AN3044
Application note

ECU level diagnostic with SPC563Mx and SPC564Ax

Introduction

The continuous need for quality improvements in automotive is creating focus on ECU (Engine Control Unit) level diagnostic tools as the analysis of traceability information and first level diagnostic should be done as early as possible to improve analysis cycle time.

This application note is intended to clarify the different tools possible with SPC56x microcontrollers, the different possibilities offered when the flash is protected and how to read the traceability information that is stored inside SPC56x flash. This will then allow OEMs (Original Equipment Manufacturer) to define and implement the right strategy for their ECUs.

As IP protection and engine tuning is of high importance, the application note is focused on identifying configurations (software, tools) that do not require to have access to the flash password.

This application note is only valid for powertrain devices: SPC563Mx and SPC564Ax.

Contents

List of tables

List of figures

1 Introduction to ECU level diagnostic

ECU level diagnostic in the field has 2 important goals:

- Collect traceability information of the ECU configuration
- Run ECU level self tests

The additional difficulty for ECU level diagnostic is access to the PCB as in most of cases, the PCB is hidden inside the ECU housing; the application note will propose different possibilities, for ECU level (i.e.: with housing) and PCB level (i.e.: without housing).

1.1 ECU level traceability information

ECU level traceability involves several data:

- Hardware configuration information: hardware revision number, traceability information of the key components of the ECU
- Software configuration information: software revision number of each top level software component

In addition to this, date code of hardware or software components is also sometimes collected.

1.2 ECU level self tests

When an ECU is suspected of wrong behavior, it is required to be able to run ECU level diagnostic to confirm or not the suspicion.

2 ECU level diagnostic

ECU level diagnostic does not usually allow access to the JTAG pins. The ECU level diagnostic is describing diagnostic possibilities with only the pins involved for serial boot (i.e.: without using JTAG pins).

2.1 SPC563Mx and SPC564Ax traceability information

Each SPC56x has traceability information stored inside the Flash (shadow Flash).

The traceability information is stored in memory locations that are accessible by the application software so that user application software can be used to read them and send them over the selected diagnostic communication interface, UART,LIN, CAN or FlexRay.

To send the traceability information, users can use a user defined diagnostic mode that is entered by the application software or use a serial boot mode. This is mainly driven by the password strategy of the OEM.

Case1: using a user defined diagnostic mode

When the Flash is protected, the application software can always read the traceability information. The application software may have then specific diagnostic mode that allows to send this information over a predefined communication interface (CAN, LIN, K-line, FlexRay). This operation does not require people to know the Flash password to do this in the field.

Case2: using a serial boot mode and by allowing users to know the password

The traceability information stored inside SPC56x can be read by an external tool using the password defined by users.

Case3: using a serial boot mode and avoiding users to know the password

To allow to read traceability information using the serial links only, the following procedure must be implemented:

- Reserve locations in the RAM (preferably with highest addresses)
- Use the application software to make systematically a copy of the locations in the Flash to selected locations in the RAM during the power on procedure
- Start the ECU in application mode first
- Apply then a reset without supply shutdown and ensure the MCU is brought in serial boot mode
- Use the software uploaded in RAM to export the traceability information using the selected serial link

In this mode (i.e.: serial boot control word is not 0x55AA and censor control word is not 0x55AA), the Flash is not enabled in boot mode and external JTAG pins do not allow direct access to the device when the Flash is enabled.

2.2 ECU self tests

ECU level self tests can be run either from the application software (ex: selection from ECU pins) or from the microcontroller RAM.

Even when high protection level is enabled, SPC563Mx and SPC564Ax allow to upload software in RAM and to run it without having to distribute any private password to users. This allows then to execute MCU specific self tests, ASSP self tests, power stage self tests.

2.3 Tools for ECU level diagnostic

Different tools can be used for ECU level diagnostic.

Third party tools allowing connection to the selected communication interface can be used (ex: Vector, Etas,..). Some of those tools are also available for external use and for extended temperature range.

Tools from STMicroelectronics

STMicroelectronics has developed tools for SPC56x for use in lab environment (indoor, 25 typ. ambient temperature). The tools to be used for ECU level diagnostics are:

- Target connection = ST Combox (CAN or UART)
- PC software = CPUcommander

Please, note that the use of CPUcommander requires development of few software to select the right communication parameters and to define the format of traceability information.

This tool also allows to upload a user defined software and run it in the ECU.

2.4 Summary for ECU level diagnostic

The table below is summarizing the different possibilities for ECU level diagnostic.

Table 1. ECU level diagnostic

		Comments
SPC56x traceability info from protected Flash	3 different options proposed with 2 not requiring to distribute the Flash password: either use specific diagnostic software either use serial boot mode with a specific sequence.	
Connection to ECU	ECU connector	To have 1 pin for boot mode selection and 1 pin for ECU reset without power off
Communication interface	UART, LIN, CAN, FlexRay	
ECU software for self tests #1	User software using application specific diagnostic mode.	Diagnostic mode can be selected by using GPIO. Diagnostic can send traceability information and run small ECU self tests software.
ECU software for self tests #2	Software uploaded in SPC56x RAM	Requires to have access to boot mode from ECU connector. Very flexible scheme for loading and executing self tests but require specific sequence for reading traceability information.
Proposed tools #1	Third party tools (Vector, Etas, ..) that allows connection to the selected communication interface.	
Proposed tools #2	ST Combox + CPUcommander software	Requires manual configuration of CPUcommander (com. characteristics, format of traceability info).

3 PCB level diagnostic

PCB level diagnostic allows access to any signal from the MCU like JTAG.

3.1 SPC563Mx and SPC564Ax traceability information

Each SPC56x has traceability information stored inside the Flash (shadow Flash).

The traceability information is stored in memory locations that are accessible by the application software so that user application software can be used to read them and send them over the selected diagnostic communication interface, UART,LIN, CAN or FlexRay.

To send the traceability information, users can use a user defined diagnostic mode that is entered by the application software or use a serial boot mode. This is mainly driven by the password strategy of the OEM.

Case1: using a user defined diagnostic mode

When the Flash protection' s scheme is used, the application software can always read the traceability information. The application software may have then specific diagnostic mode that allows to send this information over a predefined communication interface (CAN, LIN, K-line, FlexRay). This operation does not require people to know the Flash password to do this in the field.

Case2: by allowing users to know the password

When the Flash is protected and when it is possible to allow users to know the Flash password, tools can be used to connect to JTAG and write the password in JTAG mapped register. Once the tool is connected, the Flash is visible and the traceability information can be read.

Case3: using a serial boot mode and avoiding users to know the password

When SPC563Mx or SPC564Ax is put in serial bootstrap mode with serial boot control word different from 0x55AA, the Flash of the device is disabled and the JTAG port is enabled. To read the traceability information, the following procedure shall be implemented in user software:

- Reserve locations in the RAM (preferably with highest addresses)
- Use the application software to make systematically a copy of the locations in the Flash to selected locations in the RAM during the power on procedure.
- Start the ECU in application mode first
- Apply then a reset without supply shutdown and ensure the MCU is brought in serial boot mode
- Then use the tool with JTAG to connect to the device, load and execute software out of RAM

In this mode (i.e.: serial boot control word is not 0x55AA and censor control word is not 0x55AA), the Flash is not enabled in boot mode and external JTAG pins do not allow direct access to the device when the Flash is enabled.

3.2 PCB level self tests

PCB level self tests can be run either from the application software (ex: selection from ECU pins) or from the microcontroller RAM.

Even when high protection level is enabled, SPC563Mx and SPC564Ax allow to upload software in RAM and to run it without having to distribute any private password to users. This allows then to execute MCU specific self tests, ASSP self tests, power stage self tests.

The use of JTAG pins here allows to have high efficiency tools (debugger based tools) for this analysis.

3.3 Tools for PCB level diagnostic

Different tools can be used for PCB level diagnostic.

Third party tools allowing connection to the selected communication interface can be used. Depending on the selected communication interface, different third party tools can be used (CAN/LIN = Vector, ETAS; JTAG = PLS, Lauterbach, iSystem). Some of those tools are also available for external use and for extended temperature range.

Tools from STMicroelectronics

STMicroelectronics has developed tools for SPC56x for use for PCB level analysis and using JTAG.

- Target connection = ST Rlink
- PC software = Raisonance debugger

This tool is only for use for ambient temperature. As the tools using JTAG and based on debugger allow higher efficiency in PCB level diagnostic (ASSP, reading of analog inputs, MCU) ST is only developing MCU self tests based on JTAG tools.

3.4 Summary for PCB level diagnostic

The table below is summarizing the different possibilities for PCB level diagnostic.

Table 2. PCB level diagnostic

		Comments
SPC56x traceability info from protected Flash	3 different options proposed with 2 not requiring to distribute the Flash password: either use specific diagnostic software either use serial boot mode with a specific sequence.	
Connection to ECU's PCB	JTAG (connector only populated for PCB level self tests).	JTAG on PCB test points or not populated JTAG connector
Communication interfaces when using third party tools	JTAG, CAN, UART, FlexRay	
Communication interface when using ST tools	JTAG	
ECU software for self tests #1	User software using application specific diagnostic mode.	Diagnostic mode can be selected by using GPIO. Diagnostic can send traceability information and run small ECU self tests software.
ECU software for self tests #2	Software uploaded in SPC56x RAM	Requires to have access to JTAG. Very flexible scheme for loading and executing self tests with specific sequence for reading traceability information.
Proposed tools #1	Third party tools (Vector, Etas, lauterbach, PLS, iSystem..) that allows connection to the selected communication interface.	
Proposed tools #2	ST JTAG tool + Raisonance debugger	

4 SPC563Mx / SPC564Ax and Flash protection

SPC56x allows to restrict access to the Flash content. There was a specific focus for the powertrain variant as due to the type of application, car makers want to prevent engine tuning.

The following table is summarizing the different possibilities offered by the powertrain variants of SPC56x when the Flash is protected.

Table 3. SPC56x Flash access from serial boot mode or from JTAG

	Censor control word is not 0x55AA	Censor control word is 0x55AA
Serial boot control word is not 0x55AA	In boot mode[1], Flash is disabled but the rest of the device (CPU, RAM, peripherals are enabled) and JTAG accesses are enabled. In run mode, JTAG accesses are disabled unless private password is entered	In boot mode, Flash is disabled and JTAG accesses are enabled. In run mode, JTAG accesses are enabled.
Serial boot control word is 0x55AA	In boot mode, Flash is enabled and JTAG access is enabled; private password required for serial boot or to enable JTAG accesses. In run mode, JTAG accesses are disabled unless private password is entered	In boot mode, Flash is enabled and JTAG access is enabled; private password required for serial boot or to enable JTAG accesses. In run mode, JTAG accesses are enabled.

1. This mode allows ECU and PCB level diagnostic without requiring the knowledge Flash password.

The table shows that when censor control word is not 0x55AA and when the serial control word is also not 0x55AA, SPC563Mx and SPC564Ax have a very good protection and it is not necessary to send the Flash password for ECU and PCB level diagnostics.

Debugging and analysis of specific cases that require access to the content of the Flash is still possible to restricted users.

4.1 Flash protection, shadow Flash and reprogramming

The password is stored in the shadow Flash.

This part of the Flash should have, as far as possible, static information to prevent the need to reprogram the shadow Flash in the field. Shadow Flash reprogramming is possible but should be done with care. Mistakes made during the reprogramming sequence can render the ECU unusable and unaccessible for new reprogramming.

5 SPC563Mx / SPC564Ax and traceability information

SPC563Mx and SPC564Ax have traceability information stored in the shadow Flash. This information is a unique MCU identification made of wafer lot number and MCU ID for this lot.

The coding and the memory map of the traceability information is reported in the datasheet of those products. As the goal here is to focus on showing the technical possibilities to read the traceability information, the memory map is not showed again here (please refer to the relevant product documentation).

5.1 Traceability code and Flash protection

As described in the product documentation, each device has a unique die identification made by the concatenation of wafer information and a unique ID per wafer.

This can be used to generate, during ECU programming in the production line, a unique password based on the encryption of the unique ID (readable by the application software in the shadow Flash) and of the password selected by the OEM.

Then, for ECU level diagnostics, the OEM can send the password for the ECU wherever analysis is needed.

6 STMicrolectronics self test tool

ST has developed tools for SPC56x self tests. This chapter describes the tool platform for ECU and PCB level diagnostics.

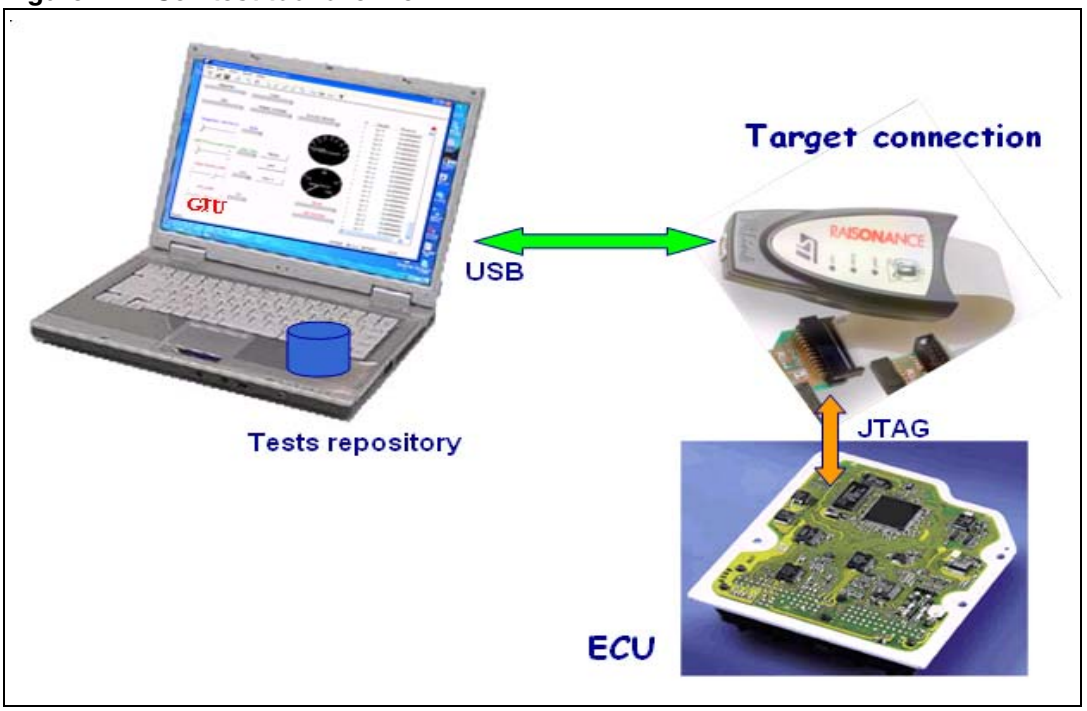
6.1 Self test tool overview

The figure below is showing the overview of the self test tool designed for SPC56x.

It includes:

- An hardware tool for target connection (USB to JTAG)
- A software tool for user interface
- A test repository

Figure 1. Self test tool overview



6.2 Self test tool GUI

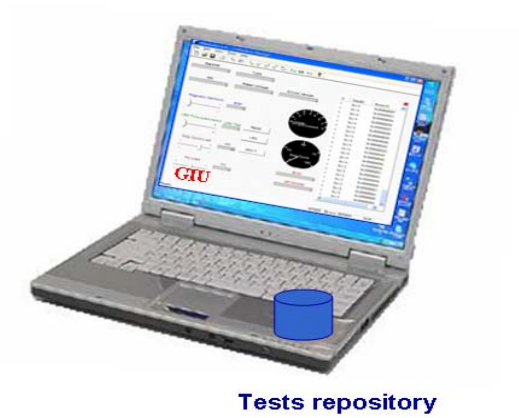
The GUI allows to connect to the MCU of the target ECU

- To read the MCU traceability information
- To run the self test or a selection of tests of the self tests and run then 1 time or a predefined number of times
  - Self tests should not activate outputs to avoid possible issue on the MCU
- To show the results of the self tests

6.3 Test repository

ST is developing a test repository for SPC56x. The tests repository is the set of self tests for a given SPC56x.

Figure 2. Test repository



Self tests are functional tests downloaded into the RAM of the target MCU.

They are then executed using the debugger. The MCU must be put into bootstrap loader mode to allow the JTAG tool to connect, load the self test, execute it, collect the results and show them on the PC.

They include memory tests, CPU self tests and peripheral self tests.

6.4 Target connection

ST has developed a target connection tool with Raisonance. This tool is powered from the USB port of the PC and connects to the JTAG port of SPC56x.

Figure 3. Target connection

6.5 ECU connection

The PCB of the ECU should be prepared for JTAG connection. In most cases, JTAG test pads can be placed close to the MCU for PCB testing. A good alternative is to have a provision for an SMD JTAG connector as it can be used for ECU testing and for PCB level diagnostic.

Figure 4. ECU connection

7 Other tools from ST

ST has developed a Combox to allow to interface from the USB port of a PC and a UART or CAN interface in an ECU.

7.1 Combox

The Combox is powered by the USB port. It has been designed for use in lab environment. It allows CAN communication.

Figure 5. Combox

The usage of the Combox shall be limited to reading traceability information.

MCU selftests are using the JTAG based tool.

7.2 SPC56x and autobaud

Some SPC56x support autobaud on CAN and UART; some do not.

The autobaud is supported by each powertrain variant (SPC563Mx, SPC564Ax) and by each chassis variant (SPC560Px, SPC564ELx, SPC567HKx). The autobaud allows to avoid to know the crystal of an ECU before connection.

For products without autobaud, users must know before connection the value of the crystal used in the ECU and then compute the needed baudrate on the Combox side.

8 Revision history

Table 4. Document revision history

Date	Revision	Changes
03-Sep-2009	1	Initial release.
17-Sep-2013	2	Updated Disclaimer.