Legend Reference for Dinex RTMWin Ladder Logic

Here is an explanation of the symbols that are used to construct each ladder rung. The basic ideas stem from the use of relay based controls. A common way to describe the arrangements of those relays came into use more than 100 years ago. The finished description is known as ‘Ladder Logic’.

The notion of wired-in logic began with simple on-off switch circuits. A vertical line on the left hand side of the diagram represents the power supply voltage wire or ‘buss bar’. A vertical line on the right hand side represents the power supply ground wire or chassis ground connection. A line from the left hand side to the right hand side represents a dead short circuit. This type of line is rarely used. The idea of using wires is to hook up some type of load between the power supply and the ground. A circle is used to represent the load. The load could be a lamp, a solenoid valve, a relay, or a starter solenoid (relay).

A line from the left hand side to a circle, then from the circle to the right hand side, represents an electrical circuit that is always on. Each line segment represents a wire. As long as the power is on, the load item is energized and this represents the most basic form of a ladder rung used for control.

In order to control the load, an engineer would wire a switch contact in between the power connection and the load. Suppose that the load is solenoid valve. When the switch is on then the valve is energized, so the valve opens. Then some compressed air flows through the valve into a pneumatic cylinder and opens the door with a whoosh sound of air. If the switch is turned off then the solenoid valve closes, air leaks away and the door closes. The compressed air connections to the valve and pneumatic cylinder are not shown on the ladder logic diagram! Only the electrical part of the control system is shown here. In a solenoid valve the load is an electro-magnet, so when the electricity is flowing then the resulting magnetism causes the pneumatic part of the valve to shift position. All of those details are lumped into the circle marked, ‘LOAD’.

AND logic. Here is where the logic comes in. In a transit vehicle there is an automated ‘Speed Switch’ that is ON when the vehicle is traveling slowly, less that three miles per hour. There is a requirement that the door cannot be opened if the vehicle is going faster than three MPH. We want to construct a ladder of switch logic because the door should not open unless the ‘Door Open’ switch is on AND the ‘Speed Switch’ is on. A horizontal connection on the ladder rung represents the logical idea of AND.

More switches may be added across the rung to make more complex logic of this AND this AND this…
OR logic. The next main idea in logic is the use of two or more switches to control one item. The idea is that the door can be opened using the ‘Door Open’ switch OR a ‘Door Emergency’ switch. The logical construction of an idea of this OR that comes from using vertical lines on the ladder drawing. Horizontal lines represent the logic idea of AND, while vertical lines represent the logical idea of OR. In this example, the door will open if the LOAD1 is energized by the logical combination of (‘Door Open’ switch OR ‘Door Emergency’ switch) AND ‘Speed Switch’.

```
POWER    ‘DOOR OPEN’ SWITCH     ‘SPEED SWITCH’     LOAD1    GROUND

‘DOOR EMERGENCY’
```

In the example above, the two DOOR switches are manually activated by the bus driver. The SPEED switch is automatically generated by the transmission or ABS system. Ladder logic is a combination of automation and manual switches. Nothing can happen unless the manual switches are being activated by the bus driver. However, there are various safety features which are automatically generated and these features are sometimes called ‘interlock’ or ‘interlocking’. In order to have interlock functions, an idea about the state of LOAD1 being on or off must cause some sort of action on another ladder rung.

Combinatorial logic. The next example shows the same rung as above using ladder contact symbols instead of switch symbols. As these examples are becoming more sophisticated, the rung output LOAD1 is now connected to a relay. The load of a relay is an electromagnet. When the relay is energized, then electricity is flowing and the resulting magnetism causes the internal contacts of the relay to shift position. One set of relay contacts would be wired to the solenoid valve (which is not shown). One set of relay contacts is used as a contact in another rung. In theory there could be a lot of contacts on each relay which could be used throughout the ladders of logic.

```
POWER    ‘DOOR OPEN’ SWITCH     ‘SPEED SWITCH’     LOAD1    GROUND

‘DOOR EMERGENCY’

LOAD1     BRAKE INTERLOCK
```

When the output LOAD1 is on, then the BRAKE INTERLOCK will also turn on.
In the previous example, when the door is open then the braking system is supposed to be activated so as to prevent the vehicle from moving. Using the Dinex system, or any PLC, the state of LOAD1 is kept in the electronic memory as a bit, as a ‘one’ or as a ‘zero’. This allows for unlimited amounts of combinatorial logic. When using relays, the cost is directly proportional to the number of contacts. When the number of contacts is greater than two, then relay logic using actual relays starts to cost a lot of money. In a typical transit bus ladder of logic, there may be a need for ten or fifteen ladder contacts. The Dinex system uses virtual relays and the resulting contacts are connected using simple mouse click items on a PC. The virtual connections don’t cost much time or money. When the entire control system would use ten or more relays then it may be cheaper to use the Dinex system instead of discrete relays.

Latches and Normally Closed Contacts. The next example brings two new concepts. Having a memory can be useful, and in ladder logic the concept of memory is called a latch. In order for the latch to work, some method is needed to unlatch and turn it off. A normally closed contact is used. The normally closed contact of a relay (or virtual relay) can conduct electricity on the ladder when it is off. When it turns on, it becomes open circuit and breaks the flow (or virtual flow) of electricity. In this example the ‘Door Override’ switch also causes an alarm. The alarm will stay on until both the ‘Door Emergency’ switch is off AND the ‘Reset’ switch is turned on.

In the example above the ‘Reset’ contact is normally closed. The reset switch is momentary contact so it will be off unless the bus driver is actively pressing the switch actuator. When the bus driver turns on the ‘Door Emergency’ switch then the Door Alarm output will be turned on. Thus the ‘Door Alarm’ contacts will be on. Then, if the bus driver turns off the ‘Door Emergency’ switch then the Door Alarm output will stay on because of the ‘Door Alarm’ contact. This is a latch. The output will remain latched ON even though the input ‘Door Emergency’ has turned off. There is a current path (or virtual path) from POWER through the ‘Door Alarm’ contact, then through the ‘Reset’ contact, then through the load to GROUND.

Once the Door Alarm output is on and latched on, the only way to turn it off is to depress the ‘Reset’ switch. Then the normally closed ‘Reset’ contact will be open and there will be no current path from POWER to the DOOR ALARM and then to GROUND.

The admittedly contrived examples above are intended to give a general idea of ladder logic functions in hopes of providing a basis toward understanding of ladder logic as shown in the Dinex RTMWin. The RTMWin can show the state of a ladder rung in real time. The PC display will show actual ladder rungs that are controlling various functions on the vehicle. The active contacts are shown with color highlighting while inactive contacts have a normal color background. It is easy to see which contacts are providing ladder outputs at any given time.

Main Ideas: Real physical relays have been replaced by virtual relays because of cost and also because the virtual relay ladder may be up-dated to allow the vehicle to comply with a changing set of law which governs vehicle operations. Various rules that govern operations are being encoded into the ladder of logic. Normally closed contacts and latches make a simple ladder more complicated yet are often necessary for compliance with rules and regulations. Things become more complicated when we add the concept of timers in the ladder rungs.
This page contains a description of the timer items. In the timing diagrams shown, time moves from left to right. Simple timing diagrams are intended to show the relationship between timer inputs and resulting timer outputs. The Dinex system has four types of timers: Flasher, Delay-On, Delay-Off, and Time-on (pulse) timers.

Flash Timer. A flasher turns on and off for the time specified. The flasher continues to turn on and off continuously when it is energized. When first energized, the output turns on immediately.

- **INPUT**
- **TIME**

Delay On Timer. A timer that turns on after a delay for the time specified. When the input is energized, the output remains off during the timing period, then it turns on forever until the input is de-energized. Subtle point: The timer may be set to zero during start-up if that is desired to make the output to be on immediately on start-up, but then give a delay only after it is de-energized then re-energized. Depends on actual ladder rung configuration.

- **INPUT**
- **DELAY**

Delay Off Timer. A timer that stays on during a delay for the time specified. When the input is energized, the output turns on immediately. Later when the input is de-energized, the output will stay on throughout the timing period. Subtle point: The timer may be set to zero during start-up if that is desired to make the output to be off immediately on start-up, but then give a delay only after it is de-energized then re-energized. This type of timer is usually set to zero during start-up.

- **INPUT**
- **DELAY**

TurnOn Timer. A timer that turns on for the time specified, then turns off, but if and only if the input stays true for a time period that is longer than the time specified in this timer. It may be called a pulse timer. If the input is de-energized, then the output turns off immediately. If the input stays on forever then the output will be on for the time specified, then turn off and stay off forever. If the input is de-energized then re-energized then the output will once again turn on for the time specified, then turn off. Subtle point: The timer may be set to zero during start-up if that is desired to make the output to be off immediately on start-up, but then give a pulse time output only after it is de-energized then re-energized.

- **INPUT**
- **TIME ON**
The programming environment consists of ladder logic symbols, representations of the I/O List items that are arranged in the left hand column of the Dinex GUI language display. These symbols are used to populate the ladder rungs that are constructed using the tools and a few mouse clicks. The programmer may select one item in the left hand column, then click in a slot on the ladder logic diagram; the resulting symbols are summarized here. The RTMWin does NOT include this programming environment. The PC display of RTMWin can show one ladder rung at a time. Here is what the ladders look like.

Input Normal On. Dinex input physical connection as contact "Normally Open". This is a physical switch or an input from the Engine Control Unit. Also known as “input contact”.

Input Normal Off. Dinex input physical connection as contact "Normally Closed". This is a physical switch or an input from the Engine Control Unit. Also known as “input contact”.

Output Normal On. Dinex output (physical connection) as contact "Normally Open". The state of an output, whether ON or OFF, will be available to use in the ladder logic, similar to the way that internal flags are used. Also known as “output contact”.

Output Normal Off. Dinex output (physical connection) as contact "Normally Closed". The state of an output, whether ON or OFF, will be available to use in the ladder logic, similar to the way that internal flags are used. Also known as “output contact”.

Flag Normal On. Internal flag as contact "Normally Open". The flag must be defined in the ‘out’ column of another rung, before it can be used as a contact on a ladder rung. Also known as “flag contact”.

Flag Normal Off. Internal flag as contact "Normally Closed". The flag must be defined in the ‘out’ column of another rung, before it can be used as a contact on a ladder rung. Also known as “flag contact”.

An internal flag is similar to an output except that there is no actual wire, only a virtual one.
Advanced programming items appear on this page. Please consult with the I/O Controls technical support department for additional information or help with programming these items, should that become necessary. This next part is a brief overview of the concepts involved with the Code item.

Code. Represent manually written inserted code. The Dinex system uses simple computer code to represent the ladder items that are described above. There are some complex functions which are difficult to program using the ladder symbols alone. Those functions may be incorporated into hidden code, and the symbol may be turned on and off using ladder contacts. An engineer may write the computer code, or reuse a piece of typical code which has been proven to work in the past. The system kernel exists as a series of Code items which provide support for the RTMWin, RF-PALM-PC, or Built-In-Self-Test. A full description of these items is beyond the scope of this text.

The code item may be placed in a ladder rung which will make an output or flag turn on and off. The code will only run when energized, like a timer.

The code item may be placed in a ladder rung by itself or with other contacts, in an input column and also in the out column. This is a stand alone type of code which will be run when energized or in this example, it will run all the time because it will always be energized.

A code item may be placed in a ladder rung by itself with a horizontal short in the output column. It will be put into the resulting compiled Dinex program at the end, in the higher part of program memory. It will never run by itself. This type of code contains a bunch of subroutines. Other code items may call these subroutines. The reason for this special type of code construction is to allow the programmer to use a large piece of code that is more than 256 bytes in length. In this example, the code item MBC01 calls subroutines in the code item MBC02. These two codes work together for purposes of RTMWin support and Built In Self Test.

The code item that gives the results of the Built In Self Test causes a flag to turn on and off a number of times. The result may be shown on a lamp in the dashboard display, using the flag from this rung. A mechanic may count the number of flashes, then consult with system documentation to find out the meaning of the flash codes. In general, one flash means module #64, two flashes indicates a problem with module #65, three flashes indicates module #66, etcetera, and sixteen flashes indicates module #79.

Technical support for the code items is available from I/O Controls Corporation. A series of standardized Dinex Kernel codes have already been developed and tested. These codes may be customized to suit almost any application. The RTMWin interface exists in the MBC02 code.
The Dinex system has some ‘Built In Self Test’ capability using a low current technique. The amber LEDs will show the state of the continuity. When the output (red LED) is OFF then the feedback (amber LED) should be on. If both LEDs are off then this indicates that the output is open or disconnected. If both LEDs are on then this indicates good continuity but there is a blown fuse.

Some of the modules do not have it. The gateway modules do not have the output continuity test. Gateway outputs are low current outputs. The ‘sink’ modules do not have the output continuity test. Sink modules have an ‘S’ in the designation such as T2-DIO-808S. The sink module is a switch to ground when the Dinex output is ON.

The RTMWin or RF-PALM-PC will test the outputs and will show those outputs as being failed because there are no amber LED feedback signals. To fix this problem, set the module type to ‘Sink’ for any DIO-808S and any DIO-168. That way, the RTMWin or RF-PALM-PC will not show a false fail indication.

The DIO-168 does not have the output continuity test. All sixteen inputs are available to be wired to input switches. When programming the DIO-168 or any ‘Sink’ module, it is important to select module type ‘Sink’ in the Dinex GUI language. Right click on the module in the left hand column, select ‘Modify Node’, then make sure that the ‘Source/Sink’ designation is Sink.

For other modules T2-DIO-888 or G2A-DIO-888, the RTMWin or RF-PALM-PC will show that an output has failed, if both the Red LED (Output) and Amber LED (feedback) are both off. The RTMWin or RF-PALM-PC will show that an output fuse has failed, if both the Red LED (Output) and Amber LED (feedback) are both on. There is a way to use the Dinex feedback inputs as contacts in ladder rungs.

Feedback Normal On. Dinex amber (feedback) LED as contact "Normally Open"

Feedback Normal Off. Dinex amber (feedback) LED as contact "Normally Closed"
System Flags. The Dinex GUI language has two ‘System’ flags. These flags have fixed position in the Dinex map. V17(15) is FSCAN flag. The FSCAN flag is false during the first scan through the logic. After that the FSCAN flag will be true. The FSCAN flag may be used in any diagram.

Scan Order. There is no need to write a ladder for the FSCAN flag. It already exists in the Dinex program. The FSCAN flag will be false until the System ladder rungs are solved. For this reason, it is very important that SYSTEM should be the last node in the scan order.

Scan Order: The default scan order is alphabetical order and it also has the SYSTEM in the last position, on the bottom of the list. The scan order can be changed so that certain ladder rungs are solved earlier in the scan, for purposes of synchronization such as synchronizing the flasher lamps. If the scan order is changed then be sure to put the SYSTEM at the bottom of the list.

In this example the flasher timer is in A1 (the MBC), and then the J1939 gateway items are solved. The J1939 gateway items are called GI and GO. The GI and GO have been moved to the top of the list in order to maximize signal throughput and minimize the time lag between Dinex inputs and outputs as compared with J1939 inputs and outputs. A key point here is that SYSTEM is at the bottom of the Node Scan Order list.

On start-up, the FSCAN flag is false. The MBC (or master HCNC) will read each module’s inputs. There are sixteen bits in the MBC V2 register and each bit represents one Dinex module, bits 0 – 15 represent Dinex modules #64 - #79. If any module fails to respond to the MBC then a bit will be zero in the V2 register. If all the modules are responding OK then the V2 register will contain all ones.

The MBC solves the logic of all of the ladder rungs depending on the list of modules in the ‘Scan Order’ shown on the Project screen. This will set the state of each flag, including ‘outputs used as contacts’, but it will not actually turn on any Dinex outputs during the first scan. This can remove one possible way to have a glitch or pulse on a Dinex output during start-up. After going through all the ladders, then the flags and ‘outputs used as contacts’ will be correctly set as ON or OFF depending on the actual state of the Dinex inputs at the moment of start-up. After this initial scan the FSCAN flag will be true.
The MBC will read all the inputs and set the bit(s) for communication OK or FAIL in V2, then the MBC will solve the logic of all of the ladder rungs in the order as shown on the list of modules in the ‘Scan Order’ pop-up list shown on the Project screen. It will actually turn the Dinex outputs on or off as required by the solution of each rung. The MBC will continue with this routine, reading all the inputs then solving all the ladder rungs. Then reading all the inputs then solving all the ladder rungs, forever.

The other system flag is the FOK flag, in fixed position V17(14). It is used by the ‘Code MBC02’ in terms of RTMWin support and RF-PALM-PC support. If the FOK flag is false, then the RTMWin or RF-PALM-PC will not be able to take control from the MBC.

System Flag on the System Diagram: Force Points OK Flag. The Dinex GUI language has two ‘System’ flags. These flags have fixed position in the Dinex map. V17(14) is FOK flag. The name of this flag means ‘Force points is OK to do if desired’. The Dinex system may be connected to a higher-level system such as a Laptop PC “RTMWin” or a Palm Pilot “RF-PALM-PC”. The higher level system does have the capability to turn the various Dinex outputs on or off. This feature may be useful during troubleshooting or demonstrations. The FOK flag controls the idea of whether the MBC may submit to external control, or whether those external controls will be locked out. If the FOK flag is false, then the RTMWin or RF-PALM-PC will not be able to take control from the MBC.

If the FOK flag is TRUE, then the RTMWin or RF-PALM-PC will indeed be able to take control during ‘Output’ force points mode which allows the mechanic to turn the Dinex outputs on and off from a PC or from a Palm Pilot. The FOK rung is a safety rung. In this example, the Master Switch #1 must be OFF in order to use the Force Points Mode. If an operator turns the Master Switch #1 to the ON position then the ‘Code MBC02’ will kick the MBC out of force points mode, and the MBC will once again solve the ladders of logic. Force Points Mode may be useful during troubleshooting although there are no safety features during Force Points Mode, so the mechanic must be careful not to energize a ‘Door Open’ output and a ‘Door Closed’ output at the same time.

More input contacts may be added to this FOK rung as required for safety purposes.