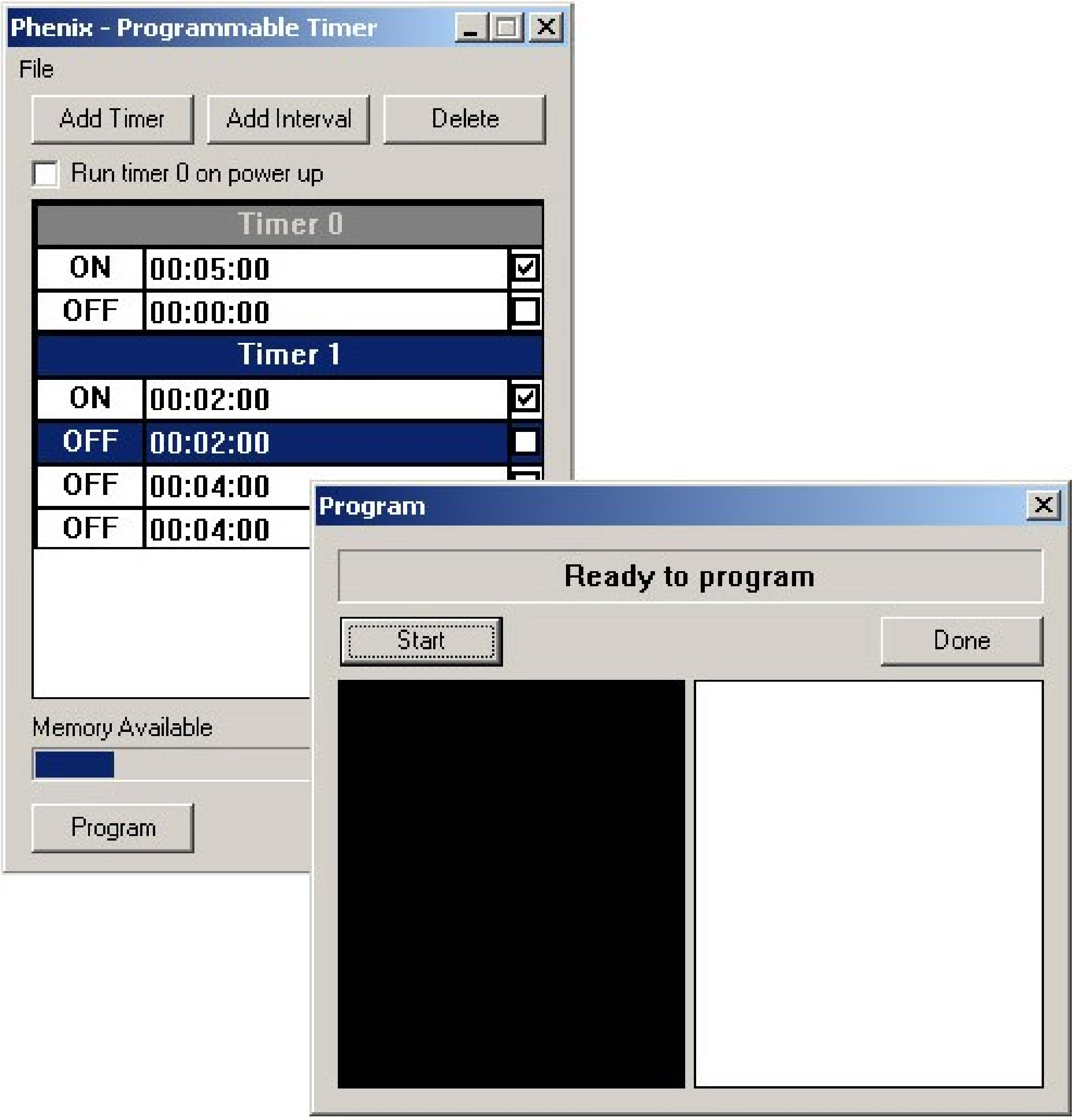
PHENIX

Windows Programmable Timer Application
Version 1.0

Phenix Programmable Timer v1.1 Manual

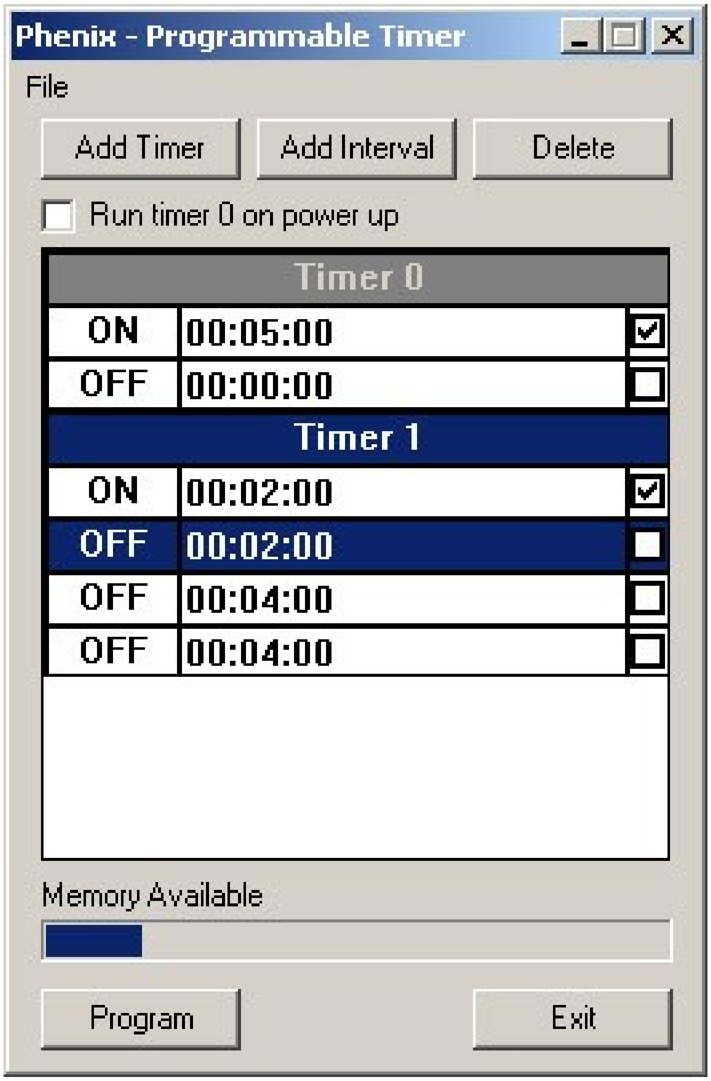


Timer 0		
ON	00:05:00	☑
OFF	00:00:00	□
Timer 1		
ON	00:02:00	☑
OFF	00:02:00	□
OFF	00:04:00	
OFF	00:04:00	

Ready to program

Table of Contents

Software Layout



□ Run timer 0 on power up

Timer 0		
ON	00:05:00	☑
OFF	00:00:00	□
Timer 1		
ON	00:02:00	☑
OFF	00:02:00	□
OFF	00:04:00	□
OFF	00:04:00	□

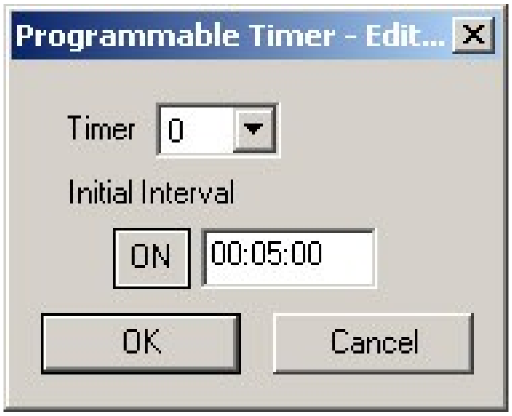
Software Functions

File Menu

The file menu contains the following functions:

- Open – Opens an existing timer program
- Save – Save the current program using the previously defined file name
- Save As – Save the current program and select the file name and folder location
- Close – Clear the current program from the display
- Exit – Exit the program

Add Timer



Timer 0

Initial Interval

ON 00:05:00

Add Timer Dialog

Activates the Add Timer Dialog. Select the desired timer from the pull-down list. The starting interval for the selected timer may be entered. The OK button adds the timer to the current program. The CANCEL button exits the Add Timer Dialog.

Selecting Timers and Intervals

To select a timer or interval, click the left mouse button over the desired timer or interval within the program window. Selecting an interval within a timer will also select that timer. The current selection is displayed by highlighting the Timer or Interval.

Add Interval

A new interval is added to the selected Timer. The new Interval is highlighted.

Delete

Delete the selected Timer or Interval. If only the Timer is selected, all intervals will be deleted.

Timer 0 Power Up Option

Toggle Timer 0 Power Up Option by left clicking on the check box or text. Setting this option will allow the programed device to execute Timer 0 when first receiving power.

Program

Activates the Screen Programming Dialog. The data from the current program will be transmitted. For more information, see Using the On-Screen Programmer section.

Program Design

The flexibility of the Phenix Programmable Timer fills the needs of a wide range of applications. Therefore, it is important to understand the features of the software to ensure that your program fits the desired application.

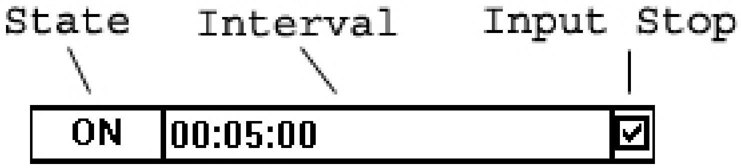
Selecting Timers

Selecting a timer is the first step when creating a program. The number of different timers needed is dependent on the number of inputs connected to the hardware. For applications with a single or no inputs, only 1 timer is required. Applications with two or more inputs require multiple timers. Timers are activated through the coded infrared input terminal. For example, when a code 3 is sent to the input, the corresponding Timer 3 is activated. Additional hardware is required to send this coded input.

Most applications will only require a single timer. Timer 0 is designed as a special case timer. A programmable option is available to activate Timer 0 when the device first receives power. Additionally, Timer 0 is activated when an infrared pulse is sent to the device input. This provides use for a single input without additional input coding hardware.

Timers contain a single or multiple intervals. When the last interval in a timer has been executed, the timer will automatically loop back to the first interval.

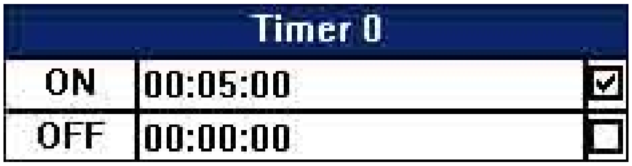
Defining Intervals



Intervals consist of three parts: state, interval, and input stop. The interval state can be toggled between ON and OFF. The ON state closes the relay contacts and OFF opens them. The interval time is represented using three, two digit numbers in the format HH:MM:SS (hours, minutes, and seconds). The maximum range of an interval is 36 hours. The smallest resolution is 1/60th of a second (see Addressing Intervals Below One Second). Larger ranges may be achieved using a sequence of the same state. An interval of 00:00:00 defines an infinite time and will continue running using the current state until receiving an input.

The interval Input Stop check box defines which intervals are considered when an input is received. Starting from the current interval upon receiving an input, the device will move through the current timer until reaching an input stop. If no Input Stops are found, the device will begin at the start of the timer. If a code is received for a different timer, the device will move to the first time of the input code. For example while executing an interval in Timer 2, a code 4 will move the device to the first interval in Timer 4.

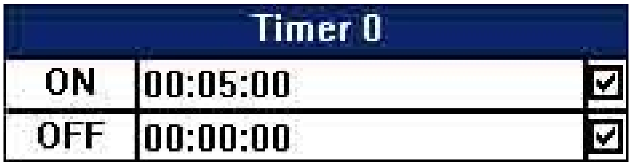
Changing the value of the Input Stop check box can greatly affect the behavior of the device. Consider the following program:



Timer 0		
ON	00:05:00	☑
OFF	00:00:00	□

Input stop example 1

During execution of the first interval (ON – 00:05:00) when a code 0 is received, the device will restart the 5 minute interval. When the first interval has completed the 5 minute countdown, the device will change the state to OFF and remain in that state until receiving an input. Only after receiving a code 0 during execution of the second interval, the first interval will begin its 5 minute countdown. Using the stop input check box, consider an alternate program:

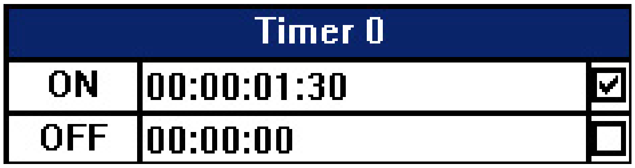


Timer 0		
ON	00:05:00	☑
OFF	00:00:00	☑

Input stop example 2

During execution of the first interval (ON – 00:05:00), an input code 0 will start the second interval (OFF – infinite). The device will remain in the OFF state until receiving another code 0. Upon receiving code 0, the first timer will begin its 5 minute countdown.

Addressing Intervals Below One Second



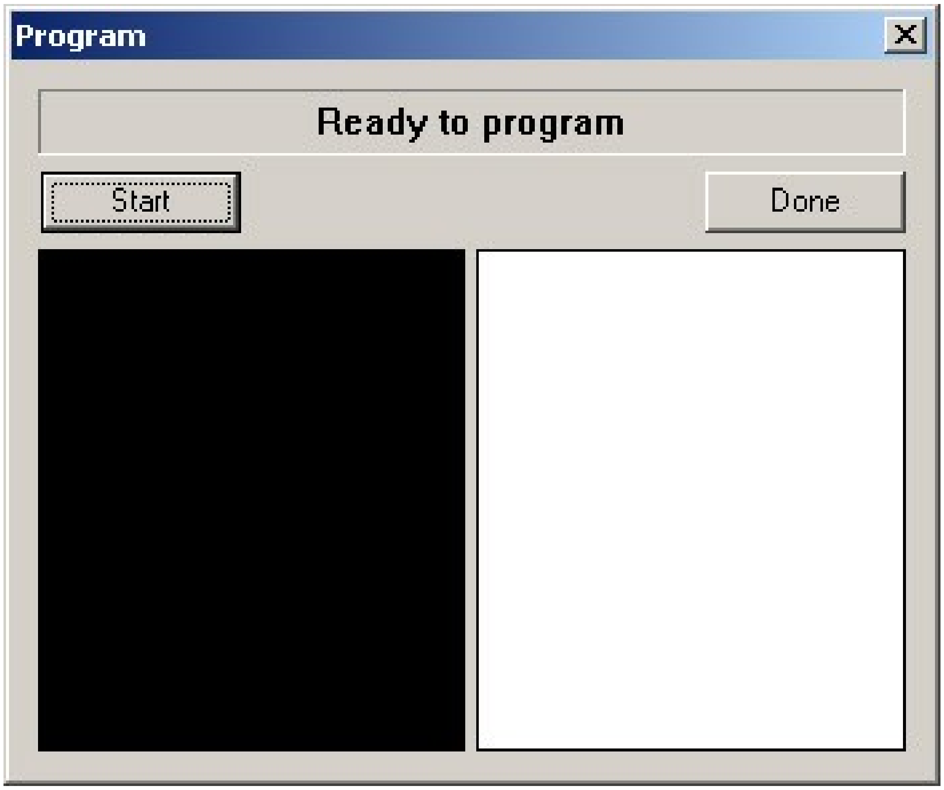
Timer 0		
ON	00:00:01:30	☑
OFF	00:00:00	□

Fractional second example

Below one second resolution is accessed by a fourth 2 digit number at the end of a HH:MM:SS interval. The fourth digit represents 1/60th of a second. In the fractional second example, the interval 00:00:01:30 defines one and a half seconds. Fractional second addressing is optional.

Fractional second addressing is available in version 1.1 of the Phenix Programmable Timer software. Although all timer hardware may be programmed with intervals below one second, not all hardware can respond within that time. Timers which use mechanical relays, such as 9206 and 9621, can not open and close the relay fast enough to guarantee times below one second. However, solid state relay timers, such as 9845 and 9791, are accurate within 1/60th of a second.

Using the On-Screen Programmer



Ready to program

Programming Dialog

To transfer the current program to the remote, press the Program button to activate the programming dialog. Click the Start button to begin transmission. Press and hold the button on the remote. A three second delay is provided to bring the remote to the screen. Hold the remote about an inch from the screen; keeping the center of the remote between the two boxes. The progress bar will fill as the program is being transferred. If the LED on the remote turns OFF or begins to blink before the progress bar has been filled, the transfer failed. When the progress bar has completely filled, the remote will blink the LED indicating a successful transfer.

The remote detects visible light emitted from the screen. Many factors, such as ambient room light and monitor brightness, can affect the transmission. The following guidelines will help ensure a successful transmission:

- Hold the remote close to the screen without hitting it (avoid damaging the screen)
- Keep the remote centered between the blinking black and white boxes
- Keep the remote steady throughout the entire transmission
- Have the remote ready before starting the transmission to avoid missing the start