Enable Secure element A71CH on LS1012A with Layerscape SDK

Introduction

The A71CH Host SW API encapsulates the APDU calls supported by the A71CH security module. The standard A71CH security module supports the following functionality:

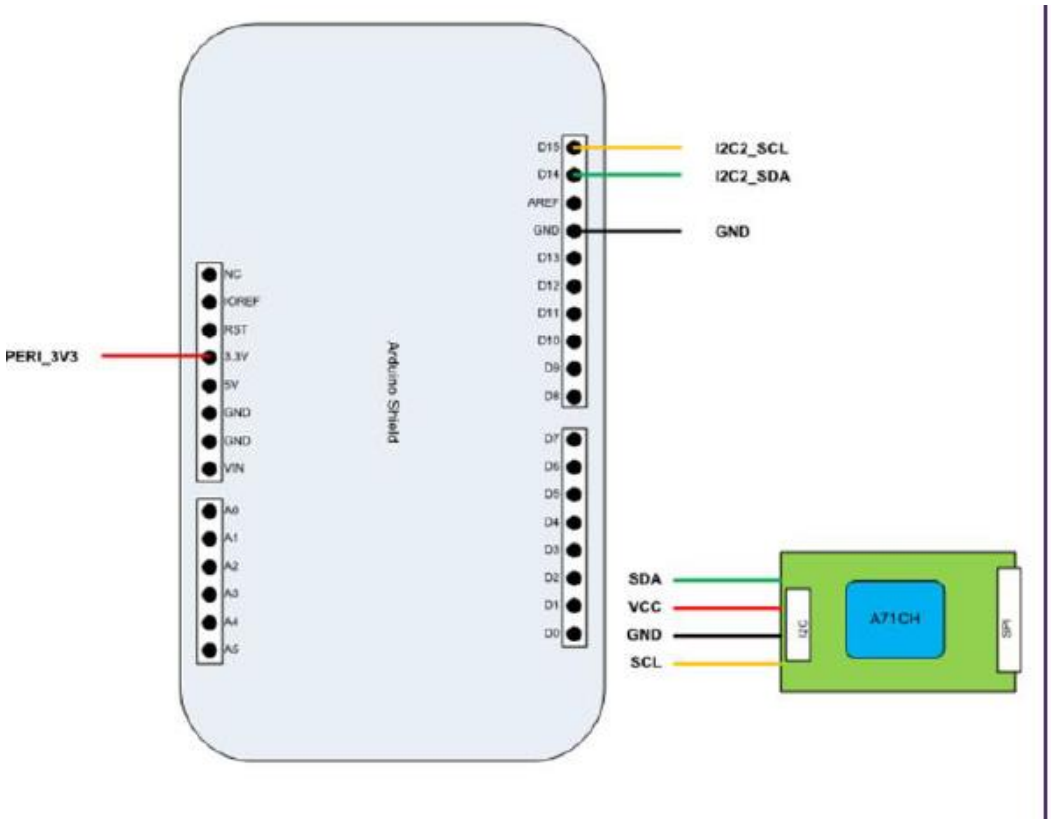
- Secure storage, generation, insertion or deletion of ECC key pairs (ECC NIST P-256).
- Secure storage, insertion or deletion of ECC public keys.
- Signature generation and verification (ECDSA)
- Shared secret calculation for Key Agreement (ECDH or ECDH-E)
- Secure storage and use of monotonic counters (32 bits each)
- Secure storage, insertion or deletion of symmetric keys (128 bits); symmetric keys can be concatenated to form longer keys
- Retrieval of unique chip ID.
- HDKF using the symmetric secrets as key, Extract & Expand or Expand only.
- HMAC SHA256 calculation
- Freezing of credentials (= OTP behavior)
- An optional secure channel with the host MCU (conform Global Platform SCP03).

The Debug Mode variant of the A71CH security module, which can be ordered on evaluation kits, supports the following additional functionality

- A set of debug commands to facilitate integration of the A71CH in a host application.
- Possibility to permanently disable these debug commands

Connecting illustrated

The following drawing illustrates the pins to connect (using the Arduino shield pin naming conventions):



I2C pins on LS1012ARDB board's Arduino-Shield compatible connector

Installation secure module package

1. **Get the package and install it.**

   $wget https://s3-us-west-

2.amaonaws.com/edgescale.org/release/package/arm64/se/A71CH-LAYERSCAPE-ARM64.sh

$ sh A71CH-LAYERSCPAE-ARM64.sh

$ cd src_package/

$ dpkg -i lsse-ng_0.1.0_arm64.deb

2. **Using openssl to create ECC private key and CSR file.**

   a) Create private key

   $ openssl ecparam -out ecckey.key -name prime256v1 -genkey

   b) Create CSR file

   $ openssl req -new -sha256 -key ecckey.key -nodes -out eccCsr.csr

   c) Create public key

   $ openssl ec -in ecckey.key -pubout -out ecpubkey.pem

3. **use "lsse_store" to storage private key into secure module(A71CH) and create reference key file.**

   $ lsse_store

4. **Export environment variable and verification secure module sign and verify, for example**

   $ export OPENSSL_CONF="/etc/ssl/opensslA71CH_i2c.cnf"

   $ openssl dgst -sha256 -sign ecckey.key -out filename.sha256 filename

   $ openssl dgst -sha256 -verify ecckeypub.key -signature filename.sha256 filename

5. **Store certificates**

   $ a71ch_i2c_cert –set ca.crt

   $ a71ch_i2c_cert –info [certificate byte size]

Remark

Regarding the "A71CH Arduino compatible", you can be found in here.

https://www.nxp.com/products/identification-and-security/authentication/om3710-a71chard-a71ch-arduino-compatible-development-kit:OM3710-A71CHARD