Introducing Integrated Development Tool “MicroPecker”

Sunny Giken, Inc.

MicroPecker is a tool platform with a built-in single-wire interface that aims to address the specifications which is being promoted by the standards body JASPAR. It supports FUJITSU’s single-wire interface and provides an optimal solution for our customers who adopt FUJITSU microcontrollers. Various other functions can be addressed through the addition of optional software programs and cables.

* Origin of the name MicroPecker: The small pecker (MicroPecker) that eats (pecks at) bugs (defects in system development).

Introduction

In recent years, the microcontrollers utilized in automobiles, home appliances, and industrial machinery have diversified considerably. Many different development tools, including debuggers, matching tools, and network tools, are required for the development of electronic control units (ECUs) for automobiles.

The group of products based on the MicroPecker platform introduced in this article solves the issues in the development of microcontroller application products as well as ECUs and realizes reduced customer burden in terms of development tool investment for the introduction of new tools.

MicroPecker

MicroPecker is a tool platform with a built-in single-wire interface that aims to address the specifications which is being promoted by the standards body JASPAR. It supports FUJITSU’s single-wire interface and provides an optimal solution for our customers who adopt FUJITSU microcontrollers. This product is used by connecting the main unit to the host PC via USB2.0. Various functions can be addressed through the addition of optional software programs and cables. We are currently developing or planning the following products using MicroPecker as the common tool platform.

- Eclipse-based software development environment
  In Europe, Artop (the AUTOSAR Tool Platform) has been proposed as a tool platform for utilizing development support tools that conform to AUTOSAR. Eclipse, which is adopted by Artop, is a software development environment developed
by IBM (as an integrated development environment) that is essentially the global standard platform in web development. In addition to being open source software, Eclipse also enables the simple development of plug-in software programs to expand its functions and the various plug-in modules that have already been released.

By adopting Eclipse in the MicroPecker software development environment, it will be possible to utilize the numerous plug-in software programs (modeling, architecture, configuration control, simulator, XML tools, and so forth) that we have thus far accumulated. Furthermore, we will develop a special plug-in module for the development of embedded applications in order to support various manufacturers’ microcontrollers. In this way, the practical standardization and internationalization of automotive embedded tool chains can be realized. It is also expected that liaison among various development processes and design support tools may be necessary in order to address software function safety (ISO-26262) in future automotive software development.

Since Eclipse is capable of addressing function expansion in a simple manner using plug-in modules and incorporating new functions while succeeding the existing development environment, we expect that it will enable developers to work on the development of next-generation products without discarding already accumulated assets.

**Flash microcontroller programmer**

Our Flash microcontroller programmer has been introduced at more than 100 domestic and overseas companies, including automobile, home appliance, and industrial device manufacturers as well as specialized microcontroller writing companies. We plan to provide an on-board writing programmer with a single-wire interface by utilizing our accumulated know-how.

**Automotive ECU matching tool**

While conventional ECU test processes mainly comprise algorithm tests, the current trend is to spend more evaluation time matching with actuators and so forth. It is believed that the following two factors underlie this trend:

1. **As high-performance microcontrollers are becoming increasingly available at low cost and as the tendency to compensate for actuator precision through software control is growing, the matching process to adjust to the target value dynamically is becoming more important in vehicle development.**

2. **Expenses for repairs and warranties to address problems with “unidentified causes of failure” have been increasing for automobile manufacturers. This has also led to increased matching time in the development stage. In this context, the demand for useful matching devices with high reliability is growing.**

Conventional tools have restrictions in both physical conditions (location of installation, size, distance between the measuring device and the host) and environmental conditions (temperature, humidity, dust) in the matching process. Problems also arose due to a failure to deliver sufficient evaluation results, including the overlooking of problems that should have been discovered.

The integrated development tool we are currently developing (MicroPecker) is equipped with a single-wire interface to solve these problems.

This single-wire interface enables high-speed communication of 30Mbps or higher while ensuring a long clearance (up to 10m) from the measuring device in addition to resistance to external noise through the adoption of coaxial cable. This single-wire interface enables an ideal matching process that is free of the aforementioned restrictions.

**Automotive network analyzer**

Our analyzer products, including FlexRay bus analyzers, CAN bus analyzers, and LIN bus analyzers, have been used by many domestic and overseas customers for a decade. We were one of the first companies to participate in standardization consortiums.

- FlexRay Consortium (2005)
- JASPAR (2006)
- AUTOSAR (2007)

The FlexRay communication middleware that was jointly developed with Nagoya University and VITs, Inc. received the IP Design Award in 2006.

By applying such know-how and our accumulated technology in MicroPecker, we will realize an ultrasmall multi-channel synchronous bus analyzer. Complete synchronous monitoring of an automotive network with multiple channels can also address the analysis and evaluation of buses with CAN-CAN gateways or CAN-LIN gateways.

**Product Specification and Product Lineup**

MicroPecker has been developed with a focus on portability and physical restrictions by assuming application in various scenes. It realizes miniaturization, weight reduction, and the use of a single-wire interface.

| Table 1 presents the product specifications, and Table 2 the product lineup. |
| The debugger products include software programs that address tool chain development for the integrated Eclipse development environment through joint development with Monami Software. |
They enable a series of development processes, including configuration control, programming, compiling, building, and debugging, to be carried out seamlessly and efficiently.

We are capable of planning, development, sales, and support for all of our products using our proprietary technology and we are thus able to support our customer needs flexibly and promptly by providing customization to unique customer specifications and extensive support. This group of products is currently under development. We plan to release them successively beginning in or after spring 2010.

### Connection Examples

Since connection to MicroPecker only requires connection of the ultrasmall and light-weight main unit between the host PC and the target, it allows developments in small spaces and simple debugging at travel destinations. In addition, the use of high-quality coaxial cable instead of inexpensive twisted pair cable as the connection cable provides measures against distance and space from the actual device and external noise. This product is also an effective on-board debugging interface for evaluation, various parameter adjustments, and log collection in a built-in status on the actual device.

---

### Table 1 Specifications

<table>
<thead>
<tr>
<th>Name</th>
<th>MicroPecker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported MCU</td>
<td>MCU with built-in single line interface</td>
</tr>
<tr>
<td>Debugging I/F</td>
<td>Single line interface</td>
</tr>
<tr>
<td>Target connection cable</td>
<td></td>
</tr>
<tr>
<td>100Ω coaxial cable</td>
<td></td>
</tr>
<tr>
<td>50Ω coaxial cable</td>
<td></td>
</tr>
<tr>
<td>Twisted pair cable</td>
<td></td>
</tr>
<tr>
<td>Operating environment for host</td>
<td>Supported models</td>
</tr>
<tr>
<td></td>
<td>IBM PC/AT-compatible machines</td>
</tr>
<tr>
<td>Supported OS</td>
<td>WindowsXP/Vista</td>
</tr>
<tr>
<td>Host I/F</td>
<td>USB2.0 High Speed (480Mbps)</td>
</tr>
<tr>
<td>Power supply</td>
<td>DC5V (supplied via USB port)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>30 (W) × 62 (H) × 12 (D) mm</td>
</tr>
<tr>
<td>Weight</td>
<td>35g</td>
</tr>
</tbody>
</table>

### Table 2 Product Lineup

<table>
<thead>
<tr>
<th>Product lineup</th>
<th>Product model</th>
<th>Supported FUJITSU microcontroller</th>
<th>Main functions</th>
</tr>
</thead>
</table>
| Debugger for FR81S (under development) | S810-R1D-FR81S | FR81S                             | • Integrated platform software (PizzaFactory manufactured by Monami Software, *Eclipse-based integrated development environment software)  
• FR81S development package (SOFUNE C/C++ Compiler, FR81S Plug-in) |
| Flash programmer for FR81S (in planning) | S810-R1F-FR81S | FR81S                             | • Erase, blank check, write/read, verify execution possible                    |
| ECU conformance tool for FR81S (in planning) | S810-R1C-FR81S | FR81S                             | • Real-time RAM editing function  
• Real-time RAM monitoring function  
• Logging function                      |
| CAN bus analyzer (in planning) | S810-R1-CAN   | —                                 | • CAN communication parameter setting function  
• CAN bus monitoring function  
• Transmission emulation function       |
Figure 1 presents the integrated development environment (debugger) screen.

NOTES
* The products introduced in this article are currently under development or in planning; their specifications are subject to change without prior notice.
* Windows is a registered trademark of the U.S. Microsoft Corporation in the U.S. and other countries.