Phenix Programmer v3

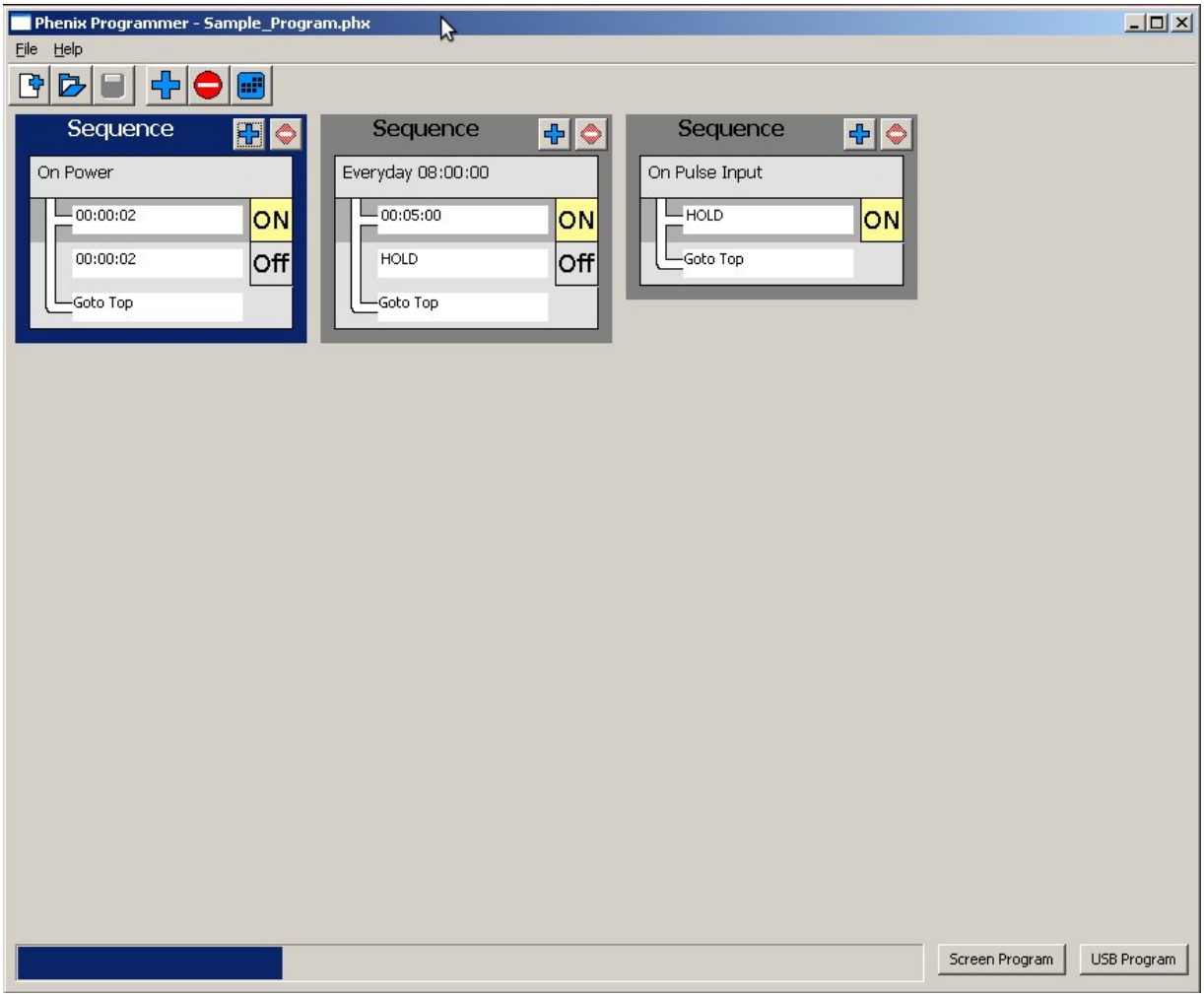
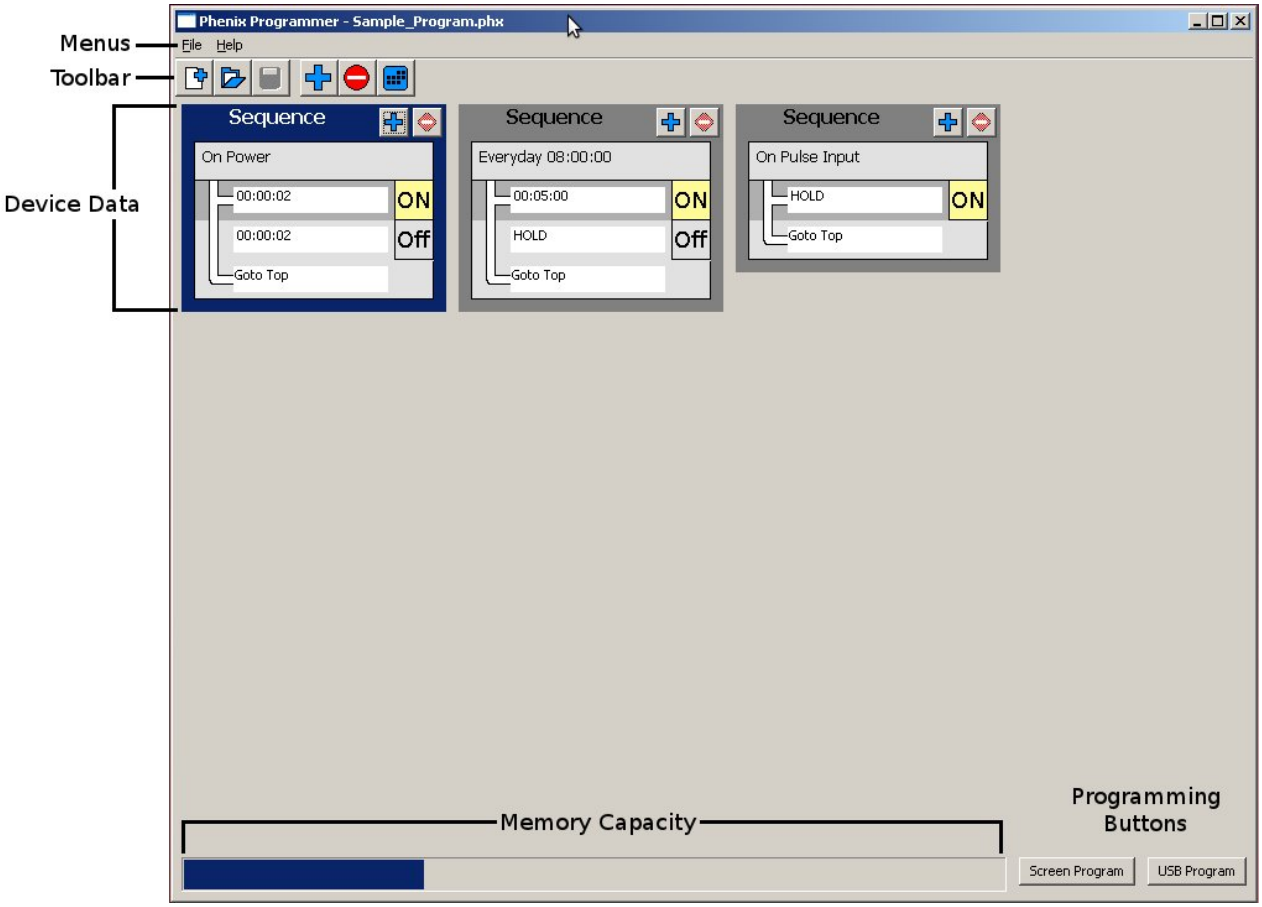


Table of Contents

Program Layout



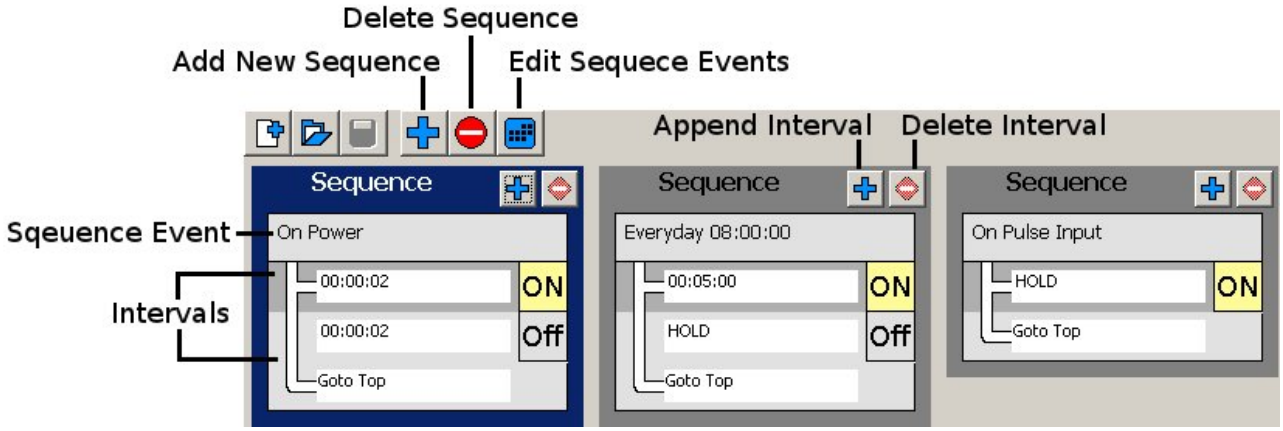
Program Layout

The Phenix Programmer software provides a graphical interface to enter settings for Phenix devices. From the *File* menu, create new, open, or save device data. The *Help* menu displays information about the Phenix Programmer software, such as the software version. The *toolbar* provides buttons for creating new, opening, and saving device data as well as function buttons for manipulating device data. The *toolbar* will change depending on the device.

The memory capacity bar indicates how much memory will be required for the current device data and how much space is available in the device. The memory bar starts on the left side and fills toward the right. The Programmer software will allow programs to exceed the available memory, but will display an error when attempting to program a remote. A smaller program will take less time to transfer into remotes and devices.

Once all the desired data has been entered, the programming buttons will transfer the device settings to a remote programmer through either the screen programming or USB interface.

Phenix Timer Devices



Sample Timer Program

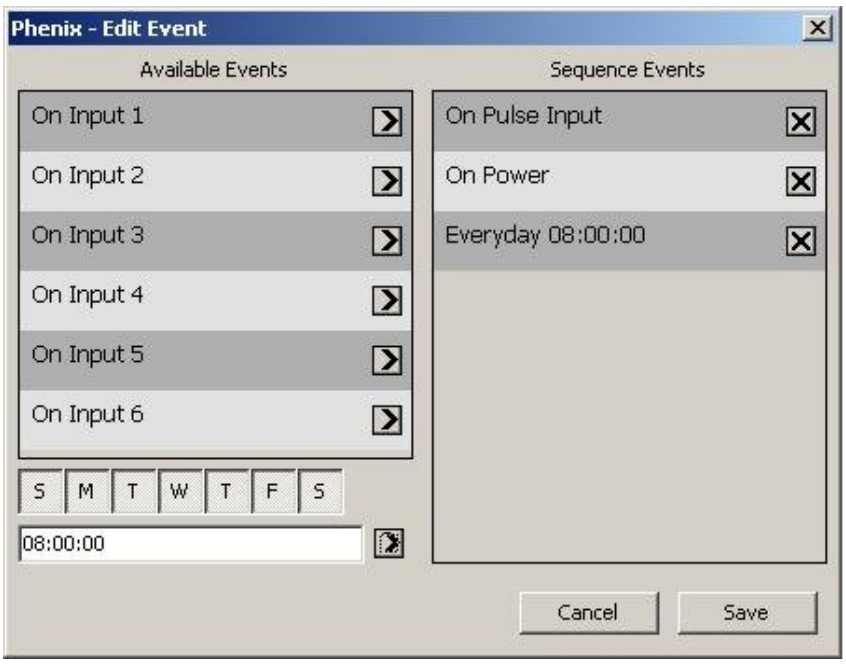
Phenix timer devices separate timer data into sequences. A sequence is a list of intervals that are activated when an event occurs, such as when the timer powers on or from a time of day. Each sequence consists of an event section followed by intervals. There may be more than one event that can trigger an interval sequence. When an event occurs, the first interval in the series is activated. The output will be set to the output of the interval and a countdown of the interval will begin. When the interval has expired, the next interval in the sequence will be loaded and the same process is repeated. After the last interval has been executed, the timer will automatically repeat the sequence.

Timer Intervals

Sample Intervals

An interval consists of time, output(s) state, and flow columns. The time column is a text entry box that defines the amount of time the output(s) will be in a state. Time is represented in DD HH:MM:SS.FF format, days, hours, minutes, seconds, and fractions of seconds. The Programmer software will automatically convert text into the time format. A time of 00:00:00 is a zero time and represents that the outputs(s) will be held in output(s) state indefinitely, or until an event. A time of zero will be displayed as “HOLD” in the time field. The output(s) column represents the state the output(s) will be in for the duration of the time. ON states will be highlighted in yellow. Left clicking on the output will alternate the state. The flow column is used to distinguish which intervals will react to an event. For further information on flow, see Timer Flow Example. Left clicking on the flow column will alternate the flow of the interval.

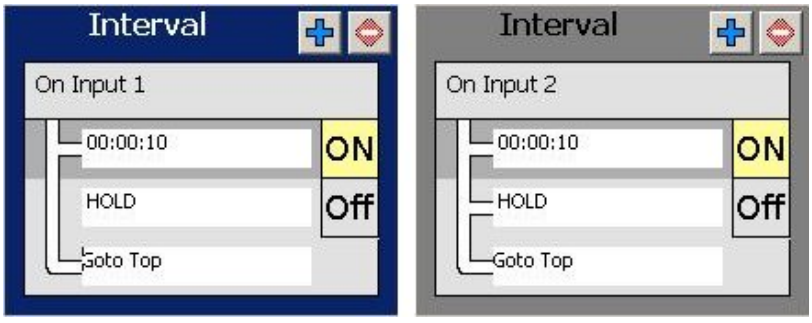
Timer Events



Timer Event Sample

When adding a new sequence, an Event dialog will be displayed. The event dialog contains a list of available events for the current device. The list of available events differs for each device. Clicking on the arrow will add the event to the list of events that will activate the sequence. The “x” button next to sequence events will remove the event from the sequence and add it back to the list of available events. Clock events are added with the arrow located in the lower left. Clock events are in the format HH:MM:SS and can be set for each day of the week.

Timer Flow Example



Sequence Flow Example

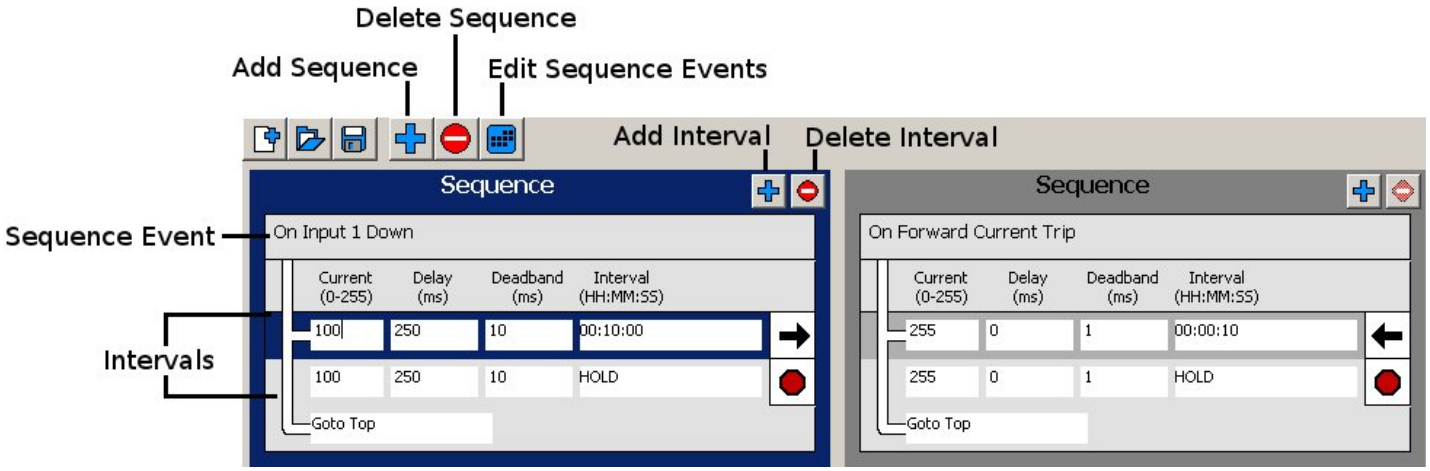
The sequence flow is a specialized feature of Phenix timers and most applications are unaffected by these settings. The white bar down the left hand side of the sequence indicates the flow of the sequence. The selected sequence(left) has an

event for *input 1*. When the timer device receives an *input 1* event, the output will be turned ON and a 10 second countdown will begin. After 10 seconds, the output will be turned OFF and hold at this location until another event occurs. An *input 1* event will turn the output ON and start the 10 second countdown once again. If an *input 1* occurs during the 10 second countdown, the sequence will advance to the next interval with a flow line set. In this example, the countdown will be reset to 10 seconds and the output ON.

When an *input 2* event occurs, the unselected sequence(right) will start. The output will be set and the first interval, 10 seconds, will begin to countdown. If an input 2 event occurs during the countdown, the sequence will advance to next flow location, in this case, the second interval. The flow line provides logic control when events occur. In the sequence flow example, *input 1* will reset the 10 second countdown whenever the event occurs, however, an *input 2* will alternate between 10 seconds ON and hold OFF.

If an event occurs for a sequence that is not currently running, the first interval of the sequence will be loaded. Clock events will always load the first interval and are not effected by the sequence flow.

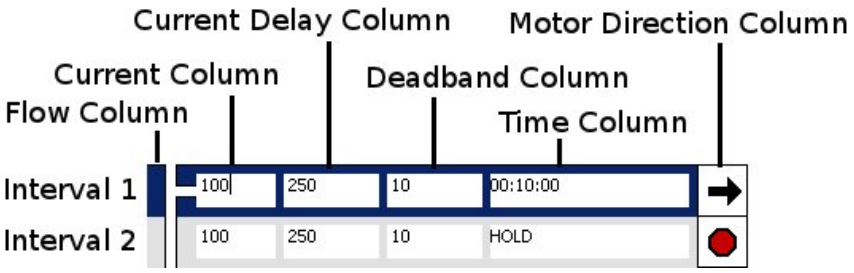
Phenix Motor Devices



Motor Sample Program

Phenix motor devices separate motor control data into sequences. A sequence is a series of intervals and motor settings that are activated when an event occurs, such as when the motor controller powers on, when the motor reaches a load threshold, or when a clock event occurs. Each sequence consists of an event section followed by intervals and motor settings. There may be more than one event that can trigger a sequence. When an event occurs, the first interval in the series is activated. The motor device will start counting down the interval. After the deadband period has expired, the motor will start running in forward or reverse direction, or remain off. After the current delay time has expired the motor device will check the motor current. If the motor exceeds the current setting, the motor will be stopped but the interval will continue to countdown. Once the interval has expired, the motor will be stopped and the next interval in the sequence will be loaded and the same process is repeated. After the last interval has been executed, the timer will automatically repeat the sequence.

Motor Intervals and Settings



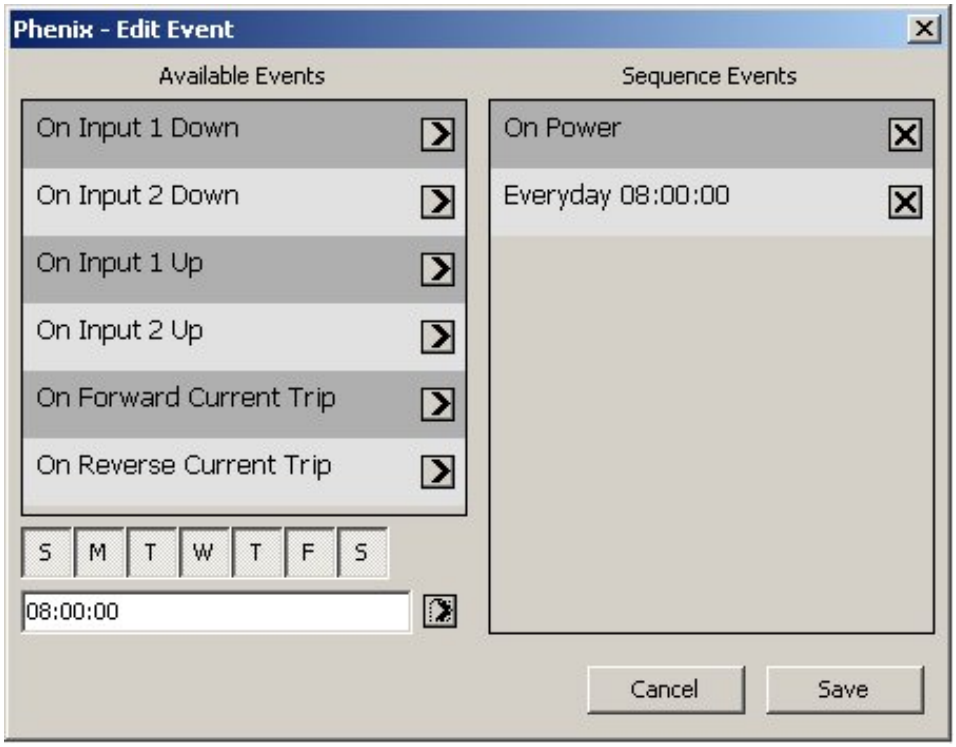
An motor interval consists of the following columns:

Column	Description
Flow	Intervals with disconnected flow will be activated when the previous interval has expired but skipped when an event occurs. This setting only applies when receiving an event while already executing a sequence with that event.
Current	The current threshold or maximum amount of motor load allowed before stopping the motor. The current is a value between 0 – 255. A value of 255 will deactivate current sensing. The value is the current limit for this interval. When the motor current exceeds the current value, the motor will stop running. The entire interval duration will still be executed. Current limits can be used to trigger an event sequence.
Current Delay	The amount of time, in milliseconds, to allow the motor to run before testing the current limit. The value may be between 0 and 4095. This setting allows the motor to come up to running speed before checking the current. The current is much higher when the motor first starts in a direction and even higher when switching from one direction to the reverse.
Deadband	The amount of time, in milliseconds, to delay before starting the motor. The value may be between 1 and 4095. The deadband allows for a period of time to discharge the energy in the motor before running in the opposite direction and helps prevent part damage.
Time	The amount of time to run the motor forward, reverse, or idle. The time is in the format HH:MM:SS.
Direction	Direction the motor will travel. A right arrow represents forward, a left arrow represents reverse, and a octagon represents idle.

The time column is a text entry box that defines the amount of time the motor will run forward, reverse, or remain idle. Time is represented in DD HH:MM:SS.FF format: days, hours, minutes, seconds, and fractions of seconds. The Programmer software will automatically convert text into the time format. A time of 00:00:00 is a zero time and represents that the motor will be held in current state indefinitely, or until an event occurs. A time of zero will be

displayed as “HOLD” in the time field. The motor direction column defines the direction the motor will drive or be idle for the duration of the time. A right arrow represents forward, a left arrow represents reverse, and a red octagon represents idle. Clicking on the motor direction column will alternate between the different motor states.

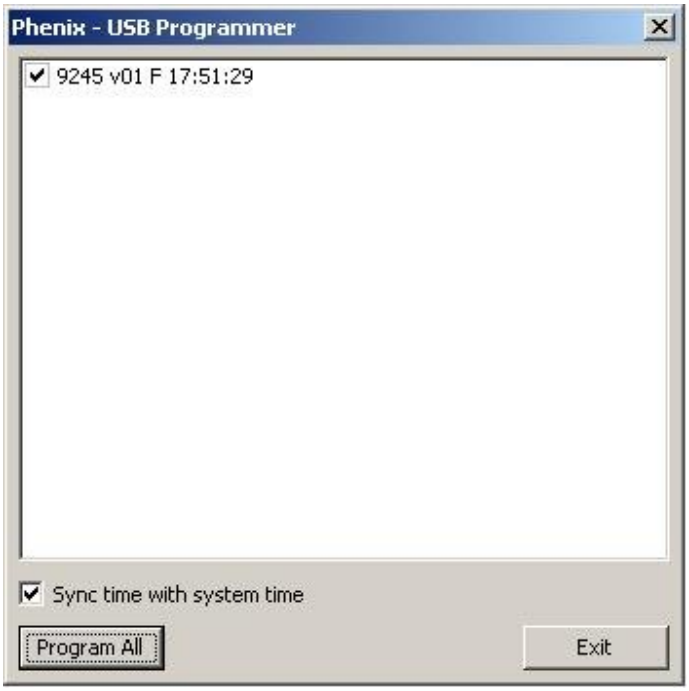
Motor Events



Motor Event Sample

When adding a new sequence, an Event dialog will be displayed. The event dialog contains a list of available events for the current device. The list of available events differs for each device. Clicking on the arrow will add the event to the list of events that will activate the sequence. The “x” button next to sequence events will remove the event from the sequence and add it back to the list of available events. Clock events are added with the arrow located in the lower left. Clock events are in the format HH:MM:SS and can be set for each day of the week.

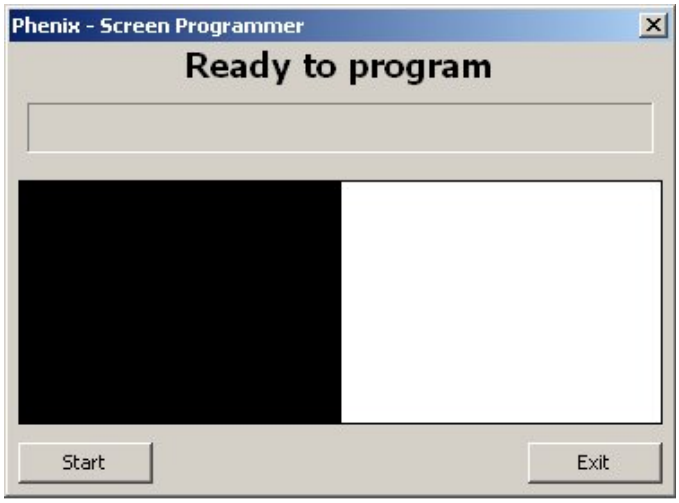
USB Programming



After the device behavior has been setup, the values can be transferred into a Phenix device through the USB interface. The Phenix Programmer software will automatically scan for available USB devices and display the device model, version number, and time, if applicable. Some devices contain a clock and the current time can be transmitted by checking the “Sync time with system time” box. The device will display the current time if the time was programmed. Setting the time on remote programmers will then send the current time to all the devices programmed by the remote.

To program a Phenix USB device, click the check box left of the device to program a single device or click the “Program All” to send the settings to all available devices. A check box to the left of the device will display if the device has been programmed with the current settings.

Screen Programming



After the device behavior has been setup, the values can be transferred into a Phenix device. This is accomplished using the screen programmer interface and an infrared remote. The Phenix Programmer software will blink two boxes on screen. An infrared remote can then read the values from the blinking boxes. The transfer can be stopped at anytime and restarted if needed. Once the remote has successfully received the values, the remote can then transmit the values into any compatible timer device. To send the values into an infrared remote, please use the following procedure:

1. Press the button on remote. The red LED on the remote will light solid then blink rapidly. The remote will indicate readiness to program by blinking the LED momentarily every three seconds.
2. Hold the remote up to the screen. Position the remote in the center of the black/white box region. The center infrared transmitter on the remote should be on the line where the black and white boxes meet.
3. Press the start button on the programming screen. After a three second countdown, the black and white boxes will alternate rapidly. After a few black/white swaps, the LED on the remote will turn on solid.
4. Hold the remote steady in the center of the black/white boxes throughout the duration. The progress bar at the top of the screen will move from left to right as the program is transferred. At the end of the transfer, the remote will indicate a successful transfer by blinking the remote LED rapidly.

If any of the following occur, the transfer to the remote was NOT successful:

- The LED on the remote does not turn on solid after a few transitions
- The LED on the remote turns off before the progress bar is full
- The LED on the remote does not blink rapidly when the progress bar is full