

XFS4IoT SP-Dev Workgroup

5 April 2022

- Recap from previous meeting
- Framework migration to .Net6
- Cash acceptor release
- Cash acceptor demo (KAL)
- Framework release 1.0.0
- TPM framework

- Frameworks are now completed to support a complete Cash Out ATM.
- All now available with C# and C++ sample code released and demos on YouTube.
 - ☑ Card Reader with support for dispensing
 - ☑ Cash Dispenser without end-to-end security
 - ☑ Text Terminal Unit
 - ☑ EPP Key Management and Crypto classes
 - ☑ Keyboard and PinPad classes
 - ☑ End-to-end security partially complete
 - ☑ Printer / Guide lights
 - ☑ Vendor Mode and Vendor Application
 - ☑ Auxiliaries
 - ☑ Barcode reader (Feb-2022)
 - ☑ Biometrics - All biometric data types supported (Feb-2022)

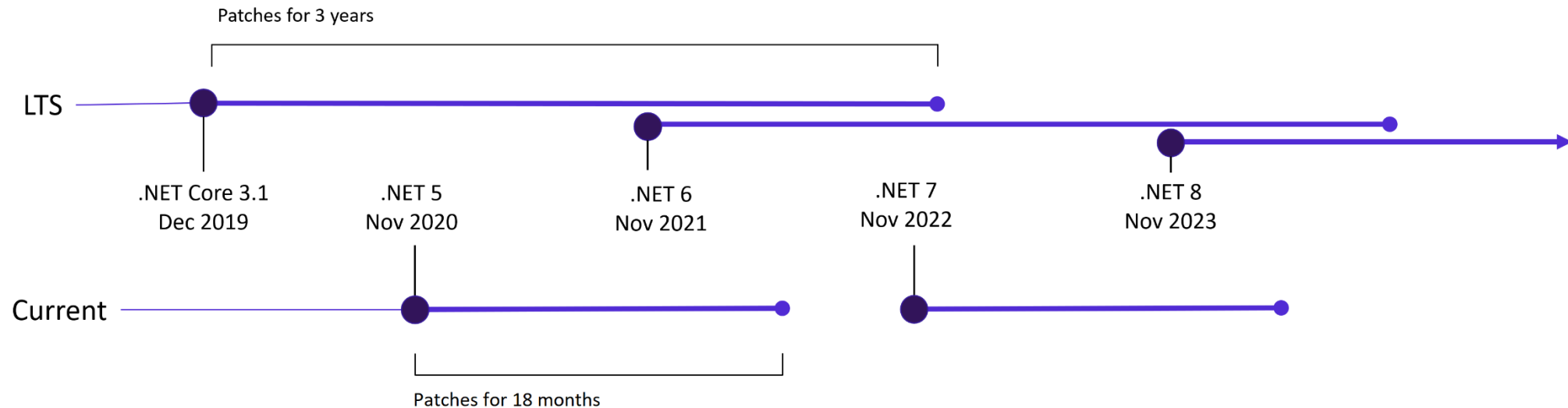
.NET 6 for SP-DEV framework

- XFS4IoT SP-Dev framework is built on .NET, written in C#
- Initial code was based on .NET 5 – March 2021
- Latest version has been updated to .NET 6

Microsoft .NET lifecycle

- Every other version of .NET is “long term support” – three years
- Other versions supported for 18 months

	End of life
.NET core 3.1	December 2022
.NET 5	May 2022
.NET 6	November 2024



[.NET and .NET Core official support policy \(microsoft.com\)](https://microsoft.com/.NET)

- Different .NET versions support new C# language versions
 - .NET 5 C# 9
 - .NET 6 C# 10
 - Record structs
 - Improvements of structure types
 - Interpolated string handlers
 - global using directives
 - File-scoped namespace declaration
 - Extended property patterns
 - Improvements on lambda expressions
 - Allow const interpolated strings
 - Record types can seal ToString()
 - Improved definite assignment
 - Allow both assignment and declaration in the same deconstruction
 - Allow AsyncMethodBuilder attribute on methods
 - CallerArgumentExpression attribute
 - Enhanced #line pragma

[The history of C# - C# Guide | Microsoft Docs](#)

- More than 500 performance improvements
 - Execution Speed and optimisation
 - Memory use
 - Start up speed
 - File performance

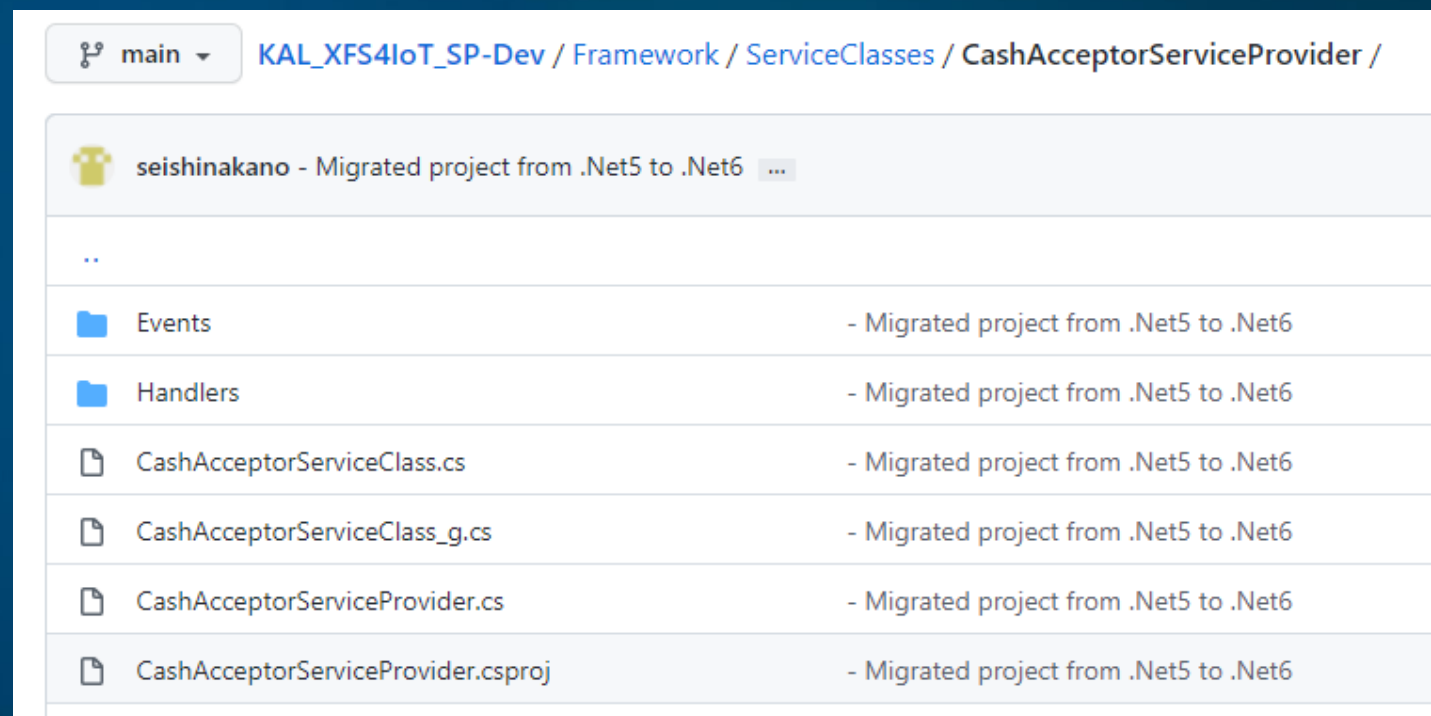
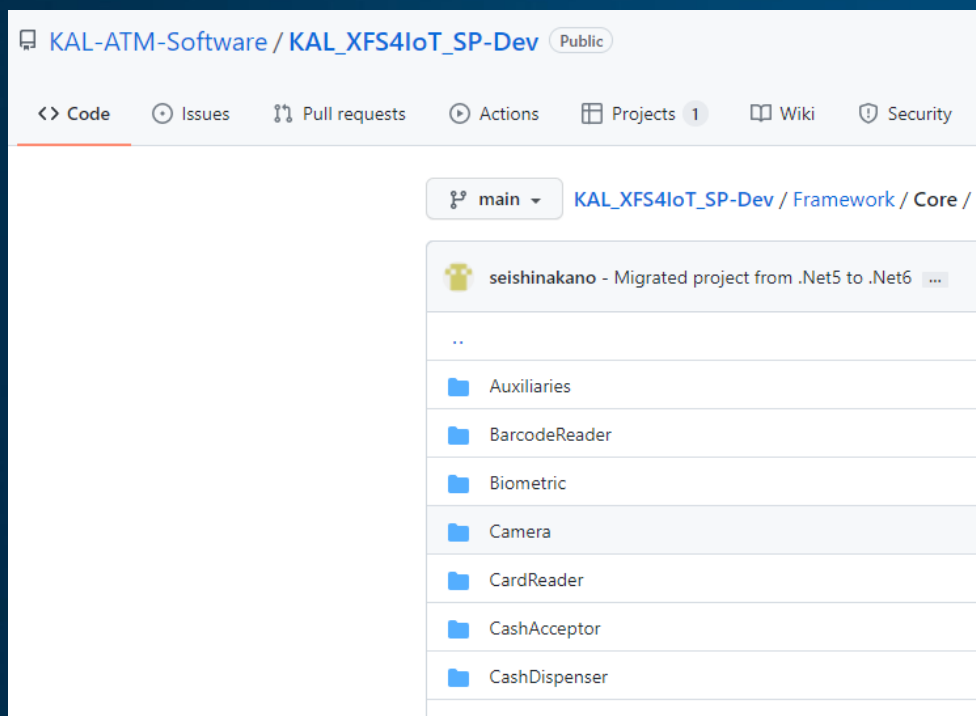
[Performance Improvements in .NET 6 - .NET Blog](#)

Some possible things to experiment with (No guarantees!)

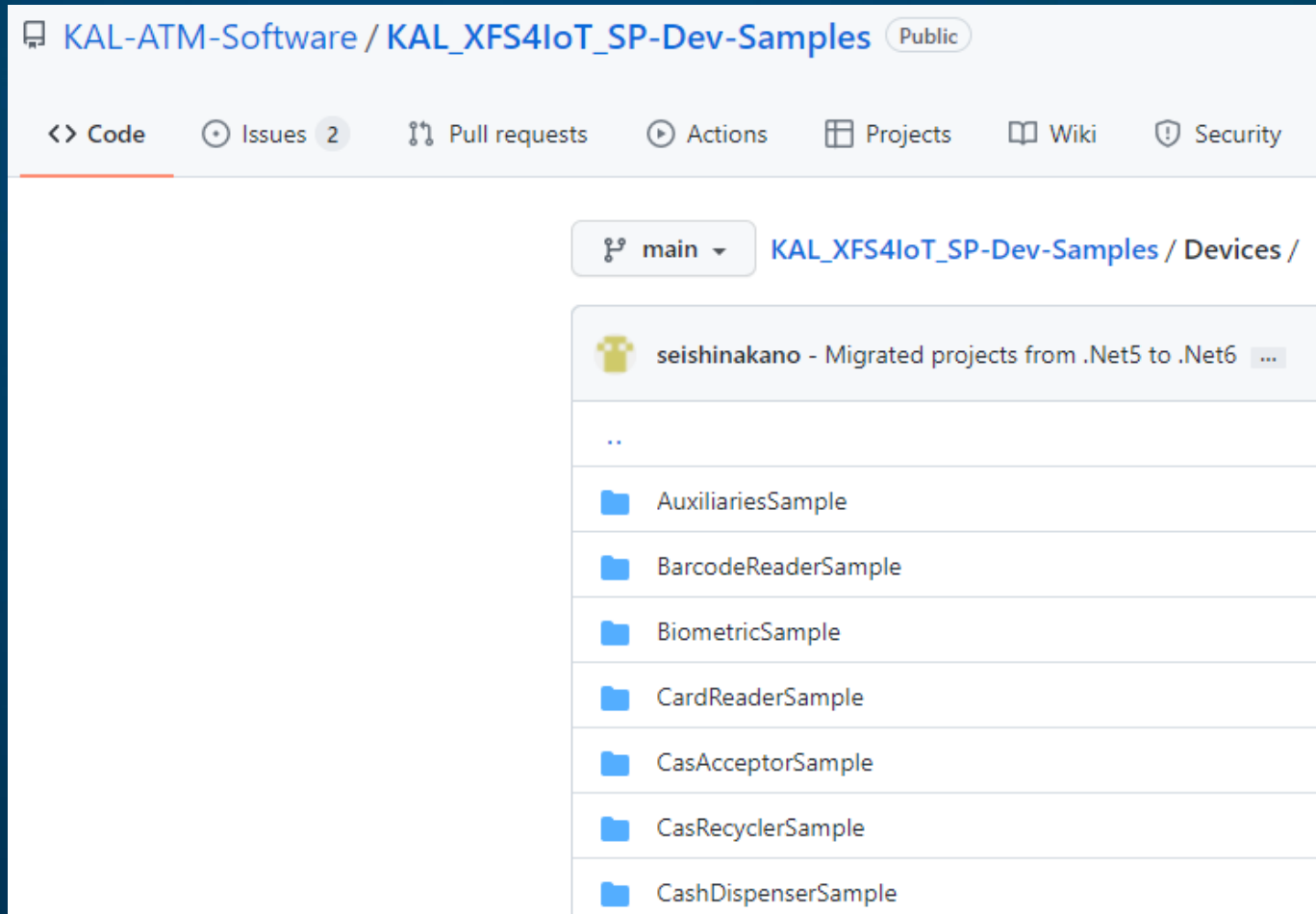
- Source generators for JSON. Smaller, faster handling of messages
- Ahead-of-time compilation – native code without JIT

Cash acceptor class release

- Framework class now available:



- Acceptor and recycler samples both available



- CashAcceptor class supports all methods required to:
 - Accept cash (*CashInStart, CashIn, CashInEnd*)
 - Return and present cash (*CashInRollback, PreparePresent, PresentMedia*)
 - Get cash status and counters (*GetCashInStatus, CashUnitCount*)
 - Manage counterfeit/suspicious notes (*CreateSignature, CompareSignature*)
 - Report various events such as cash inserted, items refused and more...
- All this now enables support for accepting and recycling capabilities in the framework

Demos with real devices

All demo videos with real devices will be available on YouTube.

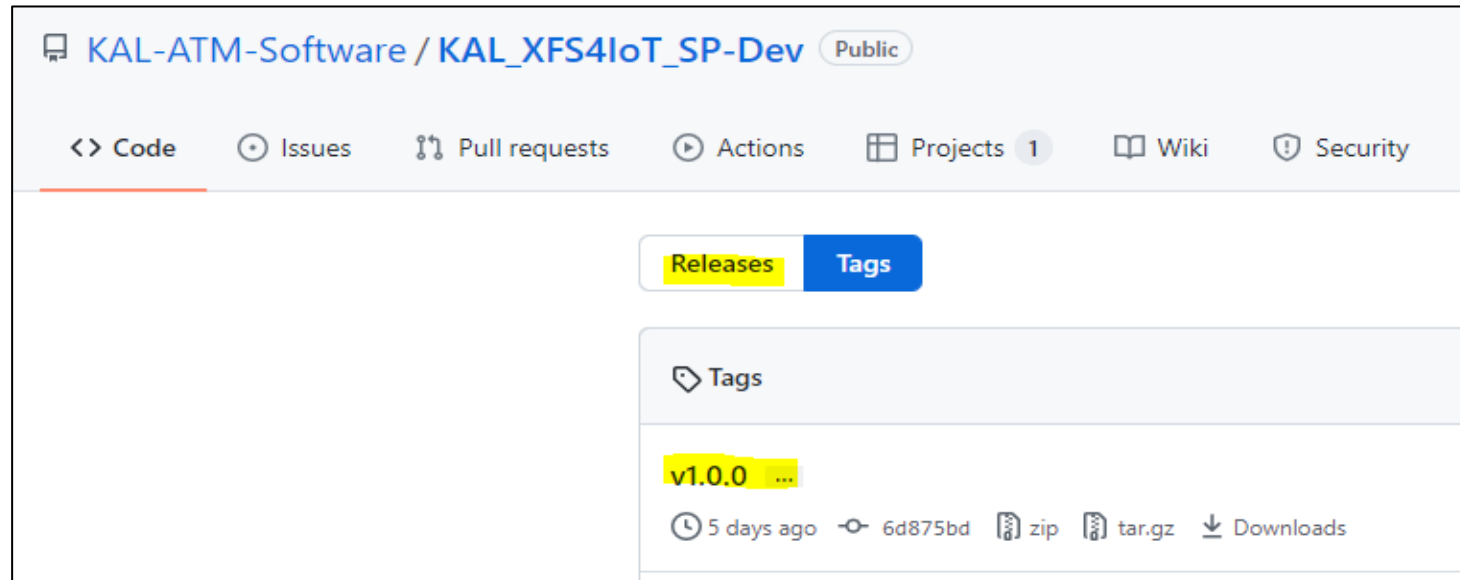
*All previous demo videos can be found on the KAL ATM Software
YouTube channel:*

<https://www.youtube.com/user/ATMsoftware/videos>



Framework Release 1.0.0

- Framework now supports all devices for CashOut, CashIn and Recycler devices
- Time for version 1.0.0




- No more pre-release
 - NuGet packages updated.
- All projects should be using
Release v1.00
- Published in NuGet.org

Releases / v1.0.0

v1.0.0













Latest

 github-actions released this 5 days ago · 1 commit to main sinc

Merge branch 'main' of https://github.com/KAL-ATM-Software/KAL_XFS

...T_SP-Dev

▼ Assets 47

-  [KAL.XFS4IoT.SP-Dev.Framework.Auxiliaries.1.0.0.nupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.Auxiliaries.1.0.0.snupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.BarcodeReader.1.0.0.nupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.BarcodeReader.1.0.0.snupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.Biometric.1.0.0.nupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.Biometric.1.0.0.snupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.Camera.1.0.0.nupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.Camera.1.0.0.snupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.CardReader.1.0.0.nupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.CardReader.1.0.0.snupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.CashAcceptor.1.0.0.nupkg](#)
-  [KAL.XFS4IoT.SP-Dev.Framework.CashAcceptor.1.0.0.snupkg](#)

TPM Integration – XFS4IoT Framework

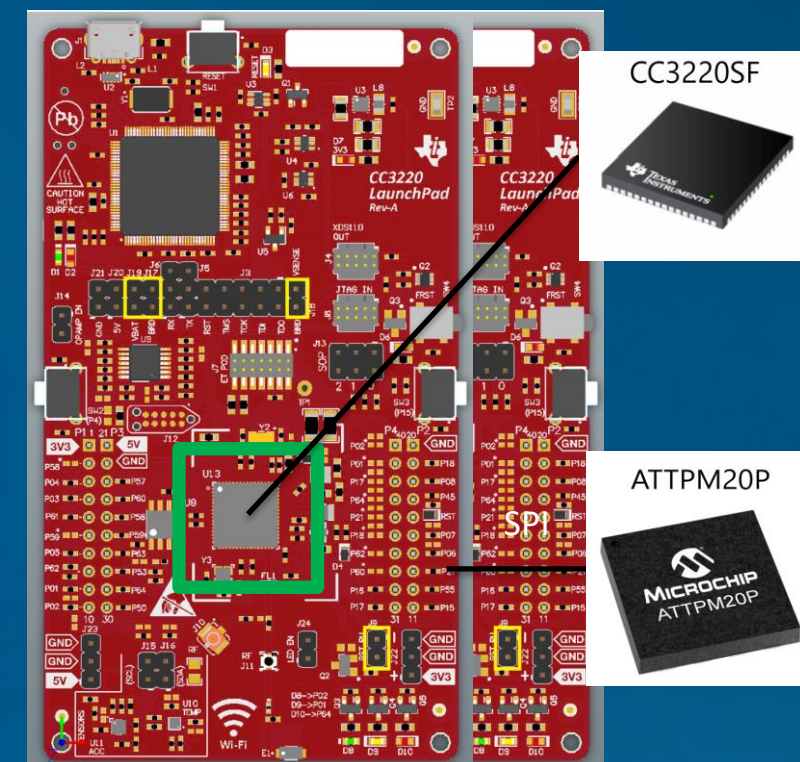
TPM (Trusted Platform Module)

- Resource constrained chip
- Secure crypto-processor
- Secure key generation and key storage
- Hardware Asymmetric/Symmetric Crypto Engine
- Cryptographic Support for: RNG, HMAC, AES, SHA, ECC, RSA, etc.
- Two generations: TPM1.2 and TPM2.0
- Trusted Computing Group compliance

TPM = Security

E2E Demo

Hardware components



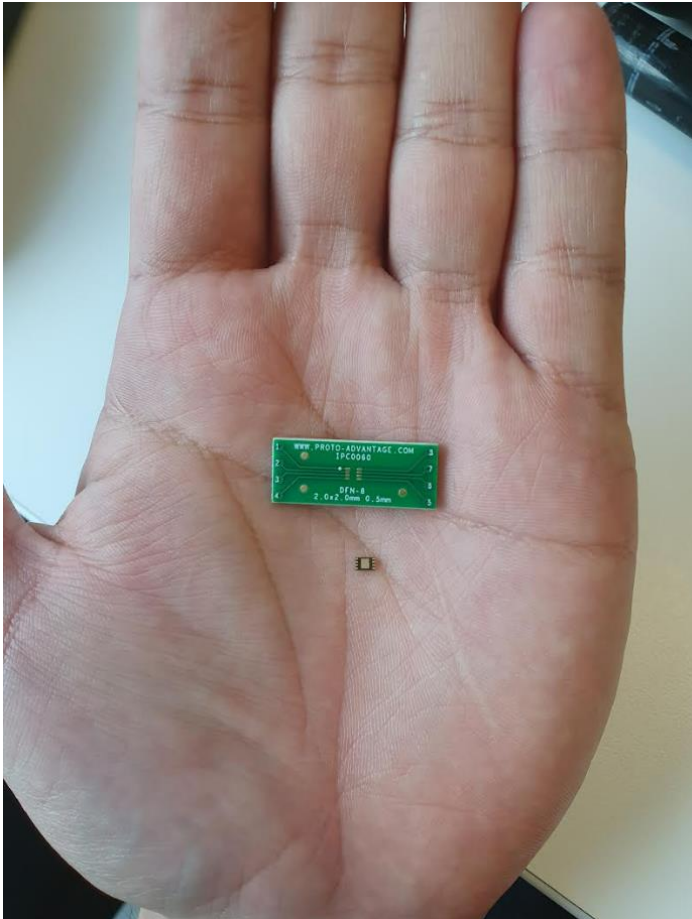
Ti LaunchPad development board

References:

<https://www.ti.com/launchpad>

<https://www.microchip.com/en-us/product/ATTPM20P>

Confidential. This document must not be copied or distributed without KAL's written permission



TPM in real perspective

- True size: 1.9 x 1.5mm
- Cost: Cost less than 2USD
- Version: 2.0 of TCG.
- Interface: (SPI) Protocol up to 36 MHz

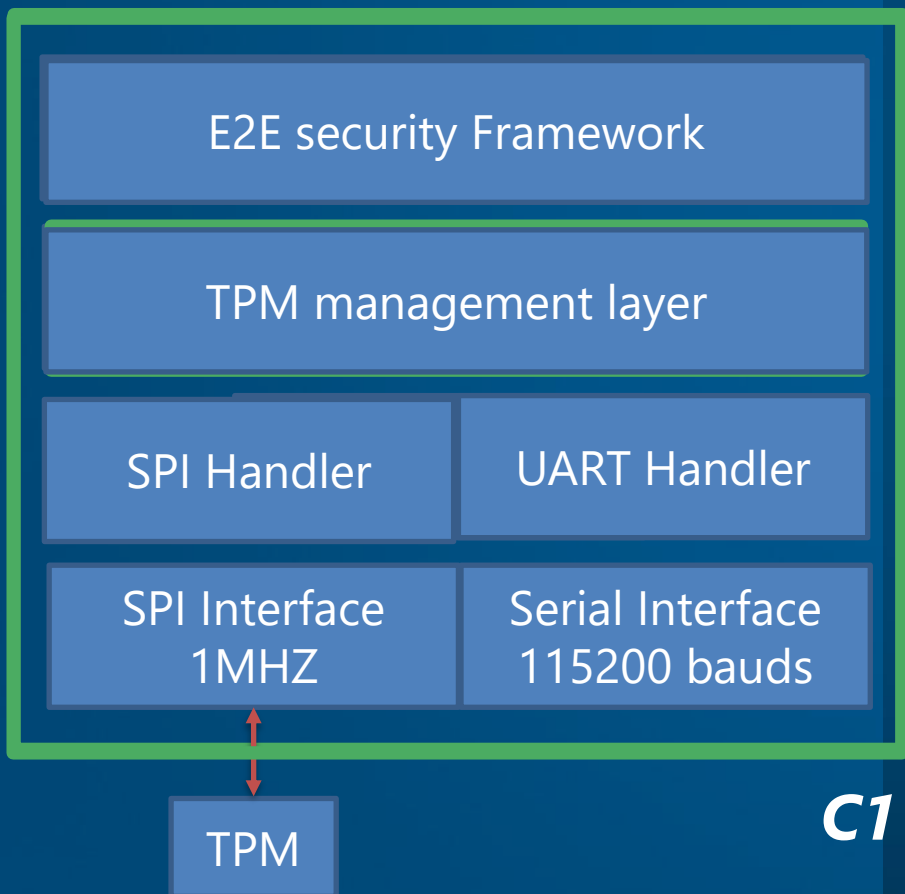
Demo video is available on YouTube.

YouTube channel:

<https://www.youtube.com/user/ATMsoftware/videos>

Where we are?

CC3220SF - Microcontroller



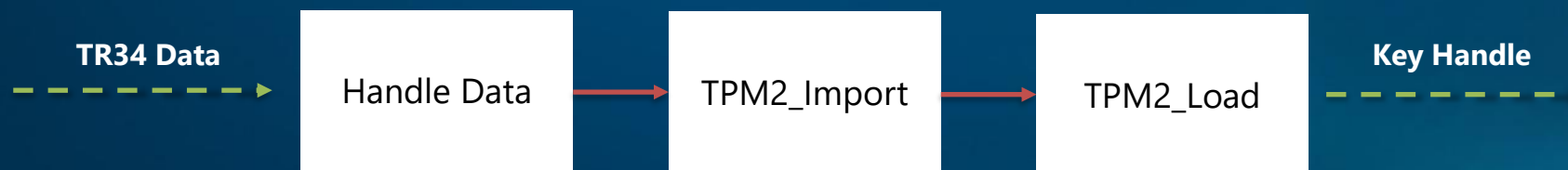
C11

Already Implemented on dev board

- FIFO interface implementation (SPI protocol).
- TPM management layer (TPM commands).
- Dispense token validation (HMAC validation)
- Known Symmetric key directly loaded in TPM.
- RNG managed by TPM.
- HMAC calculation managed by TPM.
- Token verification done by E2E framework implementation
- E2E framework implementation inside MCU the entire SP framework is running in another device.

Currently working on

- Extending functionality to support TR34.
 - Implementing key hierarchy structure to store Asymmetric RSA key used to handle TR34 data.
 - Implementing required TPM commands sequence to avoid handling raw key values in the system memory.
 - Implementing TPM mechanism to import and load the received symmetric key.



Raw key not handled in system memory

What's next?

- TR34 Demo implementation on real hardware.
- Evaluating feasibility to include the TPM implementation into XFS4IoT SP framework (not fully confirmed yet).
 - HW Target → TPM2.0.
 - Implementation of FIFO interface through SPI protocol.
 - Main TPM commands implementation.
 - Make the TPM available to be used as a Crypto-processor for any device class Cash Dispenser, Card Reader, Receipt Printer, etc.

- We will enhance and maintain the framework (i.e. TPM support)
- We will keep the framework up to date with the latest CEN XFS4IoT Specification
- Members of the workgroup are and will still be welcome to be guest speakers and share their experience about XFS4IoT and the KAL SP-Dev framework.

Zoom

- First Tuesday of each month at 1300 UK time for 30 mins

Next call: 3rd May 2022, 1300 UK, 0800 US EST, 2100 Tokyo time

Calls going forward will be reduced to 30mins each month.

We will continue to use Zoom

(Interpretation in Japanese, Chinese and Spanish using Zoom's interpretation feature)