

Microcomputer-based Acquisition of Atmospheric Pressure Data via a Sensor-equipped Remote Monitoring System and Related Application to Education in Electrical Engineering

Kouji SHIBATA[†], Kazuma HANADA^{††} and Hidehiro SEKI^{†††}

ABSTRACT

A previous paper by the authors reported on the development of an ultra-small, low-power, economical sensor-equipped remote monitoring system featuring low operational cost and stable operability. The system incorporates an ARM-based microcomputer (Raspberry Pi) running on Linux OS and a USB-connected mobile broadband modem to enable 4G cellular network connection combination. It can be operated with a private IP address via VPN for easy negotiation of NAT and firewall protection thanks to the installation of an embedded VPN program on the microcomputer. Studies by the authors have also shown that camera images and information on temperature/humidity from sensor-remote locations can be securely acquired through a web browser using smart devices such as smartphones and tablet PCs by connecting a commercially available and economical USB web camera and a temperature/humidity sensor to the system. The proposed system can be set up anywhere and provides an uninterrupted connection for remote monitoring. Always-on mobile broadband Internet access can be provided via 4G-LTE cellular networks and Mobile Virtual Network Operators (MVNO), even in mountainous areas and at sea, at ultra-low operational cost. The outcomes of verification work indicate that the system can be applied for a range of purposes, including electricity consumption monitoring, remote crop management in agriculture, and off-site evaluation of coastal aquaculture growth in fisheries. It can also be used in machine-to-machine (M2M) application and for Internet of Things (IoT) application. Examples include fixed-point observation using telemeters and an instrument shelter for combination with a stand-alone electricity supply (such as solar cell generation) and an electricity storage system, and the connection of atmospheric sensors. The system is also useful in mass-teaching of advanced configuration technology for computer networks and server programs to students via their own digital devices. Against this background, there is also demand for a method enabling effective acquisition of more sensor information using versatile interfaces. This will support the construction of telemetry systems for defense against natural disasters and the application of data in agriculture and fisheries based on outdoor installation of the proposed remote monitoring system. This paper outlines a method for the acquisition of atmospheric pressure data via an ultra-small remote monitoring system using a Linux-based microcomputer developed by the authors' research group. The results of the study conducted indicate that longitudinal data on atmospheric pressure can be easily acquired by connecting a sensor module featuring I2C interfaces and a number of electronic parts to a microcomputer and implementing program control.

Keywords: VPN, cellular network, remote monitoring, Raspberry Pi, computer network education, embedded Linux

キーワード: VPN, 携帯電話網, 遠隔監視, ラズベリーパイ, コンピュータネットワーク教育, 組み込みLinux

1. Introduction

Systems for the collection of sensor information from remote locations can today be easily made at relatively low cost thanks to the recent popularization of high-speed and

平成 29 年 1 月 5 日受付

† 工学部電気電子システム学科・准教授

†† 工学部電気電子システム学科・講師

††† 工学部電気電子システム学科・教授

low-latency mobile Internet connection services, whereas conventional systems require a costly dedicated line¹⁾. This evolution has also enabled the transmission of large volumes of data online from mobile units. As a result, a wealth of information (such as camera data from sensor-connected devices) from mobile objects can be acquired²⁾. In addition, such communication systems with microcomputer-operated sensor devices can be used to directly exchange information via a LAN or a public network such as the Internet. Their use is expected to expand into a wide range of applications, not only for sensor networks in factories and homes but also in agriculture, fishery, transportation, healthcare and tourism. The technology is also attracting increasing attention for use in M2M (machine-to-machine) application and for IoT (the Internet of Things) application. Specific examples of usage include moisture management in botanical gardens or home vegetable gardens and transmission of information by combining GPS (Global Positioning System) data from a mobile object based on installation of IoT units on cars and ships. The system is expected to expand to a wide range of applications in cooperation with big data analytics. Against this background, the authors' research group developed an extremely compact, inexpensive and versatile remote monitoring system that transmits information via the Internet at low operational cost. The system combines a Linux microcomputer with an embedded VPN program and various USB-connected sensors. The need for a costly dedicated VPN router is eliminated by the assignment of a dynamic private IP address by an Internet service provider thanks to the installation of a VPN program that traverses firewalls and NAT (network address translation). The system allows secure P2P connection between IoT terminals and the server at low operational cost. The need for sensor information acquisition equipment is also eliminated thanks to the direct acquisition of sensor information via a microcomputer using various connected sensors, including an A/D converter, via a USB interface. This technical combination allows simple acquisition of camera and temperature/humidity data from remote locations using

laptop PCs, tablet computers and similar at low operational cost³⁾. The use of ARM cores for the system's CPU cuts related costs drastically by boosting volume efficiency based on the capacity for embedding on commercial smartphones, tablet PCs and similar. The multi-tasking Linux operating system was chosen for its efficient use of storage capacity, its high processing speed, its lower CPU load and its FOSS (Free and Open Source Software) status. This technological combination enables cost-effective sensor network system creation in multiple programming languages, and allows free addition of various types of sensors via USB or I2C interfaces. The wide range of potential applications demonstrates the system's high versatility. The authors constructed a sensor network system based on a VPN that can be operated with a private IP address dynamically assigned to the IoT terminal without static from the Internet service provider. This is proposed as a solution that allows stable, non-stop prolonged operation and uninterrupted communication between the IoT terminal and another server. Its performance is based on the system's capacity for instantaneous switching among fixed-line, Wi-Fi and cellular network connections if any of these paths is interrupted in a disaster situation or similar. However, there is a need to acquire, aggregate and analyze more sensor information to a microcomputer to support the installation of the system outdoors, the acquisition of data for use in agriculture and fishery, and the construction of telemeters for defense against natural disasters. In this study, a method for the acquisition of time-series data on atmospheric pressure via an ultra-small remote monitoring system using a Linux-based microcomputer developed by the authors' research group was studied with reference to previous publications^{4) - 6)}. A method of disclosure for such longitudinal data on atmospheric pressure using an HTTP server was also studied. The authors additionally constructed a digital barometer by which temporal changes in atmospheric pressure can be viewed on the Internet.

2. System summary

The system evaluated in this study consists of a microcomputer for command control and an atmospheric pressure sensor module as shown in **Figures 1 and 2**. It was constructed with reference to ^{7)–9)}, and an LPS331 absolute pressure sensor module (Strawberry Linux) was adopted for the acquisition of atmospheric pressure data. The module features an LPS331AP single-chip atmospheric pressure sensor measuring 3 x 3 mm with a thickness of 1 mm (ST Microelectronics, Ltd.) and a printed board, and is compatible with I2C and SPI serial communication interface protocols. Either of two addresses addressed to the module can be chosen when the I2C protocol is used. The sensor part of the chip was produced using MEMS technology, and includes a 24-bit A/D converter to support high-precision sensing. The resolution and precision of the chip are 0.020 mbr and ± 0.1 hPa, respectively¹⁰⁾. These specifications enable high-accuracy measurement of changes in atmospheric pressure. In the proposed system, an I2C interface was adopted for sensor data transfer between the microcomputer and sensor module based on the module's capacity for serial data transmission. The sensor chip and the microcomputer were electrically connected as described below.

The SCL (No. 4) and SDA (No. 6) terminals of the sensor chip were connected to the SCL (No. 5) and the SDA (No. 3) terminals of microcomputer. Next, the SDO/SAO (No. 7), CS (No. 8), VDD (No. 14) and VCCA (No. 15) terminals of the sensor chip were connected to the 3.3 V (No. 1) terminal of the microcomputer. The GND (No. 16) of the sensor chip was connected to the GND (No. 6) of the microcomputer. In addition, capacitors were connected to the electrical circuit to reduce data transmission errors, and resistors were connected for pull-up. A breadboard was used for wiring between the sensor module and the microcomputer and to connect the electrical circuit element, as the study's purpose was to enable the acquisition of basic environmental information by connecting sensors with an I2C interface. The wiring of the breadboard is shown in **Figure 3**, which highlights the

efficient location of electrical parts such as the IC module, resistors and capacitors.

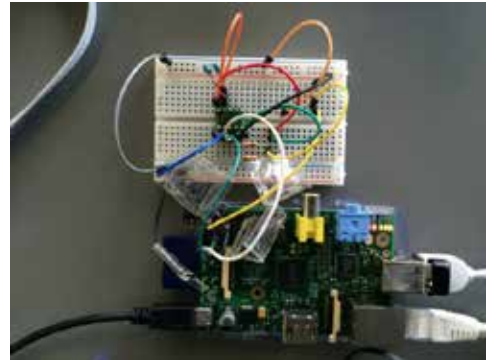


Figure 1 Overview of the proposed system

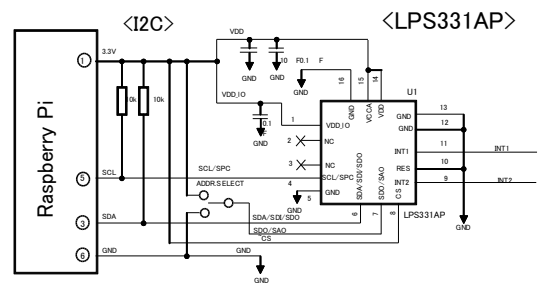


Figure 2 Circuit diagram of the proposed system

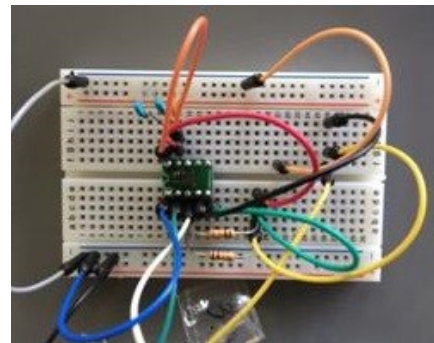


Figure 3 Atmospheric pressure module wiring

3. Adjustments to enable I2C communication

Next, the following command was input after connection of a sensor to the Raspberry Pi¹¹⁾ via I2C terminals to enable I2C connection in the above sequence:

sudo nano /etc/modules

The device name “i2c-dev” was then added to the module setting file to enable recognition by Raspberry Pi. The default configuration for modules compatible with Raspberry Pi was also edited, and the black list was adjusted to disable I2C and SPI communication interfaces. The blacklist configuration file was first loaded using

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

Next, the line showing “i2c-bcm2708” was removed from the black list by adding “#” to the start of the line. Finally, the file was saved and blacklist file editing was ended using the commands “Ctrl+o” and “Ctrl+x.”

Next, i2c-tools functionality was installed on the Raspberry Pi. I2C (Inter-integrated Circuit) is a serial data communication protocol for processor-device communication using SCL (serial clock) and SDA (serial data for bidirectional communication) wires. The circuit consists of a master device and a slave device, and a bus line is also included to accommodate multiple devices. First, the master device chooses the address of the slave device as set in advance when actual communication is made. The master device is then able to communicate with the slave device. The I2C protocol also provides a bit-rate choice of fast mode or high-speed mode for transmission. The microcomputer was set up by executing the two commands shown below and installing I2C and Python software because the proposed system is designed for the acquisition of atmospheric pressure data via the microcomputer from a sensor module using I2C interfaces.

```
sudo apt-get install i2c-tools
```

```
sudo apt-get install python-smbus
```

After installation of this software, the I2C function of the microcomputer was activated using the advanced options of “raspi-config” (the command for basic microcomputer settings; this process is very important). Once the microcomputer was able to communicate with various sensors using I2C, the following command was input to check the device address:

```
sudo i2cdetect -y 1
```

This resulted in output of the hexadecimal expression “0x5d” as shown in **Figure 4**, which confirmed that the

microcomputer recognized the LPS331AP atmospheric pressure sensor module. The following command was also executed to check communication between the microcomputer and various sensors via the I2C interface:

```
sudo i2cget -y 1 0x5d 0x0f
```

This resulted in a response of “0xbb” from the sensor to the microcomputer, thereby confirming correct communication between the two units as shown in **Figure 5**.

```

25:44 12:48:22 ~ # i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
01: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
02: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
03: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
04: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
05: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
06: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
07: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
08: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
09: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
11: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
12: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
13: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
14: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
15: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
16: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
17: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
18: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
19: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
1a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
1b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
1c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
1d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
1e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
1f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
21: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
22: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
23: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
24: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
25: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
26: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
27: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
28: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
29: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
2a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
2b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
2c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
2d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
2e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
2f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
31: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
32: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
33: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
34: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
35: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
36: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
37: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
38: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
39: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
3a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
3b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
3c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
3d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
3e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
3f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
41: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
42: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
43: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
44: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
45: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
46: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
47: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
48: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
49: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
4a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
4b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
4c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
4d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
4e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
4f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
51: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
52: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
53: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
54: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
55: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
56: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
57: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
58: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
59: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
5a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
5b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
5c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
5d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
5e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
5f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
61: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
62: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
63: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
64: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
65: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
66: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
67: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
68: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
69: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
6a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
6b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
6c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
6d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
6e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
6f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
71: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
72: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
73: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
74: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
75: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
76: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
77: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
78: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
79: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
7a: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
7b: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
7c: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
7d: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
7e: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
7f: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

```

Figure 4 Confirmation of I2C device address

```

25:44 12:48:22 ~ # i2cget -y 1 0x5d 0x0f
0xbb

```

Figure 5 Confirmation of communication via I2C

4. Acquisition of atmospheric pressure data

Serial communication between the microcomputer and the atmospheric pressure sensor via the I2C interface was enabled by the work described in the previous section. The following two commands were then input to activate the atmospheric pressure sensor module:

```
sudo i2cset -y 1 0x5d 0x20 0x90
sudo i2cset -y 1 0x5d 0x20
```

These commands need to be input again when the Debian OS is shut down and re-booted to ensure correct readings of atmospheric pressure values from the sensor module. After the activation of the above device, three-byte atmospheric pressure data from the sensor module can be read using the following commands:

```
sudo i2cget 1 0x5d 0x28
sudo i2cget 1 0x5d 0x29
sudo i2cget 1 0x5d 0x2a
```

From these commands, atmospheric pressure values with the least significance, medium significance and the highest significance are read out via the I2C interface from the 0X28, 0X29 and 0X2a resistors of the sensor module, for which the 5d interface is assigned. In addition, atmospheric pressure data consisting of $2^{12} = 4,096$ numbers as transmitted from the 24-bit A/D converter are stored in the internal resistors of the sensor module. Three bytes of data (0x28=0x68, 0x29=0x1f and 0x2a=0x3f) were actually read out from the sensor module via the I2C interface based on input of the above commands. The values were translated from three-byte hexadecimal form into digital values by inputting following command:

```
sudo perl -e 'print(0x Readed 3byte Value/4096)'
```

Then, when “sudo perl -e ‘print(0x3f1f68/4096)’” was entered, the atmospheric pressure value “1009.9628 (hPa)” was read out as shown in **Figure 6**.

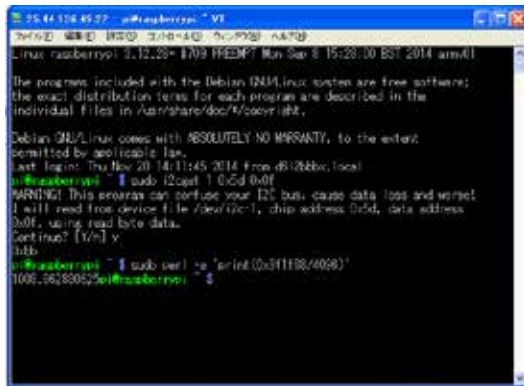


Figure 6 Acquisition of atmospheric pressure data from the sensor module

The results of atmospheric pressure monitoring from January 26th to February 3rd 2015 using the above architecture and procedure are shown in **Table 1**. A slight difference is seen between values obtained by the proposed system and data from the Japan Meteorological Agency (JMA) for the central part of Hachinohe City. The reasons for this difference are discussed below.

First, JMA’s atmospheric pressure values were corrected using data on atmospheric pressure levels (1,013 hPa) at 0 m above sea level for reference based on measured values. Meanwhile, the microcomputer atmospheric pressure data were acquired in the laboratory on the fourth floor of the Hachinohe Institute of Technology (HIT), which stands on land at an altitude of 98 m. Assuming that the height per floor is 3 m, the ASL altitude of the laboratory is 110 m. As atmospheric pressure changes by 1 hPa for every 10 m of altitude, this would account for an error of about 11 hPa.

Table 1 Relationships linking temporal variations in atmospheric pressure at HIT, temperature and weather

Date (2015)	Atmospheric pressure (hPa)		Temperature (°C)	Weather
	Proposed system	JMA value		
January 26th	996	1,019	8.9/-2.5	Cloudy with intermittent rain
January 27th	992	1,006	7.9/-1.9	Cloudy with intermittent rain
January 28th	1,002	1,013	-1/-4.5	Cloudy
January 29th	1,012	1,025	2/-4.9	Cloudy with intermittent sunshine
January 30th	1,005	1,019	3.9/-3	Cloudy with snow later
January 31st	994	1,005	4/-0.5	Cloudy with intermittent snow
February 1st	1,001	1,011	2.6/-3	Sunny
February 2nd	1,006	1,020	3/-3	Cloudy with intermittent sunshine
February 3rd	1,010	1,024	2/-1	Sunny with intermittent cloud

Note: JMA values were recorded in the center of the city

5. Script-based automatic acquisition of data

The setting and input of commands described in the previous section resulted in the successful output of atmospheric data from the sensor module. However, such output was not possible without manual input of these commands. Accordingly, a script was created to automate atmospheric pressure data acquisition with reference to [5]. First, the following setting file was opened using editor software:

```
sudo nano /usr/modules/lps331ap.sh
```

Next, the script shown in **Figure 7** was added to the file. Finally, “Ctrl+o” was used to overwrite the file, and editing was ended with “Ctrl+x.” An instruction to output the data to the text file “/home/pi/test1” was added as shown in the following script because acquisition of atmospheric pressure information is performed using Munin software as described later:

```
#!/bin/bash

WHO_AM_I='sudo i2cget -y 1 0x5d 0x0f'
if [ $WHO_AM_I != "0xbb" ]; then
    echo "device NG"
    exit 1
fi

### set active mode
sudo i2cset -y 1 0x5d 0x20 0x90

### read pres data

PressOut_XL='sudo i2cget -y 1 0x5d 0x28'
PressOut_L='sudo i2cget -y 1 0x5d 0x29'
PressOut_H='sudo i2cget -y 1 0x5d 0x2a'

RawDatHex='echo
"0x${PressOut_H:2:2}${PressOut_L:2:2}${PressOut_XL:2:2}"'

RawDatDec='printf %d $RawDatHex'

echo "scale=2;$RawDatDec/4096" | bc
echo "scale=2;$RawDatDec/4096" | bc > /home/pi/test1

### set power down
sudo i2cset -y 1 0x5d 0x20 0x00
```

Figure 7 Edited content of the script file

As only the root user can execute the generated script file “lps331ap.sh,” execution authority was granted to all users with the following command from the Debian OS command line:

```
sudo chmod u+x ping.sh
```

Next, access authority was extended to other users with the following command:

```
cd /usr/modules
sudo chmod 774 lps331ap.sh
```

Finally, the following command was input:

```
sudo /usr/modules/lps331ap.sh
```

This resulted in the output of “1001.44 hPa” as shown in **Figure 7**. Thus, atmospheric pressure data in decimal form was successfully acquired with single-command input.

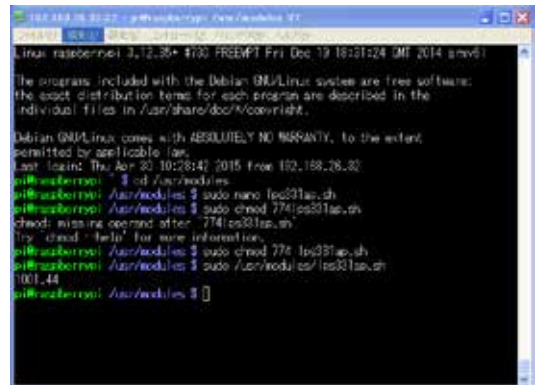


Figure 7 Automatic acquisition of atmospheric pressure data using a script file

5. Acquisition of time-series data on temperature, humidity and atmospheric pressure

Next, environment-setting for the Munin software was performed using the four commands shown below to display time-series data on temperature and humidity from the USB-connected sensor module in a web browser.

```
sudo chmod 755 /usr/share/munin/plugins/usbrh
sudo ln -s /usr/share/munin/plugins/usbrh/etc/munin/plugins
sudo /etc/init.d/munin-node restart
```

The Linux OS was also rebooted with the following command:

```
sudo service munin-node restart
```

As a result, temperature and humidity information was reflected in the setting of the proposed system once Munin had been rebooted. Finally, the IP address assigned to the microcomputer was typed into the browser’s address bar to check the above setting conditions and the time-series data from the sensor module. This can be done on a PC or smart device in a remote location via the Internet. The

input IP address was assigned from the Hamachi network.

<http://25.44.136.49/sensors-day.html> (Acquired data by day)

<http://25.44.136.49/sensors-week.html> (Acquired data by week)

Graphs showing time-series wave forms of temperature and humidity were displayed in the browser using Munin as shown in **Figures 8** and **9**.

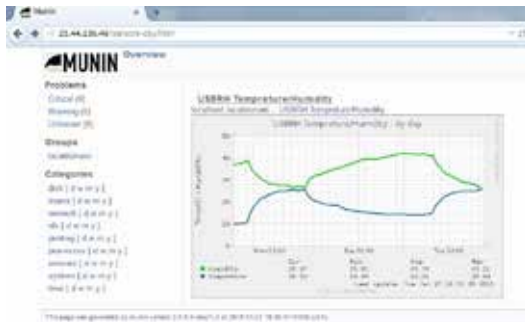


Figure 8 Time-series data on temperature and humidity by day

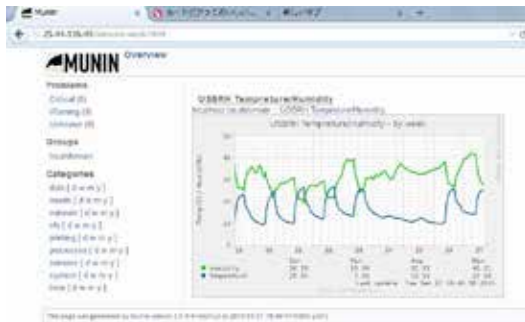


Figure 9 Time-series data on temperature and humidity by week

The script was also edited to enable output and display of time-series atmospheric pressure data from the sensor module (LPS331) and stored in the file at the directory location “/home/pi/tesxt1” using Munin as follows:

```
#!/bin/bash

#/# family=auto
#/# capabilities=autoconf

LPS331="/usr/modules/lps331ap.sh"
available="yes"
case $1 in
config)
echo "graph_title LPS331 Atmospheric Pressure"
echo "graph_category sensors2"
echo "graph_vlabel Pressure (hPa)"
echo "graph_args --rigid -l 900 -u 1100"
echo "pressure.label Atmospheric Pressure"
```

```
echo "pressure.draw LINE2"
exit 0
;;
autoconf)
if [ "Savailable" = "yes" ]; then
echo "yes"
exit 0
else
echo "no (daemon isn't running)"
exit 1
fi
;;
snmpconf[suggest)
exit 0
;;
*)
;;
esac
# Readout of the data
values=(`$LPS331`)

OLDIFS=$IFS
IFS=
exec </home/pi/tes1
while read LINE
do
echo "pressure.value" $LINE
done

IFS=$OLDIFS
```

Next, this script file was saved to “/usr/share/munin/plugins/lps331b” and access authority was changed using the command “chmod 775 lps331b.” A symbolic link was also generated using the following commands:

```
cd /etc/munin/plugins/
sudo ln -s /usr/share/munin/plugins/lps331b lps331b
```

Munin was restarted using the “sudo munin-node restart” command after the previous process had been killed. Finally, the script file below was edited and periodically and automatically executed in the same way as the previously presented script by inputting “sudo nano crontab -e” and editing the file using the cron command. This is the standard Linux command for acquisition of atmospheric pressure data from the sensor module.

```
00 * * * * /usr/modules/sudo lps331ap.sh
20 * * * * /usr/modules/sudo lps331ap.sh
40 * * * * /usr/modules/sudo lps331ap.sh
```

As a result, a time-series graph of atmospheric pressure was successfully displayed in the browser as shown in **Figures 10** to **12**. **Figure 10** shows that atmospheric pressure changed from 998 to 1,020 hPa, and **Figures 11** and **12** show weekly and monthly variations in atmospheric pressure, respectively. In future work, it will

be necessary to study the correlation between height differences or continuous variations in geographical features and atmospheric pressure using the system in combination with GPS.

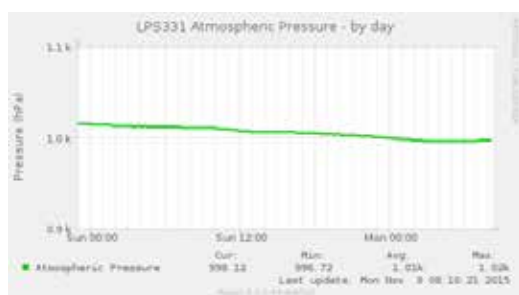


Figure 10 Changes in atmospheric pressure by day

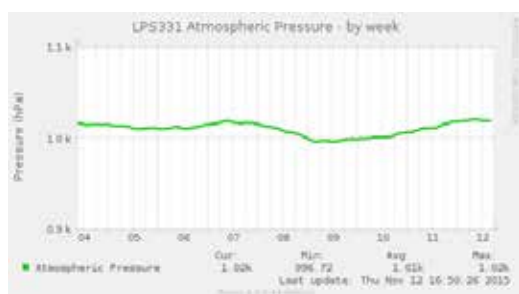


Figure 11 Changes in atmospheric pressure by week

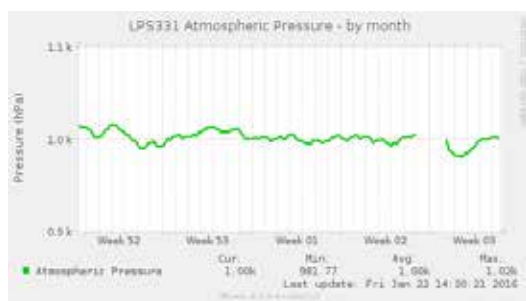


Figure 12 Changes in atmospheric pressure by month

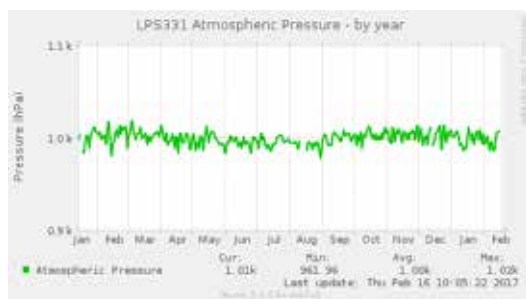


Figure 13 Changes in atmospheric pressure by year

6. Conclusion

In the research reported here, a method for atmospheric pressure data acquisition was studied. The aim was to enable the collection of an even wider variety of sensor data for use in fishery and agriculture (as compared to the authors' previously proposed remote monitoring system) by installing equipment outdoors toward the creation of a telemeter system that could also be used for defense against natural disasters. The results indicated that atmospheric pressure data could be easily acquired from a sensor module connected to a microcomputer via I2C interfaces. In future work, it will be necessary to study the correlation between differences in height or continuous variations in geographical features and atmospheric pressure using the system in combination with GPS. The equipment also needs to be improved by mounting the electrical parts on a printed board and using a waterproof case for long-term outdoor operation.

Acknowledgements

The authors wish to acknowledge Mr. Yuuma Akatsuka for his support in this research during his period of study at Hachinohe Institute of Technology.

References

- 1) Kenji Watanabe, Makoto Otani, "The TANADA-Online Project" IEICE Technical Report Vol. 108 (74), IA2008-9, pp. 43-48, 2008-5.
- 2) Kouji Shibata et al., "A Remote Monitoring System for High-Speed Sensor Information Based on a Stand-Alone VPN over a WiMAX Network," The Bulletin of the Hachinohe Institute of Technology, 32, pp. 129-134, 2013-3 (in Japanese).
- 3) Kouji Shibata, Kazuma Hanada, "Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation," The Bulletin of the Hachinohe Institute of Technology, 33, pp. 115-120, 2014-3.
- 4) Kouji Shibata, Kazuma Hanada, Masahiro Iino, Misato Take, Yuuma Akatsuka, "Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation and its Application for

- Educational Purposes,” IEICE Tokyo Japan, Tech. Rep., Vol. 114, No. 441, ET2014-83, 2015-1 (in Japanese).
- 5) Kouji Shibata, Masahiro Iino, Misato Take, Yuuma Akatsuka, Kazuma Hanada, “Development of a Remote Monitoring System with an Embedded VPN for Sensor Information Application to Education with a Linux Microcomputer for Sensor-based Disaster Information,” Proceedings of the Society Conference of IEICE, D-15-5, 2015-3 (in Japanese).
- 6) K. Shibata, K. Hanada “Development of an Ultra-small Sensor Information Remote Monitoring System with an Embedded VPN and Linux Microcomputer Operation,” Proceedings of International Conference on Engineering and Applied Science, ICEAS2015, ICEAS-3784, Sapporo, Japan, pp. 169-179, 2015-7.
- 7) First reference to website on data acquisition via I2C: <http://mutsumi3memo.blog.fc2.com/blog-entry-25.html>
- 8) Second reference to website on data acquisition via I2C: <http://44781184.at.webry.info/theme/0532575e5c.html>
- 9) Third reference to website on data acquisition via I2C: <http://news.mynavi.jp/articles/2014/08/19/raspberry-pi3/>
- 10) MEMS pressure sensor (LPS331AP) data sheet: <http://www.st-japan.co.jp/st-web-ui/static/active/jp/resource/technical/document/datasheet/DM00036196.pdf>
- 11) Raspberry Pi website: <http://www.raspberrypi.org/>.

要 旨

筆者らは以前、ARM-LinuxマイコンであるRaspberry Piと3.9Gの携帯電話網に接続可能なUSBモデムを組み合わせ、プライベートIPアドレスでも動作する超小型で低消費電力かつ安価な運用コストの低い NAT越えが容易な組込み型VPNを構築した。さらに格安な市販のWEBカメラや温湿度センサを接続してタブレットコンピュータやスマートフォン（スマホ）などのスマートデバイスを用い、センサから遠く離れた場所からでもWebブラウザでセキュアにカメラ画像や温度・湿度の情報が取得可能であることを示した。本システムは高品質なMVNOなどのLTEの4G携帯電話回線キャリアを介しモバイルでインターネットに接続するため、山中や海上など任意の場所へユビキタスに設置が可能である。成果は電気の使用量の把握や農業では農作物などの遠隔管理、漁業では海岸での海産物の盗難の監視、さらに気圧センサや太陽光パネルなど独立電源との組み合わせでテレメータや百葉箱などの定点観測などMachine to Machine (M2M) や Internet of Things (IoT)用途へ応用できるほか、学内のネットワーク教育にも有用と考える。これらを屋外に設置して農業や漁業に活用できるデータを取得したり、防災用のテレメータシステムを構築するためには、さらに多くのセンサ情報を汎用的なインターフェースを介し、効率良く取り込む必要がある。そこで本研究では、筆者らが開発したLinuxマイコンを用いた超小型遠隔監視システムへの気圧データの取り込み法につき検討した。その結果、I2Cインターフェースを有するセンサモジュールを付加した電子部品とともにマイコンに接続し制御することで、気圧の時系列データが容易に取得出来ることが確認できた。

キーワード：VPN, 携帯電話網, 遠隔監視, ラズベリーパイ, コンピュータネットワーク教育, 組込みLinux