Shortening CAN debugging

The E2 emulator by Renesas shortens the time required for CAN communication and power consumption debugging.

The Renesas MCU development environment (Photo: Renesas)

Renesas Electronics Corporation announced the E2 emulator, a new-generation on-chip debugging emulator. The E2 is intended as a development environment for the latest devices in the Renesas RH850, RX, and RL78 families of micro-controllers (MCUs), and for a selection of automotive system(s)-on-chip (SoCs). The product supports the extended debugging functionality of the RH850 family and contributes to shortening the time required for CAN communication debugging and the current consumption debugging.

Recently, use of electronics for automotive control is increasing to improve the fuel efficiency, safety, and comfort of vehicles, and the number and variety of built-in sensors is growing. This leads to an increased size of control programs of MCUs used in electronic control units (ECUs). Since each ECU performs communication control via automotive networks such as CAN, LIN, and Flexray, and dozens of ECUs are installed in a single vehicle, communication control has become more complicated. Additionally, considering that many ECUs are installed in a vehicle, it has become more important to precisely understand current consumption when each ECU operates and optimizes the control programs to achieve low current consumption in automotive systems.

One challenge with existing ECU software development is that it is necessary to download the program from the emulator to the MCU. And each time the debugging code needs to be added to the control program, it increases the program size and download time. In the case of CAN communication, a significant contributor to this repetitive debug cycle is the need to use a CAN analyzer along with the emulator—these devices have not traditionally operated synchronously. In the case of low power optimization, the emulator is typically used side-by-side with an ammeter or current probe, and it is difficult to link the current consumption peak specifically to the matching point of time in the running code.

To address this challenge, Renesas has integrated additional debugging features required for software development as solutions into the E2. Specifically, the emulator reduces the time required to determine the cause of an error during CAN communication by providing a connection to the network that allows message timing to be matched with executing code; it also makes it easier to determine the cause of current consumption peak by monitoring the current draw of the device and correlating that with the code. According to the company, this enables system developers to potentially reduce development time in these areas to as little as one-tenth compared with the existing method of repeated program modification and downloads. Even when the modification/download cycle is necessary, the increased download speed of the E2 lessens its impact on the speed of the development cycle.

The E2 emulator features faster communication speeds between the MCU and emulator and parallelization of flash rewriting and data communication. These capabilities enable control program downloads up to twice as fast as Renesas' existing E1 emulator. As a CAN communication debugging function, the emulator can stop the program when it detects that the interrupt response time has exceeded a specified limit, and record and display trace data relating to CAN communication reception and interrupt response processing simultaneously. According to Renesas, the debugging function is provided as an industry first solution for measuring CAN communication response time that allows for clearer correlation between CAN communication and program operation.

This solution for measuring CAN communication response time can allow trace analysis of CAN communication and program operation to be performed by the emulator alone, and can reduce the time required to determine the cause of an error. The product can detect the current consumption peak and stop the program when the emulator detects that the current consumption has exceeded set limits for a certain period of time. This allows system developers to determine the conditions in the program that cause it to exceed the expected current consumption levels. Additionally, the emulator can display the program operation and current consumption together, making it easier to identify areas where current reduction should be targeted. This shortens the tuning time for reducing current consumption.

The product offers a native hot plug connection on supported devices with no adapter required. It also supports the CS+ integrated development environment, which is designed to make the inevitable cycles of editing, building, and debugging in the software development.

CW