HOPERF AN260

CMT2186A / CMT2187A Develop Environment Construction and Debugging

Overview

This document mainly introduces the develop environment construction and debugging methods of two single-shot SoC products of CMT2186A & CMT2187A (hereinafter referred to as CMT218xA). It is aimed to help users quickly handle the tool chain and debugging methods of this type of single-shot SoC product.

Applicable part numbers are shown in the table below.

Table 1. Product Models Covered in this Document

Part Number	PA	CMP	MTP	XRAM	RAM	EEPROM	WDT	TMR	Interface	GPIO	Package
CMT2186A-ESR14	Single-ended	x 2	4 kB	512 B	256 B	64 B	√	x 4	UART x 1 SPI x 1	9	SOP14
CMT2186A-ESR16		x 2								11	SOP16
CMT21867A-ESR		--								6	SOP14

Note: This document does not involve the specifications, performance and parameters of each model. If you need to know, please refer to the data sheet of each model; If you need to know the package size, silk printing and ordering information, please also refer to the data sheet of each model; For more information of the product specific functions, please refer to the using guide of each model.

Catalogue

1 Develop environment construction

1.1 Basic Introduction

The core of CMT218xA product uses enhanced 8051, and the core comes with a 1-Wire simulation debugging interface function. The Keil C51 integrated develop environment applied with the widely familiar IDE of the 51 system. And all the dynamic link driver is supported by the 1-Wire debugging interface developed for the Keil C51 platform, so this document takes the Keil C51 integrated development environment as the main development platform for introduction. In addition, it is recommended to use the Keil4 version.

1.2 Establish Project

- Step 1: Open the keil 4 development platform as shown in the figure below.

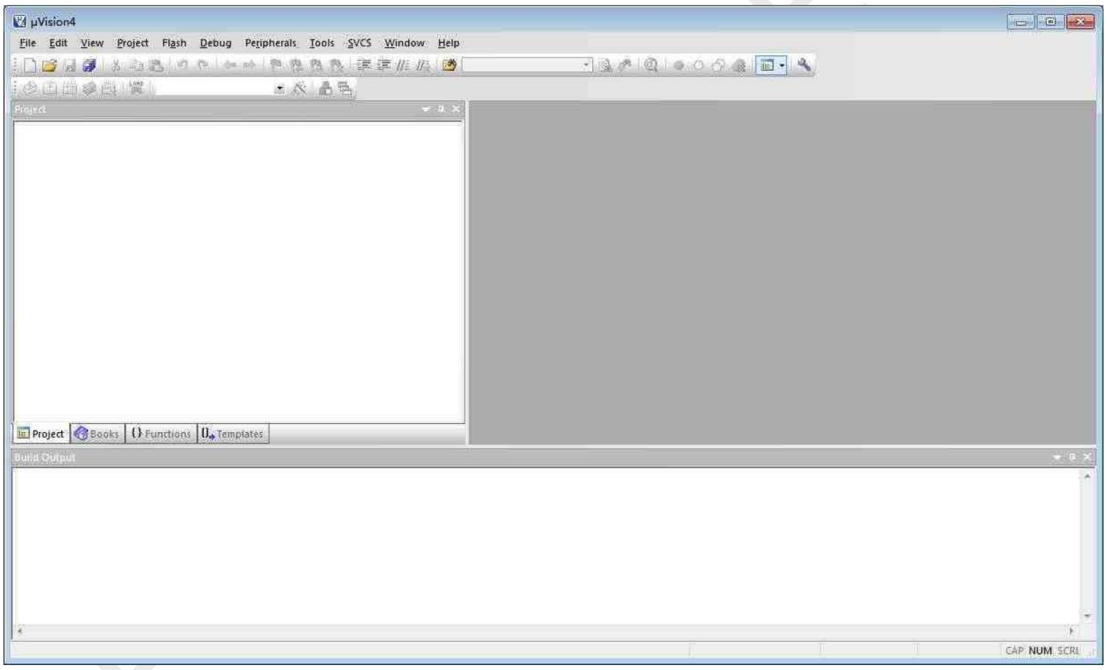

µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Project Books Functions Templates
Build Output
CAP NUM SCRL

Figure 1-1. Keil 4 Development Platform Interface

- Step 2: Select "Project" in the menu bar, and then select "New µVision Project …" in the Project drop-down menu, and the new Project window will pop up, as shown in Figures 1-2 and 1-3 below.

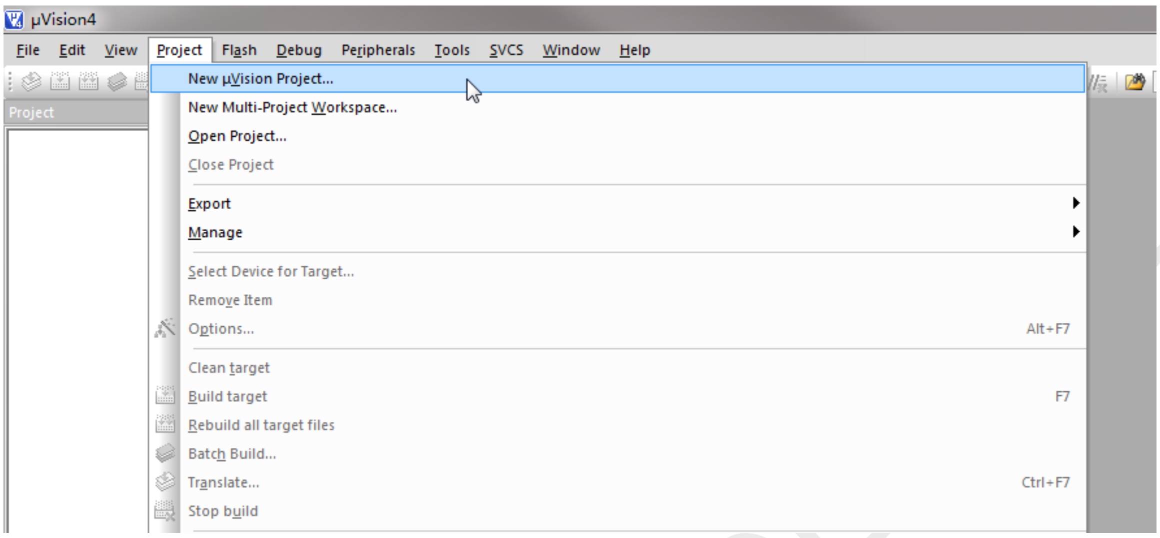


Figure 1-2. Project Drop-Down Menu

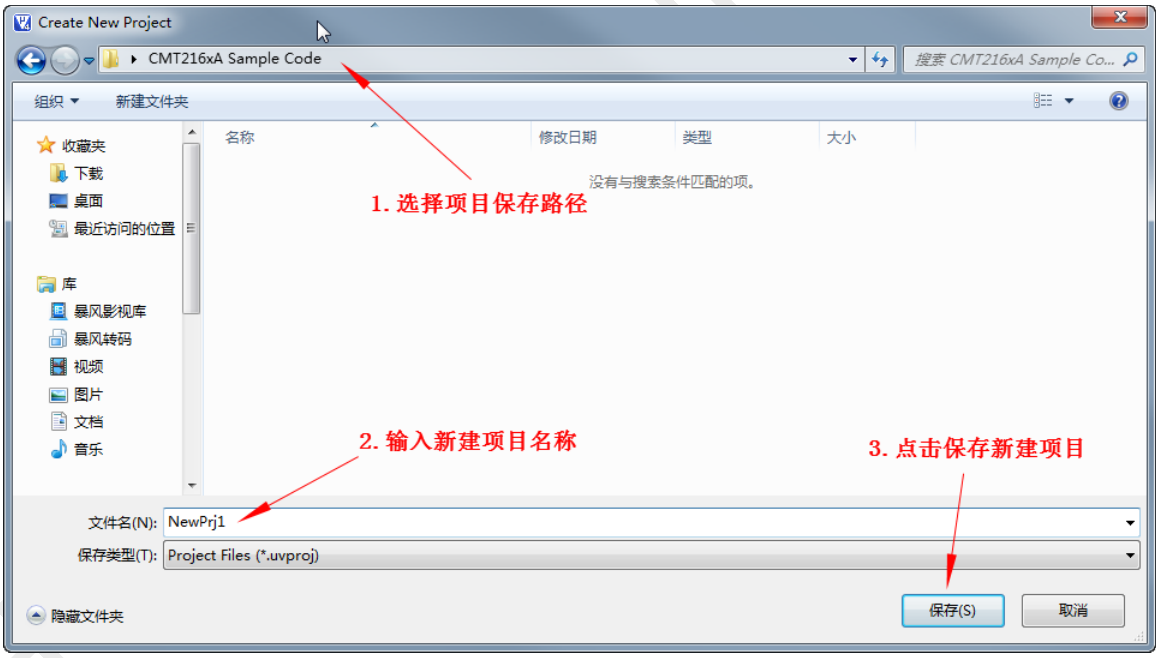
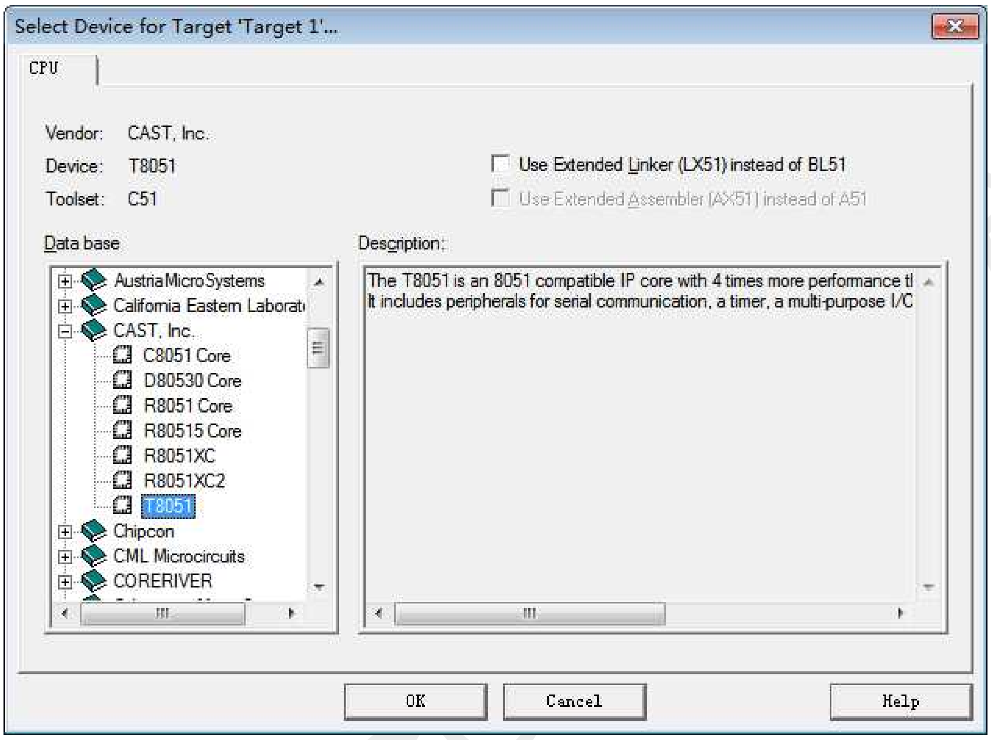


Figure 1-3. New Project Window

In the New Project window,

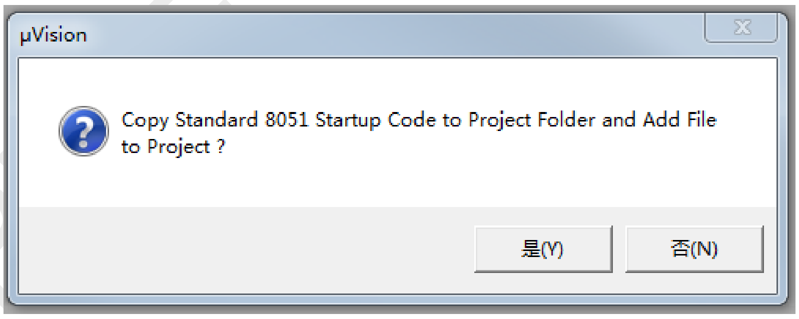
1. select the saving path for the new project;
2. Input the name of the new project;
3. Click Save;

- Step 3: Continue the previous step. After the project is successfully saved, the CPU device selection window will pop up. As shown in Figure 1-4 below, select T8051 core of CAST company and click the OK button. Then pop up interface of whether the STARTUP.A51 file is automatically generated, as shown in Figure 1-5.



The T8051 is an 8051 compatible IP core with 4 times more performance tl
It includes peripherals for serial communication, a timer, a multi-purpose I/C

Figure 1-4. Select CPU Model



Copy Standard 8051 Startup Code to Project Folder and Add File to Project ?

Figure 1-5. Generate Startup Code Window

Note: The STARTUP.A51 file can be generated automatically, and the user can also copy and import from the blank project or other routine of the CMT218xA.

- Step 4: Add *system.c* and *INTERRUPT _ PROTEST.a51* file to the project, as shown in Figure 1-6.

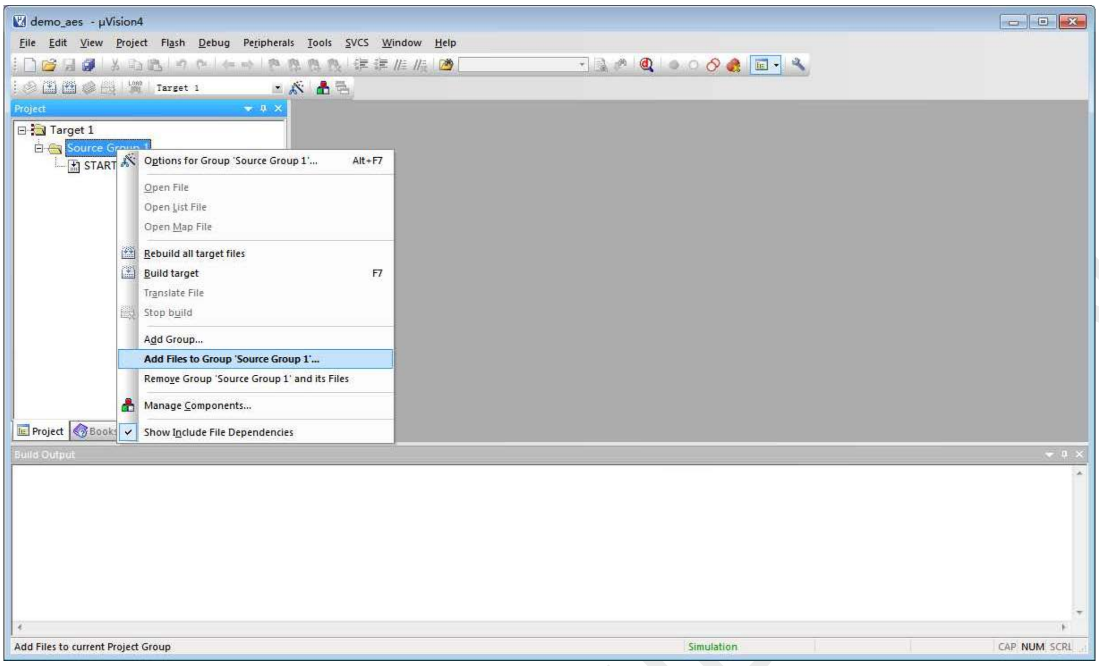
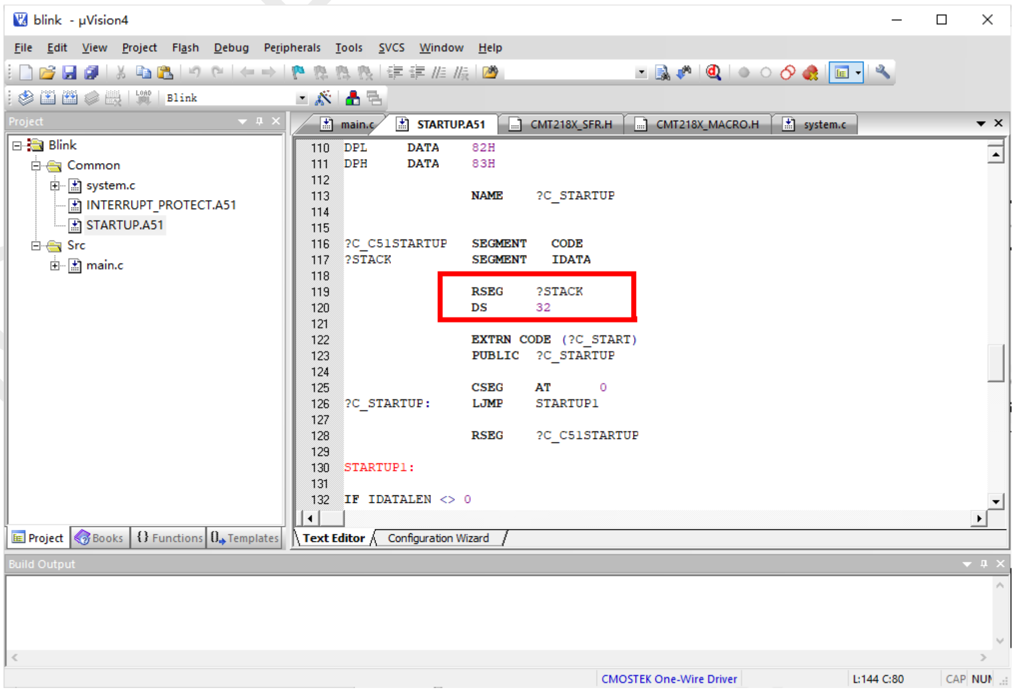

demo_aes - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
Source Group 1
STARTUP
Options for Group 'Source Group 1'... Alt+F7
Open File
Open List File
Open Map File
Rebuild all target files
Build target F7
Translate File
Stop build
Add Group...
Add Files to Group 'Source Group 1'...
Remove Group 'Source Group 1' and its Files
Manage Components...
Show Include File Dependencies
Project
Books
Build Output
Add Files to current Project Group
Simulation
CAP NUM SCRL

Figure 1-6. Add the Necessary Files

Note: system.c and INTERRUPT _ PROTEST.a51 must be added to any project. Users can copy it from a blank project or other routine of the CMT218xA.

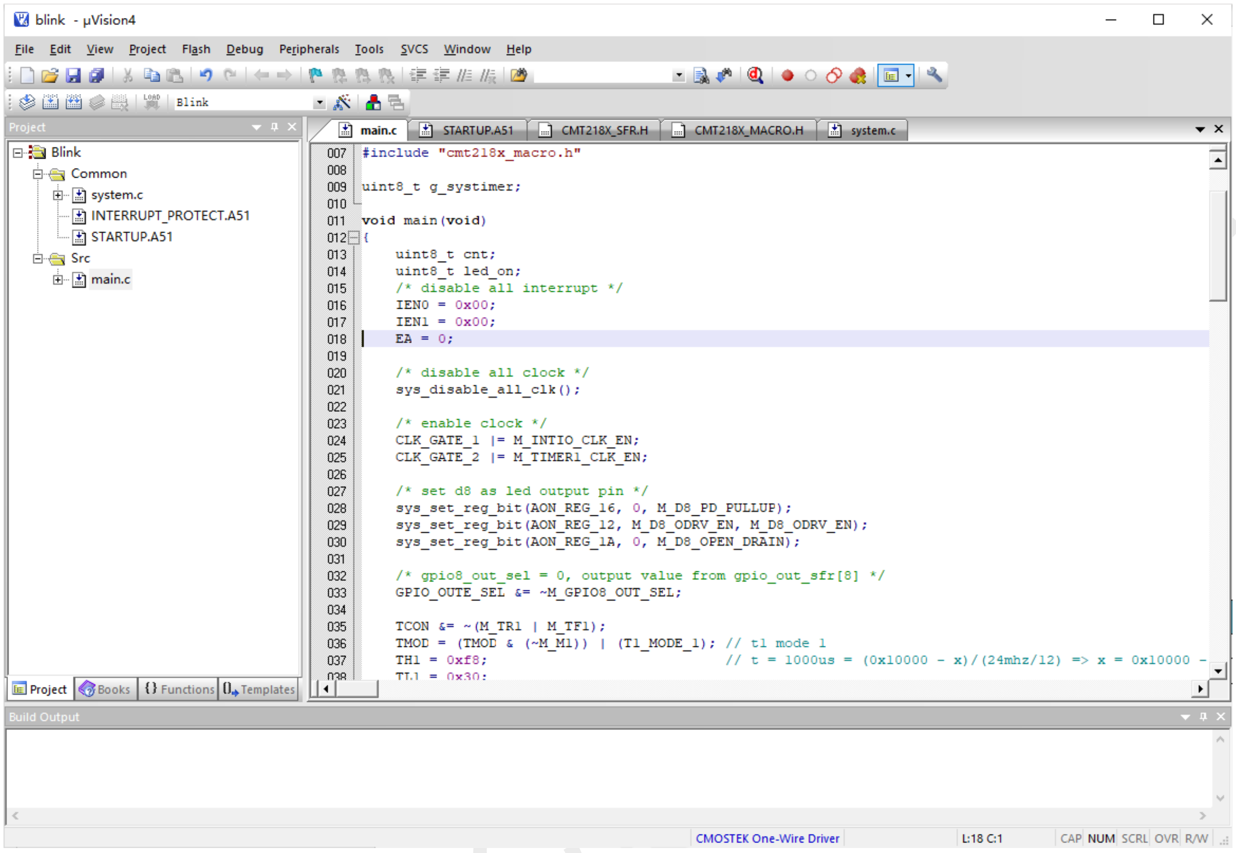
- Step 5: Open the STARTUP.A51 file and reserve the stack space size, as shown in Figure 1-7.



```
DPL     DATA    82H
DPH     DATA    83H

                NAME    ?C_STARTUP


?C_C51STARTUP   SEGMENT CODE
?STACK          SEGMENT IDATA

                RSEG    ?STACK
                DS      32

                EXTRN CODE  (?C_START)
                PUBLIC  ?C_STARTUP

                CSEG    AT      0
?C_STARTUP:     LJMP    STARTUP1

                RSEG    ?C_C51STARTUP

STARTUP1:

IF IDATALEN <> 0
```

Figure 1-7. Set the Reserved Stack Space Size

- Step 6: Setup a main program file (including main function), add it to the project, and then program it to implement related functions, as shown in Figure 1-8 (main.c is the main program file in the figure).


blink - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Blink
Project
Blink
Common
system.c
INTERRUPT_PROTECT.A51
STARTUP.A51
Src
main.c
Project Books Functions Templates
main.c STARTUP.A51 CMT218X_SFR.H CMT218X_MACRO.H system.c
Build Output
CMOSTEK One-Wire Driver
L:18 C:1
CAP NUM SCRL OVR R/W

```c
#include "cmt218x_macro.h"

uint8_t g_systimer;

void main(void)
{
    uint8_t cnt;
    uint8_t led_on;
    /* disable all interrupt */
    IEN0 = 0x00;
    IEN1 = 0x00;
    EA = 0;

    /* disable all clock */
    sys_disable_all_clk();

    /* enable clock */
    CLK_GATE_1 |= M_INTIO_CLK_EN;
    CLK_GATE_2 |= M_TIMER1_CLK_EN;

    /* set d8 as led output pin */
    sys_set_reg_bit(AON_REG_16, 0, M_D8_PD_PULLUP);
    sys_set_reg_bit(AON_REG_12, M_D8_ODRV_EN, M_D8_ODRV_EN);
    sys_set_reg_bit(AON_REG_1A, 0, M_D8_OPEN_DRAIN);

    /* gpio8_out_sel = 0, output value from gpio_out_sfr[8] */
    GPIO_OUTE_SEL &= ~M_GPIO8_OUT_SEL;

    TCON &= ~(M_TR1 | M_TF1);
    TMOD = (TMOD & (~M_M1)) | (T1_MODE_1); // t1 mode 1
    TH1 = 0xf8;                            // t = 1000us = (0x10000 - x)/(24mhz/12) => x = 0x10000 -
    TL1 = 0x30;
```

Figure 1-8. Add Main Function File

So far, a new project has been established. Users can also directly add the code files that need to be loaded and compiled through the blank project (i.e. project template) of CMT218xA, which can omit the steps of manually creating new projects mentioned above.

Note: The necessary files required for CMT218xA new project are shown in Figure 1-9 below. These files need to be copied to the project folder and included in the program for compilation. The meaning of these documents is as follows.

1. *Cmt218x. h, one of the header files, mainly contains the definition of related structures of chip peripherals, etc.;*
2. *Cmt218x _ types.h, one of the header files, mainly contains the typedef definition; Users can also add the required typedef description according to their own preferences;*
3. *Cmt218x _ SFR.h, one of the header files, mainly contains the SFR*

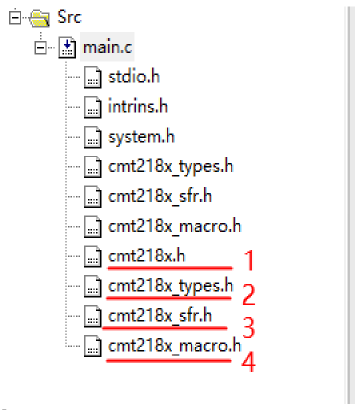


Figure 1-9. Necessary Documents for the Project

register definition;

4. *Cmt218x _ macro.h, one of the header files, mainly contains register macro definitions;*

2 1-WIRE online debugging

2.1 1-WIRE hardware construction

As shown in Figure 2-1, the hardware construction of CMT218xA simulation environment requires:

- CMT218xA debugging target board, as shown on the left side of the diagram, CMT2186A-EB or CMT2186A-DM is the debugging target ;
- CMT2186A/CMT2187A Simulator debugger, as shown in the middle of the diagram;
- Computer and USB cable (Type A to B)

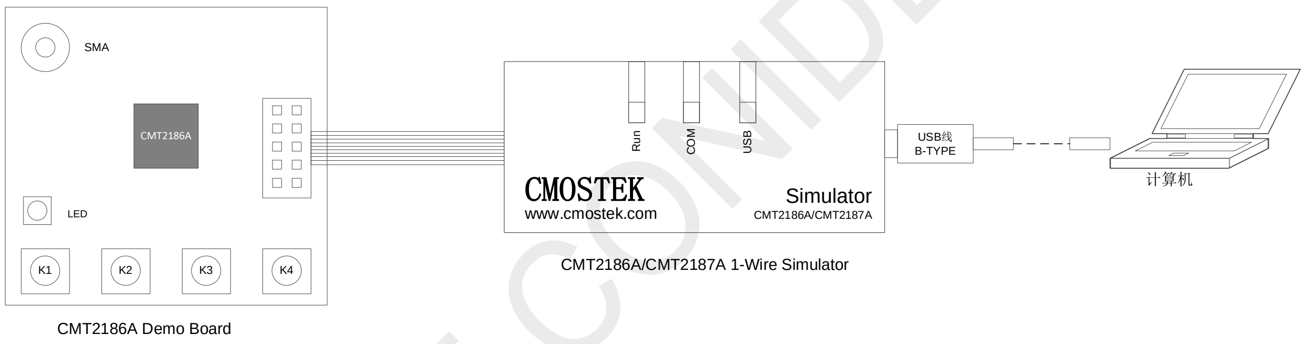


Figure 2-1. CMT218xA Debugging Hardware Connection Schematic

Although the Simulator uses a simple 10-pin arrangement, the actual CMT218xA series single-shot SoC products only need to be connected to 3 wires (VIN, GND, 1-Wire/D10) to be debugged (the RSTn/D0 reset pin can be disconnected). Therefore, when users need to make their own target debugging boards, only three connection points maintained to save PCB area. The 10-pin simple horn arrangement of the simulator is shown as followed.

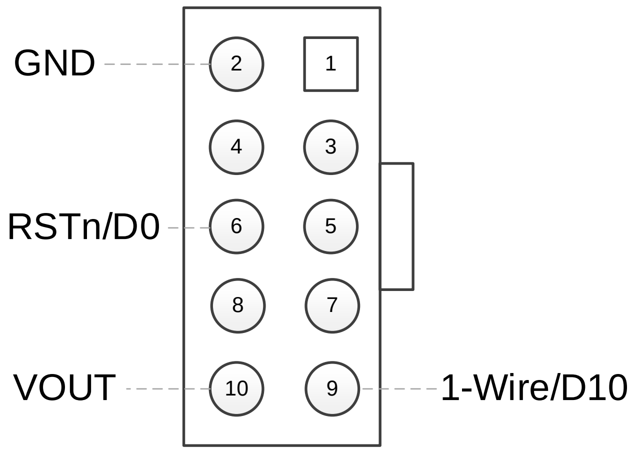


Figure 2-2. Simulator 10-pin Function Definition

Note: It is the front view of the Simulator 10-pin horn, facing the 10 holes.

2.2 1-WIRE debugging mode

The basic process of 1-WIRE debugging mode is shown in the figure below.

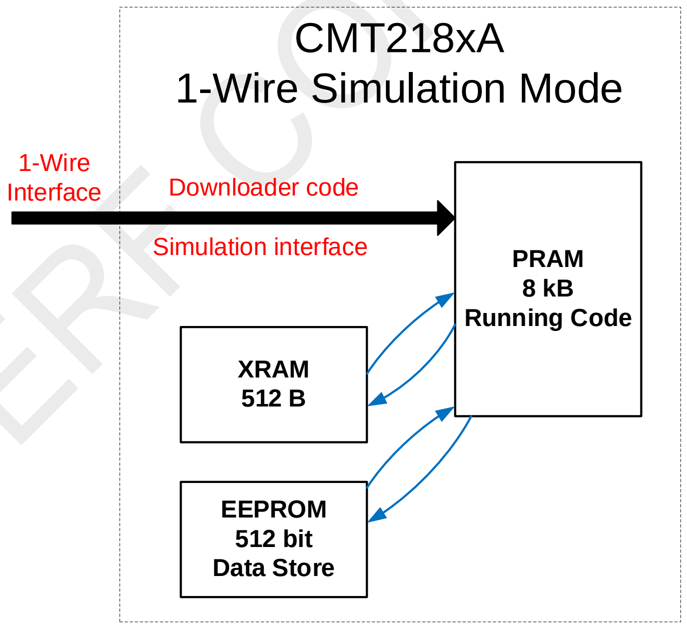


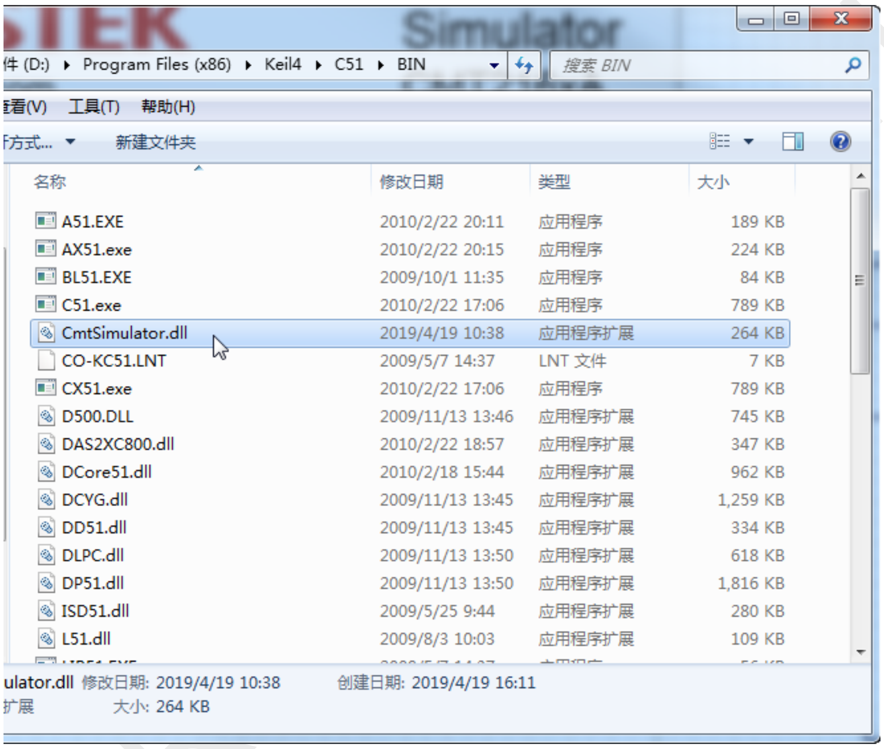
Figure 2-3. Schematic of 1- WIRE Debugging Process

- The debugging object code is directly downloaded to the CMT218xA on-chip MTP through the 1-WIRE interface;
- Under the monitoring of 1-WIRE interface, the code runs in MTP;

2.3 1-WIRE simulation settings

Before performing 1-WIRE simulation of CMT218xA, it is required to perfprmed the following Keil IDE software-related settings.

- Step 1: Copy the CmtSimulator.dll driver file to the Keil installation directory under the path: "…\ Keil\ C51\ BIN\", as shown in Figure 2-4. This DLL file is a library file for the Keil IDE environment driver Simulator debugger.



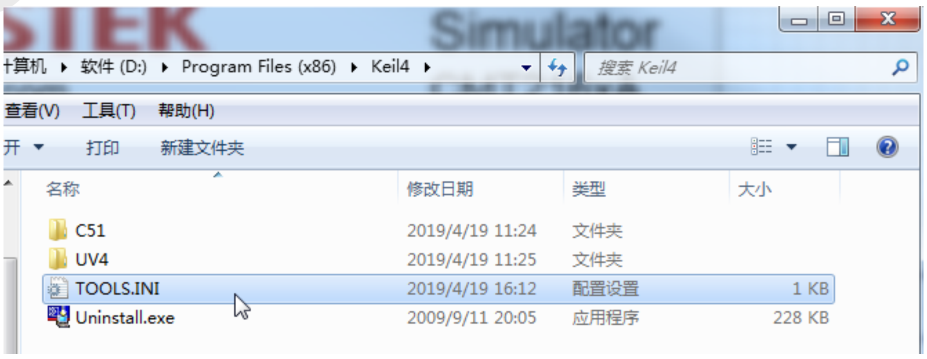
件 (D:) ▸ Program Files (x86) ▸ Keil4 ▸ C51 ▸ BIN

名称	修改日期	类型	大小
A51.EXE	2010/2/22 20:11	应用程序	189 KB
AX51.exe	2010/2/22 20:15	应用程序	224 KB
BL51.EXE	2009/10/1 11:35	应用程序	84 KB
C51.exe	2010/2/22 17:06	应用程序	789 KB
CmtSimulator.dll	2019/4/19 10:38	应用程序扩展	264 KB
CO-KC51.LNT	2009/5/7 14:37	LNT 文件	7 KB
CX51.exe	2010/2/22 17:06	应用程序	789 KB
D500.DLL	2009/11/13 13:46	应用程序扩展	745 KB
DAS2XC800.dll	2010/2/22 18:57	应用程序扩展	347 KB
DCore51.dll	2010/2/18 15:44	应用程序扩展	962 KB
DCYG.dll	2009/11/13 13:45	应用程序扩展	1,259 KB
DD51.dll	2009/11/13 13:45	应用程序扩展	334 KB
DLPC.dll	2009/11/13 13:50	应用程序扩展	618 KB
DP51.dll	2009/11/13 13:50	应用程序扩展	1,816 KB
ISD51.dll	2009/5/25 9:44	应用程序扩展	280 KB
L51.dll	2009/8/3 10:03	应用程序扩展	109 KB

ulator.dll 修改日期: 2019/4/19 10:38 创建日期: 2019/4/19 16:11
扩展 大小: 264 KB

Figure 2-4. Copy the CmtSimulator.dll Driver File

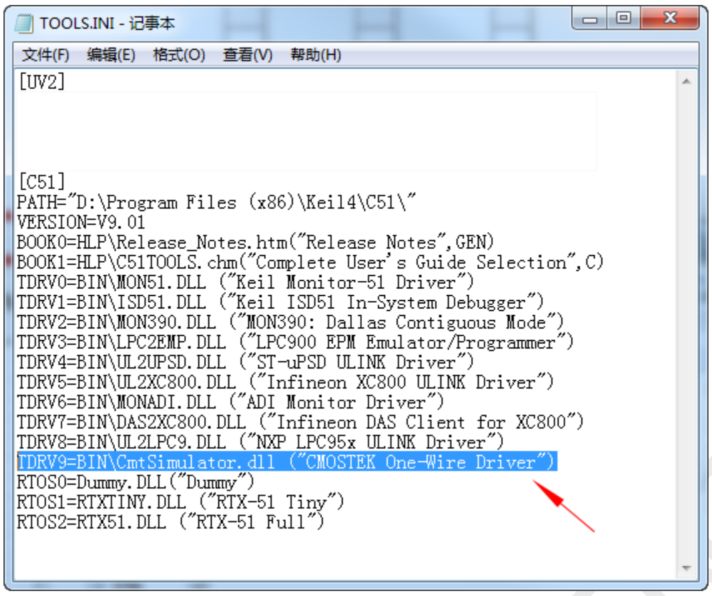
- Step 2: Modify the TOOLS.INI file in the Keil installation directory to add the CMT218xA Simulator debugger to Keil's debugger selection list. As shown in Figure 2-5 and 2-6.



十算机 ▸ 软件 (D:) ▸ Program Files (x86) ▸ Keil4 ▸

名称	修改日期	类型	大小
C51	2019/4/19 11:24	文件夹	
UV4	2019/4/19 11:25	文件夹	
TOOLS.INI	2019/4/19 16:12	配置设置	1 KB
Uninstall.exe	2009/9/11 20:05	应用程序	228 KB

Figure 2-5. TOOLS.INI File Path

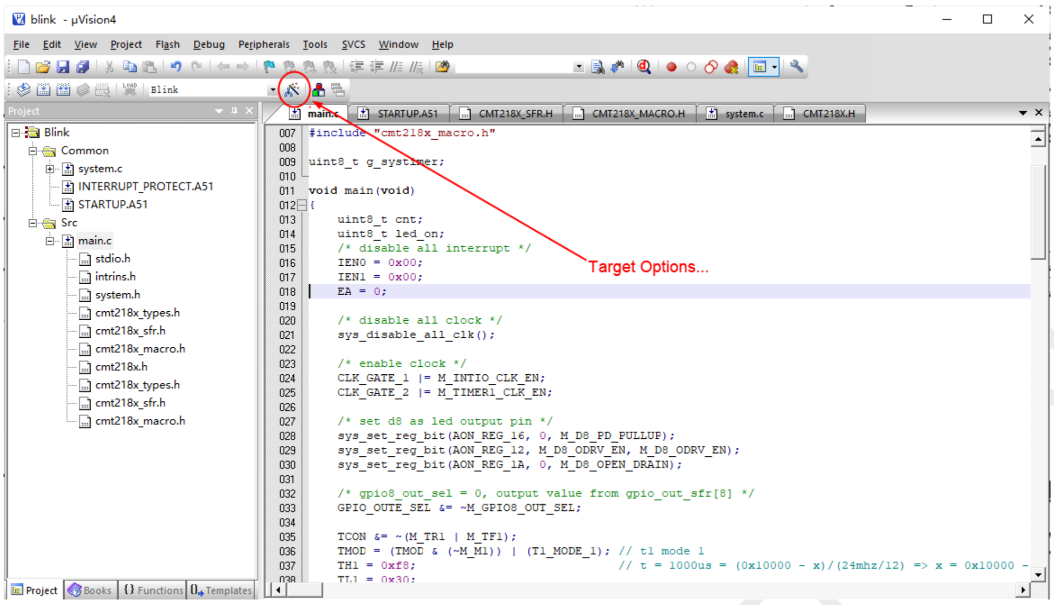


```
[UV2]


[C51]
PATH="D:\Program Files (x86)\Keil4\C51\"
VERSION=V9.01
BOOK0=HLP\Release_Notes.htm("Release Notes",GEN)
BOOK1=HLP\C51TOOLS.chm("Complete User's Guide Selection",C)
TDRV0=BIN\MON51.DLL ("Keil Monitor-51 Driver")
TDRV1=BIN\ISD51.DLL ("Keil ISD51 In-System Debugger")
TDRV2=BIN\MON390.DLL ("MON390: Dallas Contiguous Mode")
TDRV3=BIN\LPC2EMP.DLL ("LPC900 EPM Emulator/Programmer")
TDRV4=BIN\UL2UPSD.DLL ("ST-uPSD ULINK Driver")
TDRV5=BIN\UL2XC800.DLL ("Infineon XC800 ULINK Driver")
TDRV6=BIN\MONADI.DLL ("ADI Monitor Driver")
TDRV7=BIN\DAS2XC800.DLL ("Infineon DAS Client for XC800")
TDRV8=BIN\UL2LPC9.DLL ("NXP LPC95x ULINK Driver")
TDRV9=BIN\CmtSimulator.dll ("CMOSTEK One-Wire Driver")
RTOS0=Dummy.DLL("Dummy")
RTOS1=RTXTINY.DLL ("RTX-51 Tiny")
RTOS2=RTX51.DLL ("RTX-51 Full")
```

Figure 2-6. Add CmtSimulator.dll File Reference

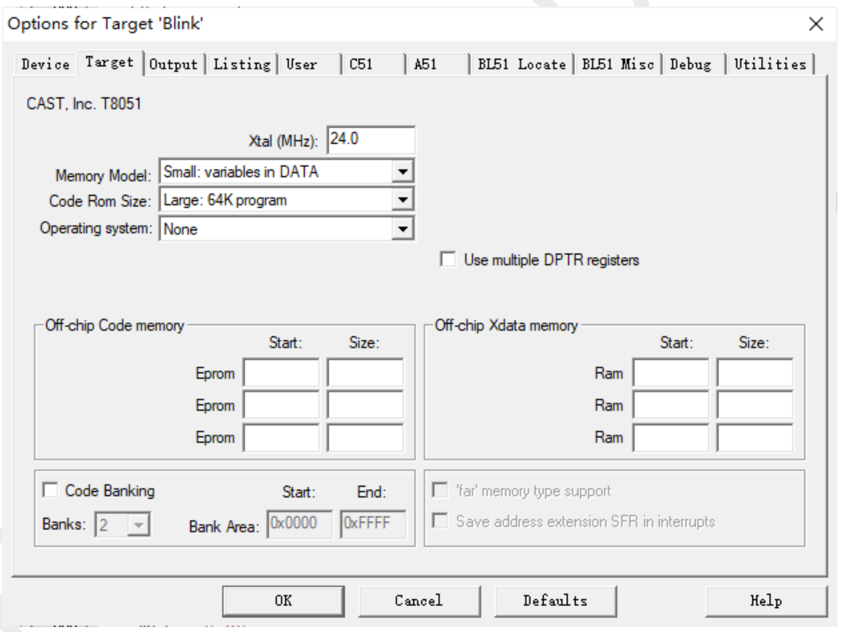
Note: The above picture is for reference only, and users need to modify it according to the current configuration file of Keil in the system while using. As shown in the above figure, before adding CmtSimulator.dll, the last item is TDRV8, so when adding CmtSimulator.dll, you need to add TDRV9. In addition, the right side of the TDRV9 equal sign is the path to CmtSimulator.dll. Finally, the content in the double quotation marks in the rightmost brackets is the tool description, which will be displayed simultaneously in the optional menu of Keil simulation debugger (the content is consistent), so users can also fill in specific content according to their needs.

- Step 3: Return to the Keil project engineering interface, select "Target Options..." (i.e. click the magic wand), and the "Options for Target 'Target 1'" interface will pop up, as shown in Figure 2-7 and Figure 2-8.



```
#include "cmt218x_macro.h"

uint8_t g_systimer;

void main(void)
{
    uint8_t cnt;
    uint8_t led_on;
    /* disable all interrupt */
    IEN0 = 0x00;
    IEN1 = 0x00;
    EA = 0;

    /* disable all clock */
    sys_disable_all_clk();

    /* enable clock */
    CLK_GATE_1 |= M_INTIO_CLK_EN;
    CLK_GATE_2 |= M_TIMER1_CLK_EN;

    /* set d8 as led output pin */
    sys_set_reg_bit(AON_REG_16, 0, M_D8_PD_PULLUP);
    sys_set_reg_bit(AON_REG_12, M_D8_ODRV_EN, M_D8_ODRV_EN);
    sys_set_reg_bit(AON_REG_1A, 0, M_D8_OPEN_DRAIN);

    /* gpio8_out_sel = 0, output value from gpio_out_sfr[8] */
    GPIO_OUTE_SEL &= ~M_GPIO8_OUT_SEL;

    TCON &= ~(M_TR1 | M_TF1);
    TMOD = (TMOD & (~M_M1)) | (T1_MODE_1); // t1 mode 1
    TH1 = 0xf8;                            // t = 1000us = (0x10000 - x)/(24mhz/12) => x = 0x10000 -
    TL1 = 0x30;
```

Figure 2-7. Select Target Options...


Options for Target 'Blink'
Device Target Output Listing User C51 A51 BL51 Locate BL51 Misc Debug Utilities
OK Cancel Defaults Help

CAST, Inc. T8051

Xtal (MHz): 24.0

Memory Model: Small: variables in DATA

Code Rom Size: Large: 64K program

Operating system: None

[ ] Use multiple DPTR registers

Off-chip Code memory

	Start:	Size:
Eprom		
Eprom		
Eprom		

Off-chip Xdata memory

	Start:	Size:
Ram		
Ram		
Ram		

[ ] Code Banking Start: End:

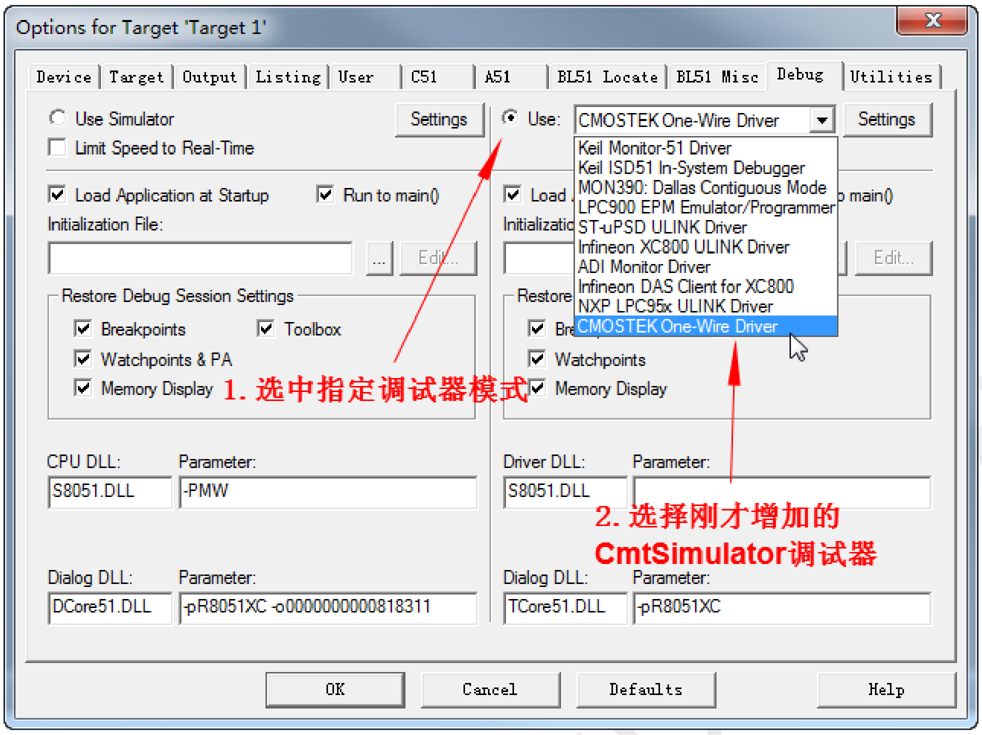
Banks: 2 Bank Area: 0x0000 0xFFFF

[ ] 'far' memory type support

[ ] Save address extension SFR in interrupts

Figure 2-8. Options for Target /Target 1 Interface

- Step 4: Select the "debug" tab page on the "Options for Target 'Target 1" interface, select the specified debugger mode (right side), and select the CmtSimulator debugger just added in the drop-down menu, as shown in Figure 2-9.



Options for Target 'Target 1'

Device | Target | Output | Listing | User | C51 | A51 | BL51 Locate | BL51 Misc | Debug | Utilities

( ) Use Simulator
[ ] Limit Speed to Real-Time
[✓] Load Application at Startup
[✓] Run to main()
Initialization File:
Restore Debug Session Settings
[✓] Breakpoints
[✓] Toolbox
[✓] Watchpoints & PA
[✓] Memory Display

CPU DLL: S8051.DLL
Parameter: -PMW
Dialog DLL: DCore51.DLL
Parameter: -pR8051XC -o0000000000818311

(•) Use: CMOSTEK One-Wire Driver

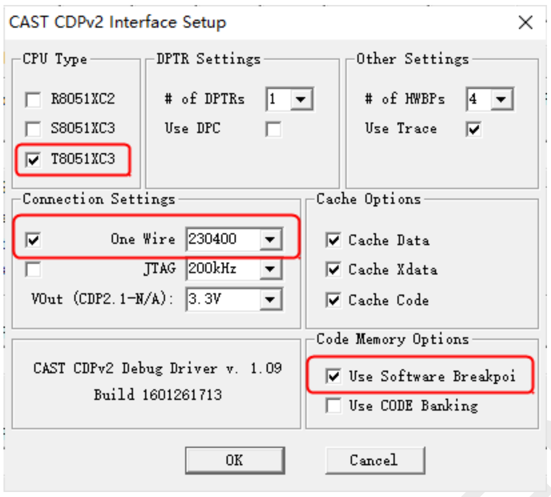
Keil Monitor-51 Driver
Keil ISD51 In-System Debugger
MON390: Dallas Contiguous Mode
LPC900 EPM Emulator/Programmer
ST-uPSD ULINK Driver
Infineon XC800 ULINK Driver
ADI Monitor Driver
Infineon DAS Client for XC800
NXP LPC95x ULINK Driver
CMOSTEK One-Wire Driver

[✓] Load
[✓] Watchpoints
[✓] Memory Display

Driver DLL: S8051.DLL
Parameter:
Dialog DLL: TCore51.DLL
Parameter: -pR8051XC

Figure 2-9. Select the CmtSimulator Debugger

- Step 5: Click the "settings" button on the right side of the selection menu to pop up the settings window, as shown in Figure 2-10.



CAST CDPv2 Interface Setup

CPU Type
- [ ] R8051XC2
- [ ] S8051XC3
- [✓] T8051XC3

DPTR Settings
- # of DPTRs: 1
- Use DPC: [ ]

Other Settings
- # of HWBPs: 4
- Use Trace: [✓]

Connection Settings
- [✓] One Wire: 230400
- [ ] JTAG: 200kHz
- VOut (CDP2.1-N/A): 3.3V

Cache Options
- [✓] Cache Data
- [✓] Cache Xdata
- [✓] Cache Code

CAST CDPv2 Debug Driver v. 1.09
Build 1601261713

Code Memory Options
- [✓] Use Software Breakpoi
- [ ] Use CODE Banking

Figure 2-10. Set debugger parameters

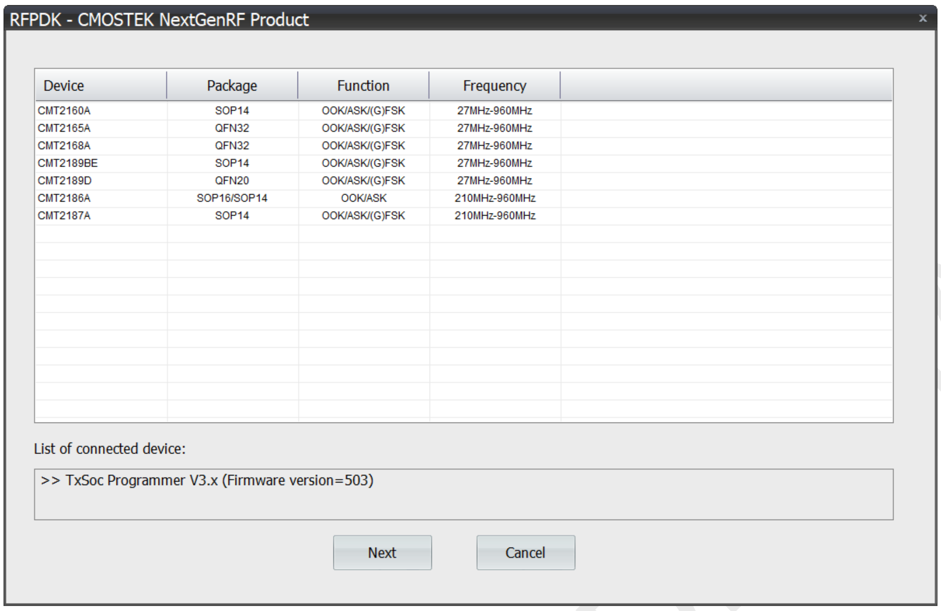
As shown in the window above, you first need to select the correct target core, that is, select T8051XC3. Then, leaving the "One Wire" interface selected by default in "Connection Settings", user can select a suitable debug baud rate. Finally, be sure to check "Use Software Breakpoint" in Code Memory Options. It is recommended to configure other options by default as shown in the figure above.

Note:

1. *When CMT218xA uses a higher running speed at baud rate debugging (i.e. when the internal RC is not divided, the default rate is 24MHz), you can select baud rate of 460800, 230400, and 115200, all of which can have a better debugging result; Lower debug baud rates are not recommended under the condition.*
2. *Similarly, when the CMT218xA uses a lower running speed, the debugging baud rate needs to be reduced. Otherwise, if the CMT218xA core runs slowly with a high debugging baud rate, it is easy to automatically exit the simulation mode due to simulation abnormalities.*

Up till now, 1-WIRE debugging related settings have been completed. After ensuring that the Simulator is connected to the target board, you can enter the Keil online simulation debugging mode after the debugging target code is compiled.

3 TxSoC RFPDK Introduction



RFPDK - CMOSTEK NextGenRF Product

Device	Package	Function	Frequency
CMT2160A	SOP14	OOK/ASK/(G)FSK	27MHz-960MHz
CMT2165A	QFN32	OOK/ASK/(G)FSK	27MHz-960MHz
CMT2168A	QFN32	OOK/ASK/(G)FSK	27MHz-960MHz
CMT2189BE	SOP14	OOK/ASK/(G)FSK	27MHz-960MHz
CMT2189D	QFN20	OOK/ASK/(G)FSK	27MHz-960MHz
CMT2186A	SOP16/SOP14	OOK/ASK	210MHz-960MHz
CMT2187A	SOP14	OOK/ASK/(G)FSK	210MHz-960MHz

List of connected device:

```
>> TxSoc Programmer V3.x (Firmware version=503)
```

Figure 3-1. TxSoC RFPDK Interface

TxSoC RFPDK is a product GUI tool software for the CMT216x and CMT218x series. It integrates code burning (OTP, MTP), Sub-1G transmission configuration parameter generation and export functions.

3.1 MTP Burning Interface

3.1.1 MTP Burning Hardware Construction

As shown in Figure 3-2, when burning MTP with CMT218xA, you need to prepare the following:

- The target chip/board to be burned is CMT2186A-DM board (with SOP burning socket) as the target in the illustration. The DM board supports the chip placed in the burning socket for connection burning;
- Programmer (for TxSoC) burner, noted that this burner is an online burner, which requires a PC computer to burn through the operation of the host computer. It is suitable for burning verification function in the research and development stage; If users need an offline mass-produced burner or get more related content, please refer to the operation instructions of offline burners of CMT series products. This article only describes the online type, and so does the burning interface and precautions.
- Computer and USB cable (Type A to B)

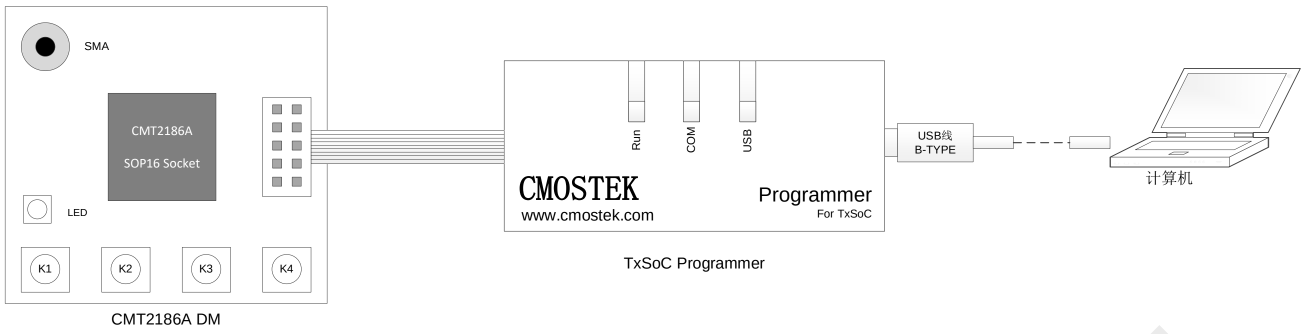


Figure 3-2. CMT218xA Online Burning Schematic

Note:

1. *If users need to burn on the board, they need to reserve the burning interface of S3S and RSTn/D0 pins in advance to ensure that other peripherals on the board do not affect the normal operation of these burning pins;*
2. *The corresponding relationship between the S3S burning interface and the RSTn/D0 pin as well as the 10-pin horn arrangement is shown in Figure 3-3 below.*

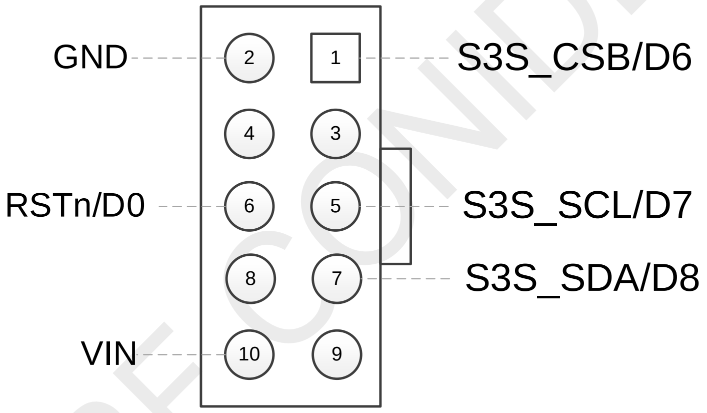
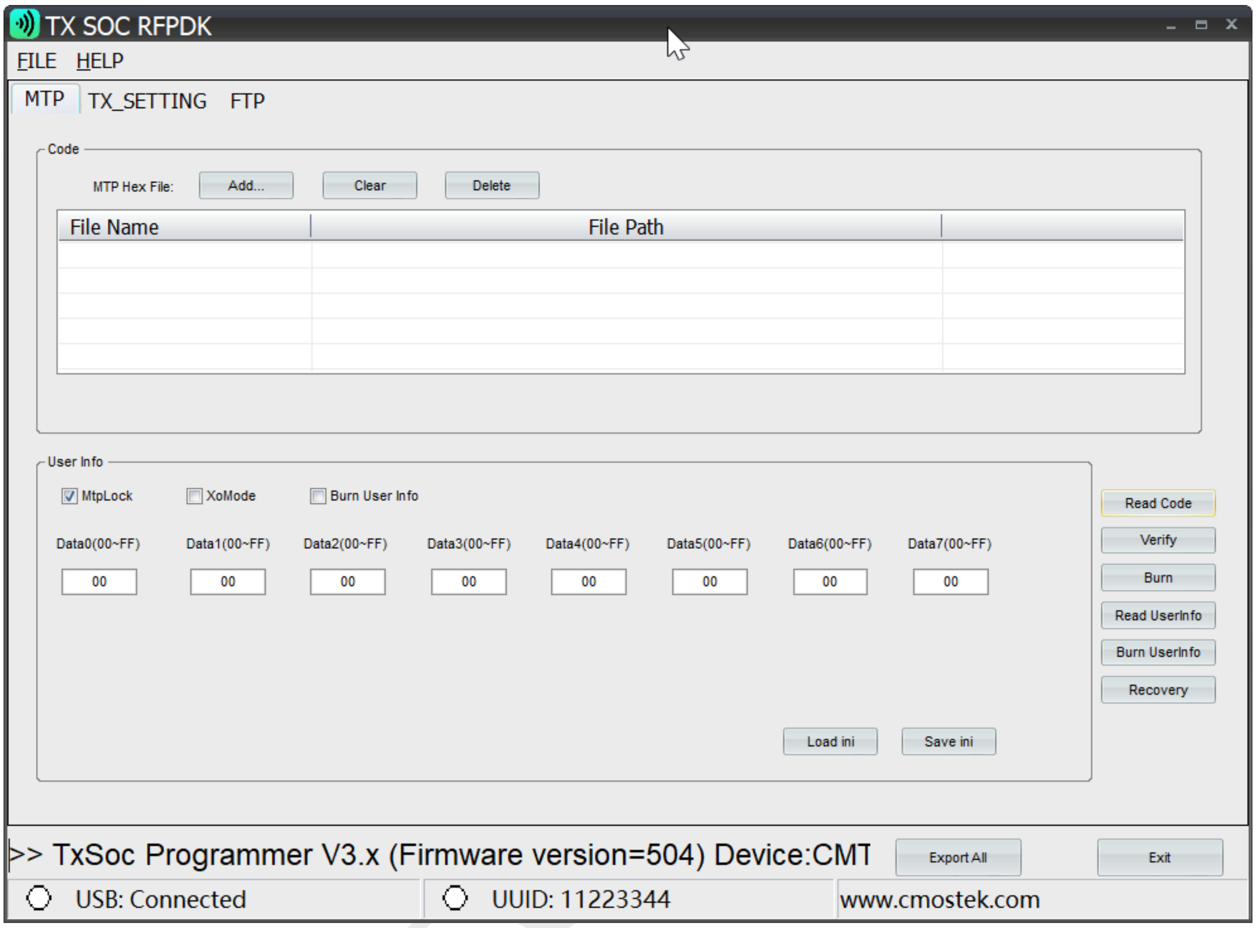


Figure 3-3. CMT218xA 10-pin Horn Burning Arrangement

3.1.2 MTP Burning Configuration

After the burning hardware is built in the above way, the burning interface is shown in the following figure.



>> TxSoc Programmer V3.x (Firmware version=504) Device:CMT

USB: Connected | UUID: 11223344 | www.cmostek.com

Figure 3-4. CMT218xA online MTP burning interface

Wherein,

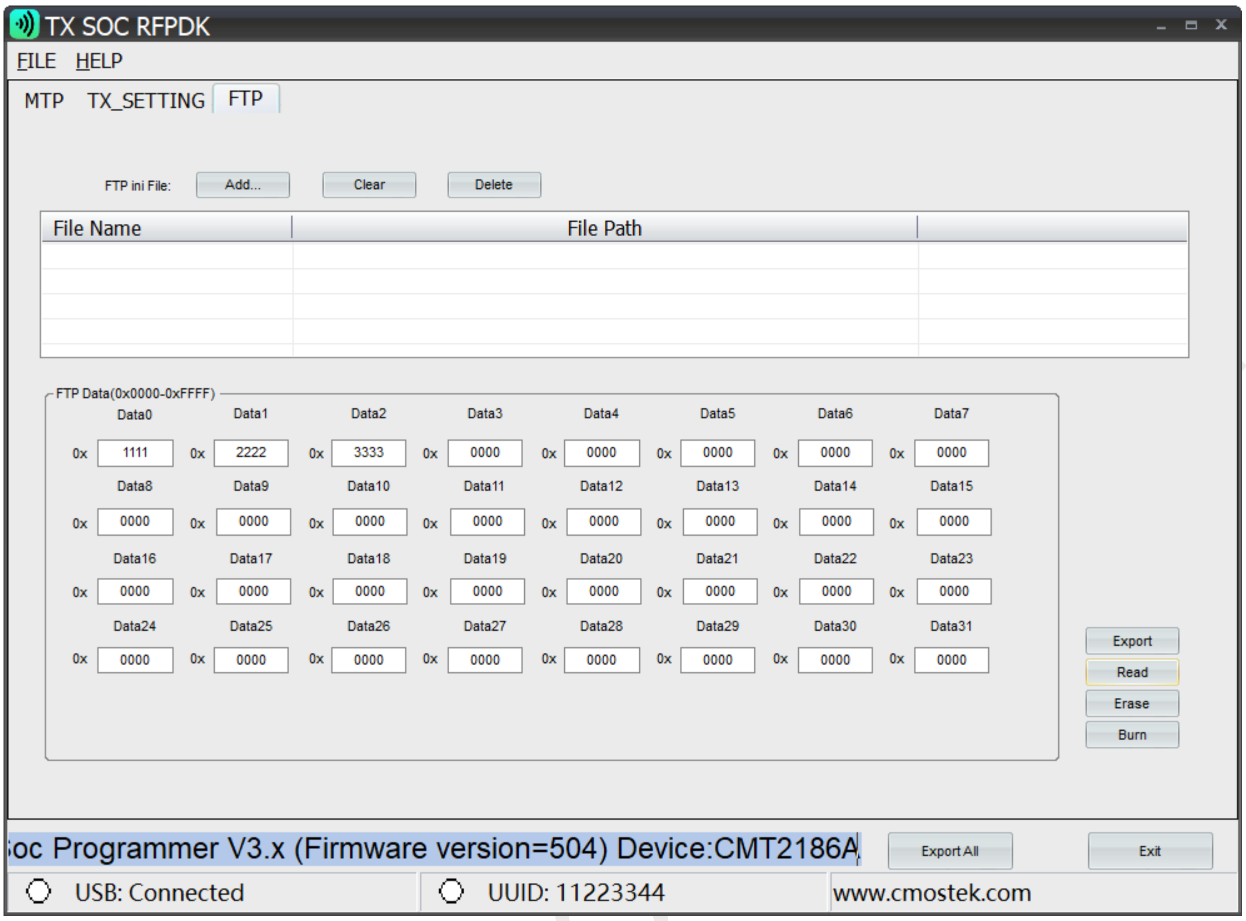
- Click "Add" to Add the HEX file to be burned;
- Click "Clear" to Clear the imported HEX file contents;
- Click "Delete" to Delete the HEX file information in the list;
- In the User Info area, burning options is provided here:
  - MtpLock, check to enable MTP burning lock, that is, reading is not supported;
  - XoMode, check to enable the chip to use an external 26MHz crystal as the main clock operating mode summary;
  - Burn User Info, check to enable the User Info area information. For specific content, you can fill in the text box below. User Info supports 8 bytes. In this area, although MtpLock is checked, it still supports erasing, writing and reading, which can be used by users to identify the burning firmware version or other information.
  - The Load ini button is the ini information file exported before loading. This file contains the relevant settings of User Info;

  - The Save ini button exports the current User Info settings, which is convenient for next import without re-entering the settings;
- The Read Code button reads the MTP content of the target chip (it can only be read when MtpLock is not effective);
- The Verify button is used to Verify the MTP content of the target chip (it can only be verified when MtpLock is not effective);
- The Burn button is to Burn the target chip (if the current target chip is in the MtpLock effective state, it needs to be Recovery before it can be burned again);
- The Read User Info button reads User Info information;
- The Burn User Info button burns User Info information;
- The Recovery button restores the factory state of the chip and is suitable for unlocking the chip, erasing the code in MTP, and clearing the User Info area information;
- The Export All button exports all current settings as a burned project file, and this burned project file supports import and use in TxSoC Wirter Config (TxSoC offline burner interface software).

Note: Due to the version upgrade, the burning interface will upgrade accordingly. If the burning interface does not match with the document, please check the options according to the interface description.

3.2 EEPROM burning interface

After the burning hardware is built in the above way, you can open the EEPROM burning interface, as shown in the figure below.



FTP ini File:

File Name	File Path

FTP Data(0x0000-0xFFFF)

Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7
0x 1111	0x 2222	0x 3333	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000
Data8	**Data9**	**Data10**	**Data11**	**Data12**	**Data13**	**Data14**	**Data15**
0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000
Data16	**Data17**	**Data18**	**Data19**	**Data20**	**Data21**	**Data22**	**Data23**
0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000
Data24	**Data25**	**Data26**	**Data27**	**Data28**	**Data29**	**Data30**	**Data31**
0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000

oc Programmer V3.x (Firmware version=504) Device:CMT2186A

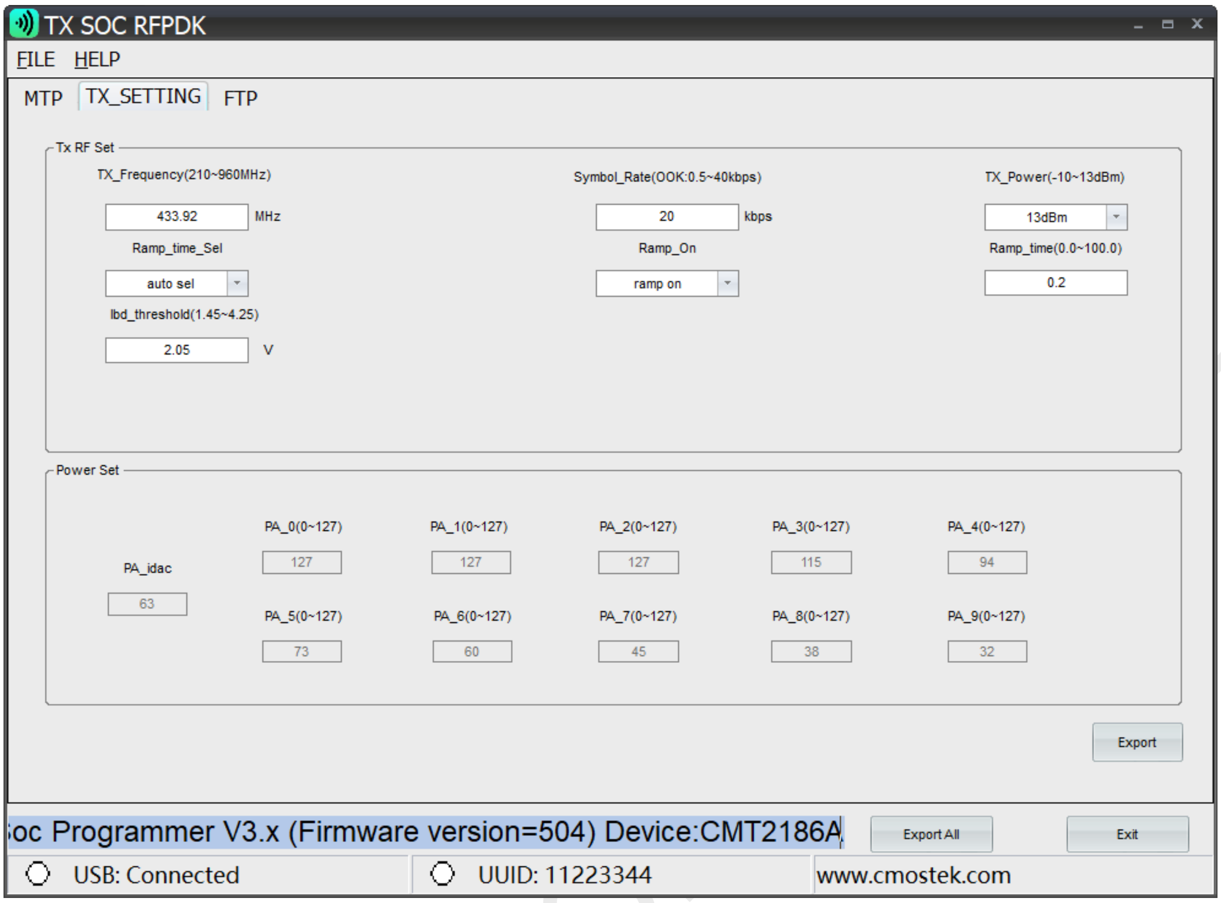
USB: Connected | UUID: 11223344 | www.cmostek.com

Figure 3-5. CMT218xA Online EEPROM Burning Interface

Wherein,

- Click "Add..." to Add the ini file of the EEPROM to be burned (Export through the Export button);
- Click "Clear" to Clear the list of ini files added below;
- Click "Delete" to Delete the files in the ini file list;
- The "EEPROM Data" area is a 64 Bytes EEPROM content filling area;
- The "Export" button is to Export the current EEPROM filling content into an ini file;
- The "Read" button reads the EEPROM content of the current target chip;
- "Erase" button is used to Erase the EEPROM content of the current target chip;
- The "Burn" button is to Burn the content filled in the current EEPROM Data area to the target chip;

3.3 Sub-1G Transmitting Configuration Interface



Tx RF Set

TX_Frequency(210~960MHz): 433.92 MHz
Symbol_Rate(OOK:0.5~40kbps): 20 kbps
TX_Power(-10~13dBm): 13dBm
Ramp_time_Sel: auto sel
Ramp_On: ramp on
Ramp_time(0.0~100.0): 0.2
lbd_threshold(1.45~4.25): 2.05 V

Power Set

PA_idac	PA_0(0~127)	PA_1(0~127)	PA_2(0~127)	PA_3(0~127)	PA_4(0~127)
63	127	127	127	115	94
	PA_5(0~127)	PA_6(0~127)	PA_7(0~127)	PA_8(0~127)	PA_9(0~127)
	73	60	45	38	32

oc Programmer V3.x (Firmware version=504) Device:CMT2186A

USB: Connected UUID: 11223344 www.cmostek.com

Figure 3-6. CMT218xA Sub-1G configuration parameter interface

Wherein,

- Click "Export" to Export the Sub-1G RF configuration file suitable for CMT2186A/2187A (with c code format, which can be compiled and used directly);

4 Introduction to each model evaluation board

4.1 CMT2186A-EB

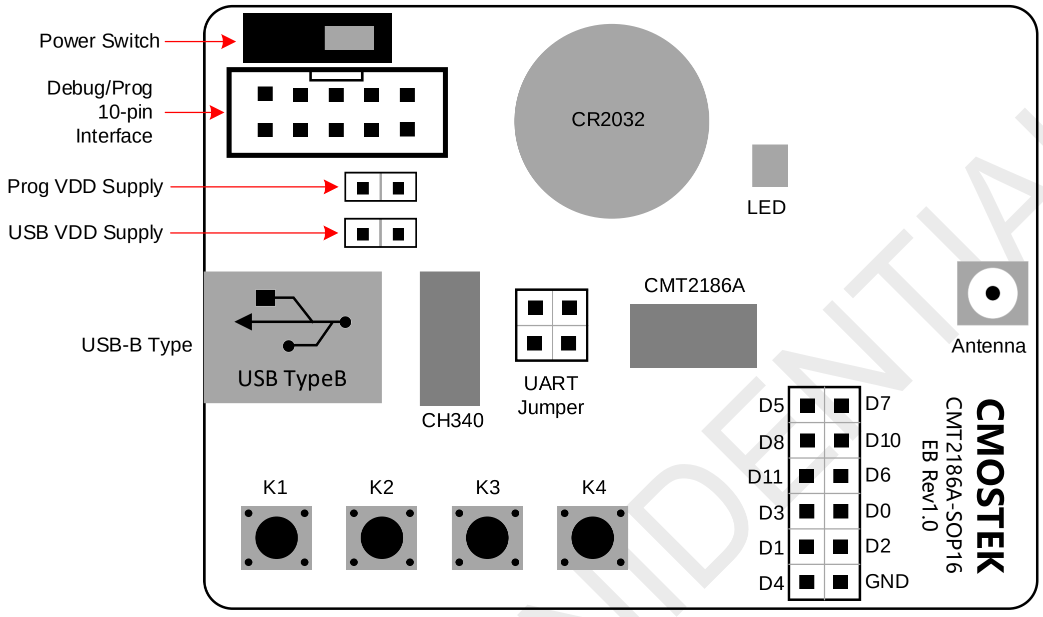


Figure 4-1. Schematic of CMT2186A-EB Board

The CMT2186A-EB board (see the figure above) is a development board designed based on the CMT2186A and integrates on-line simulation debugging and off-line testing and verification. Supports the following functions:

1. Provide 1 cathode drive LED as an indication, controlled by the D8 pin;
2. Four keys are provided as input interfaces, from K0 to K4: RSTn/D0, D6, D7 and D11 respectively;
3. UHF high-frequency transmission adopts a single-ended design and can be connected to an antenna or device test through SMA;
4. The onboard CH340G chip can realize USB to TTL UART, and connect the hardware through the UART Jumper next to it;
5. The external power supply supports two input sources, one is USB power supply (regulated to 3.3 V through the voltage regulator tube), and the other is from the Debug/Prog 10-pin Interface, which directly draws power from the simulation debugger;
6. The PowerSwitch can be powered externally (as mentioned in point 5 above) or on-board CR2032 batteries;
7. Debug/Prog 10-pin Interface, this interface is a composite port: 1-Wire online debugging interface, MTP S3S burning interface and RSTn/D0 reset pin control;
8. Other GPIOs are drawn out through the onboard CN4, which can be used for testing and analysis, and can also be used as sockets for expanded functions.

4.2 CMT2187A-EB

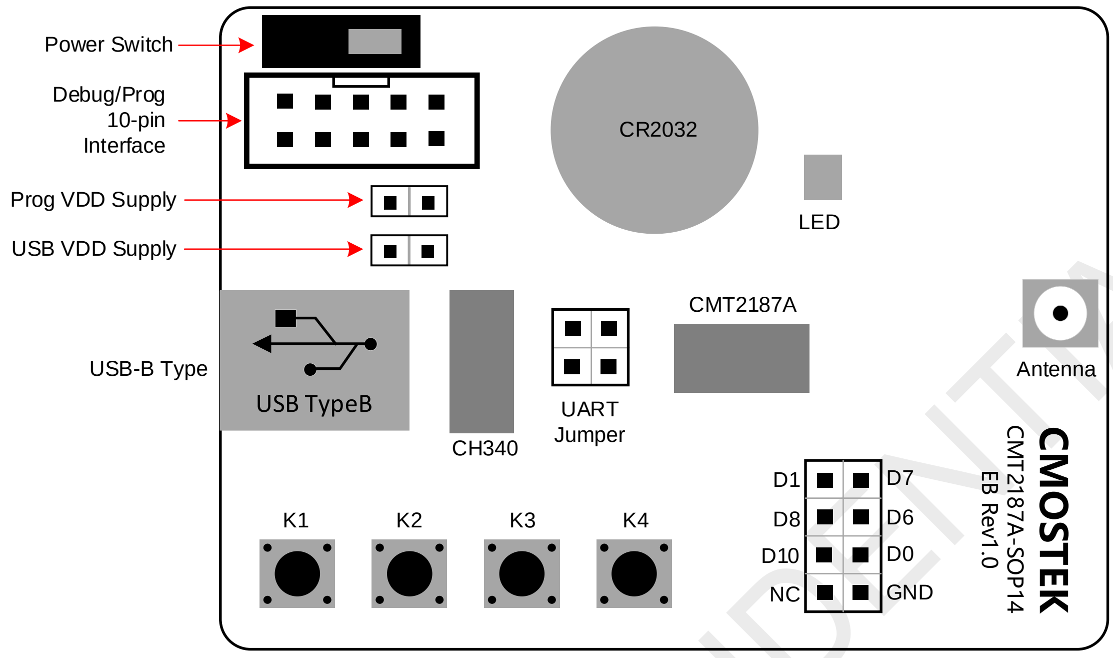


Figure 4-12. Schematic of CMT2187A-EB board

The CMT2187A-EB board (see the figure above) is a development board designed based on the CMT2187A and integrates on-line simulation debugging and off-line testing and verification. It supports the following functions:

1. Provide 1 cathode drive LED as an indication, controlled by the D8 pin;
2. Four keys are provided as input interfaces, from K0 to K4: RSTn/D0, D6, D7 and D1 respectively;
3. UHF high-frequency transmission adopts a single-ended design and can be connected to an antenna or device test through SMA;
4. The on-board CH340G chip can realize USB to TTL UART, and connect the hardware through the UART Jumper next to it;
5. The external power supply supports two input sources, one is USB power supply (regulated to 3.3 V through the voltage regulator tube), and the other is from the Debug/Prog 10-pin Interface, which directly draws power from the simulation debugger;
6. The PowerSwitch can be powered externally (as mentioned in point 5 above) or on-board CR2032 batteries;
7. Debug/Prog 10-pin Interface, this interface is a composite port: 1-Wire online debugging interface, MTP S3S burning interface and RSTn/D0 reset pin control;
8. All GPIOs are also extracted through the onboard CN4, which can be used for testing and analysis, and can also be used as sockets for expanded functions.

5 Revise Record

Table 5-1. Revise Record

Version No.	Chapter	Change Description	Date
0.1	All	Initial version	2024-11-11

6 Contacts

Shenzhen Hope Microelectronics Co., Ltd.

Address:	30th floor of 8th Building, C Zone, Vanke Cloud City, Xili Sub-district, Nanshan, Shenzhen, GD, P.R. China
Tel:	+86-755-82973805 / 4001-189-180
Fax:	+86-755-82973550
Post Code:	518052
Sales:	sales@hoperf.com
Website:	www.hoperf.com