by the PC connection device. In this application, this device is connected to the heater and the desk light. This device has relays which are connected by the DA port of the wireless sensor device. The detailed procedure is explained as follows.

The electric appliance control device sets the "Class of Device" as the electric appliance control device. This device becomes the inquiry and page scan enable status with "Write Scan Enable Command". It the connection requested device is the PC control device, the device accepts the connection request with "Accept Connection Request".

According to the received control packets from the PC connection device, the device controls the heater and the desk light.

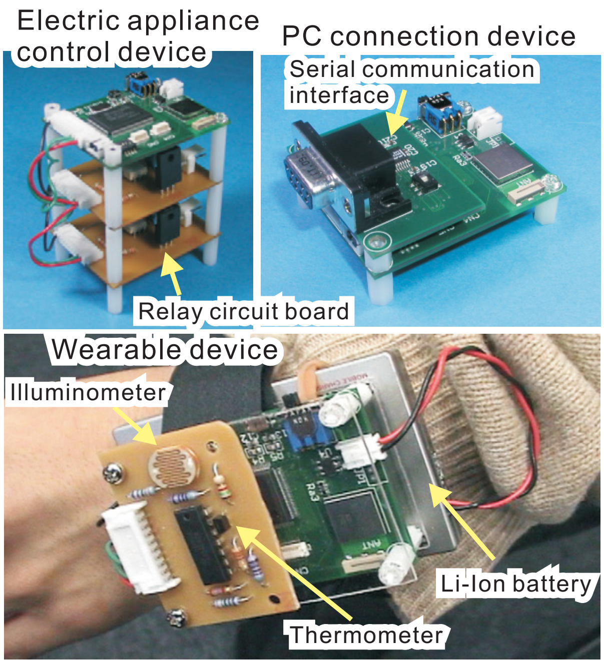

Fig. 10 Top left is the electric appliance control device. Top right is the PC connection device. Bottom is the wearable device.

Under the above configuration, the subject meets the guest and returns to his dark office through the cold outside. Then PC displays the name card information of the guest. The heater and the desk light come on. The wireless sensor network works well as we expected. The effectiveness of the realized device and the network can be

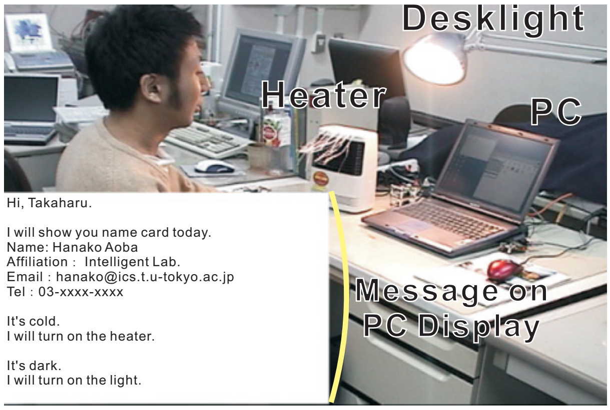

Fig. 11 A heater and a desk light are turned on by the electric appliance control device. Messages are displayed from the device connected to PC. These results depend on the information from the wearable device.

demonstrated. **Fig. 10** shows these devices which were used in this application. **Fig. 11** shows the scene of the demonstration.

6. Conclusion

In this paper, we realized a portable Bluetooth-equipped device for the wireless sensor network. This device is designed to be able to connect various sensors and control external devices. We implement not only the ad hoc network to realize a small network but also the scatternet to construct a large scale network by connecting and switching the small network. We evaluate the throughput, the communication delay and the power consumption. We also demonstrate applications using the devices for wearable sensors, environmental information sensors and controllers home electric appliances. These evaluations and demonstrations prove that the realized device has sufficient performance for the wireless sensor network and effectiveness in the daily life.

Construction of the large network using the scatternet, solving communication routing problem, implementing the power consumption control and realizing the human motion measurement application are future works.

References

1) J. M. Kahn, R. H. Katz, and K. S. J. Pister: Emerging Challenges: Mobile Networking for Smart Dust, J. of Commun. and Networks, vol. 2, no. 3, pp. 188-196, September (2000)
2) S. Hollar: COTS Dust, Master's thesis, University of California, Berkeley, (2000)
3) J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister: System Architecture Directions for Networked Sensors, In Proceedings of the ninth international conference on Architectural Support for Programming Languages and Operating Systems, pp. 93-104, (2000)

4) J. Rabaey, J. Ammer, J.L. da Silva Jr., and D. Patel: PicoRadio: Ad-Hoc Wireless Networking of Ubiquitous Low-Energy Sensor/Monitor Nodes, IEEE Computer Society Annual Workshop on VLSI (WVLSI'00), pp. 9-14, (2000)
5) A. Savvides and M. B. Srivastava: A Distributed Computation Platform for Wireless Embedded Sensing, Proceedings of International Conference on Computer Design (ICCD), Freiburg, Germany, pp.220-225, September (2002)
6) R. Min, M. Bhardwaj, S. H. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan: An Architecture for a Power-Aware Distributed Microsensor Node, IEEE Workshop on Signal Processing Systems (SiPS '00), pp. 581-590, October (2000)
7) Y. Kawahara, M. Minami, H. Morikawa, and T. Aoyama: Design and Implementation of a Sensor Network Node for Ubiquitous Computing Environment, IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall) ,10 (2003)
8) N. Bulusu, D. Estrin, L. Girod, and J. Heidemann: Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems, In Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA '01), Ambleside, Lake District, UK. July 15-20, (2001)

Tatsuya Harada (Member)

He received his M.E. and Ph.D. degrees in Mechanical Engineering from the University of Tokyo, in 1998 and 2001, respectively. From 2000 to 2001, he was a Research Fellow of the Japan Society for the Promotion Science. From 2001, he is a Research Associate at Graduate School of Information Science and Technology, the University of Tokyo. His research interests include human motion observation, modeling, perception, sensors, and embedded systems. Ph.D. Harada is an active member of the IEEE, the Robotics Society of Japan, and the Society of Instrument and Control Engineers.

Takaharu Nagai

He received his B.E. degree in Mechano-Informatics from the University of Tokyo, in 2004. From 2004, he is a master course student at Graduate School of Information Science and Technology, the University of Tokyo. His research interests include sensors, embedded systems, and micro electro mechanical system.

Taketoshi Mori (Member)

He received his Ph.D. in Information Engineering from the University of Tokyo, in 1995. From 1995 to 1998, he was a Research Associate at Research Center for Advanced Science and Technology of the University of Tokyo. From 1998, he was an Assistant Professor at the university. Now, he is an Associate Professor at the university. His research interests include recognition, robot vision, image processing, and pervasive computing. Ph.D. Mori is an active member of the IEEE Robotics and Automation Society, the IEEE Computer Society, the Robotics Society of Japan, the Japan Society of Mechanical Engineers, the Institute of Electronics, Information and Communication Engineers, Japanese Society for Artificial Intelligence, and the Society of Instrument and Control Engineers.

Tomomasa Sato (Member)

He received the B.S., M.S. and Ph.D. degrees in mechanical engineering from the University of Tokyo, Japan, in 1971, 1973 and 1976 respectively. Since 1976, he has been with the Electrotechnical Laboratory (ETL) of the Ministry of Industrial Science and Technology. In 1991, he moved to the Research Center for Advanced Science and Technology (RCAST) of the University of Tokyo. From 1998, he is a Professor of the Department of Mechano-Informatics of the university. His current research interests include intelligent machine, human symbiosis robot and environmental type robot systems. Ph.D. Sato is an active member of the IEEE Robotics and Automation Society, the Japan Society of Mechanical Engineers, the Society of Instrument and Control Engineers, the Japanese Society for Artificial Intelligence, and the Robotics Society of Japan.

Reprinted from Trans. of the SICE
Vol. E-S-1 No. 1 2/10 2006