# CLICK Programming Software

## FREE software!

CLICK Programming Software can be downloaded at no charge or a CD version may be purchased.

The CLICK Programming Software is designed to be a user-friendly application, and the tools, layout, and software interaction provide ease-of-use and quick learning.

The simple operation of this software allows users to quickly develop a ladder logic program. The online help file provides information that will help you get acquainted with the software quickly.

## PC requirements

CLICK PLC Windows-based programming software works with Windows® 2000 Service Pack 4, XP Home or Professional, or Vista (32 bit). These are the minimum system requirements:

- Personal Computer with a 333 MHz or higher processor (CPU) clock speed recommended; Intel Pentium/Celeron family or AMD K6/Athlon/Duron family, or compatible processor recommended
- SVGA 800x600 pixels resolution. (1024x768 pixels resolution recommended)
- 150MB free hard-disk space
- 128MB free Ram (512MB recommended)
- CD-ROM or DVD drive for installing software from the CD
- 9-pin serial port or USB port for project transfer to PLC (USB port communications also requires USB-to-serial converter)

## C0-PGMSW

### CLICK PLC Programming Software

Includes software on CD-ROM and software installation manual. Free download available from the Web includes the manual in pdf format. Cable sold separately. Windows 2000/XP(Home/Pro)/Vista required.

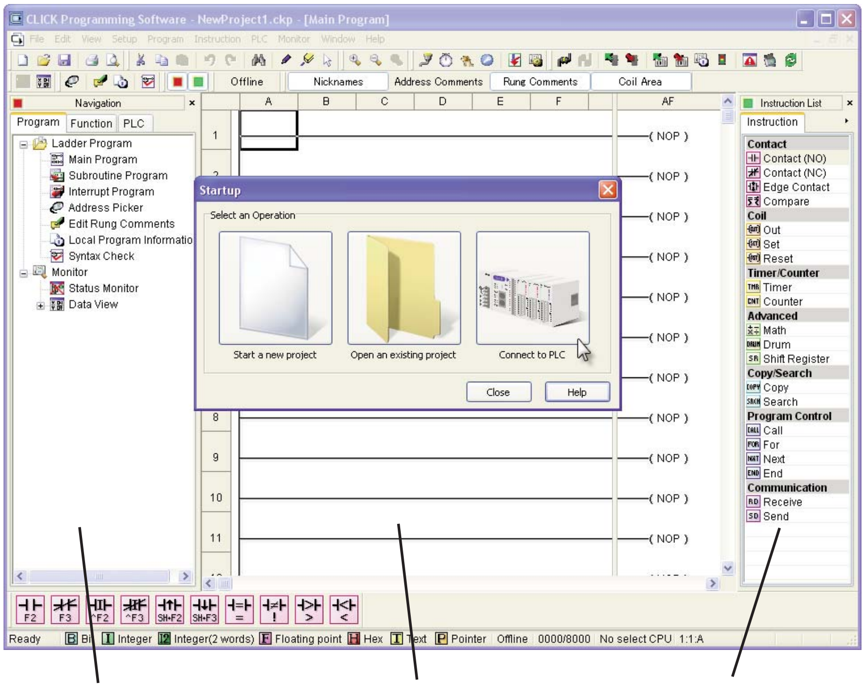
The CLICK Programming Software can be downloaded free at the *AutomationDirect* Web site:

www.support.automationdirect.com/downloads.html

**NOTE: CLICK PLCs cannot be programmed using *Direct*SOFT5 programming software, which is used to program our *Direct*LOGIC PLCs; you must use the CLICK programming software, C0-PGMSW.**

### Main window

The Main Window is displayed when the program opens. It is divided into Menus, Toolbars, and Windows that work together to make project development as simple as possible.



**Navigation Window**

**Ladder Edit Window**

**Instruction List Window**

# CLICK Programming Software

## Instructions

The 21 easy-to-use instructions are listed at the end of this section.

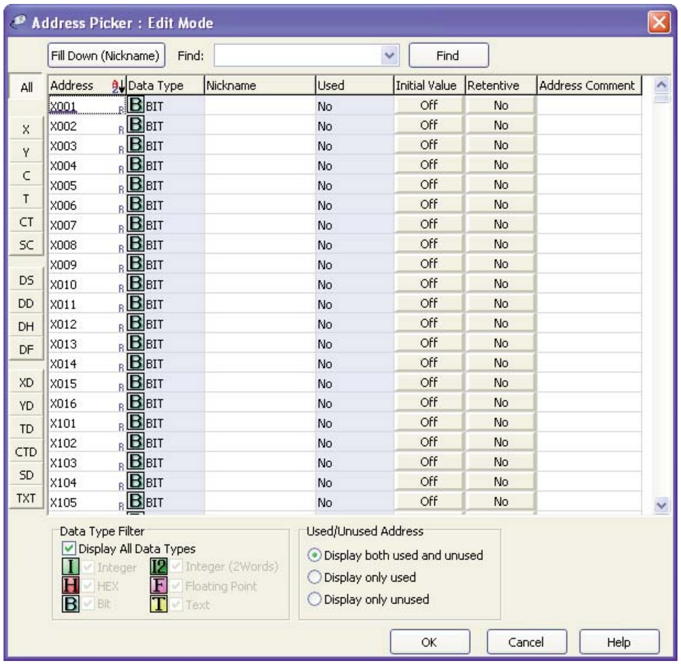
## Powerful features!

CLICK Programming Software has amazingly powerful features for a free software product, such as

- Address picker
- Separate subroutine and interrupt routine features
- Rung comment feature

## Address picker

The Address Picker is a powerful multi-function memory table which can be used to assign nicknames, create address comments, and establish initial values for specific memory locations. It can assign specific memory locations to be retentive during power outages. The Address Picker also has powerful tools for sorting the memory table and making it easier to use.



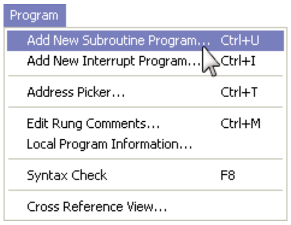
| Address | Data Type | Nickname | Used | Initial Value | Retentive | Address Comment |
|---|---|---|---|---|---|---|
| X001 | BIT | | No | Off | No | |
| X002 | BIT | | No | Off | No | |
| X003 | BIT | | No | Off | No | |
| X004 | BIT | | No | Off | No | |
| X005 | BIT | | No | Off | No | |
| X006 | BIT | | No | Off | No | |
| X007 | BIT | | No | Off | No | |
| X008 | BIT | | No | Off | No | |
| X009 | BIT | | No | Off | No | |
| X010 | BIT | | No | Off | No | |
| X011 | BIT | | No | Off | No | |
| X012 | BIT | | No | Off | No | |
| X013 | BIT | | No | Off | No | |
| X014 | BIT | | No | Off | No | |
| X015 | BIT | | No | Off | No | |
| X016 | BIT | | No | Off | No | |
| X101 | BIT | | No | Off | No | |
| X102 | BIT | | No | Off | No | |
| X103 | BIT | | No | Off | No | |
| X104 | BIT | | No | Off | No | |
| X105 | BIT | | No | Off | No | |

Data Type Filter
☑ Display All Data Types
I ☑ Integer  I2 ☑ Integer (2Words)
H ☑ HEX  F ☑ Floating Point
B ☑ Bit  T ☑ Text

Used/Unused Address
● Display both used and unused
○ Display only used
○ Display only unused

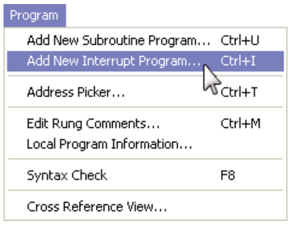
### Subroutine feature

Subroutines can be created to isolate a body of program code that is run selectively.


Program
Add New Subroutine Program... Ctrl+U
Add New Interrupt Program... Ctrl+I
Address Picker... Ctrl+T
Edit Rung Comments... Ctrl+M
Local Program Information...
Syntax Check F8
Cross Reference View...

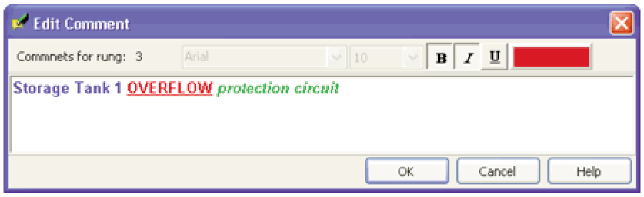
### Interrupt routine feature

Interrupt programs are created and named using the New Interrupt Program dialog, which is accessed from the Program Menu.


Program
Add New Subroutine Program... Ctrl+U
Add New Interrupt Program... Ctrl+I
Address Picker... Ctrl+T
Edit Rung Comments... Ctrl+M
Local Program Information...
Syntax Check F8
Cross Reference View...

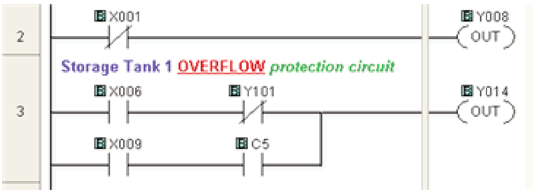
### Rung comment feature

Easily create and edit rung comments with colors and three text styles.



Commnets for rung: 3

Storage Tank 1 OVERFLOW *protection circuit*



# CLICK Instruction Set

## Instructions

### Normally Open Contact

The Normally Open Contact mimics the behavior of a physical contact and changes in response to the status of a Bit Memory Address. The Normally Open Contact is ON when the related bit is ON.

### Normally Closed Contact

The Normally Closed Contact mimics the behavior of a physical contact and changes in response to the status of a Bit Memory Address. The Normally Closed Contact is ON when the related bit is OFF.

### Edge Contact

The Edge Contact turns ON when the related bit transitions from OFF to ON (Rising Edge) or ON to OFF (Falling Edge).

### Compare Contact

The Compare A with B instruction uses a Mathematical Operator as a basis of comparison of two data values. A mathematical statement is developed using the instruction dialog. When the Data A value satisfies the selected mathematical relationship with Data B, the associated Bit Memory Address is turned ON.

### Out Coil

An Out instruction turns ON its associated Bit Memory when the status of the rung is true. The Out instruction turns OFF its associated Bit Memory when the status of the rung is false. An Out instruction is capable of turning on and off more than one Bit Memory Address at the same time.

### Set Coil

The Set instruction turns ON the associated Bit Memory when the status of the rung is true. The Set instruction can turn ON more than one Bit Memory at the same time. When a user selects the Set instruction from the Instruction List pane and drops it on a rung of the Ladder Editor, the Set dialog appears.

### Reset Coil

The Reset instruction shall turn off the associated bit memory when the status of the rung is true. A Reset instruction can turn off more than one bit memory at the same time. When a user selects Reset instruction from instruction list pane and drops on the specific rung of ladder editor, the Reset dialog shall be invoked.

### Timer

An ON Delay Timer measures a user-specified time duration that begins with a transition of the enable rung from OFF to ON. Beyond this transition point, the Timer increases the Current Value until it reaches the Set Point, when the Timer Bit is turned ON.

An OFF Delay Timer measures a user-specified time duration that begins with a transition of the enable rung from ON to OFF. Beyond this transition point, the Timer increases the Current Value reaches the Set Point, when the Timer Bit is turned OFF.

### Counter

When enabled, a Counter instruction counts up or down (depending on user settings) until it reaches the Set Point. The Counter counts in response to the transition from OFF to ON of the enabling rung. If the user selects Count Up & Down, the Counter will have two enabling rungs, one for up counts and one for down counts.

The current count is held in the register shown in the Current Value field. When the Current Value reaches the Set Point, the Completion Bit is turned ON. The Completion Bit is turned OFF when the Reset rung is enabled.

When a user selects a Counter instruction from the Instruction List pane and drags and drops it to the selected rung of the Ladder Editor, the Counter dialog appears.

### Decimal Math Instruction

The Math instruction serves as a powerful calculator to perform mathematical operations during the execution of the Ladder Program. Two sets of mathematical operators are available within the Math instruction. One set is appropriate for use with decimal values, and the other is for use with hexadecimal values.

The formula pad of the  Math instruction dialog is used to develop the mathematical expression that will be solved during the CPU scan.

The mathematical expression can be developed using the onscreen keypad, the keyboard, and Address Picker, combining constants and stored variable values, as necessary for the application. The Sum and Modulo operators are available for either decimal or hexadecimal data values.

When the user selects the radio button for Decimal data values, the available operators include: standard arithmetic and algebraic operators, parentheses for grouping terms, and transcendental functions and operators.

Selecting the radio button for Hexadecimal data values makes a different set of operators available. The logical operators AND, OR, and XOR are available only for Hex math. The bit operations Shift Left and Shift Right (LSH and RSH) and Roll Left and Roll Right (LRO and RRO) are also available only for Hex math.

Parenthetical expressions can be nested up to eight levels deep.

If the Floating Point Data Type is employed in any operation, then all operations will be based on Floating Point math. The Result will be stored in the data format selected for the Result.

## Instructions

### Hex Math Instruction

The Math instruction serves as a powerful calculator to perform mathematical operations during the execution of the Ladder Program. Two sets of mathematical operators are available within the Math instruction. One set is appropriate for use with decimal values, and the other is for use with hexadecimal values.

The formula pad of the  Math instruction dialog is used to develop the mathematical expression that will be solved during the CPU scan.

The mathematical expression can be developed using the onscreen keypad, the keyboard, and Address Picker, combining constants and stored variable values, as necessary for the application. The Sum and Modulo operators are available for either decimal or hexadecimal data values.

When the user selects the radio button for Decimal data values, the available operators include: standard arithmetic and algebraic operators, parentheses for grouping terms, and transcendental functions and operators.

Selecting the radio button for Hexadecimal data values makes a different set of operators available. The logical operators AND, OR, and XOR are available only for Hex math. The bit operations Shift Left and Shift Right (LSH and RSH) and Roll Left and Roll Right (LRO and RRO) are also available only for Hex math.

Parenthetical expressions can be nested up to eight levels deep.

If the Floating Point Data Type is employed in any operation, then all operations will be based on Floating Point math. The Result will be stored in the data format selected for the Result.

### Drum Instruction

The Drum instruction shall be used to simulate an electro-mechanical drums sequencer. It shall work on time base or event base.

### Shift Register

The Shift Register instruction shifts a range of control bits with each OFF-to-ON transition of  the Clock pulse. If the starting address is lower than ending address the shift register will shift from left to right. If the ending address is lower than starting address then shift register will shift from right to left.

### Single Copy

The Single Copy instruction is used to copy a data or text value from its Source location to a specified Destination register. The Source location of the data or text can be another register, identified by its Memory Address, or it can be a constant value typed directly into the Source field on the dialog.

### Block Copy

The Block Copy instruction is used to copy a data or text value from its Source location to a specified sequence of Destination registers. The Source location of the data or text is identified by its Memory Address, and the Destination registers are identified by the first register in the sequence.

A powerful feature is included in the Block Copy instruction allowing you to copy numerical text values to one or more data registers as integer values.

### Fill

The Fill instruction is used to copy a data or text value from its Source location to a specified range of Destination registers. The Source location of the data or text can be another register, identified by its Memory Address, or it can be a constant value typed directly into the Source field on the dialog. Constant data or text values must follow the typing conventions for that Data Type.

### Search Instruction

The Search instruction is used to search for a data value that meets specified conditions and that is located within a specified range of data registers. A successful search returns the Memory Address of the value that satisfies the conditions.

### Call and Return Instructions

The Call instruction is required to call a Subroutine program from the Main Program. The Call instruction resides in the Coil Area of the Ladder Program. A Subroutine program must have a Return instruction to return to the Main Program.

### For and Next Instructions

The For instruction indicates the starting point of a For - Next loop, and based on its user setting, determines how many times the For - Next loop will be executed. Between the For instruction rung and the Next instruction rung, place the rungs of logic that should be repeated multiple times. Placing one For-Next loop inside another For-Next loop is not permitted.

The Next instruction indicates the end of a For loop.

The Next instruction must be the only instruction on its rung. No enabling contacts are allowed on the same rung with the Next instruction.

### End Statement

The  End instruction marks the termination point of the normal program scan.

Every program must have an End statement to tell the CPU that there are no more rungs to be processed. The End statement must reside in the Main Program. It cannot reside in a Subroutine or Interrupt program.

### RD ASCII

The Port ID defaults to Port 2.