

Politechnika Wroclawska

Technika cyfrowa 2

wykład 13

Programy narzędziowe

Katedra Metrologii Elektronicznej i Fotonicznej
Andrzej Stępień



Program - assembler

instrukcja

mnemonik instrukcji [operand1], [operand2], ..

[etykieta:]

[komentarz]

dyrektywa

etykieta - symboliczny adres

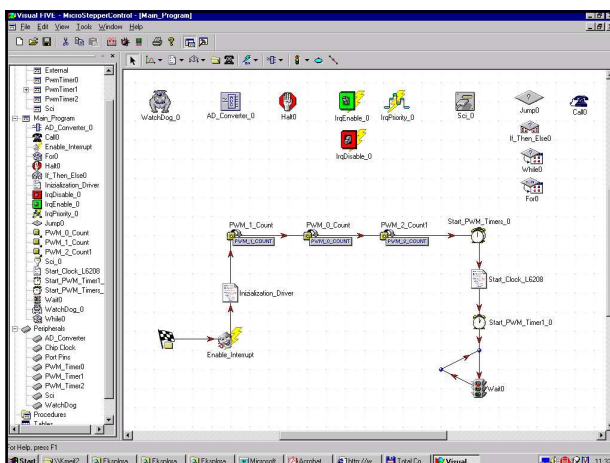
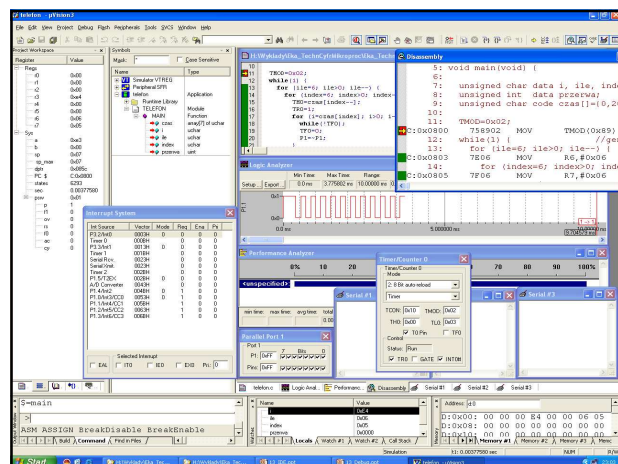
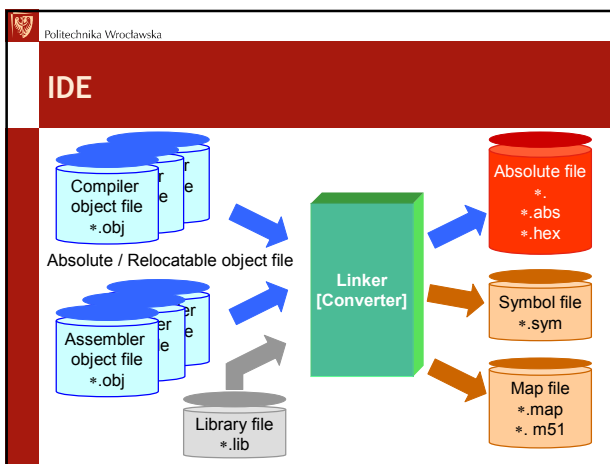
instrukcja - rozkaz wykonywany przez procesor


mnemonik - symboliczna nazwa instrukcji

operand - argumenty instrukcji

dyrektywa - polecenie dla assemblera

komentarz - tekst pomijany przez assembler, od ';' do CR LF





Code Architect

(<http://www.codearchitect.org> - Code Generation Technology by Embedded Systems Academy (1))

- Code Architect is an online and offline technology that allows quick and simple generation of code or data in situations where manual generation may be time consuming or complex.
- With Code Architect, the familiar user interface of web pages are used to provide an intuitive and easy to follow method of configuring and customizing the code or data to be generated. With one click the customized code or data is instantly generated and may then be saved directly into a project. Code Generated may be in any language available (C, Pascal, Modula-2, Visual C++). Data generated may be in any text-based format.
- Code Architect is a generic open-ended system making it completely independent of:
 - The Web Page graphics and appearance
 - The HTML form to input configuration and customization details
 - The algorithm to generate the code or data
 - The actual code or data generated

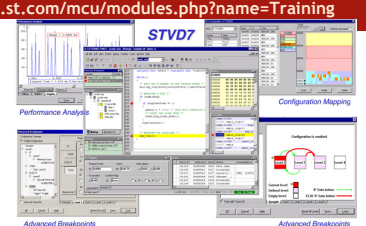
Politechnika Wroclawska

Code Architect (<http://www.codearchitect.org> - Code Generation Technology by Embedded Systems Academy (2))

- This allows Code Architect to be used for a wide range of applications. For example:
 - Configuration of Microcontroller Peripherals
 - Configuration of software stacks such as TCP/IP stacks and CANopen stacks
 - Configuration of projects
 - Generation of lookup tables
 - Generation of raw ASCII data for further processing or use in other applications
- The following links will take you to the current versions of Code Architect available (the "Classic" version is the previous version of Code Architect):
 - Code Architect for Philips P89LPC9xx Microcontrollers Online Version
 - Code Architect for Philips P89LPC9xx Microcontrollers Windows Version
 - Classic Code Architect for Philips P87LPC7xx Microcontrollers

Politechnika Wroclawska

ST7 Training <http://mcu.st.com/mcu/modules.php?name=Training>



Free Solution available on the ST web site or CD ROM. The ST Visual Debug 7 is the ST generic user interface for all ST7 microcontrollers. It is used to build, write and debug the application inside the same friendly environment. Applications can be developed in both Assembly code and/or C language. The STVD7 easily interfaces with Cosmic and Metrowerks C Compilers. For more information, please refer to ST7 Visual Debug User Manual available on ST web site or CD-ROM. STVD7 drives all the ST development tools plus the InDart series.

Politechnika Wroclawska

Microcontroller Programming methods

- In System Programming (**ISP**) occurs when an outside device causes the processor, rather than executing its normal application code, to execute code at a location that contains memory erase and programming routines. This programming takes place with the microcontroller in its normal hardware environment. i.e., soldered on a printed circuit board.
- In Application Programming (**IAP**) is similar to in system programming described above, but rather than having an external device initiate the programming process the normal application code branches to the memory erase and program routines. Similar to In System Programming this operation takes place with the microcontroller in its normal hardware environment.
- Parallel Programming** requires an external programming device. In general it can only be accomplished on a part that is not in its normal hardware environment.

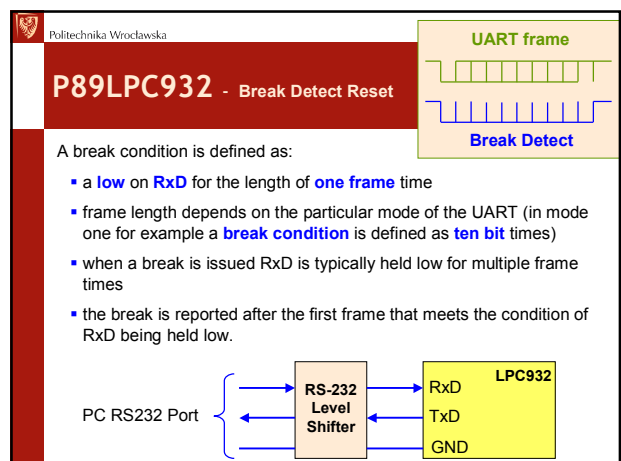
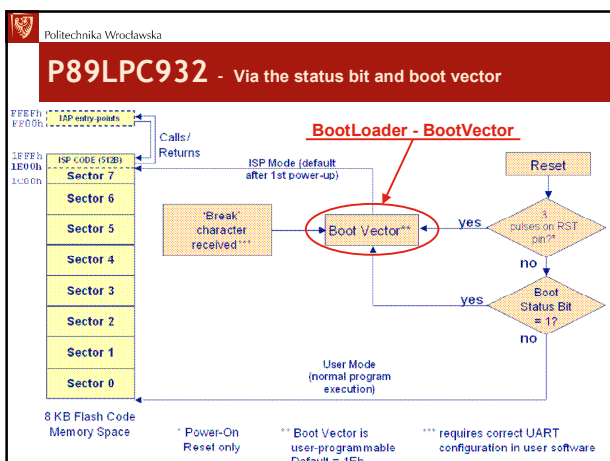
Politechnika Wroclawska

ISP - P89LPC932

- P89LPC932A1.** 8-bit microcontroller with accelerated two-clock 80C51 core, 8 kB 3 V byte-erasable flash with 512-byte data EEPROM. Rev. 02 — 10 May 2005, Philips Semiconductors
- Richard Soennichsen: AN10221.** In-system programming (ISP) with the Philips P89LPC932 microcontroller. 2003 Sep 08, Philips Semiconductors
- AN10258.** How to use the LPC900 In-circuit programming (ICP). Rev. 02 - 21 October 2004, Philips Semiconductors

Entering ISP Mode - There are three ways to enter the ISP mode:

- Via the status bit and boot vector. (Default condition on initial power-up.)
- Through a break detect reset.
- By pulsing the reset pin upon power-up. (Hardware Activation)

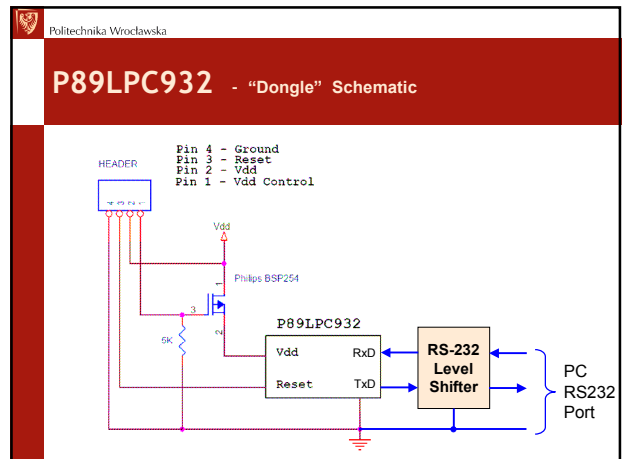


Politechnika Wrocławska

P89LPC932 - Hardware activation

- This mode of ISP entry is always available regardless of user code or the state of the status bit. (Assuming that the boot loader code is intact and the boot vector is 1EH)
- By presenting a timed waveform of low-going pulses to the reset pin after power up, the part will begin code execution at the address pointed to by the boot vector.
- This entry mode has the same effect as having a non-zero status byte.

SYMBOL	PARAMETER	Min	Max
t_{VR}	RST delay from VDD active	50	—
t_{RH}	RST HIGH time	1	32
t_{RL}	RST LOW time	1	—



Politechnika Wrocławska

TUSB3410 - Texas Instruments

TUSB3410, TUSB3410I. USB to Serial Port Controller. Data Manual. September 2005, Texas Instruments, SLLS519D

- The TUSB3410 contains onboard ROM microcode (BootLoader), which enables the MCU to enumerate the device as a USB peripheral. The ROM microcode can also load application code into internal RAM from either external memory via the I²C bus or from the host via the USB.
- After reset, the TUSB3410 is disconnected from the USB.
- The TUSB3410 checks the I²C port for the existence of valid code; if it finds valid code, then it uploads the code from the external memory device into the RAM program space. Once loaded, the TUSB3410 connects to the USB and enumeration and configuration are performed. This is the most likely use of the device.
- If the valid code is not found at the I²C port, then the TUSB3410 connects to the USB and then an enumeration and default configuration are performed.

Politechnika Wrocławska

ISP - AVR Family (1/2)

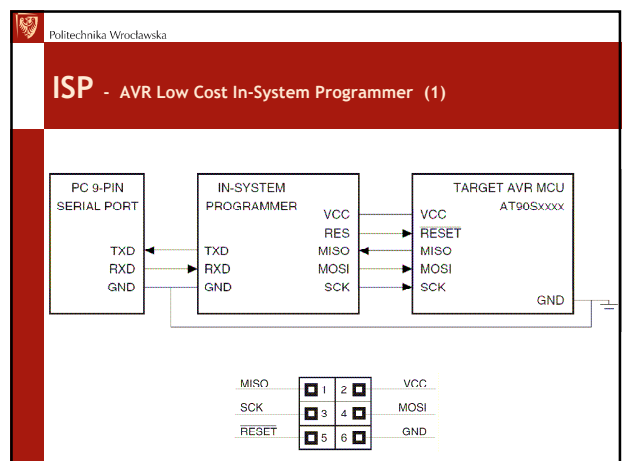
Application Note AVR910. In-System Programming. Rev. 0943C-11/00, Atmel Corporation

- In-System programming allows programming and reprogramming of any AVR microcontroller
- Using a simple 3-wire SPI interface, the In-System programmer communicates serially with the AVR microcontroller, reprogramming all nonvolatile memories on the chip.
- The Serial Peripheral Interface (SPI) consists of three wires:
 - Serial Clock (SCK),
 - Master In – Slave Out (MISO)
 - Master Out – Slave In (MOSI).
- When programming the AVR, the In-System programmer always operate as the master, and the target system always operate as the slave.

Politechnika Wrocławska

ISP - AVR Family (2/2)

- To allow programming of targets running at any allowed voltage (2.7V - 6.0V), the programmer can draw power from the target system (VCC).
- The target AVR microcontroller will enter Serial programming mode only when its reset line is active (low).
- Immediately after Reset has gone active, the In-System programmer will start to communicate on the three dedicated SPI wires SCK, MISO and MOSI.
- The AVR microcontroller will automatically set all its I/O pins to inputs, with pull ups disabled, when Reset is active.





ISP - AVR ATmega8(L) - Device Clocking & Serial Programming

JTAG (1/2)

JTAG (2/2)

- ## JTAG - ATmega16(L) Microcontroller with 16K Bytes ISP Flash

4

Politechnika Wroclawska


In-Circuit Emulator® - SIGNUM SYSTEMS (2)

USP-51 Emulator:

- Non-intrusive emulation up to 60 MHz
- Replaceable processor probes (PODs) support virtually all 8051 derivatives from Philips, Atmel, Atmel W & uC, Infineon etc.
- Parallel and Serial Port connectivity

USP-51 Emulator:

- Trace Buffer with timestamp
- Advanced Complex Events (Events, Counters, Sequencer, address and data match Comparators)
- Unlimited number of breakpoints
- Banked memory support
- Dual-ported RAM to access memory while application is running



Politechnika Wroclawska

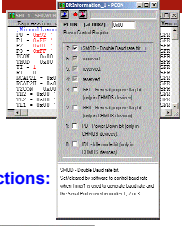
In-Circuit Emulator® - SIGNUM SYSTEMS (4)

Windows used for assembly debug:

- Source window for assembly language code display, breakpoint setting and other
- Watch, Status and SFR windows allow examine/ modification of CPU and CPU peripherals registers
- SFR Bit Information window shows detailed information about SFR bits
- Symbol Explorer window allows to declare user defined symbols and browse/search for symbols

Assembly language execute actions:

- CPU instruction single step
- CPU instruction step over
- Subroutine step out
- Go till CPU instruction at mouse cursor
- Go till breakpoint(s)
- Skip CPU instruction without execution
- Set new execution point at mouse cursor
- In-line code assembly



Politechnika Wroclawska

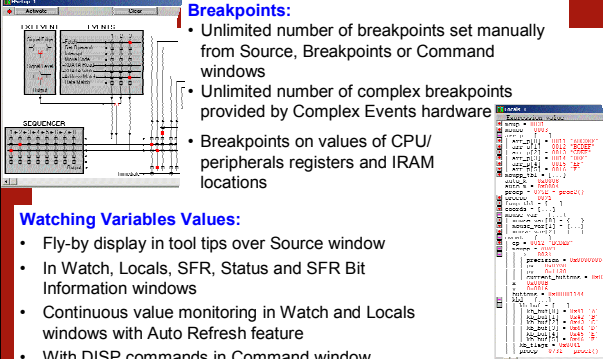
In-Circuit Emulator® - SIGNUM SYSTEMS (5)

Breakpoints:

- Unlimited number of breakpoints set manually from Source, Breakpoints or Command windows
- Unlimited number of complex breakpoints provided by Complex Events hardware
- Breakpoints on values of CPU/ peripherals registers and IRAM locations

Watching Variables Values:

- Fly-by display in tool tips over Source window
- In Watch, Locals, SFR, Status and SFR Bit Information windows
- Continuous value monitoring in Watch and Locals windows with Auto Refresh feature
- With DISP commands in Command window



Politechnika Wroclawska

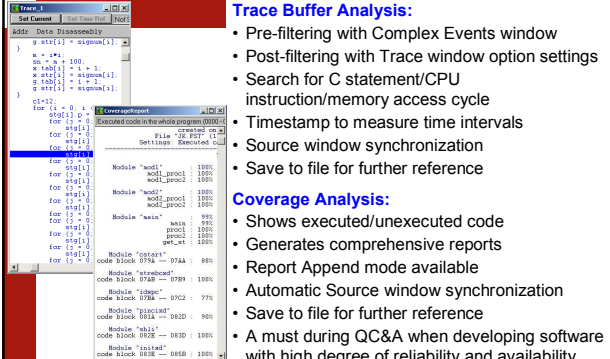
In-Circuit Emulator® - SIGNUM SYSTEMS (6)

Trace Buffer Analysis:

- Pre-filtering with Complex Events window
- Post-filtering with Trace window option settings
- Search for C statement/CPU instruction/memory access cycle
- Timestamp to measure time intervals
- Source window synchronization
- Save to file for further reference

Coverage Analysis:

- Shows executed/unexecuted code
- Generates comprehensive reports
- Report Append mode available
- Automatic Source window synchronization
- Save to file for further reference
- A must during QC&A when developing software with high degree of reliability and availability



Politechnika Wroclawska

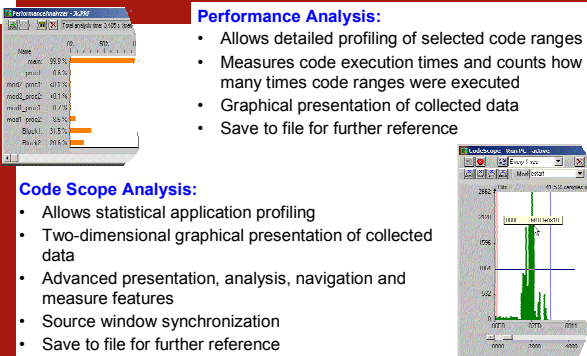
In-Circuit Emulator® - SIGNUM SYSTEMS (7)

Performance Analysis:

- Allows detailed profiling of selected code ranges
- Measures code execution times and counts how many times code ranges were executed
- Graphical presentation of collected data
- Save to file for further reference

Code Scope Analysis:

- Allows statistical application profiling
- Two-dimensional graphical presentation of collected data
- Advanced presentation, analysis, navigation and measure features
- Source window synchronization
- Save to file for further reference



Politechnika Wroclawska

In-Circuit Emulator® - SIGNUM SYSTEMS (8)

Problem: How can I trap my program on attempts to read un-initialized memory locations?

Solution:

Before loading your program fill the XDATA memory (where the program variables are located) with a constant pattern (e.g. 0x66) and set Complex Events breakpoint whenever data read from XDATA is equal to the 0x66.

Although this does not fully guarantee that the memory was not initialized, finding the right constant value for filling XDATA may give excellent results. Here breakpoint was triggered when `stg[0].tab[s]` uninitialized variable was used.

