42 Sim Project Objectives

This project enables you to run a spacecraft attitude controller in a cFS app in a closed-loop simulation using the 42 simulator. Closed-loop simulation is a method of running a system by connecting a controller to a simulator that models the time-varying behavior of the spacecraft and its environment.

Detailed project instructions with videos can be found at

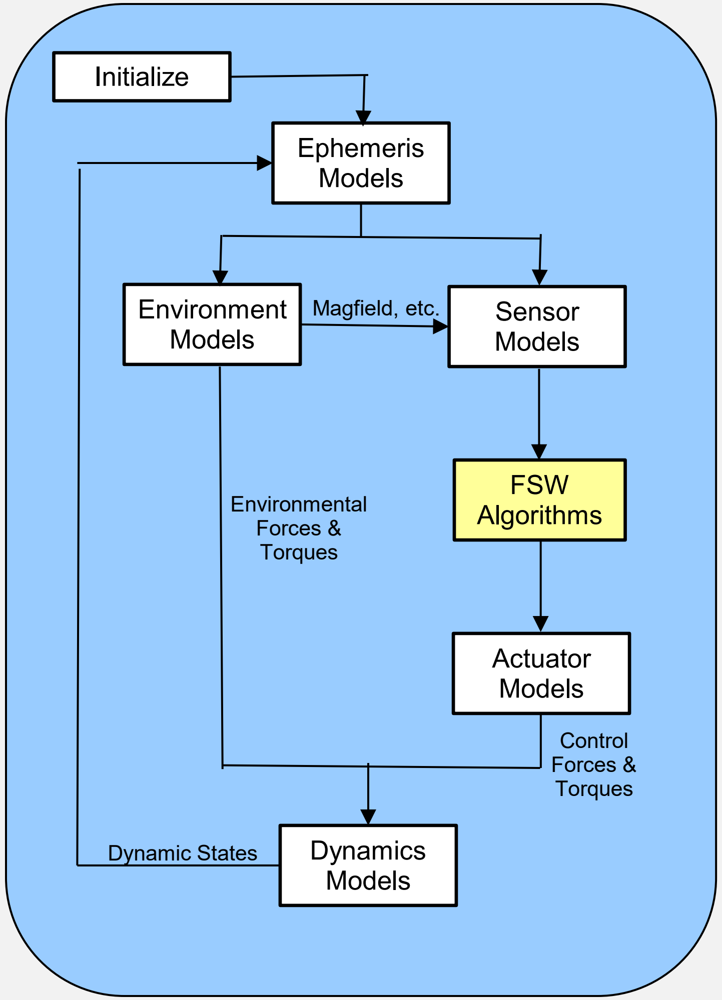
https://spacesteps.com/TODO/

Software Installation (1 of 2)

This project requires a cFS library and two apps to be installed in the cFS target. It also requires the 42 Simulator to be downloaded from gihub, configured and built in its

1. Use Basecamp tCreate 42 Simulation project using the Create Project tool
   a. TODO: Add project
   b. Automation will require expanding the current project

2. Start the cFS

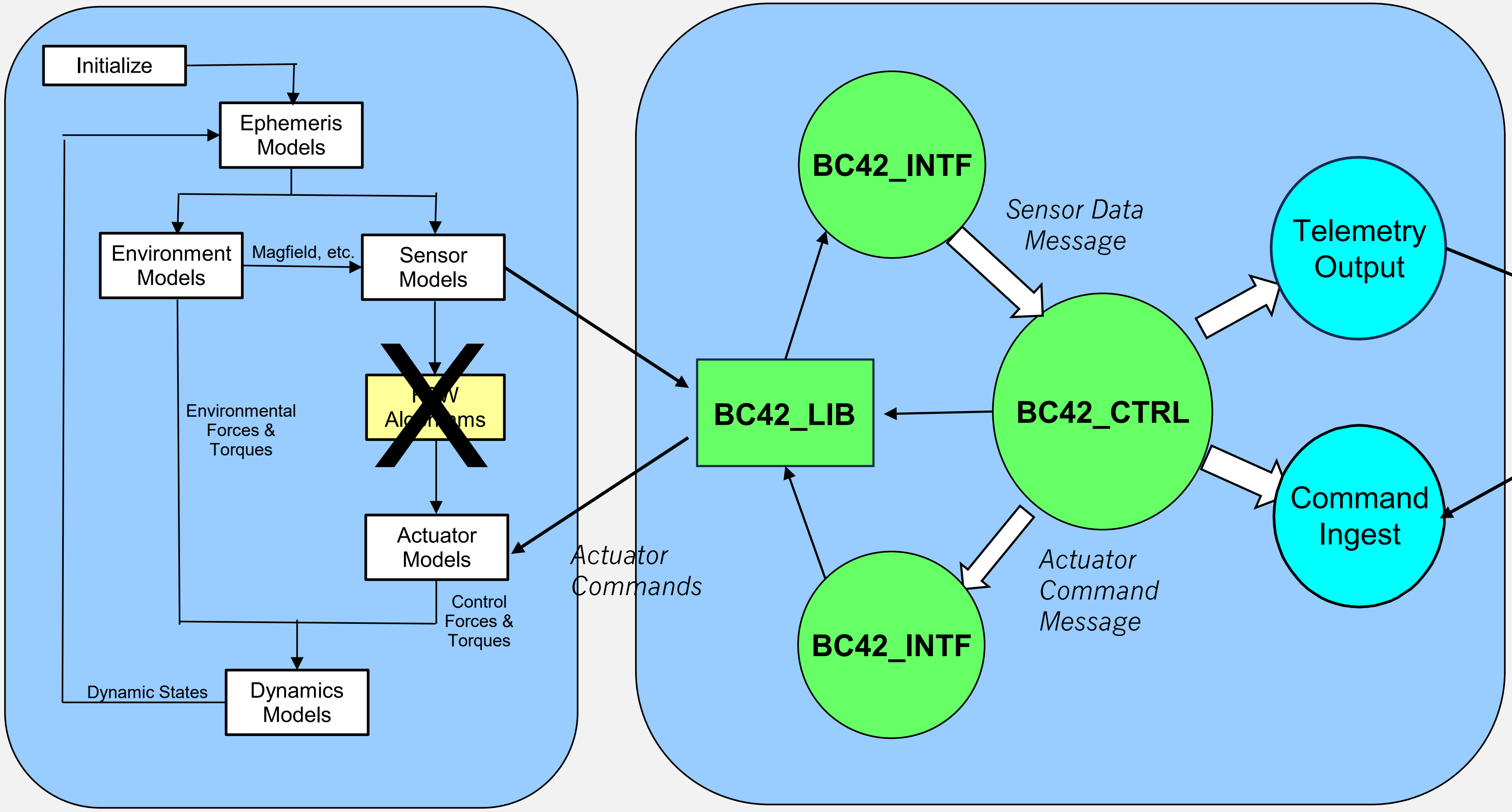

Initialize
Ephemeris Models
Environment Models
Magfield, etc.
Sensor Models
FSW Algorithms
Environmental Forces & Torques
Actuator Models
Control Forces & Torques
Dynamic States
Dynamics Models

This app receives sensor data from the 42 simulator and sends actuator commands to 42. 42 models the spacecraft and its environment helps answer the common cFS question, “How should I design a cFS app to manage a science payload?” There are multiple approaches towards solving this problem. This project is not intended to provide a survey of designs, rather it shows a specific design that you can evaluate within the context of your situation.

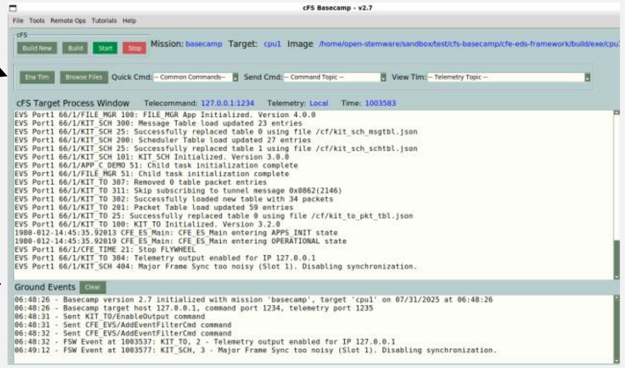
A secondary objective is to show how a cFS library and application can be used to simulate a payload. This effective strategy lets you run your payload manager app prior to having a test configuration with the target hardware. It’s also a flexible environment that lets you test error paths.

42 Simulator

cFS Target



Basecamp GUI



Mission: basecamp Target: cpu1 Image /home/open-stemware/sandbox/test/cfs-basecamp/cfe-eds-framework/build/exe/cpu

Quick Cmd: -- Common Commands-- Send Cmd: -- Command Topic -- View Tlm: -- Telemetry Topic --

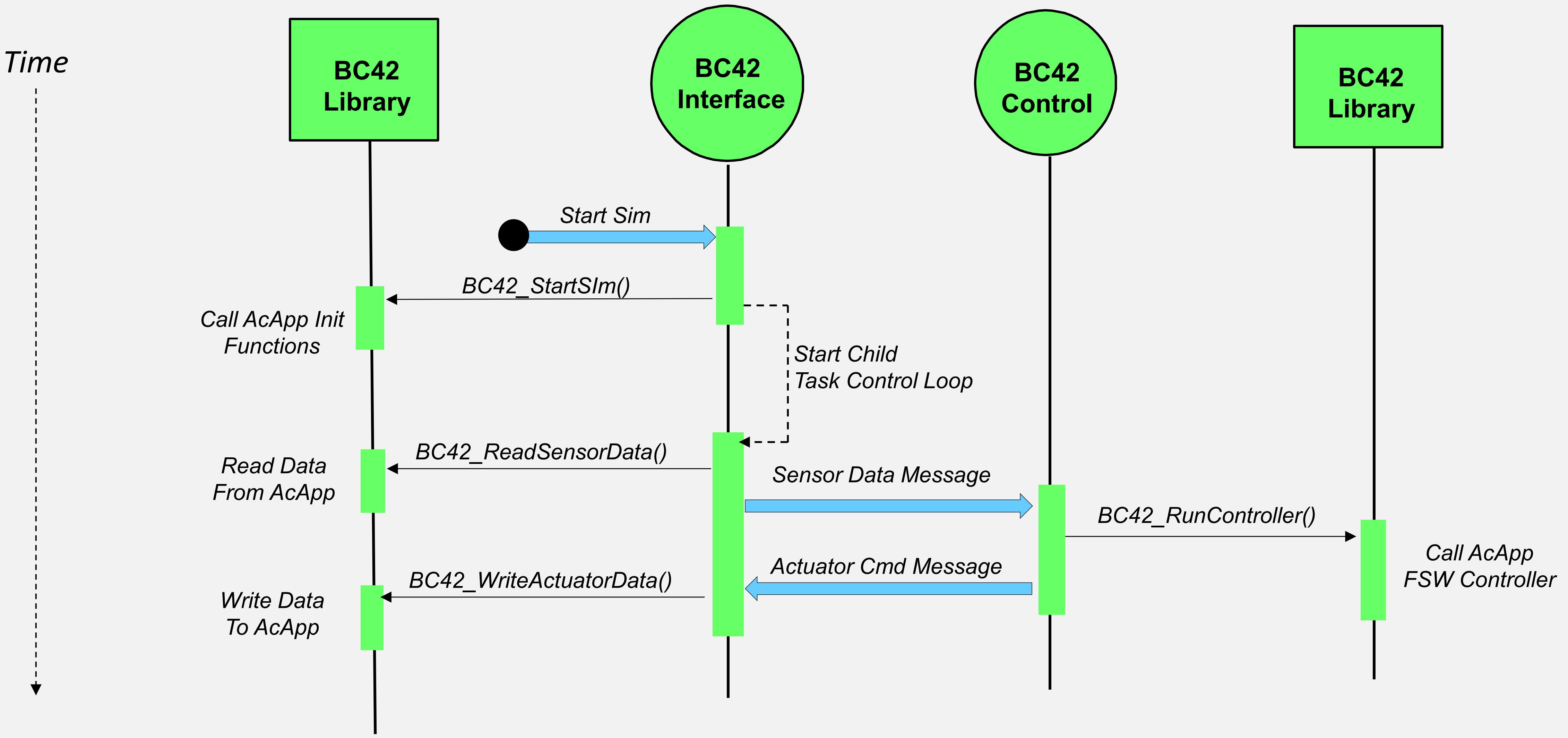
cFS Target Process Window Telecommand: 127.0.0.1:1234 Telemetry: Local Time: 1003583

```
EVS Port1 66/1/FILE_MGR 100: FILE_MGR App Initialized. Version 4.0.0
EVS Port1 66/1/KIT_SCH 300: Message Table load updated 23 entries
EVS Port1 66/1/KIT_SCH 25: Successfully replaced table 0 using file /cf/kit_sch_msgtbl.json
EVS Port1 66/1/KIT_SCH 200: Scheduler Table load updated 27 entries
EVS Port1 66/1/KIT_SCH 25: Successfully replaced table 1 using file /cf/kit_sch_schtbl.json
EVS Port1 66/1/KIT_SCH 101: KIT_SCH Initialized. Version 3.0.0
EVS Port1 66/1/APP_C_DEMO 51: Child task initialization complete
EVS Port1 66/1/FILE_MGR 51: Child task initialization complete
EVS Port1 66/1/KIT_TO 307: Removed 0 table packet entries
EVS Port1 66/1/KIT_TO 311: Skip subscribing to tunnel message 0x0862(2146)
EVS Port1 66/1/KIT_TO 302: Successfully loaded new table with 34 packets
EVS Port1 66/1/KIT_TO 201: Packet Table load updated 59 entries
EVS Port1 66/1/KIT_TO 25: Successfully replaced table 0 using file /cf/kit_to_pkt_tbl.json
EVS Port1 66/1/KIT_TO 100: KIT_TO Initialized. Version 3.2.0
1980-012-14:45:35.92013 CFE_ES_Main: CFE_ES_Main entering APPS_INIT state
1980-012-14:45:35.92019 CFE_ES_Main: CFE_ES_Main entering OPERATIONAL state
EVS Port1 66/1/CFE_TIME 21: Stop FLYWHEEL
EVS Port1 66/1/KIT_TO 304: Telemetry output enabled for IP 127.0.0.1
EVS Port1 66/1/KIT_SCH 404: Major Frame Sync too noisy (Slot 1). Disabling synchronization.
```

Ground Events

```
06:48:26 - Basecamp version 2.7 initialized with mission 'basecamp', target 'cpu1' on 07/31/2025 at 06:48:26
06:48:26 - Basecamp target host 127.0.0.1, command port 1234, telemetry port 1235
06:48:31 - Sent KIT_TO/EnableOutput command
06:48:31 - Sent CFE_EVS/AddEventFilterCmd command
06:48:32 - Sent CFE_EVS/AddEventFilterCmd command
06:48:32 - FSW Event at 1003537: KIT_TO, 2 - Telemetry output enabled for IP 127.0.0.1
06:49:12 - FSW Event at 1003577: KIT_SCH, 3 - Major Frame Sync too noisy (Slot 1). Disabling synchronization.
```

Start Simulation



● Start Event

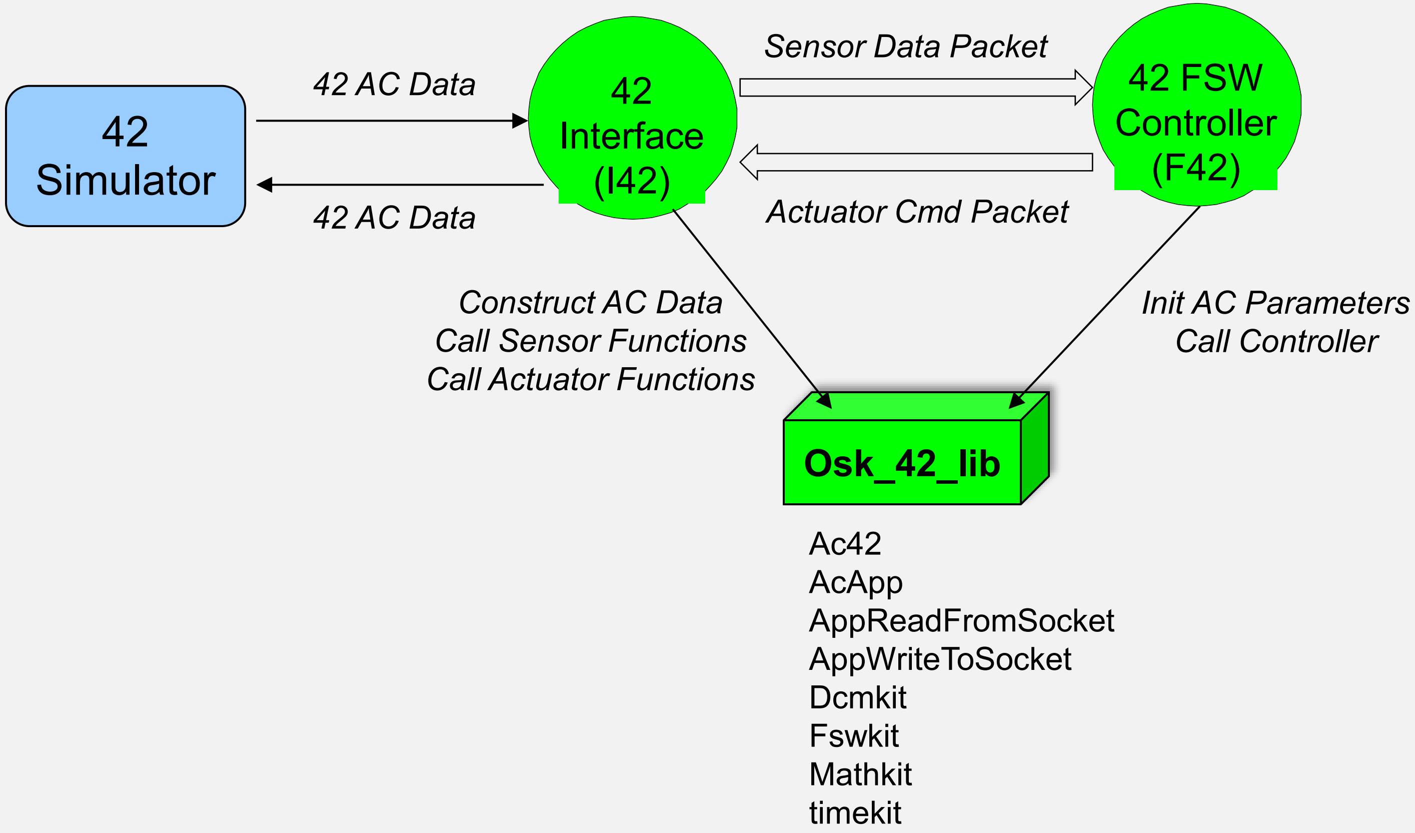
Software Bus Message

Library Call

- **42 release 2042 includes a standalone “AcApp”. AcApp has been integrated into an OSK library called osk_42_lib. The I42 app provides the interface to the 42 simulator. It constructs osk_42_lib’s “Ac struct” and calls the sensor data processing and actuator command functions defined in AcApp. The F42 app calls the controller function in AcApp.**

- **I created the I42 and F42 apps to demonstrate how the sensor data processing, attitude determination and actuator commanding are often distributed across apps.**

- **OpenSatKit/training/OSK-Apps-ADC.pdf**



42 Simulator

42 AC Data

42 AC Data

42 Interface (I42)

Sensor Data Packet

Actuator Cmd Packet

42 FSW Controller (F42)

Construct AC Data
Call Sensor Functions
Call Actuator Functions

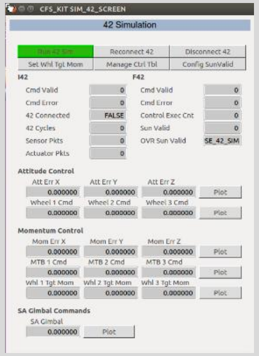
Init AC Parameters
Call Controller

Osk_42_lib

Ac42
AcApp
AppReadFromSocket
AppWriteToSocket
Dcmkit
Fswkit
Mathkit
timekit

COSMOS

Screens



42 Simulation

I42

Cmd Valid 0
Cmd Error 0
42 Connected FALSE
42 Cycles 0
Sensor Pkts 0
Actuator Pkts 0

F42

Cmd Valid 0
Cmd Error 0
Control Exec Cnt 0
Sun Valid 0
OVR Sun Valid SE_42_SIM

Attitude Control

Att Err X	Att Err Y	Att Err Z
0.000000	0.000000	0.000000

Wheel 1 Cmd	Wheel 2 Cmd	Wheel 3 Cmd
0.000000	0.000000	0.000000

Momentum Control

Mom Err X	Mom Err Y	Mom Err Z
0.000000	0.000000	0.000000

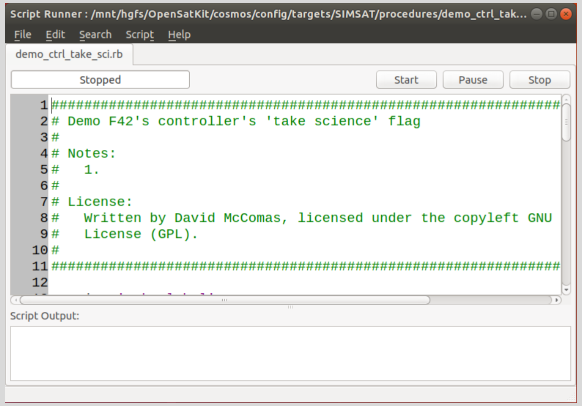
MTB 1 Cmd	MTB 2 Cmd	MTB 3 Cmd
0.000000	0.000000	0.000000

Whl 1 Tgt Mom	Whl 2 Tgt Mom	Whl 3 Tgt Mom
0.000000	0.000000	0.000000

SA Gimbal Commands

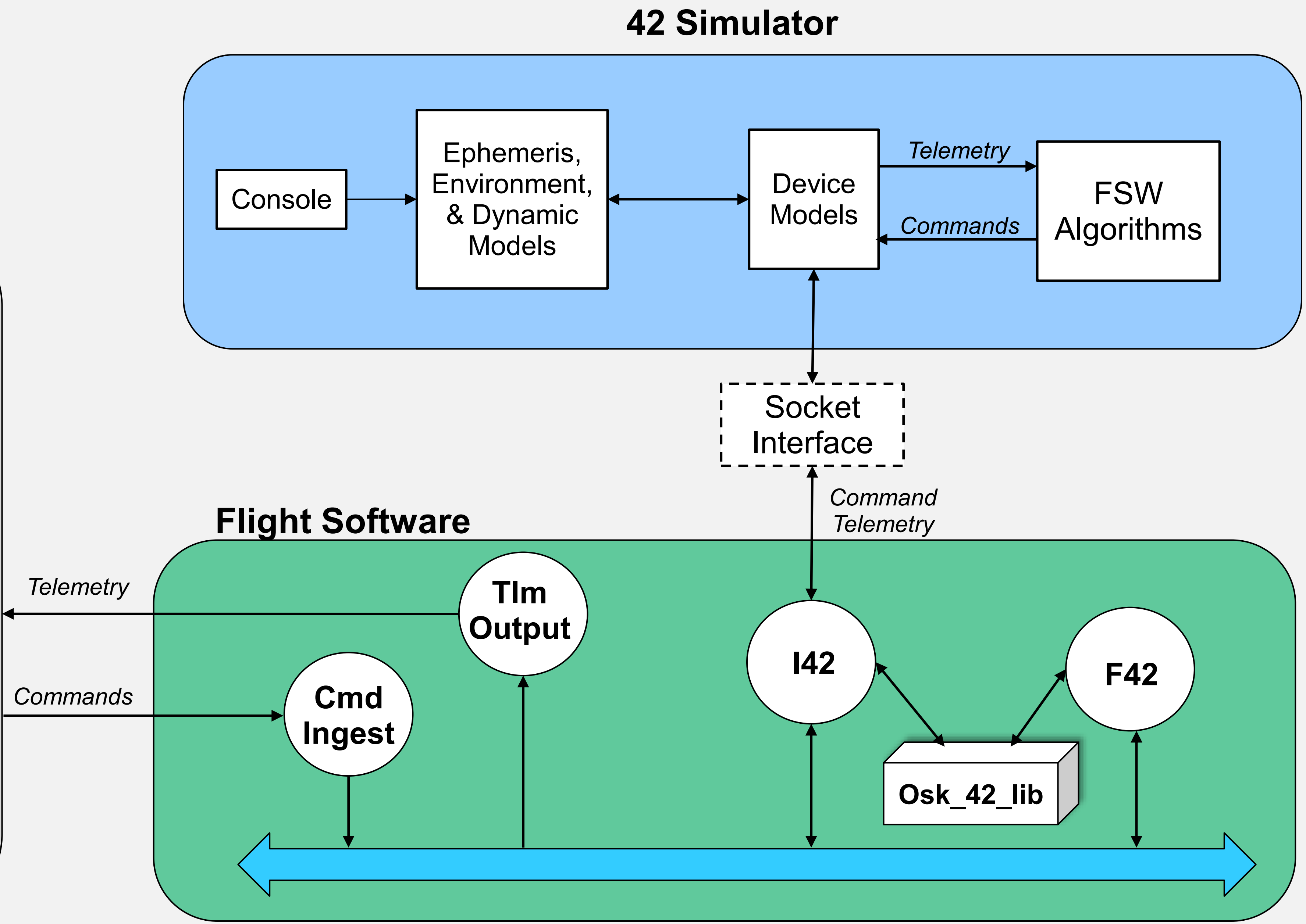
SA Gimbal 0.000000

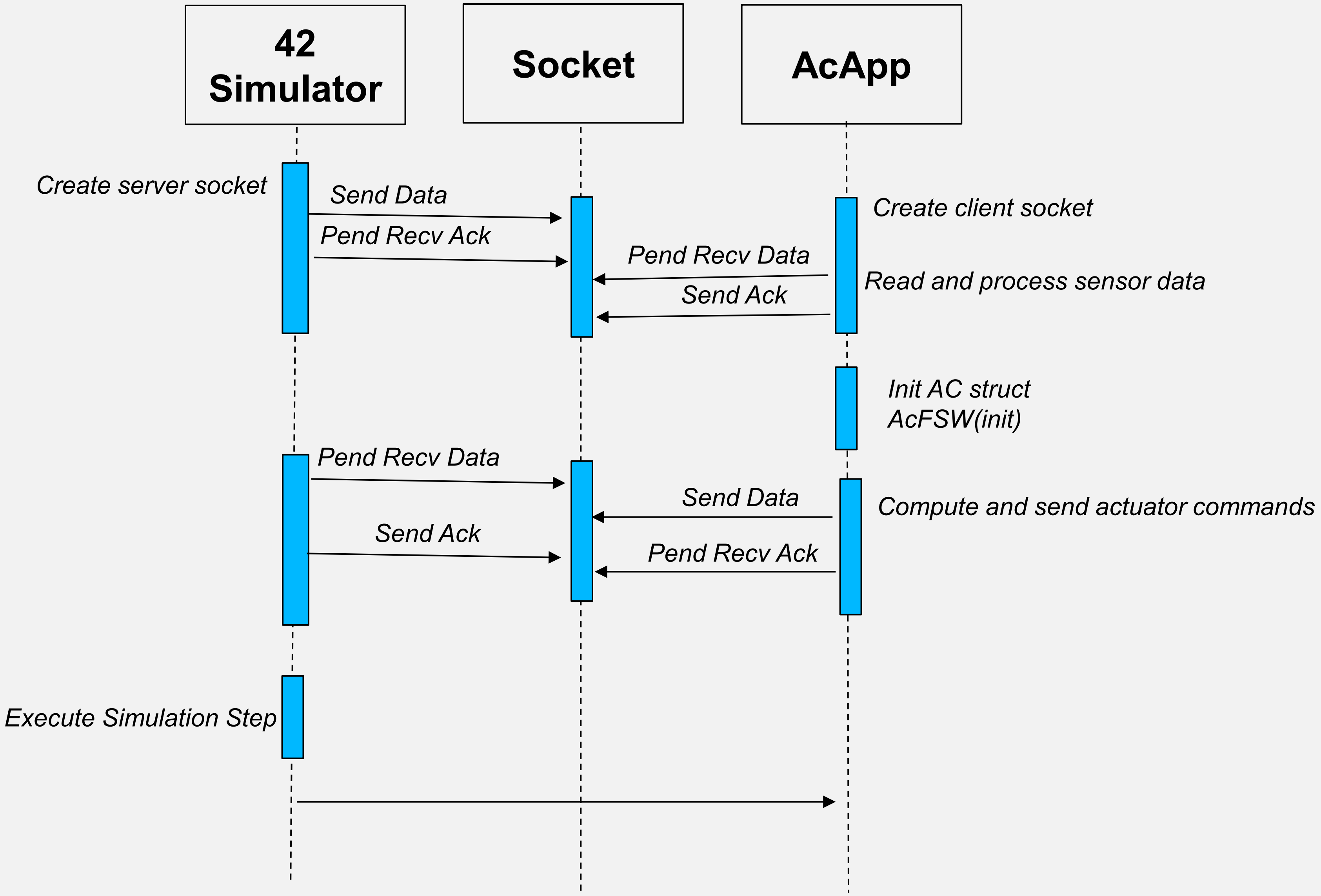
Scripts


Script Runner : /mnt/hgfs/OpenSatKit/cosmos/config/targets/SIMSAT/procedures/demo_ctrl_tak...
File
Edit
Search
Script
Help
demo_ctrl_take_sci.rb
Start
Pause
Stop
Script Output:

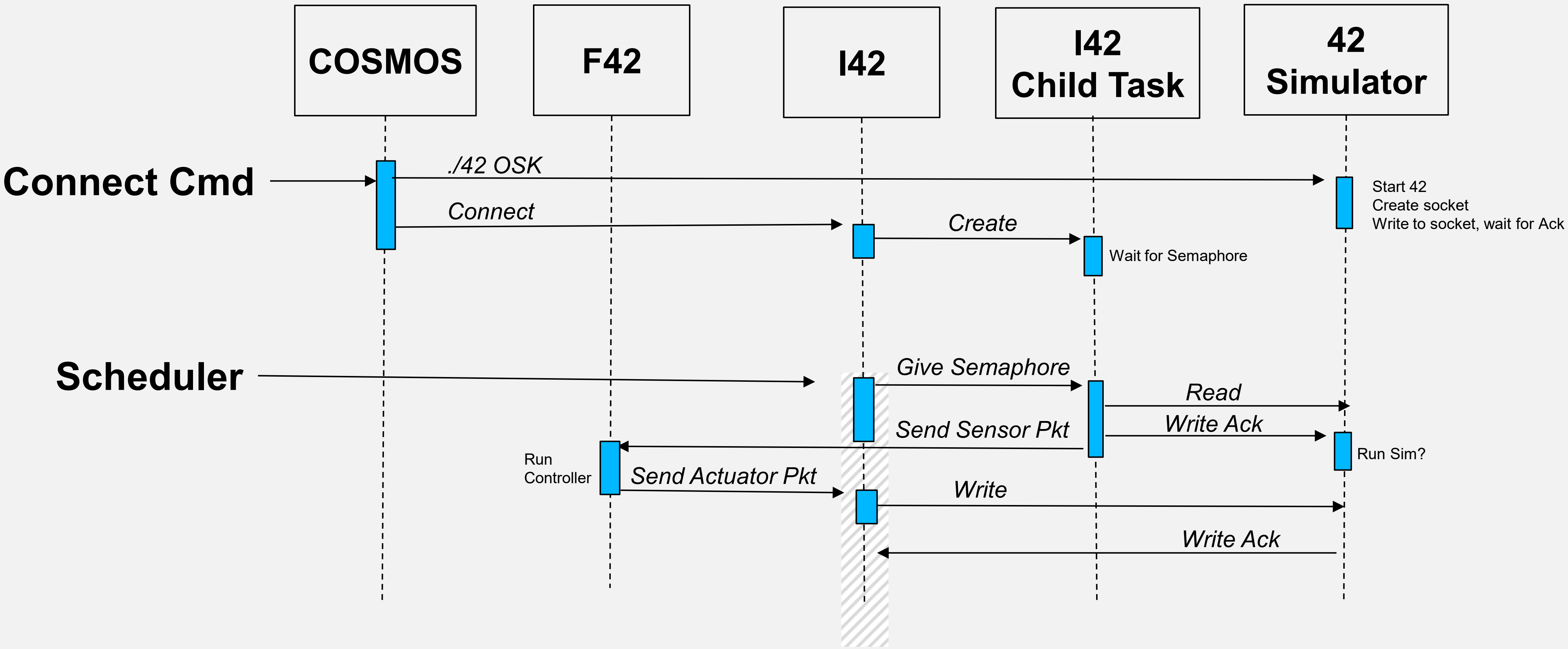
Stopped

```
################################################################
# Demo F42's controller's 'take science' flag
#
# Notes:
#   1.
#
# License:
#   Written by David McComas, licensed under the copyleft GNU
#   License (GPL).
#
################################################################
```





42 Simulator
Socket
AcApp
Create server socket
Send Data
Pend Recv Ack
Create client socket
Pend Recv Data
Read and process sensor data
Send Ack
Init AC struct
AcFSW(init)
Pend Recv Data
Send Data
Compute and send actuator commands
Send Ack
Pend Recv Ack
Execute Simulation Step



COSMOS

F42

I42

I42 Child Task

42 Simulator

Connect Cmd

./42 OSK

Start 42
Create socket
Write to socket, wait for Ack

Connect

Create

Wait for Semaphore

Scheduler

Give Semaphore

Read

Write Ack

Send Sensor Pkt

Run Sim?

Run
Controller

Send Actuator Pkt

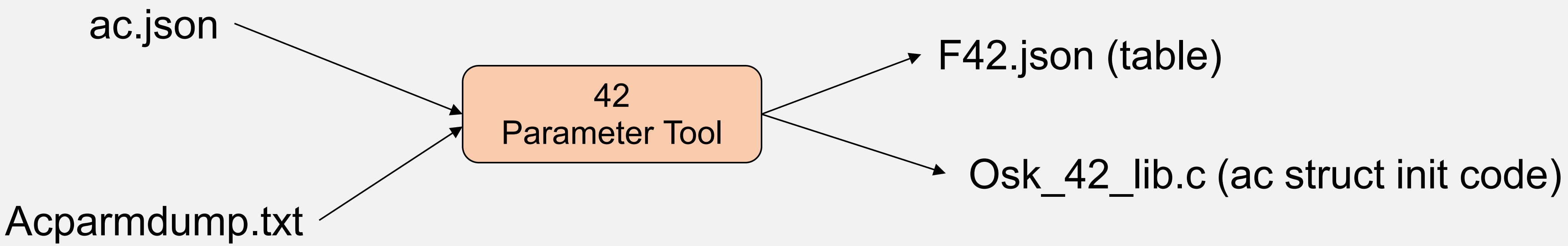
Write

Write Ack

I42 is client, 42 Simulator is server

▨ I42 Manage Execution

Ideas for a tool to generate the osk_42_lib code that initializes the ac struct


ac.json
Acparmdump.txt
42
Parameter Tool
F42.json (table)
Osk_42_lib.c (ac struct init code)