C for AVR® Microcontrollers CD ROM

What does it do?

This is a self contained CD ROM for learning C programming for AVR microcontrollers
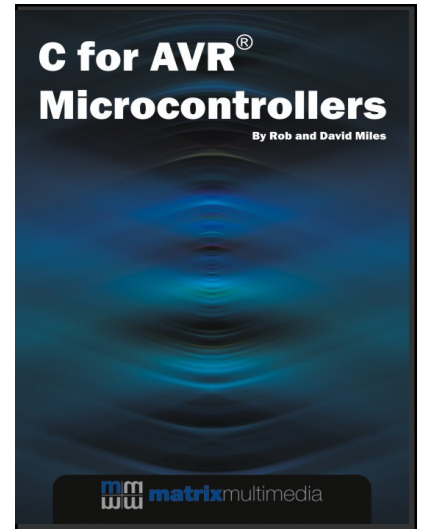
Benefits

- Unique microcontroller Simulations shortens learning curve
- This is a complete solution for learning including tutorials, IDE and compiler

Features

- A full course in C programming
- Includes C compiler and IDE
- Includes virtual ATmega32 microcontroller simulations
- Links to E-blocks™ AVR multiprogrammer

CD ROM is shipped in a DVD case

Description

This new CD ROM is designed for those who want to learn how to program embedded microcontrollers in C. The CD contains a full course as well as all the software tools needed to create hex code for a range of AVR ATmega devices - including a full C compiler and an Integrated Development Environment (IDE). The CD ROM makes extensive use of a virtual C microcontroller which uses code simulation to explain how C works: students can step through programs to see the effects of each line of code on the microcontroller. This product is designed to work seamlessly with our E-blocks AVR Multiprogrammer, but will also operate with third party programmers. Written by David and Rob Miles of Durham and Hull Universities.

Learning time

Approximately 40 hours

Prerequisites

- An understanding of digital electronics
- Windows skills
- Elementary programming skills

Included on the CD ROM

- Complete course in C programming with exercises
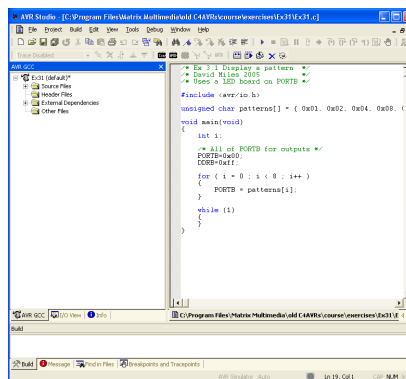- AVR Studio IDE
- GCC C compiler

How to use this CD ROM

This CD ROM can be used as a programme of self study at home or in industry. It is also suitable for use with undergraduates as part of a structured course, thereby freeing up lecturers to provide one-to-one tutorial assistance.

Screen images
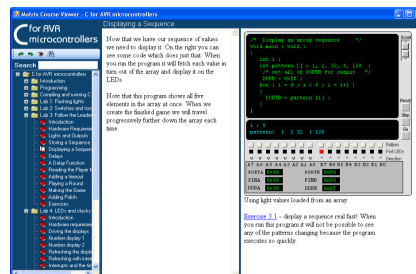
Typical tutorial screen showing the Virtual C microcontoller

IDE and compiler screen

Also consider

EB194 AVR programmer
EB219 AVR starter pack
EB343 Deluxe AVR starter pack

AVR Multiprogrammer

AVR Deluxe E-blocks starter pack

C for AVR® Microcontrollers CD ROM

Learning objectives

Study of the CD ROM—will achieve the following objectives:

- Gain a thorough understanding of C programming for microcontrollers from basic techniques through to advanced concepts such as serial communication, and interrupts.
- Develop the skills and techniques required to write C programs of some complexity from scratch.
- Develop a good understanding of how electronic systems are controlled.

CD ROM contents

The CD ROM is divided into two sections: a suite of Labs with real applications, and a section on C programming:

Labs contents

Lab 1: flashing lights
VOID, statements, hexadecimal, output to ports, WHILE, FOR, #INCLUDE, DDRA, variables

Lab 2: Switches and torches
INPUT, IF, ==, !, debouncing inputs, functions

Lab 3: Follow the leader
Arrays, delays, unary operators, break

Lab 4: LEDs and Clocks
Const, interrupt, 7-segment displays, multiplexing

Lab 5: LCDs and libraries
Serial data comms, libraries of code

Lab 6: Mystic LCD
Random numbers, shift register, MOD, test harnesses and debugging

Lab 7: The E lock
EEPROM memory, sleep modes,

Lab 8: reaction timer
Accurate timing

Lab 9: Inspect your morse
Converting state diagrams to programs

C Programming contents

What is a C program?
Introduction, Comments, The main function header, The main function body, The end of the line, Exercises,

Variables
Introduction, Types, Floating point, Characters, Integers, Choosing variables, Identifiers, Declaration, Multiple declarations, Initialization, Names, Assigning statements, Expressions, Operators and Operands, Constant operands, Operators and divide, Working on bits, Shifting bits, Shortcut operators, Unary operators, Casting, Exercises

Conditional Statements
Introduction, Logical operators, Equality, Using else, Cunning conditions, Complicated conditions, Unary operators in conditions, Exercises.

Statements and Blocks
Introduction , Blocks in Blocks, Global and local variables, Local variable scope, Global variables, Exercises.

Looping the loop
Introduction, While loop, Counting with the while loop, The for loop, Breaking out of loops, Continuing loops, Exercises.

Functions
Introduction, When to use a function, Functions which return values, Functions which accept parameters, Functions which return values, More than one parameter, Exercises.

Arrays
Introduction, The need for arrays, Declaring an array, elements in an array, Sorting array elements, Into the next dimension, Exercises.

Switches
Introduction, The switch condition, Switches and breaks, Exercises.

Pointers
Introduction, Arrays and pointers, Declaring a pointer, Using a pointer, Comparing a pointer, Null pointers, Functions and pointers, Finding out the size, Strings, Exercises.

Structures
Introduction, Creating structures, How structures work, Pointers in structures, Structures in structures, Exercises.

The pre-processor
Introduction, The #include directive, magic numbers and #define, Conditional compilation, Exercises.

Software engineering in C
Introduction, Deciding on specification, Deciding how to test, creating the functions, Using the functions, Making a project, External data.

Sections on AVR device reference also included.

Versions available

E:LCVRST Student/home version
ELCVRSI Single user version
ELCVRSL Site licence version

Note that student versions are missing selected exercises and content more applicable to institutions. Student/home version are not available to educational institutions or companies.

System requirements

PC with CD ROM drive and Windows 98 or greater. Site licence version is compatible with all major network configurations.

Hardware requirements

Atmel AVR programmer with ATmega32 device and download utility—or E-blocks AVR Multiprogrammer.