Trustworthy Platform Module

An attempt to create open-source firmware for TPM

FOSDEM 2023

Maciej Pijanowski

3MDEB

Maciej Pijanowski
Engineering Manager

- @macpijan
- maciej.pijanowski@3mdeb.com
- linkedin.com/in/maciej-pijanowski-9868ab120

- 7 years in 3mdeb
- Open-source contributor
- Interested in:
  - build systems (e.g. Yocto)
  - embedded, OSS, OSF
  - firmware/OS security

Who we are ?

- Poland-based company, over 7 years in the market
- Open-source firmware, Embedded Linux
- coreboot licensed service providers since 2016 and leadership participants
- UEFI Adopters since 2018
- Yocto Participants and Embedded Linux experts since 2019
- Official consultants for Linux Foundation fwupd/LVFS project since 2020

Agenda

- What is TwPM project?
- Why did it start?
- TPM modules pinouts
- How to start such a project?
- Expected challenges
- Roadmap
- Current state
- Q&A

What is TwPM?

Trustworthy vs Trusted

TwPM project aims to increase the trustworthiness of the TPM module (hence the TwPM), by providing the open-source firmware implementation for the TPM device, compliant* to the TCG PC Client Specification.

TPM modules enable measured boot and support verified boot, Dynamic Root of Trust for Measurement, and other security features.

The project is funded through the NGI Assure Fund, a fund established by NLnet foundation.

https://nlnet.nl/project/TwPM

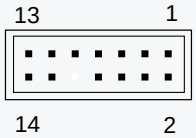
NGI ASSURE nlnet FOUNDATION

https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/

Why did it start?

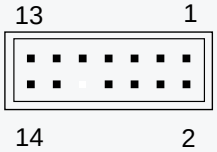
- Traditional TPMs are dedicated microcontrollers with proprietary firmware
  - can't be audited
  - bugs can't be fixed if TPM vendor doesn't care
  - capabilities of TPM are defined by the vendor and can't be modified by user (e.g. to include newer hash algorithms)
- Different interfaces
  - LPC (older PCs, still commonly used)
  - SPI (new PCs, mobile, IoT)
  - I2C (mobile, IoT)
- Each mainboard vendor has different pinout for module
  - some look the same and mechanically can be installed to incompatible boards, but their electrical connections are different
  - doing so may **physically damage your mainboard**

Examples of TPM pinouts



Pin No.	Definition	Pin No.	Definition
1	LPC Clock	2	3V Standby Pwr
3	LPC Reset	4	3.3V Power
5	LPC address & data pin 0	6	Serial IRQ
7	LPC address & data pin 1	8	5V Power
9	LPC address & data pin 2	10	No Pin
11	LPC address & data pin 3	12	Ground
13	LPC Frame	14	Ground

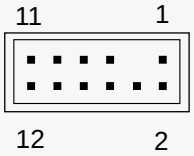
MSI - B75MA-E33 (14-1)



Pin No.	Definition	Pin No.	Definition
1	F_LAD0	2	+3V
3	F_LAD1	4	+3V
5	F_LAD2	6	C_PCICLK_TPM
7	F_LAD3	8	GND
9	F_FRAME#	10	No Pin
11	F_SERIRQ	12	S_PCIRST#_TBD
13	F_CLKRUN	14	+3VSB

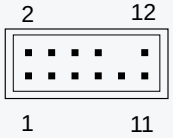
Asus - MAXIMUS IX FORMULA (14-1)

Examples of TPM pinouts



Pin No.	Definition	Pin No.	Definition
1	Data output	2	Power 3.3V
3	No Pin	4	NC
5	Data Input	6	CLK
7	Chip Select	8	GND
9	IRQ	10	NC
11	NC	12	RST

GIGABYTE Z590 AORUS MASTER (12-1)



Pin No.	Definition	Pin No.	Definition
1	SPI Power	2	SPI Chip Select
3	Master In Slave Out (SPI Data)	4	Maste Out Slave In (SPI Data)
5	Reserved	6	SPI Clock
7	Ground	8	SPI Reset
9	Reserved	10	No Pin
11	Reserved	12	Interrupt request

MSI - Z590 PRO WIFI (12-1)

Examples of TPM pinouts

19 1
20 2

Supermicro MBD M12SWA-TF-O Server Motherboard

Pin No.	Definition	Pin No.	Definition
1	LCLK	2	GND
3	LFRAME#	4	No Pin
5	LRESET#	6	NC
7	LAD3	8	LAD2
9	3.3V	10	LAD1
11	LAD0	12	GND
13	NC	14	NC
15	3.3V_STBY	16	SERIRQ
17	GND	18	NC
19	NC	20	NC

19 1
20 2

Supermicro X10DAL-I Server MTB

Pin No.	Definition	Pin No.	Definition
1	LCLK	2	GND
3	LFRAME#	4	<(KEY)>
5	LRESET#	6	+5V (X)
7	LAD3	8	LAD2
9	3.3V	10	LAD1
11	LAD0	12	GND
13	SMB_CLK4	14	SMB_DAT4
15	3.3V_DUAL	16	SERIRQ
17	GND	18	CLKRUN# (X)
19	LPCPD#	20	LDRQ# (X)

19 1
20 2

Gigabyte - GA-970A-UD3P (20-1)

Pin No.	Definition	Pin No.	Definition
1	LCLK	2	GND
3	LFRAME	4	No Pin
5	LRESET	6	NC
7	LAD3	8	LAD2
9	VCC3	10	LAD1
11	LAD0	12	GND
13	NC	14	ID
15	SB3V	16	SERIRQ
17	GND	18	NC
19	NC	20	SUSCLK

1

Asrock H170M PRO4 (18-1)

Pin No.	Definition	Pin No.	Definition
1	PCICLK	2	GND
3	FRAME	4	SMB_CLK_MAIN
5	PCIRST #	6	SMB_DATA_MAIN
7	LAD3	8	LAD2
9	+3V	10	LAD1
11	LAD0	12	GND
13	No Pin	14	S_PWRDWN#
15	+3VSB	16	SERIRQ#
17	GND	18	GND

19 1
20 2

Asus - M5A99FX pro r2.0 (20-1)

Pin No.	Definition	Pin No.	Definition
1	PCICLK	2	GND
3	LFRAME	4	No Pin
5	PCIRST#	6	NC
7	LAD3	8	LAD2
9	+3V	10	LAD1
11	LAD0	12	GND
13	NC	14	NC
15	+3VSB	16	SERIRQ
17	GND	18	CLKRUN
19	PWRDWN	20	NC

19 1
20 2

Asus - MAXIMUS VII HERO (20-1)

Pin No.	Definition	Pin No.	Definition
1	PCICLK	2	GND
3	LFRAME	4	No Pin
5	PCIRST#	6	NC
7	LAD3	8	LAD2
9	+3V	10	LAD1
11	LAD0	12	GND
13	NC	14	NC
15	+3VSB	16	SERIRQ
17	GND	18	CLKRUN
19	PWRDWN	20	NC

13 1
14 2

MSI - B85M-E45 (14-1)

Pin No.	Definition	Pin No.	Definition
1	LPC Clock	2	3V Standby Pwr
3	LPC Reset	4	3.3V Power
5	LPC address & data pin 0	6	Serial IRQ
7	LPC address & data pin 1	8	5V Power
9	LPC address & data pin 2	10	No Pin
11	LPC address & data pin 3	12	Ground
13	LPC Frame	14	Ground

11 1
12 2

GIGABYTE TPM Header pinout (12-1)

Pin No.	Definition	Pin No.	Definition
1	LAD0	2	VCC3
3	LAD1	4	No Pin
5	LAD2	6	LCLK
7	LAD3	8	GND
9	LFRAME	10	NC
11	SERIRQ	12	LRESET

13 1
14 2

Asus - B550 PLUS (14-1)

Pin No.	Definition	Pin No.	Definition
1	VCC SPI	2	S_SPI_TPM_IRQ#
3	S_PLTSRT#	4	S_SPI_TPM_CS2#
5	F2_SPI_CS1#_R	6	F_BIOS_WP#_R
7	+3V_SPI	8	GND
9	F_SPI_CS0#_R	10	T_SPI_CLK
11	T_SPI_MISO	12	T_SPI_MOSI
13	F_SPI_HOLD#_R	14	No Pin

13 1
14 2

Asus - TPM-M R2.0 (14-1)

Pin No.	Definition	Pin No.	Definition
1	F_LAD0	2	+3V
3	F_LAD1	4	+3V
5	F_LAD2	6	C_PCICLK_TPM
7	F_LAD3	8	GND
9	F_FRAME#	10	No Pin
11	F_SERIRQ	12	S_PCIRST#_TBD
13	LPCPD#	14	+3VSB

13
14 2

Asus ROG STRIX X570-F GAMING SPI TPM

Pin No.	Definition	Pin No.	Definition
1	+1.8V TPM	2	SPI TPM IRQ#
3	PLTRST#	4	SPI TPM CS#
5	-	6	SPI FLASH WP#
7	+18V SPI	8	GND
9	SPI CS#	10	SPI CLK
11	SPI MISO	12	SPI MOSI
13	SPI HOLD#	14	No Pin

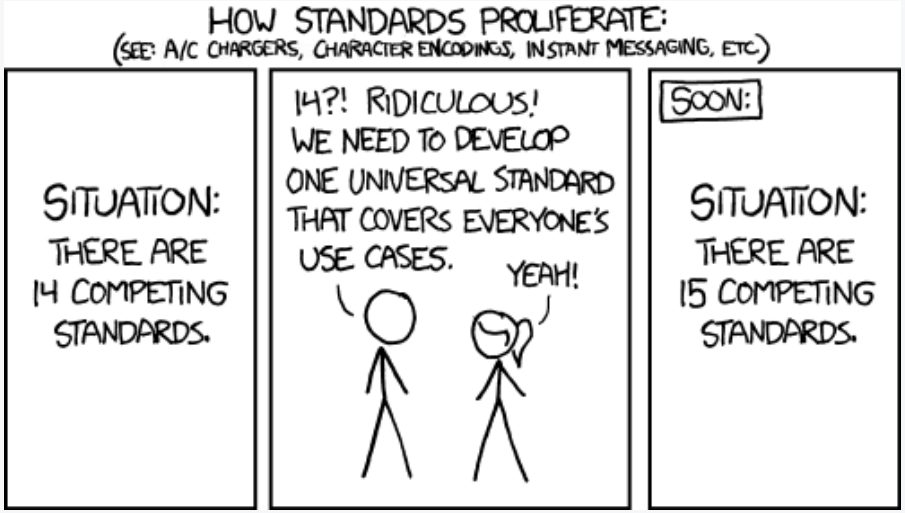
17 1
18 2

AsRock Rack X470-D4U Server Motherboard
AsRock Rack ROMED8-2T Server Motherboard
AsRock Rack EPC621D8A Server Motherboard

Pin No.	Definition	Pin No.	Definition
1	CK_33M_TPM	2	GND
3	LFRAME#_L	4	SMB_CLK_MAIN
5	TPM_RST#	6	SMB_DATA_MAIN
7	LAD3_L	8	LAD2_L
9	+3V	10	LAD1_L
11	LAD0_L	12	GND
13	No Pin	14	S_PWRDWN#
15	+3VSB	16	SERIRQ#
17	GND	18	F_CLKRUN#

... and many more!

Examples of TPM pinouts



- At first we wanted to choose most commonly used connector and use it with custom board
  - variety is bigger than anticipated
  - connector would have to support different interfaces (LPC, SPI)
- We may aim to tackle the hardware problem in the future
  - firmware is enough of a challenge for a start

Image source: https://xkcd.com/927/

How to start?

- There are **a few** open-source stacks for processing TPM commands
- ms-tpm-20-ref
  - https://github.com/microsoft/ms-tpm-20-ref
  - implementation from Microsoft
  - simulator for Windows / Linux / MacOS
  - some others
    - fTPM Trusted Application for ARM Trust Zone
    - samples for STM32 **Nucleo L476RG / L4A6RG**
- ibmswtpm2
  - https://sourceforge.net/projects/ibmswtpm2/
  - implementation from IBM
  - simulator

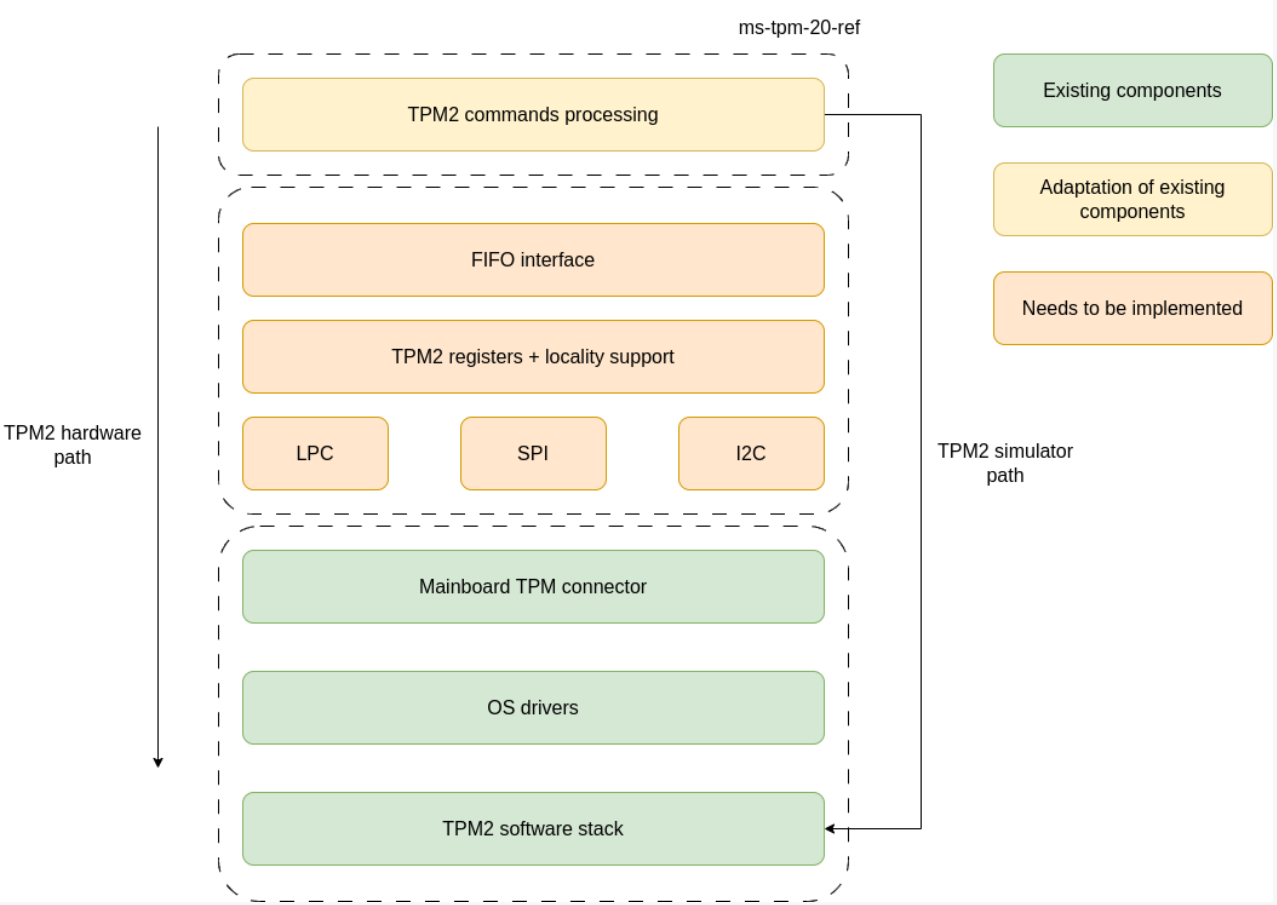
Nucleo sample review

- Code for Nucleo samples was contributed 4 years ago
- It was created using the Atollic TRUEStudio for STM32
  - such software no longer exists
  - STM32CubeIDE has replaced it
- Code under directory was contributed at some point in the past
  - it is not maintaned - it may or may not work
  - https://github.com/microsoft/ms-tpm-20-ref/issues/62

Closer look at the Nucleo sample

- We have converted the project into STM32CubeIDE
- We were able to build it after some modifications
- There is some VCOM application for Windows
  - https://github.com/microsoft/ms-tpm-20-ref/tree/main/Samples/Nucleo-TPM/VCOM
  - it was used for testing this sample code
- The STM32 code can accept TPM command via USB CDC
  - it can process it
  - it can return response
- There is some custom protocol involved there
  - no interperability with existing tools, such as tpm2-tools
  - no interoperability with existing TPM interfaces (e.g. SPI, LPC, I2C)
- The STM32 was low on resources when running this code

High-level overview



Challenges

- In the meantime, the global shortage of chips happened
  - even if we wanted to use STM32L4, they were not available
  - the other chips, were also at low-availability
- It was difficult to asses the precise hardware requirements at this point
- The application was using HAL, so switching hardware requires rewrite
- It would be nice to use some OS to switch between boards more easily
- We chose Zephyr as OS for TwPM
  - all-rounded RTOS with decent portability between smaller devices

Zephyr™

Challenges

- Some TPM registers must return valid state without delay
  - may be hard or impossible to reply on time if it has to pass through interface FIFO and possibly kernel/userspace boundary
  - FPGA would help with returning register values on time, but we're experimenting with other, cheaper options
- FPGA will be required for LPC protocol - LPC isn't supported by most MCUs
- Reference implementation [1] doesn't implement NV RAM in a secure way
  - only TPM emulator is officially supported, it doesn't have physical flash so no protections implemented
  - wear leveling also has to be considered
- Full compliance with TPM specification may be impossible
  - strict initialization time and power consumption requirements
  - no vendor ID assigned

1: https://github.com/microsoft/ms-tpm-20-ref

Roadmap and current state

1. Public site with documentation:

https://twpm.dasharo.com/

Trustworthy Platform Module (TwPM) | Search | GitHub 2 0

Roadmap

TwPM project is funded by the NLnet Foundation via the NGI ASSURE.

3mdeb has proposed to implement the following tasks under the grant agreement No 957073.

1. Public site with documentation

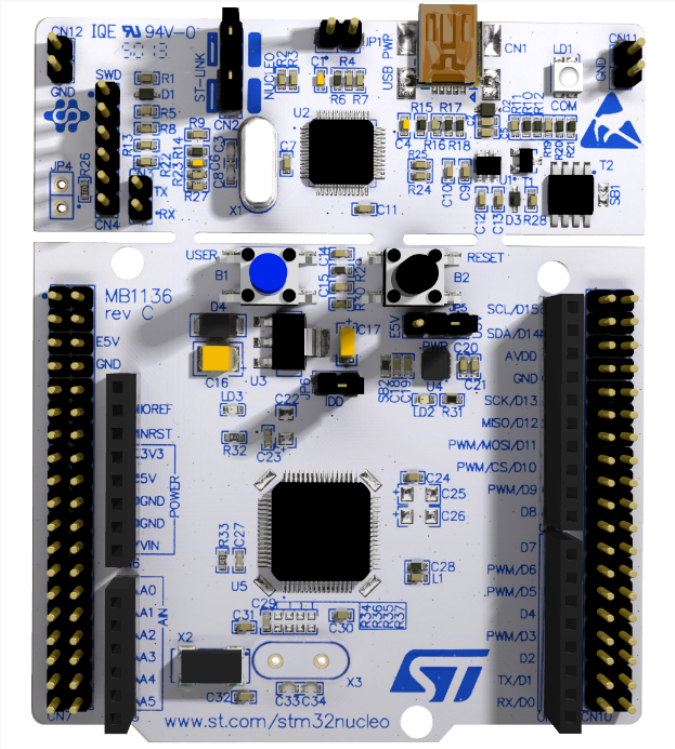
All of the documentation produced during this project should be publicly available to users and developers.

Milestones:

- create repositories for the project
- prepare server and domain name
- describe project's purpose and phases
- create placeholders for description of deviations from TPM specification, software stack and documentation changelog that will be updated after each of the remaining phases

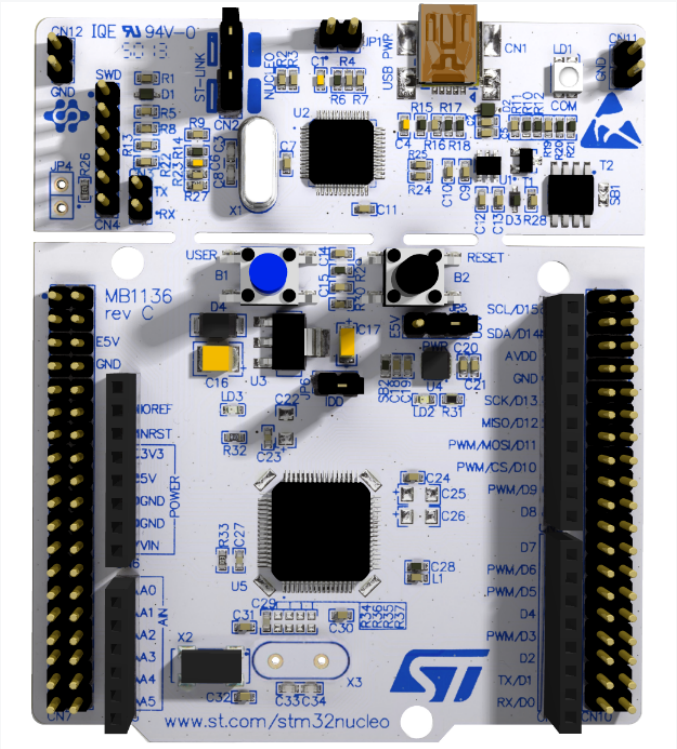
Roadmap and current state

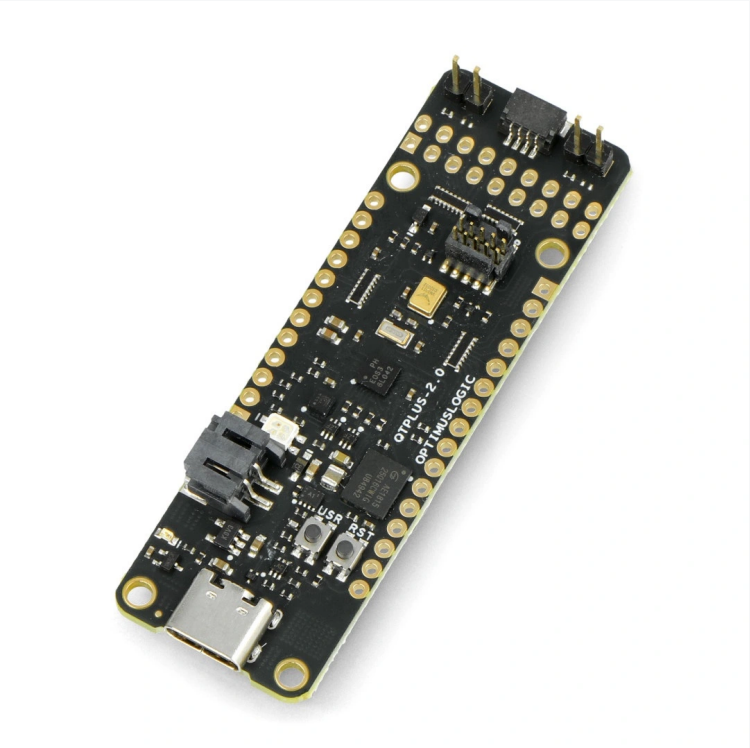
2. Gather hardware requirements and choose target board



Roadmap and current state

2. Gather hardware requirements and choose target board





Roadmap and current state

3. Implement LPC protocol in FPGA

- currently in progress
- targeted for PC only

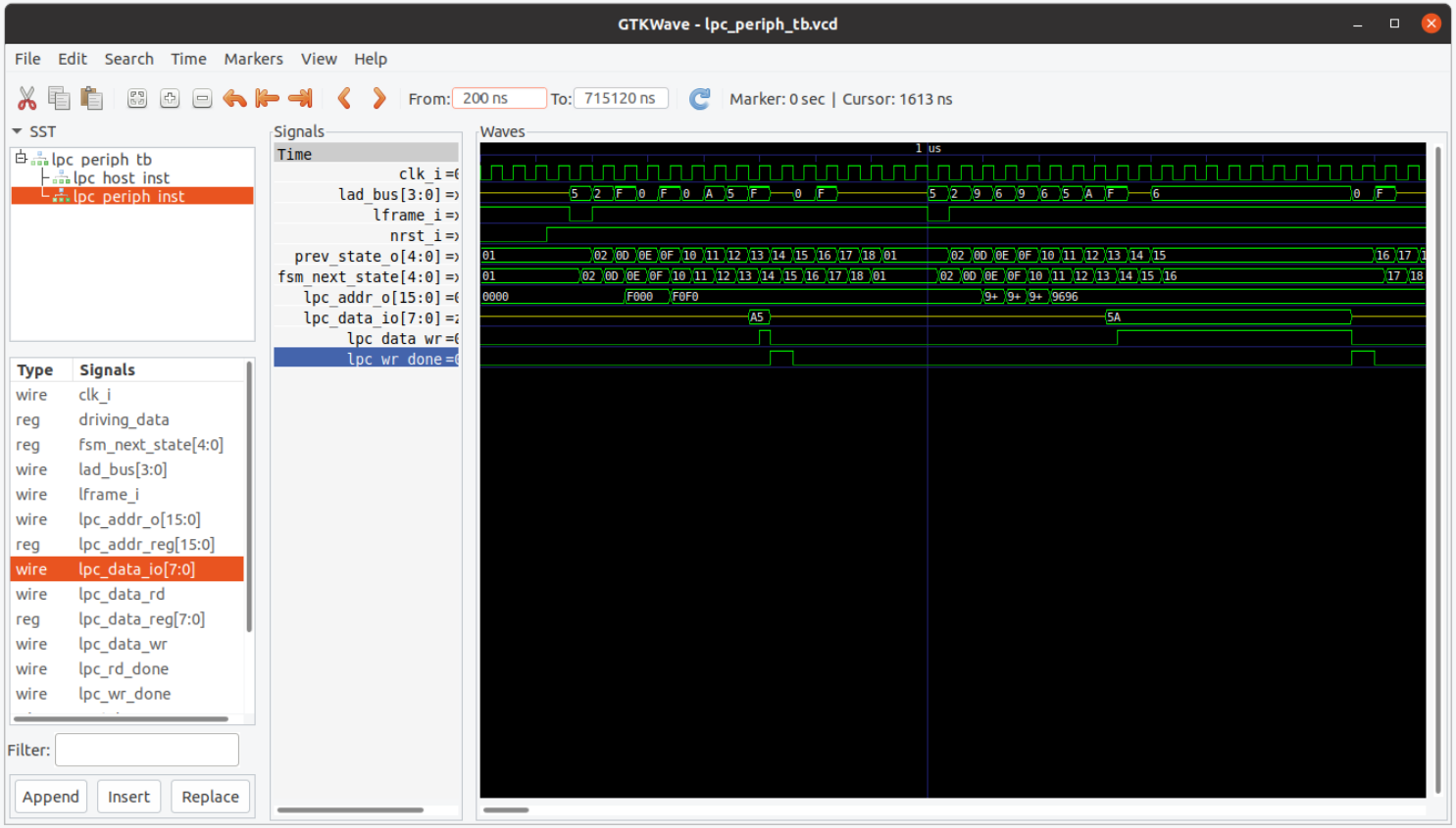
4. Implement basic TPM registers in FPGA

- some registers must be readable without delay
- implementing TPM locality state machine in FPGA significantly simplifies interface between FPGA and MCU

5. Implement TPM command parsing and communication between FPGA and MCU

- completing this step will produce first semi-usable version with limited capabilities
- applications requiring persistent storage like sealing data to PCR values or persistent key creation won't be possible yet

Roadmap and current state



Roadmap and current state

6. Base tests

- test suites mostly documented: https://github.com/Dasharo/docs/pull/447
- automation in progress
- tests results will be added to documentation

7. Implement SPI TPM protocol

- repetition of steps from previous slide for another interface
- it is likely, that again FPGA might be required to meet timing requirements

8. Explore the usage of using simpler hardware platform

- it may or may not be possible to use board without FPGA (usually cheaper)
- potential benefits make exploration worthwhile
- already happens in parallel to other tasks

Roadmap and current state

9. Flash driver for TPM stack

- nonvolatile storage for user- and vendor-defined data
- will open the way for additional use cases
- more test suites
- protections are required for compliance with specification, but at this point we put it in nice-to-have category

10. Unique identification and randomness source

- implementation may depend on chosen hardware, hence left for later
- uniqueness required for primary seeds, used to generate primary keys
- primary seed is required to have at least twice the number of bits as the security strength of any symmetric or asymmetric algorithm implemented on the TPM
- TPM should have at least one internal source of entropy
- FPGA can be used if everything else fails

Roadmap and current state

11. Manufacturing process

- each TPM has unique Endorsement Key (EK)
- vendor issues certificate for EK that should be committed to NVRAM
- this step will describe the process in detail and try to automate it

12. Customizable configuration

- prepare easy to use build system integrating whole stack
- build-time configuration including:
    - interface (SPI or LPC)
    - hash algorithms supported by TPM
    - amount of NVRAM
    - whether to include RNG entropy source from FPGA or not
- goal: making transition between different boards easier

Where to find out more

Work currently in progress:

- https://github.com/3mdeb/verilog-lpc-module
  - LPC module
  - TPM registers probably will also be implemented in this repo
- https://github.com/3mdeb/zephyr
  - exploration and abusing of SPI drivers takes place here
- https://github.com/Dasharo/twpm-docs
  - source for https://twpm.dasharo.com
  - will be progressively filled with results of each step

Interested about further development? Want to participate in the project? Join TwPM channel in Dasharo Matrix space:

Where to find out more

Interested about further development? Want to participate in the project? Join TwPM channel in Dasharo Matrix space:

https://matrix.to/#/#twpm:matrix.org

We need you

Want to join our team and work with open-source firmware on a daily basis?

- use contact links from next slide
- approach me directly

Contact us

- Join Dasharo Matrix space: https://matrix.to/#/#dasharo:matrix.org
- contact@3mdeb.com
- facebook.com/3mdeb
- @3mdeb_com
- https://fosstodon.org/@3mdeb
- linkedin.com/company/3mdeb
- https://3mdeb.com
- Book a call
- Sign up for the newsletter

Q&A