

An Embedded Integration Prototyping System based on Component Technique

Youngjin Jung¹; Jeongbae Lee¹; Jinbaek Kwon¹; Keewook Rim¹; Sangyoung Cho²

¹Department of Computer Science, Graduate School, Sunmoon University,
Kalsan-ri, Tangjeong-myeon, Asan-si, ChungNam, 336-840, Korea,

²Department of Computer Science and Engineering, Hankuk University of Foreign Studies,
89, Wangsan-ri Mohyun, Yongin-si, Kyounggi-do, 449-791, Korea
yjung.kr@gmail.com, {jblee, jbkwon, rim}@sunmoon.ac.kr, sycho@san.hufs.ac.kr

Abstract. Nowadays in the development of embedded system, cutting-edge embedded system products are quickly disappearing from the markets because of their short product development period which shortens the product life cycle. Therefore, strengthening its competitiveness and minimizing its development cost can be said to be one of the most important factors. For this motive, an Embedded Integration Prototyping (IP) system based on Component Technique was designed and implemented through this paper. The system is composed of Physical Prototyping (PP) providing the environment in which the product can be tested by using Actuator(Motor), Sensor and reusable Blocks, and Virtual Prototyping (VP) in which visual test on the product can be carried out by applying various components and libraries based on technique related to the computer. And, IP System was built in order to mutually compensate for drawbacks latent in both of physical and virtual prototyping environment by making use of component module. The module will be able to enhance the product competitiveness, through spending less time in developing kinds of the component owning almost same features, using it again for different embedded system products, and accordingly minimizing spent cost and time for developing the component.

Keywords: Component, Embedded System, Integration Prototyping (IP), Physical Prototyping (PP), Virtual Prototyping(VP), Simulation

1. Introduction

The Embedded System has been widely used in a diversity of industry fields including military affairs, aerospace, information appliance, etc. In general, it is the complicated combination of hardware and software equipped with unique operation environment, independent architecture, special interface, and so on. And, because embedded products applying latest technology have short life and development cycle for them, they are disappearing from the market in fast speed, after their launch into the market.

As result of these reasons, prototyping technology used for developing embedded system product was needed to strengthen its competitiveness, with lessening the cost to be spent for initial development stage, as well as applying Time-to-Market shorter than before [1]. And, any problematic matters or errors possibly to take place at built-in function and hardware after product's launching into the market can be found in advance by using prototyping technology and applying it in the product's features and function from the stage of designing hardware. Besides, such found errors can be immediately modified at found time points to carry out test on them again. Like this, prototyping technology enables product to have higher level of reliability and performance until its launch by gradually repeating these series of process. Prototyping technology already known until present includes: PP in whose technology real embedded system built in factory automation system, vending machine, washing machine, mobile phone, etc is tested and produced by using Embedded System Prototyping Suit (ESPS)[2]; VP in whose technology using computer-related technology 3D model product can be visually created by providing libraries and various components; IP under currently active research in whose technology physical and virtual prototyping can be integrated. But, PP technology can't support diverse components and libraries to closely control small-sized appliances built with embedded system. On the other hand, VP technology can't support practical test environment for embedded system, even if it is possible for the system to execute virtual simulation with using computer technology. IP system at current times is implemented to give and receive only simple data, by using communication technology between physical and virtual prototyping. Besides, to be troublesome, it needs additional tasks initializing and declaring data set up in actuator and sensor to let the system operate.

To solve these problems, Component-based IP system was designed and developed at this paper in order to make it easier and promptly to reuse actuator and sensor built in currently developed IP system to different embedded systems. This system set up Double Rock Spin (DRS) system built in embedded system as its target, and constructed PP environment using ESPS, while it made use of RapidPLUS[3] tool visually providing 3D object to carry out simulation, and constructed VP environment. The remainder of this paper is structured as follows. In the next section, we discuss related works on embedded system prototyping. In section 3, we describe the design of our integration prototyping component system. Section 4 presents the implementation of integration prototyping component system for DRS, and section 5 provides a final discussion on our work and plans for future research.

2. Related Works

2.1 Physical Prototyping

A Prototype is miniature or real sized model product made during product development prior to production. A prototype is applied to test product's appearance and performance before its launch, and belongs to the part of production. As computer technology is developing in recent days, the process in which testable prototype equal to real product is created by applying advanced visual and engineering technology based on computer is called "Prototyping" [4-6]. PP can produce and test practically embedded product such as model of factory automation system or vending machine. To do so, PP supports operation environment similar to target model, by modifying real-time operation system, Real-Time(RT) Linux kernel and producing board and device driver for turning the product into drive. And, it can be applied as a tool of promoting mutual understanding among professional developers in various fields required due to the characteristic of embedded system in order to work together and to invent efficient solution to settle problems. Using PP allows hardware and software developers to carry out respective tasks at the same time in the middle of development stage, and allows problems possibly occurring (if software may be applied) to be directly communicated to hardware developer in order to solve them, as well as allowing developers to improve in their mutual understanding. In addition, it guarantees end product's quality from its development stage. This PP enables smooth communication among developers, together with lessening cost, and thus has merit in product's price competitiveness, whereas it can strengthen ability of coping with any faults in hardware found at the stage of developing software. But, PP environment can not completely support various components or fully controlling libraries of small information appliance product equipped with embedded system program, unlike VP environment can do. Lots of development tools have been studied and developed, as importance has been put on this PP concept in recent years.

2.2 Virtual Prototyping

VP using computer technology provides varied components and libraries. It supports visually simulating real product with visual 3D modeling on the computer in order to allow for the convenience in design modification and full control. Accordingly, with technology development and application expansion related to computer, engineers currently make use of VP technology in developing complicated systems. Current physical prototype needs new prototype in order to change product's appearance and adding another function, but virtual prototype can easily change design by using modeling method on GUI on condition that the design still exists on the computer. And it also can add new function without difficulty, because a function is linked to appearance expressed in formal specifications. But, it can not guarantee certainty of correct simulation in embedded system environment, because it adopts visual simulation with sense of sight using computer technology. VP like this is utilized in developing appliance, designing car audio system, building visual factory, designing airport's control system, other validation field, etc, and is expected to enhance productivity by developing product through it in most of production realm, afterwards. Tools of developing it includes RapidPLUS, ASADAL[7], Rhapsody[8] and Virtio-made Virito[9].

3. The Design of Integration Prototyping Component System

This section explains the architecture built up in component-based IP system to be applied to various embedded system fields. Because this component module exists in the middle of physical and virtual prototyping environment, it can add, delete, or modify actuator and sensor component's data and property set through TCP/IP Socket. Data values set in component module and API(Application Programming Interface) already produced for actuator and sensor are used in PP environment. User-Defined Object (UDO) and RapidPLUS are used to be applied to values set in component module in VP environment. Fig. 1 shows the architecture of component-based IP system.

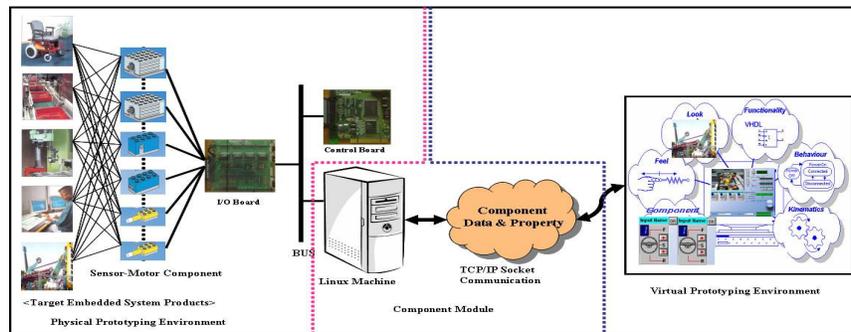


Fig. 1. The Architecture of IP Component System

3.1 The Design of Physical Prototyping

This section explains architecture of PP environment required for IP component system. This system constructed simulating environment turning actuator and sensor into operate, by using ESPI-API and data set in Component Module. Fig. 2 below displays the architecture of device driver and API for controlling several SMC(Sensor Motor Controll) devices. Each SMC device driver acts like real-time task for operating each device in RTLinux environment. This device drivers control sensor and motor to operating via I/O port, Interrupt and DMA to be assigned to each oneself. Also, each device driver communicate with user layer, higher-layer via RT-FIFO. And SMC API has Application Programming Interface for using easier device driver of user application.

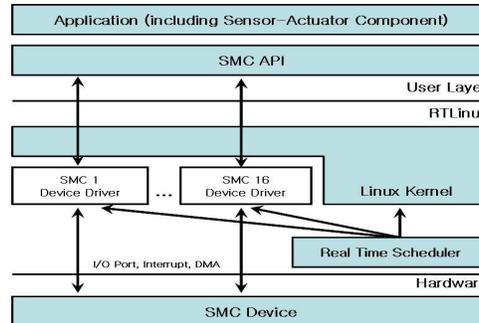


Fig. 2. The Architecture of Device Driver and API

Fig.3 describes the role of device driver and API for PP. In user level, ESPS-API carries out its role in connecting RT-Task and Linux process. RT-Task in RT kernel carries out its role as device driver. RT-FIFO is needed as method of communicating to this device driver. ESPS-API existing in kernel level executes its communication to RT-Task, by internally using FIFO. ESPS-API was developed so that user could easily control LEGO board. ESPS-API can directly call LEGO device's driver task from kernel level. It is available for precise control, but has difficulty in programming.

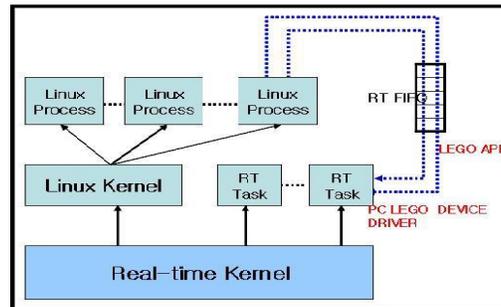


Fig. 3. The Role of Device Driver and API for PP

3.2 The Design of Virtual Prototyping

This section explains the construction of VP environment, by using RapidPLUS - tool available for simulating embedded system product on User Interface (UI). Target product is designed with RapidPLUS, as the process steps below.

- ① Placement: Place various objects provided for the appearance of targeted embedded system product, dependent on the Layout.
- ② Design: Use objects to create Product appearance, and then design Mode, Transition, Trigger, Activity, etc to turn product into work.
- ③ As final step, use prototype provided in VP environment to test former process and simulate product with debugged errors and problems.

Fig.4 expresses the relationship between application and UDO for constructing VP component to be built in IP component system. It is possible to create/control actuator and sensor in PP environment by setting up component in VP environment using UDO.

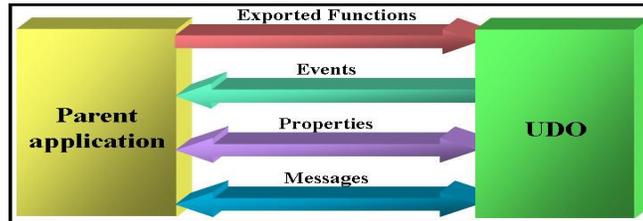


Fig. 4. The Relationship between Application and UDO

- Exported Functions: Sole function usable from other application
- Events: UDO reports its status change to parent application.
- Properties: Some types of data on object held.
- Messages: Able to send message to both of UDO and parent application by defining its structure type.

3.3 The Design of Component Module

This section describes component module connecting physical respective PP and VP components designed as above. Fig. 5 shows the design of component module between PP and VP. Component module for actuator and sensor was designed by using PP and VP constructed in former clause. Component module is organized by each class including actuator, sensor, and SensorMotor Component, as shown in class diagram in Fig. 5. The Class is composed of:

- Actuator Class: Holds properties related to actuator channel number, direction, and speed;
- Sensor Class: Sensor channel number and direction;
- SensorMotor Component Class: Holds objects of Actuator and Sensor Class.

This system was designed to create and control components in PP and VP environment, by using three classes set up in component module.

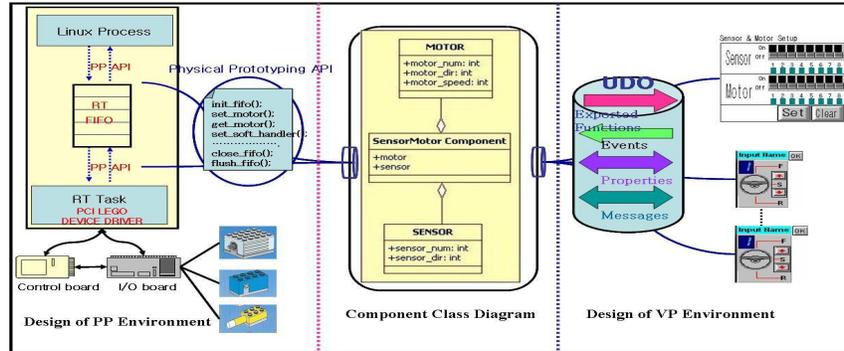


Fig. 5. The Design of Component Module between PP and VP

Fig. 6 describes the architecture of sensor and actuator's component for IP system. This component has property for changing specific status(channel, dir and speed) of sensor and actuators and interface(sm_stop, sm_start, sm_pause, sm_restart, and set motor, and so on) for setting/getting status of the component between physical and virtual environment. And it has event for setting sensor and actuator's status or using or calling some information from outside.

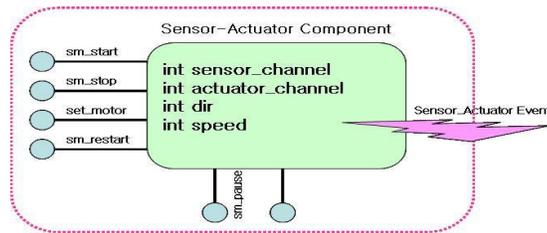


Fig. 6. The Architecture of Sensor and Actuator for IP System

Also, component module classes require data specification for connecting between PP and VP. This data specification is shown in Table 1. The data specification is classified by Component, Function and Property for Actuator and Sensor.

Table 1. The Data Specification of Component Module for IP System

Type	Factor	Definition
Component	Actuator/Sensor	Select Actuator/Sensor Component
Function	Initialize	Initialize properties of Actuator and Sensor
	Start/Stop	Set up Start/Stop Function
	Order	Set up Auto/Manual Mode
Property	Num	Set up Actuator/Sensor's Channel
	Dir	Set up Actuator/Sensor's Direction
	Speed	Set up Actuator's Speed

4. The Test and Evaluation of Integration Prototyping Component System

This section describe the implementation of PP and VP for setting up targeted product as DRS among embedded system products, and implementation of IP component system through designing component-based system. We also describe the results using component-based IP System in this section. Fig. 7 shows the test scenario for component-based IP System. Above all, we select component to be used and component's channel number (①-Selection). After that, we connect to PP and VP each using socket communication for the selected components and then set up direction and speed (②-Setup). Lastly, we operate function such as initialize, start, stop, and pause (③-Operation).

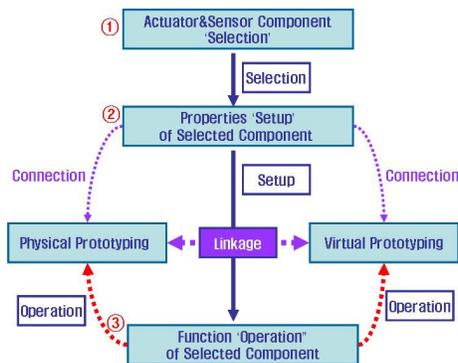


Fig. 7. The Test Scenario

4.1 System Test

(1) Physical Prototyping Side

Fig. 8 below displays PP for DRS already developed toward IP component system. In this PP, four actuators are used, and they are operated by ESP8266-API. Actuators included here works so that DRS can revolve inward and outward. And sensors are used to control work of the above defined motors.

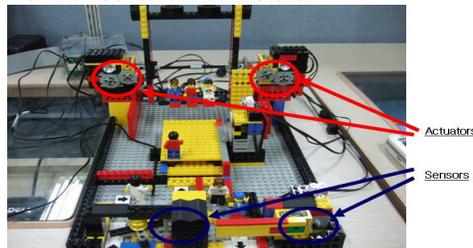


Fig. 8. The DRS PP

(2) Virtual Prototyping Side

For DRS, VP can more easily and faster implements UI than PP, by realizing it with using computer technology. Besides, it can test functions like almost real product, and can easily find and modify problems. Fig. 9 describes functions of Actuator and Sensor Component UI and shows that actuator and sensor components of VP environment required for the system are selected and setup. In this system, we use channel number 1, 2 for Actuator and Sensor respectively. Because each of these components has its own individual property and data, it is needless to create component again, for other embedded system products. Furthermore, it will be able to lessen time and cost to be spent in producing embedded system product, because it reuses components.

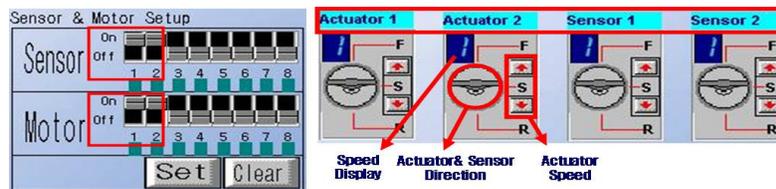


Fig. 9. Selection (left) and Setup (right) Actuator and Sensor Component UI

4.2 Evaluation

Table 2 below defines value, property and function of used actuator and sensor component in IP System. We use two motors and two sensors to handle the motor work in this system. And this system initializes value of defined actuator and sensor component's property. It also defines Start/Stop/Pause functions about work of the system and function of order operation such as Auto and Manual Mode.

Table 2. The Used Simulation Data in IP System

Type	Factor	Descriptions (Value)
Component	Sensor	Select Sensor Component
	Motor	Select Motor Component
Function	Initialize	Initialize values of the above Actuator and Sensor (4)
	Start/Stop	Set up Start/Stop (1/2)
	Order	Set up Manual Mode (2)
Property	Num	Set up available Sensor/Motor's Channel(1,2/1,2)
	Dir	Set up selected Sensor/Motor's Direction (0,1)
	Speed	Set up selected Motor's Speed (0x0, ..., 0xF)

The result of simulated data shows that components implemented between PP side and VP side are synchronized. And we can simulate fast and easily component-based IP system to handle operation and change direction and speed of selected component using setting Actuator and Sensor components.

5. Conclusion and Future Studies

Previously, IP system was an interlocking system to simulate target product in virtual and physical environment. In developing embedded system, strengthening its competitiveness and minimizing its development cost can be said to be one of the most important factors. For this motive, Component-based IP System was designed and implemented through this paper. Implementing component-based system using sensor and actuator that is highly reusable could make it possible to easily and fast implement IP systems for various embedded system products. Reliable embedded system products pertinent to Time-to-Market will be able to be launched into markets, by applying component-based IP system. In addition to it, it will make contribution to strengthening product competitiveness, too.

Implemented component-based IP system will be made in formal specification to expand its application to more industry fields, and debugger finding and then modifying any errors will be additionally developed, afterwards. And, further studying the method of assessing functionality built in embedded system product is for future study plan.

Acknowledgment. This study was supported by the Ministry of Information and Communication of Korea under the Information Technology Research Center Support Program supervised by the Institute of Information Technology Assessment (ITA-2006-C1090-0603-0020)

References

1. Gaver. B., Dunne. T, and E.Pacenti. Cultural Probes. *Interactions* 6(1), 1997, pp21-29.
2. ESPS, <http://www.artsystem.co.kr>
3. RapidPLUS, <http://www.e-sim.com>
4. Jung-Hee Song, Institute of Information Technology Assensment, "Korea Information Science Society," vol. ISSN 1229-6821, no. 23, pp. 7-11[*Dig. 9th Annual Conf. Magn. Jpn.*, p. 301, 1982].
5. Lawson. B, "How Designers Think," *Architectural Press.*, 1997.
6. SONY, AIBO, <http://www.sony.net/Products/aibo/>
7. ASADAL, http://selab.postech.ac.kr/realtime/public_html/
8. Rhapsody, <http://www.ilogix.com>
9. Virtio, <http://www.virtio.com>