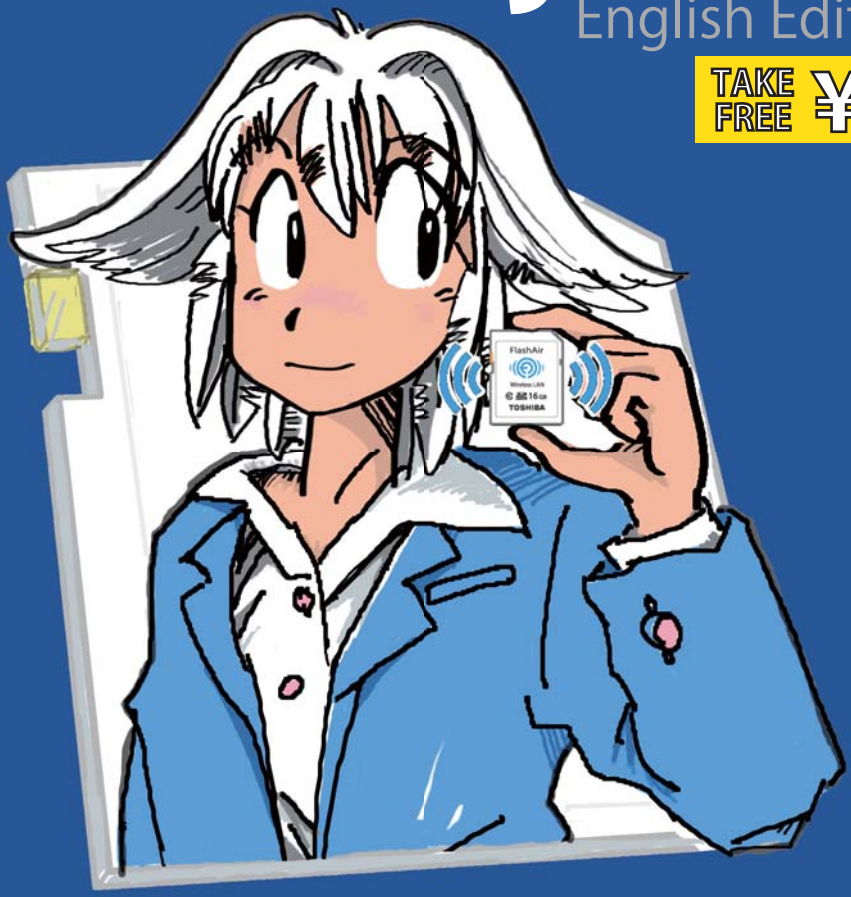
FlashAir™ Doujinshi

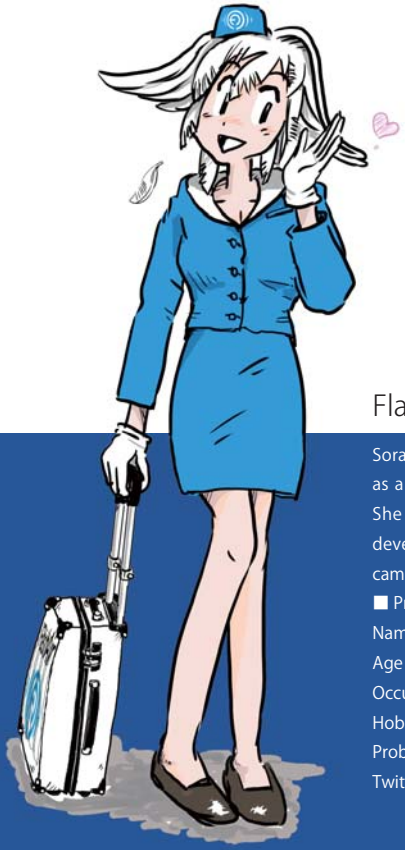
English Edition

TAKE FREE ¥0



A slightly unusual way of using the SD card FlashAir with built-in wireless LAN explained by people involved. Get ready for take off!

Contents



FlashAir Support Character "Sora Hirameki"

Sora Hirameki is the unofficial support character of FlashAir. She works as a cabin attendant at an airline company - so she is frequently away. She spends her off time working with electronics, doing application development, and taking pictures with her mirror-less single-lens reflex camera.

■ Profile

Name	Sora Hirameki
Age	23
Occupation	(Trainee) cabin attendant
Hobby	Electronic work, application development
Problem	Electronic equipment not usable on airplane
Twitter	@Hirameki_Sora

Maker Faire Tokyo 2014

Takada

After many discussions and studies on how to best use FlashAir, we realized that not every new application is being developed on the product manufacturing side. We've been searching for a killer application and working in partnership with other companies, but activities between companies are not well promoted unless the business prospect is in sight at a very early stage. Even though we were holding idea discussions, nothing was happening – are there no other ways to find this killer application? Can't we work with people who can freely suggest their ideas? With these thoughts in mind, we looked into the Open Source Conference. What passion the people there have for making (even if it's often understated)! Their honest expectations and comments on products! There, I found a totally different world from the business development centered on proposals from big companies. They search for technology itself driven purely by an individual engineer's interests and concerns - and they have so much energy!

I timidly stepped forward to this totally different world. First of all, products must interest people. For Maker Faire Tokyo 2014, many works of electronic work were collected, actually much more than expected, so that people take interest in them. Please come and see the many of exhibits in the small booth. If you're interested, please play with FlashAir and show us if your work can be achieved with it.

The above activities led us to the idea that FlashAir is not a product that will just survive with one killer application. I cannot help believing that it will transform into another shape in the near future. It may become a thing that always exists close to us and helps solve dissatisfaction in business and everyday life.

I hope our activities will lead to all sorts of utilization ideas, whether they are games or toys, something relaxing or hilariously absurd, a learning tool, or even just something useful in life.

I want people to say “We can play a lot with FlashAir.” From there onward, under my secret ambition of introducing IoT from Japan to the world, I want to make our activities long-standing.

FlashAir and GPIO

Ito

I have a three-year-old son. His name is K. Generally boys are slower than girls in learning words, and recently he started talking at last. Whenever I talk of ghosts and monsters, he gets scared and asks me to stop, crying. I enjoy playing with him a lot at weekends.

This is a story of the time he was born. Just two weeks after he was born, a sample of original card-shaped FlashAir was completed. This also "new born" card could do nothing yet.

Now, FlashAir is known as an SD card with built-in wireless AP and Web server "which transfers photos you take by a digital camera to smart phone on the spot". However, at first, we had no such idea, and actually even no software to run inside. Therefore I worked hard on coding, while my son was on my knees. Now, that became a topic of our drinking party.

Recently my son K got to do small things I asked him such as “switch on the light” or “fetch a picture book to go to bed”. Though they are very simple things, I feel as if I could communicate with him and was happy as his parent. I feel him growing and understanding words, though he is still very young.

Thinking of these small things I can ask my three-year-old son to do, I led FlashAir, another child, into doing such things, and that is a story of attaching GPIO to FlashAir.

GPIO is the abbreviation of General Purpose Input Output. Another explanation is that it can convert electric signals from a microcomputer to 1 or 0. If the GPIO of a microcomputer is connected, we may be able to switch on the electric fan or light LED.

GPIO can realize such small things, and how to use it is the same as asking a three-year-old child. For example:

* Instead of asking K in a voice, ask FlashAir using wireless LAN
* Instead of K’s switching on, convert FlashAir SD interface to 1 or 0
* Instead of saying “please”, input a URL in browser

These are GPIO functions of FlashAir but unlike K, FlashAir is obedient to my requests. What it can convert to 0 or 1 is countable by one hand and that is like my son. Another similarity may be that sometimes it becomes unable to communicate. I will omit the details and leave more explanations to other people. I feel glad if everyone can play with FlashAir and tweet in Twitter, putting the above in mind.

* This story is fiction.

The Birth History of Sora-chan and GPIO with OSC

Pochio

Now, in a booth in the exhibition, we can meet many people who say they know FlashAir. However, a year ago, it was a rare case to meet such people. In fact, I was one of them, and until recently I didn't know of its existence even though it was our product.

Encounter with FlashAir

My encounter with FlashAir came suddenly. In mid-June, 2013, my ex-Senior Manager came to consult my next co-worker on how to use FlashAir, bringing a package of it with him. Then, for the first time, I saw the blue package of FlashAir my co-worker showed me.

FlashAir is a product in consideration of use by digital camera. On hearing it is an SD card with wireless LAN, what came into my mind was Raspberry Pi. Raspberry Pi is the popular world-wide mini-computer where Linux runs. As interface, GPIO (General Purpose Input/Output) terminal, two USB ports, wired LAN port and SD card slot are equipped. When we want to use wireless LAN, we have to insert its module into the USB port. However, if we can make a wireless LAN communication from FlashAir inserted into the SD card slot, blocking up of USB port of Raspberry Pi is not necessary, so, I thought, Raspberry Pi users all over the world would definitely use it.

I hurriedly made the explanatory document on combination of FlashAir and Raspberry Pi and distributed to those concerned. Then I was called to a meeting at Head Office. There, I found we cannot use FlashAir like a wireless LAN card and my plan went wrong. Nevertheless, I still believed in some possibility of its combination with Raspberry Pi and consulted my old friend, Dr. K who was using it. His valuable suggestions were:

(1)GPIO should be attached to FlashAir.

(2)Program coded by users should be made run on FlashAir.

(3)How about advertising by joining OSC[1], as API of FlashAir is open to the public.

He suggested (1) and (2) are necessary functions as embedded utility, but they are not easy to realize. As for (1), he suggested "making a hole at the end of the

1 OSC (Open Source Conference) is an exhibition and seminar related to open source which is held around Japan throughout the year.

card and pull out a terminal from it," but doing this is against the SD standard. As it is our company that made it, anything against it is prohibited. As for (2), RAM mounted in FlashAir is too small to carry out a program made by users. FlashAir is closely packed with the parts and has no room to load a larger RAM. The remaining suggestion (3) is realizable, so, accompanied by Dr. K. and some people concerned with FlashAir, I visited Begi.net Inc. administering OSC and decided to join the exhibition of "OSC Tokyo" held in October, 2013.

First Exhibition in OSC

After our exhibition was decided, trial and error continued on the contents. As a novelty, we tried to make a FlashAir-type notebook, but no manufacturer was found who can cut the right upper corner so that it has a shape of an SD card, so our alternative idea of making sticky label representing FlashAir was adopted (Fig.3-1). In the booth, FlashAir function was explained and in another room a seminar by FlashAir Developers[2] was held. Originally designed FlashAir (Fig.3-2) was presented to the visitors by lottery but most of them did not know it. On our explanation that it is an "SD card with Wi-Fi," their question was "how does it differs from a famous orange SD card installing Wi-Fi ?" Here, we keenly realized the gap of popularity.

Fig.3-1: Sticky label with a design of FlashAir (Above: front, Below: back)

Fig.3-2: First original design of FlashAir

New Year started, and I was shocked at the web-news. The Company called a "giant" in the semi-conductor world released "Edison". At first, it was an SD card type computer mounting SoC, memory, 2GB storage and Wi-Fi, with

2 https://flashair-developers.com/en/

GPIO terminal seen in the back side hole of the card. Yes, it was exactly what Dr. K. suggested before. "Didn't I tell you so?" he pointed out in Twitter. I began to worry and conveyed this to Mr. Ito who is developing FlashAir. Unexpectedly, he seemed not to care and only said "Electricity consumption seems high, doesn't it?"

Birth of Sora-chan

Spring OSC in Tokyo 2014 was the 100th memorial and a costume play convention of supporting companies' mascot characters was planned. It was to be relayed by live broadcasting on the Internet and no doubt will get lively by participation of popular "Konoha-chan" and "Cloudia-san". We were invited to the event, but frankly, I worried. Needless to say, we had no cute mascot character for FlashAir. However, there might be no other chance of promotion like this. Recently various industries and manufacturers have been developing advertisements with their own mascot characters. I proposed that we make one and join the event with "try-it-anyway spirit", and was permitted.

Fig.3-3: Initial design of Sora-chan

Fig.3-4: Final design of Sora-chan

Soon after that, we discussed the character setting and agreed on the concept that "she is twenty-three year old cabin attendant (CA), an electronic enthusiast and her problem is that electronic equipment is not usable in a plane during flight". The answer to why CA is because, after the first FlashAir promotion, a suit of CA was left which was the only clothes we could prepare. Gm-san who designed the character named her Sora Hirameki. It was named after FlashAir using Japanese translation Hirameki for flash and Sora for air. Sora-chan designed by gm-san first (Fig.3-3) was changed like Fig.3-4 to suit her clothes. He attached more originality to her in that her hair is made of feathers (Fig.3-5).

At the end of January, a meeting was held to choose a model for the costume play.[3] From ten candidates, Miss Arisa Kunii was chosen as the prettiest. We requested her to act as a model and she agreed. With her clothes already prepared, what was left was to realize the fantastic idea that her hair is made of feathers! Realization of this will be the Leading Innovation in the costume play world! Gm-san called many hair salons producing wigs and finally found one in Akihabara who granted our wishes.

The day of actual performance came on 1st March. After seven in the morning, I arrived at the meeting place in Meisei University, the venue of OSC Tokyo, and found Miss Kunii already there and gm-san too, with the finished wig with him (Fig.3-6). Miss Kunii, in the wig fitted by a professional make-up artist, was really Sora-chan herself, and there was no word other than wonderful. In the seminar, the same as the previous time, we presented FlashAir designed like Fig.3-7. Thus by joining OSC twice in spring and autumn, we successfully raised the popularity of FlashAir among geeks.

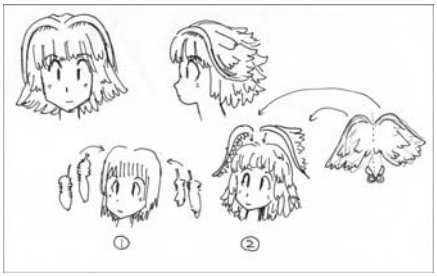


Fig.3-5: Sora-chan's hair setting

Fig3-6: Sora-chan's wig (trial product)

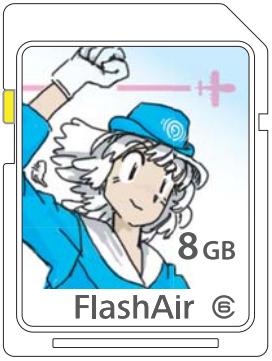


Fig.3-7: Second original design of FlashAir

Opening of FlashAir Make Idea Contest and GPIO Installation

In June, we decided to collect ideas of utilizing FlashAir by holding a "FlashAir Make Idea Contest" on the FlashAir Developers' site. We made a Japanese fan out of thick paper as a handbill to notify the event and gave it to all visitors in OSC Nagoya held in July. At first, the design motif was created from the image of Sora-chan's

3 She was actress appearing on TV and on the stage.

Fig.3-8: Japanese fan distributed in OSC Nagoya and Kyoto

debut single record and the music was titled by me "Electronic Work by a Maiden Aboard a Plane". That was rejected before I knew it, but the appearance of the debut single record was preserved (Fig.3-8).

After the exhibition in OSC Nagoya, I heard from gm-san that Ito-san installed GPIO in FlashAir. "He couldn't make a hole, could he?" I asked. The answer was "He didn't." I never thought of transforming the SD terminal to GPIO function without making a hole! In case of using the SD terminal as GPIO, reading and writing of the data to flash memory inside FlashAir is not possible from the host equipment (possible from Wi-Fi) but nothing is different from the fact that GPIO was installed. It took one year to realize what Dr. K. suggested was necessary. The reason why Ito-san seemed not to care before was, I think, because he had already been prepared.

Make of Sora-chan's Printed Circuit Board

Fig.3-9: Wireless crane game

We planned to advertise the GPIO function in OSC Kansai @Kyoto and gm-san created a wireless crane game as a utilization example of FlashAir (Fig3-9). It gained a good reputation. In the process of the game creation, he found it better to prepare the interface for users to connect what they want to connect with FlashAir so that they can use the GPIO function easily. Surely there is no other SD card slot to cope with GPIO function of FlashAir in the world. Considering embedded utility, gm-san designed a printed circuit board (PCB) also usable as Arduino shield (Fig. 3-10). His idea was to have it used by people who want to play with it, in the next OSC Tokyo (autumn) in October and to have their utilization examples reported in Twitter. By

the way, he illustrated the PCB with Sora-chan's picture, influenced by yone2-san of Doujin Circle "Respon" who skillfully illustrated Doujin PCB with a girl, using wiring of the PCB (Fig. 3-11).

At first, distribution of the PCB included worries that those who failed in making interface with it might make a claim. At the last moment, the discussion was raised to stop distribution to evade risks. However, it is waste of the PCB already designed, so I suggested gm-san to consult yone2-san.

Fig.3-10: Example of interface PCB production for GPIO

A few days passed when I received sudden shocking mail from gm-san saying that yone2-san was one of our colleagues! In the end, we temporarily distributed the PCB under the condition that it should be made definitely with care according to instructions, and the result should be reported in Twitter. Reports appeared one after another in Twitter. Most people worked on making the LED flicker.

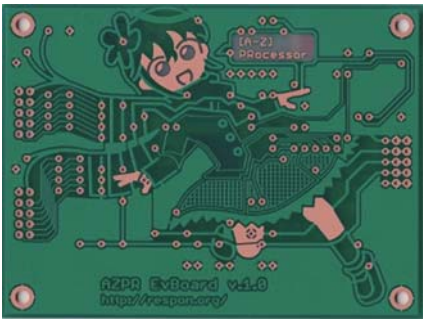


Fig.3-11: "PCB Girl" designed by yone2

Later, yone2-san designed an interface PCB independently for GPIO use, and his friend, Muraguchi-san made equipment for displaying to liquid crystal the image transferred from outside to FlashAir, and succeeded in drawing Sora-chan on liquid crystal with the lower transfer speed of 1.8 bps. Thus, he showed an example of new application. You can read the articles written by yone2-san and Muraguchi-san in this Doujinshi.

Conclusion

This doujinshi distributed at the Maker Faire Tokyo 2014, was completed with the aid of yone2-san who had experiences in publishing doujinshi. This is the whole story of FlashAir team activities through OSC. How did you feel? FlashAir still has room for development in the future with users' voices.

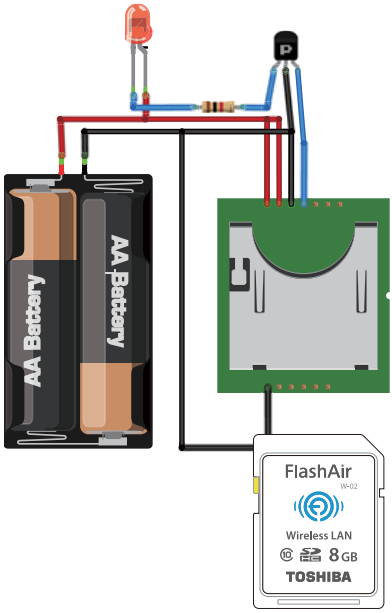
Designing of FlashAir Eva Board, Airio

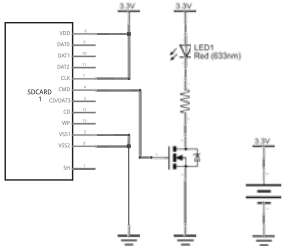
yone2@Respon

Hi everyone! I am yone2. Usually I am designing PCB and other equipment in “circle respon”, doujin hardware circle. This time I made the FlashAir evaluation board “Airio” which can easily test FlashAir’s GPIO function and is easily connectable to Arduino. This is a record of production and not a detailed explanation of design. I put the instruction manual, schematic, software, etc. in the webpage, so please refer to it if necessary. (http://yone2.net/)

FlashAir’s GPIO function

Originally, we used FlashAir with microcomputer and other equipment, but by rewriting the configuration file in the hidden folder, we can use its SD interface terminal as GPIO. With this function, we can operate the terminal of FlashAir with smartphone. Also, we can easily make the schematic for LED-flicker as below. This function is usable in Class 10 of FlashAir (FW Ver.2.00.03). Please refer to FlashAir Developers (https://flashair-developers.com) for detail.





※ Image is generated by fritzing(http://fritzing.org)

Planning

I started designing the PCB for LED-flicker to test FlashAir's GPIO function. The final goal was to make a small number and distribute them at the event. My memo just after the start reads “for low price: with only FlashAir +LED/ switch + power supply”. Later, I thought only connecting FlashAir and LED is not interesting and let the PCB have the functions of an Arduino shield. I put original rough sketch in Fig.4-1.

I aimed at the lowest possible cost in view of distribution of the products in the event. Power is supplied from USB and the PCB can be directly inserted into the USB connector. It was designed to fit inside 5x5cm, a minimum size manufacturable at overseas PCB facilities. Five centimeters square PCB is barely enough for the Arduino shield and I considered connecting it with an ISCP connector (Fig.4-2).

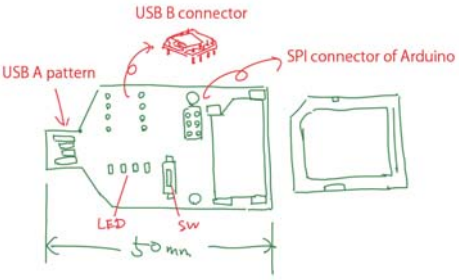


Fig.4-1: Rough sketch of initial design

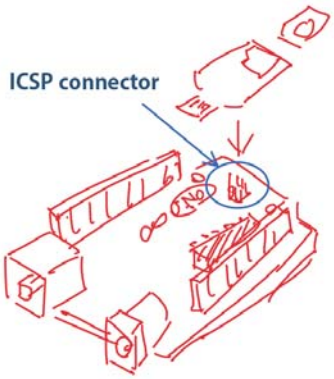


Fig.4-2: Connect Arduino with ISCP connector

Design

It took about two weeks from the selection of parts to the circuit designing and layout. Considering implementation by hand for distribution in the event, I did not use narrow pitched parts under 0.5mm and selected parts easily available. KiCad was used for circuit design and layout.

Power supply surroundings

First, I will summarize power supply surroundings. Input voltage of PCB is USB feeding or 5V feeding from Arduino. Signal level of Arduino is 5V. On the other hand, SD card is driven by 3.3V, so a regulator is necessary for power supply and a level converter for the signal. As regulator, I chose NJU7223DL1 of JRC. Its output

electric current is 500mA. Signal level conversion was carried out simply by resistive voltage divider instead of using exclusive IC.

Circuit driving LED

We drive the LED with FlashAir via FET. Full color LED has high forward voltage, so drive voltage was set at 5V. LED can be driven either with FlashAir or Arduino by putting FET in parallel disposition (Fig.4-3). As for current limiting resistance, brightness was confirmed on a solderless breadboard after calculation on the desk and the value was set.

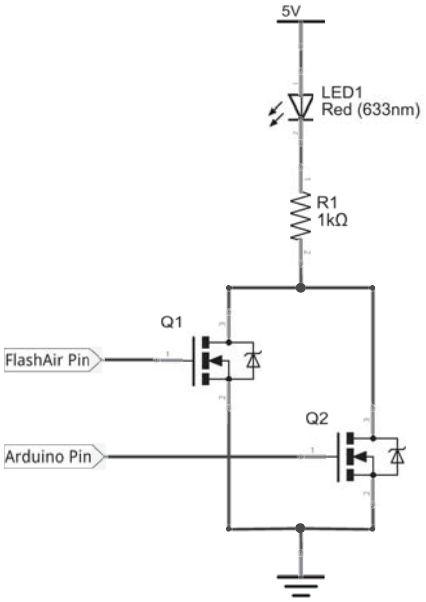


Fig.4-3: Schematic of LED driving part

Selection of the SD card receptacle

It took quite a bit of time and labor to choose the SD card receptacle. The PCB area was required to be as short as possible and to have the smallest possible metal area for implementation, but there were few dealers able to achieve this. Digi-key sells it, but at high cost, and what is obtainable at Akihabara has a problem of stock amount. Then I found the SD card receptacle used in Raspberry Pi (Fig.4.4). In Japan we can buy it at Leocom.

Fig.4-4: The same SD slot as Raspberry Pi

Pin assignment of FlashAir

The SD card uses SD I/F to communicate with digital camera and other equipment, but uses SPI to connect to microcomputer like Arduino. In case of using the GPIO function of FlashAir, a different terminal from the one used in SPI is assigned. Details are shown in Fig.4-1. In Airio, both SPI and GPIO are supported.

Table.4-1: Pin assignment of FlashAir

ピン	SD I/F	SPI	GPIO
8	DAT1		0x04
7	DAT0	DO	0x02
6	Vss2		
5	CLK	SCLK	
4	Vcc		
3	Vss1		
2	CMD	DI	0x01
1	DAT3	CS	0x10
9	DAT2		0x08

Circuit layout

The Layout drawing started after the circuit was designed. At this stage, direct insertion of PCB into USB connector was found not to fit 5 x 5cm, so USB-A connector was attached to PCB. The final layout is in Fig.4-5.

A double-sided PCB was used for wiring and was manufactured by Fusion PCB. Due to small lot distribution, the parts are implemented with the soldering reflow oven remodeled from the toaster at home.

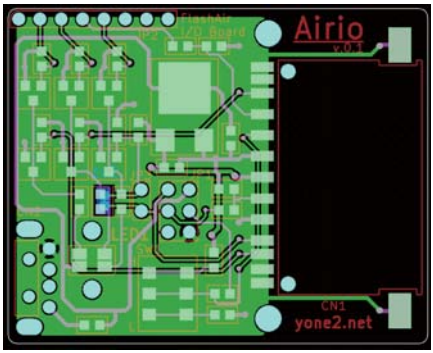


Fig.4-5: Layout of Airio

Test operation

Confirmation of GPIO function

To test GPIO function, we save HTML file on FlashAir and get access with smartphone. Full color LED has seven colors. The button of each color is located which, if tapped, will make its own color shine. (Fig.4-6)

Connection with Arduino

To connect with Arduino, it is required to attach the pin header and cut solder jumper on the back side of PCB. If sample code is written to Arduino, SD card becomes accessible. At the time of connection to Arduino, Aruduino can drive full color LED with PWM.

Fig.4-6: Operation check of GPIO function

Closing remarks

As I had only two weeks from the start of designing to the completion (gerber-out), the product was made in a rush, but I felt relieved, after confirming the safe operation of PCB. When I make hardware, I care much more about selection and supply of parts than circuit design itself. I want to compile the know-how about that in the future.

Let Aruduino control FlashAir !

Doi

Introduction

FlashAir is a super SD memory card where wireless LAN adaptor and Web server are integrated. As it has a built-in Web server, we can take out stored files on FlashAir via wireless LAN and its terminal can be used as GPIO by sending requests to special URL. However, applicability of FlashAir is not limited to that. In fact, there is a way of controlling wireless LAN freely. That is "iSDIO", an extended command of SD interface. If we use it, we can highly and precisely control wireless LAN function of FlashAir with SD host equipment such as microcomputer board. FlashAir becomes the HTTP communication co-processor.

FlashAir Developers (https://flashair-developers.com) introduces in the tutorial form the procedure from scanning and connection of wireless LAN AP to downloading data in website, controlling FlashAir by iSDIO client function implemented in Arduino. This article is a digested version. What sort of thing we can do? I hope you will get to know some possibility of FlashAir.

iSDIO

iSDIO (Intelligent SDIO) is a new standard set by the SD Association, the standardization body of SD memory card, in order to control SD memory card with attached extended function like FlashAir.

Generally in case of attaching wireless LAN function to microcomputer equipment like Arduino, a wireless LAN adaptor and driver software are necessary. However, if we use iSDIO, we can let FlashAir perform wireless LAN communication instead of microcomputer equipment. We may be able to regard FlashAir as integrated parts of wireless LAN adaptor and driver software. The simplified version of SD standard (Simplified Specifications –SD Association) is open to non-member companies of the SD Association and iSDIO specification is defined in Part E7 Intelligent SDIO Simplified Specification.

Actually, iSDIO specification is a sort of framework setting only the procedure of exchanging commands (protocol), and the details of usable function, argument, etc. are set for each kind of extended function (application) in Addendum. FlashAir

Memory mapping of iSDIO register
(Excerpt from SD Specifications Part E7 iSDIO Simplifed Specification Version 1.10)

Address	Name	Short Description	Type
00000h	Command Write Register Port	Data Port to write the iSDIO Command Write Data	W/O
00001h - 001FFh	Reserved		
00200h	Response Data Register Port	Data Port to read the iSDIO Command Response Data	R/O
00201h - 03FFh	Reserved		
00400h - 005FFh	Status Register	Memory Area for iSDIO Status Register	Table 2-7
00600h - 007FFh	Capability Register	Memory Area for iSDIO Capability Register	R/O
00800h - 00FFFh	Reserved		
01000h - 01FFFh	Reserved for Vendor		
02000h - 1FFFFh	Reserved		

Table 2-6 : iSDIO Register Map

conforms to “Wireless LAN Addendum” for wireless LAN built-in card.

Commands of iSDIO are exchanged by read/write in iSDIO register. Write to address 000h ~ 1FFh issues commands. Command response is returned to address 200h ~ 3FFh. Address 400h ~ 4FFh is reserved as common status where the state of command execution is obtained. Address 500h ~ 5FFh is the area defining the status of each application set in Addendum, and various information of wireless LAN function can be obtained for FlashAir.

Status scan

First, let's read the status of FlashAir. We read a memory block starting from the address 400h and display it according to the status map.

```
card.readExtMemory(1, 1, 0x400, 0x200, buffer);
...The codes displaying each status readably follows....
```

If this is executed, the following output can be obtained.

```
== iSDIO Status Registers ==
[0400h] Command Write Status:
[0420h] iSDIO Status: CRU
[0422h] iSDIO Int Enable:
[0424h] Error Status:
[0426h] Memory Status: MEX
[0440h] Command Response Status #1: id = 3, sequence id = 0, status = Process
Succeeded
... (snip) ...
```

Start-up in AP mode

Let's switch on the wireless LAN function in AP mode with an "Establish" command (ID 03h). At the same time, server function of HTTP and DHCP start up.

```
uint8_t* p = buffer;
p = put_command_header(p, 1, 0);
p = put_command_info_header(p, 0x03, sequenceId, 3);
p = put_str_arg(p, "sdiotest");
p = put_str_arg(p, "12345678");
p = put_u8_arg(p, 0x06);
put_command_header(buffer, 1, (p - buffer));
card.writeExtDataPort(1, 1, 0x000, buffer);
```

2nd line Making command header with helper function. In the last argument, we put command data length (number of bytes) but calculate later and overwrite due to argument of variable length.

3rd line Making header of command information with helper function. Designating establish command (ID0x03), sequence ID (sequenceID) and the number of argument (3)

4th line Writing the first argument"SSID". Using helper function to write character string argument.

5th line Similarly writing the second argument "network key".

6th line Writing the third argument "security mode". Here, designating 0x06 representing WPA2-PSK and AES.

7th line With all data written and the number of bytes determined, overwriting command header.

8th line Writing command data to FlashAir. When finished, the processing starts. Now we try to carry it out and display status. WLAN status reads AP, and IP address reads 192.168.01, so we can confirm FlashAir started up in AP mode.

```
[0440h] Command Response Status #1: id = 3, sequence id = 3, status = Process
Succeeded
... (snip) ...
[0506h] WLAN: No Scan, No WPS, Group Client, AP, Infrastructure, No Connection,
[0508h] SSID: sdiotest
[0528h] Encryption Mode: WPA2-PSK and AES
[0529h] Signal Strength: 0
[052Ah] Channel: 11
[0530h] MAC Address: E8E0B758A7FB
[0540h] ID:
[0550h] IP Address: 192.168.0.1
[0554h] Subnet Mask: 255.255.255.0
[0558h] Default Gateway: 192.168.0.1
[055Ch] Preferred DNS Server: 192.168.0.1
[0560h] Alternate DNS Server: 0.0.0.0
```

Wireless LAN scan

With Scan command (ID 01h), let's try to search SSID of wireless LAN AP in the vicinity. After the command execution, we can find SSID, signal strength and the cryptographic method of wireless LAN.

```
uint8_t* p = buffer;
p = put_command_header(p, 1, 0);
p = put_command_info_header(p, 0x01, sequenceId, 0);
put_command_header(buffer, 1, (p - buffer));
card.writeExtDataPort(1, 1, 0x000, buffer);
```

We can obtain the result of scanning by reading the result register (address 200h) after the command execution was finished.

```
Number of APs: 5
mynetwork, B86B23663750, 70, WPA2
flashair_led, B86B23583750, 64, NoSec
flashair203r, B86B23005049, 61, WPA2
HWD14_VEGETA, C40528C98B0C, 52, WPA2
FlashairT5-v2, E8E0B744A7FB, 49, WPA2
```

Download of webpage

Let's try to receive HTML data from flashair-developers. HTTP communication command has several versions based on the difference of handing over protocol or data. This time, we use the command SendHTTPSSLMessageByRegister (ID 23h) due to use of HTTP over SSL (Secure Socket Layer).

```
uint8_t* p = buffer;
p = put_command_header(p, 1, 0);
p = put_command_info_header(p, 0x23, sequenceId, 2);
p = put_str_arg(p, "flashair-developers.com");
p = put_str_arg(p,
"GET /en/ HTTP/1.1¥r¥n"
"Host: flashair-developers.com¥r¥n"
"User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/36.0.1985.125 Safari/537.36¥r¥n"
"¥r¥n");
put_command_header(buffer, 1, (p - buffer));
card.writeExtDataPort(1, 1, 0x000, buffer);
```

4th line Designating the domain name of the server. We can designate IP address.

5th line- Generating HTTP request header.

Receiving responses

We read the obtained data after the command execution was completed. Please connect beforehand to the wireless LAN capable of Internet communication. Let's display returned data written in address 200h ~ 3FFh from server.

```
HTTP/1.1 200 OK
Date: Tue, 21 Oct 2014 06:25:10 GMT
Server: Apache/2.2.15 (CentOS)
... (snip) ...
<title>FlashAir Developers - Home</title>
... (snip) ...
```

Refer to FlashAir Developers for further detail

I introduced on the run how to use wireless LAN function with Arduino, controlling FlashAir with iSDIO. Disconnection of wireless LAN, connection in station mode, sample codes of helper function, etc. are explained in the website http://flashair-developers.com./. Please refer.

Wireless SPI Master with FlashAir

Muraguchi

Introduction

This article introduces “FlashAir, also usable as wireless SPI Master”. First, I will explain the outline of SPI. Then, giving a connection example of FlashAir and SPI LCD (liquid crystal display) module, I will introduce a software controlled example. When FlashAir is controlled by Arduino and other equipment, it functions as an SPI Slave but please note that this article explains use of FlashAir as SPI Master.

Let’s widen electronic work.

About SPI

SPI (Serial Peripheral Interface) is full duplex serial communication bus specification proposed by Motorola Inc. (now Freescale Semiconductor). It consists of four kinds of signal lines, SCLK, MOSI, MISO and SS. Table 6-1 explains each signal line.

Table.6-1: Signal lines of SPI

Name of signal	Transmitting side	Receiving side	Explanation
SCLK (Serial Clock)	Master	All Slave	Data is transmitted, synchronizing with SCLK. Phase relation between edge and data of SCLK has four variations in SPI mode mentioned later.
MOSI (Master Out Slave In)	Master	All Slave	MSerial data is transmitted from Master to Slave.
MISO (Master In Slave Out)	Each Slave	Master	Serial data is transmitted to Master from Slave selected by Slave Select. Non-selected Slave is required to fix MISO at Hi-Z not to disturb other Slave communication.
SS (Slave Select)	Master (Normally, independent SS for each Slave is required.)	Each Slave	Negative-true logic. L shows selecting state of target Slave and H non-selecting state. The number of SS required is the same as Slave devices.

We added pull-up resistor

The SPI Master side can control timing of a master clock, so SPI Master can be implemented with a software controlled GPIO. In most cases, the actual SPI slave devices may have more additional signals which is not defined in SPI specification. For example, reset signals and so on. These additional signals should be also controlled for SPI communication.

SPI bus constitution

Only one Master can be placed in the SPI bus and we cannot make multi-bus Master constitution. The number of Slaves placeable in SPI bus is not limited, but an independent SS line for each Slave is required. All Slaves share SCLK, MOSI and MISO but the number of SS signal lines is proportional to the number of Slave devices, so the system with many Slave devices has a problem of increasing signal numbers. General constitution of SPI bus is shown in Fig.6-1.

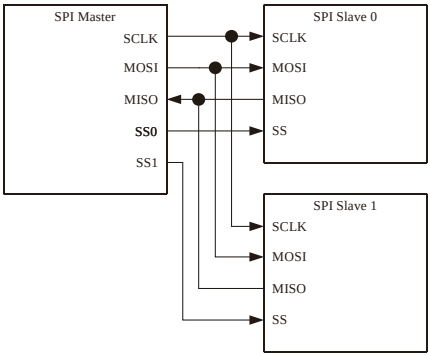


Fig.6-1: Signal lines of SPI

SPI operation mode

There are four operation modes of SPI due to the difference of clock polarity or phase relation. The relation between SPI mode, clock polarity CPOL and clock phase CPHA is shown in Table 6-2.

Table.6-2: Relation between SPI mode, clock polarity and clock phase

SPI mode	Clock polarity CPOL setting	Clock phase CPHA setting	Explanation
0	0	0	During L state of SCLK, MSB data is transmitted Latched, synchronizing with first rise-up of SCLK. Next data is transmitted, synchronizing with fall-down of SCLK.
1	0	1	MSB data is transmitted, synchronizing with first rise-up of SCLK. Data is latched, synchronizing with fall-down of SCLK. Data is transmitted, synchronizing with rise-up of SCLK.
2	1	0	During H state, MSB data is transmitted. Latched, synchronizing with first fall-down of SCLK. Data is transmitted, synchronizing with rise-up of SCLK.
3	1	1	MSB data is transmitted, synchronizing with first fall-down of SCLK. Data is latched, synchronizing with rise-up of SCLK.

Clock polarity CPOL shows whether clock toggle starts from rising (CPOL=0) or from falling (CPOL=1) at the time of starting data transmission.

Clock phase CPHA shows the phase relation between clock and data, and CPHA=1 setting delays the data phase of CPHA=2 setting by 180 degrees behind. Fig.6-2 shows the relation of timing waveform between CPOL and CPHA.

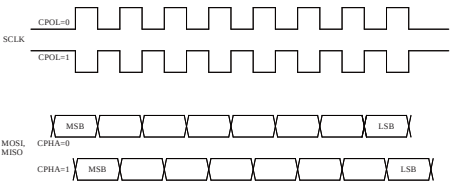


Fig.6-2: Timing waveform in SPI mode

Connection of FlashAir and LCD Module

This time I used "STN liquid crystal module (100x32) HQ077BC" available at Aitendo as of November 2014. This liquid crystal module uses KS0713 as controller. It can be driven by 3.3V single power supply and be controlled by SPI. SPI specifications are word size 8 bits and SPI mode 0. This liquid crystal module needs an IO control of five lines in total: reset terminal RESETN and register select terminal RS in addition to SCLK terminal, MOSI terminal and SS terminal. There is no MISO terminal. FlashAir also operates with 3.3V and has five GPIO lines, so it can be connected as it is. LCD module and FlashAir were connected like Table 6-3.

The terminal of this liquid crystal module has a flexible flat cable of 19 pins with 0.5mm pitch, so in order to convert to 2.54mm pitch, a breakout board is used. We added pull-up resister or pull-down resister on the CLK pin of FlashAir.

Table.6-3: LCD module connection

LCD module pin		Connection
1	VSS	GND
2	VDD	3.3V
3	CS1B(SPI SS)	FlashAir Pin2, CMD
4	CS2(L or Hf ixed)	H fixed
5	RS(0: command, 1: data)	FlashAir Pin7, D0
6	RD(L or H fixed)	H fixed
7	WR(L or H fixed)	L fixed
8	PS(0: serial, 1: parallel)	L fixed
9	RESETN	FlashAir Pin8, D1
10	MI(0: 8080, 1: 6800)	Lfixed
11	DB7(SPI MOSI)	FlashAir Pin9, D2
12	DB6(SPI SCLK)	FlashAir Pin1, D3
13 ～ 18	DB5 ～ DB0	NC
19		NC

GPIO control

The ruby script controls FlashAir GPIO state. It is running on the PC connected to FlashAir through WiFi. Whenever it changes GPIO state, it always transmits the HTTP request whose HTTP query string has the GPIO state. As Ruby “open” can handle HTTP and is blocked up until the processing finishes, the processing order is maintained. The following shows a code example to issue HTTP requests by Ruby “open”.

```
Code description example:
require 'open-uri'
...
open(sprintf("http://flashair/command.cgi?op=190&CTRL=0x1f&DATA=0x%02x",
        @clock<<4 | @data<<3 | @reset<<2 | @rs<<1 | @csb))
...
```

Continuously, we make reset cancel sequence, register write function and data transmission function to make sequence initializing LCD module

The liquid crystal module used this time has a whole liquid crystal divided into four regions called a page, and 100x8 dots constitute one page. Renewal of 1x8 dots unit in a page is possible with write access of one word of SPI. Addresses in the page are incremented automatically, so write access of 100 words makes one page drawing possible. Renewal of the whole screen of 100x32 dots needs sequence initializing liquid crystal module, write access of 400 words and four times of page address designation.

Drawing of Sora Hirameki with FlashAir

First we prepared a monochromatic image of 100x32 dots. Then, using ImageMagick and Cairo library, we took out pixel information of each drawing unit of 1x8 dots and drawing in order leads to Fig.6-3.

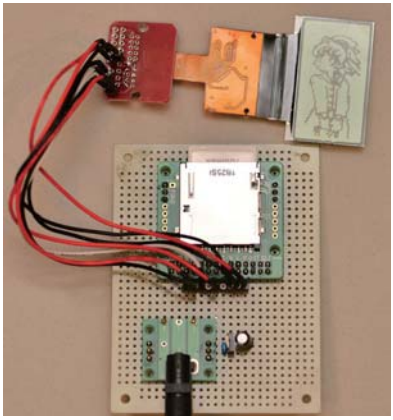


Fig.6-3: Display of Sora Hirameki with FlashAir

Hello, World ! with FlashAir

To perform displaying “Hello, World”, we made minimal ASCII font of 5x5 dots. With a space of 1 dot between characters, we tried to draw one character in 6x6 dots. With this font, we can draw maximum 80 characters in 16 characters x 5 lines on the screen of 100x32 dots. When you store the font data in array, storing in order of ASCII code is convenient to use the font data. The result of drawing became like Fig.6-4.

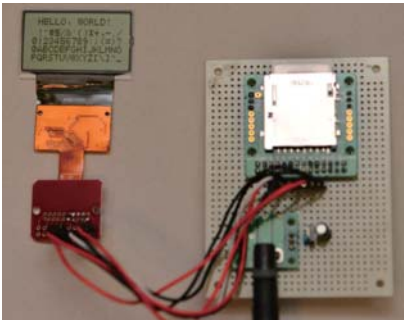
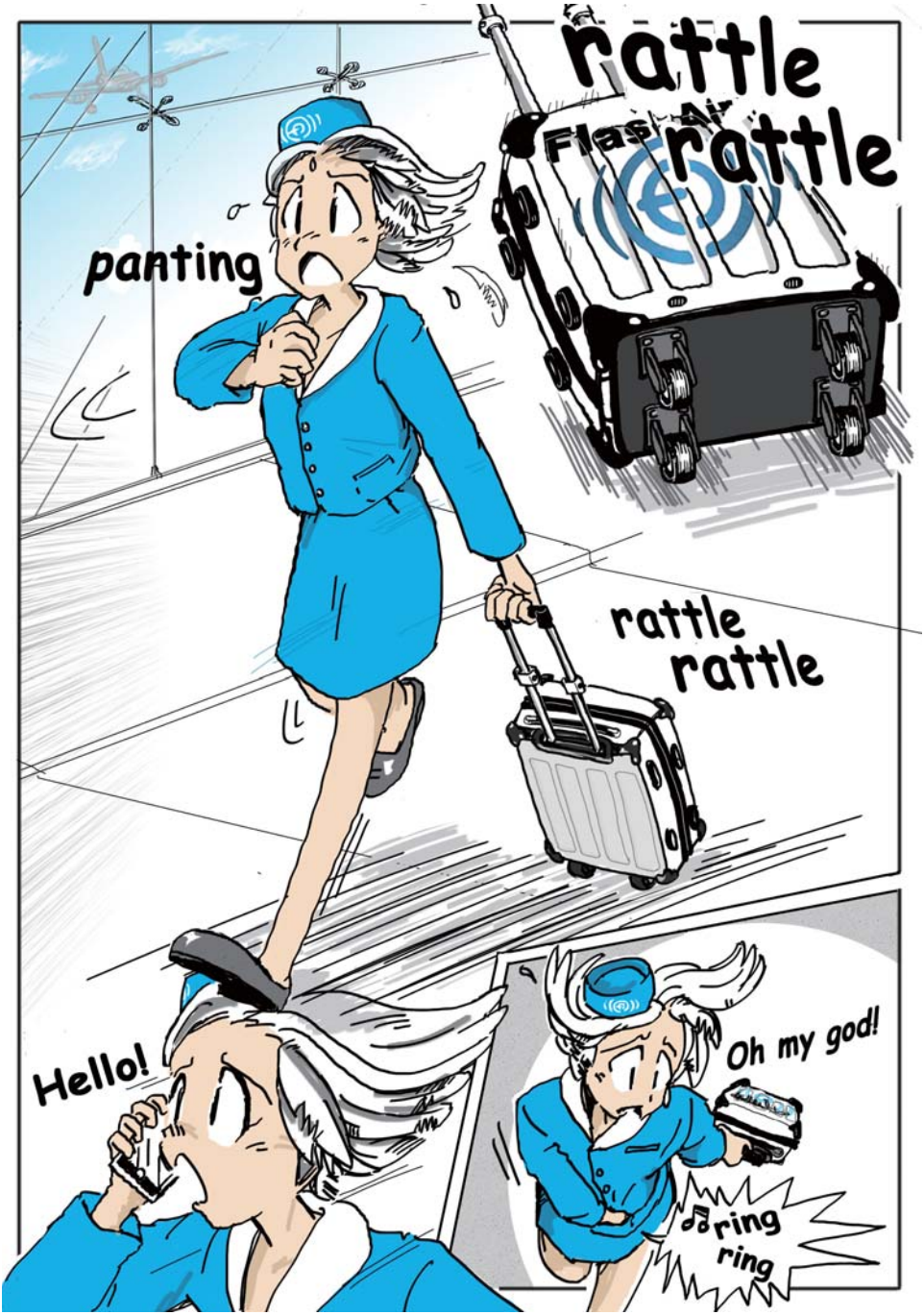
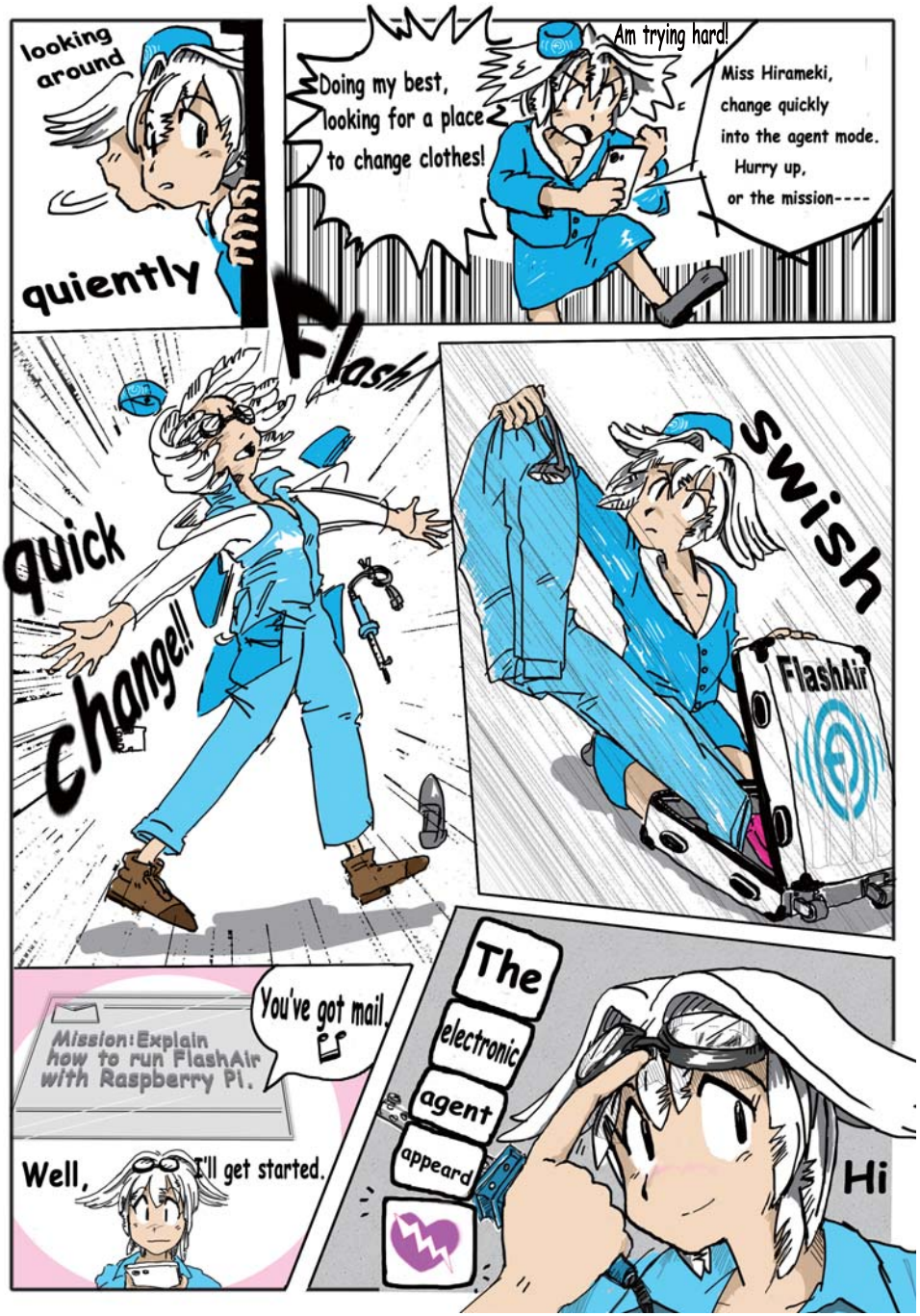


Fig.6-4: “Hello, world!” with FlashAir

SPI transmission rate by wireless GPIO of FlashAir

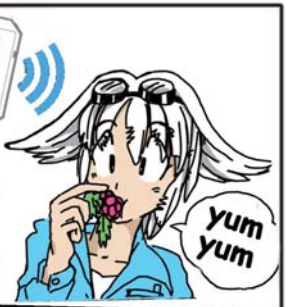
Software SPI using wireless GPIO of FlashAir needed 7724 times of IO toggle (HTTP request) to renew the whole screen of 100x32 dots and it took about 35 minutes. SPI transmission rate using wireless GPIO of FlashAir is about 1.8bps but there is no problem in using it for low-speed M2M.


rattle
Flash Air
rattle
panting
rattle
rattle
Hello!
Oh my god!
ring
ring

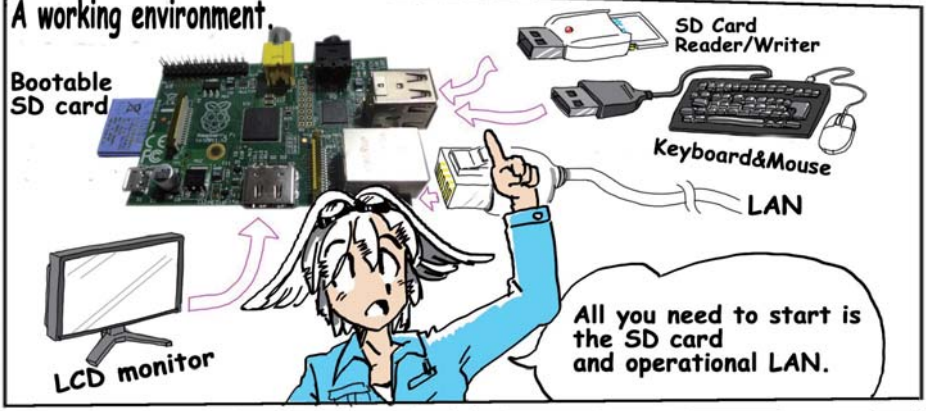

looking around
quiently
Doing my best, looking for a place to change clothes!
Am trying hard!
Miss Hirameki, change quickly into the agent mode. Hurry up, or the mission----
Flash
quick change!!
swish
FlashAir
You've got mail.
Mission: Explain how to run FlashAir with Raspberry Pi.
Well, I'll get started.
The electronic agent appeard
Hi

Let's run FlashAir with Raspberry Pi.

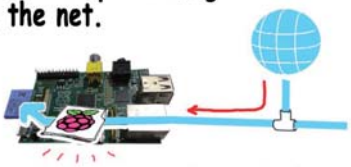
Please refer to the following URL, too.
https://flashair-developers.com/ja/documents/tutorials/users/1/



A working environment.



Download Raspbian image from the net.

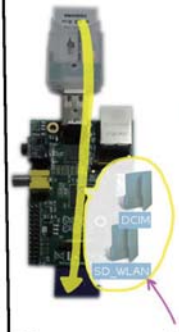


Copy the image file to FlashAir with the dd command.

<Example>

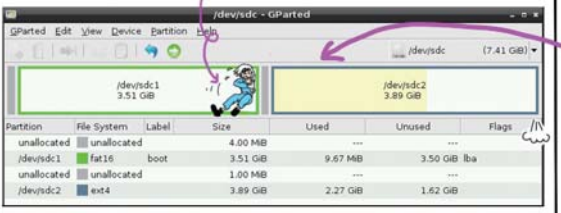
$sudo dd if=/home/pi/Desktop/flashair/
2014-09-09-wheezy-raspbian.img of=/dev/sdc

Insert FlashAir just initialized by "PC Setting tool" in TOSHIBA site into the USB reader/writer, and copy two folders ("/DCIM","/SD_WLAN") to the bootSD card side.

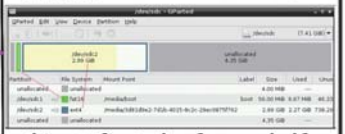


Please remember to be careful with the hidden folder "/SD_WLAN".

Using partition editing application (=gparted), change each partitioned area (sdc1/sdc2) of FlashAir free of mounting.

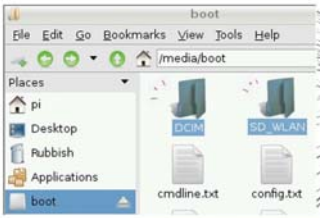


In the image development of Raspbian, the partitioned configuration was like



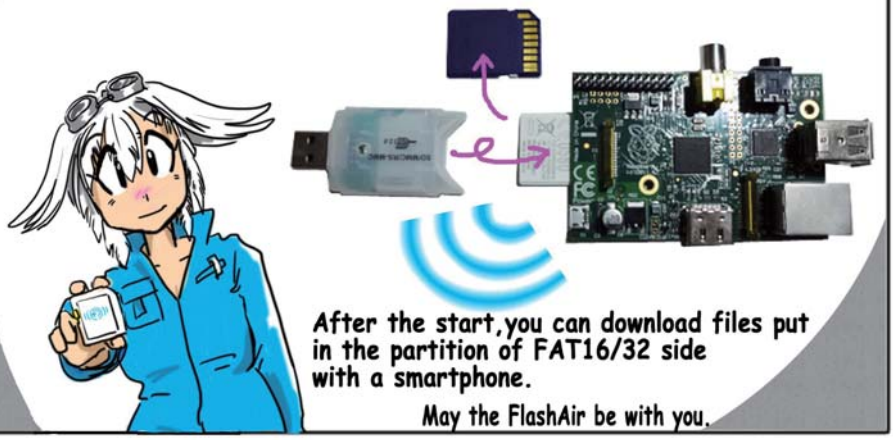
this at first:the former half of "FAT16"(yellow green) and the latter half of "ext4"(blue green).

It is something like putting ext4 area of sdc2 behind, and enlarging FAT16 area of sdc1 in the open space(*)



*FAT16 set at the capacity of over 4G get to have the format of FAT32.
At this time,"/dev/sdc1" file disappears, so all the data must be temporarily retreated and after formatting,
copy it and put it back if you want to do so.

Pull out FlashAir from the USB card reader/writer and reinsert it as the boot SD card of Raspberry Pi.
When Raspberry Pi is booted,FlashAir will start up too.



After the start,you can download files put in the partition of FAT16/32 side with a smartphone.

May the FlashAir be with you.

■ FlashAir Doujinshi

First Edition Aug. 11, 2015

Author: Takada, Ito, Pochio, yone2, Doi, Muraguchi, gm
Cover Illustrator: gm
Cover Designer: yone2, gm
Editor: yone2
Translator: Tama
Translation Checker: Peter Matthews, Alexandra Robin Clodge
Publisher: FlashAir Developers
Mail: support@flashair-developers.com
Printing: Souei Corporation
10 9 8 7 6 5 4 3 2

Postscript

This contribution was a good opportunity to consider the significance of our activities. It is a fond memory that I called a meeting which led to our activities, saying “a meeting with geeks in Akihabara”.

Takada

I want to do more with FlashAir.

Ito

I had very valuable experiences through this expanded sales activities, telephoning talent agencies, being interviewed by voice actors, etc. In the future I hope FlashAir will be used as a versatile built-in utility. Please put production examples in Twitter, attaching #FlashAir.

Pochio

I did my best in writing this and am glad if you could enjoy it. By the way, my main business is introduced in the back cover. I hope it will be of interest to you too.

yone2@Respon

Aiming at the construction of community, FlashAir Developers started and at last Doujinshi was published. I am so impressed and feel grateful. Please read Arduino tutorial in the website too.

Doi

FlashAir can be used as a wireless SPI Master too. I am happy if you feel excited at this expanse of possibility.

Muraguchi

What is different from Microsoft or GMO is that in-house production rate (*) of Character is 100%. This shows Toshiba spirit, doesn’t it?(*Of all the parts constituting our own company’s products, the occupancy rate of the parts manufactured by our own company without outsourcing)

gm

The FlashAir project got started by just a few colleagues. Big attractions of the Card brought many supporters in many places. We promise to develop the Card to more attractive product !

Sayako.K