



XAPP365 (v1.0) October 10, 2001

Automotive Scan Tool

Summary

This document describes the implementation of the Automotive Scan Tool design submitted to the recently publicized "Cool Module Design Contest". All development for this contest was performed using the Insight Springboard™ development platform which allows for rapid development of Handspring modules. This development platform incorporates the reprogrammable Xilinx CoolRunner™ XPLA3 CPLD and uses the Handspring Visor™ PDA expansion slot. Low power CoolRunner CPLDs are the ideal programmable logic solution for portable, handheld applications.

The Automotive Scan Tool design is a portable test device which interfaces with the On-Board Diagnostics (OBD) system found in new cars to display data and trouble codes generated by the OBD. The CPLD and Handspring design files for the Automotive Scan Tool are available and can be found at **Download Pack, page 8**.

A visual presentation of the Automotive Scan Tool is provided to show operation and functionality. This on-demand video is available at: <http://www.xilinx.com/apps/video.htm>.

Introduction

In the winter of 2001, Xilinx, Handspring and Portable Design Magazine collaborated on a design contest to highlight the use of Xilinx CoolRunner XPLA3 CPLDs in quickly developing Springboard modules for the Handspring Visor PDA. About 250 contest registrations were received and nearly 100 contest design ideas were submitted. From these submissions, ten were chosen to receive a Handspring Visor and an Insight Springboard development kit. These ten finalists were given three months to complete their designs in order to compete for a "winner takes all" grand prize of \$10,000. All finalists were required to include a written description of the project, all design files, and the necessary software to make their Springboard module prototypes operate. All the finalists did a great job. This application note is derived from the submission supplied by George Kaldis.

Appendix A, page 8 outlines existing Xilinx application notes that are appropriate for understanding this application note. These are available on the Xilinx website at: <http://www.xilinx.com/>.

Appendix B: References, page 9 describes additional appropriate references that may be of interest.

On-Board Diagnostics Overview

The following questions and answers are reprinted from the Environmental Protection Agency's website on On-Board Diagnostics. The discussion describes what OBD is and helps explain why it is valuable on today's cars.

What is OBD And How Does It Work?

By the early 1980s, numerous vehicles were using electronics and on-board computers to control many of the engine's control systems, such as fuel and ignition. Vehicle manufacturers had to develop ways to diagnose problems generated by the new electronic hardware found under the hood. Thus, the first OBD systems were developed by auto manufacturers in the early 1980s as electronic systems replaced mechanical systems.

The engines in today's vehicles are largely electronically controlled. Sensors and actuators sense the operation of specific components (e.g., the oxygen sensor) and actuate others (e.g.,

the fuel injectors) to maintain optimal engine control. An on-board computer, known sometimes as a "powertrain control module" or an "engine control unit," controls all of these systems. With proper software, the on-board computer is capable of monitoring all of the sensors and actuators to determine whether they are working as intended. It can detect a malfunction or deterioration of the various sensors and actuators, usually well before the driver becomes aware of the problem through a loss in vehicle performance or driveability. The sensors and actuators, along with the diagnostic software in the on-board computer, make up what is called "the OBD system."

How Does OBD Inform Drivers Of Problems?

When the OBD system determines that a problem exists, a corresponding "Diagnostic Trouble Code" is stored in the computer's memory. The computer also illuminates a dashboard light indicating "Service Engine Soon" or "Check Engine" or displays an engine symbol. This light, usually yellow in color, serves to inform the driver that a problem has been detected and vehicle service is needed. When the car is delivered to the repair shop, a service technician can quickly retrieve the stored diagnostic trouble codes from the computer memory of the vehicle using newly developed diagnostic tools. Since the diagnostic trouble codes will specifically identify the problem, the service technician can more quickly and accurately make the proper repair.

How Does OBD Help Consumers?

OBD systems are designed to alert drivers when something in the emission control system begins to deteriorate or fails. Early diagnosis followed by timely repair can often prevent more costly repairs on both emission control systems and other vehicle systems that may affect vehicle performance such as fuel economy. For example, a poorly performing spark plug can cause the engine to misfire, a condition sometimes unnoticed by the driver. This engine misfire can, in turn, quickly degrade the performance of the catalytic converter. With OBD detection of the engine misfire, the driver would be faced with a relatively inexpensive spark plug repair. However, without OBD detection, the driver could be faced with an expensive catalytic converter repair as well as the spark plug repair. In addition, manufacturers have increased incentive to build a higher quality vehicle with better performance, reduced emissions, and more efficient powertrains to prevent problems that can lead to OBD detection. OBD systems will also provide far more information than ever before to help auto technicians diagnose and properly repair vehicles during their first visit to the repair shop, saving time and money for consumers.

Automotive Scan Tool Design Summary

This design is an automotive scan tool. It uses the Insight Electronics, CoolRunner-based development board to create an interface to the On-Board Diagnostic (OBD) port that is in all new cars. The OBD-II communications standard defines three standard interfaces: J1850 VPW, J1850 PWM and ISO 9141-2. This specific design targets ISO 9141-2. With this OBD-II interface it is possible to request diagnostic data, sensor data and Diagnostic Trouble Codes (DTCs) from the car's Electronic Control Units (ECUs). It is also possible to clear the trouble codes and turn off the Malfunction Indicator Lamp (MIL), also known as the "Service Engine Soon" lamp. The design was tested, and data retrieval from a Subaru was successful.

Figure 1 is a photograph of the Visor, Development board, additional circuitry and cable. The open end of the cable attaches to the OBD-II connector on the vehicle.



Figure 1: Visor-Based Automotive Scan Tool

Implemented Features

The Auto Scan Tool reads and interprets diagnostic and sensor data. It displays a list of information describing the condition of the vehicle. The tool can also display or graph a single sensor value in “real time”. It can display a list of the active Diagnostic Trouble Codes (DTCs) and clear them.

Note: DTC testing was not completed as the target vehicle does not support DTCs.

The SAE (Society of Automotive Engineers) 1979 standard, which defines the packet protocol, allows for multiple responses per request and multiple ECUs responding to the same request. Currently the design accepts the first packet, which allows for three DTCs to be reported.

Operation Instructions

The main window has five “radio buttons” that implement five functions. Select the desired function and a new window appears. The buttons are:

1. "All" — This displays all of the diagnostic and sensor data.
2. "Text" — This displays a single sensor value in “real time.”
3. "Graph" — This graphs a single sensor value in “real time.”
4. "Get" — This displays a list of the active DTCs.
5. "Clear" — This clears the active DTCs.

See [Figure 2](#) for the main window button interface.

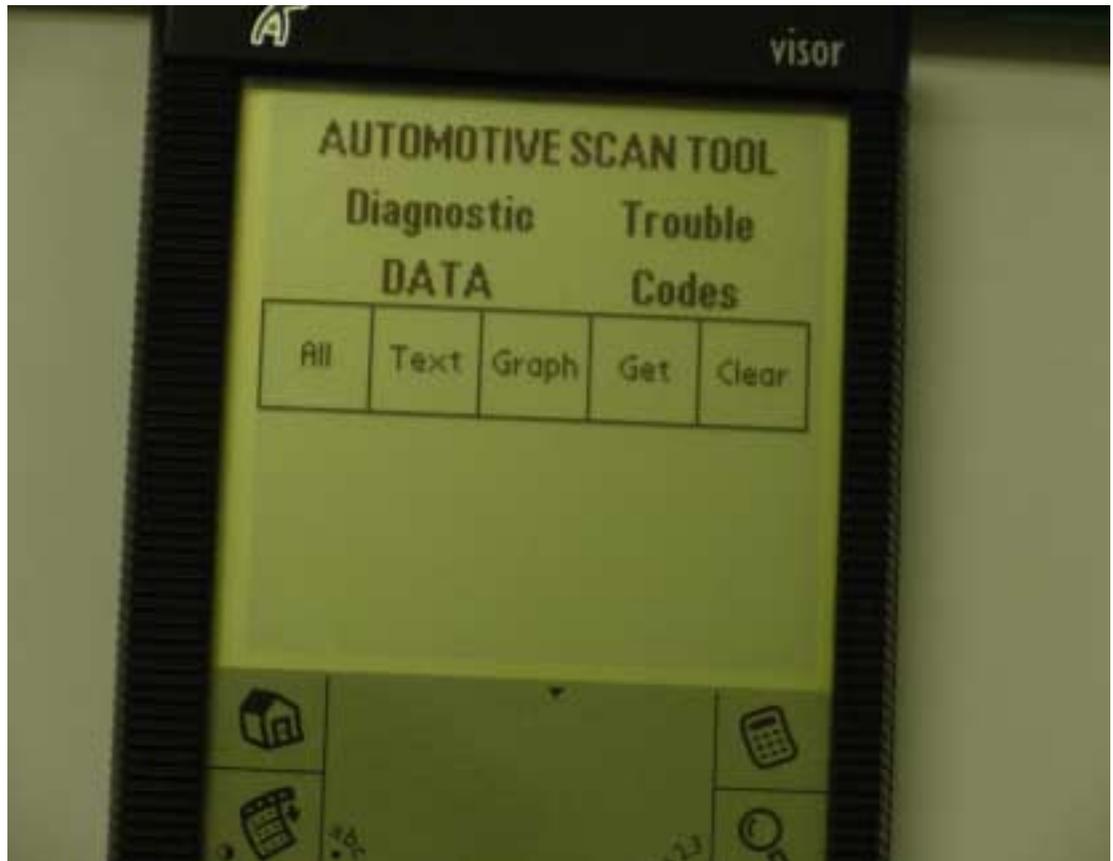


Figure 2: Main Window Display

After selecting one of the buttons and before bringing up the new window, the scan tool tries to establish communication with the ECU. It will display "Initializing". If it succeeds, it displays the new window. If it fails, it keeps retrying. Currently there is no "abort" button, so users must quit the application to exit the loop.

The All, Get and Clear windows bring up listboxes with appropriate data. To return to the main window select the first line in the listbox that contains the text "Main Screen".

The Text and Graph windows deliver dropdown menus from which to select the sensor data to display or graph. Again, to return to the main window, select the first line in the dropdown menu containing the text "Main Screen".

If at any time while receiving data communication with the ECU it is "lost," the software attempts reinitializing.

Hardware Description

Additional electronic circuitry is required to interface through the XPLA3 to the Visor bus. Specifically, an ISO 9141-2 transceiver, voltage regulator, pullup resistor, bypass capacitor, cable and connector are needed as shown in Figure 3. Figure 4 shows the J1962 connector pins and Table 1 describes their functions.

- Si9241EY — a monolithic bus transceiver providing bi-directional serial communication in ISO 9141-2 applications.
- 5V DC Voltage Regulator 78L05. The 78L05 provides a stable +5 Volt output.
- 470 Ohm pullup resistor
- 10 μ F bypass capacitor
- DB-9 male connector

- J1962 cable (Part number unknown).

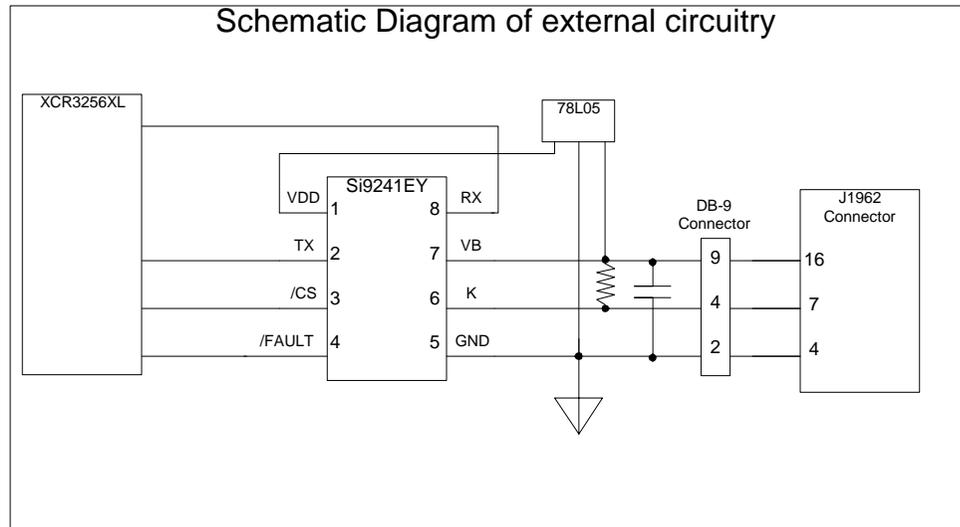


Figure 3: Additional External Circuitry Schematic



Figure 4: SAE J1962 Connector Pins

Table 1: SAE J1962 Pin Definitions

Pin #	Description
1	Manufacturer's discretion
2	SAEJ1850 Lime (Bus+)
3	Manufacturer's discretion
4	Chassis ground
5	Signal ground
6	SAE J2284 (CAN High)
7	K Line of ISO 9141-2 & ISO/DIS 4230-4
8	Manufacturer's discretion
9	Manufacturer's discretion
10	SAE J1850 Line (Bus-)
11	Manufacturer's discretion
12	Manufacturer's discretion

Table 1: SAE J1962 Pin Definitions (Continued)

13	Manufacturer's discretion
14	SAE J2284 (CAN Low)
15	L Line of ISO 9141-2 & ISO/DIS 4230-4
16	Unswitched Vehicle Battery Positive

CPLD Design Description

The CPLD design consists of 4 VHDL files: iso9141.vhd, iso_clk_divider.vhd, txmit.vhd, and rcvr.vhd.

Specific code descriptions are as follows:

- txmit.vhd and rcvr.vhd provide simple uart functionality. They are based on the Xilinx application note [XAPP 341: UARTS in Xilinx CPLDs](#) but have been modified to meet the specific requirements of ISO 9141-2.
- iso_clk_divider.vhd provides the uart clock by dividing down the 2.5 MHz clock from the A/D converter chip. It generates a 166.4 kHz clock for data transmission at 10.4 kbps and an 80 Hz clock for initialization transmission at 5 bps. It also generates a 5 ms, 9-bit bus idle timer that is used to time the bus transactions.
- iso9141.vhd is the top level of the design. It instantiates the other three components, generates the decoding for the chip registers, and buffers the FLASH/SRAM pins to the springboard pins.

Figure 5 shows a block diagram of the relationships of the VHDL blocks contained in the XPLA3 CPLD.

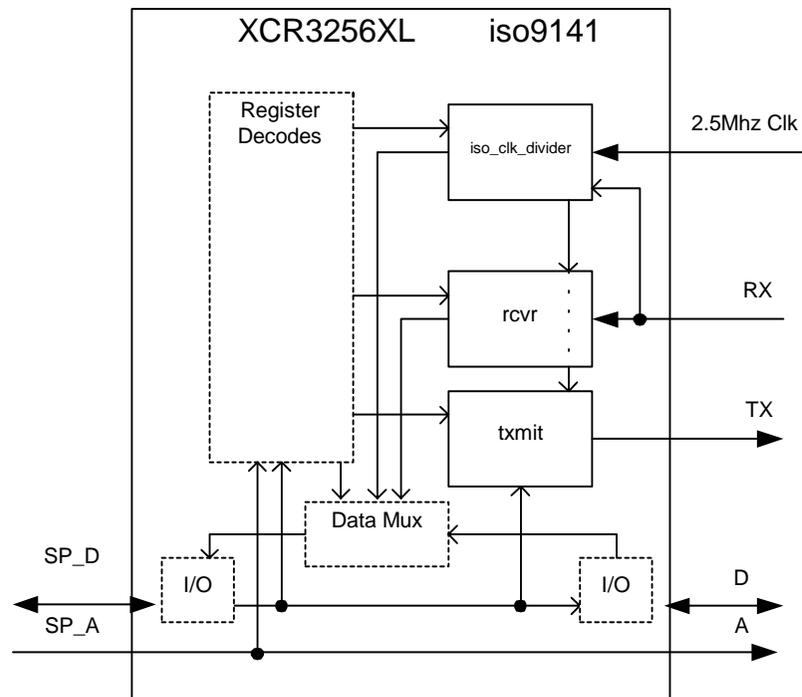


Figure 5: XPLA3 Block Diagram

Handspring Software Description

The application was created using PocketC and Ccontrols. The following files make up the application: `autoscan.pc`, `autoscan_func.pc`, `autoscan_pids.pc`, `autoscan_controls.pc`, `autoscan_data.pc`, `autoscan_sensor.pc`, `autoscan_graph.pc`, `autoscan_dtc.pc` and `autoscan.h`. (See [Download Pack](#), page 8).

Specific software functionality is as follows:

Autoscan.pc creates the main window with radio buttons to select the function to be run.

Autoscan_func.pc provides the basic functions for reading/writing data packets and initializes the interface.

Autoscan_pids.pc interprets the diagnostic and sensor data.

Autoscan_controls.pc implements the radio buttons, listbox and dropdown controls with Ccontrols.

Autoscan_data.pc creates the listbox with all of the diagnostic and sensor data.

Autoscan_sensor.pc creates a text display to show the real time value of one sensor.

Autoscan_graph.pc graphs the real time value of one sensor.

Autoscan_dtc.pc interprets the trouble codes, creates a listbox with all of the trouble codes and creates a listbox to clear the trouble codes.

[Figure 6](#) shows the relationship of the various software modules with the overall flow.

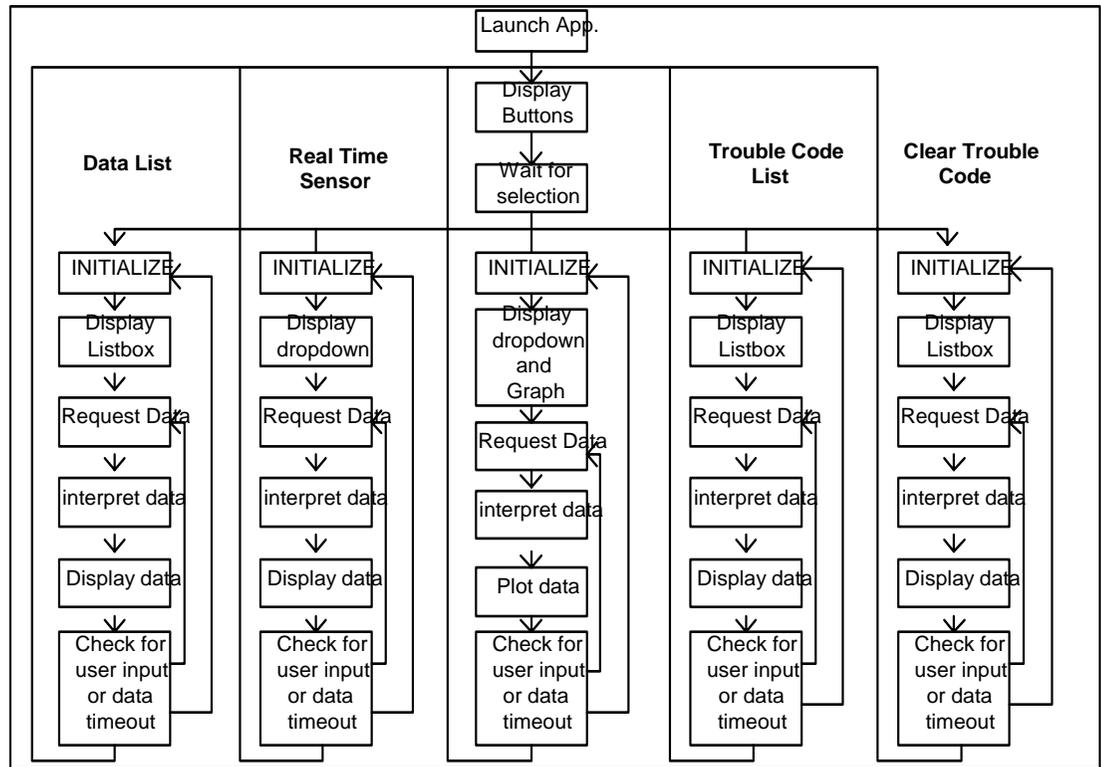


Figure 6: Software Flow Diagram

Download Pack

THIRD PARTIES MAY HAVE PATENTS ON THE AUTOMOTIVE SCAN TOOL. BY PROVIDING THIS HDL CODE AS ONE POSSIBLE IMPLEMENTATION OF THIS DESIGN, XILINX IS MAKING NO REPRESENTATION THAT THE PROVIDED IMPLEMENTATION OF THIS DESIGN IS FREE FROM ANY CLAIMS OF INFRINGEMENT BY ANY THIRD PARTY. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, AND XILINX SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE, THE ADEQUACY OF THE IMPLEMENTATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OR REPRESENTATION THAT THE IMPLEMENTATION IS FREE FROM CLAIMS OF ANY THIRD PARTY. FURTHERMORE, XILINX IS PROVIDING THIS REFERENCE DESIGNS "AS IS" AS A COURTESY TO YOU.

XAPP365 - <http://www.xilinx.com/products/xaw/coolvhdlq.htm>

Conclusion

The development tools for both the Handspring PDA and the Xilinx CoolRunner CPLD are easy to understand and use. Interfacing to the Visor port is straightforward as it is essentially the processor bus. The Automotive Scan Tool design submission is a portable, low power design that interfaces with the Handspring Springboard expansion slot. It is an ideal application for the Xilinx XPLA3 CoolRunner CPLD, as no other programmable logic device in the world is capable of delivering the Xilinx patented Fast Zero-Power (FZP) Technology, the combination of ultra-low power and uncompromised speed.

Appendix A

Appendix A lists appropriate Xilinx CoolRunner CPLD application notes. These application notes can be found by searching the Xilinx website and keying on the specific XAPP#. Many include appropriate driver software along with high level design code. All have been constructed and work.

PDA Springboard Design

[XAPP147: Low Power Handspring Springboard Module Design with CoolRunner CPLDs](#)

[XAPP359: Understanding the Insight Springboard Development Kit](#)

[XAPP357: CoolRunner Visor Springboard LED Test](#)

[XAPP355: Serial ADC Interface Using a CoolRunner CPLD](#)

[XAPP146: Designing an Eight Channel Digital Volt Meter with the Insight Springboard Kit](#)

[XAPP149: Designing an Oscilloscope for the Insight Springboard Development Kit](#)

[XAPP349: Pocket Oscilloscope](#)

Additional Application Notes

[XAPP341: UARTs in Xilinx CPLDs](#)

Appendix B: References

Harrison, Dan. "Routine Checkup: An Automotive Diagnostic Scan." *Circuit Cellar* 123 (October 2000).

Handspring website: <http://www.handspring.com/>

Multiplex Engineering, Inc.: <http://www.multiplex-engineering.com/>

The OBD-II Homepage: <http://www.obdii.com/>

Discussion Forum: <http://www.obdii.com/cgi-bin/Ultimate.cgi>

EPA Glossary Site: <http://www.epa.gov/otag/epg/keyterm.htm>

Baum Tools Unlimited, Inc.: www.baumtools.com

AutoTap: www.autotap.com

Ease Diagnostics: www.easesim.com

Delphi Automotive Systems: www.delphiauto.com

Dearborn Technology Group: www.dgtech.com

Interro Systems, Inc.: www.interro.com

Vetronix Corporation: www.vetronix.com

Vishay Intertechnology: www.vishay.com

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/10/00	1.0	Initial Xilinx release.