E-LAB Computers

AVRco

Pascal System

Features

Pascal Compiler for Atmel's AVR 8Bit RISC CPU's, 90S2313, 90S8515, Tiny13, Mega128 etc. Nearly complete implementation of the Pascal standard

Floating point: SIN, COS, PWR, PWR10, SQRT, LOG2, LOG10

Additional strong support of bit manipulation.

Many features especially for embedded control applications.

Most of the On-Chip peripherals are implemented by comfortable library routines:

UART, ADC, PWM, Timer

Additional drivers implemented by software: LED7Seg Display, LCD-display, I2C-BUS, Stepper, SwitchPorts, Triggered ports, LAN

High level support of internal EEprom

Complete **PID**-controllers impl.
Multiprocessing Kernel

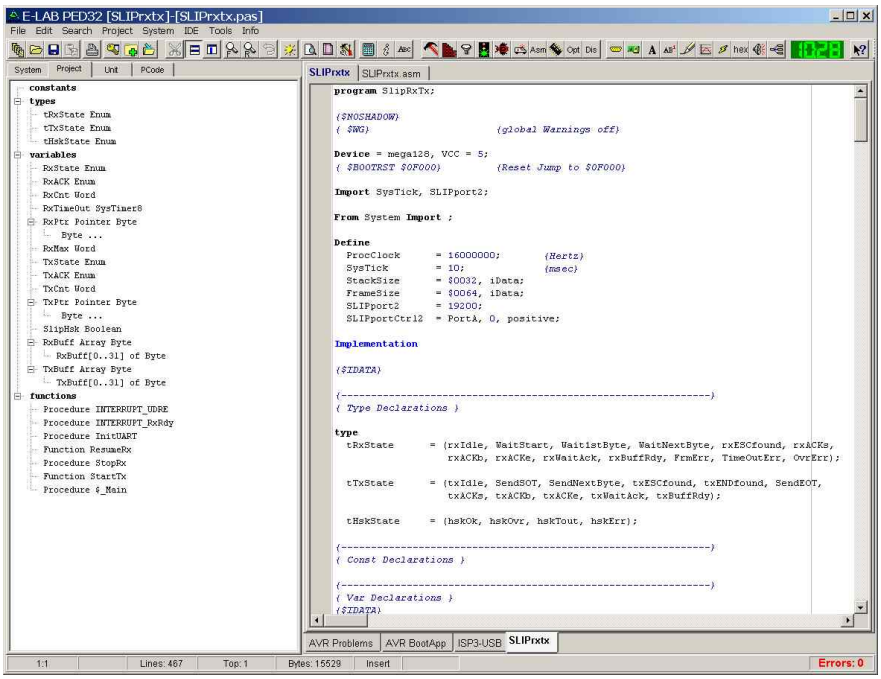
Included Simulator/JTAG ICE debugger on high level language. Single step on Pascal statements

Assembler statements can be included in Pascal source.

Multi-window editor with comfortable project management. Syntax and error highlighting.

Completely configurable environment Simulator included. Application Wizzard

12 months free update
free demo versions



Product Information

The AVRco is a **Multi-Task** Pascal development system for the AVR family of microcontrollers. All devices excl. 90S1200 are supported. The system was specially designed for the needs of the designer of embedded systems. Therefor many extensions for single chips are included.

AVRco consists of a mighty multi window editor, an Application Wizzard, the Pascal compiler, the assembler, simulator and an Incircuit-programmer/ICE.

The editor resp. IDE is completely configurable, has configurable syntax and error highlighting, unlimited filesize, multiwindow. It's heavily project oriented. Automatic reload of faulty source files. Cursor will be positioned to the incorrect syntax position. Online help of the editor functions. Context sensitive online help for the Pascal syntax. Nearly unlimited undo/redo.

The Pascal compiler supports all standard definitions and many more. **Types:** Bit, Boolean, Byte, Char, Enum, Word, Integer, Pointer, String, Array, Record, Float, SysTimer, Pipe, Semaphore.

Operators: not, div, mod, and, or, xor, shl, shr, rol, ror, in
System functions: Lo, Hl, Abs, Bit, Incl, Excl, Toggle, SetBit, Inc, Dec, Swap, Odd, Length, SizeOf, Delay, FillBlock, CopyBlock, WatchDog, Sleep, CpuSleep, EnableInts, DisableInts, IntToStr, ByteToStr, IntToHex, ByteToHex, FloatToStr, StrToFloat, StrToInt etc.

Statements: Begin, Return, End, If, Then, Else, Elsif, Endif, Label, Goto, Case, For, While, Repeat, Loop, Type, Const, Var, Procedure, Function, With, True, False, Nil, Import.

Complete interrupt support and handling. Optional runtime errorhandling of software stack and string/array checks. Internal EEprom can be accessed as normal defined vars or as an array of bytes.
The compiler supports structured constants, forward declaration, conditional compile and also assembler statements.

The simulator/debugger/emulator is completely integrated. Single steps and breakpoints on Pascal statements. Program variables can be examinated with their Pascal names in the watch window.
Can be used as an in-Circuit emulator ICE

Contact:

E-LAB Computers

Grombacherstr. 27
D74906 Bad Rappenau
Germany
Tel. (49) 7268 9124 0
Fax. (49) 7268 9124 24
email: info@e-lab.de
http://www.e-lab.de

AVRco Pascal AVR Multitask Development

Types Boolean, Byte, Char, Bit, String, Array, Word, Integer, Enum, Procedure, Pointer, LongInt, LongWord, Float, Record, Semaphore, Pipe, SysTimer, PIDcontrol

Operants not, div, mod, and, or, xor, shl, shla, shr, shra, rol, ror, in, •, +, -

Keywords Program, From, Import, Device, Define, Const, StructConst, Var, Nil, Implementation, Procedure, Function, Process, Task, Interrupt, Trap, true, false, Begin, Return, Exit, End, ASM, EndASM, if, then, else, elsif, endif, while, repeat, break, until, Loop, ExitLoop, EndLoop, for, to, downto, by, case, exit, Label, Goto

System Library Lo, Hi, Abs, Odd, Swap, UpCase, Val, Ord, Min, Max, Random, Length, SizeOf, Incl, Excl, SetBit, Bit, Toggle, Inc, Dec, Lower, Higher, WithIn, Trunc, Round, Frac, Sqr, Sqrt, Pow, Pow10, Exp, Deg2Rad, Rad2Deg, ArcTan, Tangens, Sin, Cos, Log2, Log 10, FillBlock, CopyBlock, StrToInt, StrToFloat, FloatToStr, IntToStr, ByteToStr, LongToStr, ByteToHex, IntToHex, LongToHex, PipeSend, PipeRecv, PipeStat, PipeFull, WaitPipe, PipeFlush, EnableInts, DisableInts, CPUSleep, Sleep, WatchDogInit, WatchDogTrig, Suspend, Resume, Lock, UnLock, Priority, Main_Priority, Schedule, WaitSema, SendSema, SemaStat, Addr, FlushBuffer, PID.Required, PID.Actual, PID.pFactor, PID.iFactor, PID.dFactor, PID.sFactor, RunTimeErr, SwitchP1, SwitchP2, Port_Stable1, Port_Stable2, Inp_Stable1, Inp_Stable2, Inp_Raise1, Inp_Raise2, mDelay, uDelay, Write, Read, LCD, LCDout, LCDinp, LCDctrl, LCDstat, RX_Buff, TX_Buff, SERinp, SERout, SERstat, ADCport, GetADC, PWMPort1, PWMPort2, PWMout1, PWMout2, I2CPort, I2Cinp, I2Cout, I2Cstat, Disp7Seg, Eeprom, Stepper.

Processes and **Tasks**
AVRco includes a multitasking system, which is supported by an amount of functions and procedures. Up to 15 processes and tasks can be defined, which are periodically invoked by the scheduler, dependend on their priority and status.

Jobs can be done in background without intervention of the main program. The processes can communicate via pipes and semaphores. Tasks are specialized processes, which are cyclically invoked with a fixed time delay, so they can do such periodic jobs like PIDcontrols.

Multitasking is an extraordinary way to solve many problems appearing in embedded applications. The most development systems don't support it or the multitasking is sold with extra cost. AVRco **includes Multi Tasking** in a streamlined way.

Hardware support
The most hardware functions, like UART, ADC, PWM etc. are completely supported with functions and procedures by the system library. The internal EEprom, if present, is treated like a normal variable, but the special access-modes of the CPU are used. So the user must not take care about it. Initialization and dealing with such CPU-parts in most cases is not an easy thing and doing this in HLL is slow and very ROM consumptive.

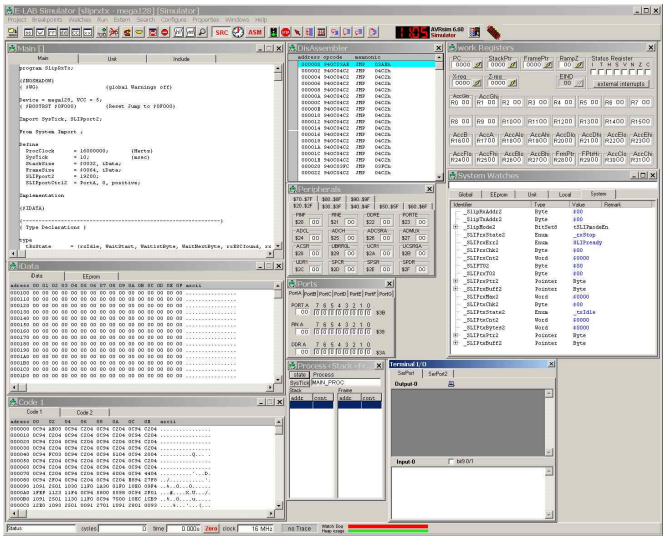
Several additional hardware functions, which are not supported by the CPU, like I2C, LCD, 7seg-Display, STEPPER-motor etc are also supported by the library. The PID-controller for temperature, speed or position control, which is very hated by the programmers, is also implemented. Each of this function is completely written in assembler and therefore very compact and fast. They also make the source code, because of simple function calls, shorter and readable.

Normally, these system functions are as double so fast as HLL handmade and take the half of the ROM space as their counterparts. So these features of the compiler are a big advantage against „pure" compilers.

Standard Version
This version includes all functions and drivers listed above.

Profi Version
This version includes all drivers and functions of the Standard Version but provides modular programming with Units.

In addition there are such powerful drivers like LCD Graphic, Ethernet, CAN, FAT16, ModBus, USB etcetc.



Free Demo Version, free Mega8/Mega88 Version

Prices Compiler, IDE, Assembler, Simulator, Application Wizard, several Tools, 5 Manuals, Incircuit-programmer/ICE

Standard Version	€481.- +ship
Profi Version	€940.- +ship

12 months free updates. If expired then 25% of the actual price.

E-LAB Computers Grombacherstr. 27 D74906 Bad Rappenau Tel 07268/9124-0 http://www.e-lab.de/