

Torch In More Detail

Current Source

$$\begin{aligned} Poly &\in Cat \\ Interface &= Iy^A \\ Interface &\in Poly \end{aligned}$$

$$\begin{aligned} StateSystem &= Sy^S \\ StateSystem &\in Poly \end{aligned}$$

$$\begin{aligned} \Phi &= StateSystem \rightarrow Interface \\ &= Sy^S \rightarrow Iy^A \end{aligned}$$

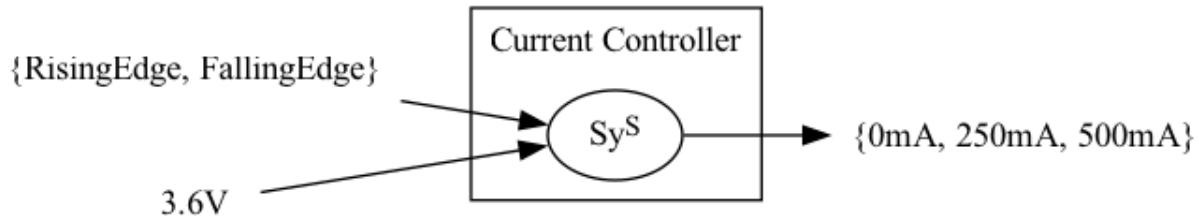
$$\begin{aligned} V &= \{3.6V\} \\ A &= V \times \{RisingEdge, FallingEdge\} \\ I &= \{0mA, 250mA, 500mA\} \\ S &= \{Off, Low, High\} \end{aligned}$$

$$\begin{aligned} \Phi^\# &: S \times A \rightarrow S \\ \Phi_0^\# &= (s, a) \rightarrow Off \\ \Phi_1 &: S \rightarrow I \end{aligned}$$

$$\begin{aligned} \Phi_1(Off) &= 0mA \\ \Phi_1(Low) &= 250mA \\ \Phi_1(High) &= 500mA \end{aligned}$$

$$\begin{aligned} \Phi^\#(Off, 3.6V, FallingEdge) &= Low \\ \Phi^\#(Low, 3.6V, FallingEdge) &= High \\ \Phi^\#(High, 3.6V, FallingEdge) &= Off \\ \Phi^\#(Off, 3.6V, RisingEdge) &= Off \\ \Phi^\#(Low, 3.6V, RisingEdge) &= Low \\ \Phi^\#(High, 3.6V, RisingEdge) &= High \end{aligned}$$

Diagrams of the State System



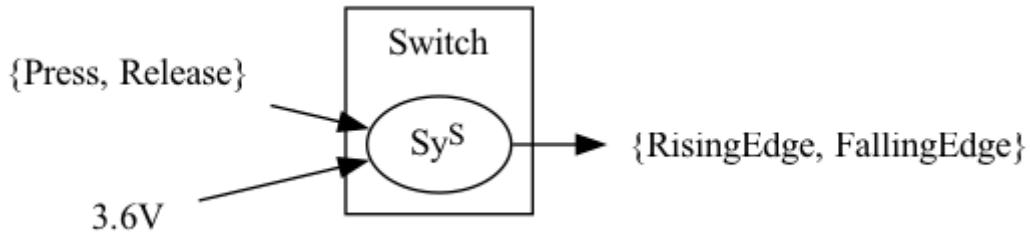
The Switch

$$\begin{aligned}
\Phi &= Ty^T \rightarrow Jy^B \\
T &= \{\text{Conducting}, \text{NotConducting}\} \\
V &= C \\
B &: V \times \{\text{Press}, \text{Release}\} \\
B &= \{\{3.6V, \text{Press}\}, \{3.6V, \text{Release}\}\} \\
J &= \{\text{RisingEdge}, \text{FallingEdge}\}
\end{aligned}$$

$$\begin{aligned}
\Phi^\# : T \times B \rightarrow T \\
\Phi_0^\# = (t, b) \rightarrow \text{NotConducting} \\
\Phi_1 : T \rightarrow J
\end{aligned}$$

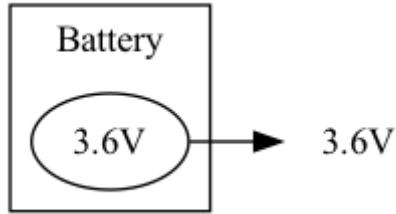
$$\begin{aligned}
\Phi_1(\text{Conducting}) &= \text{RisingEdge} \\
\Phi_1(\text{NotConducting}) &= \text{FallingEdge}
\end{aligned}$$

$$\begin{aligned}
\Phi^\#(-, 3.6V, \text{Press}) &= \text{Conducting} \\
\Phi^\#(-, 3.6V, \text{Release}) &= \text{NotConducting}
\end{aligned}$$



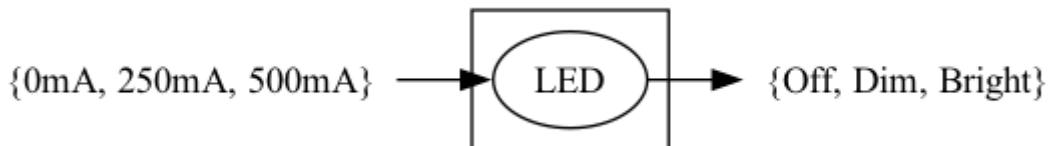
The Battery

$$C = 3.6V$$

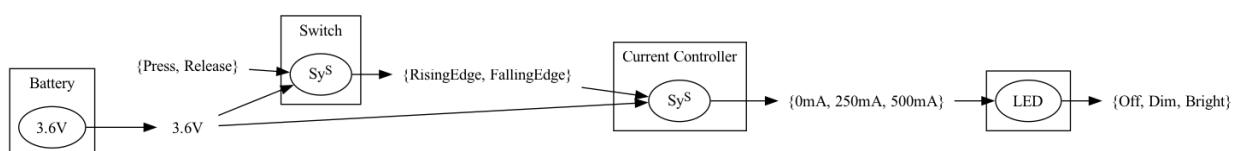


LED

$$\begin{aligned}
L &= \{\text{Off}, \text{Dim}, \text{Bright}\} \\
f : I \rightarrow L \\
f(0mA) &= \text{Off} \\
f(250mA) &= \text{Dim} \\
f(500mA) &= \text{Bright}
\end{aligned}$$



Combined System



System as Combined by Gemini

Based on the mathematical definitions provided, you are describing a discrete dynamical system composed of two state machines wired in series, culminating in a physical output.

Here is the logical reconstruction of the total system.

Ambiguities and Assumptions

Before defining the combined system, I have resolved the following ambiguities using standard systems theory assumptions:

1. **Naming:** You used the symbol Φ for both the "Button" system and the "Controller" system. In the output, I have renamed them Φ_{btn} and Φ_{ctrl} respectively.
 2. **Timing/Composition:** There is an ambiguity regarding *when* the Controller receives the signal from the Button.
 - *Assumption:* I have modeled this as a synchronous composition where the **new state** of the button determines the input to the controller.
 - *Reasoning:* If the controller read the *old state* of the button, the system would react to a button press before the button state actually updated.
 3. **Voltage Injection:** The Controller input A requires a voltage V , but the Button output J only provides an "Edge".
 - *Assumption:* The constant $C = 3.6V$ is applied structurally to bridge the connection between the Button output and Controller input.
-

Combined System Definition

The combined system, Ψ , represents a **single-button dimmable lighting controller**. The system changes light intensity (Off \rightarrow Low \rightarrow High) specifically upon the **release** of the button (Falling Edge).

Let Ψ be the Composite System in *Cat*
 $\Psi : (T \times S)y^{(T \times S)} \rightarrow Ly^B$

1. Composite State and Interface Spaces

Inputs (User) : $B = (3.6V, Press), (3.6V, Release)$
States (Internal) : $\Sigma = T \times S$
 $\Sigma = Conducting, NotConducting \times Off, Low, High$
Outputs (Physical) : $L = Off, Dim, Bright$

2. Internal Composition Logic

We define the projection of signals between the two sub-systems:

Let $\gamma : J \rightarrow A$ be the wiring map
 $\gamma(j) = (3.6V, j)$

3. Composite Functions

The system Ψ is defined by a readout function Ψ_1 and an update function $\Psi^\#$.

The Readout Function (Ψ_1):

This maps the internal state to the physical light brightness.

$$\begin{aligned}\Psi_1 : \Sigma &\rightarrow L \\ \Psi_1(t, s) &= f(\Phi_{ctrl,1}(s))\end{aligned}$$

Explicitly:

$$\begin{aligned}\Psi_1(-, Off) &= Off \\ \Psi_1(-, Low) &= Dim \\ \Psi_1(-, High) &= Bright\end{aligned}$$

The Update Function ($\Psi^\#$):

This maps the current combined state and user input to the next combined state.

$$\begin{aligned}\Psi^\# : \Sigma \times B &\rightarrow \Sigma \\ \Psi^\#((t, s), b) &= (t_{next}, s_{next})\end{aligned}$$

Where the state transitions are derived sequentially:

$$\begin{aligned}t_{next} &= \Phi_{btn}^\#(t, b) \\ s_{next} &= \Phi_{ctrl}^\#(s, \gamma(\Phi_{btn,1}(t_{next})))\end{aligned}$$

4. System Behavior Table (Truth Table)

Evaluating $\Psi^\#$ for the specific interactions defined in your input:

Interaction 1: Pressing the Button

$$\begin{aligned}&\text{Input } b = (3.6V, Press) \\ &\text{Resulting } t_{next} = Conducting \\ &\text{Intermediate Signal } j = RisingEdge \\ &\text{Controller Reaction } s_{next} = s \quad (\text{Since } \Phi_{ctrl}^\#(-, RisingEdge) \text{ holds state}) \\ &\implies \Psi^\#((-s), Press) = (Conducting, s) \quad (\text{No Light Change})\end{aligned}$$

Interaction 2: Releasing the Button

$$\begin{aligned}&\text{Input } b = (3.6V, Release) \\ &\text{Resulting } t_{next} = NotConducting \\ &\text{Intermediate Signal } j = FallingEdge \\ &\text{Controller Reaction } s_{next} = \text{Next State in Cycle } (Off \rightarrow Low \rightarrow High) \\ &\implies \Psi^\#((-Off), Release) = (NotConducting, Low) \\ &\implies \Psi^\#((-Low), Release) = (NotConducting, High) \\ &\implies \Psi^\#((-High), Release) = (NotConducting, Off)\end{aligned}$$

Next Step

Would you like me to generate a **diagram** of this system (e.g., a state transition diagram or a block diagram) to visualize the wiring between the button and the controller?