WHITEPAPER

Connecting to the target without a cable

Wireless Debugging as a part of an embedded software test concept
for corded and cordless electric tools

Table of Contents

1 Motivation

Embedded applications run in diverse environments. How to test the embedded software for mobile controls? A question many companies developing embedded systems are facing today. How to detect sporadic software bugs while the machine is in use? A wireless debugger, build in an embedded systems, is needed to set breakpoints and record data during runtime, which is then transmitted wirelessly to the PC where the testing is done. Development time may be shorter, because tests and analysis are performed under real field conditions. Additionally to flexible test conditions, such debugger provides galvanic isolation that is required in test systems, where big potential fluctuations are possible. This paper describes a wireless, Bluetooth-based solution for debugging, which addresses these concerns.

Tags: wireless; Bluetooth; debugging; trace; printf debugging; galvanic isolation; harsh test environment; mobile test system;

2 Introduction

Traditionally a debugging hardware is directly connected to the target system (e.g., via a JTAG debug interface) and then using an USB or Ethernet cable to the computer on which the debugging and analysis is done. iSYSTEM has been asked by an industrial customer to develop a solution for cases, where a debugger needs to communicate with the computer wirelessly. In this paper, case studies are presented, in which it is necessary to use a wireless debugger. iSYSTEM's Bluetooth-based wireless debugger (iONE-BT) was created as a response to a growing demand for such tools.

3 Case Studies

Several use-cases were studied during the development of the wireless debugger. This paper focuses on a case, where galvanic isolation is mandatory and another case, where the target mobility is required and therefore cannot be debugged with a traditional, wired debugger. Additionally, cases where target system disassembly is not possible and where the target system runs in harsh environments were considered.

3.1 Uninterruptable power supply (UPS)

Wireless debugging is a solution for cases, where galvanic isolation is required. Galvanic isolation is a principle of isolating parts of the electrical system to prevent current flow where it is not permitted. It is commonly used in cases, where two parts of the electrical system must communicate, but their grounds may be at different potentials.

Galvanic isolation is required when testing systems, where big fluctuations in the potentials are expected. This is common for example in UPS systems, which need to be stress-tested with disruptions such as lightning strikes or sudden blackouts. Figure 1 shows an example of an UPS system with iONE-BT attached.

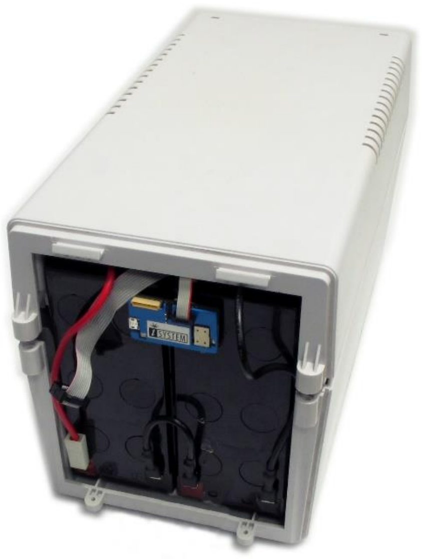

Figure 1 UPS with iONE-BT debugger attached

3.2 Robot vacuum, a wireless, free-moving device

In cases, where the target hardware moves around wirelessly or it is unacceptable to disassemble the machine each time changes need to be done to the debug and analysis system, wireless technology is needed to enable debugging of such systems.

Figure 2 shows another user case with built-in iONE-BT, where an application running on a robot vacuum needs to be tested and verified. Debugging hardware should be small enough to fit in the robot vacuum casing and light enough not to affect the power consumption of the device.

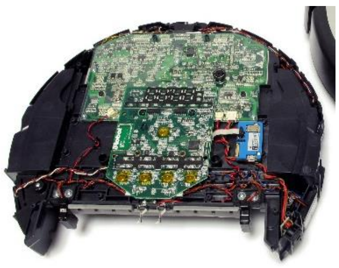
Figure 2 Mobile Embedded System with iONE-BT debugger attached

4 Wireless Debugger

Solution for these cases is a Bluetooth-based wireless debugger, iONE-BT, developed by iSYSTEM. Its compact size (figure 3) enables manufacturers to place the debugger directly in the hardware housing, allowing the hardware to move freely.

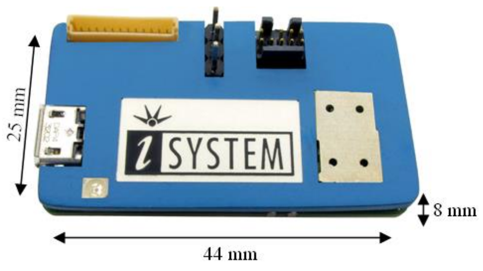

Figure 3 iONE-BT size characteristics

iONE-BT enables embedded developers to wirelessly debug and test the embedded application. To control the wireless debugger hardware iSYSTEM provides iSYSTEM winIDEA, a development and test environment, running on Microsoft Windows and/or Eclipse (with a winIDEA plugin). winIDEA gives detailed overview and control of the application, offers support for several real-time operating systems and enables the user to create a trace recording during the application execution. For cases, where advanced tests are required, a highly customizable testing tool (unit, integration and system test), testIDEA, is available. For advanced users there is a possibility of scripting the tests with the provided application programming interface (API) that is available for almost all types of external applications. For cases, where one wants to monitor certain variables during application execution, real-time data acquisition is supported.

4.1 Wireless debugging

As shown in figure 4, the target processor connects to iONE-BT via a debug port supporting multiple debug protocols (i.e., JTAG). iONE-BT then communicates via Bluetooth with iBridge, which is a USB dongle connected to the PC, where the debug software runs. Communication reliably works at a distance up to 10m.

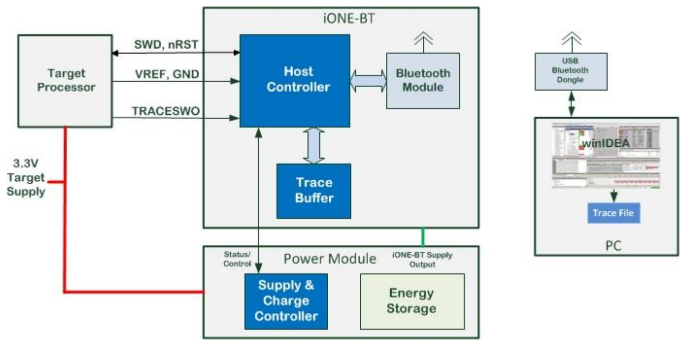

Figure 4 Schematic view of a wireless debug session setup

This allows debugging the application from a safe distance, as well as it provides galvanic separation from the hardware in which the application is running. This is critically important in cases, where high voltages are used or where the grounds may be at different potentials. Traditionally, a wired debugger might cause an unwanted current flow, which could result in unwanted hardware / application behavior or even damage to the hardware.

During iONE-BT development the goal was to minimize the wireless communication to a minimum, to perform most of the operations on the debugger side. However, Bluetooth still dictates the speed with which operations such as application download are performed. iONE-BT supports transfer rates of average 600kb/s

4.2 Wireless data trace and printf style debugging

iONE-BT currently supports ARM™ Cortex-M based processors, which offer Single Wire Output (SWO). Trace messages are outputted through the SWO pin. Instrumentation Trace Macrocell (ITM) and Data Watchpoint and Trace unit (DWT) can be used to make a data trace recording and to make a basic execution trace recording with the instrumentation of the application source code. This offer a possibility of program execution reconstruction (e.g. task profiling) based on the trace record of the instrumented source code. In cases where this is not possible, printf style debugging is possible by wiring the CPU's UART module to the SWO pin.

4.3 Real-time data acquisition

For cases, where a variable needs to be monitored in real-time, iONE-BT offers data acquisition functionality, which records the values of the selected variable(s) in real-time. Values are stored in the iONE-BT's internal buffer and are uploaded to the PC, making continuous monitoring possible. In case visualizing of the monitored data is desired, daqIDEA (an application integrated in iSYSTEM's winIDEA) offers a simple user interface for continuous display of monitored variables in forms of self-incrementing graphs and charts.

4.4 Performance and power consumption

Depending on the requirements and possibilities of the target hardware, iONE-BT offers two power modes. For target systems, which can provide sufficient current, performance mode is available. In this mode, iONE-BT operates on the highest possible frequency and enables fastest trace functionality, real-time data acquisition and watchdog service in addition to the debugging functionality. In this power mode power consumption averages on 100mA.

In case such power consumption is unacceptable, iONE-BT offers a low power mode, where it runs on decreased frequency and enters low power mode whenever possible. This mode supports the debugging functionality, as well as trace, although trace is slower due to the decreased frequency. Low power mode doesn't support real-time data acquisition and watchdog service. Power consumption in this mode averages on 55mA. After certain time of inactivity iONE-BT enters deep sleep mode, in which its power consumption drops below 1mA, in order to preserve energy sources when not actively debugging.

4.5 Power supply considerations

Depending on the possibilities on the embedded system (unit under test), iONE-BT may be powered either by the target hardware or with an external power supply. iONE-BT operates on 3.3V power supply. If the target hardware cannot provide required voltage, there are three possibilities for external power supply – a module with two AA batteries, a CR123A battery or a super capacitor module (Supercap). The three modules differ in size and autonomy. Battery modules have longer autonomy, ranging from 24 to 40 hours, where super capacitor module only offers 4 min autonomy. However, super capacitor can be quickly recharged directly from the target system and therefore permanently sealed in the device, whereas battery modules can't be recharged and the batteries need to be replaced when depleted. Comparison between the autonomies of the external power supply modules can be seen on figure 5.

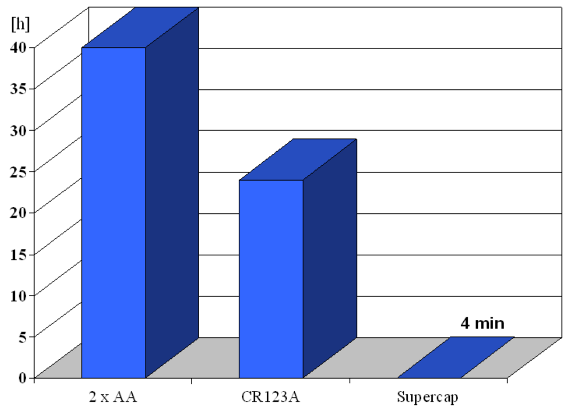

Figure 5 Autonomy in hours, when in low power mode

In case the debugger and the power supply need to be enclosed in the test system, their size plays a vital role when deciding on the power supply module. Figure 6 shows the differences in size between the modules.

Module	***length [mm]***	***width [mm]***	***height [mm]***
2xAA	72	35	20
CR123A	44	25	19
Supercap	44	23	14

Figure 6 Size specification for external power supply modules

5 HILTI Use Case

How to test the embedded software for mobile controls? A question the company Hilti is facing developing hand-held electric machines. Within these machines of the famous red suitcase microcontrollers work in the harshest environmental conditions. Such controllers control modern brushless motors and collect data from various sensors in real time. Now the question is how to detect sporadic software bugs while those machines are in use? Quite annoying are USB cable connections, apart from the technical safety for corded electric tools. Wireless debugging allows recording of data from the machine by setting breakpoints or tracing the parts of the program execution during runtime and via radio. Thus, the development time is shortened because such tests and analysis are executed under real customer conditions. Very new test use cases may be driven on the real hardware and will influence the reliability and robustness of such products dramatically.

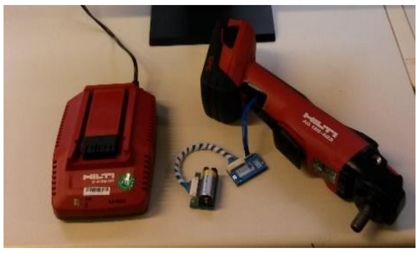

Figure 7 iONE-BT being used within Hilti's embedded software test concept for corded and cordless electric tools

6 Summary

The wireless, Bluetooth-based debugger iONE-BT was developed as a response to an industrial customer's request for a wireless debugger, which could be enclosed in the target system and operated from distance. Galvanic isolation, which is the second most requested feature for debuggers, is realized with the option of external power supplies for the debugger and Bluetooth connection to the PC. Additionally to basic debugging functionality, iSYSTEM's integrated development and test environment winIDEA adds testing functionality (unit, integration and system test with the integrated test tool testIDEA) as well as a tool for graphical display of large data sets, daqIDEA. This empowers the user to create diverse tests and presentations for the embedded application, among which are unit, integration and system tests, test scripts, real-time data acquisition and graphic display.

Currently it supports ARM™ Cortex-M architecture and is therefore available to use in diverse target systems. A WiFi version of this debugger is planned for 2014.

7 Appendix

Revision History

Date	Version	Author	Description
Date	14.01	Anja Visnikar, Primoz Alic, Matej Antonijevic, Erol Simsek	Final Paper