Features

- **Includes:**
  - **Emulation Software and Hardware**
  - **Development Chip Similar to Production Device**
  - **Built-in Card Reader to Program Card Samples**
- **Final Phase of Development can be Done using a Smart Card Sample**
  - **Flash-based Microcontroller-Production can Start Immediately**
- **Supported on PC (Windows)**
- **Based on IAR Development Chain**

Description

One of the essential characteristics of integrated circuits for Smart Cards is the high level of system security. They have only one input/output through which the interior of the chip is neither visible nor controllable. This, in addition to the problem of security of access, makes application code difficult to test and design.

Atmel's Smart Card Development Kit (SDK) includes a complete set of tools designed to facilitate the development of an AT90SCC application.

The hardware emulator is built around a development version of the AT90SCC device called the Development Chip. The Development Chip is similar in terms of functionality to the production device, but has direct access to internal buses and the possibility of disabling the built-in security features. During emulation, the code runs on the Development Chip CPU. An external RAM, available on the SDK, is used to emulate the internal nonvolatile memory (Flash/EEPROM) during debugging phases. The emulator also features an embedded reader which allows an application to be programmed directly onto card samples or the first production cards, thus making a finished product available shortly after completion of the code. Associated with an External Reader Interface which provides a physical link with an external card reader, this tool allows the user to test the AT90SCC functionality in a real application.

The document "AT90SCC Smart Card Development Kit" describes the Emulator and is provided with the tools.

Smart Card Development Kit (SDK)

AT90SCC SDK Summary

Complete User Guide Available Under NDA and Delivered with Tools

Rev. 1172AS–09/98

Application Validation Flow

Following a rigorous methodology is the most secure route for the validation and integration of software into the AT90SCC devices.

The flow presented below ensures that all the elements of the AT90SCC Development Kit are used in the most efficient manner and guarantees that the application software is fully validated.

Stage 1: Development Chip Emulation with External Memory

A ROM monitor on the SDK is activated and connected with the debugger on the PC.

The application software is loaded into the external RAM memory (used to emulate the internal Flash/EEPROM memory) which is connected to the Development Chip through the additional bus interface. *The AT90SCC development products are capable of emulating the Flash/EEPROM* even while using the external RAM memory.

The ISO Interface of the Development Chip is connected to a remote reader through the External Reader Interface. The application software is then run on the Development Chip.

If the developer is careful to use only the addresses available in the AT90SCC device, the ensuing code is very similar to the final code.

Stage 2: Development Chip External Memory

From this stage, the debugger is no longer used.

The application software is loaded into the emulation RAM memory of the SDK. The ISO Interface of the Development Chip can be connected to a remote reader through the External Reader Interface.

At this point, the user has the possibility of adding a debug code to the external RAM. Code execution can be tracked through an additional RAM.

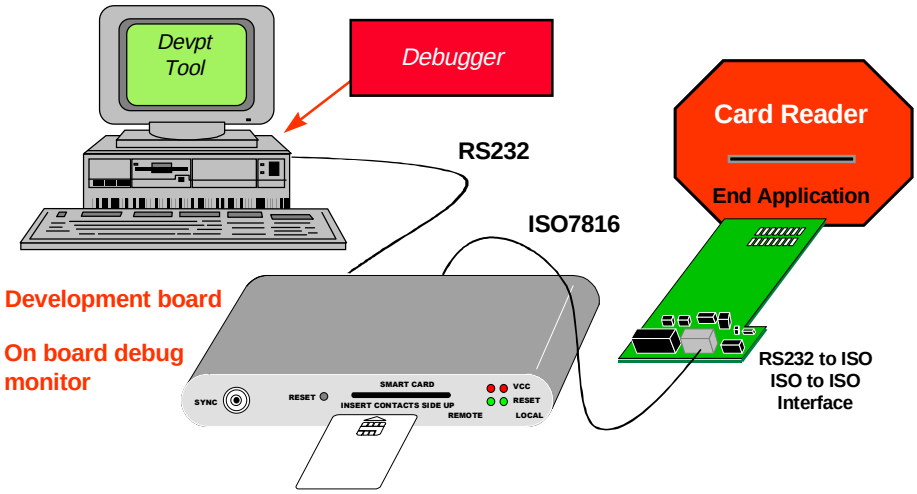
The developer can progressively remove the debugging routines to reach the final code. The program is then transferred from the RAM to the non volatile memory of the Development Chip.

Stage 3: Production AT90SCC

The application is loaded and programmed into the Production Chip through the Embedded Smart Card Reader which can then be inserted in the standard reader for normal use.

Hardware Emulator

Figure 1. AT90SCC Hardware Emulator



The AT90SCC hardware emulator is based on the AT89C52 microprocessor. It includes the following:

- A microcontroller (AT89C52) connected to a PC by an RS232 interface
- An AT90SCC Development Chip
- SRAM memory for SDK supervising
- An EEPROM for development tool storage
- An ISO Interface which allows the AT89C52 to be connected with either the Development Chip or with a Production Chip through the local reader
- A RAM for the breakpoint management
- Peripherals for system configuration
- RAM memory emulation
- A DPRAM for debugger channel
- Line drivers and level shifts for ISO to CMOS conversion

The **DPRAM** allows data to be transferred between the debugger on the PC (connected to the serial port of the 89C52) and the ROM monitor (executed by the Development Chip on the daughter board).

RAM memory emulation can replace the internal memory of the Development Chip to avoid premature aging.

The **breakpoint** memory allows hardware breakpoints to be implemented. When a breakpoint is set by the user, it is stored in the memory at the location corresponding to the address where the breakpoint is set. When the instruction at this address is fetched, a hardware signal is sent to the development chip as an interrupt and stops the program execution. The breakpoints are available only on instruction fetch accesses. The breakpoints on memory read/write are not available.

"SYNC" is an output which can be driven by the application software on the Development Chip. It is connected with the I/O Register of the Development Chip. It can be used to synchronize or trigger an external scope or analyzer. Its management is under user control.

External Reader Interface

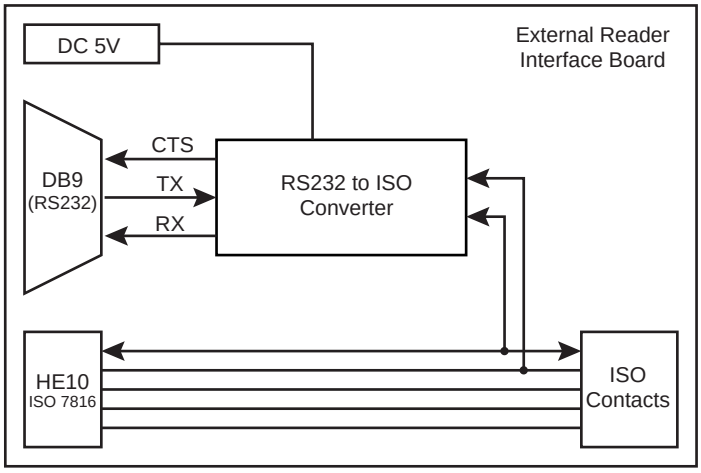
The External Reader Interface is a board which allows an external reader to be connected with a Development Chip on the SDK (ISO to ISO).

It is composed of:

1. An ISO footprint contact for reader connection
2. An HE10 with ISO signals for SDK connection
3. A DB9 for RS232 connection with a PC
4. An ISO to RS232 converter
5. A connector jack to supply power

It can also be used to connect a serial analyzer or spy equipment.

Figure 2. RS232 to ISO Interface Board



Debug Mode

The Smart Card Development Kit has been designed to support the IAR development chain. The emulator is activated by starting the IAR C-SPY debugger.

The ROM monitor and the IAR debugger allow the standard features of an emulator to be executed except trace buffering. The standard features include the ability to:

- Load an application
- Display and modify the contents of the registers
- Display and modify the contents of the data memory
- Disassemble and modify the contents of the code memory
- Set breakpoints
- Execute the application software by stepping or continuous running

Emulation on the SDK allows the application to be validated in a real environment. The External Reader Interface board provides the interface with the remote reader. The SDK is connected to:

- The debugger on the PC by an RS232 cable
- The interface board by an ISO cable (the interface board is plugged into a reader)

In Debug Mode the Development Chip switches between the following:

- User application (the development chip fully emulates the volume chip)
- ROM Monitor (communicates with the IAR C-SPY debugger on the PC)

Switching from the user application to the ROM Monitor is performed by:

- Halting (break) the application from the ROM Monitor
- Stepping the user application
- Encountering a breakpoint when fetching an instruction in the user application

To switch from the ROM Monitor to the user application the commands “step” and “go” are used.

Macro Definition

The C-SPY debugger implements a macro definition to quickly execute complex sequences. These macros have been used to add commands specific to the AT90SCC. These macros are:

- XAF: displays the current value of the XAF register
- READER: sets and displays the ISO Interface connected to the Development Chip.

Breakpoint Management

The breakpoints are managed by hardware. Therefore, the application code is not modified by setting a breakpoint (the user can include a checksum control on the application).

When the AT90SCC encounters an instruction with a breakpoint, the application software is suspended before the breaking instruction is executed. All registers are saved in order to be restored when the application is resumed.

Step Management

The step-by-step execution is also managed by hardware. When executing a step, the true code is executed. There is no software action used. Therefore, executing a step does not modify the code.

ISO Interface Selection

When the ROM monitor is activated at the RESET of the SDK, the Development Chip is systematically connected with the internal ISO Interface on the mother board. The VCC, RESET and CLOCK signals allow the development chip to run but the I/O line is not usable.

Afterwards, two modes of operation are possible:

- Step or run with the ISO VCC of the local interface on the mother board
- Run with the ISO interface of the remote reader through the interface board

Synchronization

The SDK provides an output which synchronizes the external equipment (logic analyzer or scope).

This output is set/cleared by the user application software by setting/clearing the bit TPRT of the I/O register TRIG.

Program Mode

Program Mode requires that a PC be connected to the RS232 serial link. The software must be a terminal emulation (e.g. Hyper Terminal for Windows 95). The baud rate is 19200.

With the Program Mode, the SDK offers a set of menus which allow the user to:

- Test the SDK
- Program an application on the chip
- Load and execute test bench software
- Update the SDK software parts in the EEPROM

These different features are activated by entering commands from the keyboard.

A set of software tools is provided with the SDK. These tools are executed either on the target or on the host.

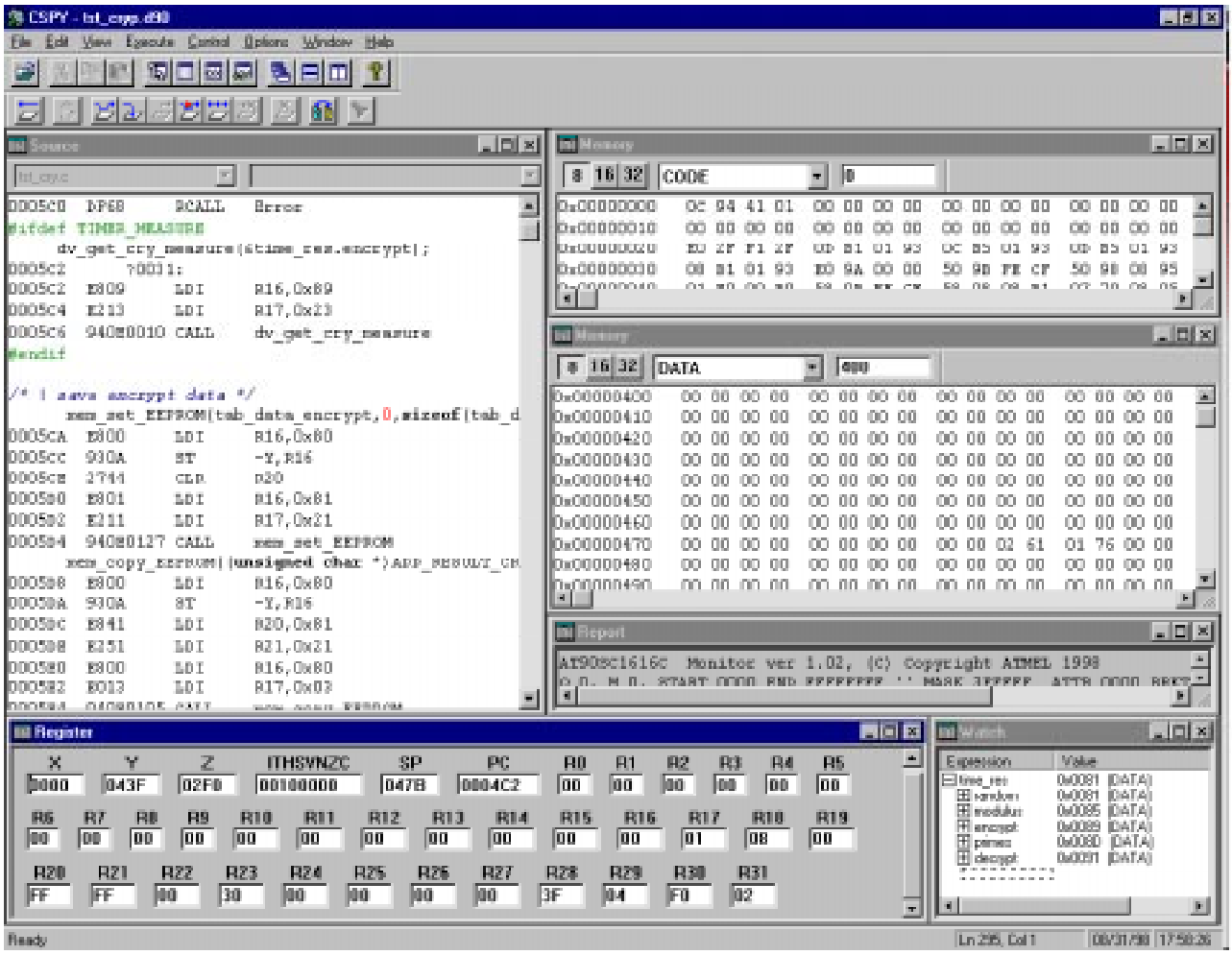
SDK Commands

The SDK is connected to a terminal Emulator on the PC by an RS232 link. The different features are activated by entering commands on the keyboard. These commands are grouped in menus according to their purpose.

Table 1. Summary of Major Development Kit Commands

Command	Description
Dump	Displays on the terminal the contents of the current page (XDATA), showing both hexadecimal and ASCII values
Fill	Fills a part of the current page (XDATA) with a defined value
Load	Uploads data to current page (XDATA). The data can be a: - Test bench or Tool to be activated with GO command - Application in daughter board memory emulation - Application to load onto AT90SCC via the ISO interface
Go	Activates a Test bench or Tool previously uploaded
Page	Modifies the mother board memory mapping
Memory Space	Selects the lower or the upper 64K bytes of the SRAM of the mother board as XDATA space
Window Smartcard	Selects the memory reached on the daughter board
Reader	Selects the reader connected with the Development Chip
Smart Card Control	Activates the menu for the smart card features
Features in EERPOM	Activates the menu for the EEPROM utilities
Tests Hardware	Activates the menu for the tests of the SDK
...	...

Figure 3. Windows Interface of the AT90SCC Software Development Kit



Requirements

Hardware

PC 486 (or above) with 10M bytes Hard Disk Space and one free RS232 port.

Software

- Windows 95 or Windows 3.1
- Note that the use of the SDK requires the user to pre-install the IAR C-SPY debugger. The products of the IAR development chain are the Embedded Workbench (EWA90) and the C-SPY Simulator (the ordinary CWA90). These products are included in the package EWA90 EECCR.

Deliverables

AT90SCC Smart Card Development Kit

- SDK box
- ISO cable for extension board

Diskette

The diskette contains the following:

- AT90SCC User's Manual
- Software tools
- Files which can be saved as models
- Files necessary to update the standard C-SPY debugger with the ROM monitor for the AT90SCC.