State Machine - MachineLogic

Contents

Introduction

Introducing the **State Machine** command to MachineLogic. You can now add an instruction that simplifies your program to a set number of states that you have defined. Based on the current state and a given input, the state machine performs state transitions and produces outputs. A simple state machine and the MachineLogic equivalent is shown in *Figure 1* and *Figure 2* below.

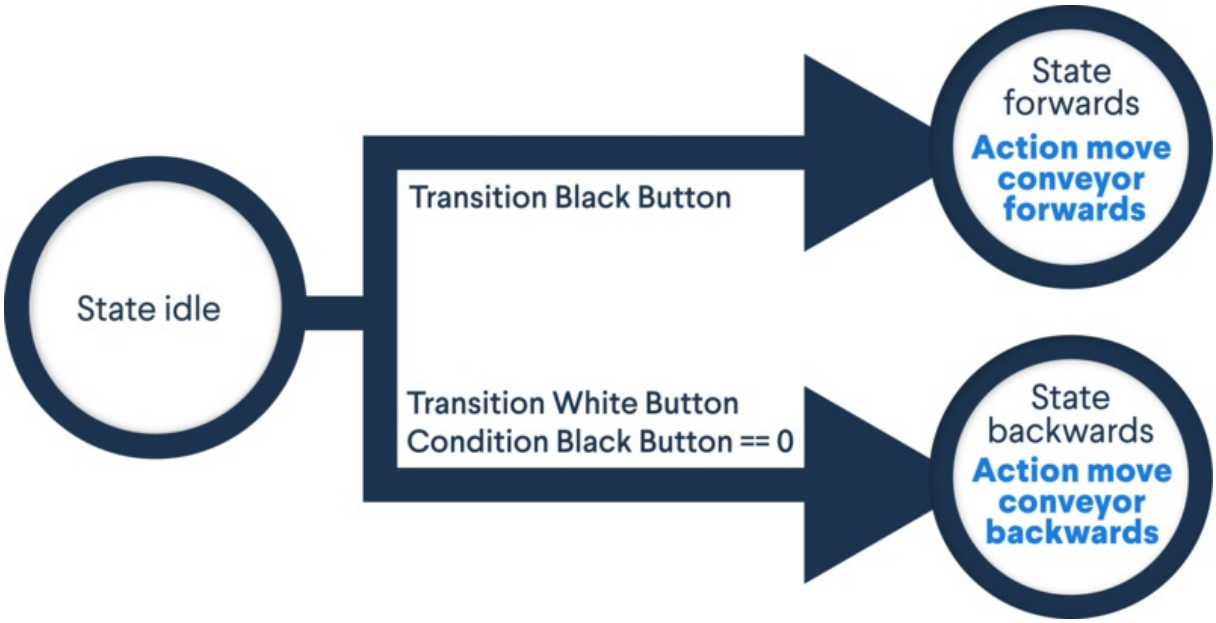


Figure 1: Simple State Machine

Figure 2: Simple State Machine in MachineLogic

Take a deeper dive into MachineLogic State Machines below.

Definitions

State

A State is a **description of the status of a machine**. A machine can only be in exactly one State at any given time, and is always waiting to transition to a possible next state, following the transitions defined in the list of Next States. A state can optionally define an Action to be executed when entering that state. See *Figure 3* below for more details.

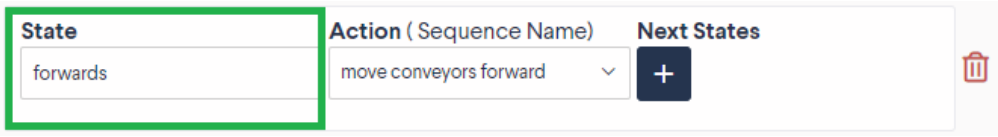


Figure 3: State Field

Action

An Action is a **Child Sequence to be executed** when entering a given state. Please note that although only one State can be active at any given time, several Actions could be executing, as the State Machine does not wait for the Action to complete in order to transition to the Next State. A child sequence can be selected from the Action dropdown, as seen in *Figure 4*.

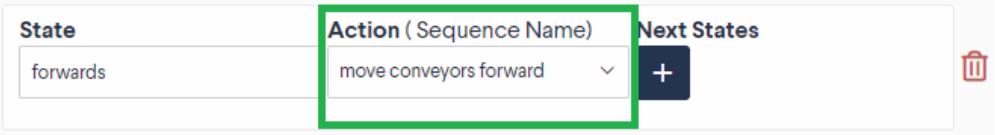


Figure 4: Action Dropdown

Next State

Next State lists **one or many allowable states** the current state can transition to. Each of these next states transition defines an event to wait for, along with an optional condition to be met, in order to trigger the transition to a next state. The modal seen below in *Figure 5* defines a next state and the transition criteria.

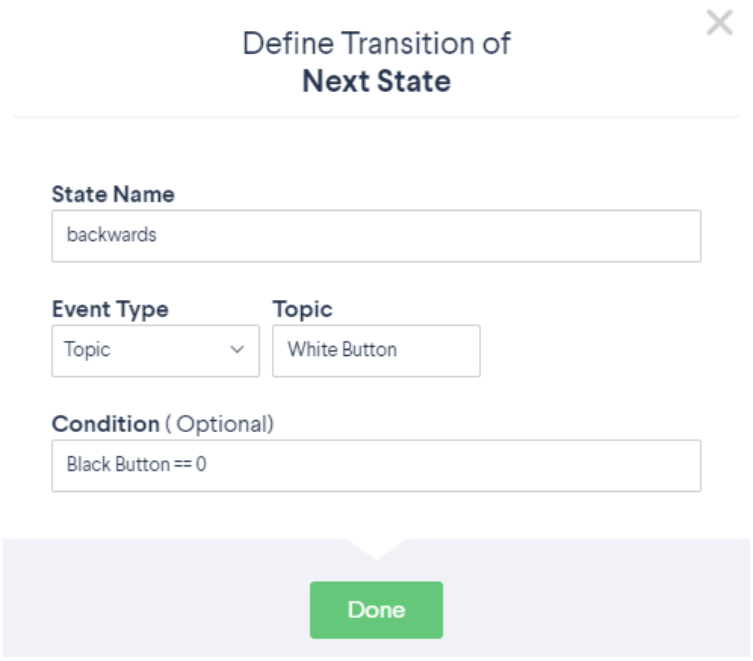


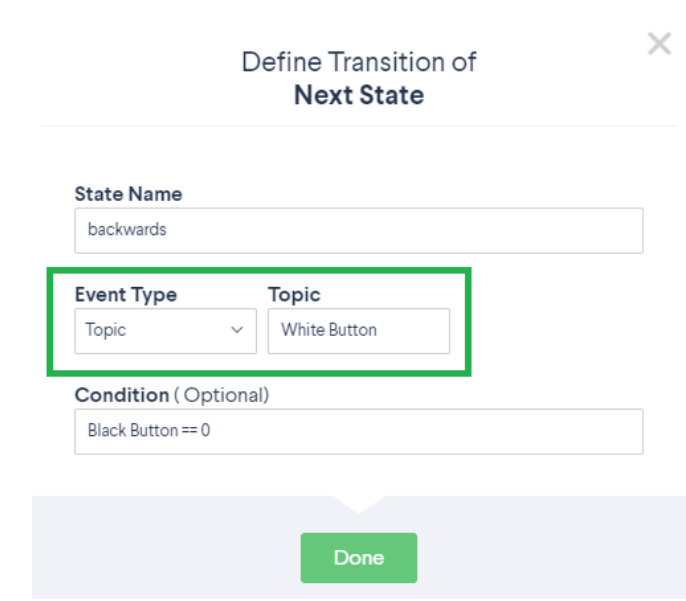
Figure 5: Next State Modal

Transition Event Type

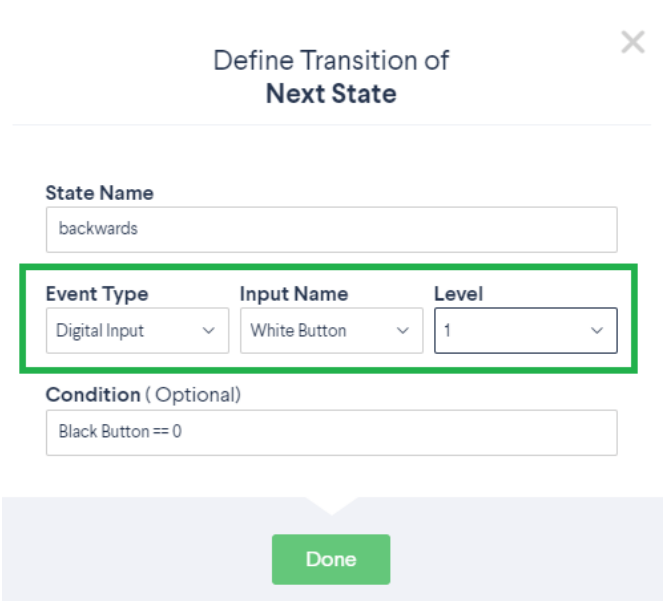
The event that triggers a transition to the Next State can take two forms:

- a **Topic**, which waits for an event to be generated with the exact event topic defined in the Topic field. Events can be generated, for example via the Generate Event command, or a UI Builder button widget. The topic names must match, and any message on this topic will trigger the transition. See *Figure 6a* for more details.
- a **Digital Input**, which waits for a specific digital input to have the defined level in order to transition. See *Figure 6b* for more details.

Topic



Digital Input



Condition

Also known as a guard, a Condition is an **expression that must be met in order to allow the transition** to the Next State. The Condition field can accept a general expression using variables, arithmetic, and/or Lambda functions. If a condition is not met, the transition will not occur and the state machine will remain in the same state. A condition can be inputted in the Condition field, as seen below in *Figure 7*.

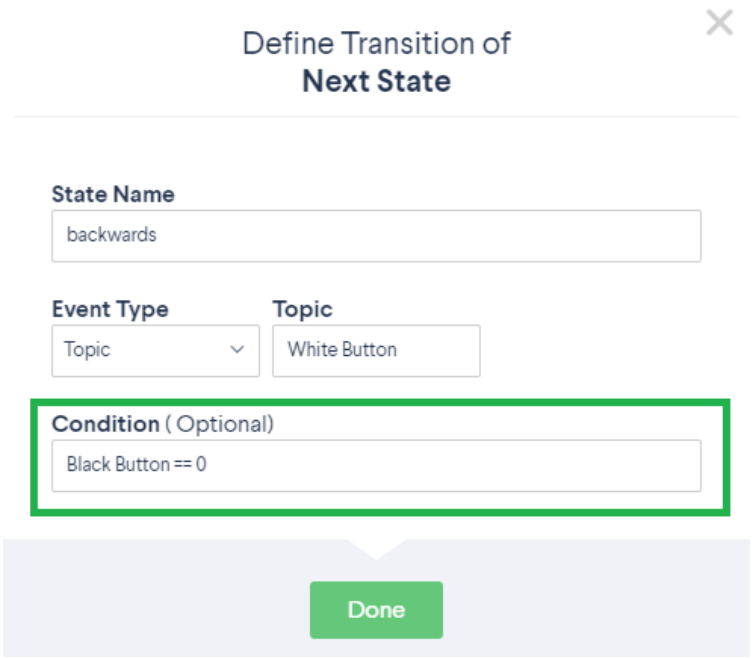


Figure 7: Condition