Tiny Runtime to Run Model-Based Software on CubeSats

Model Based Space Systems and Software Engineering - MBSE2020

2020

Agenda

- Project background and objectives
- Implementation
  - Kazoo code patterns
  - PolyORB removal
  - Optimizations and fixes
- Validation (DemoSat)
- Ada support

BACKGROUND AND OBJECTIVES

TASTE by ESA

- "A tool-chain targeting heterogeneous embedded systems, using a model-based development approach"
- System architecture – **AADL**
- Data – **ASN.1** (with ACN)
- Behaviour – **SDL** (but can also be **C**, **Ada**...)
- Can generate executables targeting:
  - x86
  - SPARC (e.g. Leon3)
  - ARM (e.g. STM32F407)
  - and now **MSP430** (MSP430FR5969)!
- https://taste.tools and https://taste.tuxfamily.org/wiki/

MSP430 by Texas Instruments

- **General purpose**, **ultra-low-power**, **affordable**, mixed-signal MCUs
- **Radiation hardened** versions available (50krad for MSP430FR5969-SP)
- Good choice for certain subsystems and CubeSats
- But also challenging:
  - **16-bit** ISA
  - **low memory** (e.g. 64 kB)
  - low speed (e.g. 16 MHz)
- https://www.ti.com/microcontrollers/msp430-ultra-low-power-mcus/overview.html

N7 Space

- N7 Space develops both ground support and flight software
- TASTE is used extensively:
  - PROBA3
  - ASN.1/ACN Modelling IDE
  - PUS C deployment
  - ARM BSP
  - CoreSight
  - MBSE Implement
- N7 Space is a member of the TASTE Steering Commitee
- http://n7space.com/

Goal

- Provide support for **MSP430 in TASTE**
- Validate it using a realistic (**CubeSat** class) use-case
- An (intended) by-product: various optimizations which enable **targeting other low-resource platforms** in the future, extending the applicability of model-based approaches

IMPLEMENTATION

Kazoo

- Kazoo uses a set of templates to transform **AADL** architecture model into **code**, **build scripts** and other artefacts
- Kazoo was improved to better handle many generation targets
  - to avoid lots of **@@IF@@**
- A set of templates that use **FreeRTOS** primitives was added
- A new property: `TASTE_IV_Properties::Default_CodeGen`
  - Ada support was not yet available (but it is now!)

PolyORB

- TASTE targets rely on code generated by Ocarina taking advantage of PolyORB-HI/Ada|C middleware for abstraction
- **PolyORB is very resource intensive** (for platforms with tens of kB)
- **A lightweight replacement** was provided by implementing all the primitives required by TASTE using **FreeRTOS** features
- The solution can be applied to different targets
- Interop between PolyORB and non-PolyORB partitions requires a dedicated TASTE device driver

Other

- C support in OpenGEODE required some maintenance
  - case-sensitivity
  - naming collisions
- ASN1SCC received some optimizations to target 16-bit platforms
  - Ada base type sizes now match C ones
  - consistency between type sizes is now less compiler-dependent
- Large memory model support was added to the FreeRTOS MSP430 port

VALIDATION

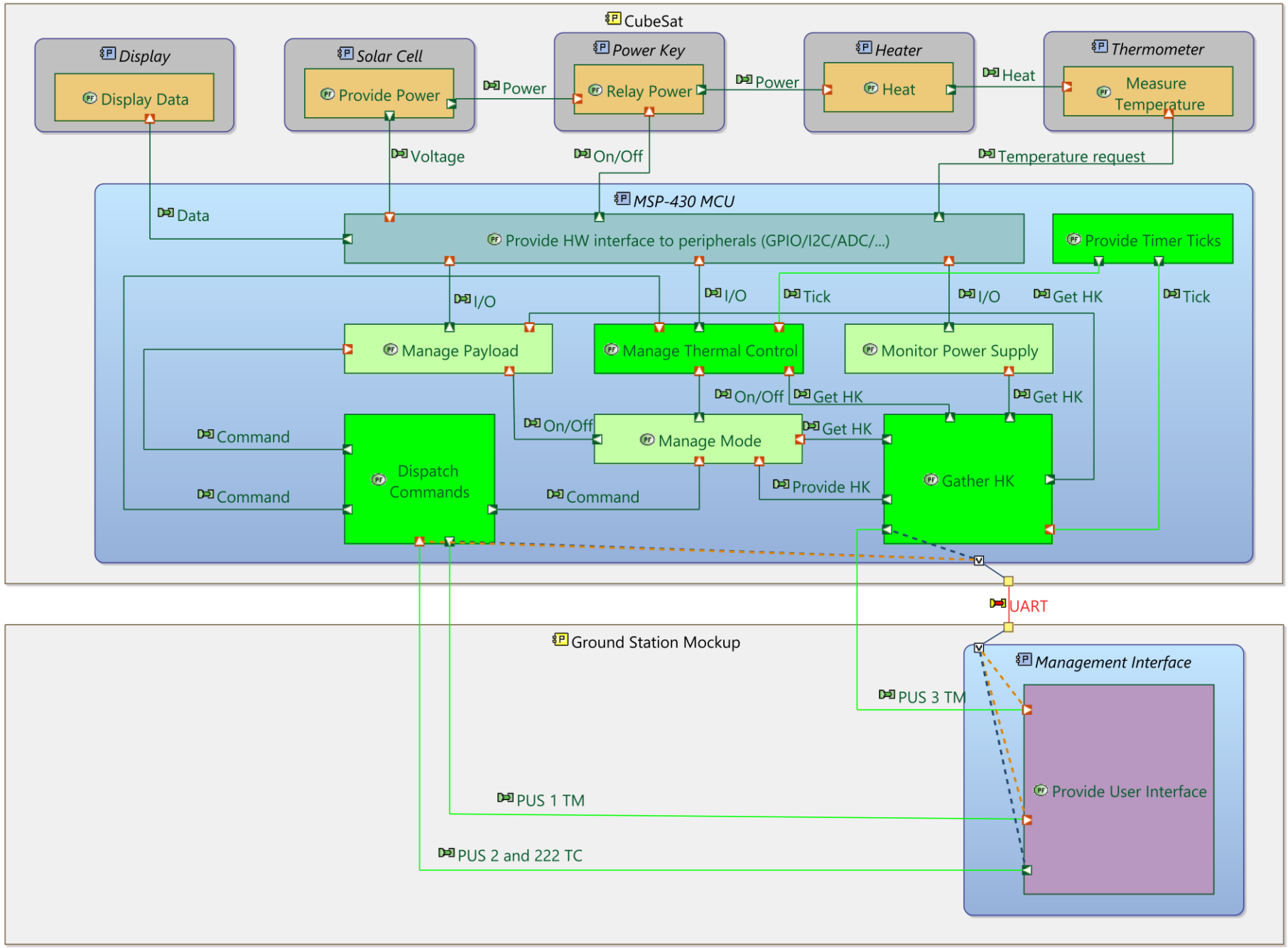
Demo-Sat

- Demonstration of the new runtime
- MSP430FR5969 Launchpad Evaluation Kit serves as the on-board computer
  - 16-bit
  - 16 MHz
  - 2kB of SRAM and 64kB of FRAM
- Design was to be agreed during the project execution - a high-level idea was iterated in Capella, agreed with ESA and then remodelled in TASTE
- Code automatically generated using TASTE and deployed onto the hardware

Capella model

- Capella allowed **fast iteration** and allowed to reach an **unambiguous** agreement between a highly **distributed team** (all personnel were working from home, which was a new experience at the time)
- The model includes hardware peripherals and non-software related exchanges (**missing from the later TASTE model**, but critical for cost estimation, procurement...)
- Capella provides **high-level** modelling capabilities
- Low learning curve (if you know where to start)
- Capella **does not provide code-generation** capabilities*
- Model **verification** in Capella is quite **weak**

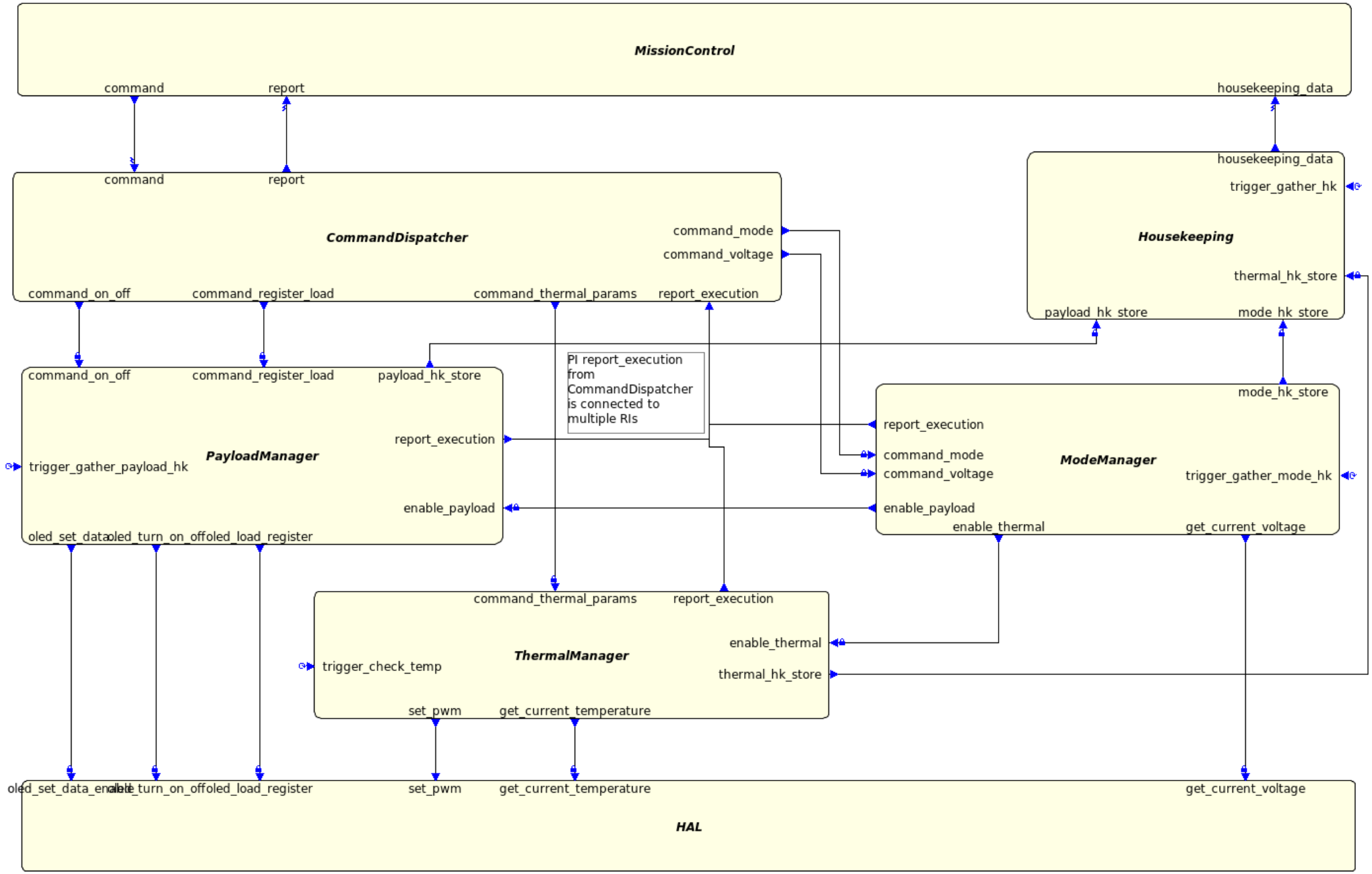
*plugins may, but to a limited extent


CubeSat
Display
Display Data
Solar Cell
Provide Power
Power
Power Key
Relay Power
Power
Heater
Heat
Heat
Thermometer
Measure Temperature
Voltage
On/Off
Temperature request
MSP-430 MCU
Data
Provide HW interface to peripherals (GPIO/I2C/ADC/...)
Provide Timer Ticks
I/O
I/O
Tick
I/O
Get HK
Tick
Manage Payload
Manage Thermal Control
Monitor Power Supply
On/Off
Get HK
Get HK
Command
On/Off
Manage Mode
Get HK
Dispatch Commands
Gather HK
Command
Command
Provide HK
UART
Ground Station Mockup
Management Interface
PUS 3 TM
Provide User Interface
PUS 1 TM
PUS 2 and 222 TC

TASTE model

- TASTE is software focused
- TASTE provides **mid-to-low-level** modelling capabilities
- TASTE **provides code-generation** capabilities
- Model **verification** in TASTE is **better***
- No need to include "Provide Timer Ticks" function (cyclic interface)
- The model was iterated to optimize for memory consumption
- The resulting model is still quite similar to a subset of the original one

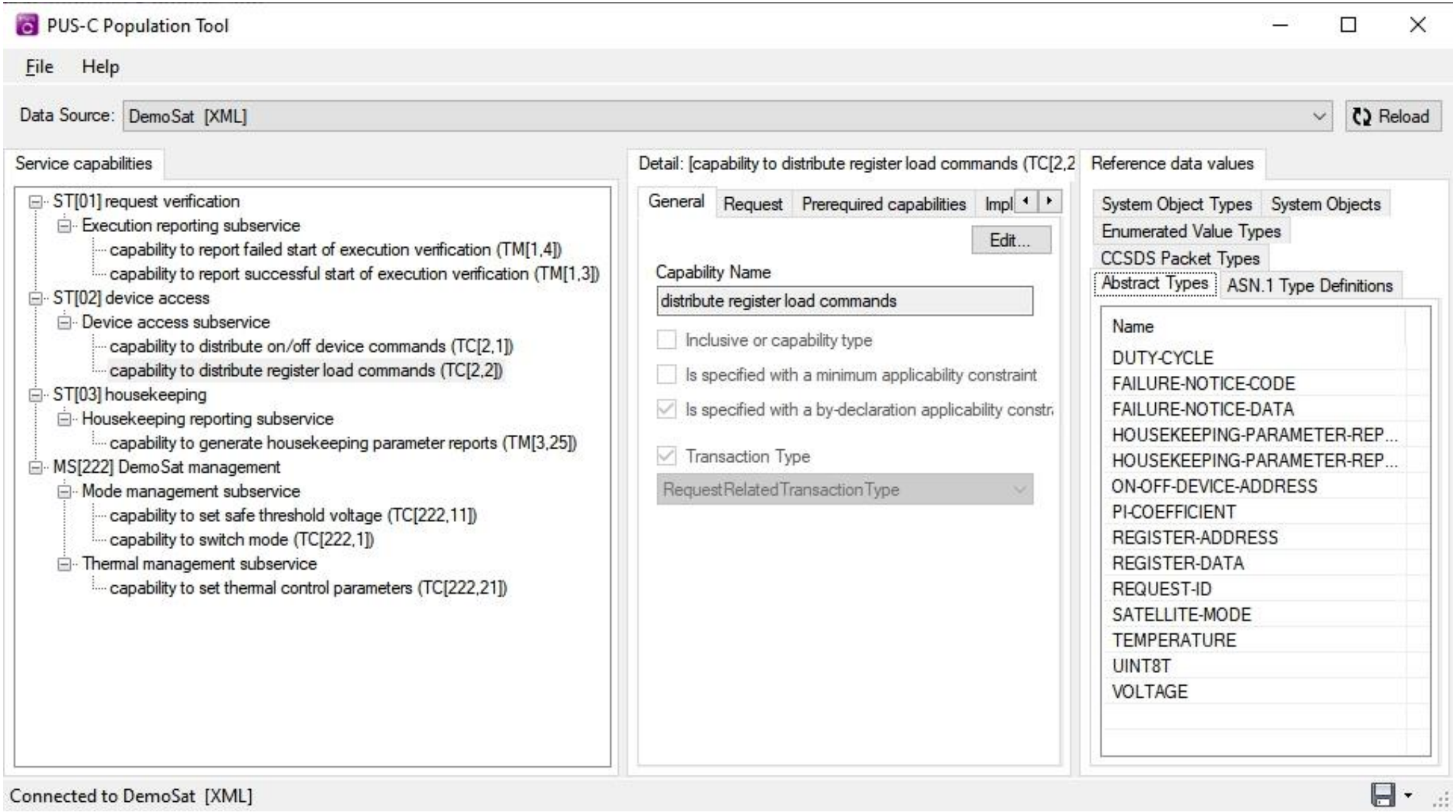
*there is room for improvement


MissionControl
command
report
housekeeping_data
command
report
CommandDispatcher
command_mode
command_voltage
command_on_off
command_register_load
command_thermal_params
report_execution
housekeeping_data
trigger_gather_hk
Housekeeping
thermal_hk_store
payload_hk_store
mode_hk_store
command_on_off
command_register_load
payload_hk_store
PI report_execution from CommandDispatcher is connected to multiple RIs
mode_hk_store
report_execution
PayloadManager
report_execution
command_mode
command_voltage
ModeManager
trigger_gather_payload_hk
trigger_gather_mode_hk
enable_payload
enable_payload
enable_thermal
get_current_voltage
oled_set_data
oled_turn_on_off
oled_load_register
command_thermal_params
report_execution
enable_thermal
ThermalManager
trigger_check_temp
thermal_hk_store
set_pwm
get_current_temperature
oled_set_data_enable
oled_turn_on_off
oled_load_register
set_pwm
get_current_temperature
get_current_voltage
HAL

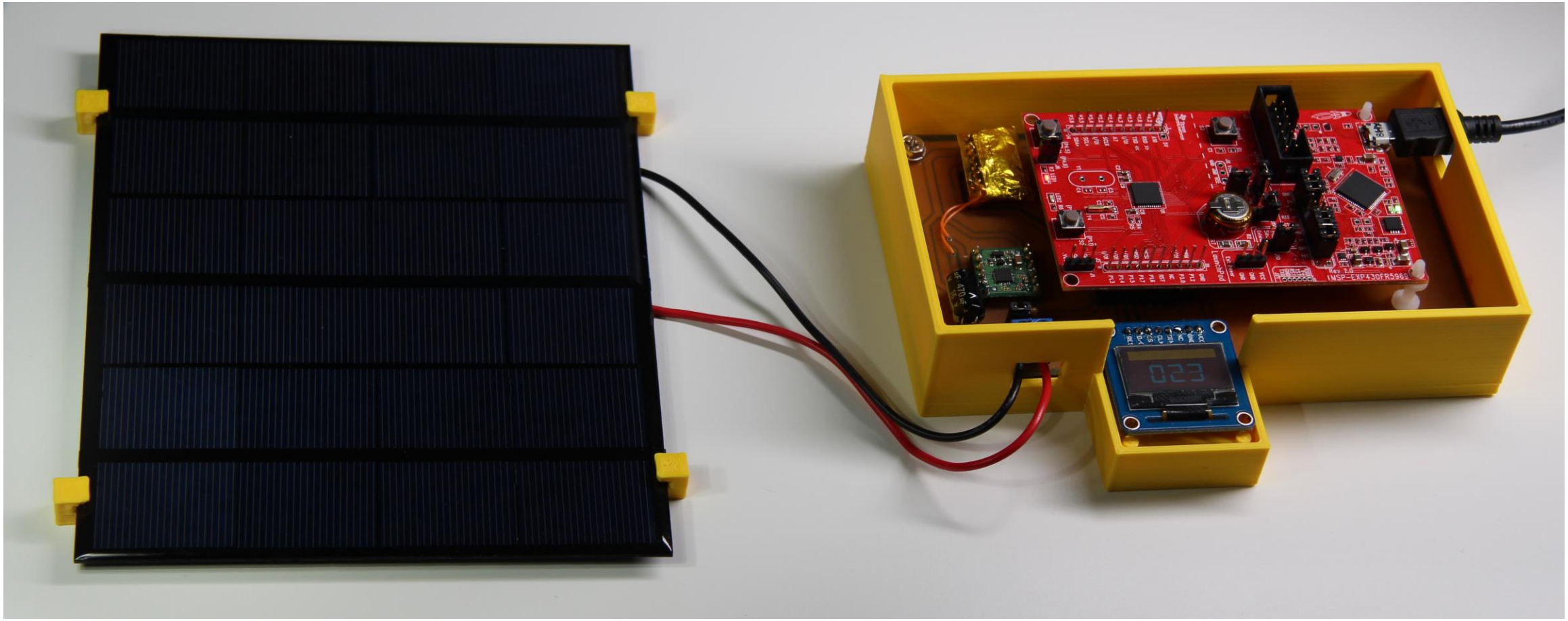
TASTE model - data

- DemoSat uses a **tailored PUS-C compliant protocol**
- The required ASN.1 model was generated using **PUS C ASN.1-GEN** tool developed within the scope of the **PUS-C Deployment** project
- The used population data schema ensures compliance with the **PUS Foundation Model** (presented in Chapter 5 of the ECSS-E-ST-70-41C standard, elaborated during the PUS-C Deployment project using ORM)
- The generator used a tailored "implicit knowledge provider" ("replacement" of a part of Chapter 7 of the standard) to have a lighter, non-CCSDS encapsulation
- Services 1, 2 and 3 were tailored (trimmed to a bare minimum*)
- Service 222 is custom

*PUS 1 minimum capability set is still an overkill for some applications


PUS-C Population Tool
File
Help
Data Source:
DemoSat [XML]
Reload
Service capabilities
ST[01] request verification
Execution reporting subservice
capability to report failed start of execution verification (TM[1,4])
capability to report successful start of execution verification (TM[1,3])
ST[02] device access
Device access subservice
capability to distribute on/off device commands (TC[2,1])
capability to distribute register load commands (TC[2,2])
ST[03] housekeeping
Housekeeping reporting subservice
capability to generate housekeeping parameter reports (TM[3,25])
MS[222] DemoSat management
Mode management subservice
capability to set safe threshold voltage (TC[222,11])
capability to switch mode (TC[222,1])
Thermal management subservice
capability to set thermal control parameters (TC[222,21])
Detail: [capability to distribute register load commands (TC[2,2
General
Request
Prerequired capabilities
Impl
Edit...
Capability Name
distribute register load commands
Inclusive or capability type
Is specified with a minimum applicability constraint
Is specified with a by-declaration applicability constr
Transaction Type
RequestRelatedTransactionType
Reference data values
System Object Types
System Objects
Enumerated Value Types
CCSDS Packet Types
Abstract Types
ASN.1 Type Definitions
Name
DUTY-CYCLE
FAILURE-NOTICE-CODE
FAILURE-NOTICE-DATA
HOUSEKEEPING-PARAMETER-REP...
HOUSEKEEPING-PARAMETER-REP...
ON-OFF-DEVICE-ADDRESS
PI-COEFFICIENT
REGISTER-ADDRESS
REGISTER-DATA
REQUEST-ID
SATELLITE-MODE
TEMPERATURE
UINT8T
VOLTAGE
Connected to DemoSat [XML]

Demo-Sat Hardware



housekeeping_data

Field	Value
housekeeping_data	
packet_data	tm_3_25_housekeepingParameterReport
tm_3_25_housekeepingParameterReport	
housekeepingParameterNotificationN...	
data	hk_id_all
hk_id_all	
mode	satellite-mode-operational
thermal_enabled	True
payload_enabled	True
input_voltage	6.6124877930
current_temperature	45.0000000000
target_temperature	45.0000000000
pi_integral	44.3750000000
thermal_duty_cycle	0.4437499940
payload_data_enabled	True
payload_data	42

Plot | Meter

command

Field	Value	Pr
command		
packet_data	tc_2_2_distributeRe...	
tc_2_2_distributeRegisterLoadCommands		
instructionsToDistributeARegisterLoadCommand	1	
elem_0		
registerData	register_address_di...	
register_address_display_data	42	

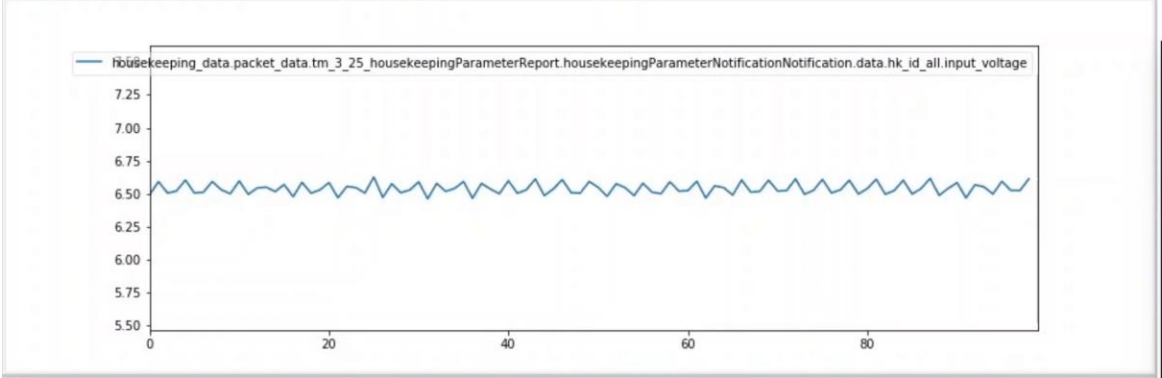
Send TC

report

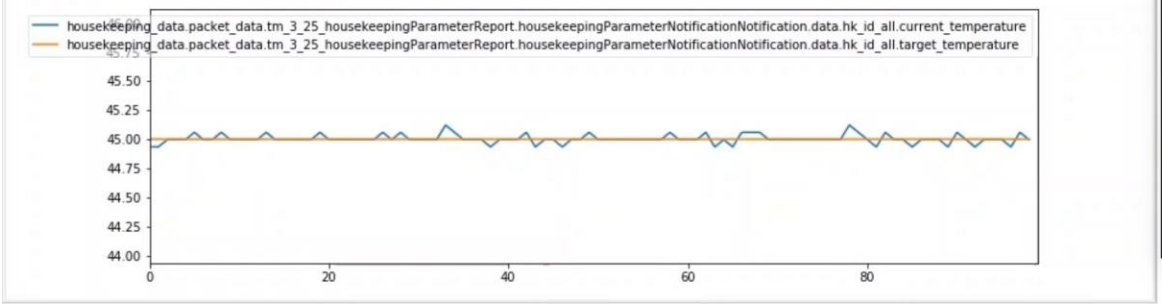
Field	Value
report	
packet_data	tm_1_3_successfulStartOfExecutionVerificationReport
tm_1_3_successfulStartOfE...	
successfulStartOfExecut...	
requestID	request-id-distribute-register-load-command

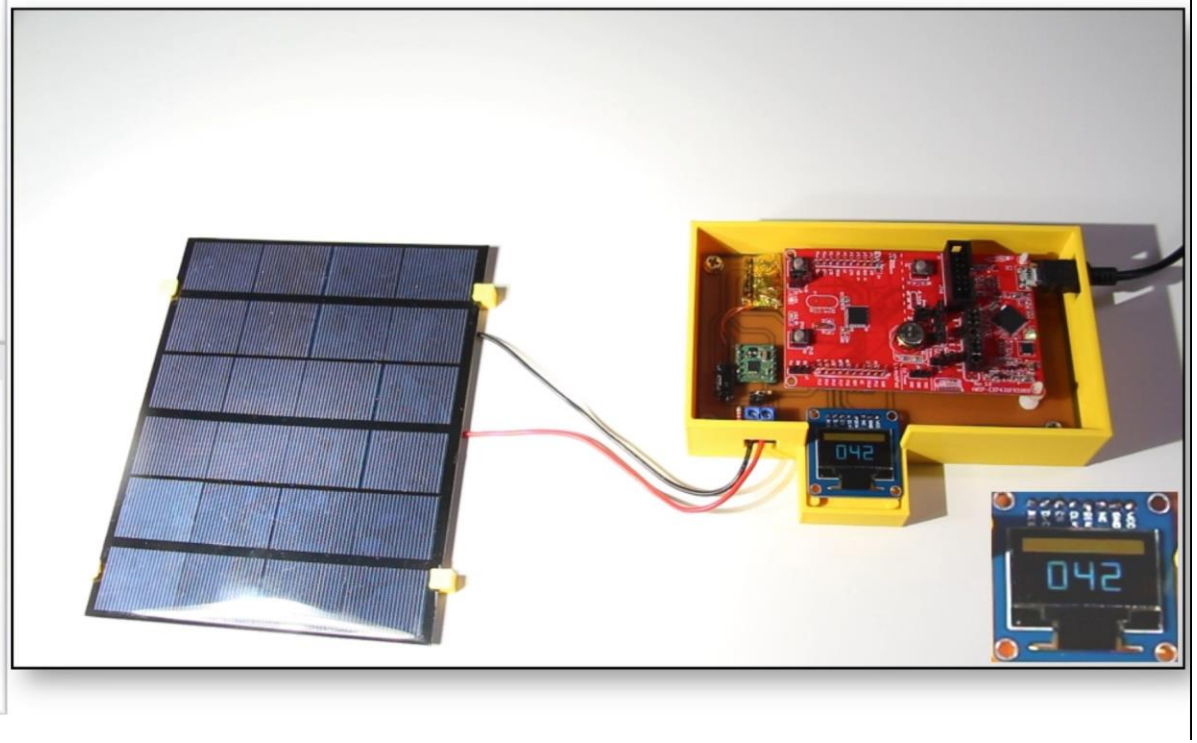
Plot | Meter

Plot 1



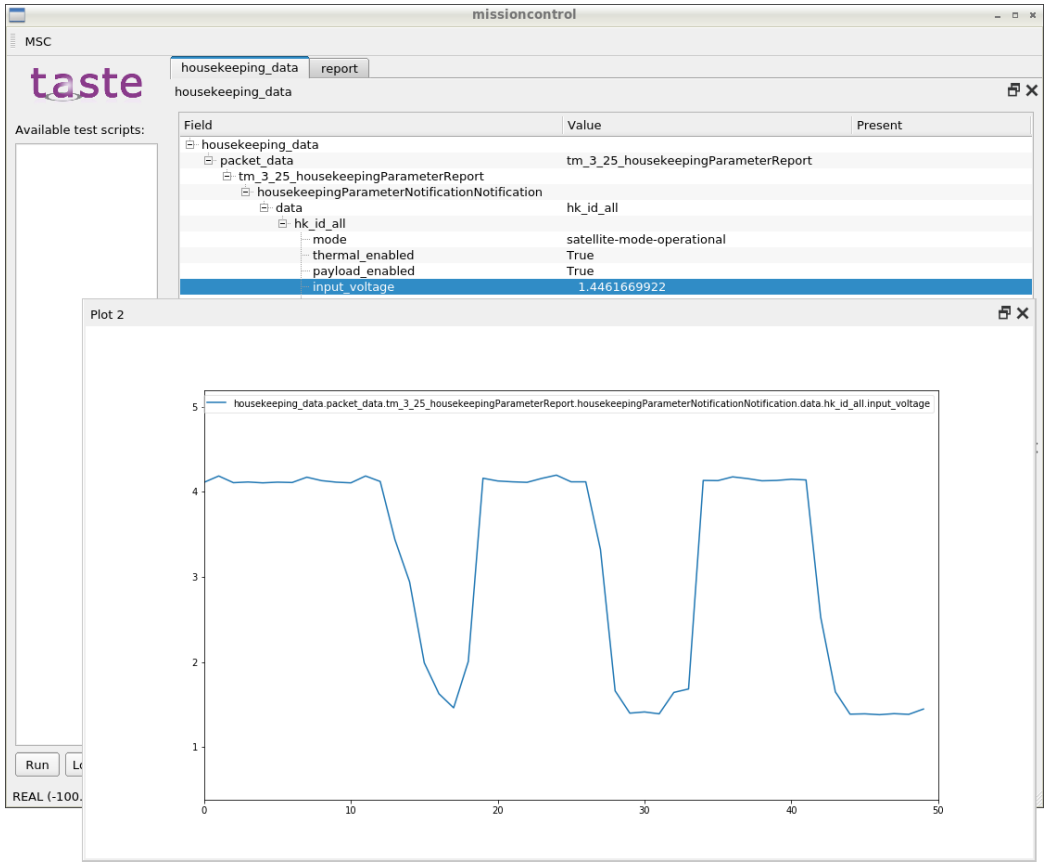
Plot 2





Demo-Sat Summary

- MSP430FR5969 can handle TASTE models
- About 10k of free memory for other features
- Demo-Sat is public on TASTE GitLab
- Documentation is available on TASTE Wiki
- A demonstration movie is coming soon to an internet near you!



ADA SUPPORT

Custom Ada compiler

- OpenGEODE can translate SDL into both Ada and C
- Ada and C can be used instead of SDL
- GNU GCC compiler available in Debian repositories is outdated and contains bugs affecting MSP430FRXXX chips
- AdaCore GNAT Community Edition does not support MSP430
- A new toolchain was assembled based on existing compilers:
  - LLVM
  - TI GCC
  - AdaCore GNAT LLVM project - https://github.com/AdaCore/gnat-llvm

Custom Ada compiler

- msp430-elf-adac compilation steps:
  - ADB (Ada) to LLVM BC using GNAT LLVM
  - LLVM BC to ASM using LLC (part of LLVM)
  - ASM postprocessing (Ada initializers are put into .init_array)
  - ASM to binary using TI GCC
- Linking is performed using TI GCC and its linker scripts
- Required libraries (available both for MSP430 and MSP430X ISAs):
  - GNAT runtime
  - LLVM runtime (just a few functions missing from GCC runtime)
- One important limitation – large memory model is not supported

SUMMARY

Project achievements

- **TASTE**, an MBSE toolchain, can now target **MSP430**, a **low-power**, **affordable**, **hardened**, but also a **low-resource platform**
- Supporting a new platform was (relatively) easy thanks to the **Kazoo** templating engine
- The validation use-case, **DemoSat**, features:
  - mode management, thermal control, payload management
  - **PUS C** compliant interface
  - all in **64kB** of total (code + data) system memory, using an MBSE approach
- The implemented optimizations will benefit future projects
- TASTE is free, open-source and available now

Thank you for your attention

Rafał Babski
rbabski@n7space.com

Michał Kurowski
mkurowski@n7space.com

Konrad Grochowski
kgrochowski@n7space.com

+48 22 299 20 50
www.n7space.com