NXP Semiconductors
Application Note

Document Number: AN12272
Rev. 0, 10/2018

Use of Unique 40-bit MAC Address in Kinetis® Wireless Microcontrollers

1. Introduction

For IEEE standard based wireless devices, device address is required to be unique to ensure transmitted data is received by the intended recipient.

To meet this requirement, NXP's Kinetis wireless microcontroller series (KW) ships a pre-programmed address that can be used to form a global unique address. This feature allows a static firmware image to be programmed during manufacturing, alleviating the need for companies to program a dynamic firmware image which only differs by the address.

This application note describes how IEEE MAC addresses are handled in the Bluetooth low energy (BLE) and IEEE 802.15.4 that are the foundation of Thread and Zigbee mesh networking protocols.

It also describes how these addresses are handled in NXP's protocol stacks and how to utilize the unique 40-bit MAC address factory programmed by NXP into each Kinetis KW device to generate a global unique address for a companies end product.

Contents

2. IEEE Addresses for BLE and IEEE 802.15.4

Before explaining how to use the unique 40-bit MAC address provided in NXP Kinetis Wireless Microcontrollers Series, it is useful to know how addresses are formed for BLE and IEEE 802.15.4 based wireless protocols.

BLE addresses (BD_ADDR) are unique 48-bit addresses assigned to each Bluetooth device by the manufacturer.

The upper half of a BLE address (most-significant 24 bits) is called the Organizationally Unique Identifier (OUI) and is used to determine the manufacturer of a device.
OUI prefixes are assigned by the Institute of Electrical and Electronic Engineers (IEEE) and are unique to each company. A company is assigned a block of addresses and thus will have multiple OUIs.

The lower half (LAP) of a BLE address (least-significant 24 bits) is where a unique value is programmed into each device manufactured by the company. This provides 2^24 or over 16 M unique addresses. If a company ships over 2^24 devices to the market, the OUI will need to change to ensure devices with the same address are not shipped.

Figure 1 shows a BLE address diagram.

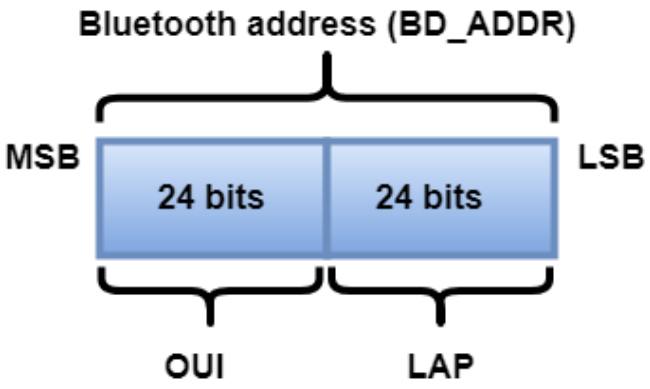

Figure 1. BLE Address

For information on the company assigned numbers, visit:
https://www.bluetooth.com/specifications/assigned-numbers

IEEE 802.15.4 based wireless protocols, like Thread and Zigbee, use a unique 64-bit MAC address which consists of a 24-bit Organizationally Unique Identifier (OUI), and a 40-bit unique value provided by the company which assures 2 ^ 40 unique addresses for the devices shipped by each company.

If a company ships over 2^40 devices to the market, the OUI will need to change to ensure devices with the same address are not shipped.

Figure 2 shows a 64-bit IEEE 802.15.4 MAC address.

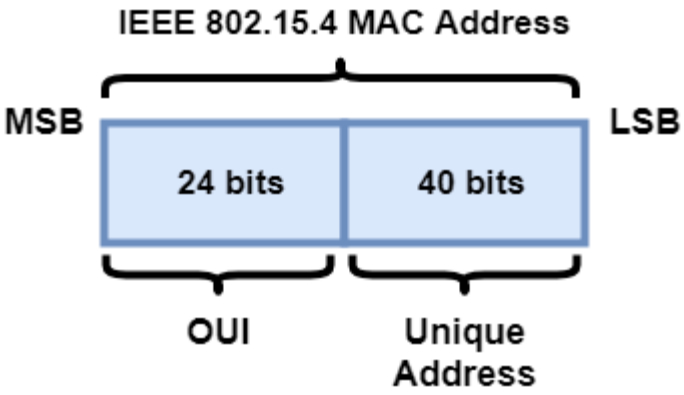

Figure 2. IEEE 802.15.4 MAC Address

3. Kinetis W Factory Programmed 40-bit Unique MAC Address

3.1 Uniqueness

NXP's Kinetis wireless microcontroller series (KW) include a unique 40-bit MAC address factory programmed which can be used to form a global unique address for BLE and IEEE 802.15.4 based wireless protocols.

It is included in KW40Z/30Z/20Z, KW41Z/31Z/21Z, KW35A/36A, KW35Z/36Z and K32W0x microcontrollers.

40-bit MAC address ensures uniqueness and tracks the device ID with the unique value associated for that device. NXP guarantees no duplication across the Kinetis Wireles connectivity devices for the unique 40-bit MAC address for the first 2^40 shipped microcontrollers.

3.2 Registers

For KW41Z/31Z/21Z, KW35A/36A, KW35Z/36Z and KW40Z/30Z/20Z microcontrollers, the 40-bit MAC address is stored in **MAC_ADDR_LSB** and **MAC_ADDR_MSB** read only registers as shown in *Figure* 3.

Field	Function
31-0 MAC_ADDR_LSB	Radio MAC Address LSB The Radio MAC Address is loaded from the Flash IFR during the SoC Power on Reset sequence. The MAC Address is a unique ID that is stored in the Flash during factory test.

Field	Function
31-8 Reserved	Reserved
7-0 MAC_ADDR_MSB	Radio MAC Address MSB The Radio MAC Address is loaded from the Flash IFR during the SoC Power on Reset sequence. The MAC Address is a unique ID that is stored in the Flash during factory test.

Figure 3. . MAC_ADDR_LSB and MAC_ADDR_MSB

For K32W0x microcontrollers, the 40-bit MAC address is stored in **RFADDRL** (*Figure* 4) and **RFADDRH (***Figure* **5**) registers.

Field	Function
31-24 MACADDR3	MACADDR3 RF address. Loaded from IFR
23-16 MACADDR2	MACADDR2 RF address. Loaded from IFR
15-8 MACADDR1	MACADDR1 RF address. Loaded from IFR
7-0 MACADDR0	MACADDR0 RF address. Loaded from IFR

Figure 4. RFADDRL

Field	Function
31-8 —	Reserved
7-0 MACADDR4	MACADDR4 RF address. Loaded from IFR

Figure 5. RFADDRH

3.3 Software

Addresses handling in the NXP connectivity protocol stacks for all Kinetis Wireless microcontrollers series (KW) could make use of the UID (Unique Identification for the Device) to generate address instead of the 40-bit MAC address.

Unique Identification value can be found in all Kinetis families but it's not a tracked ID so it is not recommended to form a global unique address for the wireless protocol. The recommendation is to use the unique 40-bit MAC address which is a tracked ID.

NOTE

40-bit MAC address is not included for KW01Z and KW2xD microcontrollers. UID can be used in this case.

Below examples show how addresses are generated to make use of the 40-bit MAC address in KW41 KSDK for BLE and Zigbee wireless protocol stacks.

Example 1. Change the Bluetooth Address using the UID

An example taken from KW41 BLE connectivity stack “SDK_2.2_FRDM-KW41Z” downloaded from https://mcuxpresso.nxp.com make use of the function “ControllerSetBD_ADDR” to change the Bluetooth Address using the UID as shown in Figure 6.

```
static void ControllerSetBD_ADDR(void)
{
    sha256Context_t mCtx;
    uint8_t uid[16] = {0};
    uint8_t len = 0;
    uint8_t i=0;

    NV_ReadHWParameters(&gHardwareParameters);
    while( (i<6) && (gHardwareParameters.bluetooth_address[i] == 0xFF) ){ i++; }
    if( i == 6 )
    {
        BOARD_GetMCUUid(uid, &len);
        SHA256_Hash(&mCtx, uid, len);
        FLib_MemCpy(gHardwareParameters.bluetooth_address, (uint8_t *)(&mCtx.hash), 3);
        FLib_MemCpy(&gHardwareParameters.bluetooth_address[3], (uint8_t *)gBD_ADDR_OUI_c, 3);
        NV_WriteHWParameters(&gHardwareParameters);
    }
    FLib_MemCpy(gBD_ADDR, gHardwareParameters.bluetooth_address, 6);
}
```

Figure 6. UID to set ble address

Example 2. 40-bit MAC address implementation

Figure 7 shows an example of 40-bit MAC address implementation where “ControllerSetBD_ADDR” function was modified to support it.

```
static void ControllerSetBD_ADDR(void)
{

    uint64_t mac;

     mac = (RSIM->MAC_LSB) | (RSIM->MAC_MSB<<32);
     FLib_MemCpy(gHardwareParameters.bluetooth_address, (uint8_t *) &mac, 3);
     FLib_MemCpy(&gHardwareParameters.bluetooth_address[3], (void *)gBD_ADDR_OUI_c, 3);
     NV_WriteHWParameters(&gHardwareParameters);


    FLib_MemCpy(gBD_ADDR, gHardwareParameters.bluetooth_address, 6);


}
```

Figure 7. 40-bit MAC address to set bluettoth addres

Other example taken from the same KW41 connectivity software, it is for Zigbee where the MAC address is generated in “APP_vSetMacAddr” function that makes use of the UID. For this example, MAC address is loaded in **u64IeeeAddr** variable.

If 40-bit MAC address implementation is needed, then “APP_vSetMacAddr” function needs to be modified by only setting **u64IeeeAddr** with the 40-bit MAC address and with the OUI portion.

4. Conclusion

The use of unique address in each device trough the diverse wireless protocols is a necessity in the market; To fill this need, each microcontroller must have a unique identifier to be used for this purpose.

NXP Kinetis Wireless Microcontrollers Series helps to provide devices in market with unique identifiers to assure uniqueness in each device by including a 40-bit MAC unique address programmed in mcu’s flash.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number: AN12272
Rev. 0
10/2018