MT793X IoT SDK for ECC User Guide

Version: 1.0

Release date: 2021-07-29

Use of this document and any information contained therein is subject to the terms and conditions set forth in Exhibit 1. This document is subject to change without notice.

Version History

Version	Date	Description
1.0	2021-07-29	Official release

Table of Contents

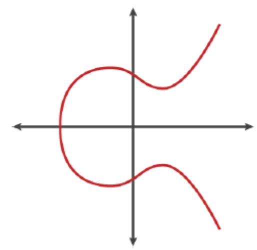
1 Getting Started

This chapter introduces the MT7933 ECC project and gives you an idea of what you need to prepare to get started.

1.1 Overview

The Elliptic Curve Cryptography (ECC) driver is designed to process signature and verification based on the Elliptic curve Digital Signature Algorithm (ECDSA). The ECC module supports 192/224/256/384/521 curve based on National Institute of Standards and Technology(NIST).

Curve Name	Curve Function
NIST P-192	$y^2 = x^3 - 3x + b \pmod{p}$
NIST P-224	
NIST P-256	
NIST P-384	
NIST P-521	



1.2 Code Layout

driver\chip\mt7933\src\hal_ecc_api.c
driver\chip\inc\hal_ecc.h

1.3 ECC APIs

The ECC module provides 2 APIs for signature and verification.

hal_ecc_ecdsa_sign

```
/** @brief Elliptic curve digital signature algorithm sign function.
* Calculate an ECDSA signature using elliptic curve digital signature
 * algorithm with the specified curve, private key, random number, and data.
*
* @param curve[in]  An elliptic curve of choice. See hal_ecc_curve_t for more information.
 * @param d[in]       The private key.
 * @param k[in]       The random data.
 * @param e[in]       The data which has already been hashed to create a signature.
 * @param r[out]      The first part of the signature result.
 * @param s[out]      The second part of the signature result.
*
* @return       #HAL_ECC_STATUS_OK is returned if signature is generated
 *              using the specified curve. Otherwise, see descriptions in hal_ecc_status_t.
*/
hal_ecc_status_t hal_ecc_ecdsa_sign(
            const  hal_ecc_curve_t    curve,
            const  uint32_t         *d,
            const  uint32_t         *k,
            const  uint32_t         *e,
```

```
                    uint32_t        *r,
                    uint32_t        *s);
```

hal_ecc_ecdsa_verify

```
/**@brief Elliptic curve digital signature algorithm verify function.
 * Verify an ECDSA signature using elliptic curve digital signature
 * algorithm with the specified curve, public key, signature, and data.
 *
 * @param curve      An elliptic curve of choice. See hal_ecc_curve_t for more information.
 * @param r[in]      The first part of a signature result.
 * @param s[in]      The second part of a signature result.
 * @param Qx[in]     The first part of a public key.
 * @param Qy[in]     The second part of a public key.
 * @param e[in]      The data which has already been hashed to verify a signature.
 * @param v[out]     The output of verification.
 *
 * @return      #HAL_ECC_STATUS_OK is returned if signature is generated
 *              using the specified curve. Otherwise, see descriptions in hal_ecc_status_t.
 */
hal_ecc_status_t hal_ecc_ecdsa_verify(
            const  hal_ecc_curve_t    curve,
            const  uint32_t        *r,
            const  uint32_t        *s,
            const  uint32_t        *Qx,
            const  uint32_t        *Qy,
            const  uint32_t        *e,
                   uint32_t        *v);
```

2 ECC Sample Use Case

```
/* - Trigger ECC to do signature and verification.
 *  - Step1: Call hal_ecc_init() to initialize the ECC clock.
 *  - Step2: Call hal_ecc_ecdsa_sign() to generate the ECDSA signarure or hal_ecc_ecdsa_verify() to verify the ECDSA signarure.
 *  - Step3: Call hal_ecc_deinit() to de-initialize the ECC clock.
 *  - Sample code:
 *  @code
 *     // ECC needs 32 bytes(32 * 8 bits) length when using NIST P-256 curve. Little endian format.
 *     // You can test sign/verify funtion by using the following golden data.
 *     // e: {0x0377BCC0, 0x26681592, 0x5F3CDF14, 0xC64E5D61, 0xC535C273, 0x637536F7, 0x19F5BF25, 0x1FDA2156}.
 *     // d: {0xA112ED54, 0xFDAF4EE1, 0x4DC4192F, 0x7C7A9947, 0xF013D563, 0x84335DD3, 0x3B51E0FC, 0xACEC122D}.
 *     // k: {0xFD11A53D, 0x0AEBFE6D, 0x3694C98E, 0xA3CE7B21, 0x8566A7E8, 0x2DEA7054, 0x1958A428, 0xC8BDD79F}.
 *     // r: {0xF345B5B5, 0x8926F457, 0xFDAB95A9, 0xBD362686, 0x253EB72A, 0xD33E3511, 0xB21737AE, 0x2F350F06}.
 *     // s: {0x5313B579, 0x814492C3, 0x135D7EF3, 0xA686FD6E, 0xCED6F8A5, 0x0749A6B2, 0x151E00C0, 0x338AE2FA}.
 *     // Qx:{0xC3E79B79, 0x8F335540, 0x684E285C, 0xAAAA74F1, 0x6AE6900E, 0x65455B8E, 0xE75F70CD, 0x5AF2E9D1}.
 *     // Qy:{0x3016AC86, 0x50FDF6D9, 0xB69BA98B, 0xC5EC1D8B, 0x9A296177, 0x32F97CCB, 0xD8565D9D, 0xEC52712F}.
 *
 *     uint32_t  e[8]; // input data
 *     uint32_t  d[8]; // input data
 *     uint32_t  k[8]; // input data
 *     uint32_t  r[8]; // output data for signature, input data for verification
 *     uint32_t  s[8]; // output data for signature, input data for verification
 *     uint32_t  v[8]; // output data
 *     uint32_t  Qx[8]; // input data
 *     uint32_t  Qy[8]; // input data
 *
 *     // Initializes the ECC clock.
 *     if(HAL_ECC_STATUS_OK != hal_ecc_init()) {
 *         //error handle
 *     }
 *     // Generate the ECDSA signarure based on NIST P-256 curve.
 *     if(HAL_ECC_STATUS_OK != hal_ecc_ecdsa_sign(HAL_ECC_CURVE_NIST_P_256, d, k, e, r, s)) {
 *         //error handle
 *     }
 *     // Verify the ECDSA signarure based on NIST P-256 curve.
 *     if(HAL_ECC_STATUS_OK != hal_ecc_ecdsa_verify(HAL_ECC_CURVE_NIST_P_256, r, s, e, Qx, Qy, v)) {
 *         //error handle
 *     }
 *     if(memcmp(v, r, 4 * 8)) {
 *         //error handle
 *     }
 *     // De-initialize the ECC clock.
 *     hal_ecc_deinit();
*/
```

Exhibit 1 Terms and Conditions

Your access to and use of this document and the information contained herein (collectively this “Document”) is subject to your (including the corporation or other legal entity you represent, collectively “You”) acceptance of the terms and conditions set forth below (“T&C”). By using, accessing or downloading this Document, You are accepting the T&C and agree to be bound by the T&C. If You don’t agree to the T&C, You may not use this Document and shall immediately destroy any copy thereof.

This Document contains information that is confidential and proprietary to MediaTek Inc. and/or its affiliates (collectively “MediaTek”) or its licensors and is provided solely for Your internal use with MediaTek’s chipset(s) described in this Document and shall not be used for any other purposes (including but not limited to identifying or providing evidence to support any potential patent infringement claim against MediaTek or any of MediaTek’s suppliers and/or direct or indirect customers). Unauthorized use or disclosure of the information contained herein is prohibited. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for Your unauthorized use or disclosure of this Document, in whole or in part.

MediaTek and its licensors retain titles and all ownership rights in and to this Document and no license (express or implied, by estoppels or otherwise) to any intellectual propriety rights is granted hereunder. This Document is subject to change without further notification. MediaTek does not assume any responsibility arising out of or in connection with any use of, or reliance on, this Document, and specifically disclaims any and all liability, including, without limitation, consequential or incidental damages.

THIS DOCUMENT AND ANY OTHER MATERIALS OR TECHNICAL SUPPORT PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT, IF ANY, ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, COMPLETENESS OR ACCURACY AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MEDIATEK SHALL NOT BE RESPONSIBLE FOR ANY MEDIATEK DELIVERABLES MADE TO MEET YOUR SPECIFICATIONS OR TO CONFORM TO A PARTICULAR STANDARD OR OPEN FORUM.

Without limiting the generality of the foregoing, MediaTek makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MediaTek assume any liability arising out of the application or use of any product, circuit or software. You agree that You are solely responsible for the designing, validating and testing Your product incorporating MediaTek’s product and ensure such product meets applicable standards and any safety, security or other requirements.

The above T&C and all acts in connection with the T&C or this Document shall be governed, construed and interpreted in accordance with the laws of Taiwan, without giving effect to the principles of conflicts of law.