Lab 2: Timers, Counters, and One-Shot Instructions

# Worksheet

## Timers

1. Provide a screenshot of your on-delay timer implementation.

A screenshot of a computer

Description automatically generated

1. Comment on your observations of how the on-delay timer operates.

The on-delay timer is off for the preset amount of time and then turns on.

1. Provide a screenshot of your off-delay timer implementation.

A screenshot of a computer

Description automatically generated

1. Comment on your observations of how the off-delay timer operates.

The off-delay timer DN output will energize immediately when the rung is true, and then will wait some specified preset time after the rung is de-energized for the DN bit to de-energize as well.

1. Provide a screenshot of your retentive on-delay timer implementation.

A white and blue line

Description automatically generated

1. Comment on your observations of how the retentive on-delay timer operates.

The retentive on-delay timer will remember how much time it has counted even when the rung becomes false. The timer is only reset when the reset command is activated.

1. Provide a screenshot of your “blinker” program.

A blue line on a white background

Description automatically generated

## Challenge: Traffic Light Intersection

1. Provide a screenshot of your program. Also, upload your code to the Dropbox (as an RSLogix project).

A computer screen shot of a diagram

Description automatically generatedA screenshot of a white board

Description automatically generatedA screenshot of a computer

Description automatically generatedA white background with blue text

Description automatically generated

1. What type of traffic pattern/orientation would require that the lights on one road be controlled separately depending on which direction the traffic flows?

Supposing that the roads in question are fairly busy, there could be a crosswalk or a left turn signal that gives priority with an arrow and blocks the opposing traffic with a red signal. In those cases, there would need to be more individual control over the lights. Alternatively, on a narrow road there may only be room for one direction of traffic to pass, so either the east, west, north, or south direction individually.

## Counters

1. Provide a screenshot of your up-counter implementation.

A blue rectangle with white background

Description automatically generated

1. Comment on your observations of how the up-counter operates.

The up counter increases by 1 for every time the increment input bit is switched on. It ignores each time the bit is switched off or left in one state continuously.

1. Provide a screenshot of your down-counter implementation.

A screenshot of a white board

Description automatically generated

1. Comment on your observations of how the down-counter operates.

The down-counter increments the accumulated value down by one every time the rung is true and resets to 0 when the reset command is active.

1. Provide a screenshot of your up/down counter implementation.

A blue lines on a white background

Description automatically generated

1. Comment on your observations of how the up/down counter operates.

The increment input switch will count up for every rising edge and the decrement input switch will count down for every rising edge. The reset command resets to counter to 0. The updown setup is configured by addressing an up counter and a down counter to the same counter file, and so when the preset is reached or exceeded, the DN bit for both counters is active.

## Challenge: Parking Garage

1. Provide a screenshot of your program. Also, upload your code to the Dropbox (as an RSLogix project). A screenshot of a notebook

   Description automatically generatedA blue lines on a white background

   Description automatically generatedA white rectangle with blue lines

   Description automatically generated

## One-Shot Instructions

1. Screenshot your demonstrations and provide them in the worksheet.

Demonstrated throughout the lab in different exercises

## Challenge: Efficient Motor Forward and Reverse Control

1. Provide a screenshot of your program. Also, upload your code to the Dropbox (as an RSLogix project).

A diagram of a program

Description automatically generatedA person holding a rectangle

Description automatically generated with medium confidenceA blue line on a white background

Description automatically generatedA blue rectangle with a white background

Description automatically generatedA blue rectangle with a white background

Description automatically generated