Low Power Optimization for Wearable Devices

Siquan Hu[1,2, a*], Qianyue Zhao[1,b] and Chundong She[3,c]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China

[2]Chengdu Visual Microcosm Technology Ltd. Chengdu , 610072 , Sichuan, China

[3]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[a]husiquan@ustb.edu.cn, [b]qianyue365@163.com, [c]shurcd@vip.sina.com

Keywords: Low power consumption; Wearable device; EFM32; SIM800H.

Abstract. In recent years, the wearable devices are attracting more and more interests because of small volume, portability, multiple functions. However, such devices haven't had a significant development. One of the reasons is that the battery can't provide long time power supply due to the size and functional limitations. In this paper, we combined EFM32 and SIM800H to design a simple wearable device system, and researched the low power optimization from the perspective of software design. A communication mode between the master chip and SIM800H will be proposed in this system, which mainly about maximizing the working hours in low power mode. Experimental results show that the power consumption of this system is obviously reduced.

Introduction

With the vigorous development of the Internet, our field of vision is from traditional electronic devices such as PADs, Navigation, to the present portable like Glasses, Watches [1], and the demand for wearable devices are beginning to expand. As we know, most wearable devices are related to sports and human body features detection. In many cases, the wearable devices can provide a solution to monitor the vital parameters of a person during daily activities. Wearable devices can be used in many occasions, such as medical, sports and fitness. And one of the advantages of wearable devices is that they can be used to monitor the health status of the elders [2].

Now a lot of wearable devices, which have rich functionality, have emerged in the electronic market such as Smart Bracelets, Watches and Glasses, but some of the user experience is not very good. A short battery life significantly restricts the widespread use of these devices. There are two critical aspects for the wearable technologies: reliability and battery lifetime. Several wireless communication protocols for low power communication have been developed [3,4]. Many electronics companies are also looking for low power design strategies.

For these reasons, this paper investigated the power consumption behavior of each module in a simple wearable system, and proposed some strategies to reduce system power consumption mainly from the software design. This research will be useful on the development of low-power technologies.

System Design

We only focus on the software optimization. The whole system can be divided into three modules:

Microcontroller. EFM32G230F128 is the master chip, it is based on ARM Cotex-M3 kernel. The EFM32G consumes as little as 180 μA/MHz in run mode, and 900 nA with a RTC running [5].

Communication Module. SIM800H is a complete Quad-band GSM/GPRS solution and supports Quad-band 850/900/1800/1900 MHz. We use it to communicate with master chip and server, mainly about transmitting voice, SMS, location and other information with low power consumption. However, the low power optimization mainly aimed at SIM800H. In addition, the way SIM800H communicates with master is AT commands.

Sensor and Display modules. GPS, LCD and sensors are the third module, they are mainly to collect location information, display useful items and other health information.

Due to the demand of low power design, the master controller doesn't have too much process of dealing with received data, which will be transmitted to the server through the SIM800H. The work process mainly includes the following steps:

Initialization. The initialization work includes EFM32 and SIM800H, especially for the SIM800H. Before working, we need to determine whether the SIM card in SIM800H registered to the GSM network, and whether it can access to GPRS network. Other modules won't be initialized until used.

Collecting Data. We need to collect base station information, GPS data, sports and pulse data. Each module collects the necessary information after receiving the instruction from the master controller.

Transporting Data. For every five minutes, the master will collect data, and send it to the server according to the prepared protocol via SIM800 every half an hour, mainly use GPRS.

Low Power Optimization

Software structure. A struct is designed to encapsulate the system functions, and a loop to check the time task queue and handled each task, the timer is Systick. The main function is shown in Fig. 1.

```
typedef struct
{
                long cnt;
                long limit;
                bool finish;
                void (*function)(void);
}Task_t;
Task_t PulseTask = {0, PULSE_DETECTION_COLLECT_PERIOD/SYSTICK_UNIT, false, PulseCollect};
Task_t MPU6050Task = {0, SLEEP_DETECTION_COLLECT_PERIOD/SYSTICK_UNIT, false, SleepDetectionCollect};
Task_t SIM800TransTask = {0, SIM800_TRANS_PERIOD/SYSTICK_UNIT, false, SIM800TransInfo};
Task_t SIM800TimeTask = {0, SIM800_TIME_PERIOD/SYSTICK_UNIT, false, SetTimeNeedUpdate};
while (true)
 {
      for(int i = 0; i < TASK_NUM; i++)
      {
          if(TaskPtrArray[i]->finish == true)
          {
              TaskPtrArray[i]->finish = false;
              TaskPtrArray[i]->function();
          }
     }
     EMU_EnterEM2(false);
 }
```

Fig. 1 Main function code

Low power strategy. We considered from the following aspects to optimize the power consumption: clock frequency, GPIO, the running time of the active mode.

Clock frequency is nothing more than fast or slow. Low frequency means long working hours, while the high will be fast but the power consumption will increase. Because most of the peripherals we use can run in low power mode, we choose the low frequency clock—LFA_Clock.

To open GPIO or not has a great impact on power consumption. If a IO stay in openning but not be used, the interference level will affet the power consumption and the performance of the system. So we don't enable them until we use, and disable them after using. All unconnect pins of the EFM32 should be configured to be disabled through setting the GPIO->P[x]. MODEL/MODEH to 0.

The system communication core is the SIM800H, and it consumes a lot of electricity. We connect it to the server for transmitting. The transmition process will be affected by network signal intensity, so we use TxStatus and RxStatus flags to control the communication between the master and SIM800H. After sending AT commands to the SIM800H, there will be a period of time waiting for SIM800H's feedback. The power consumption will increase if the EFM32 is waiting in the full speed running state. Considering that the serial port (LEUART) we use can work in low power mode, so we let EFM32 enter into EM2 (low power mode) in the waiting time. LEUART doesn't work until it detects signls. The process of sending AT commands is shown in Fig. 2. This strategy can greatly reduce the running time of the active mode, so as to reduce the power consumption of the entire system.

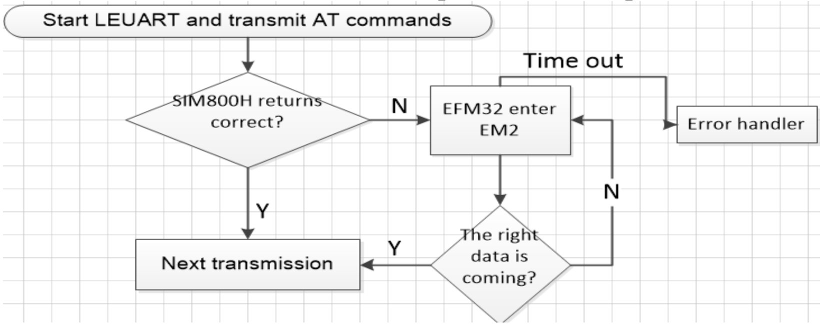

Fig. 2 Send AT commands to SIM800H

The data transmission mode is also a problem to be considered. We use KEYSIGHT N6705B to monitor power consumption, the effect is shown in Fig. 3 and Fig. 4. LCD and other module work as usual because they have already been in low power operation mode.

●Sim800H obtains the location information and send data to server every five minutes, then it will enter sleep mode. At the same time, EFM32 will enter EM2 mode.

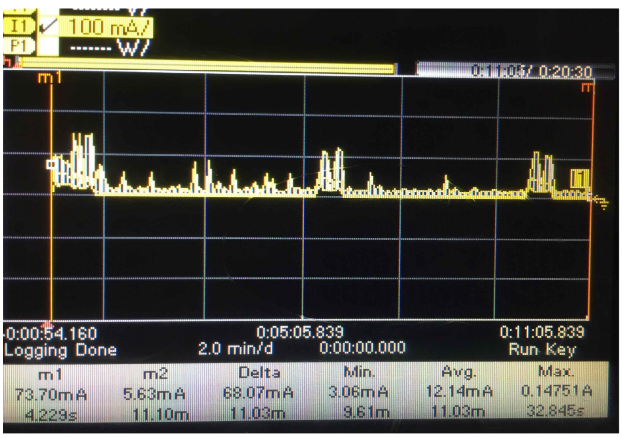

Fig. 3 Not disable LEUART after sending

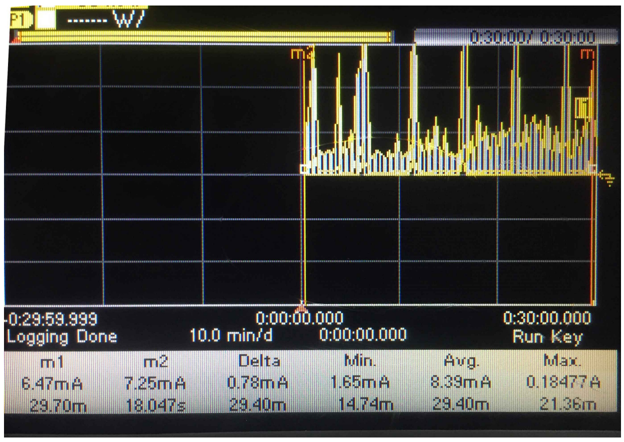

Fig. 4 Disable LEUART after sending

●Sim800H obtains the location information and send data to server every five minutes, then it will power down. At the same time, EFM32 will enter EM2 mode. The power consumption of this process is shown in Fig. 5 (from now on we use line charts to compare).

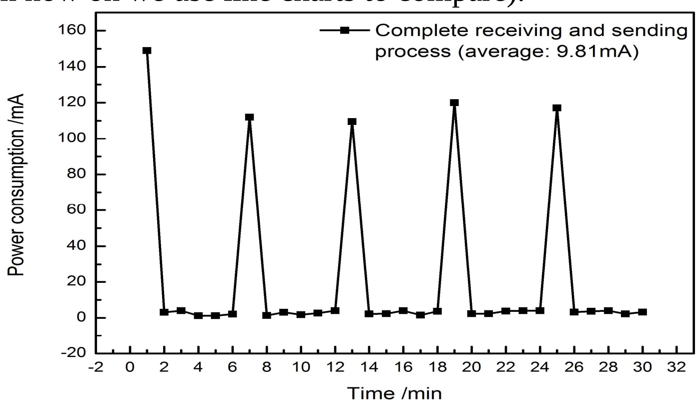

Fig. 5 Power down SIM800H after sending

●Sim800H obtains the location information without GPRS every five minutes, then it will enter sleep mode. At the same time, EFM32 will enter EM2 mode. These data will be sent to server every half an hour. Fig. 6 and Fig. 7 show the power consumption of this process.

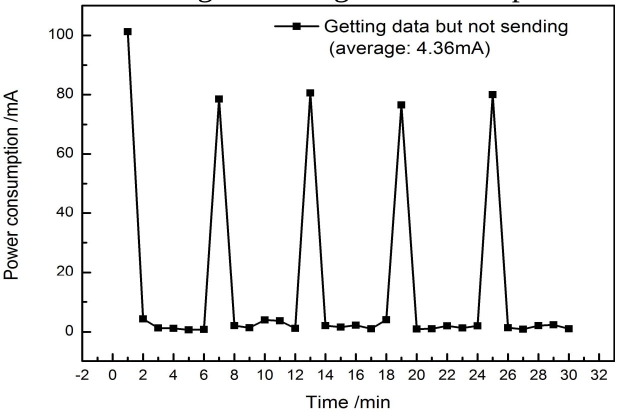

Fig. 6 Locate but not send data

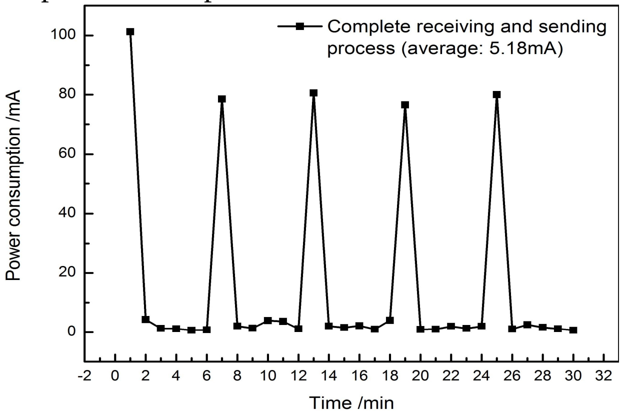

Fig. 7 Send data every half an hour

It is meaningless to upload data every five minutes if users don't request it. But we need to record the various states of users, it is enough to upload every half an hour (six groups of data).

Results and Conclusions

We have successfully designed a wearable device system, and took some strategies to do the low power optimization job. The pictures above showed the results of our work.

By closing the IO and peripherals that are not be used, the average power consumption is reduced from 12.14 mA to 8 mA. There is a fact that SIM800H also interacts with GSM network even if in sleep mode. So we try to power down it after transmitting. However, when SIM800H powers on, the behavior of searching for the network will increase the power consumption (the average is 9.81 mA). The last we found out that we can obtain the location information in the case of closing the GPRS, which will save most of electricity. The average power consumption is 4.36 mA, and 5.18 mA when we collect six groups of data and uploaded them.

We have to emphasize that all the data we have measured was include the process of starting the network state, but the actual operation only needs one start-up process, the power consumption should be lower in actual work.

Acknowledgements

The study was supported by the National Natural Science Foundation of China (No. 91438120) and the Sichuan Science and Technology Support Program (No.2016GZ0339).

References

[1] M.H. Cho, C.H. Lee: IEEE Trans on Consumer Electronics Vol. 56 (3) (2010), p. 1602-1609.

[2] A. Dionisi, D. Marioli, E. Sardini and M. Serpelloni: IEEE International Symposium on Medical Measurements and Applications, MeMeA 2015-Proceedings (2015), p. 537-542.

[3] C.H. Wang, W. Lu, M.R. Narayanan, S.J. Redmond and N.H. Lovell: Biomedical Engineering Letters. Vol. 5 (1) (2015).

[4] D. Manstretta: 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2010), p. 3437-3438.

[5] Information on http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32-RM.pdf.