

# AQDT-GEN: AADL and QEMU-based Digital Twin generation for IoT testing

Mónica M. Villegas, Hernán Astudillo, Fáber Giraldo,  
[monica.villegas@sansano.usm.cl](mailto:monica.villegas@sansano.usm.cl), [hernan.astudillo@unab.cl](mailto:hernan.astudillo@unab.cl) , [fdgiraldo@uniquindio.edu.co](mailto:fdgiraldo@uniquindio.edu.co)  
ADEPT 2024



# Table of contents

## 1 Introduction

## 2 AQDT-GEN

## 3 Case Study

## 4 Questions?

AQDT-GEN

Villegas,  
Astudillo and  
Giraldo

Introduction

AQDT-GEN

Case Study

Questions?

## ① Introduction

## ② AQDT-GEN

## ③ Case Study

## ④ Questions?

# Introduction

- Digital twin = simulated and emulated environment that closely mimics the behavior of real-world Internet of Things (IoT) deployments
- Digital twins allow to build scenarios to analyze and test products before their physical implementation
- IoT digital twins help to identify potential defects in systems through an end-to-end implementation and specific analysis in each part of the system

AQDT-GEN

Villegas,  
Astudillo and  
Giraldo

Introduction

AQDT-GEN

Case Study

Questions?

# Introduction

- AQDT-GEN = AADL<sup>1</sup> and QEMU<sup>2</sup>-based Digital Twin GENerator
- Designed specifically to test IoT systems
- AQDT-GEN builds the Digital Twin of an IoT system using the configurations, parameters and specifications of its AADL model

---

<sup>1</sup><http://www.openaadl.org/>

<sup>2</sup><https://www.qemu.org/>

AQDT-GEN

Villegas,  
Astudillo and  
Giraldo

Introduction

AQDT-GEN

Case Study

Questions?

## 1 Introduction

## 2 AQDT-GEN

## 3 Case Study

## 4 Questions?

# Using AQDT-GEN

- ① Create the AADL model of the IoT system to be tested, specifying components, connections and properties
- ② Generate the required deployment files, parameters, and launching scripts, for both emulation and simulation, using the AADL model and the generation rules
- ③ Deploy the digital twin, by launching the emulation and simulation process

# AQDT-GEN Components

- ① Deployment files generator:** takes the AADL model of the embedded system, and generates the deployment files and codes
- ② Simulator/Emulator:** receives the generated files and codes, and triggers the emulation/simulation process

# AQDT-GEN QEMU Parameters

- The launch parameters of the QEMU emulator (OS, image path and name, kernel image, CPU type, and RAM size) are configured in the annex properties of the AADL model

# AQDT-GEN Files (1/2)

AQDT-GEN has four elements:

- **AQDTGENGUI.py**: GUI to guide the generation of the required files for the simulation
- **initsimulation**: script that starts the simulation inside the QEMU emulator
- **QemuCommand**: script generated by the sim-generator.py that modifies the Raspbian OS image and launches QEMU with the modified image
- **rpi\_stable.img**: image of Raspbian OS used for simulation

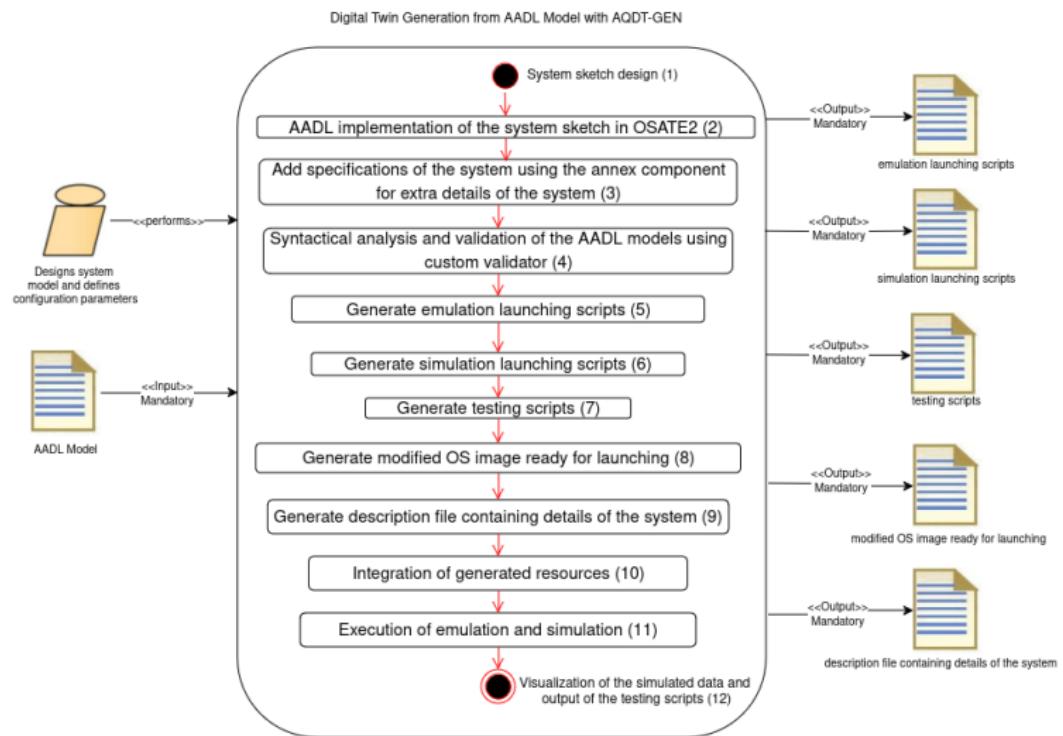
## AQDT-GEN Files (2/2)

- **rules.json**: configuration rules of the QEMU commands and AADL language
- **sim-generator.py**: Python program that takes the AADL model and `rules.json`, and generates `QemuCommand` and `simulation.conf`
- **simulation.conf**: configuration file that describes the sensors modeled in AADL
- **simulation.c**: program launched from `initsimulation` on the QEMU emulator, that uses the data from `simulation.conf` to simulate the sensors

# AQDT-GEN detailed execution

- ① Design system sketch
- ② Use OSATE2 to define AADL model
- ③ Add system specs in AADL model annex properties
- ④ Use custom validator to analyze the AADL model annex
- ⑤ Generate emulation launching scripts
- ⑥ Generate simulation launching scripts
- ⑦ Generate testing scripts
- ⑧ Generate modified OS image, ready for launching
- ⑨ Generate description file containing system details
- ⑩ Integrate generated resources
- ⑪ Execute emulation and simulation
- ⑫ Visualize the simulated data and testing output

# Steps for DT generation



[AQDT-GEN](#)

Villegas,  
Astudillo and  
Giraldo

[Introduction](#)[AQDT-GEN](#)[Case Study](#)[Questions?](#)

# Digital Twin Generation Tool

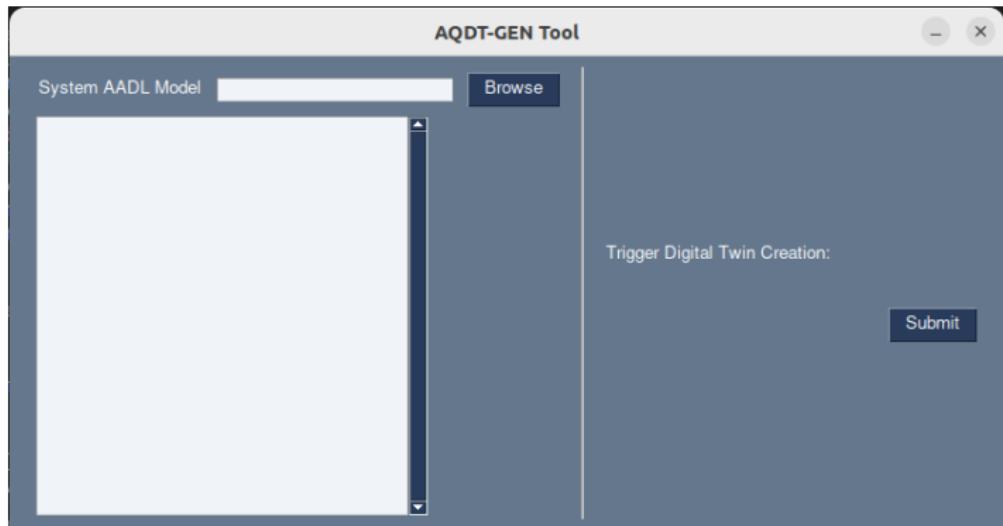


Figure 1: AQDT-GEN GUI

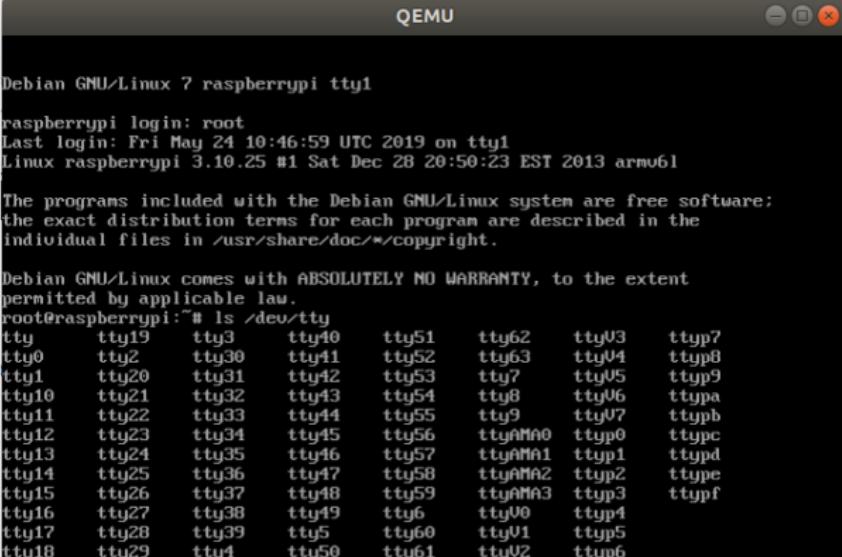
# Deployed Digital Twin

Introduction

AQDT-GEN

Case Study

Questions?



The terminal window shows the following text:

```
Debian GNU/Linux 7 raspberrypi tty1
raspberrypi login: root
Last login: Fri May 24 10:46:59 UTC 2019 on tty1
Linux raspberrypi 3.10.25 #1 Sat Dec 28 20:50:23 EST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@raspberrypi:~# ls /dev/ttys
tty     tty19   tty3    tty40   tty51   tty62   tty83   ttyp7
tty0    tty2    tty30   tty41   tty52   tty63   tty84   ttyp8
tty1    tty20   tty31   tty42   tty53   tty7    tty85   ttyp9
tty10   tty21   tty32   tty43   tty54   tty8    tty86   ttyna
tty11   tty22   tty33   tty44   tty55   tty9    tty87   ttypb
tty12   tty23   tty34   tty45   tty56   ttyAM0  ttyp0   ttypc
tty13   tty24   tty35   tty46   tty57   ttyAMA1 ttyp1   ttypd
tty14   tty25   tty36   tty47   tty58   ttyAMA2 ttyp2   ttype
tty15   tty26   tty37   tty48   tty59   ttyAMA3 ttyp3   ttypf
tty16   tty27   tty38   tty49   tty6    tty80   ttyp4
tty17   tty28   tty39   tty5    tty60   tty81   ttyp5
tty18   tty29   tty4    tty50   tty61   tty82   ttyp6
```

Figure 2: DT deployment

AQDT-GEN

Villegas,  
Astudillo and  
Giraldo

Introduction

AQDT-GEN

Case Study

Questions?

## 1 Introduction

## 2 AQDT-GEN

## 3 Case Study

## 4 Questions?

# Case study - Weather station

- Data output format:  
c000s000g000t086r000p000h53b10020
- c000: air direction in degrees
- s000: air speed in miles per hour
- g000: air speed in miles per hour
- t086: temperature in Fahrenheit
- r000: rainfall of 1 hour in inches
- p000: rainfall of 24 hours in inches
- h53: about humidity value
- b10020: atmosphere in hPa

[AQDT-GEN](#)[Villegas,  
Astudillo and  
Giraldo](#)[Introduction](#)[AQDT-GEN](#)[Case Study](#)[Questions?](#)

# First weather station

```
device weatherStation1           -- MODULE 1
    features
        data_pk_out1:out data port;
    properties
        Period=>2000ms;
    annex extras {** MAC = "FF0211"; mean = "20"; deviation = "20" **};
end weatherStation1;
```

Figure 3: AADL description of the first weather station

[AQDT-GEN](#)[Villegas,  
Astudillo and  
Giraldo](#)[Introduction](#)[AQDT-GEN](#)[Case Study](#)[Questions?](#)

# Last weather station

```
-----  
device weatherStation15           -- MODULE 15  
  features  
    data_pk_out1:out data port;  
  properties  
    Period=>1000ms;  
  annex extras {** MAC = "FF02F1"; mean = "2"; deviation = "20" **};  
end weatherStation15;  
-----
```

Figure 4: AADL description of the last weather station

# Serial receiver

```
-----  
device SerialRadio1  
  features  
    data_pk_sensor1:in data port;  
    data_pk_sensor2:in data port;  
    data_pk_sensor3:in data port;  
    data_pk_sensor4:in data port;  
    data_pk_sensor5:in data port;  
    data_pk_sensor6:in data port;  
    data_pk_sensor7:in data port;  
    data_pk_sensor8:in data port;  
    data_pk_sensor9:in data port;  
    data_pk_sensor10:in data port;  
    data_pk_sensor11:in data port;  
    data_pk_sensor12:in data port;  
    data_pk_sensor13:in data port;  
    data_pk_sensor14:in data port;  
    data_pk_sensor15:in data port;  
    bus_serial1:requires bus access SerialConnection1;  
end SerialRadio1;  
  
-----  
bus SerialConnection1 extends Buses::UART::UART  
end SerialConnection1;
```

Figure 5: AADL description of the serial receiver

[AQDT-GEN](#)[Villegas,  
Astudillo and  
Giraldo](#)[Introduction](#)[AQDT-GEN](#)[Case Study](#)[Questions?](#)

# Serial receiver

```
root@raspberrypi:~# ls /dev/ttys*
/dev/ttys0  /dev/ttys1
```

Figure 6: Serial interface of the simulated serial receiver

# Weather stations output

```
root@raspberrypi:~ cat /dev/tty00
FF02D1c67s59g68t62r63p16h94b3
FF02C1c7s82g25t35r13p98h76b81
FF0261c88s2g43t47r17p65h99b1
FF02E1c28s79g72t89r16p29h74b74
FF0291c79s50g55t76r79p75h87b48
FF02A1c16s83g5t57r86p49h4b9
FF02B1c14s3g62t79r39p70h61b65
FF0281c15s34g56t94r36p64h22b16
FF0231c91s9g64t80r11p32h16b69
FF0241c48s25g5t57r52p20h91b9
FF0271c18s93g15t32r49p78h12b88
FF02F1c25s89g96t16r51p12h48b14
FF0251c44s64g83t34r18p53h79b34
FF02C1c95s13g41t62r93p66h3b89
FF02D1c12s50g17t12r27p22h69b80
FF0261c10s63g51t48r91p15h50b60
FF02E1c53s82g73t97r47p56h83b17
FF0291c35s1g9t30r66p50h44b59
FF02B1c68s99g0t2r6p53h37b79
FF02A1c87s86g53t49r1p57h49b44
FF0281c24s51g5t4r81p92h57b90
FF0211c93s18g82t13r17p34h16b23
FF0231c39s5g54t41r41p94h79b94
FF0221c51s45g77t76r48p34h32b29
FF0241c26s89g72t0r12p17h18b46
FF0271c80s98g59t19p3p65h12b44
FF02F1c59s44g39t7r76p42h52b5
```

Figure 7: Output of weather stations connected to serial receiver

AQDT-GEN

Villegas,  
Astudillo and  
Giraldo

Introduction

AQDT-GEN

Case Study

Questions?

## 1 Introduction

## 2 AQDT-GEN

## 3 Case Study

## 4 Questions?

AQDT-GEN

Villegas,  
Astudillo and  
Giraldo

Introduction

AQDT-GEN

Case Study

Questions?



## Contact us

**Mónica M. Villegas, Hernán Astudillo, Fáber Giraldo**  
monica.villegas@sansano.usm.cl, hernan.astudillo@unab.cl , fdgiraldo@uniquindio.edu.co  
Valparaíso, Chile