SMH Technologies®

14/03/2025

Driver v. 5.14

Moreno Ortolan

Interfacing FlashRunner 2.0 with NXP KINETIS

SMH Technologies S.r.l.

NXP KINETIS Families

KINETIS K32 L Series Arm Cortex M4/M0+

K32 L Series: Ultra-Low Power Microcontrollers (MCUs) Optimized for Low-Leakage Applications

Based on the Arm® Cortex®-M4 and Cortex-M0+ architectures, K32 L series MCUs provide software and tool enablement via MCUXpresso suite of tools, and scalability making this portfolio an ideal fit for IoT applications.
The K32 L series of MCUs is part of NXP's EdgeVerse™ edge computing platform.

	CPU		Packages		Comms		HMI	Security	
Family	Arm Cortex-M	Memory	Type	Pin Count	SDHC	USB Full-Speed	FlexIO	HW Encryption	Tamper Detection
K32 L3 Energy Efficient and Secure	Cortex-M4 and Cortex-M0+ 72 MHz	Up to 1.25 MB Flash 384 kB SRAM	VFBGA, MBGA, LQFP, QFN, LQFP	64-176	✓	✓	✓	✓	✓
K32 L2 Ultra-Low-Power, Highly Integrated	Cortex-M0+ 72 MHz	64-512 kB Flash, 32-128 kB RAM	QFN, LQFP, MAPBGA	32-100	-	✓	✓	✓	-

KINETIS K Series Arm Cortex M4

Kinetis® K Series: High-Performance Microcontrollers (MCUs) Based on Arm® Cortex®-M4 Core

Kinetis K series MCUs offer optimized performance, scalable integration and low-power capabilities.

	CPU			Comms			HMI		Security		
Sub-Families	Arm Cortex-M4	Memory	Packages	Ethernet	CAN	USB	SVGA LCD	Segment LCD	HW Encryption	Tamper Detection	Firmware Upd.
K8x Scalable and Secure	150 MHz	256 KB Flash 256 KB SRAM, XIP QuadSPI	LQFP, MAPBGA, WLCSP	-	-	✓	-	-	✓	✓	✓
K7x Graphic LCD	120–150 MHz	1 MB Flash 128 KB SRAM	MAPBGA	✓	✓	✓	✓	-	✓	✓	-
K6x Ethernet	100–180 MHz	256 KB–2 MB Flash 64–256 KB SRAM	LQFP, MAPBGA, WLCSP	✓	✓	✓	-	-	✓	✓	-
K5x Measurement	72–100 MHz	160–512 KB Flash 32–128 KB SRAM	LQFP, MAPBGA	✓	-	✓	-	✓	✓	-	-
K4x USB and segment LCD	72–100 MHz	64–512 KB Flash 16–128 KB SRAM	LQFP, MAPBGA	-	✓	✓	-	✓	-	-	-
K3x Segment LCD	72–100 MHz	64–512 KB Flash 16–128 KB SRAM	LQFP, MAPBGA	-	✓	-	-	✓	-	-	-
K2x / KS2x USB	50–180 MHz	32 KB–2 MB Flash 8 KB–1 MB SRAM, XIP QuadSPI	LQFP, QFN, MAPBGA, WLCSP	-	✓	✓	-	-	✓	✓	-
K1x Mainstream	50–120 MHz	32 KB–1 MB Flash 8–128 KB SRAM	LQFP, QFN, MAPBGA	-	✓	-	-	-	✓	✓	-
K0x Entry-level	100 MHz	64–128 KB Flash 16 KB SRAM	LQFP, QFN	-	-	-	-	-	-	-	-

KINETIS KL Series Arm Cortex M0+

Kinetis® L Series: Ultra-Low Power Microcontrollers (MCUs) Based on Arm® Cortex®-M0+ Core

Kinetis® L series MCU portfolio includes more than 200 compatible, low-power, high-performance 32-bit MCUs. This series combines the low-power performance and energy-efficiency of the Arm® Cortex®-M0+ core with the peripheral sets, enablement and scalability of the Kinetis MCU portfolio of solutions for internet of things (IoT) applications.

	CPU		Packages		Comms		HMI		Security	
Products	Arm Cortex-M0+	Memory	Type	Pin Count	I²S	USB Full-Speed	FlexIO*	Segment LCD	HW Encryption	Tamper Detection
KL8x Security	72 MHz (Up to 96 MHz)	128 KB Flash 96 KB SRAM	LQFP, MAPBGA, WLCSP	64 – 121	✓	✓	✓	✓	✓	✓
KL4x USB and Segment LCD	48 MHz	128 – 256 KB Flash 16 – 32 KB SRAM	LQFP, MAPBGA	64 – 121	✓	✓	✓	✓	-	-
KL3x Segment LCD	48 MHz	32 – 256 KB Flash 4 – 32 KB SRAM	LQFP, MAPBGA	64 – 121	✓	-	✓	✓	-	-
KL2x USB	48 MHz / 72MHz (Up to 96)	32 – 512 KB Flash 4 – 128 KB SRAM	LQFP, QFN, MAPBGA, XFBGA, WLCSP	32 – 121	✓	✓	✓	-	✓	-
KL1x Mainstream	48 MHz	32 – 256 KB Flash 4 – 32 KB SRAM	LQFP, QFN, MAPBGA, XFBGA, WLCSP	32 – 80	✓	-	✓	-	-	-
KL0x Entry-Level	48 MHz	8 – 32 KB Flash 1 – 4 KB SRAM	LQFP, QFN WLCSP	16 – 48	-	-	-	-	-	-

KINETIS KV Series Arm Cortex M4/M0+/M7

KV Series: Real-Time Motor Control and Power Conversion MCUs Based on Arm® Cortex®-M0+/M4/M7

KV Series of MCUs are part of the EdgeVerse™ edge computing platform and are designed to support a wide range of BLDC, PMSM and ACIM motor control as well as digital power conversion applications.

	CPU	Memory		Timers						Analog			Connectivity		Security	
Products	Arm	Flash / SRAM	Packages	FlexTimers	eFlexPWM	PDB	Quad Dec	Quad Enc	Hi-Res Timer	ADC	DAC	ACMP	Ethernet	CAN	CRC	HW Encryption
KV5x Power Conversion	Cortex-M7 240 MHz	512 KB – 1 MB / 128 – 256 KB	LQFP, MAPBGA	2 x 8ch 2 x 2ch	2 x 12ch	✓	✓	✓	1 x 12ch (260pS)	4 x 12bit (5 Msps)	✓	✓	✓	✓	✓	✓
KV4x High Performance Motor	Cortex-M4 168 MHz	64 – 256 KB / 16 – 32 KB	LQFP	2 x 8ch 1 x 2ch	1 x 12ch	✓	✓	✓	1 x 12ch (312pS)	2 x 12bit (4.1 Msps)	✓	✓	-	✓	✓	-
KV3x Sensorless Motor Control	Cortex-M4 100 – 120 MHz	64 – 512 KB / 16 – 96 KB	LQFP, QFN	2 x 8ch 2 x 2ch	-	✓	✓	-	-	2 x 16-bit (1.2 Msps)	✓	✓	-	-	✓	-
KV1x Entry-Level	Cortex-M0+ 75 MHz	16 – 128 KB / 8 – 16 KB	LQFP, QFN	2 x 6ch 2 x 2ch	-	✓	✓	-	-	2 x 16-bit (1.2 Msps)	✓	✓	-	✓	✓	-

KINETIS KE Series Arm Cortex M4/M0+

Kinetis® E Series: 5V, Robust Microcontrollers (MCUs) Based on Arm® Cortex®-M0+/M4 Core

The 5V KE series of MCUs are part of the EdgeVerse™ edge computing platform and are designed to maintain high-reliability and robustness in harsh electrical noise environments, targeting white goods and industrial applications.

	CPU			Communication Interfaces				
Products	Arm	Memory	Packages	I²C	SPI	LPUART	CAN	TSI
KE1xF Performance w/ CAN	168 MHz Cortex-M4 w/ FPU	256 – 512 KB Flash 32 – 64 KB SRAM	64-LQFP, 100-LQFP	2	2	3	up to 2 x FlexCAN	-
KE1xZ Mainstream w/ Touch	72 MHz Cortex®-M0+	32 – 256 KB Flash 4 – 32 KB SRAM	44-LQFP, 48-LQFP, 64-LQFP, 100-LQFP, 40-QFN	2	2	3	1	Up to 50 ch
KE06 Entry-Level w/ CAN	48 MHz Cortex®-M0+	64 – 128 KB Flash 8 – 16 KB SRAM	44-LQFP, 64-LQFP, 64-QFP,80-LQFP	2	2	3	1	-
KE04 Entry-Level w/ Mixed Signal	48 MHz Cortex®-M0+	8 – 128 KB Flash 1 – 16 KB SRAM	16-TSSOP, 24-QFN, 44-LQFP, 64-LQFP, 64-QFP, 80-LQFP, 20-SOIC WB	Up to 2	Up to 2	Up to 3	-	-
KE02 Entry-Level w/ EEPROM	20 - 40 MHz Cortex®-M0+	16 – 64 KB Flash 2 – 4 KB SRAM	32-QFN, 32-LQFP, 44-LQFP, 64-LQFP, 64-QFP	1	2	Up to 3	-	-

KINETIS KM Series Arm Cortex M0+

KM Series: Metrology Microcontrollers (MCUs) Based on Arm® Cortex®-M0+ Core

KM series low-power MCUs are part of the EdgeVerse™ edge computing platform and are designed to support single-chip designs for one-, two- and three-phase electricity meters, flow meters and other precision measurement applications.

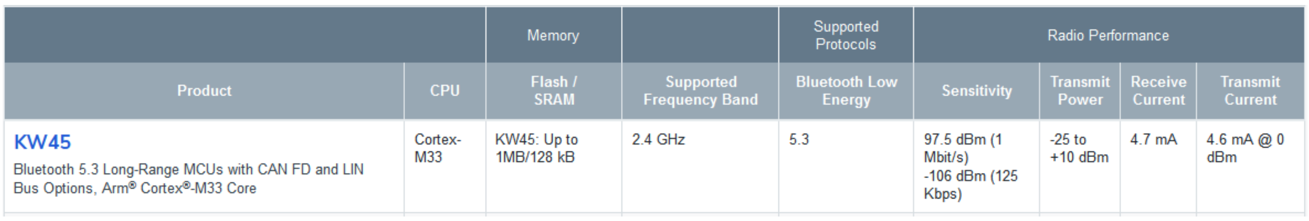
	CPU	Memory		Development Hardware	HMI	Comms			Security		
Products	Arm Cortex-M0+	Flash / SRAM	Packages	Tower Module	SLCD	UART (ISO7816/ LPUART)	I²C	SPI	RNG	CRC	mmCAU
KM1x Baseline	50 MHz	64 – 128 KB / 16 KB	44LGA	TWR-KM34Z50M(V3)	N/A	2 (2 / -)	1	2	✓	✓	N/A
KM3x Segment LCD	50–75 MHz	64 – 512 KB / 16 – 64 KB	64LQFP, 100LQFP, 144LQFP	TWR-KM34Z50M(V3) TWR-KM34Z75M TWR-KM35Z75M	Up to 56 X 8	4-5 (2 / 0-1)	2	Up to 3	✓	✓	Optional

SMH Technologies®

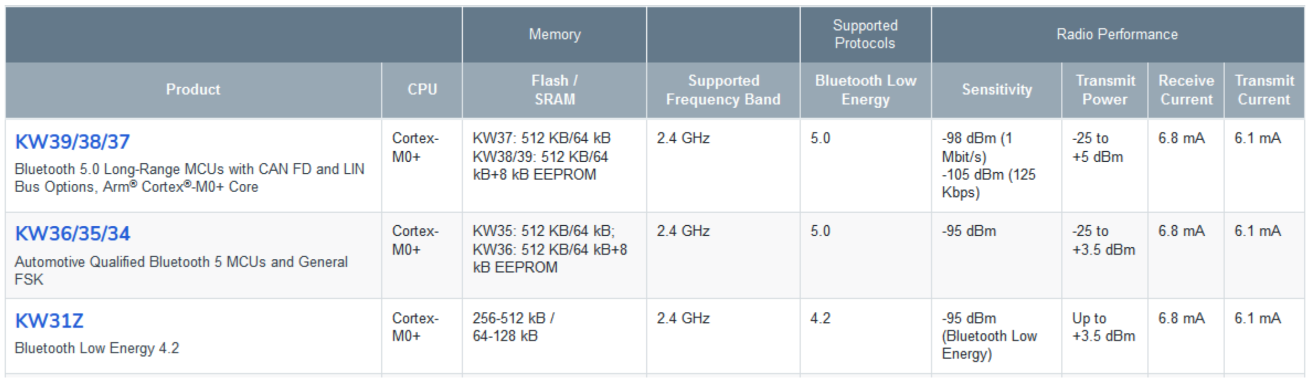
KINETIS KW Series Arm Cortex M33/M0+

KW Series: Bluetooth® Smart/Bluetooth Low Energy

Our Bluetooth Smart (Bluetooth Low Energy) solutions include highly integrated SoCs with plentiful Flash and SRAM memory utilizing Arm® Cortex®-M cores. These products enable ultra-low-power operation without compromising radio performance. In addition to providing complete SoCs, a host Bluetooth Low Energy stack is provided, as well as several GATT profiles and services.



		Memory		Supported Protocols	Radio Performance			
Product	CPU	Flash / SRAM	Supported Frequency Band	Bluetooth Low Energy	Sensitivity	Transmit Power	Receive Current	Transmit Current
KW45 Bluetooth 5.3 Long-Range MCUs with CAN FD and LIN Bus Options, Arm® Cortex®-M33 Core	Cortex-M33	KW45: Up to 1MB/128 kB	2.4 GHz	5.3	97.5 dBm (1 Mbit/s) -106 dBm (125 Kbps)	-25 to +10 dBm	4.7 mA	4.6 mA @ 0 dBm



		Memory		Supported Protocols	Radio Performance			
Product	CPU	Flash / SRAM	Supported Frequency Band	Bluetooth Low Energy	Sensitivity	Transmit Power	Receive Current	Transmit Current
KW39/38/37 Bluetooth 5.0 Long-Range MCUs with CAN FD and LIN Bus Options, Arm® Cortex®-M0+ Core	Cortex-M0+	KW37: 512 KB/64 kB KW38/39: 512 KB/64 kB+8 kB EEPROM	2.4 GHz	5.0	-98 dBm (1 Mbit/s) -105 dBm (125 Kbps)	-25 to +5 dBm	6.8 mA	6.1 mA
KW36/35/34 Automotive Qualified Bluetooth 5 MCUs and General FSK	Cortex-M0+	KW35: 512 KB/64 kB; KW36: 512 KB/64 kB+8 kB EEPROM	2.4 GHz	5.0	-95 dBm	-25 to +3.5 dBm	6.8 mA	6.1 mA
KW31Z Bluetooth Low Energy 4.2	Cortex-M0+	256-512 kB / 64-128 kB	2.4 GHz	4.2	-95 dBm (Bluetooth Low Energy)	Up to +3.5 dBm	6.8 mA	6.1 mA

KINETIS KEA Series Arm Cortex M0+

Ultra-Reliable KEA Automotive Microcontrollers (MCUs) based on Arm® Cortex®-M0+ Core

Kinetis® EA series of 32-bit Arm® Cortex® MCUs are targeted for a wide range of automotive and industrial applications requiring the highest level of quality and longevity support.
The Kinetis EA series is an entry point to the broad Arm ecosystem and features:

- A low-power Arm Cortex-M0+ core and 8–128 KB of embedded flash
- Excellent EMC/ESD, high temperature, and low radiated emissions
- Scalable, highly robust performance solution for cost-sensitive automotive applications
- A broad set of reference designs, tools and application notes to help shorten design development and speed time-to-market

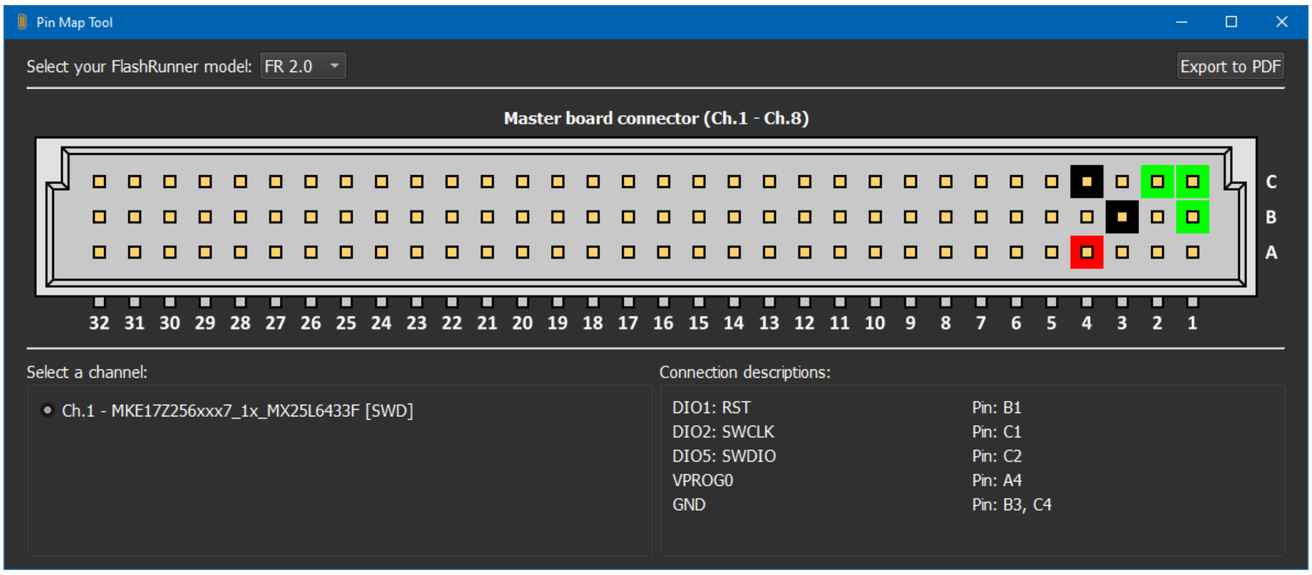
Device	Flash	RAM	EEPROM	Freq	MSCAN	SCI	SPI	ATD	Flex-Tim	IIC	GPIO	Packages
KEAZN8	8K	1K	Emulated	48MHz	0	1	1	12c12b	6c+2c 16b	1	Up to 22	16 TSSOP/24 QFN
KEAZN16	16K	2K	256B	40MHz	0	3	2	16c12b	6c+2c+2c 16b	2	Up to 57	32/64 LQFP
KEAZN32	32K	4K	256B	40MHz	0	3	2	16c12b	6c+2c+2c 16b	2	Up to 57	32/64 LQFP
KEAZN64	64K	4K	256B	40MHz	0	3	2	16c12b	6c+2c+2c 16b	2	Up to 57	32/64 LQFP
KEAZ64	64K	8K	Emulated	48MHz	1	3	2	16c12b	6c+2c+2c 16b	2	Up to 71	64/80 LQFP
KEAZ128	128K	16K	Emulated	48MHz	1	3	2	16c12b	6c+2c+2c 16b	2	Up to 71	64/80 LQFP

NXP KINETIS Protocol and PIN map

KINETIS devices support the SWD protocol.

```
#TCSETPAR CMODE <SWD>
```

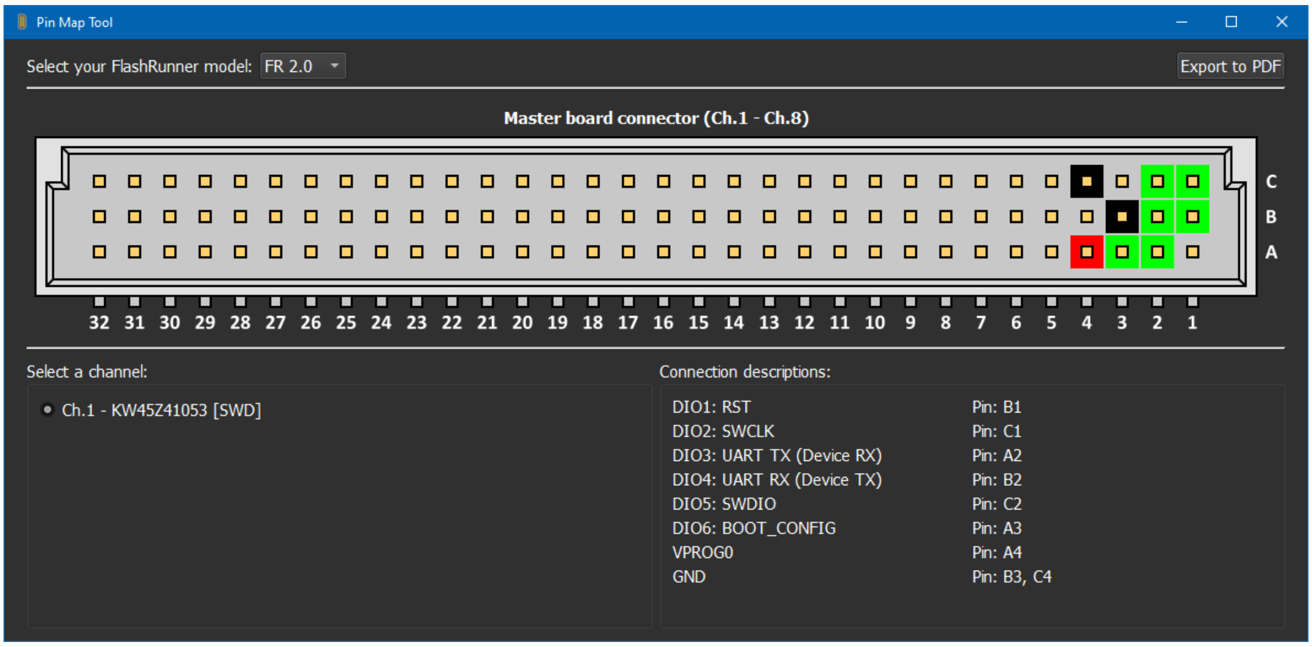
NXP KINETIS PIN MAP



Ch.1 - MKE17Z256xxx7_1x_MX25L6433F [SWD]

DIO1: RST	Pin: B1
DIO2: SWCLK	Pin: C1
DIO5: SWDIO	Pin: C2
VPROG0	Pin: A4
GND	Pin: B3, C4

NXP KINETIS PIN MAP for KW45 devices

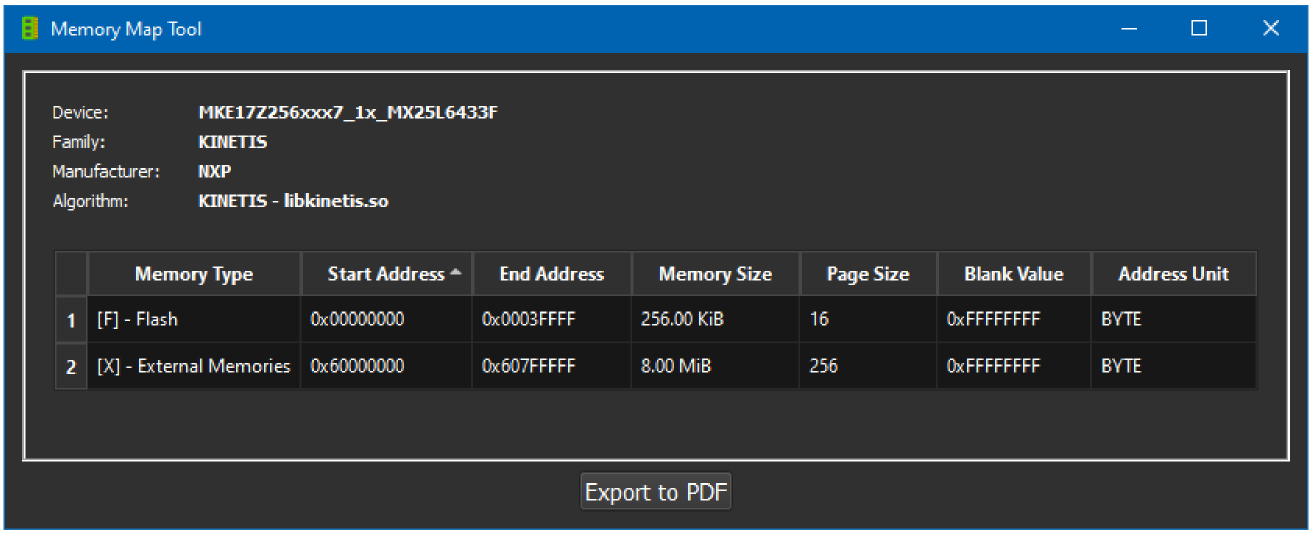


Ch.1 - KW45Z41053 [SWD]

DIO1: RST	Pin: B1
DIO2: SWCLK	Pin: C1
DIO3: UART TX (Device RX)	Pin: A2
DIO4: UART RX (Device TX)	Pin: B2
DIO5: SWDIO	Pin: C2
DIO6: BOOT_CONFIG	Pin: A3
VPROG0	Pin: A4
GND	Pin: B3, C4

NXP KINETIS Memory Map

NXP KINETIS Memory Map

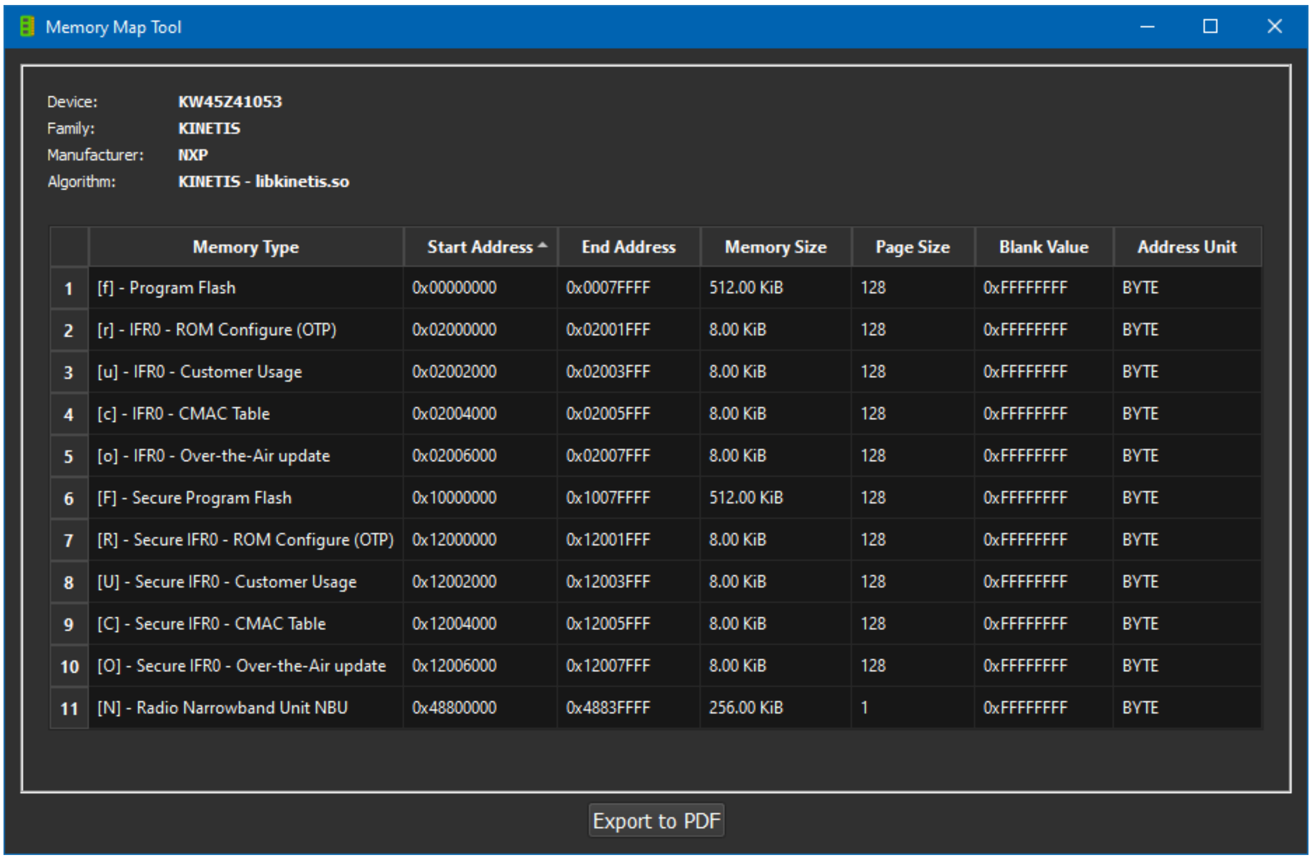


Memory Map Tool

Device: MKE17Z256xxx7_1x_MX25L6433F
Family: KINETIS
Manufacturer: NXP
Algorithm: KINETIS - libkinetis.so

	Memory Type	Start Address	End Address	Memory Size	Page Size	Blank Value	Address Unit
1	[F] - Flash	0x00000000	0x0003FFFF	256.00 KiB	16	0xFFFFFFFF	BYTE
2	[X] - External Memories	0x60000000	0x607FFFFF	8.00 MiB	256	0xFFFFFFFF	BYTE

NXP KINETIS Memory Map for KW45 devices



Memory Map Tool

Device: KW45Z41053
Family: KINETIS
Manufacturer: NXP
Algorithm: KINETIS - libkinetis.so

	Memory Type	Start Address	End Address	Memory Size	Page Size	Blank Value	Address Unit
1	[f] - Program Flash	0x00000000	0x0007FFFF	512.00 KiB	128	0xFFFFFFFF	BYTE
2	[r] - IFR0 - ROM Configure (OTP)	0x02000000	0x02001FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
3	[u] - IFR0 - Customer Usage	0x02002000	0x02003FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
4	[c] - IFR0 - CMAC Table	0x02004000	0x02005FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
5	[o] - IFR0 - Over-the-Air update	0x02006000	0x02007FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
6	[F] - Secure Program Flash	0x10000000	0x1007FFFF	512.00 KiB	128	0xFFFFFFFF	BYTE
7	[R] - Secure IFR0 - ROM Configure (OTP)	0x12000000	0x12001FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
8	[U] - Secure IFR0 - Customer Usage	0x12002000	0x12003FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
9	[C] - Secure IFR0 - CMAC Table	0x12004000	0x12005FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
10	[O] - Secure IFR0 - Over-the-Air update	0x12006000	0x12007FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
11	[N] - Radio Narrowband Unit NBU	0x48800000	0x4883FFFF	256.00 KiB	1	0xFFFFFFFF	BYTE

NXP KINETIS Flash Configuration Field

Flash Configuration Field is an area of Main Flash between the addresses **0x400** and **0x410**.
In this area you can set some configuration for the device behaviour.

A random value written here may lock the device forever and any attempt to reprogram it will be failed.

After the execution of **#TPCMD** MASSERASE UNLOCK this area is blank with 0xFF value except for the **Flash Security Byte** that is 0xFE.
So, after **#TPCMD** MASSERASE UNLOCK command, the Flash memory is totally blank except for this 0xFE value and the Blankcheck operation will give you PASS.

Here an example of Flash Configuration Field for a Kinetis MKE1xZxx device:

Flash Configuration Field Offset Address	Size (Bytes)	Field Description
0x0_0400 - 0x0_0407	8	Backdoor Comparison Key. Refer to Verify Backdoor Access Key command and Unsecuring the MCU Using Backdoor Key Access.
0x0_0408 - 0x0_040B	4	Program flash protection bytes. Refer to the description of the Program Flash Protection Registers (FPROT0-3).
0x0_040F	1	Data flash protection byte. Refer to the description of the Data Flash Protection Register (FDPROT).
0x0_040E	1	EEPROM protection byte. Refer to the description of the EEPROM Protection Register (FEPROT).
0x0_040D	1	Flash nonvolatile option byte. Refer to the description of the Flash Option Register (FOPT).
0x0_040C	1	Flash security byte. Refer to the description of the Flash Security Register (FSEC).

If the Kinetis device is not locked, so the **Flash Security Byte (FSEC)** that is set to **0xFE**, but the other protections bytes are not set to **0xFF**, the **#TPCMD** MASSERASE F command may fail.

NXP KINETIS Flash Masserase Flash and Unlock

If you execute the **#TPCMD** MASSERASE F command after the **#TPCMD** CONNECT command, it may happen that **#TPCMD** MASSERASE F fails.

This is because **Security Bytes** could be set in the Flash (Security Bytes are other bytes than the Flash Security Byte).

It is therefore recommended to always execute the **#TPCMD** MASSERASE UNLOCK command before the **#TPCMD** MASSERASE F command.

What happens is that with the **#TPCMD** MASSERASE UNLOCK command all the Flash bytes are set to **0xFF** except the value of the **Flash Security Byte** which is set to **0xFE**.

Then if **#TPCMD** MASSERASE F command is then executed, the **Flash Security Byte** is also set to **0xFF**.
So, at this point, all the Flash memory is in the erased state.

Please pay attention now.
If you perform a power off and power on of the board, the **Flash Security Byt**e is set to **0xFF**, then the Flash will be secure and therefore to re-enter the device it will be necessary to re-set this byte back to **0xFE** (this is done automatically by the **#TPCMD** CONNECT command).

NXP KINETIS Flash Blankcheck

When you run the `#TPCMD BLANKCHECK F` command you are trying to check if the Flash memory is in the erased state.

By implementation choice, the `#TPCMD BLANKCHECK F` command gives PASS both if the **Flash Security Byte FSEC** is equal to **0xFF** or if it is equal to **0xFE**.
All other bytes of Flash memory must be equal to **0xFF**.

NXP KINETIS CPU Speed Calculation

It is necessary for some specific Kinetis devices to calculate the working frequency of the core to correctly set a specific value in the Flash controller.

This is done empirically via a special procedure when executing the `#TPCMD CONNECT` command.
Below you will find an example application case:

```
---#TPCMD CONNECT
Protocol selected SWD.
Entry Clock is 1.00 MHz.
Trying Hot Plug connect procedure.
IDCODE: 0x0BC11477.
Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
ID-Code read correctly at 1.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[0] IDR: 0x04770031, Type: AMBA AHB3 bus.
AP[1] IDR: 0x001C0020, Type: JTAG connection.
AP[0] ROM table base address 0xF0002000.
CPUID: 0x410CC600.
Implementer Code: 0x41 - [ARM].
Found Cortex M0+ revision r0p0.
Cortex M0+ Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 4 [Range 4-7].
IDCODE: 0x0BC11477.
Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
ID-Code read correctly at 37.50 MHz.
Device configuration:
 * System Reset Status and ID Register: 0x02140400.
   * Family ID: 0x0 - KE0x family.
   * Sub-family ID: 0x2.
   * Revision number: 0x1.
   * Pin ID: 0x4 - 32-pin.
 * Universally Unique Identifier:
   * UID: 0x193D830630314534.
Calculated frequency 15986 KHz, expected 16000 KHz [Δ -0.087502%].
Assuming bus frequency of ~16MHz [0x0F].
Time for Connect: 0.350 s.
>|
```

NXP KINETIS FlexNVM Partitioning

The basic idea is to divide the FlexNVM memory into two distinct areas.

The first area is always used for data use in which a single copy of the data is saved in each memory address (`Data Flash`).
The second area into which the secondary flash is divided is called EEprom (`EEprom Backup`).

In reality it is always of flash technology being a part of Data Flash but the hardware behaviour of the EEprom is emulated.
In fact, the Flash technology allows the rewriting of the memory cell of about 10'000 times while the EEprom technology allows the rewriting of 1'000'000 times of the data.

Therefore, to obtain this result in the secondary memory used (after the partitioning command) for EEprom use, a data is saved and copied several times of the EEprom backup memory.

In this way an algorithm manages wear-levelling and it is possible to have the duration of a true EEprom.
To divide the memory with the partitioning command it is necessary to specify specific parameters where the dimensions of the areas are indicated.

These parameters must be appropriately chosen so that the area reserved as data flash can contain all the data present in the source image file. The remaining free space can be used as a EEprom Backup.

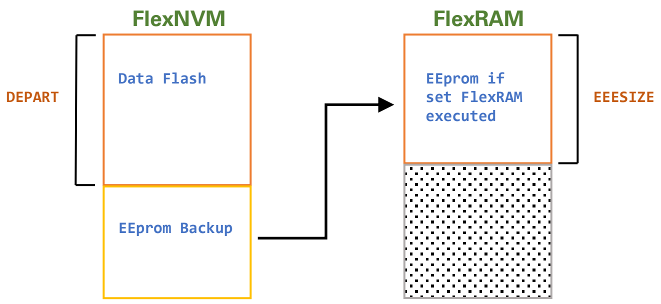
Kinetis FlexRAM Partitioning Special Commands

```
#TPCMD SET_ FLEXRAM <RAM|EEPROM>
```

The Set FlexRAM command changes the function of the FlexRAM:

- When not partitioned for EEPROM, the FlexRAM is typically used as traditional RAM.
- When partitioned for EEPROM, the FlexRAM is typically used to store EEPROM data.

```
#TPCMD SET_PARTITION <FlexRAM not loaded [1]/loaded [0]> <EEPROM Data Size> <EEPROM-backup Size>
#TPCMD REMOVE_PARTITION
```



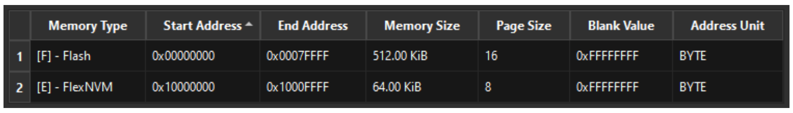
To handle varying customer requirements, the FlexRAM and FlexNVM blocks can be split into partitions:

1. **EEPROM partition (EEESIZE)** — The amount of FlexRAM used for EEPROM can be set from 0 Bytes (no EEPROM) to the maximum FlexRAM size.
   The remainder of the FlexRAM not used for EEPROM is not accessible while the FlexRAM is configured for EEPROM.
   The EEPROM partition grows upward from the bottom of the FlexRAM address space.

2. **Data flash partition (DEPART)** — The amount of FlexNVM memory used for data flash can be programmed from 0 bytes (all of the FlexNVM block is available for EEPROM backup) to the maximum size of the FlexNVM block.

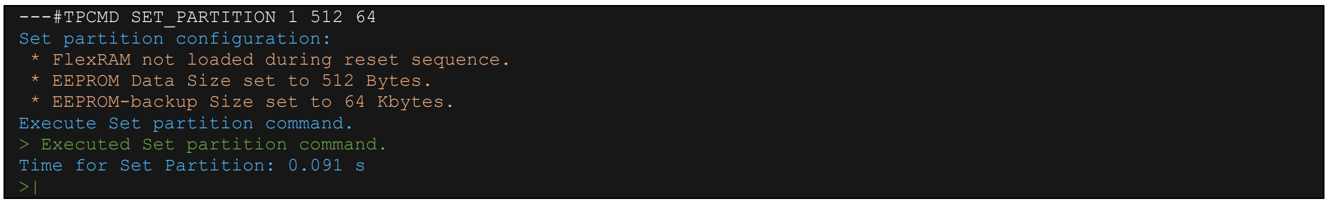
3. **FlexNVM EEPROM partition** — The amount of FlexNVM memory used for EEPROM backup, which is equal to the FlexNVM block size minus the data flash memory partition size.
   The EEPROM backup size must be at least 16 times the EEPROM partition size in FlexRAM.

Kinetis FlexNVM Partitioning Example



	Memory Type	Start Address	End Address	Memory Size	Page Size	Blank Value	Address Unit
1	[F] - Flash	0x00000000	0x0007FFFF	512.00 KiB	16	0xFFFFFFFF	BYTE
2	[E] - FlexNVM	0x10000000	0x1000FFFF	64.00 KiB	8	0xFFFFFFFF	BYTE

```
FlexRAM not loaded     [1]
EEPROM Data Size       [512]
EEPROM-backup Size     [64]
```



```
---#TPCMD SET_PARTITION 1 512 64
Set partition configuration:
 * FlexRAM not loaded during reset sequence.
 * EEPROM Data Size set to 512 Bytes.
 * EEPROM-backup Size set to 64 Kbytes.
Execute Set partition command.
> Executed Set partition command.
Time for Set Partition: 0.091 s
>|
```

We use the **#TPCMD** GET_DEVICE_INFORMATIONS to get the new device configuration:

```
---#TPCMD GET_DEVICE_INFORMATIONS
Device configuration:
 * System Device Identification Register: 0x14270087.
   * Family ID: 0x1 - KE1x Family (Enhanced features).
   * Sub-family ID: 0x4.
   * Series ID: 0x2 - Kinetis E+ series.
   * Ram Size: 0x7 - 64KB.
   * Revision number: 0x0.
   * Project ID: 0x01.
   * Pin ID: 0x07 - 100-pin count.
 * Flash Configuration Register: 0x5B054000.
   * Flash memory size: 512 KB. Protection region size: 16 KB.
   * FlexNVM size: 64KB.
   * EEE SRAM data size: 512 Bytes.
   * Depart set to 0x4 - 0 KBytes Data Flash.
   * Flash access is enabled.
   * FlexRAM is not available as normal RAM.
Time for Get Device Information: 0.003 s
>|
```

We use the **#TPCMD** SET_PARTITION to change device partition:

FlexRAM loaded [0]
EEPROM Data Size [64]
EEPROM-backup Size [32]

```
---#TPCMD SET_PARTITION 0 64 32
Set partition configuration:
 * FlexRAM loaded with valid EEPROM data during reset sequence.
 * EEPROM Data Size set to 64 Bytes.
 * EEPROM-backup Size set to 32 Kbytes.
FlexNVM is already divided into Data Flash and EEprom Backup.
We need to remove partition before set the new one.
To remove partition, we need to use Erase All Blocks command.
This operation erases Flash memory, initialize FlexRAM, verifies all memory contents, releases MCU security.
> Executed Remove partition command.
Execute Set partition command.
> Executed Set partition command.
Time for Set Partition: 0.192 s
>|
```

We use the **#TPCMD** GET_DEVICE_INFORMATIONS to get the new device configuration:

```
---#TPCMD GET_DEVICE_INFORMATIONS
Device configuration:
 * System Device Identification Register: 0x14270087.
   * Family ID: 0x1 - KE1x Family (Enhanced features).
   * Sub-family ID: 0x4.
   * Series ID: 0x2 - Kinetis E+ series.
   * Ram Size: 0x7 - 64KB.
   * Revision number: 0x0.
   * Project ID: 0x01.
   * Pin ID: 0x07 - 100-pin count.
 * Flash Configuration Register: 0x5B083000.
   * Flash memory size: 512 KB. Protection region size: 16 KB.
   * FlexNVM size: 64KB.
   * EEE SRAM data size: 64 Bytes.
   * Depart set to 0x3 - 32 KBytes Data Flash.
   * Flash access is enabled.
   * FlexRAM is not available as normal RAM.
Time for Get Device Information: 0.003 s
>|
```

We use the **#TPCMD** REMOVE_PARTITION to remove the partition already created:

```
---#TPCMD REMOVE_PARTITION
To remove partition, we need to use Erase All Blocks command.
This operation erases Flash memory, initialize FlexRAM, verifies all memory contents, releases MCU security.
> Executed Remove partition command.
Time for Remove Partition: 0.162 s
>|
```

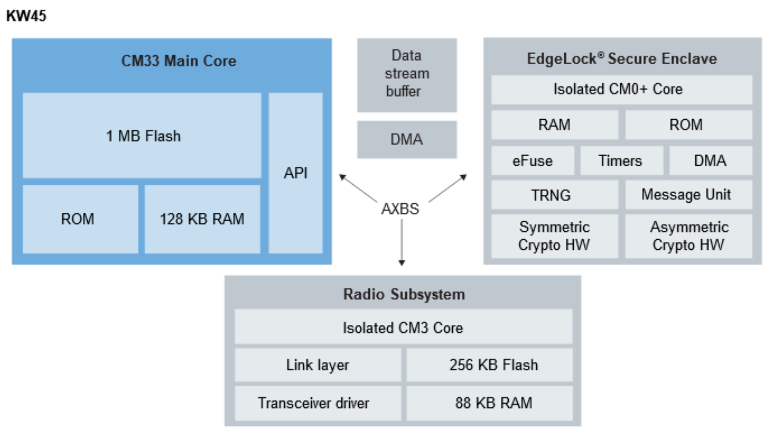
NXP KINETIS KW45 Devices

Kinetis KW45’s three-core architecture integrates a 96 MHz CM33 application core, dedicated CM3 radio core and an isolated EdgeLock Secure Enclave.
The Flash-based radio core with dedicated SRAM delivers a highly configurable and upgradeable software-implemented radio, freeing resources on the main core for customer application space.

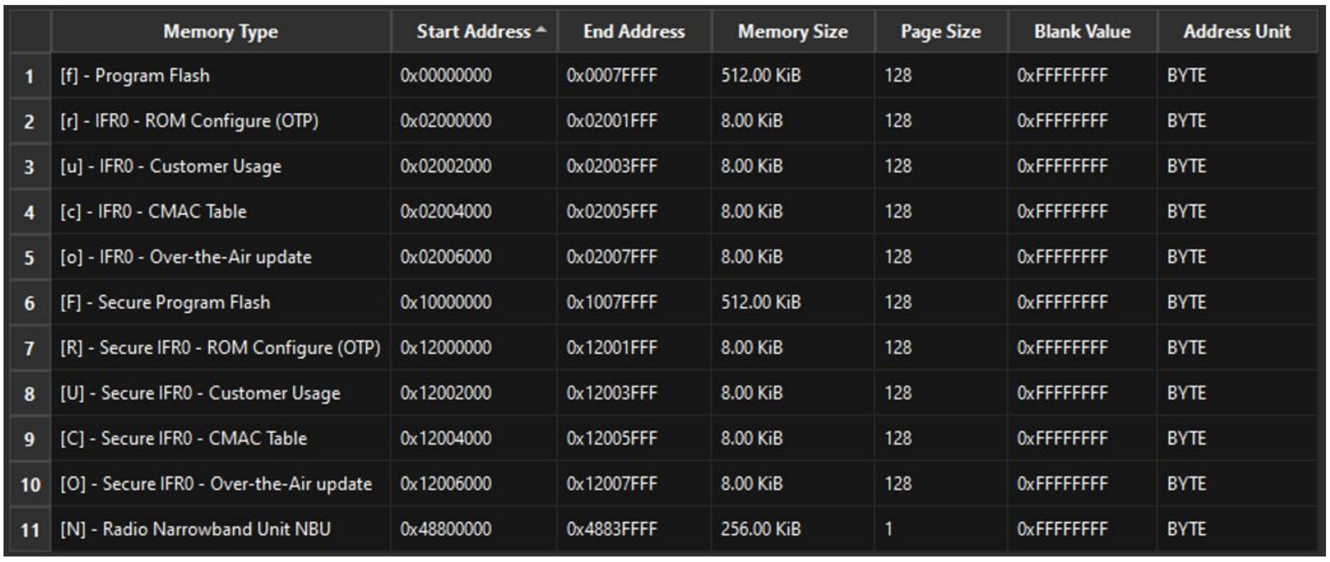
The Bluetooth Low Energy 5.3-compliant radio supports up to 24 simultaneous secure connections.
The EdgeLock Secure Enclave’s isolated execution environment provides a set of cryptographic accelerators, key store operations and secure lifecycle management that minimizes main core security responsibilities.

The KW45 MCU additionally integrates FlexCAN, helping enable seamless integration into an automobile’s in-vehicle or industrial CAN communication network.
The FlexCAN module can support CAN’s flexible data rate (CAN FD) for increased bandwidth and lower latency.

NXP KINETIS KW45 Architecture



NXP KINETIS KW45 Memory Map



	Memory Type	Start Address ▲	End Address	Memory Size	Page Size	Blank Value	Address Unit
1	[f] - Program Flash	0x00000000	0x0007FFFF	512.00 KiB	128	0xFFFFFFFF	BYTE
2	[r] - IFR0 - ROM Configure (OTP)	0x02000000	0x02001FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
3	[u] - IFR0 - Customer Usage	0x02002000	0x02003FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
4	[c] - IFR0 - CMAC Table	0x02004000	0x02005FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
5	[o] - IFR0 - Over-the-Air update	0x02006000	0x02007FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
6	[F] - Secure Program Flash	0x10000000	0x1007FFFF	512.00 KiB	128	0xFFFFFFFF	BYTE
7	[R] - Secure IFR0 - ROM Configure (OTP)	0x12000000	0x12001FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
8	[U] - Secure IFR0 - Customer Usage	0x12002000	0x12003FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
9	[C] - Secure IFR0 - CMAC Table	0x12004000	0x12005FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
10	[O] - Secure IFR0 - Over-the-Air update	0x12006000	0x12007FFF	8.00 KiB	128	0xFFFFFFFF	BYTE
11	[N] - Radio Narrowband Unit NBU	0x48800000	0x4883FFFF	256.00 KiB	1	0xFFFFFFFF	BYTE

NXP KINETIS KW45 Available ISP commands

ISP commands availability is controlled by lifecycle and token.
Table below shows the ISP commands available for each lifecycle. Details of several commands can be found in sections below.

Command	Description	Before OEM Open	At OEM Open	After OEM Open
Reset	Reset the device	Available	Available	Available
get-property <tag>	Query about various properties and settings	Available	Available	Available
set-property <tag> <value>	Change properties or options in ROM Bootloader	Available	Available	Available
receive-sb-file <file>	Receive a file in Secure Binary (SB) format	Available	Available	Available
flash-erase-region	Erase one or more sectors of the flash memory	Available	With limitation for radio flash	NA
flash-erase-all [<memoryID>]	Erase the entire flash memory specified by memoryID	Available	With limitation for radio flash	NA
read-memory <addr> <byte_count> [<file>]	Read memory at specified address	Available	With limitation for radio flash	NA
write-memory <addr> [<file> | {{<hex-data>}}]	Write memory at specified address from file or string of hex values Note: When write to SRAM, make sure the length of file or hex-data is 4-byte aligned	Available	With limitation for radio flash	NA
fill-memory <addr> <byte_count> <pattern> [word | short | byte]	Fill memory with pattern; pattern size can be word(default), short or byte	Available	With limitation for radio flash	NA
fuse-program <index> [<file> | {{<hex-data>}}]	Program fuse at the specified index from file or string of hex values	Available	Available	NA
fuse-read <index> <byte_count> [<file>]	Read fuse from the specified index	Available	Available	NA
Execute <address> <arg> <stackpointer>	Jumps to code at the provided address and does not return to the ROM bootloader	Available	Available	NA

NXP KINETIS KW45 Available Properties

Table below shows the available properties (tag) for this device.

Name	Writable	Tag value	Size in Bytes	Description
CurrentVersion	no	1	4	The current bootloader version.
AvailablePeripherals	no	2	4	The set of peripherals supported on this chip.
FlashStartAddress	no	3	4	Start address of program flash.
FlashSizeInBytes	no	4	4	Program flash size in bytes.

FlashSectorSize	no	5	4	The size of one sector of program flash in bytes.
FlashBlockCount	no	6	4	The number of blocks of the on-chip flash.
AvailableCommands	no	7	4	The set of commands supported by the bootloader.
CRCCheckStatus	no	8	4	The status of the application CRC check.
VerifyErase	yes	10	4	Controls whether the bootloader verifies erase to flash. The VerifyErase feature is enabled by default: 1 = Enable 0 = Disable
MaxPacketSize	no	11	4	Maximum supported packet size for the currently active peripheral interface.
ReservedRegions	no	12	4	List of memory regions reserved by the bootloader. Returned as value pairs (<startaddress-ofregion>, <end-addressof-region>).
RAMStartAddress	no	14	4	Start address of RAM
RAMSizeInBytes	no	15	4	RAM size in bytes.
SystemDeviceId	no	16	4	System Device ID
SecurityState	no	17	4	Security status
UniqueDeviceId	no	18	16	Unique device identification
BootStatus	no	20	4	Value of Boot Status Register
LoadableFWVersion	no	21	4	SSS loadable firmware version
FuseProgramVoltage	yes	22	4	Control the System LDO VDD Regulator Voltage Level. To program fuse, System LDO VDD regulator level needs to be regulated to Over Drive Voltage (2.5 V). The default System LDD VDO Regulator Voltage Level is regulated to Normal Voltage (1.8 V). 0 = System LDO VDO Regulator Voltage Level is related to Normal Voltage (1.8 V) 1 = System LDO VDD regulator level is regulated to Over Drive Voltage (2.5 V)
TargetVersion	no	24	4	Target version

NXP KINETIS KW45 Available eFUSE

Name	Fuse index	Description
LIFECYCLE	0x0A	Lifecycle state. Corresponding product state can be determined from Table 1. Lifecycle States
DBG_EN_LOCK	0x0B	Debug Enable Lock 0b - The debug access control registers remain open when jumping to customer code. 1b - The debug access control registers are write-locked before jumping to customer code.
DBG_AUTH_DIS	0x0C	Debug Authentication Disabled 0b - Debug Authentication enabled. 1b - Debug Authentication disabled.
TZM_EN	0x0D	Trust Zone Mode Enable 0b - TZ-M is disabled by default. 1b - TZ-M is enabled.
SERIAL_DIS	0x11	Serial Download Disabled 0b - ISP path is enabled. 1b - ISP path is disabled.
WAKEUP_DIS	0x12	Wakeup Disabled 0b - Boot-ROM LP wakeup is enabled. 1b - Boot-ROM LP wakeup is disabled.
CUST_PROD_OEMFW_AUTH_PUK_REVOKE[3:0]	0x13	Key revocation indicators of CUST_PROD_OEMFW_AUTH_PUK.
DBG_AUTH_VU[15:0]	0x15	These Debug Authentication Vendor Usage fuses are used by customers to restrict debug certificates.
IMG_KEY_REVOKE[15:0]	0x16	These fuses are used to revoke image signing keys separately from root keys.
SECURE_PHANTOM_CONFIG	0x18	Token configuration.
CUST_PROD_OEMFW_AUTH_PUK	0x1f	256-bit RoTKTH (customer trusted root key) typically used for CM33 main flash image authentication.
CUST_PROD_OEMFW_ENC_SK	0x20	256-bit encryption key used to protect confidentiality of OEM firmware. Typically required for firmware updates using sb3 (SB3KDK).
OEM_Enablement_Token	0x21	256-bit RoTKTH (root key hash) typically used for radio firmware authentication. (if valid token is intended to be used by programming SECURE_PHANTOM_CONFIG fuse)
DCFG_CC_SOCU_L1[8:0]	0x22	Debug authentication configuration for level 1 customer.
DCFG_CC_SOCU_L2[8:0]	0x23	Debug authentication configuration for level 2 customer.

SOC_VER_CNT	0x24	512-bits of version counter for various software components in SoC. Each software component version counter has dedicated fuse index 0x25 to 0x29.
CM33_S_VER_CNT	0x25	64-bit secure OEM CM33 firmware version counter (number of set fuse bits).
CM33_NS_VER_CNT	0x26	256-bit non-secure OEM CM33 firmware version counter (number of set fuse bits).
RADIO_VER_CNT	0x27	128-bit radio firmware version counter (number of set fuse bits).
SNT_VER_CNT	0x28	32-bit s200 loadable firmware version counter (number of set fuse bits).
CM33_BOOTLOADER_VER_CNT	0x29	Reserved for future usage of additional NXP software deliverable.
CM33_S_VER_CNT_VIRTUAL	0x2A	64-bit secure OEM CM33 firmware version counter (Integer value equivalent to number of set fuse bits).
CM33_NS_VER_CNT_VIRTUAL	0x2B	256-bit non-secure OEM CM33 firmware version counter (Integer value equivalent to number of set fuse bits).
RADIO_VER_CNT_VIRTUAL	0x2C	128-bit radio firmware version counter (Integer value equivalent to number of set fuse bits).
SNT_VER_CNT_VIRTUAL	0x2D	32-bit s200 loadable firmware version counter (Integer value equivalent to number of set fuse bits).
CM33_BOOTLOADER_VER_CNT_VIRTUAL	0x2E	Reserved for future usage of additional NXP software deliverable.

NXP KINETIS KW45 Specific Commands

One of the most important features of the Kinetis KW45 device is that it is not possible to perform all operations using only the **SWD** protocol.
In fact, it is also necessary to use the **UART** protocol.
Through FlashRunner you can use the **UART** and **SWD** simultaneously in the same channel.

Special commands have therefore been added for the KW45 device.

Commands for switch from SWD to UART and vice versa:

```
#TPCMD SWITCH_TO_SWD
#TPCMD SWITCH_TO_UART
```

UART commands for read and write properties:

```
#TPCMD UART_READ_PROPERTY <Property ID>
#TPCMD UART_WRITE_PROPERTY <Property ID> <Property Value>
```

UART commands for erase memory regions:

```
#TPCMD UART_FLASH_ERASE_ALL <INTERNAL_FLASH|RADIO_PFLASH|RADIO_USER_IFR>
#TPCMD UART_FLASH_ERASE_REGION <Address> <Byte Count> <INTERNAL_FLASH|RADIO_PFLASH|RADIO_USER_IFR>
```

UART commands for read memory regions:

```
#TPCMD UART_READ_MEMORY <Address> <Byte Count> <INTERNAL_FLASH|RADIO_PFLASH|RADIO_USER_IFR>
#TPCMD UART_WRITE_MEMORY <Address> <INTERNAL_FLASH|RADIO_PFLASH|RADIO_USER_IFR> <Data Word 0 - 15>
```

UART commands for read or write eFUSE:

```
#TPCMD UART_READ_FUSE <Address>
```

```
#TPCMD UART_WRITE_FUSE <Address> <Data ..> <Data ..>
#TPCMD UART_VERIFY_FUSE <Address> <Data ..> <Data ..>
```

SWD commands for read or write eFUSE:

```
#TPCMD READ_FUSE <Address>
#TPCMD WRITE_FUSE <Address> <Data ..> <Data ..>
```

UART command for special SBF procedure:

```
#TPCMD UART_SEND_SBF_FILE
```

#TPCMD SWITCH_TO_UART

Syntax: `#TPCMD SWITCH_TO_UART`

Prerequisites: none

Description: This command is used to switch from SWD to UART protocol

Examples: Correct command execution: 😊

```
---#TPCMD SWITCH_TO_UART
Attempt to connect via UART protocol (460800bps).
 * Set Boot Config Pin High to Enter ISP Mode.
 * Shuffle DIO: shuffling 2->4, 5->3.
 * Reset Device through RESET Pin (DIO1).
 * Send Init command to target device.
 * Device response correctly to Init command.
Time for Switch to UART: 0.112 s.
```

Skipped command execution: 😐

```
---#TPCMD SWITCH_TO_UART
Skipped: Fpga is already configured to use UART protocol.
Time for Switch to UART: 0.001 s.
```

#TPCMD SWITCH_TO_SWD

Syntax: `#TPCMD SWITCH_TO_SWD`

Prerequisites: none

Description: This command is used to switch from UART to SWD protocol

Examples: Correct command execution: 😊

```
---#TPCMD SWITCH_TO_SWD
Attempt to switch to SWD protocol.
* Set Boot Config Pin Low to Exit ISP Mode.
* Shuffle DIO: shuffling 4->2, 3->5.
* Reset Device through RESET Pin (DIO1).
Try to reconnect using SWD protocol.
Protocol selected SWD.
Entry Clock is 1.00 MHz.
Trying forcing hardware reset high procedure.
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 1.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[00] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[02] IDR: 0x002A0000, Type: JTAG connection.
AP[29] IDR: 0x54770002, Type: AMBA APB2 or APB3 bus.
AP[30] IDR: 0x24770011, Type: AMBA AHB3 bus.
AP[31] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[0] ROM table base address 0xE00FF000.
Used Debugger Mailbox to unlock AP buses.
 * Unlock Core M33 AP0 bus through NXP Debug Authentication Protocol.
 * Core M33 AP0 bus unlocked.
```

```
CPUID: 0x410FD214.
Implementer Code: 0x41 - [ARM].
Found Cortex M33 revision r0p4.
Cortex M33 Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 5-7].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Device Unique ID registers:
 * UID0: 0x4B4B3839, UID1: 0x00005030, UID2: 0x00000002, UID3: 0x00060028.
Device lifecycle configuration: [0x00000407].
 * Serial Wire Debug Instance ID: 0x0.
 * Revocation indicator from OEM Firmware Authentication Public Key: 0x0.
 * Boot-ROM LP wakeup is enabled.
 * Serial download path is enabled.
 * DICE is disabled.
 * Trust Zone Mode is disabled.
 * Debug Authentication is enabled.
 * The debug access control registers remain open when jumping to customer code.
 * Converged Lifecycle is [OEM Open].
Check device memories from ROM API:
 * Flash memory base address: 0x00000000.
 * Flash memory block count: 1.
 * Flash memory total size: 0x00100000 - 1024 KiB.
 * Flash IFR0 memory base address: 0x02000000.
 * Flash IFR0 memory total size: 0x00008000 - 32 KiB.
 * NBU memory base address: 0x48800000.
 * NBU memory block count: 1.
 * NBU memory total size: 0x00040000 - 256 KiB.
 * NBU IFR0 memory base address: 0x48840000.
 * NBU IFR0 memory total size: 0x00008000 - 32 KiB.
Time for Switch to SWD: 0.153 s.
```

Skipped command execution: 😐

```
---#TPCMD SWITCH_TO_SWD
Skipped: Fpga is already configured to use SWD protocol.
Time for Switch to SWD: 0.001 s.
```

#TPCMD UART_READ_PROPERTY

Syntax: `#TPCMD UART_READ_PROPERTY <Property ID>`

`<Property ID>`	Property ID, see Property table above

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: This command is used to read selected property from KW45 device
The Read Property command is used to query the bootloader about various properties and settings. Each supported property has a unique 32-bit tag associated with it.
Properties are the defined units of data that can be accessed with the Read Property or Write Property commands. Properties may be read-only or read-write.
The 32-bit property tag is the only parameter required for Read Property command

Examples: Correct command execution: 😊

```
---#TPCMD UART_READ_PROPERTY 18
Property ID 18: 28EB5DD91CB588089694FECA508CBE8E.
Time for UART Read Property: 0.003 s.
```

#TPCMD UART_WRITE_PROPERTY

Syntax: `#TPCMD UART_WRITE_PROPERTY <Property ID> <Property Value>`

`<Property ID>`	Property ID, see Property table above
`<Property Value>`	Property Value, see Property table above

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: The Write Property command is used to change or alter the values of the properties or options of the bootloader.
The command accepts the same property tags used with the Read Property command.
However, only some properties are writable.
If an attempt to write a read-only property is made, an error is returned indicating the property is read-only and cannot be changed.

Examples: Correct command execution: 😊

```
---#TPCMD UART_WRITE_PROPERTY 0x16 0x00
Time for UART Write Property: 0.002 s.
```

#TPCMD UART_FLASH_ERASE_ALL

Syntax: `#TPCMD UART_FLASH_ERASE_ALL <Memory ID>`

`<Memory ID>`	Memory ID can be `INTERNAL_FLASH`, `RADIO_PFLASH` or `RADIO_USER_IFR`

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: The Flash Erase All command performs an erase of the entire flash memory.
If any flash regions are protected, then the Flash Erase All command fails and returns an error status code.
The Flash Erase All command requires memory ID. If memory ID is not specified, the internal flash (`INTERNAL_FLASH`) will be selected as default.

Examples: Correct command execution: 😊

```
---#TPCMD UART_FLASH_ERASE_ALL
Time for UART Flash Erase All: 0.184 s.
```

#TPCMD UART_FLASH_ERASE_REGION

Syntax: `#TPCMD UART_FLASH_ERASE_REGION <Address> <Byte Count> <Memory ID>`

`<Address>`	Memory address aligned to 32-byte
`<Byte Count>`	Size aligned to 32-byte
`<Memory ID>`	Memory ID can be `INTERNAL_FLASH`, `RADIO_PFLASH` or `RADIO_USER_IFR`

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: The Flash Erase Region command performs an erase of one or more sectors of the flash memory.
The start address, and number of bytes are the 2 parameters required for the Flash Erase Region command.
The start and byte count parameters must be 32-byte aligned ([3:0] = 0000) and region specified does fit in the flash memory space or the Flash Erase Region command fails.
If any part of the region specified is protected, the Flash Erase Region command fails.

Examples: Correct command execution: 😊

```
---#TPCMD UART_FLASH_ERASE_REGION 0x00000000 0x00000100 INTERNAL_FLASH
Time for UART Flash Erase Region: 0.003 s.
```

#TPCMD UART_READ_MEMORY

Syntax: `#TPCMD UART_READ_MEMORY <Address> <Byte Count> <Memory ID>`

`<Address>`	Memory address aligned to 32-byte
`<Byte Count>`	Size aligned to 32-byte
`<Memory ID>`	Memory ID can be `INTERNAL_FLASH`, `RADIO_PFLASH` or `RADIO_USER_IFR`

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: The Read Memory command returns the contents of memory at the given address, for a specified number of bytes. This command can read any region of memory accessible by the CPU and not protected by security. The start address, and number of bytes are the two parameters required for Read Memory command. `INTERNAL_FLASH` memory will be selected as default if memory ID is not specified.

Examples: Correct command execution: 😊

```
---#TPCMD UART_READ_MEMORY 0x00000000 0x00000100 INTERNAL_FLASH
Time for UART Read Memory: 0.010 s.
```

On the Terminal Tool you can see:

```
01|Read[0x00000000]: 0x00 0x40 0x00 0x20 0x41 0x00 0x00 0x00 | 0x41 0x00 0x00 0x00 0x41 0x00 0x00 0x00
01|Read[0x00000010]: 0x41 0x00 0x00 0x00 0x41 0x00 0x00 0x00 | 0x41 0x00 0x00 0x00 0x41 0x00 0x00 0x00
01|Read[0x00000020]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x00 0x00 0x00 0x00 0x41 0x00 0x00 0x00
01|Read[0x00000030]: 0x41 0x00 0x00 0x00 0x00 0x00 0x00 0x00 | 0x41 0x00 0x00 0x00 0x41 0x00 0x00 0x00
01|Read[0x00000040]: 0x2D 0xE9 0xF0 0x4D 0x81 0xB0 0x72 0xB6 | 0x99 0x48 0x9A 0x49 0x4F 0xF0 0x00 0x02
01|Read[0x00000050]: 0x4F 0xF0 0x00 0x03 0x4F 0xF0 0x00 0x04 | 0x4F 0xF0 0x00 0x05 0x3C 0xC0 0x88 0x42
01|Read[0x00000060]: 0xFC 0xD3 0x4F 0xF4 0x00 0x54 0x60 0x68 | 0x4B 0xF2 0x59 0x38 0xCF 0xF6 0x7C 0x28
01|Read[0x00000070]: 0x40 0x45 0x2F 0xD1 0x20 0x68 0xB0 0xF5 | 0x00 0x2F 0x2B 0xD8 0xE0 0x68 0xA1 0x68
01|Read[0x00000080]: 0x40 0x1A 0x00 0xF1 0x04 0x0A 0x51 0x46 | 0xA0 0x68 0x01 0x23 0x4F 0xF0 0xFF 0x32
01|Read[0x00000090]: 0x00 0xF0 0xD1 0xF8 0x05 0x46 0x20 0x69 | 0x85 0x42 0x1B 0xD1 0x61 0x69 0xE8 0x43
01|Read[0x000000A0]: 0x88 0x42 0x17 0xD1 0x4F 0xF0 0xFF 0x32 | 0x01 0x23 0x18 0x21 0x4F 0xF4 0x00 0x50
01|Read[0x000000B0]: 0x00 0xF0 0xC1 0xF8 0x05 0x46 0xA0 0x69 | 0x85 0x42 0x0B 0xD1 0xE1 0x69 0xE8 0x43
01|Read[0x000000C0]: 0x88 0x42 0x07 0xD1 0x26 0x68 0xD6 0xF8 | 0x04 0xB0 0x5F 0x46 0x30 0x68 0x80 0xF3
01|Read[0x000000D0]: 0x08 0x88 0xB8 0x47 0x4F 0xF4 0x00 0x34 | 0x60 0x68 0x4B 0xF2 0x59 0x38 0xCF 0xF6
01|Read[0x000000E0]: 0x7C 0x28 0x40 0x45 0x30 0xD1 0x20 0x68 | 0xB0 0xF5 0x00 0x2F 0x2C 0xD8 0xE0 0x68
01|Read[0x000000F0]: 0xA1 0x68 0x40 0x1A 0x00 0xF1 0x04 0x0A | 0x51 0x46 0xA0 0x68 0x01 0x23 0x4F 0xF0
```

#TPCMD UART_WRITE_MEMORY

Syntax: `#TPCMD UART_WRITE_MEMORY <Address> <Memory ID> <Data Word 0 - 15>`

`<Address>`	Memory address aligned to 32-byte
`<Memory ID>`	Memory ID can be `INTERNAL_FLASH`, `RADIO_PFLASH` or `RADIO_USER_IFR`
`<Data Word 0 - 15>`	Data/s to be programmed

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: The Write Memory command writes data provided in the `<Data Word 0 - 15>` to a specified range of bytes in memory. However, if flash protection is enabled, then writes to protected sectors fail.
Special care must be taken when writing to Flash:

- First, any Flash sector written to must have been previously erased with a Flash Erase All or Flash Erase Region.
- Writing to Flash requires the start address to be 16-byte aligned ([3:0] = 0000).
- The byte count is rounded up to a multiple of 4, and trailing bytes are filled with the flash erase pattern (0xFF).
- If the Verify Writes property is set to true, then writes to flash also performs a flash verify program operation.

The start address and number of bytes are the 2 parameters required for Write Memory

Examples: Correct command execution: 😊

```
---#TPCMD UART_FLASH_ERASE_REGION 0x00000000 0x00000100 INTERNAL_FLASH
Time for UART Flash Erase Region: 0.004 s.

---#TPCMD UART_WRITE_MEMORY 0x00000000 INTERNAL_FLASH 0xDEADBEEF
Time for UART Write Memory: 0.003 s.

---#TPCMD UART_READ_MEMORY 0x00000000 0x00000100 INTERNAL_FLASH
Time for UART Read Memory: 0.010 s.
```

On the Terminal Tool you can see:

```
01|Read[0x00000000]: 0xEF 0xBE 0xAD 0xDE 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000010]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000020]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000030]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000040]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000050]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000060]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
```

```
01|Read[0x00000070]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000080]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x00000090]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x000000A0]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x000000B0]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x000000C0]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x000000D0]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x000000E0]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
01|Read[0x000000F0]: 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF | 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
```

#TPCMD UART_READ_FUSE

Syntax: `#TPCMD UART_READ_FUSE <Address>`

`<Address>`	eFUSE address (see table above)

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: This command is used to read selected eFUSE using UART protocol

Examples: Correct command execution: 😊

```
---#TPCMD UART_READ_FUSE 0x2B
Fuse 0x2B word 0: 0x00000000.
Fuse 0x2B word 1: 0x00000000.
Fuse 0x2B word 2: 0x00000000.
Fuse 0x2B word 3: 0x00000000.
Fuse 0x2B word 4: 0x00000000.
Fuse 0x2B word 5: 0x00000000.
Fuse 0x2B word 6: 0x00000000.
Fuse 0x2B word 7: 0x00000000.
Time for UART Read Fuse: 0.007 s.
```

Correct command execution: 😊

```
---#TPCMD UART_READ_FUSE 0x0A
Fuse 0x0A word 0: 0x00000007.
Time for UART Read Fuse: 0.003 s.
```

#TPCMD UART_WRITE_FUSE

Syntax: `#TPCMD UART_WRITE_FUSE <Address> <Data ..> <Data ..>`

`<Address>`	eFUSE address (see table above)
`<Data ..> <Data ..>`	Data to be programmed

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: This command is used to write selected eFUSE using UART protocol

Examples: Correct command execution: 😊

```
---#TPCMD UART_WRITE_FUSE 0x1F 34568097079FF27A3E8A2DA14781B922 FD8295B6C00BFA067F00E87F1A168899
Time for UART Write Fuse: 0.003 s.
```

#TPCMD UART_VERIFY_FUSE

Syntax: `#TPCMD UART_VERIFY_FUSE <Address> <Data ..> <Data ..>`

`<Address>`	eFUSE address (see table above)
`<Data ..> <Data ..>`	Data to be verified

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: This command is used to verify selected eFUSE using UART protocol

Examples: Correct command execution: 😊

```
---#TPCMD UART_VERIFY_FUSE 0x1F 34568097079FF27A3E8A2DA14781B922 FD8295B6C00BFA067F00E87F1A168899
Fuse 0x1F word 0: Expected 0x97805634, Read 0x97805634.
Fuse 0x1F word 1: Expected 0x7AF29F07, Read 0x7AF29F07.
Fuse 0x1F word 2: Expected 0xA12D8A3E, Read 0xA12D8A3E.
Fuse 0x1F word 3: Expected 0x22B98147, Read 0x22B98147.
Fuse 0x1F word 4: Expected 0xB69582FD, Read 0xB69582FD.
Fuse 0x1F word 5: Expected 0x06FA0BC0, Read 0x06FA0BC0.
Fuse 0x1F word 6: Expected 0x7FE8007F, Read 0x7FE8007F.
Fuse 0x1F word 7: Expected 0x9988161A, Read 0x9988161A.
Time for UART Verify Fuse: 0.004 s.
```

#TPCMD READ_FUSE

Syntax: `#TPCMD READ_FUSE <Address>`

`<Address>`	eFUSE address (see table above)

Prerequisites: none

Description: This command is used to read selected eFUSE using SWD protocol

Examples: Correct command execution: ☺

```
---#TPCMD READ_FUSE 0x2B
Fuse 0x2B word 0: 0x00000000.
Fuse 0x2B word 1: 0x00000000.
Fuse 0x2B word 2: 0x00000000.
Fuse 0x2B word 3: 0x00000000.
Fuse 0x2B word 4: 0x00000000.
Fuse 0x2B word 5: 0x00000000.
Fuse 0x2B word 6: 0x00000000.
Fuse 0x2B word 7: 0x00000000.
Time for Read Fuse: 0.007 s.
```

Correct command execution: ☺

```
---#TPCMD READ_FUSE 0x0A
Fuse 0x0A word 0: 0x00000007.
Time for Read Fuse: 0.003 s.
```

#TPCMD WRITE_FUSE

Syntax: `#TPCMD WRITE_FUSE <Address> <Data ..> <Data ..>`

`<Address>`	eFUSE address (see table above)
`<Data ..> <Data ..>`	Data to be programmed

Prerequisites: none

Description: This command is used to write selected eFUSE using SWD protocol

Examples: Correct command execution: ☺

```
---#TPCMD WRITE_FUSE 0x1F 34568097079FF27A3E8A2DA14781B922 FD8295B6C00BFA067F00E87F1A168899
Time for Write Fuse: 0.003 s.
```

#TPCMD UART_SEND_SBF_FILE

Syntax: `#TPCMD UART_SEND_SBF_FILE`

Prerequisites: `#TPCMD SWITCH_TO_UART` executed correctly

Description: Send SBF file command is used to do image update on CM33 flash or radio (CM3) flash when security is enforced.
KW45 device receives a secure binary (SB) file, decrypt, authenticate and program the image to the target memory.

CUST_PROD_OEMFW_ENC_SK fuse needs to be programmed correctly to ensure this command can be used successfully.

Examples: Correct command execution: 😊

```
---#TPCMD UART_SEND_SBF_FILE
Get the maximum packet size from device:
 * Maximum packet size is 512 bytes.
Configure the device to Receive SBF file:
 * SBF file size is 0x000302C8 - 197320 bytes.
Start SBF transfer file:
 * 10% of SBF file transferred.
 * 20% of SBF file transferred.
 * 30% of SBF file transferred.
 * 40% of SBF file transferred.
 * 50% of SBF file transferred.
 * 60% of SBF file transferred.
 * 70% of SBF file transferred.
 * 80% of SBF file transferred.
 * 90% of SBF file transferred.
 * 100% of SBF file transferred.
Completed transfer of SBF file.
Time for UART Send SBF file: 7.558 s.
```

NXP KINETIS S9KEAZ/S9KEAZN Specific Commands

For S9KEAZ/S9KEAZN devices, a very useful special command has been added.

#TPCMD FREQUENCY_TRIM

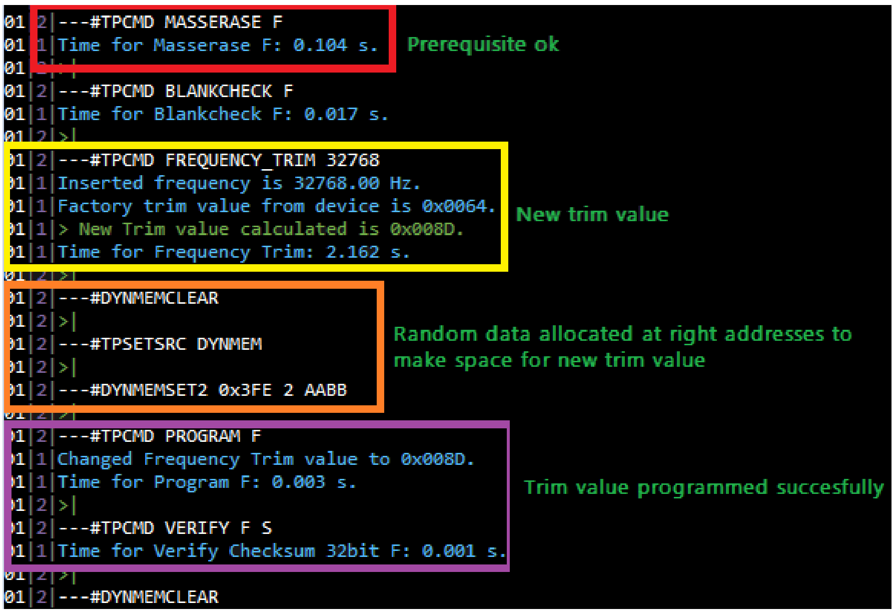
Syntax: `#TPCMD FREQUENCY_TRIM <Frequency[KHz]>`

Prerequisites: `#TPCMD MASSERASE F` executed correctly

Description: Allows the user to calculate the new trim value of the internal needed to achieve the desired <Frequency>.

Note: Allowed range for <Frequency> parameter: 31250 – 39062.50.
To flash the new trim value, it is sufficient to allocate space into the FRB file or programmer dynamic memory at addresses [0x3FE-0x3FF] (SCTRIM, SCFTRIM) together with the rest of the firmware and execute PROGRAM F command.

Examples: Correct command execution: 😊



```
01|2|---#TPCMD MASSERASE F
01|1|Time for Masserase F: 0.104 s.
01|2|>|
01|2|---#TPCMD BLANKCHECK F
01|1|Time for Blankcheck F: 0.017 s.
01|2|>|
01|2|---#TPCMD FREQUENCY_TRIM 32768
01|1|Inserted frequency is 32768.00 Hz.
01|1|Factory trim value from device is 0x0064.
01|1|> New Trim value calculated is 0x008D.
01|1|Time for Frequency Trim: 2.162 s.
01|2|>|
01|2|---#DYNMEMCLEAR
01|2|>|
01|2|---#TPSETSRC DYNMEM
01|2|>|
01|2|---#DYNMEMSET2 0x3FE 2 AABB
01|2|>|
01|2|---#TPCMD PROGRAM F
01|1|Changed Frequency Trim value to 0x008D.
01|1|Time for Program F: 0.003 s.
01|2|>|
01|2|---#TPCMD VERIFY F S
01|1|Time for Verify Checksum 32bit F: 0.001 s.
01|2|>|
01|2|---#DYNMEMCLEAR
```

Prerequisite ok
New trim value
Random data allocated at right addresses to make space for new trim value
Trim value programmed succesfully

NXP KINETIS Driver Parameters

The standard parameters are used to configure some specific options inside KINETIS driver.

#TCSETPAR ENTRY_CLOCK

Syntax:	`#TCSETPAR ENTRY_CLOCK <Frequency>`	
	`<Frequency>`	Accepted parameters 4000000, 2000000, 1000000, 500000, 100000 Hz
Description:	Set the JTAG/SWD frequency used in the Connect procedure before raising the PLL of the device, if the device PLL is available	
Note:	Default value 1.00 MHz	

#TCSETPAR PLL_ENABLED

Syntax:	`#TCSETPAR PLL_ENABLED <Value>`	
	`<Value>`	Accepted parameters YES / NO
Description:	Enable the PLL of the device at the highest possible frequency if it's available	
Note:	Default value YES	

#TCSETPAR RESET_HARDWARE

Syntax:	`#TCSETPAR RESET_HARDWARE <Value>`	
	`<Value>`	Accepted parameters YES / NO
Description:	Use Hardware reset (DIO1) into Connect procedure during halt Cortex Core Please leave this parameter to NO except when it is strictly necessary Usually, the Software Reset is enough to proceed with the reset of the device and to continue with the programming procedure	
Note:	Default value NO	

#TCSETPAR SAMPLING_POINT

Syntax:	`#TCSETPAR SAMPLING_POINT <Value>`	
	`<Value>`	Accepted values are in the range 1-15
Description:	Use this parameter to permanently set the sampling point of the FPGA It is recommended to leave this parameter with the default value	
Note:	Default value 17	

#TCSETPAR UART_CONNECT

Syntax:	`#TCSETPAR UART_CONNECT <Value>`	
	`<Value>`	Accepted parameters YES / NO
Description:	Connect to the KW45 device directly through UART protocol instead of SWD protocol	
Note:	Added from driver version **5.12** Default value NO	

#TCSETPAR UART_FREQUENCY

Syntax:	`#TCSETPAR UART_FREQUENCY <Frequency>`	
	<Frequency>	Accepted parameters 460800, 230400, 115200, 57600, 38400, 19200 or 9600 bps
Description:	Set the UART frequency used for KW45 devices	
Note:	Default value 460800 bps	

#TCSETPAR QSPI_PROTOCOL

Syntax:	`#TCSETPAR QSPI_PROTOCOL <Protocol>`	
	<Protocol>	Accepted parameters SPI or QUAD-SPI
Description:	Select if you want to communicate with the external memory via the SPI or QUAD-SPI protocol	
Note:	Default value SPI	

#TCSETPAR QSPI_FREQUENCY

Syntax:	`#TCSETPAR QSPI_FREQUENCY <Frequency>`	
	<Frequency>	Accepted parameters 25, 20, 15, 10, 5 or 1 MHz
Description:	Select the frequency to be used to communicate with the external memory via the SPI or QUAD-SPI protocol	
Note:	Default value 25 MHz	

NXP KINETIS Driver Commands

Here you can find the complete list of all available commands for KINETIS driver.

Memories for Generic Kinetis

```
F → Main Flash
E → FlexNVM or EEprom
X → External Memory
```

Memories for KW45 Kinetis

```
f → Program Flash
r → IFR0 ROM Configure (OTP)
u → IFR0 Customer Usage
c → IFR0 CMAC Table
o → IFR0 Over-the-Air update
F → Secure Program Flash
R → Secure IFR0 ROM Configure (OTP)
U → Secure IFR0 Customer Usage
C → Secure IFR0 CMAC Table
O → Secure IFR0 Over-the-Air update
N → Radio Narrowband Unit NBU
```

#TPCMD CONNECT

#TPCMD CONNECT

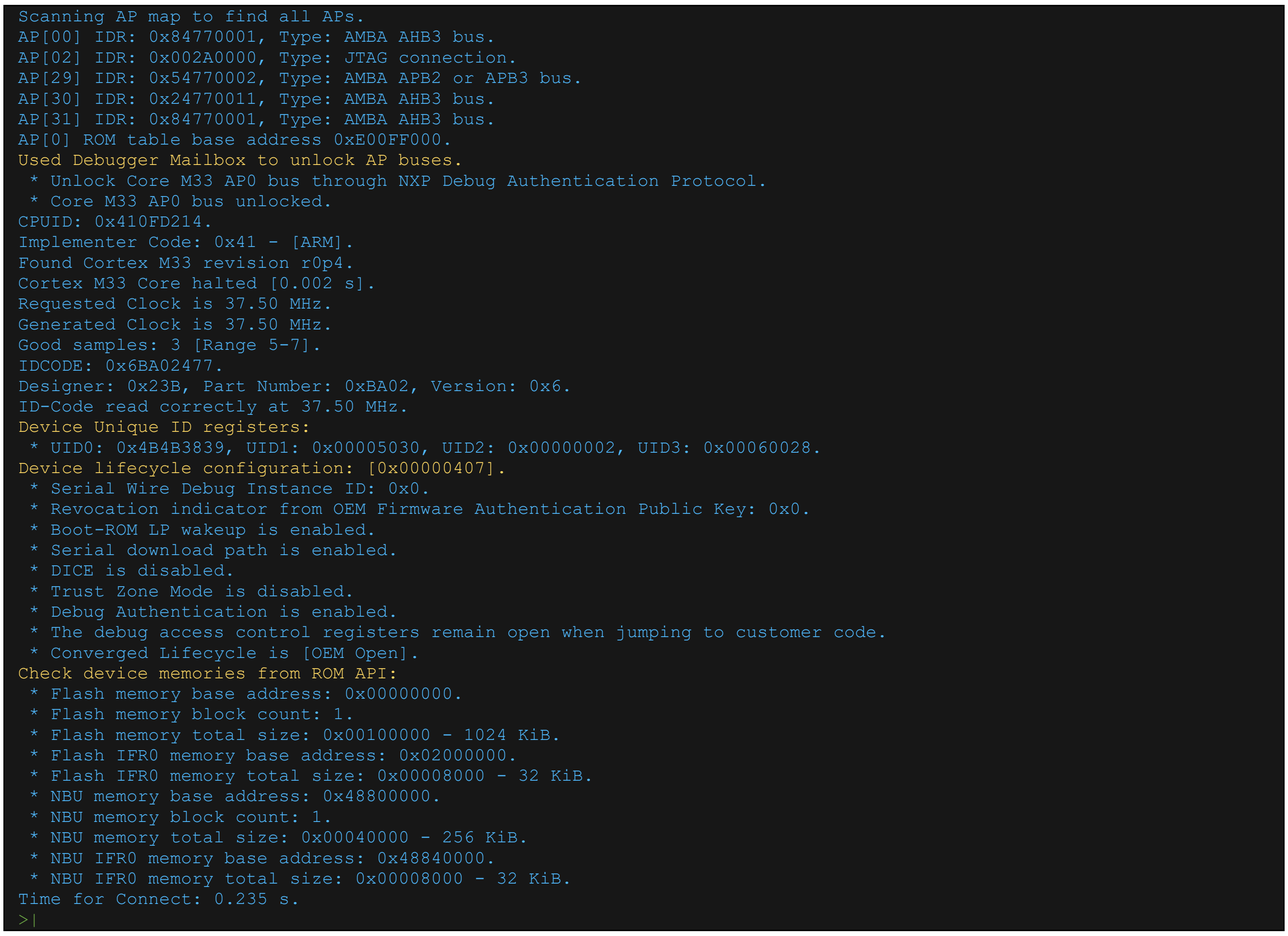
This function performs the entry and is the first command to be executed when starting the communication with the device.

Example for **MKE17Z256xxx7_1x_MX25L6433F**:

```
Protocol selected SWD.
Entry Clock is 1.00 MHz.
Trying Hot Plug connect procedure.
IDCODE: 0x0BC11477.
Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
ID-Code read correctly at 1.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[0] IDR: 0x04770031, Type: AMBA AHB3 bus.
AP[1] IDR: 0x001C0020, Type: JTAG connection.
AP[0] ROM table base address 0xF0002000.
CPUID: 0x410CC601.
Implementer Code: 0x41 - [ARM].
Found Cortex M0+ revision r0p1.
Cortex M0+ Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 4-6].
IDCODE: 0x0BC11477.
Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
ID-Code read correctly at 37.50 MHz.
Device configuration:
 * System Device Identification Register: 0x17270207.
   * Family ID: 0x1 - KE1x Family (Enhanced features).
   * Sub-family ID: 0x7.
   * Series ID: 0x2 - Kinetis E+ series.
   * Ram Size: 0x7 - 48KB.
   * Revision number: 0x0.
   * Project ID: 0x04.
   * Pin ID: 0x07 - 100-pin count.
 * Flash Configuration Register: 0x09000000.
   * Flash memory size: 256 KB. Protection region size: 8 KB.
   * Flash access is enabled.
Time for Connect: 0.107 s.
>|
```

Example for **KW45B41Z83**:

```
---#TPCMD CONNECT
Protocol selected SWD.
Entry Clock is 1.00 MHz.
Trying forcing hardware reset high procedure.
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 1.00 MHz.
JTAG-SWD Debug Port enabled.
```



```
Scanning AP map to find all APs.
AP[00] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[02] IDR: 0x002A0000, Type: JTAG connection.
AP[29] IDR: 0x54770002, Type: AMBA APB2 or APB3 bus.
AP[30] IDR: 0x24770011, Type: AMBA AHB3 bus.
AP[31] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[0] ROM table base address 0xE00FF000.
Used Debugger Mailbox to unlock AP buses.
 * Unlock Core M33 AP0 bus through NXP Debug Authentication Protocol.
 * Core M33 AP0 bus unlocked.
CPUID: 0x410FD214.
Implementer Code: 0x41 - [ARM].
Found Cortex M33 revision r0p4.
Cortex M33 Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 5-7].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Device Unique ID registers:
 * UID0: 0x4B4B3839, UID1: 0x00005030, UID2: 0x00000002, UID3: 0x00060028.
Device lifecycle configuration: [0x00000407].
 * Serial Wire Debug Instance ID: 0x0.
 * Revocation indicator from OEM Firmware Authentication Public Key: 0x0.
 * Boot-ROM LP wakeup is enabled.
 * Serial download path is enabled.
 * DICE is disabled.
 * Trust Zone Mode is disabled.
 * Debug Authentication is enabled.
 * The debug access control registers remain open when jumping to customer code.
 * Converged Lifecycle is [OEM Open].
Check device memories from ROM API:
 * Flash memory base address: 0x00000000.
 * Flash memory block count: 1.
 * Flash memory total size: 0x00100000 - 1024 KiB.
 * Flash IFR0 memory base address: 0x02000000.
 * Flash IFR0 memory total size: 0x00008000 - 32 KiB.
 * NBU memory base address: 0x48800000.
 * NBU memory block count: 1.
 * NBU memory total size: 0x00040000 - 256 KiB.
 * NBU IFR0 memory base address: 0x48840000.
 * NBU IFR0 memory total size: 0x00008000 - 32 KiB.
Time for Connect: 0.235 s.
>|
```

#TPCMD MASSERASE

#TPCMD MASSERASE <UNLOCK>

This command performs the internal unlocking routine which consists in 2 phases: permit access to the device and erase the content of all the programmable memories.
This is done only if the device is actually in secure state, otherwise the command will just pass

The reason behind this is that if a microcontroller is put in secured state, it means that the user does not want anyone to read the firmware's stored inside, so the internal erasure process prevents this, in the case unknown people gain access to the memory space through this command.

At the end of the routine, the **FSEC** byte is restored to its default value **0xFE** and the rest of the memories goes to **0xFF**.

#TPCMD MASSERASE <F|E|X>

This command is available for all Kinetis devices except the KW45.
This command performs a masserase for Main Flash, FlexNVM/EEprom or External Memory.

#TPCMD MASSERASE <f|u|c|o|F|U|C|O>

This command is available for only KW45 Kinetis devices.
This command performs a masserase for Program Flash, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.

#TPCMD ERASE

`#TPCMD ERASE <F|E>`
This command is available for all Kinetis devices except the KW45.
This command performs a sector/page erase for all Main Flash or FlexNVM/EEprom.

`#TPCMD ERASE <F|E> <start address> <size>`
This command is available for all Kinetis devices except the KW45.
This command performs a sector/page erase for selected part of Main Flash or FlexNVM/EEprom.

`#TPCMD ERASE <X> <64KB/32KB/4KB>`
This command is available for all Kinetis devices except the KW45.
This command performs a sector erase of 64KB/32KB/4KB for all External Memory.

`#TPCMD ERASE <X> <64KB/32KB/4KB> <start address> <size>`
This command is available for all Kinetis devices except the KW45.
This command performs a sector erase of 64KB/32KB/4KB for selected part of External Memory.
Enter the Start Address and Size in hexadecimal format.

`#TPCMD ERASE <f|u|c|o|F|U|C|O>`
This command is available for only KW45 Kinetis devices.
This command performs a sector/page erase for all Program Flash, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Enter the Start Address and Size in hexadecimal format.

`#TPCMD ERASE <f|u|c|o|F|U|C|O> <start address> <size>`
This command is available for only KW45 Kinetis devices.
This command performs a sector/page erase for selected part of Program Flash, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Enter the Start Address and Size in hexadecimal format.

#TPCMD BLANKCHECK

`#TPCMD BLANKCHECK <F|E|X>`
This command is available for all Kinetis devices except the KW45.
Blankcheck is only available for Main Flash, FlexNVM/EEprom and External Memory.
Verify if all memory is erased.

`#TPCMD BLANKCHECK <F|E|X> <start address> <size>`
This command is available for all Kinetis devices except the KW45.
Blankcheck is only available for Main Flash, FlexNVM/EEprom and External Memory.
Verify if selected part of memory is erased.
Enter the Start Address and Size in hexadecimal format.

`#TPCMD BLANKCHECK <f|r|u|c|o|F|R|U|C|O>`
This command is available for only KW45 Kinetis devices.
Blankcheck is only available for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Verify if all memory is erased.

`#TPCMD BLANKCHECK <f|r|u|c|o|F|R|U|C|O> <start address> <size>`
This command is available for only KW45 Kinetis devices.
Blankcheck is only available for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Verify if selected part of memory is erased.
Enter the Start Address and Size in hexadecimal format.

#TPCMD PROGRAM

#TPCMD `PROGRAM <F|E|X>`
This command is available for all Kinetis devices except the KW45.
Program available for Main Flash, FlexNVM/EEprom and External Memory.
Programs all memory of the selected type based on the data in the FRB file.

#TPCMD `PROGRAM <F|E|X> <start address> <size>`
This command is available for all Kinetis devices except the KW45.
Program available for Main Flash, FlexNVM/EEprom and External Memory.
Programs selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

#TPCMD `PROGRAM <f|r|u|c|o|F|R|U|C|O>`
This command is available for only KW45 Kinetis devices.
Program available for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Programs all memory of the selected type based on the data in the FRB file.

#TPCMD `PROGRAM <f|r|u|c|o|F|R|U|C|O> <start address> <size>`
This command is available for only KW45 Kinetis devices.
Program available for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Programs selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

#TPCMD VERIFY

#TPCMD `VERIFY <F|E|X> <R|S>`
R: Readout Mode.
S: Checksum 32 Bit Mode.
This command is available for all Kinetis devices except the KW45.
Verify Readout/Checksum 32bit available for Main Flash, FlexNVM/EEprom and External Memory.
Verify all memory of the selected type based on the data in the FRB file.

#TPCMD `VERIFY <F|E|X> <R|S> <start address> <size>`
R: Readout Mode.
S: Checksum 32 Bit Mode.
This command is available for all Kinetis devices except the KW45.
Verify Readout/Checksum 32bit available for Main Flash, FlexNVM/EEprom and External Memory.
Verify selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

#TPCMD `VERIFY <f|r|u|c|o|F|R|U|C|O> <R|S>`
R: Readout Mode.
S: Checksum 32 Bit Mode.
This command is available for only KW45 Kinetis devices.
Verify Readout/Checksum 32bit available for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
Verify all memory of the selected type based on the data in the FRB file.

#TPCMD `VERIFY <f|r|u|c|o|F|R|U|C|O> <R|S> <start address> <size>`
R: Readout Mode.
S: Checksum 32 Bit Mode.
This command is available for only KW45 Kinetis devices.
Verify Readout/Checksum 32bit available for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.

Verify selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.
`#TPCMD VERIFY F M`
This command is available for S9KEAZN8 devices and it is used to margin verify the FLASH memory.

#TPCMD READ

`#TPCMD READ <F|E|X>`
`#TPCMD READ <F|E|X> <start address> <size>`
This command is available for all Kinetis devices except the KW45.
Read function for Main Flash, FlexNVM/EEprom and External Memory.
The result of the read command will be visible into the Terminal.

`#TPCMD READ <f|r|u|c|o|F|R|U|C|O>`
`#TPCMD READ <f|r|u|c|o|F|R|U|C|O> <start address> <size>`
This command is available for only KW45 Kinetis devices.
Read function for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
The result of the read command will be visible into the Terminal.

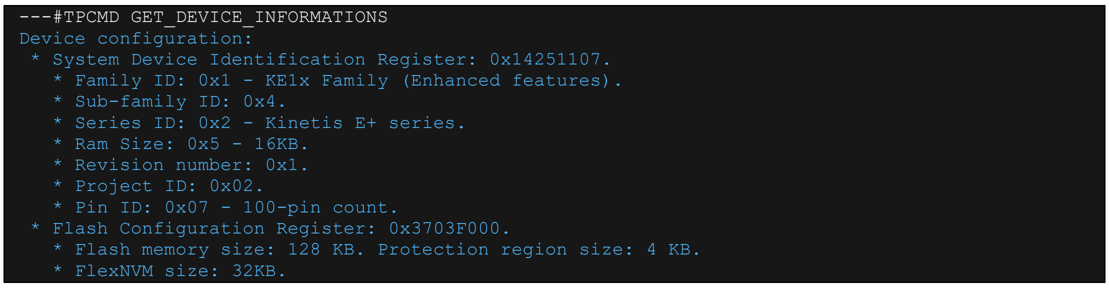
#TPCMD DUMP

`#TPCMD DUMP <F|E|X>`
`#TPCMD DUMP <F|E|X> <start address> <size>`
This command is available for all Kinetis devices except the KW45.
Dump function for Main Flash, FlexNVM/EEprom and External Memory.
The result of the dump command will be stored in the FlashRunner 2.0 internal memory.

`#TPCMD DUMP <f|r|u|c|o|F|R|U|C|O>`
`#TPCMD DUMP <f|r|u|c|o|F|R|U|C|O> <start address> <size>`
This command is available for only KW45 Kinetis devices.
Dump function for Program Flash, IFR0 ROM Configure OTP, IFR0 Customer Usage, IFR0 CMAC Table, IFR0 Over-The-Air update, Secure Program Flash, Secure IFR0 ROM Configure OTP, Secure IFR0 Customer Usage, Secure IFR0 CMAC Table and Secure IFR0 Over-The-Air update.
The result of the dump command will be stored in the FlashRunner 2.0 internal memory.

#TPCMD GET_DEVICE_INFORMATIONS

Syntax:	`#TPCMD GET_DEVICE_INFORMATIONS`
Prerequisites:	none
Description:	This function gets the device information
Note:	This command prints into Real Time Log
Examples:	Correct command execution: 😊



```
---#TPCMD GET_DEVICE_INFORMATIONS
Device configuration:
 * System Device Identification Register: 0x14251107.
   * Family ID: 0x1 - KE1x Family (Enhanced features).
   * Sub-family ID: 0x4.
   * Series ID: 0x2 - Kinetis E+ series.
   * Ram Size: 0x5 - 16KB.
   * Revision number: 0x1.
   * Project ID: 0x02.
   * Pin ID: 0x07 - 100-pin count.
 * Flash Configuration Register: 0x3703F000.
   * Flash memory size: 128 KB. Protection region size: 4 KB.
   * FlexNVM size: 32KB.
```

```
    * EEE SRAM data size: 2 KB.
    * Depart set to 0xF - 64 KBytes Data Flash.
    * Flash access is enabled.
    * FlexRAM is available as normal RAM.
Time for Get Device Information: 0.003 s.
```

#TPCMD GET_MEMORY_ID

Syntax: #TPCMD GET_MEMORY_ID

Prerequisites: none

Description: This function gets the device ID from the external memory connected to the Kinetis device

Note: This command is available from **libkinetis.so** version **5.11**
This command prints into Real Time Log and Terminal

Examples: Correct command execution: 😊

Real Time Log:

```
---#TPCMD GET_MEMORY_ID
Enabled LPSPI clock, core running at 72 MHz.
> Clock to LPSPI module is 72 MHz.
Initialize QSPI GPIO pins.
> QSPI GPIO pins are initialized.
Enabling Kinetis LPSPI peripheral.
 * QSPI frequency set to 25MHz.
 * Chip Select n.0 selected.
> Kinetis QSPI peripheral enabled.
Read external memory ID.
 * QSPI: Memory ID: 0x17609D.
Time for Get Memory ID: 0.021 s.
```

Terminal:

```
Memory ID: 0x17609D
```

#TPCMD RUN

Syntax: #TPCMD RUN <Time [s]>

<Time [s]>	Time in seconds (i.e., 2 s). This time is an optional parameter.

Prerequisites: none

Description: Move the Reset line up and down quickly if no parameter <Time [s]> is inserted.
#TPCMD RUN <Time [s]> instead moves the Reset line down and high, waits for the entered time.
This command typically can be used to execute the firmware programmed in the device.

#TPCMD READ_MEM8

Syntax: #TPCMD READ_MEM8 <Address> <Byte Count>

<Address>	Address in HEX format (i.e., 0x52002020)
<Byte Count>	Byte count in decimal format (i.e., 8 -> eight bytes)

Prerequisites: none

Description: Read memory byte per byte from target KINETIS device

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM8 0x52002020 8
Read[0x52002020]: 0xF0
Read[0x52002021]: 0xAA
Read[0x52002022]: 0x16
Read[0x52002023]: 0x14
Read[0x52002024]: 0x00
Read[0x52002025]: 0x00
Read[0x52002026]: 0x00
Read[0x52002027]: 0x00
Time for Read Mem: 0.002 s
```

#TPCMD READ_MEM16

Syntax: `#TPCMD READ_MEM16` <Address> <16-bit Word Count>

<Address>	Address in HEX format (i.e., 0x52002020)
<16-bit Word Count>	16-bit Word count in decimal format (i.e., 4 -> four 16-bit words)

Prerequisites: none

Description: Read memory 16-bit word per 16-bit word from target KINETIS device

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM16 0x52002020 4
Read[0x52002020]: 0xAAF0
Read[0x52002022]: 0x1416
Read[0x52002024]: 0x0000
Read[0x52002026]: 0x0000
Time for Read Mem: 0.002 s
```

#TPCMD READ_MEM32

Syntax: `#TPCMD READ_MEM32` <Address> <32-bit Word Count>

<Address>	Address in HEX format (i.e., 0x52002020)
<32-bit Word Count>	32-bit Word count in decimal format (i.e., 2 -> two 32-bit words)

Prerequisites: none

Description: Read memory 32-bit word per 32-bit word from target KINETIS device

Note: This command prints into Terminal and Real Time Log

Examples: Correct command execution: 😊

```
---#TPCMD READ_MEM32 0x52002020 2
Read[0x52002020]: 0x1416AAF0
Read[0x52002024]: 0x00000000
Time for Read Mem: 0.002 s
```

#TPCMD DISCONNECT

`#TPCMD DISCONNECT`

Disconnect function. Power off and exit.

NXP KINETIS Driver Examples

Here you can see a complete example of NXP KINETIS projects.

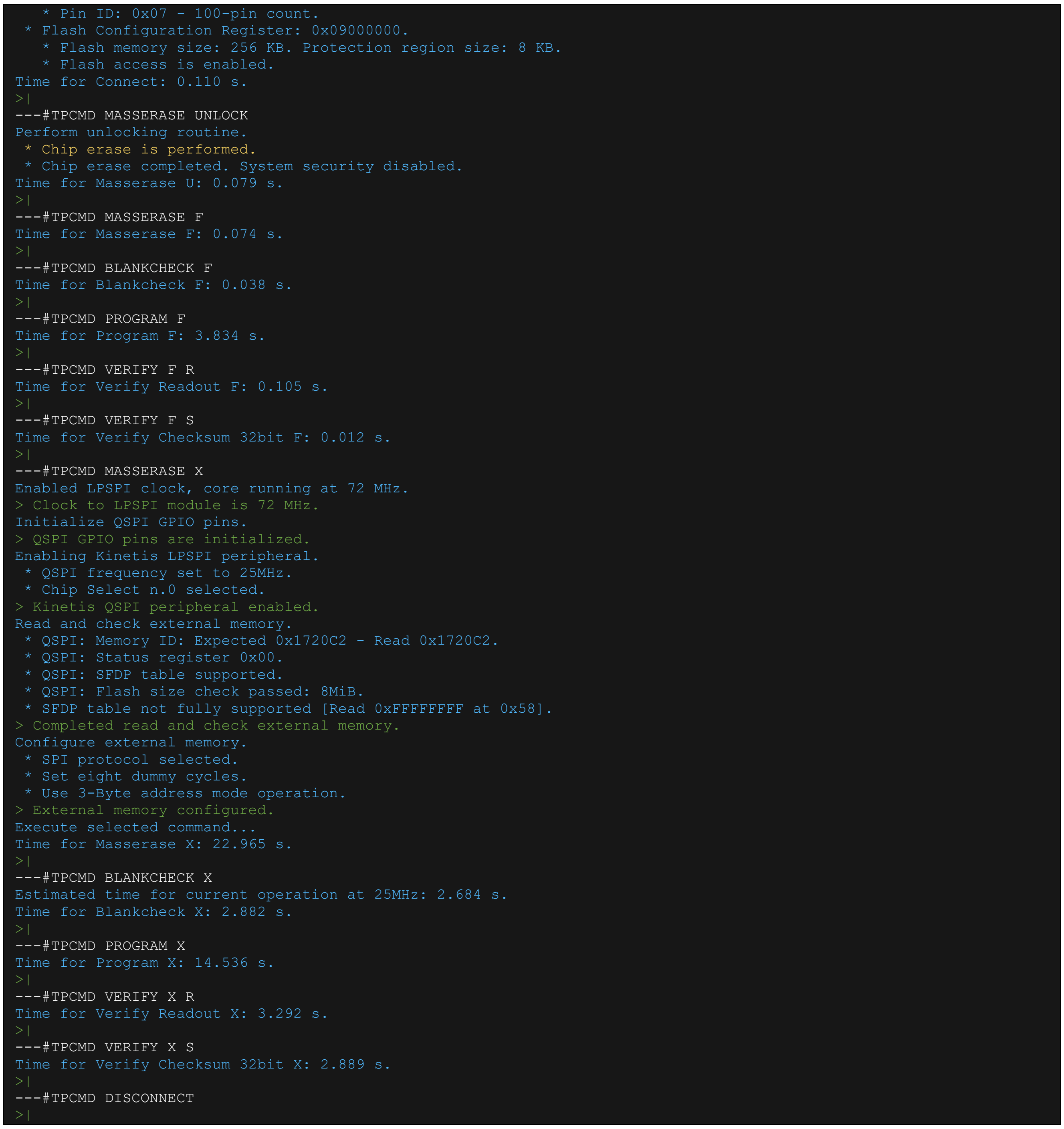
1 – NXP KINETIS MKE17Z256xxx7 256KB + 8MB example Commands

MKE17Z256xxx7 Kinetis device with **256 KB Flash** and an external memory **MX25L6433F** with **8 MiB Flash**.

```
#TCSETPAR ENTRY_CLOCK 1000000
#TCSETPAR PROTCLK 37500000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR QSPI_FREQUENCY 25MHz
#TCSETPAR QSPI_PROTOCOL SPI
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV PUSHPULL
#TCSETPAR RSTUP 100
#TCSETPAR VPROG0 5000
#TCSETPAR CMODE SWD
#TPSETSRC 256KB_8MB.frb
#TPSTART
#TPCMD CONNECT
#TPCMD MASSERASE UNLOCK
#TPCMD MASSERASE F
#TPCMD BLANKCHECK F
#TPCMD PROGRAM F
#TPCMD VERIFY F R
#TPCMD VERIFY F S
#TPCMD MASSERASE X
#TPCMD BLANKCHECK X
#TPCMD PROGRAM X
#TPCMD VERIFY X R
#TPCMD VERIFY X S
#TPCMD DISCONNECT
#TPEND
```

1 – NXP KINETIS MKE17Z256xxx7 256KB + 8MB example Real Time Log

```
---#TPSTART
Load SWD FPGA version 0x00001215.
>|
---#TPCMD CONNECT
Protocol selected SWD.
Entry Clock is 1.00 MHz.
Trying Hot Plug connect procedure.
IDCODE: 0x0BC11477.
Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
ID-Code read correctly at 1.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[0] IDR: 0x04770031, Type: AMBA AHB3 bus.
AP[1] IDR: 0x001C0020, Type: JTAG connection.
AP[0] ROM table base address 0xF0002000.
CPUID: 0x410CC601.
Implementer Code: 0x41 - [ARM].
Found Cortex M0+ revision r0p1.
Cortex M0+ Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 4-6].
IDCODE: 0x0BC11477.
Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
ID-Code read correctly at 37.50 MHz.
Device configuration:
 * System Device Identification Register: 0x17270207.
   * Family ID: 0x1 - KE1x Family (Enhanced features).
   * Sub-family ID: 0x7.
   * Series ID: 0x2 - Kinetis E+ series.
   * Ram Size: 0x7 - 48KB.
   * Revision number: 0x0.
   * Project ID: 0x04.
```



```
    * Pin ID: 0x07 - 100-pin count.
  * Flash Configuration Register: 0x09000000.
    * Flash memory size: 256 KB. Protection region size: 8 KB.
    * Flash access is enabled.
Time for Connect: 0.110 s.
>|
---#TPCMD MASSERASE UNLOCK
Perform unlocking routine.
  * Chip erase is performed.
  * Chip erase completed. System security disabled.
Time for Masserase U: 0.079 s.
>|
---#TPCMD MASSERASE F
Time for Masserase F: 0.074 s.
>|
---#TPCMD BLANKCHECK F
Time for Blankcheck F: 0.038 s.
>|
---#TPCMD PROGRAM F
Time for Program F: 3.834 s.
>|
---#TPCMD VERIFY F R
Time for Verify Readout F: 0.105 s.
>|
---#TPCMD VERIFY F S
Time for Verify Checksum 32bit F: 0.012 s.
>|
---#TPCMD MASSERASE X
Enabled LPSPI clock, core running at 72 MHz.
> Clock to LPSPI module is 72 MHz.
Initialize QSPI GPIO pins.
> QSPI GPIO pins are initialized.
Enabling Kinetis LPSPI peripheral.
  * QSPI frequency set to 25MHz.
  * Chip Select n.0 selected.
> Kinetis QSPI peripheral enabled.
Read and check external memory.
  * QSPI: Memory ID: Expected 0x1720C2 - Read 0x1720C2.
  * QSPI: Status register 0x00.
  * QSPI: SFDP table supported.
  * QSPI: Flash size check passed: 8MiB.
  * SFDP table not fully supported [Read 0xFFFFFFFF at 0x58].
> Completed read and check external memory.
Configure external memory.
  * SPI protocol selected.
  * Set eight dummy cycles.
  * Use 3-Byte address mode operation.
> External memory configured.
Execute selected command...
Time for Masserase X: 22.965 s.
>|
---#TPCMD BLANKCHECK X
Estimated time for current operation at 25MHz: 2.684 s.
Time for Blankcheck X: 2.882 s.
>|
---#TPCMD PROGRAM X
Time for Program X: 14.536 s.
>|
---#TPCMD VERIFY X R
Time for Verify Readout X: 3.292 s.
>|
---#TPCMD VERIFY X S
Time for Verify Checksum 32bit X: 2.889 s.
>|
---#TPCMD DISCONNECT
>|
```

1 – NXP KINETIS MKE17Z256xxx7 256KB + 8MB example Programming Times

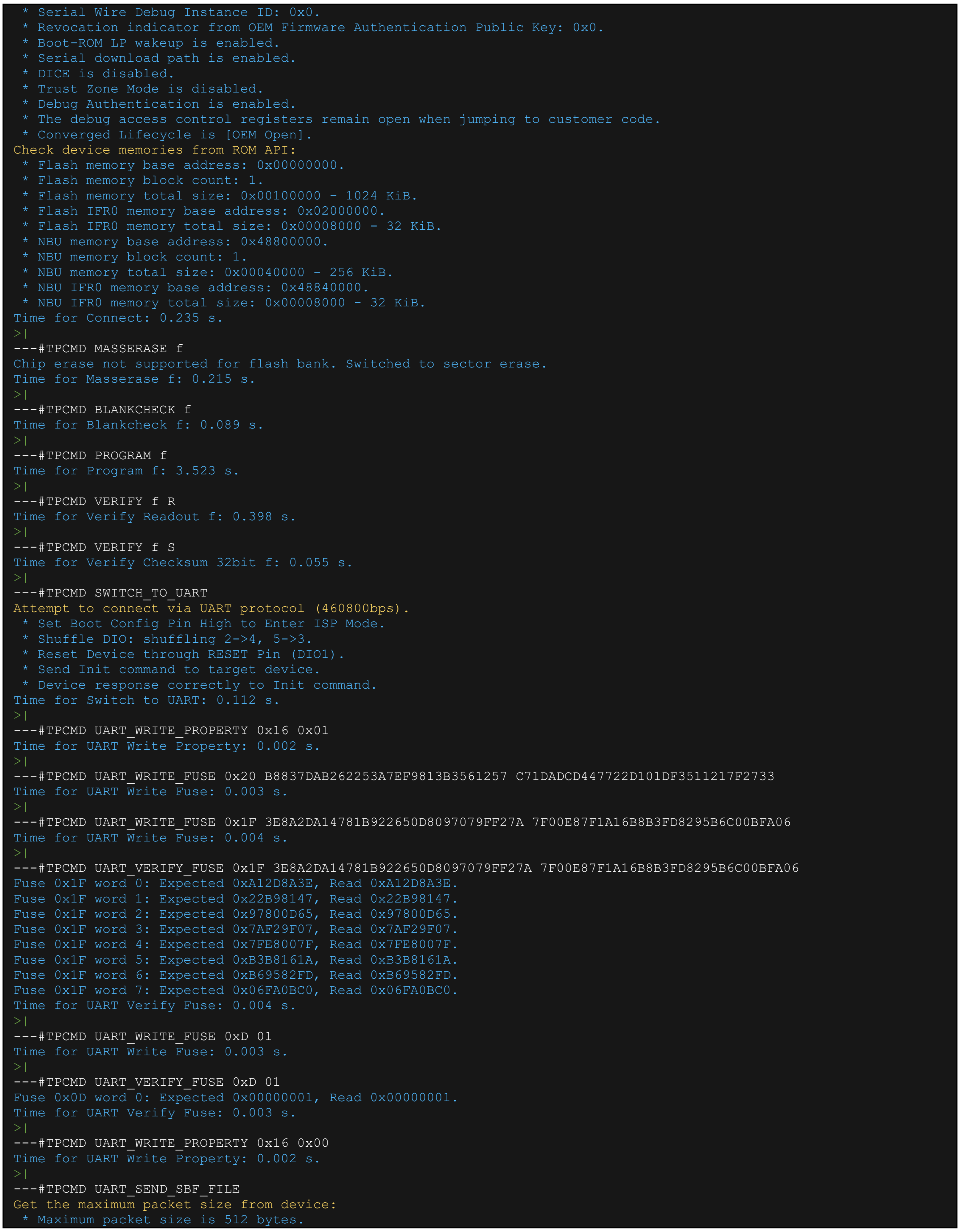
Operation	Timings FlashRunner 2.0
Time for Connect	0.110 s
Masserase Unlock	0.079 s
Masserase Flash	0.074 s
Blankcheck Flash	0.038 s
Program Flash	3.834 s
Verify Readout Flash	0.105 s
Verify Checksum Flash	0.012 s
Masserase External Flash	22.965 s
Blankcheck External Flash	2.882 s
Program External Flash	14.536 s
Verify Readout External Flash	3.292 s
Verify Checksum External Flash	2.889 s
Cycle Time	**00:50.870 s**

2 – NXP KINETIS KW45 1MB and SBF example Commands

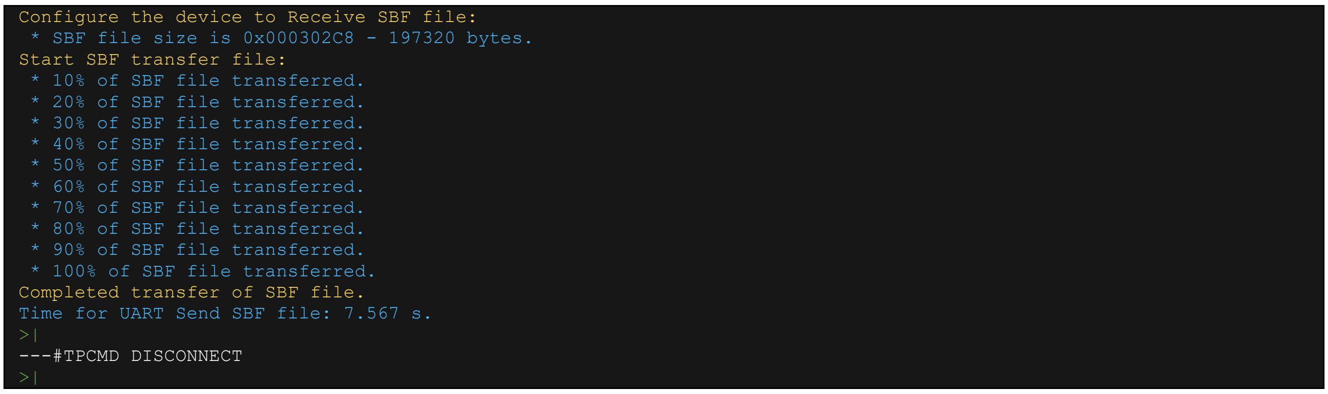
```
#TCSETPAR PROTCLK 37500000
#TCSETPAR PWDOWN 100
#TCSETPAR PWUP 100
#TCSETPAR RSTDOWN 100
#TCSETPAR RSTDRV OPENDRAIN
#TCSETPAR RSTUP 100
#TCSETPAR UART_FREQUENCY 460800
#TCSETPAR VPROG0 3300
#TCSETPAR CMODE SWD
#TPSETSRC Flash_1MB_plus_SBF.frb
#TPSTART
#TPCMD CONNECT
#TPCMD MASSERASE f
#TPCMD BLANKCHECK f
#TPCMD PROGRAM f
#TPCMD VERIFY f R
#TPCMD VERIFY f S
#TPCMD SWITCH_TO_UART
#TPCMD UART_WRITE_PROPERTY 0x16 0x01
#TPCMD UART_WRITE_FUSE 0x20 B8837DAB262253A7EF9813B3561257 C71DADCD447722D101DF3511217F2733

;Fuse 0x20 cannot be read/verified.
;Fuse 0x20 this slot is for a symmetric key used in encryption of the sb3 container.
;Therefore being of symmetric nature we do not want it to be visible to the user and keep it safe.
;#TPCMD UART_VERIFY_FUSE 0x20 B8837DAB262253A7EF9813B3561257 C71DADCD447722D101DF3511217F2733

#TPCMD UART_WRITE_FUSE 0x1F 3E8A2DA14781B922650D8097079FF27A 7F00E87F1A16B8B3FD8295B6C00BFA06
#TPCMD UART_VERIFY_FUSE 0x1F 3E8A2DA14781B922650D8097079FF27A 7F00E87F1A16B8B3FD8295B6C00BFA06
#TPCMD UART_WRITE_FUSE 0xD 01
#TPCMD UART_VERIFY_FUSE 0xD 01
#TPCMD UART_WRITE_PROPERTY 0x16 0x00
#TPCMD UART_SEND_SBF_FILE
#TPCMD DISCONNECT
#TPEND
```

2 – NXP KINETIS KW45 1MB and SBF example Real Time Log

```
---#TPSTART
Load SWD FPGA version 0x00001215.
>|
---#TPCMD CONNECT
Protocol selected SWD.
Entry Clock is 1.00 MHz.
Trying forcing hardware reset high procedure.
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 1.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[00] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[02] IDR: 0x002A0000, Type: JTAG connection.
AP[29] IDR: 0x54770002, Type: AMBA APB2 or APB3 bus.
AP[30] IDR: 0x24770011, Type: AMBA AHB3 bus.
AP[31] IDR: 0x84770001, Type: AMBA AHB3 bus.
AP[0] ROM table base address 0xE00FF000.
Used Debugger Mailbox to unlock AP buses.
 * Unlock Core M33 AP0 bus through NXP Debug Authentication Protocol.
 * Core M33 AP0 bus unlocked.
CPUID: 0x410FD214.
Implementer Code: 0x41 - [ARM].
Found Cortex M33 revision r0p4.
Cortex M33 Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 4 [Range 4-7].
IDCODE: 0x6BA02477.
Designer: 0x23B, Part Number: 0xBA02, Version: 0x6.
ID-Code read correctly at 37.50 MHz.
Device Unique ID registers:
 * UID0: 0x4B4B3839, UID1: 0x00005030, UID2: 0x00000001, UID3: 0x00150049.
Device lifecycle configuration: [0x00000407].
```



```
 * Serial Wire Debug Instance ID: 0x0.
 * Revocation indicator from OEM Firmware Authentication Public Key: 0x0.
 * Boot-ROM LP wakeup is enabled.
 * Serial download path is enabled.
 * DICE is disabled.
 * Trust Zone Mode is disabled.
 * Debug Authentication is enabled.
 * The debug access control registers remain open when jumping to customer code.
 * Converged Lifecycle is [OEM Open].
Check device memories from ROM API:
 * Flash memory base address: 0x00000000.
 * Flash memory block count: 1.
 * Flash memory total size: 0x00100000 - 1024 KiB.
 * Flash IFR0 memory base address: 0x02000000.
 * Flash IFR0 memory total size: 0x00008000 - 32 KiB.
 * NBU memory base address: 0x48800000.
 * NBU memory block count: 1.
 * NBU memory total size: 0x00040000 - 256 KiB.
 * NBU IFR0 memory base address: 0x48840000.
 * NBU IFR0 memory total size: 0x00008000 - 32 KiB.
Time for Connect: 0.235 s.
>|
---#TPCMD MASSERASE f
Chip erase not supported for flash bank. Switched to sector erase.
Time for Masserase f: 0.215 s.
>|
---#TPCMD BLANKCHECK f
Time for Blankcheck f: 0.089 s.
>|
---#TPCMD PROGRAM f
Time for Program f: 3.523 s.
>|
---#TPCMD VERIFY f R
Time for Verify Readout f: 0.398 s.
>|
---#TPCMD VERIFY f S
Time for Verify Checksum 32bit f: 0.055 s.
>|
---#TPCMD SWITCH_TO_UART
Attempt to connect via UART protocol (460800bps).
 * Set Boot Config Pin High to Enter ISP Mode.
 * Shuffle DIO: shuffling 2->4, 5->3.
 * Reset Device through RESET Pin (DIO1).
 * Send Init command to target device.
 * Device response correctly to Init command.
Time for Switch to UART: 0.112 s.
>|
---#TPCMD UART_WRITE_PROPERTY 0x16 0x01
Time for UART Write Property: 0.002 s.
>|
---#TPCMD UART_WRITE_FUSE 0x20 B8837DAB262253A7EF9813B3561257 C71DADCD447722D101DF3511217F2733
Time for UART Write Fuse: 0.003 s.
>|
---#TPCMD UART_WRITE_FUSE 0x1F 3E8A2DA14781B922650D8097079FF27A 7F00E87F1A16B8B3FD8295B6C00BFA06
Time for UART Write Fuse: 0.004 s.
>|
---#TPCMD UART_VERIFY_FUSE 0x1F 3E8A2DA14781B922650D8097079FF27A 7F00E87F1A16B8B3FD8295B6C00BFA06
Fuse 0x1F word 0: Expected 0xA12D8A3E, Read 0xA12D8A3E.
Fuse 0x1F word 1: Expected 0x22B98147, Read 0x22B98147.
Fuse 0x1F word 2: Expected 0x97800D65, Read 0x97800D65.
Fuse 0x1F word 3: Expected 0x7AF29F07, Read 0x7AF29F07.
Fuse 0x1F word 4: Expected 0x7FE8007F, Read 0x7FE8007F.
Fuse 0x1F word 5: Expected 0xB3B8161A, Read 0xB3B8161A.
Fuse 0x1F word 6: Expected 0xB69582FD, Read 0xB69582FD.
Fuse 0x1F word 7: Expected 0x06FA0BC0, Read 0x06FA0BC0.
Time for UART Verify Fuse: 0.004 s.
>|
---#TPCMD UART_WRITE_FUSE 0xD 01
Time for UART Write Fuse: 0.003 s.
>|
---#TPCMD UART_VERIFY_FUSE 0xD 01
Fuse 0x0D word 0: Expected 0x00000001, Read 0x00000001.
Time for UART Verify Fuse: 0.003 s.
>|
---#TPCMD UART_WRITE_PROPERTY 0x16 0x00
Time for UART Write Property: 0.002 s.
>|
---#TPCMD UART_SEND_SBF_FILE
Get the maximum packet size from device:
 * Maximum packet size is 512 bytes.
```



```
Configure the device to Receive SBF file:
 * SBF file size is 0x000302C8 - 197320 bytes.
Start SBF transfer file:
 * 10% of SBF file transferred.
 * 20% of SBF file transferred.
 * 30% of SBF file transferred.
 * 40% of SBF file transferred.
 * 50% of SBF file transferred.
 * 60% of SBF file transferred.
 * 70% of SBF file transferred.
 * 80% of SBF file transferred.
 * 90% of SBF file transferred.
 * 100% of SBF file transferred.
Completed transfer of SBF file.
Time for UART Send SBF file: 7.567 s.
>|
---#TPCMD DISCONNECT
>|
```

2 – NXP KINETIS KW45 1MB and SBF example Programming Times

Operation	Timings FlashRunner 2.0
Time for Connect	0.235 s
Masserase Flash	0.215 s
Blankcheck Flash	0.089 s
Program Flash	3.523 s
Verify Readout Flash	0.398 s
Verify Checksum Flash	0.055 s
Switch to UART protocol	0.112 s
UART Write Property	0.002 s
UART Write eFUSE	0.003 s
UART Write eFUSE	0.004 s
UART Verify eFUSE	0.004 s
UART Write eFUSE	0.003 s
UART Verify eFUSE	0.003 s
UART Send SBF file	7.567 s
Cycle Time	**00:12.262 s**

NXP KINETIS Driver Changelog

Info about driver versions prior to 4.00
All driver versions prior to 4.00 are to be considered obsolete, please update your driver to the latest version.

Info about driver version 5.00 - 03/02/2023
Upgraded Kinetis driver.

Info about driver version 5.01 - 21/02/2023
Fixed Conditional Masserase for External Memory through Kinetis device.

Info about driver version 5.02 - 19/04/2023
Supported KW45xx devices.

Info about driver version 5.03 - 05/06/2023
Fixed issue for S9KEAZNxx devices with small RAM.

Info about driver version 5.04 - 14/06/2023
Supported MKMx series.
Supported MKL3x and other Kinetis families.

Info about driver version 5.05 - 19/06/2023
Supported MKV1xx series.

Info about driver version 5.06 - 23/06/2023
Fixed connect command for MKW3xx series.

Info about driver version 5.07 - 03/07/2023
Completed support of MK1xx subfamily of Kinetis MK1 series.

Info about driver version 5.08 - 08/09/2023
Completed support of MKE13Z128xxx7.

Info about driver version 5.09 – 09/11/2023
Updated unlock procedure for MKW series.

Info about driver version 5.10 – 30/01/2024
Supported new Kinetis families.

Info about driver version 5.11 – 27/02/2024
Supported new external memories through Kinetis MKEx devices.
Added `#TPCMD GET_DEVICE_ID` command.

Info about driver version 5.12 – 07/05/2024
Updated error management when the user tries to send a UART command and the KW45xx device is locked.
Added `#TCSETPAR UART_CONNECT` parameter for KW45 devices to perform UART connect instead of SWD.

Info about driver version 5.13 – 13/12/2024
Updated Kinetis help driver command.
Updated KW45 cache management.
Added KW45 lifecycle management.

Info about driver version 5.14 – 14/03/2025
Updated Kinetis help driver command.
Added command