Security Model Description

Lecture 63, v04

This lecture presents a description of an example Security System model adapted from a model that was once available through MathWorks

As of 2025, the original MathWorks version of the model is no longer available

This version of the Security System model presented here has been simplified (reduced down to one type of intrusion sensor) and modified (providing enhanced management of broadcast events) for use in an educational setting

© Copyright 2022-2025 John G. Artus

Purpose

- This lecture describes a coordinated set of Simulink and Stateflow models that describe:
  - A simplified version of a Home Security System (HSS) model originally developed by MathWorks
  - This model is henceforth called the As-Is HSS Model
- The description of this model is conducted in two sections
  - A description of the As-Is HSS Simulink (environment) model (Section 1)
  - A description of the As-Is HSS Stateflow (system) model (Section 2)
- Some of the material contained herein to describe the MathWorks model is sourced from various pages on www.mathworks.com
- The copyright applied to this document refers to the arrangement of material, both from MathWorks and originally generated by J.G. Artus, into a format that is appropriate to covering the subject in a 2- to 3-week educational module

© Copyright 2022-2025 John G. Artus

Section 1

A description of the Simulink portion of the Simplified Home Security System (HSS) Model

The original MathWorks HSS Model has been “simplified” to focus only on the door sensor

© Copyright 2022-2025 John G. Artus

As-Is Home Security System Model Overview

- This model represents a home alarm system that has a single intrusion-detection sensor
  - When the system detects an intrusion, it sounds an audible “Sound” (called an “Alert”)
  - After a period of 60 seconds, a “call_police” signal is issued (called an “Alarm”)
- This model shows how to
  - Send Simulink (environment model) input events to Stateflow (system model) to simulate triggering of a system sensor
  - Broadcast local events within Stateflow to coordinate between parallel states
  - Output Stateflow (system) events to drive external blocks in Simulink (environment)
- The Simulink model represents the environment within which the security system is operationg
  - It consists of one anti-intrusion sensor attached to a door of the premises
  - The simulation operator has control over the door sensor to simulate tripping of the sensor by an intruder
- The Stateflow model represents the logic of the security system
  - It consists of two parallel states: 1) the door anti-intrusion sensor monitoring state, and 2) the alert state that sounds the audio alert and issues a call to the police
  - In each time step, the parallel states are evaluated in sequence as indicated by the numbers in the top right corners of the states
- Simulink inputs sent to the Stateflow model include
  - For the door sensor, an intrusion detection signal
- Outputs from the Stateflow model include
  - A signal to sound an audible intrusion warning
  - A signal to call the police

As-Is and To-Be are descriptors commonly used in Systems Engineering to represent 1) the original system configuration (As-Is) as it stands today, and 2) the configuration of the system that is anticipated to exist (To-Be) after the planned modifications to the As-Is configuration are completed

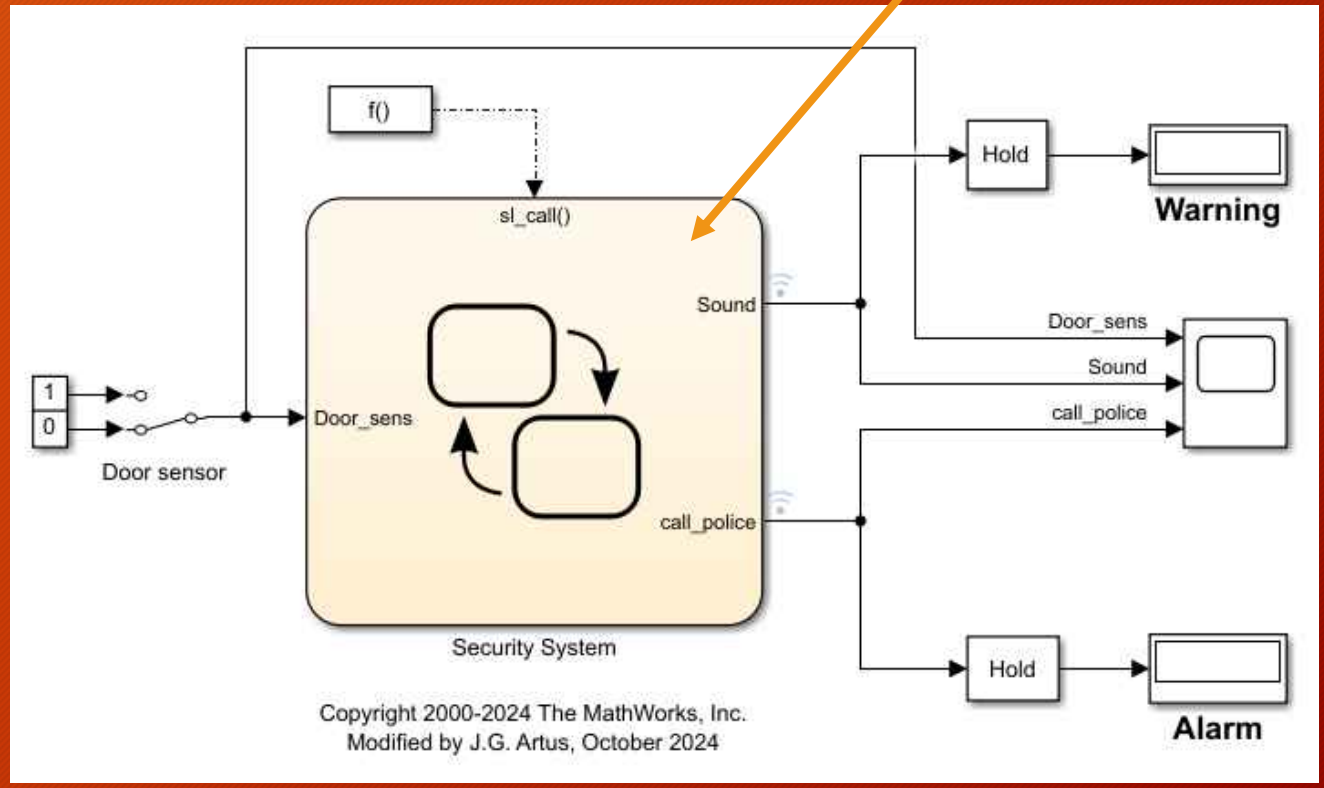
© Copyright 2022-2025 John G. Artus

HSS Environmental Model

- This model represents a simple Home Security System (HSS)
- Shown here is the Simulink portion of the model that represents the environmental simulation within which the Stateflow system model (the Chart Block) operates
- The Simulink model includes a “Chart Block” that represents the Stateflow model which is considered to be the behavioral model of the “System of Interest”
- The Stateflow model contains the state-based logic that controls the transitions of states within the modeled Security System control unit, based on the Simulink environmental model inputs – it describes the behavioral logic of the system that is the subject of study (of the System of Interest)

NOTE: This model is a much simplified version of the original MathWorks model

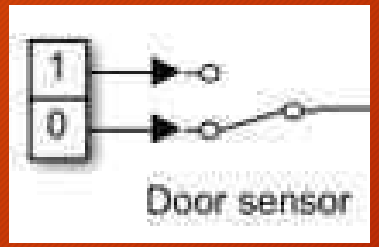
Chart Block



© Copyright 2022-2025 John G. Artus

Switch Inputs

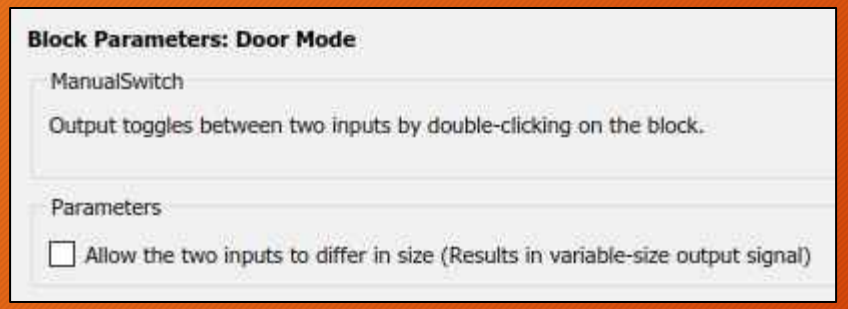
- One manual switch is provided in the Simulink model
- This switch represents the current sense of the Door sensor
  - When the switch is set to 0, this indicates that the door sensor has not been tripped (no intrusion is detected)
  - When the switch is set to 1, this indicates that the door sensor has been tripped (intrusion has been detected)
- This switch can be changed by the simulation operator prior to running of the simulation so as to set the initial conditions of the run
- This switch can be changed by the simulation operator during running of the simulation to indicate a live event occurring during the run
  - This is accomplished by double-clicking on the switch)



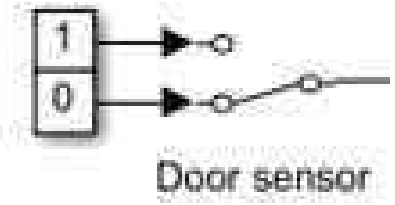
© Copyright 2022-2025 John G. Artus

Switch Arrangement

- This switch is a simple Simulink ManualSwitch Block that flips between Boolean values 0 and 1
  - Example Manual Switch for Door Mode



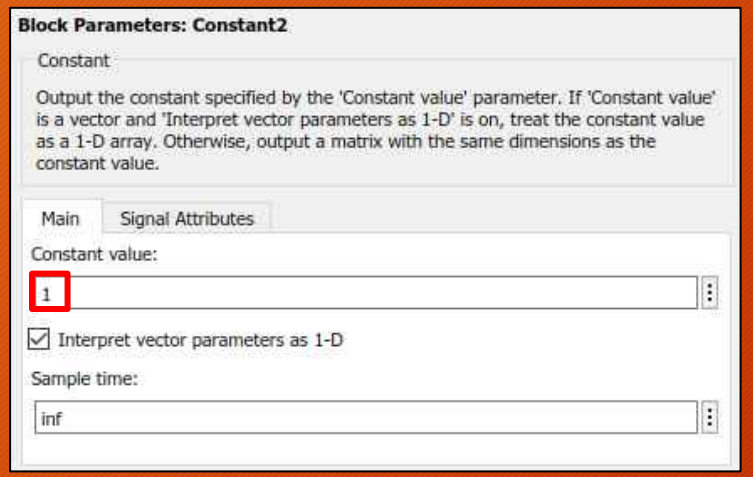
Example Value Input Implementation



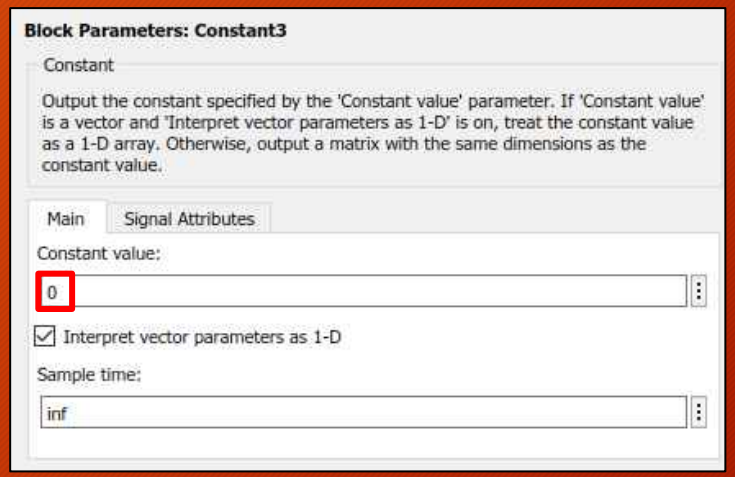
Two Simulink Constant Blocks are used to feed the switch inputs

  - Boolean constants

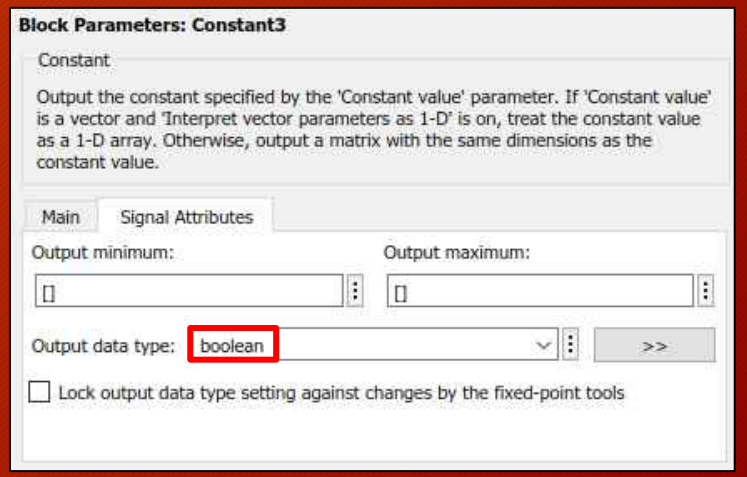
One



Zero



Both Constant Blocks in this Model are Boolean

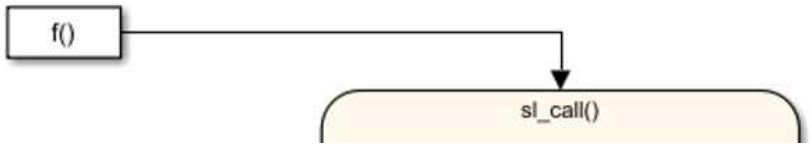


© Copyright 2022-2025 John G. Artus

Function-Call Generator

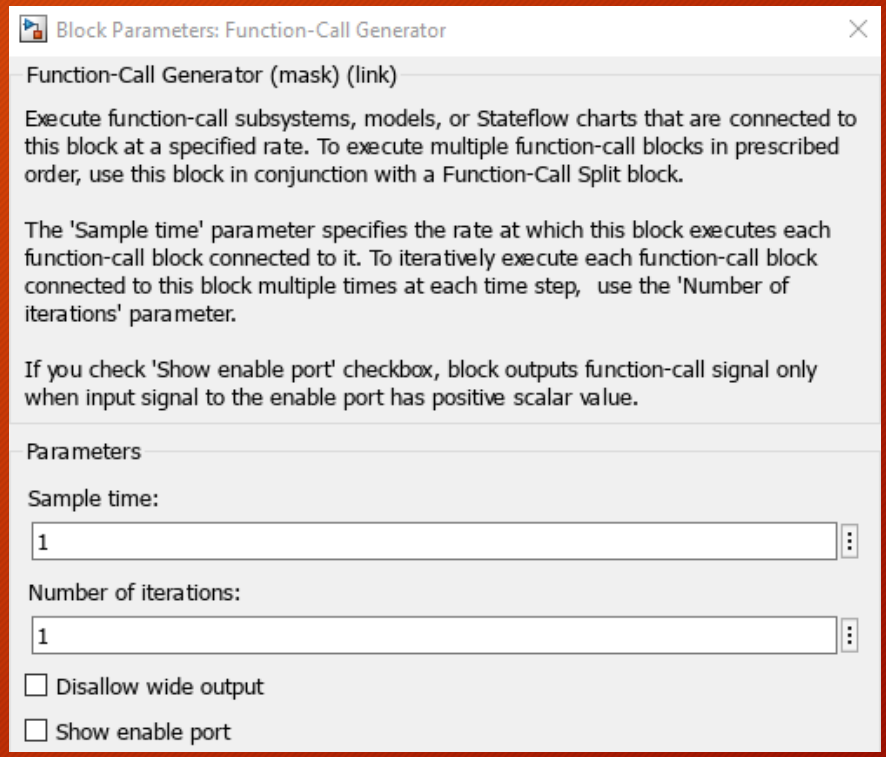
- The Function-Call generator is used to wake up the Stateflow Chart at a selected rate
- The generator for this model operates at a 1 second rate

Function-Call Generator Wakes-Up Stateflow Model



sl_call() is an input event that triggers the execution of the Stateflow Chart Block

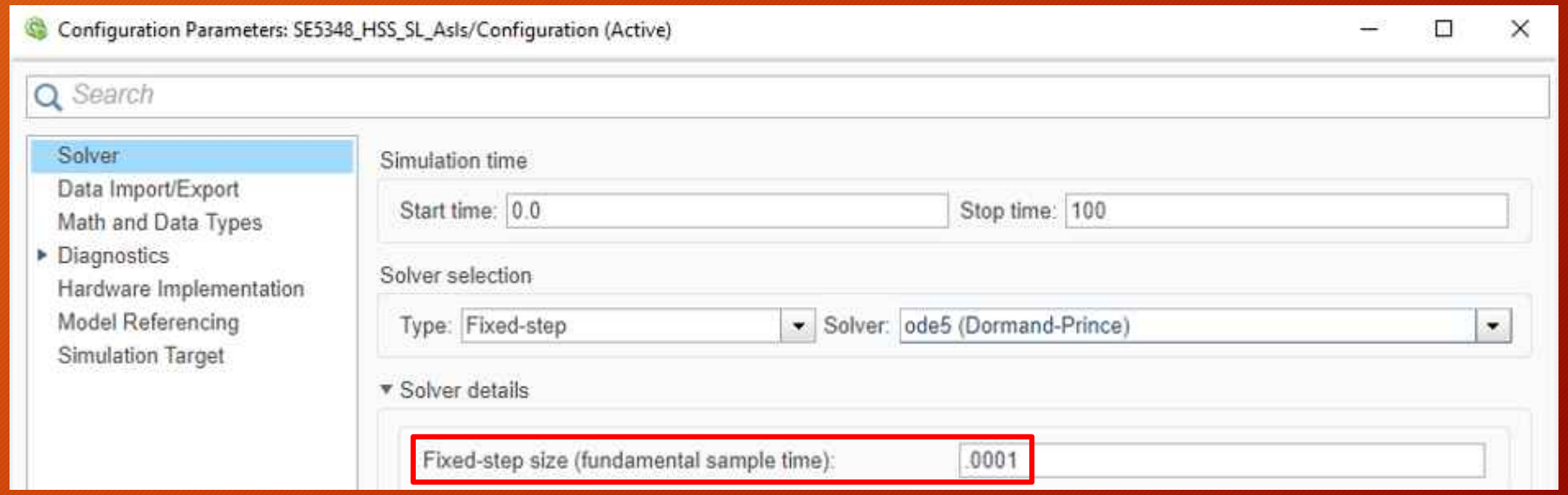
It is used here to initiate an evaluation of the state of the system using Stateflow at regular intervals, generated by the Function-Call Generator



© Copyright 2022-2025 John G. Artus

Modified Solver Time Step

- The original MathWorks model used a very small time step of 0.0001 seconds per time step
  - I believe this was due to the presence of a double-bounce detector in the motion sensor that was part of the original model
  - Since the motion sensor has been removed in this simplified version of the model and since the tight time step puts an unnecessary heat burden on CPUs, it was decided to reduce the fidelity of the time step significantly

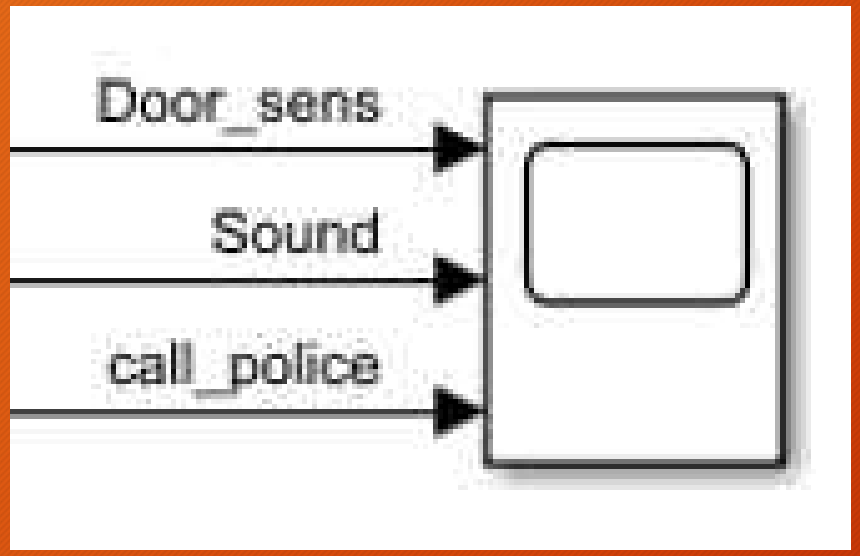


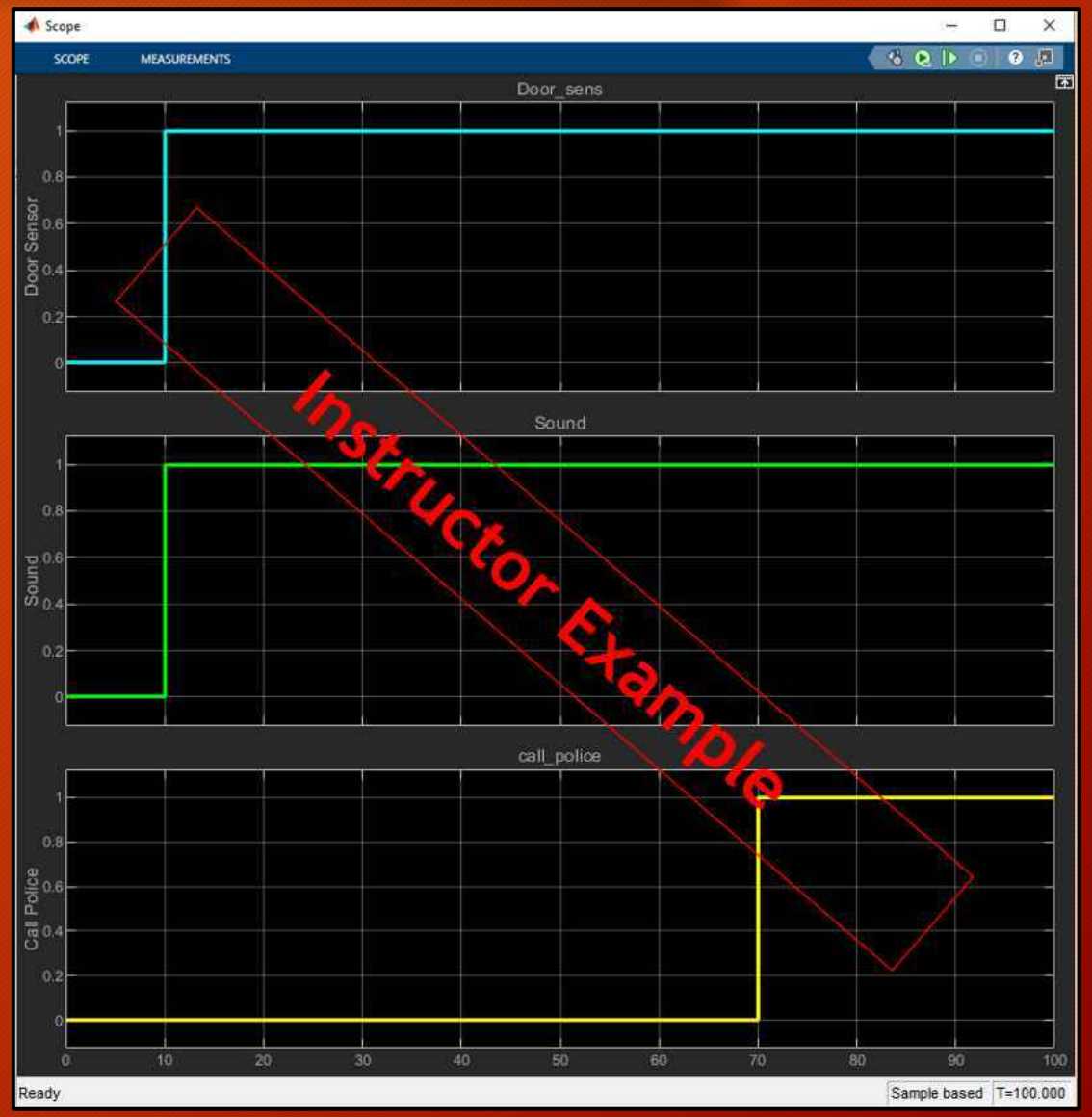
  - Instead, a revised time step of 0.1 seconds is being used in the simplified model

© Copyright 2022-2025 John G. Artus

Scope Block

- The HSS Model includes a Scope Block which plots data collected during a simulation run
- In the HSS model, output data is collected on three signals
  - Door_sens which indicates if a door intrusion is detected
  - Sound which indicates that the system is producing an audible warning (an “alert”)
  - call_police which indicates that the system has sent a call to the police to investigate the intrusion (an “alarm”)





© Copyright 2022-2025 John G. Artus

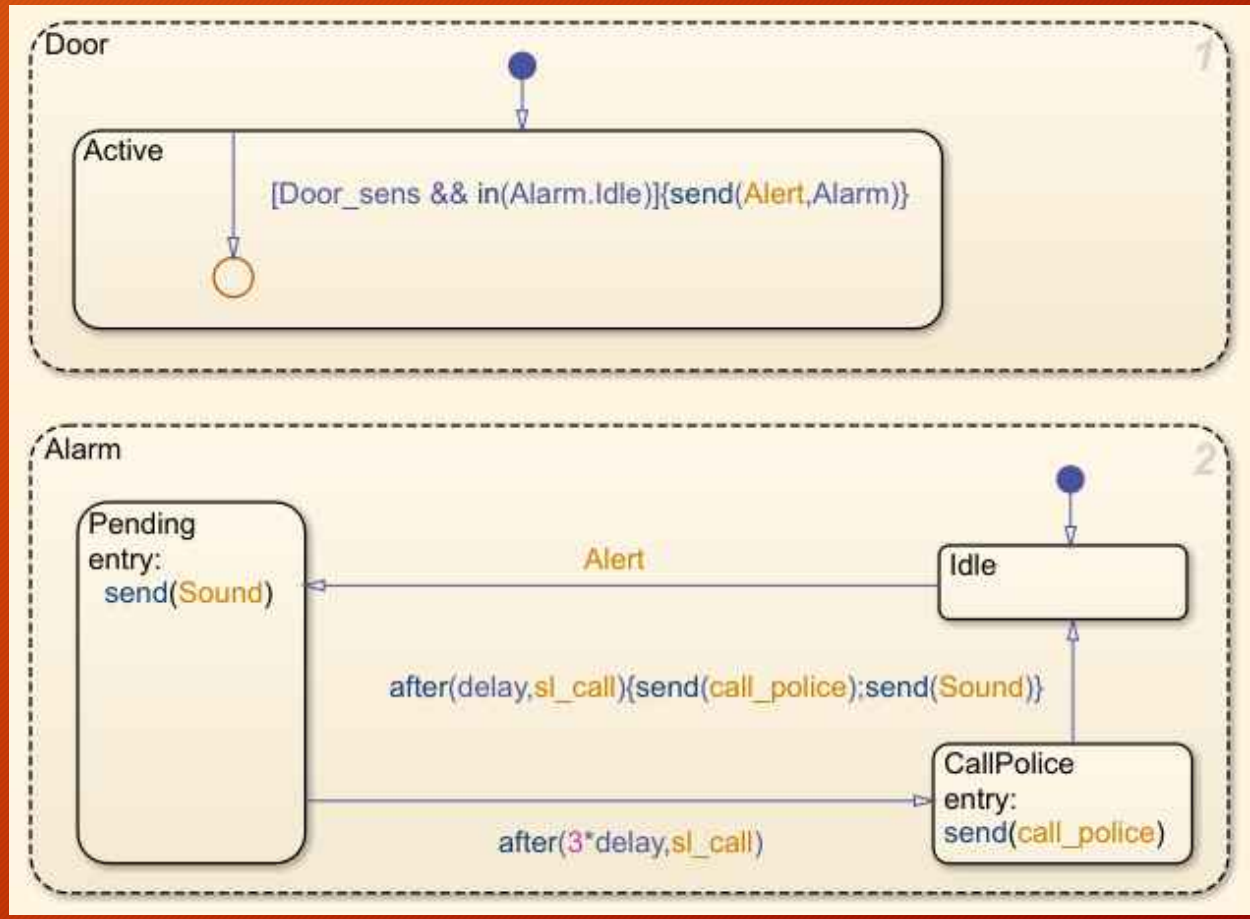
Section 2

A description of the Stateflow portion of the As-Is Home Security System (HSS) Model

© Copyright 2022-2025 John G. Artus

Stateflow Model of Security System State Behavior

- This model controls the behavior of the system based on the switch setting inputs provided by the Simulink (environment) portion of the model
- The Stateflow model controls how the system reacts to the tripping of the door sensor
- It also controls the sounding of the audible warning and calling of the police, based on operator responses following tripping of the door sensor



Stateflow simulates state-based behavior, showing how the system transitions from one state to another based on events occurring in and around the system

© Copyright 2022-2025 John G. Artus

Door Behavior

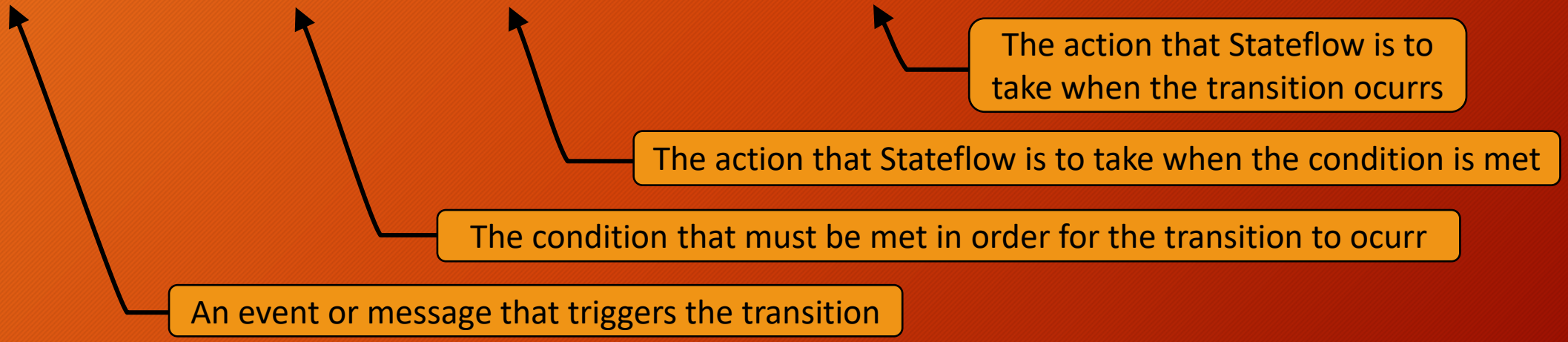
- Door has one substate: Active
  - This state does one thing: checks the status of the door sensor and reports an Alert if the sensor has been tripped
- Simulink reports the door sensor being tripped by setting Door_sens =1
- This event is triggered via the manual switch “Door sensor” in Simulink
- When the event “Door_sens” occurs, it sends an “Alert” signal to the Alarm state
  - The door sensor interprets a single positive trigger signal as an intrusion and issues an immediate alert
  - See further below for a description of the Alarm state

- If Door_sens = 1 and the parallel Alarm state is in substate Idle, then substate Active will send an Alert event to state Alarm

© Copyright 2022-2025 John G. Artus

Interpreting the Transition Label

- Transitions in Stateflow use a particular format, and understanding that format and using it correctly is critical to getting a Stateflow model to execute correctly
- The Transition Label in Stateflow is broken into parts using the following format
  - event_or_message[condition]{condition_action}/transition_action



- Specifying an event or message is optional
- The ? character is the default transition label
- You can specify multiple events using the OR logical operator (|)
- The absence of an event or message indicates that the transition takes place immediately on the occurrence of any event

© Copyright 2022-2025 John G. Artus

What are Events?

- An event is a Stateflow object that can trigger actions in a Stateflow or Simulink model
- Explicit events (defined by user) can have one of these types:
  - Input Event - Event that is broadcast to a Stateflow chart from outside the chart
  - Local Event - Event that can occur anywhere in a Stateflow chart but is visible only from within the parent object and its descendants
    - Local events are supported only in Stateflow charts in Simulink models
  - Output Event - Event that occurs in a Stateflow chart but is broadcast to a Simulink block
    - Output events are supported only in Stateflow charts in Simulink models
- Events do not have values like variables do – sending and receiving an event simply means something has happened in the scenario (that can trigger something else to happen)

© Copyright 2022-2025 John G. Artus

Use of Events in the Original HSS Model

- In state Door.Active, the transition

```
[Door_sens && in(Alarm.On.Idle)] {send(Alert,Alarm);}
```

  - uses the “send” function to transmit the event Alert to state Alarm
  - This only happens IF the system is currently in the state Alarm.Idle where the Alert event has meaning
- Inside state Alarm, the transition from substate Idle to substate Pending is triggered on reception of the Alert event

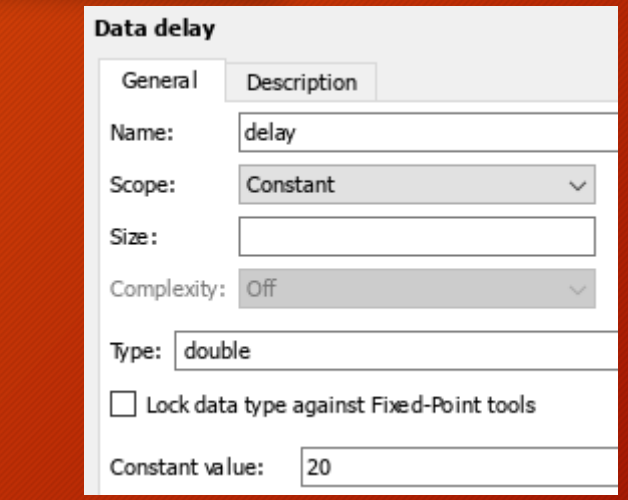
Transmit “Alert”

Respond to “Alert”

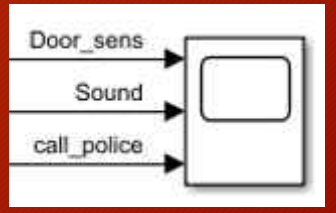
© Copyright 2022-2025 John G. Artus

Alarm Behavior

- The Alarm state has three substates: Idle, Pending, and CallPolice
- The system will transition from Idle to the Pending state when an Alert event occurs
  - The Alert event is sent (using “send(Alert,Alarm)”) to the Idle substate of the Alarm parent state (using “in(Alarm.Idle)”) by state Door
- After (3 * 20 =) 60 seconds in the Pending state, the system transitions to the CallPolice state in which the event “call_police” is broadcast
  - The input event sl_call controls the delay before the call to the police
  - In this instance, the event occurs inside a call to the temporal operator “after”, which triggers a transition after the chart receives the event “sl_call” 60 times since the associated state became active



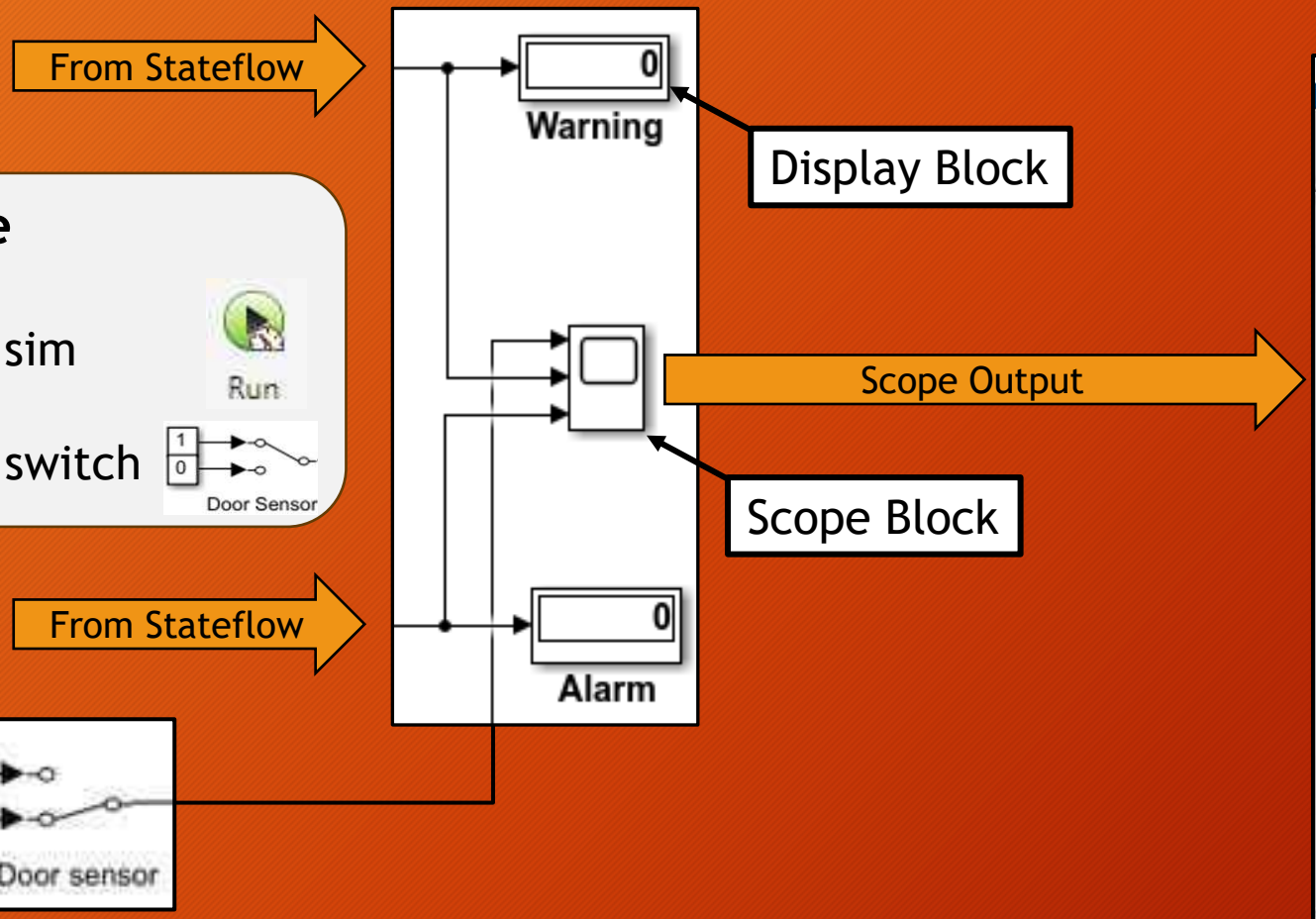
The Scope monitors these signals in order to follow what is happening in the system



© Copyright 2022-2025 John G. Artus

Monitoring System Behavior Using the Scope Block in Simulink

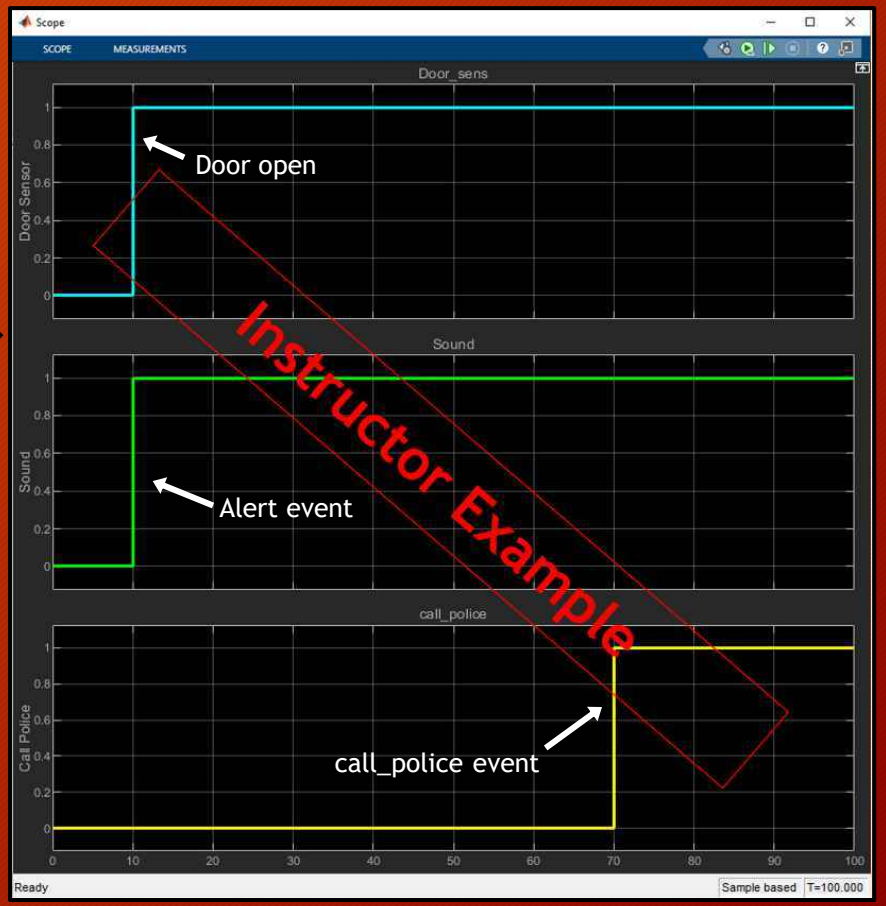
- The Scope Block is used to display signals over time
- The Display Block simply displays the current value that exists on a signal channel



Procedure

1. Run the sim

2. Flip the switch



© Copyright 2022-2025 John G. Artus

Controlling System Behavior by Event Broadcasting

- As was shown previously, parallel states can communicate with each other by broadcasting events to each other using the “send()” action
- Broadcasting of events can also be used to control the states of signals which indicate progress of internal processes
  - This approach is used to good effect in the state Alarm, shown below
- State Alarm involvs the broadcasting of two signals important to this security system process:
  - Sound – Indicating that an audible warning is being sounded to warn off intruders
  - call_police – Inidcating that a call has been placed to the police to report an intrusion into the residence
  - In this case, once one of these events has been broadcast, there needs to be a way to terminate the broadcast, else the audio warning and call to police would continue indefinitely
  - Terminating broadcasting of events can be accomplished by using a follow-up “send()” action
  - In this way, “send()” serves as a kind of toggle of the broadcasting of an event (the broadcasting of a signal)

Here, the audio warning (Sound) and the call to police (call_police) are both terminated during the transition from state CallPolice to state Idle

© Copyright 2022-2025 John G. Artus

Section 3

References

© Copyright 2022-2025 John G. Artus

References

- The MathWorks. (2022), *Synchronize Parallel States by Broadcasting Events*
  - https://www.mathworks.com/help/stateflow/gs/events.html

© Copyright 2022-2025 John G. Artus