Poster: SoEasy - A Software Framework for Easy Peripheral Control Programming in Diverse Hardware Platforms

Kwang-il Park, Jong-ha Shin, Jin-hae Lee, Seong-eun Yoo*
School of Computer and Communication Engineering
Daegu University, Gyeongsan, Gyeongbuk, Korea
{pki987, prattlesjh, centipede486}@live.daegu.ac.kr, seyoo@daegu.ac.kr

Abstract

With the help of widespread of open-source hardware platforms, many IoT (Internet of Things) applications are emerging and evolving rapidly. Most of high end open-source hardware platforms include built-in peripherals such as UART (Universal Asynchronous Receiver and Transmitter) and GPIO (General Purpose Input Output) ports, and have enough computation power to run embedded operating systems such as Linux. However, each hardware platform has its own way to configure each peripheral, and it is difficult for a programmer or a user to configure it. Therefore, we propose an easy and convenient way to configure each peripheral using a web-based software framework for programming and configuring the operation of the peripherals. Through the implementation, we show the feasibility of the proposed software framework.

1 Introduction

Recently, open-source hardware platforms are widely used in embedded systems such as Internet of Things. There are various kinds of open-source hardware platforms from the low end 8-bit microcontroller based ones to the high end 32-bit microcontroller based ones. These hardware platforms had been expensive in the past, but recently they are getting cheap and obtained easily thanks to the open-source distribution policy[1]. So, everybody, whether he is a beginner or an expert in embedded systems, is easily able to start to design and implement his own IoT application. However, when he starts to develop the application, he needs to know the detailed hardware architecture of the hardware platform, but it is not easy for a general user to configure each peripheral which is configured in a different way depending on the hardware platforms. For beginners who are unfamiliar to embedded software programming, Arduino was released, but a programming illiterate user is still difficult to develop an application with it. So we propose *SoEasy*, a software framework for an easy peripheral control programming in diverse hardware platforms envisioning that everyone even inexpert can make an IoT application as one writes a document easily with a word processor. To evaluate the software framework, we implement the proposed framework, *SoEasy*, and demonstrate the easiness and convenience of the peripheral device control programming of *SoEasy*.

*Corresponding author.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria

ISBN: 978-0-9949886-0-7

2 Purposed Software Framework

One of most popular user interfaces is a web, and we adopt the web technology as a user interface of *SoEasy*. Since most of users are familiar with a web interface, a user can make an application through the web interface easily and conveniently without writing any software program in *SoEasy*. In addition, any device with a web browser can be a development host or a monitoring device. *SoEasy* framework consists of Database, Web Interface, and Control Program as in Figure 1. Web User Interface with the help of a web server provides users with an intuitive GUI and helps inexpert users configure peripherals and make an application easily. Database stores and manages all the information for *SoEasy* including the configuration values for each built-in peripherals such as GPIO controller and add-on function implementations. Control Program is a service agent program that controls the hardware platform according to information in the database.

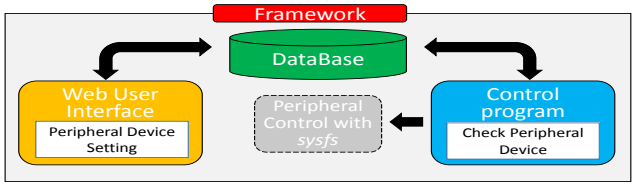

Figure 1. *SoEasy* Framework

3 Implemention

To evaluate *SoEasy* framework,we implement it as Figure 2 in Linux for two different open-source hardware platforms: Intel Galileo Gen1/2 and Raspberry Pi.

3.1 Control Program

Control Program is to perform the functions that are selected through the Web User Interface. Based on the infor-

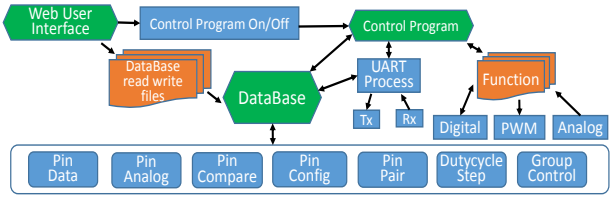

Figure 2. *SoEasy* Detailed Architecture

mation stored in the database, it configures the built-in peripherals and control registers to provide the user with the specified functions such as PWM (Pulse Width Modulation), Digital I/O, Analog Input, and UART according to the user configuration. To do that, Control Program accesses *'sysfs'* [2], a virtual file system provided by the Linux kernel, modifies the information in the files in order to reflect the user request. Control program spawns an independent child process for UART interface to prevent any loss of UART data.

Control program modifies the exported files in *sysfs* using low-level file input and output as follows. For GPIO configuration, the control program exports the selected GPIO pins in order to create a symbolic link for the GPIO pins in GPIO path in *sysfs* directory. By writing *'in'* or *'out'* in direction file for the pin in *sysfs* according to the direction specified in the database, the pin is configured to input or output mode, respectively. To read from or write on the pin, Control Program reads the *'value'* file and store it in the database or it writes the value file in *sysfs* with the value from the database. For an analog input, Control Program reads the analog value from analog path in *sysfs*. For PWM, Control Program opens period and duty_cycle files in PWM path in *sysfs* and sets the period and the duty cycle according to the specified values. For UART, Control Program uses the APIs in WiringPI library[3] to operate the UART functionality, can communicate via UART by reading and writing UART path in *sysfs* .

3.2 Web User Interface and Database

Users can configure and program the hardware platform using Web User Interface, and the configuration data is written to the database. Users set the mode (Digital, Analog, PWM, and UART) of each pin with *pin_mode*. *pin_data* stores the digital input value from a pin or the digital output value to a pin. *pin_analog* saves the analog input value from a pin. If *pin_mode* sets to PWM, dutycycle_step keeps a duty cycle changing step used in UI.

To facilitate the development of diverse projects, UI of *SoEasy* provides various services. Pin pairing service connects a source pin to a destination pin, logically. For example, if the *pin_data* of a source pin is '1', the *pin_data* of the paired destination pin is updated to the same value as the *pin_data* of the source pin. Pin Compare service is to control a target pin when the *pin_data* of a source pin meets to the condition. Using the service, we can easily implement if-else like statement. Pin Grouping service provides the multiple pin control service. The pins listed in *group_control* are controlled together.

Figure 3 shows an Atmega128 source code of 133 lines for a project to control an RC car while Figure 4 shows the web interface for *SoEasy*.

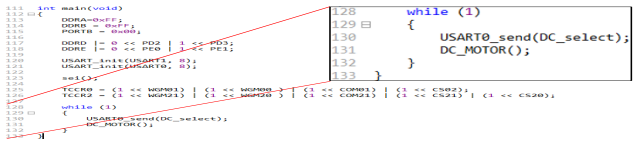

Figure 3. Conventional Programming Source Code

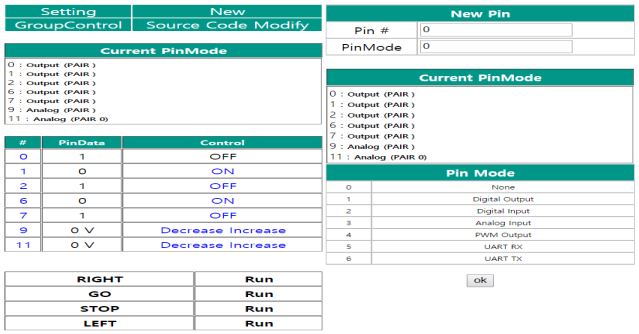

Figure 4. *SoEasy* Programming Interface

3.3 Advanced function

SoEasy offers an interesting solution for an expert user to modify the existing function or service, or to add a new function or service. That can be possible by allowing the user to edit the source codes through the UI, and to run make utility to compile *Control Program* with the help of PHP scripts. This function provides *SoEasy* with the unlimited extensibility.

4 Conclusions

Widespread open-source hardware platforms facilitate the development of diverse and novel IoT applications. However, programming on a various kinds of hardware platforms is still difficult for inexpert users, since depending on the hardware platforms the ways of configuring I/O peripherals are different. To ease an application development of beginners, a software platform for easy programming, *SoEasy* is proposed. Implementation result shows the feasibility of *SoEasy* as an easy and convenient programming environment. In the past, an application development needs editing source files and compilation process to make an executable file while *SoEasy* provides an easy programming environment based on widespread web interface with a few clicks. Another useful feature of *SoEasy* is a monitoring function. Without any mobile application, *SoEasy* provides a monitoring environment based on web technology.

5 Acknowledgments

This research was supported by Basic Science Research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. 2011-0014204)

6 References

[1] Open source hardware associatoin(oshwa). http://www.oshwa.org/faq/.
[2] P. Mochel. The sysfs filesystem. Linux Symposium, 2005.
[3] WiringPi. *GPIO Interface library for the Raspberry Pi*. http://wiringpi.com/.