Blinky

By: Ben Earle and Kyle Bjornson

# Description

This project was designed to be used as a ‘hello world’ for Real-Time (RT) Cadmium. It is for new users to walk through compiling a project, simulating it, then flashing it to the board. This will identify if there are any installation and setup problems with their RT-Cadmium. When flashed to a board blinky will turn the on-board LED on and off. When the on-board button is pressed the light will flash twice as quickly. If the user sees this behavior, then they have successfully setup RT-Cadmium on their computer and flashed a project to the board. This model also shows how to create a DigitalInput and DigitalOuput atomic model and couple them with a user made model.

# Block Diagram



# Formal Models

ATOMIC: Blinky

This model will use its internal transitions to time the flashing of the light. Its input will be used to choose the fast or slow time advance value.

**State Variables:**  
**sigma** = , **phase** = Passive;

**LightOn** = false  bool; /\* Is the LED on, used to toggle state\*/

**FastToggle** = false  bool; /\*Represents the buttons state \*/

**Formal specification:**

**X** = { in  bool } /\* Used to receive input from the button’s digital input model \*/

**Y** = { dataOut  bool } /\* Used to output light state to the LED’s digital output model \*/

**S** = { { phase, sigma, LightOn, FastToggle } }

**ext** (s= { phase, sigma, LightOn, FastToggle }, e, x = { in })) {

case x

in:

FastToggle = x;

}

**int** (s= { phase, sigma, LightOn, FastToggle }) {

case phase

busy:

LightOn = !LightOn;

passive: /\* Never happens \*/

}

****s) { send LightOn to the port dataOut }

COUPLED: Top

This is the top model, notice that it does not have any inputs or outputs. All embedded projects should not have I/O in their top models. This is because GPI/O is done inside atomic models, such as the DigitalInput and DigitalOutput models.

X = {};

Y = {};

D = {button1  DigitalInput, blinky  Blinky, led1  DigitalOutput };

EIC = {};

EOC = {};

IC = { ( button1.out, blinky.in ), ( blinky.dataOut, led1.in ) };

SELECT = ( { button1, blinky , led1 } ) = led1;

( { blinky , button1 } ) = button1;