# LadderWORK Ver. 1.20 Overview

LadderWORK

LadderWORK is the easiest way to create automation control programs. Use of LadderWORK is immediate. With the use of the mouse only, you simply place functional objects in your worksheet , connect the components with wires and configure the components property. Microprocessor assembler code will be generated at the simply push of the BUILD button.

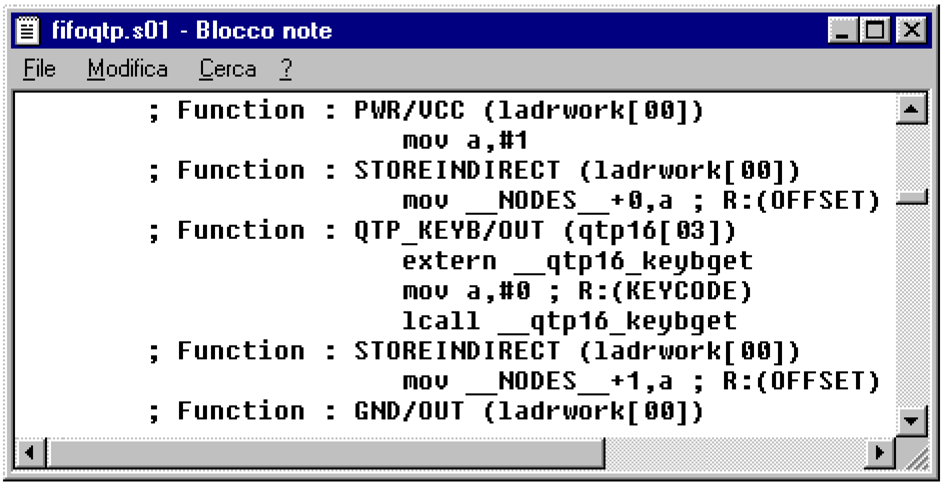
LadderWORK software integrate a powerful schematic editor with multi view feature and context-sensitive help.

LadderWORK's generated code is really efficent. Microprocessor's assembly code it's directly generated by the compiler so no other instructions charge will affect your result ( NO 'C' SOURCE GENERATED AND COMPILED PROCESS ). In this way with LadderWORK you always are sure to obtain the best size & speed optimized code.

A great number of build-in functional components are ready to be placed in your project. LadderWORK software includes a standard set of LADDER DIAGRAM (RELAY LOGIC) devices and a set of extra components, like pure-logical ports and user-programmable functions.

Full ADVANCED version includes over 70 devices : input/output devices, relays, d-type flip flops, debouncers, clock generators, delay lines, up/down counters, comparators, fifo/lifo queues, A/D & D/A converters, and/or/not logical ports and user programmable functions.

LadderWORK produce a Intel-Std HEX file as output. Also intermediate assembler and listing files are available as output of LadderWORK compile process so you can check instruction by instruction the generated code. Many PLC devices supported by LadderWORK software can directly upload the generated code simply pushing the UPLOAD button.



```
        ; Function : PWR/VCC (ladrwork[00])
                mov a,#1
        ; Function : STOREINDIRECT (ladrwork[00])
                mov __NODES__+0,a ; R:(OFFSET)
        ; Function : QTP_KEYB/OUT (qtp16[03])
                extern __qtp16_keybget
                mov a,#0 ; R:(KEYCODE)
                lcall __qtp16_keybget
        ; Function : STOREINDIRECT (ladrwork[00])
                mov __NODES__+1,a ; R:(OFFSET)
        ; Function : GND/OUT (ladrwork[00])
```

## MORE THAN A LADDER LANGUAGE

Ladder standard language is strongly rigid. Components must be forced in predeterminated cells along two rails called rungs. Moreover Ladder standard language has great limitations about feedback connections. LadderWORK broken these limitations introducing the first free schematic ladder diagram. LadderWORK includes a powerful schematic editor. Components can be placed anywhere and there isn't limitation on feedback connections. LadderWORK schematic is more similar to an electrical circuit. Moreover LadderWORK includes extra components like logical ports and flip-flops so if you are well-versed in boolean logic you can approach your problem using these traditional notations.

## ELECTRICAL NOTATION APPROACH

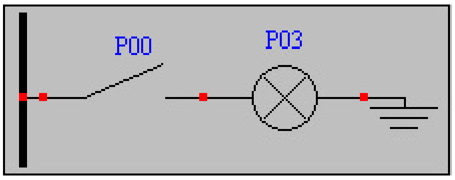
LadderWORK is remarkably intuitive !

With LadderWORK you haven't to know nothing about assembler, interrupts or hardware architectures. All you have to do is think your project as a electrical scheme where you have to disposition switches, relays and lamps. Switches means inputs, lamps means outputs and relays gives the way to create states and elementary memory cells. Many problems related to control automation can be resolved in few minutes using LadderWORK.

## PUT A PLC INTO YOUR MICROCONTROLLER !

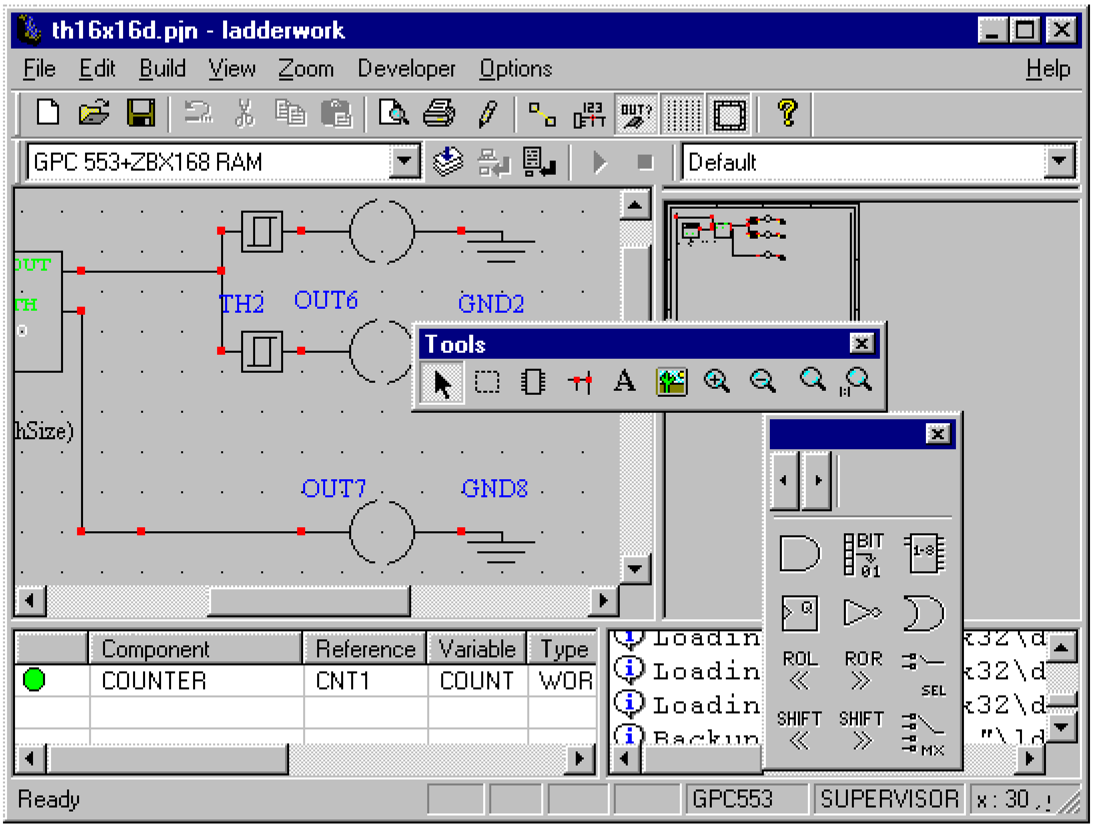
With LadderWORK you can transfrom a microcontroller in a PLC . Microcontroller support is activated simply selecting your MPU model during project setup. For example if i select the 8051 MPU the system put to disposition all the 8051 chip I/O resources. So when i configure my input or output devices i directly see the resource named P.0 .. P.7 which means that the system will drive the relative hardware pin .

The picture shows a simple schematic where the value of the 8051 pin P0.0 is transferred directly to the pin P0.3



## ROBUST 32 BIT ARCHITECTURE

LadderWORK is enterily written in C/C++ Language with full 32 bit architecture. The C compiled code gives great performances in terms of code compactness and process speed.

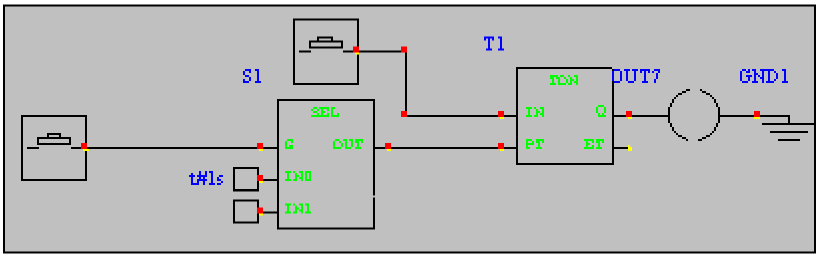


| | Component | Reference | Variable | Type |
|---|---|---|---|---|
| ● | COUNTER | CNT1 | COUNT | WOR |

Ready | GPC553 | SUPERVISOR | x: 30

## IEC1131-3 Standard programming

LadderWORK software is currently supporting many symbols compliant with the IEC/CEI 1131-3 directive. Our goal is the full implementation of this directive.



## LIBRARY COMPONENTS OVERVIEW

The table below shows the available library components according to the related software version.

| Software Version | Library Components |
|---|---|
| BASE | INPUT, EINPUT, NCINPUT, ENCINPUT, DEBOUNCE, OUTPUT, EOUTPUT, IPIN, OPIN, RELAY, PROBE |
| STANDARD | *All the components included in the BASE version plus ...* |
| | CLOCK, COUNTER, DELAY, THRESHLD, AND, FFD, NOT, OR, TP, TON, TOF, R_TRIG, F_TRIG, SR, RS, ASSIGN, READVAR, CONST, IDENT |
| ADVANCED | All the components included in the STANDARD version plus... |
| | AD_CONV, PWMOUT, FIFO, LIFO, USER1, USER2, USER3, ADD, SUB, MUL, DIV, MOD, SHL, SHR, ROL, ROR, BIT, DEC1-8, CTU, CTD, CTUD, TMI, TSQ, SEMA, DISPLAY, KEYBOARD, FIELD, SEL, MIN, MAX, LIMIT, MUX. |

| Component | Brief Description | Software Version |
|---|---|---|
| **Digital input / output functions** | | |
| EINPUT | Normally open electrical notation general purpose input | BASE |
| ENCINPUT | Normally closed electrical notation general purpose input | BASE |
| INPUT | Normally open ladder standard notation general purpose input | BASE |
| NCINPUT | Normally closed ladder standard notation general purpose input | BASE |
| EOUTPUT | Electrical notation general purpose output | BASE |
| OUTPUT | Ladder standard notation general purpose output | BASE |
| RELAY | General purpose relay coil | BASE |
| DEBOUNCE | Spike noise filter for phisical inputs | BASE |
| IPIN | Microprocessor input pin | BASE |
| OPIN | Microprocessor output pin | BASE |
| **Analogs** | | |
| AD_CONV | Analog to digital converter (1) | ADVANCED |
| PWMOUT | Digital to analog converter (1) | ADVANCED |

## Timing generators

| | | |
|---|---|---|
| CLOCK | General purpose clock generator ( up to 10Hz ) | STANDARD |
| TSQ | Square wave generator | ADVANCED |

## Counters

| | | |
|---|---|---|
| COUNTER | General purpose up/down counter with threshold output | STANDARD |
| CTU | IEC1131-3 standard up counter | ADVANCED |
| CTD | IEC1131-3 standard up counter | ADVANCED |
| CTUD | IEC1131-3 standard up/down counter | ADVANCED |

## Delay units

| | | |
|---|---|---|
| DELAY | General purpose delay unit with monostable or single-pulse working mode. | STANDARD |
| TP | IEC1131-3 standard pulse-timer | STANDARD |
| TON | IEC1131-3 standard on-timer | STANDARD |
| TOF | IEC1131-3 standard off-timer | STANDARD |
| TMI | Integral timer | ADVANCED |

## Logicals

| | | |
|---|---|---|
| AND | Logical AND function | STANDARD |
| NOT | Logical NOT | STANDARD |
| OR | Logical OR | STANDARD |
| SHL | Logical shift left | ADVANCED |
| SHR | Logical shift right | ADVANCED |
| ROL | Logical rotate left | ADVANCED |
| ROR | Logical rotate right | ADVANCED |
| BIT | Extract a single bit from a 16 bit WORD | ADVANCED |

## Mathematical

| | | |
|---|---|---|
| ADD | Addition | ADVANCED |
| SUB | Subtraction | ADVANCED |
| MUL | Multiplication | ADVANCED |
| DIV | Division | ADVANCED |
| MOD | Modulo, rest of the division | ADVANCED |
| MIN | IEC1131-3 standard. Compute the minimum value of the two applied signals | ADVANCED |
| MAX | IEC1131-3 standard. Compute the maximum value of the two applied signals | ADVANCED |
| LIMIT | IEC1131-3 standard. Limits the input value inside the applied range. | ADVANCED |

## Flip-Flops

| | | |
|---|---|---|
| FFD | Logical D-TYPE flip-flop | STANDARD |
| SR | IEC1131-3 standard set-dominant SR flip-flop | STANDARD |
| RS | IEC1131-3 standard reset-dominant SR flip-flop | STANDARD |
| SEMA | IEC1131-3 standard semaphore | ADVANCED |

| | | |
|---|---|---|
| **Decoders and multiplexers** | | |
| DEC1-8 | 1 to 8 decoder | ADVANCED |
| SEL | IEC1131-3 standard selector | ADVANCED |
| MUX | IEC1131-3 standard multiplexer | ADVANCED |
| **Comparators and threshold detectors** | | |
| THRESHLD | Threshold detector. Compares the input value with a programmed constant. | STANDARD |
| R_TRIG | IEC1131-3 standard raising-edge detector | STANDARD |
| F_TRIG | IEC1131-3 standard falling-edge detector | STANDARD |
| **Queues** | | |
| FIFO | First-in First-out method queue | ADVANCED |
| LIFO | Last-in First-out method queue | ADVANCED |
| **Display and keyboard interface** | | |
| DISPLAY | General purpose display driver block | ADVANCED |
| KEYBOARD | General purpose keyboard driver block | ADVANCED |
| FIELD | General purpose data entry block | ADVANCED |
| **Assembly user programmable functions** | | |
| USER1, USER2, USER3 | Assembly user programmable functions | ADVANCED |
| **Variables, literally and identificators** | | |
| ASSIGN | Assigns a value to a variable | STANDARD |
| READVAR | Reads a variable assigned with ASSIGN | STANDARD |
| IDENT | IEC1131-3 standard identificators | STANDARD |
| **Probing and watching** | | |
| PROBE | Probes the value of the connected path | BASE |
| | | |

**Notes :**

*(1) Functions available only for PLC with dedicated analog interfaces*

## And more ...

Full 8051 ADVANCED version includes USER FUNCTIONS so if you need a particular device you can write your assembly routine for your needing.

LadderWORK run-time kernel is really small and the system requirements are minimum. For example, in a 8051 system the hardware requiments are just the TIMER 0, used for global timing, and less than 20 bytes of internal RAM including stack area.

8051 version of LadderWORK can be configured to use internal or external RAM with customizable memory mapping.

## MINIMUM SYSTEM REQUIREMENTS

Personal computer Pentium 133 or higher
32 Mbyte RAM
20 Mbyte hard disk space
Windows 95/98 or NT4 operating system
CD-ROM drive for installing
Second serial port for PLC remote control

MicroSHADOW
RESEARCH