USR-GM3 SDK User Manual

Ver 1.0

Content

1. Introduction

The below resources are provided on GM3 SDK :

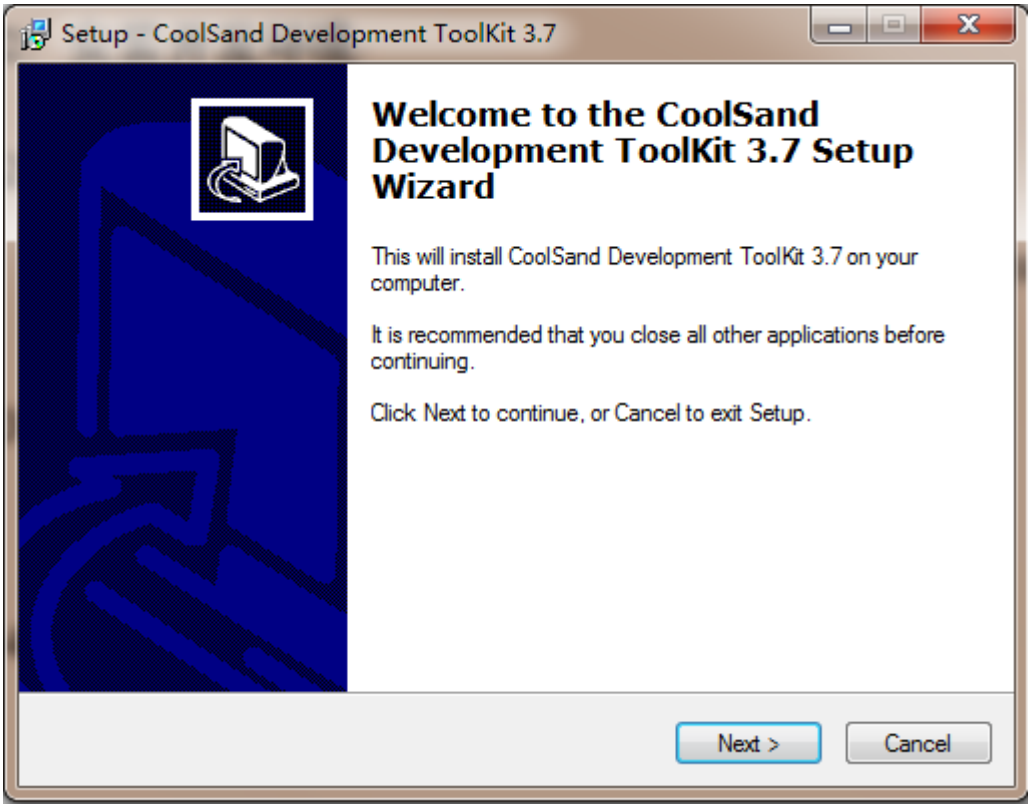
1 Universal Asynchronous Receivers and Transmitters (UART)
5 Timer
2 Simultaneous TCP or UDP Sockets
SMS Send and Receive
System AT Command
Custom AT Command
RTC Clock Interface
10 Reuse GPIO
Some System Information

2. Installation

2.1. Install SDK

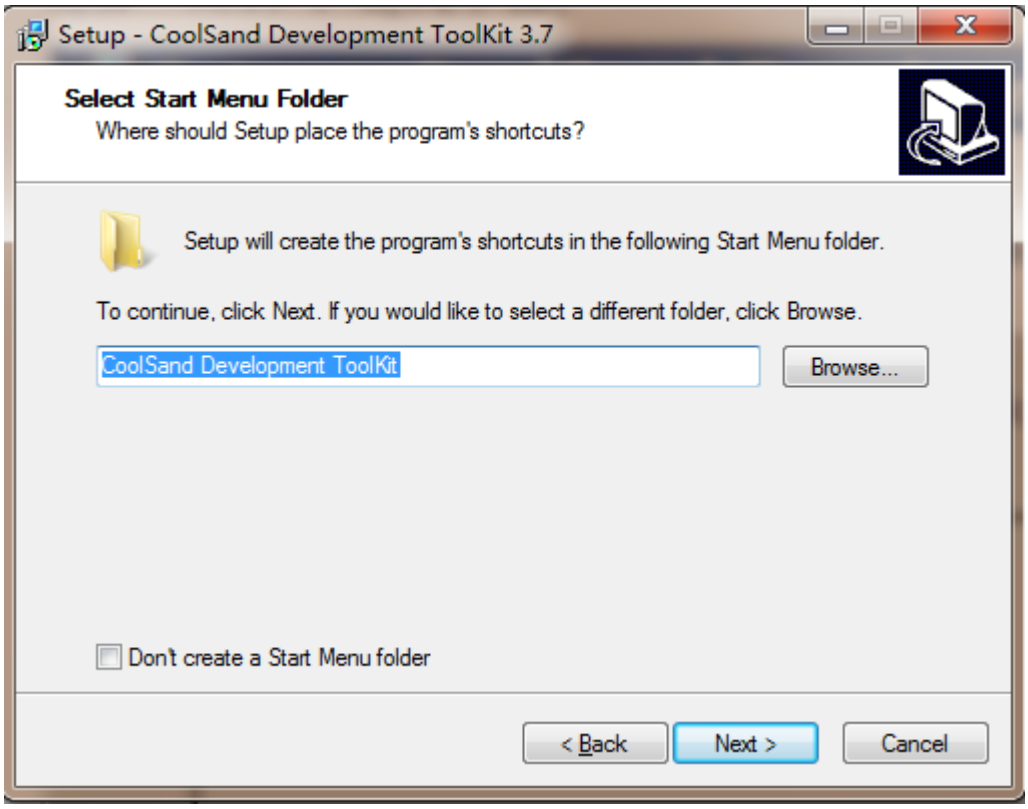
Double click on the file “CSDTK3.7_Cygwin1.5.25_Svn_1.5.4_Full_Setup.exe”, and operate according to the following steps.

（1） Choose Next.

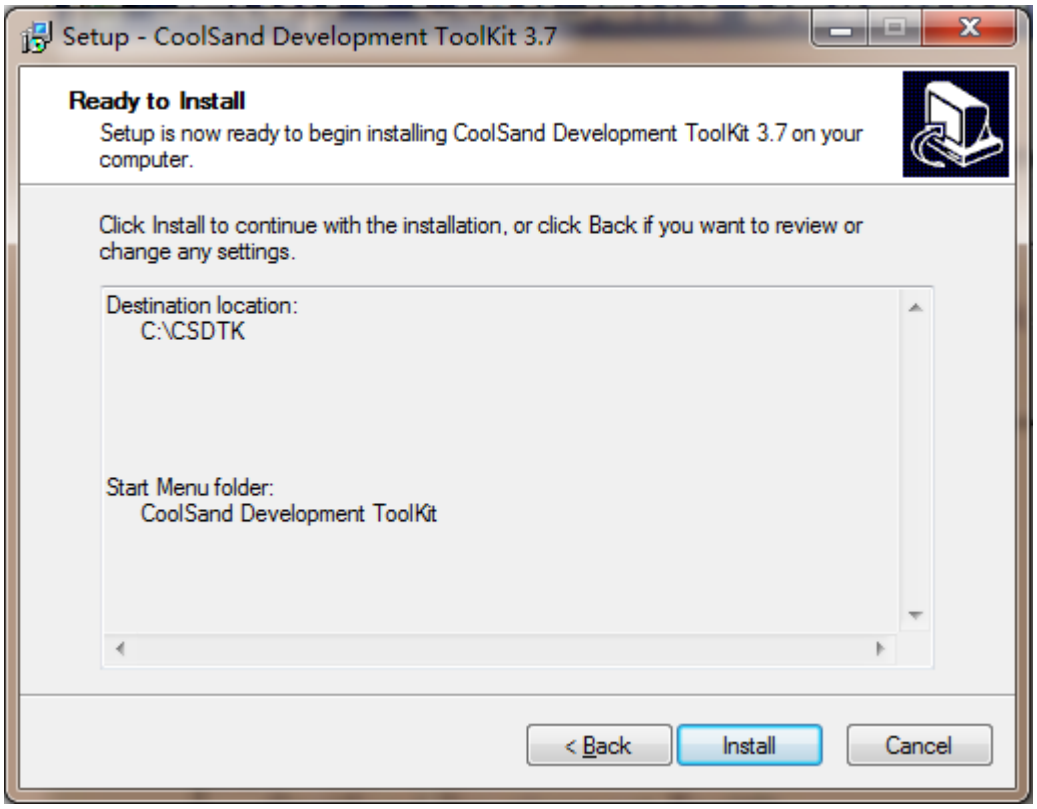


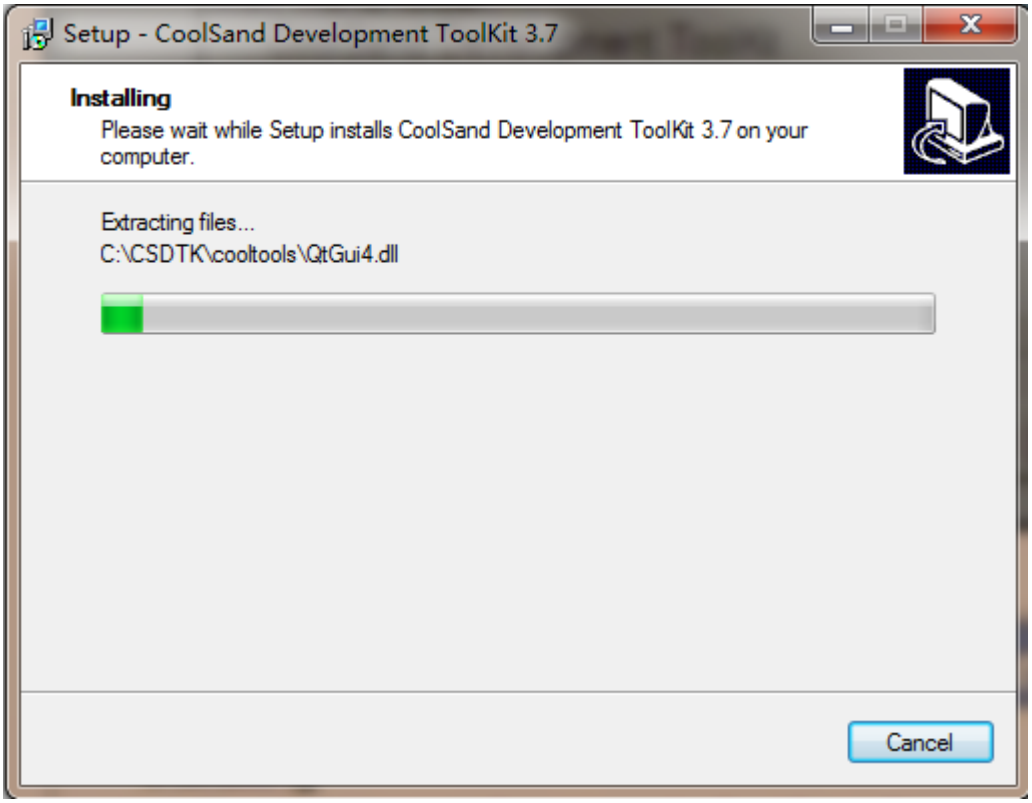
（2） Keep default，and choose next.

（3） Keep default，and choose next.

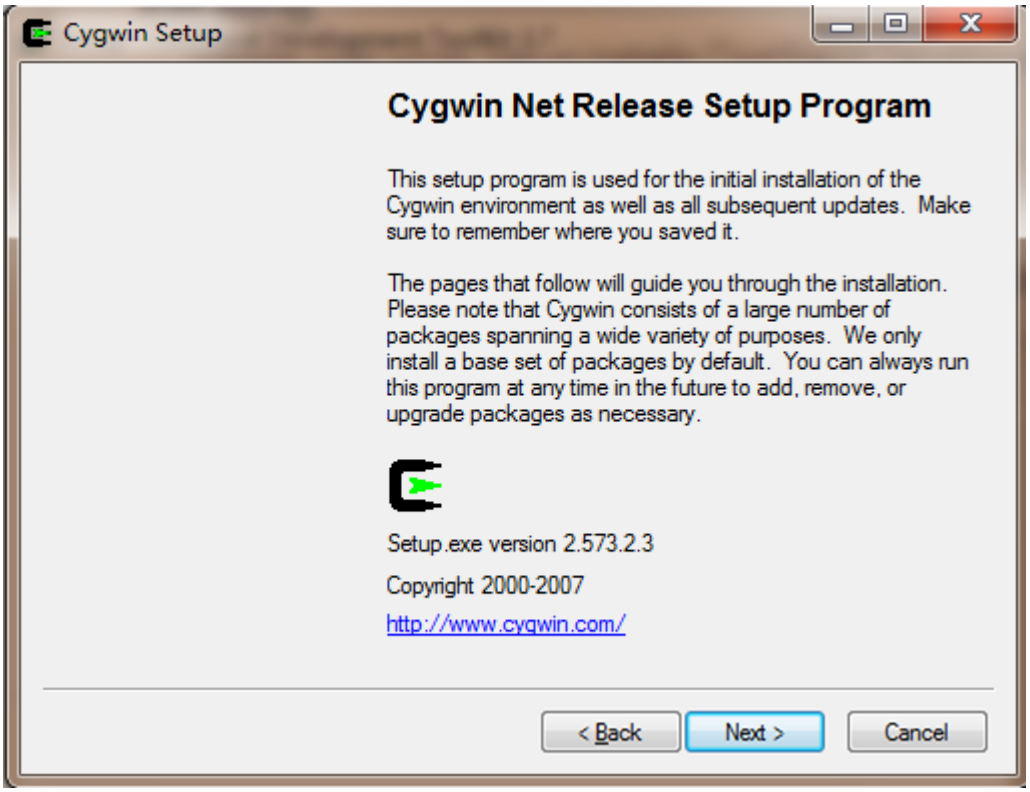


（4） Choose install.

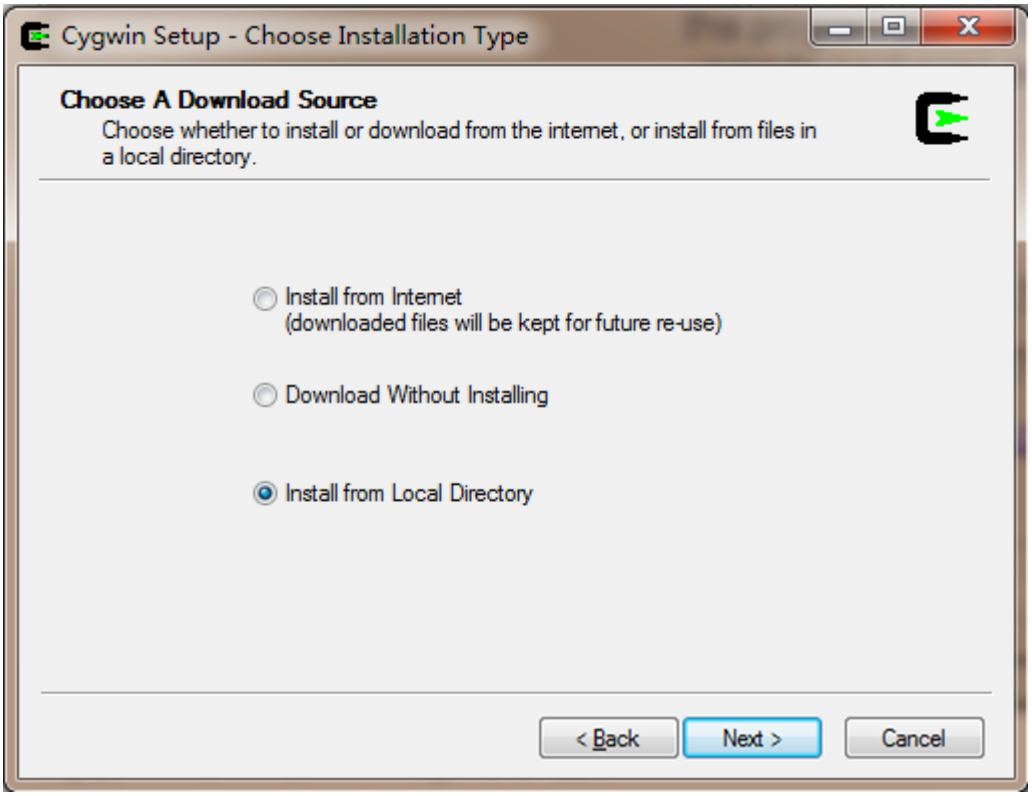




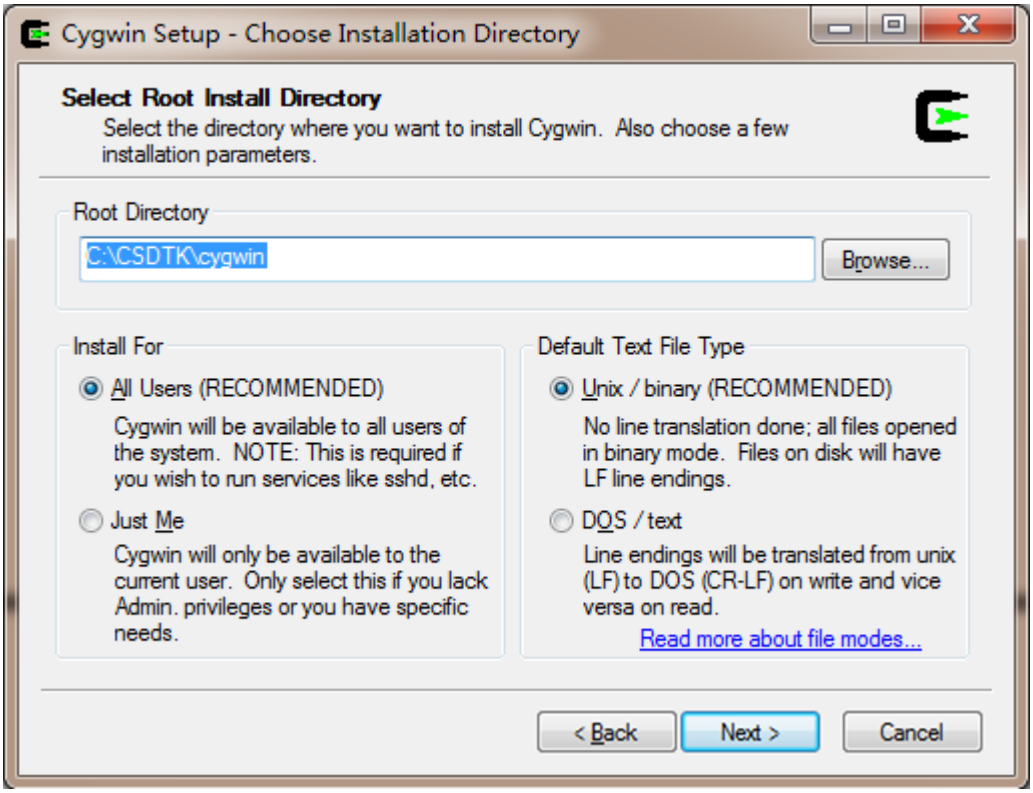
（5） Waiting the auto installation process to complete, and choose next.



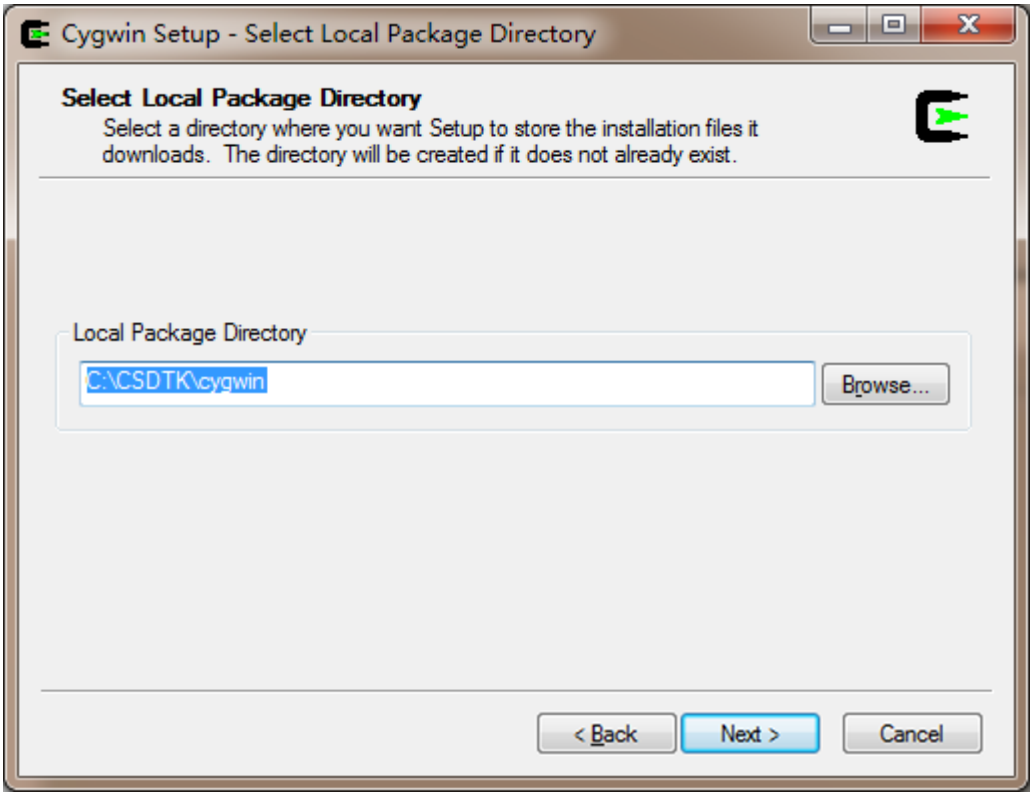
（6） Keep default,and choose next.



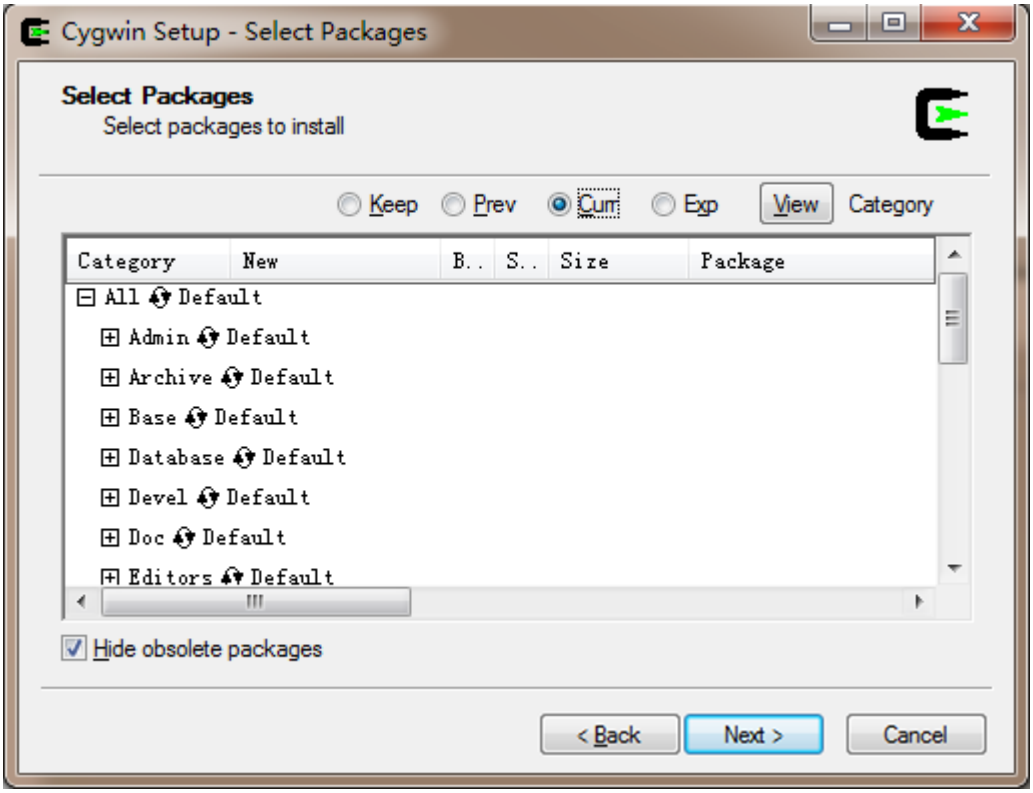
（7） Keep default，and choose next.



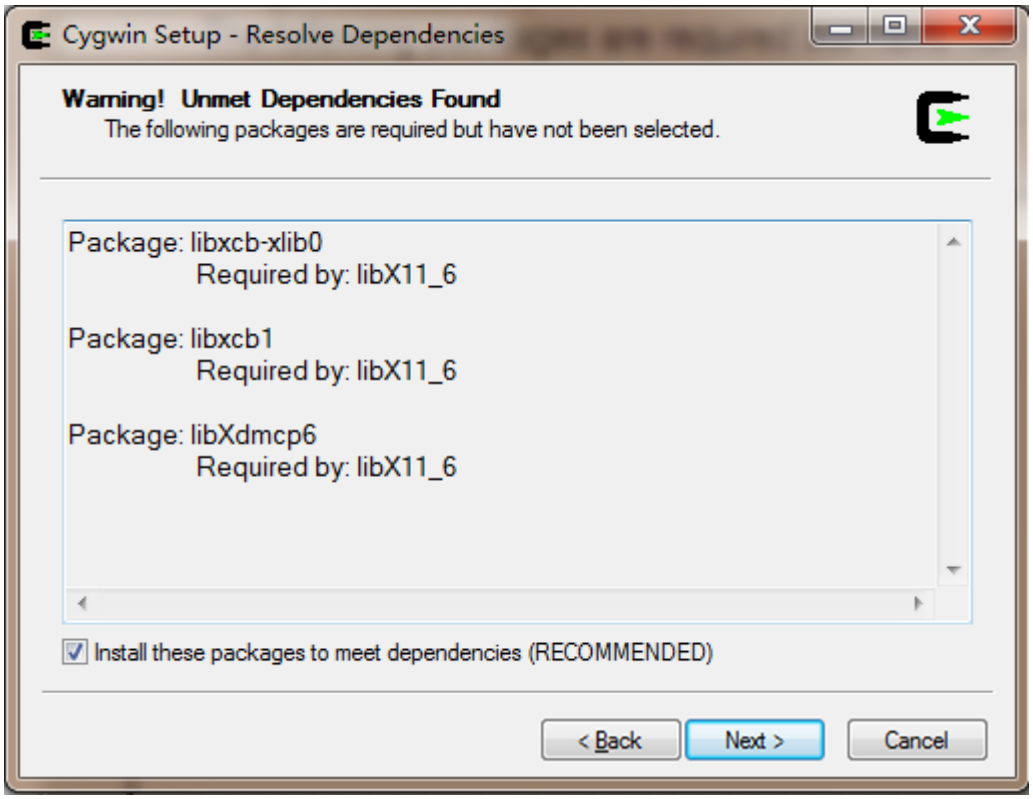
（8） Keep default，and choose next.



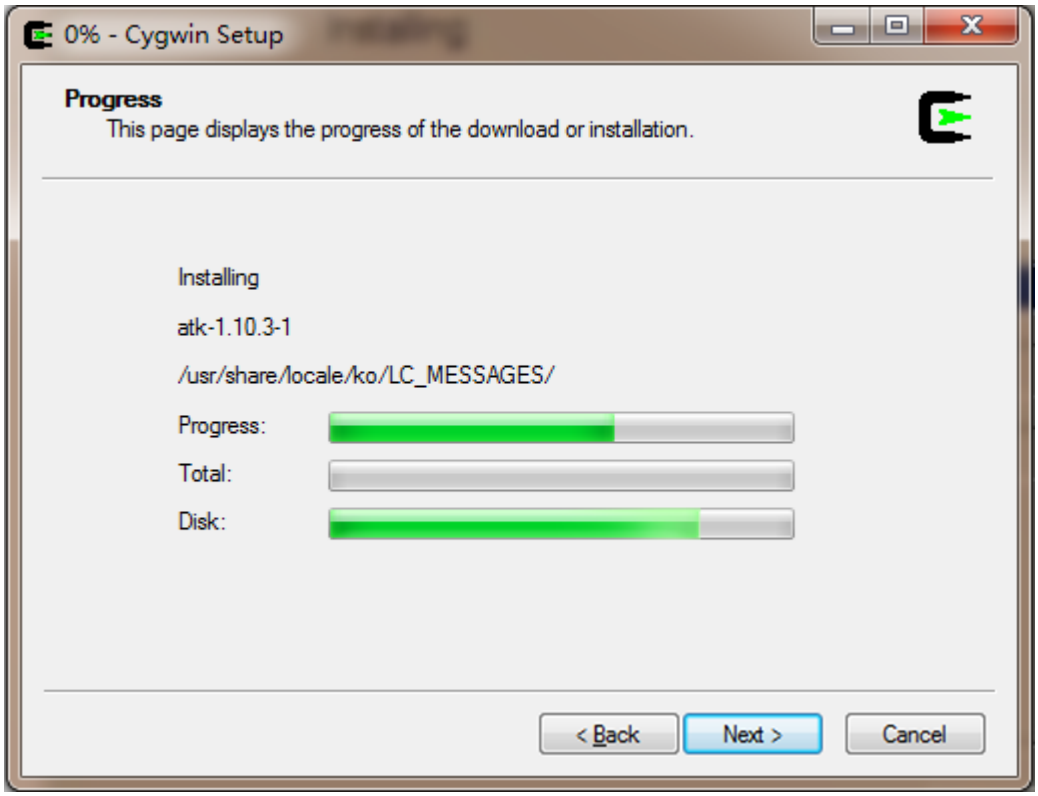
（9） Keep default，and choose next.



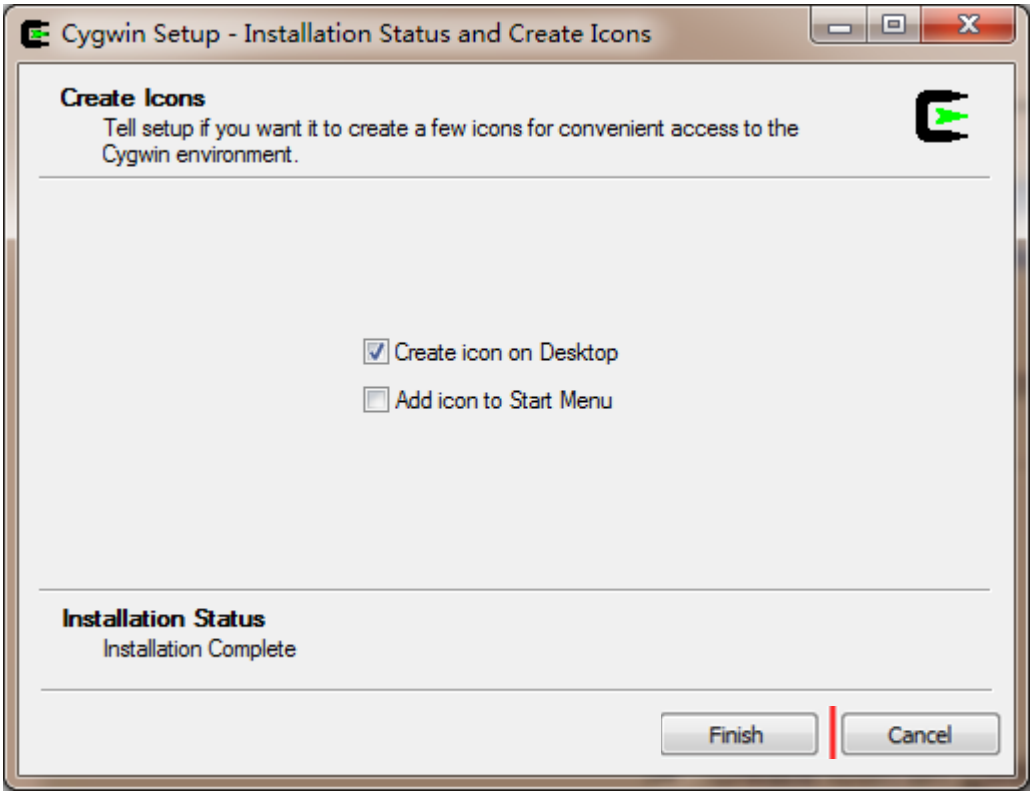
（10） Keep default，and choose next.

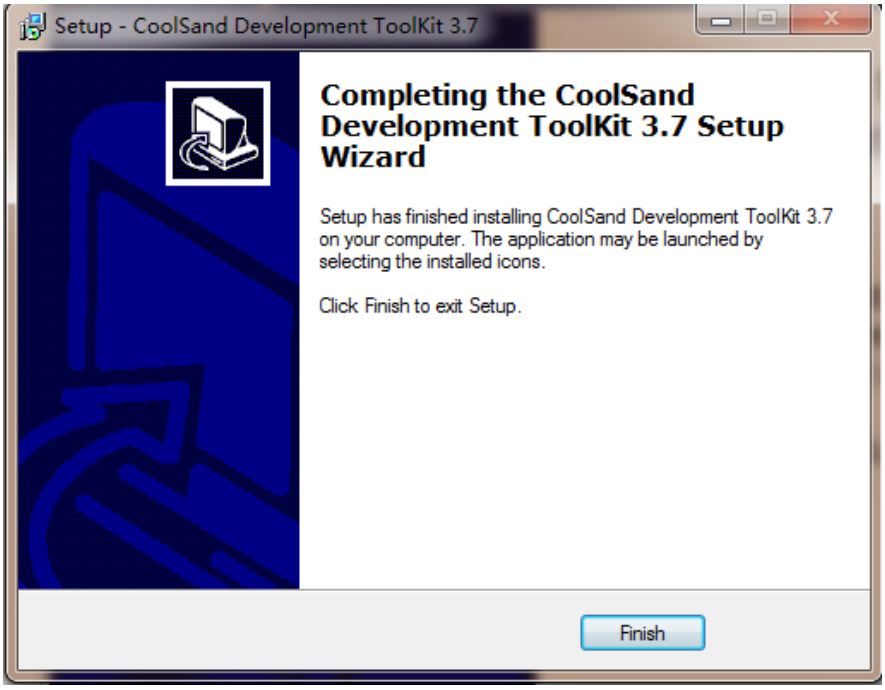


（11） Waiting the auto installation process to complete, and choose next.



（12） Waiting the auto installation process to complete, and choose finish.

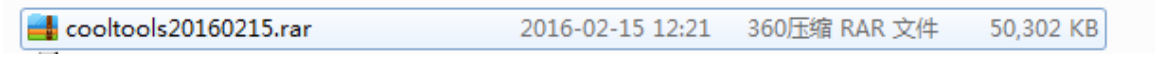




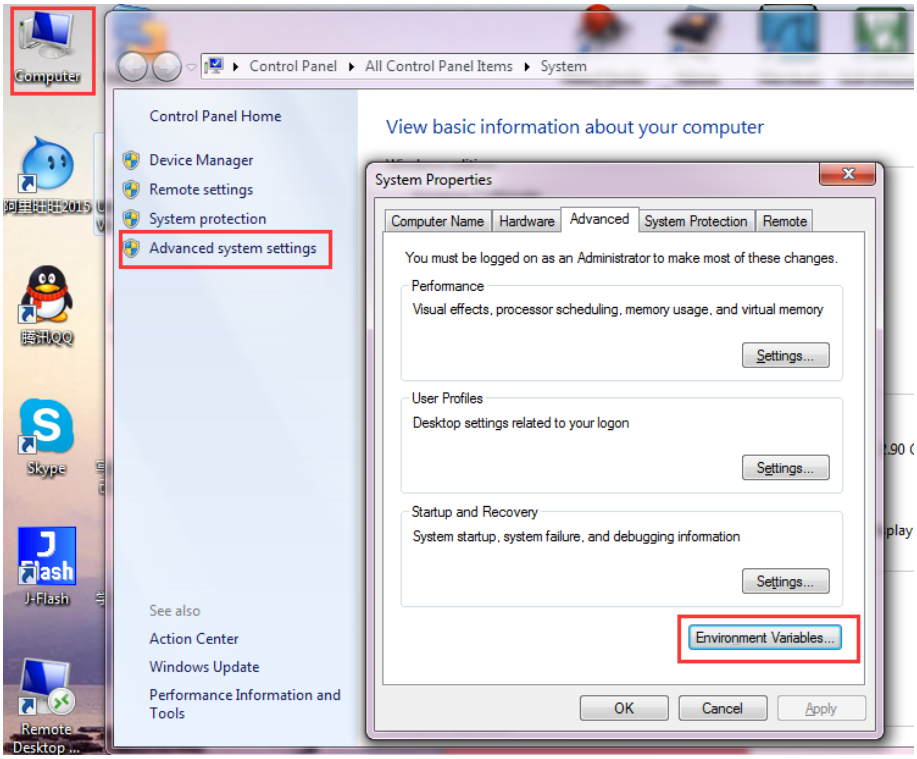
（13） Update cooltools.

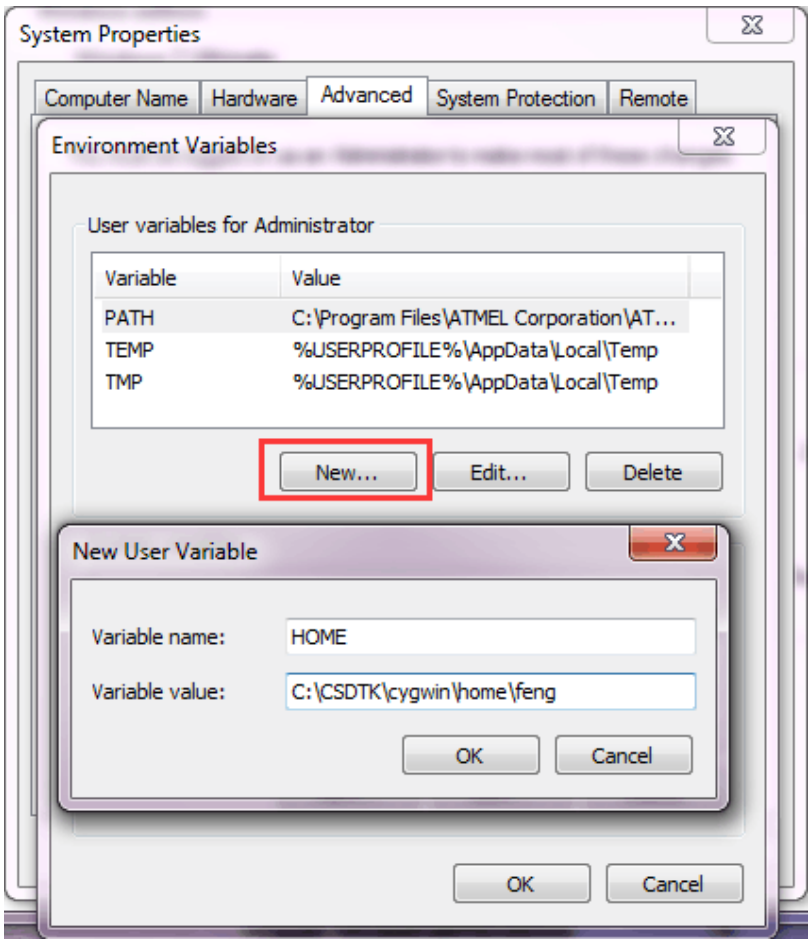
Replace the folder “C:\CSDTK\cooltools” with the new file.

Note: keep updated file in the original file path.

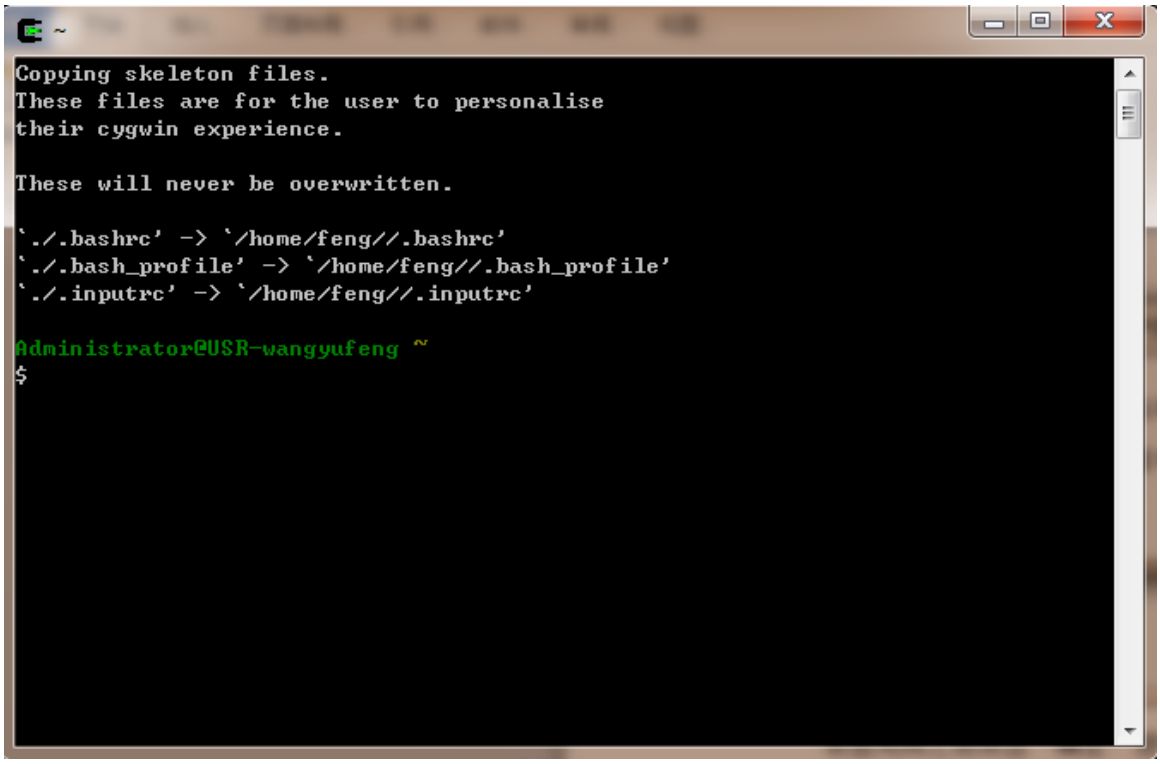


（14） Set a HOME environment variable.





（15） Start the cygwin application.



2.2. Code configuration path

（1） edit file “C:\CSDTK\cygwin\.bashrc file” with UltraEdit;If the startup tips into DOS format,choose NO

（2） Line 118:

export PROJ_ROOT=`cygpath "C:\projects"`

```
116
117 # Root directorty for all the projects
118 export PROJ_ROOT=`cygpath "C:\projects"`
119
```

Replace with:

export PROJ_ROOT=`cygpath "D:\projects"`

```
116
117 # Root directorty for all the projects
118 export PROJ_ROOT=`cygpath "D:\projects"`
119
120 # type work <project_name> to switch the environment to the required project
121 alias  work='source ./cygenv.sh'
```

（3） Line 124:

export PATH=/usr/bin:/crosscompiler/bin:/cooltools/:

```
123 # set path to include the crosscompiler and the cooltools
124 export PATH=/bin:/usr/bin:/crosscompiler/bin:/cygdrive/C/CSDTK/cooltools:
125
126 # Create the link /cygdrive/n -> /n if it does not exist
```

Replace with:

export PATH=/bin:/usr/bin:/crosscompiler/bin:/cygdrive/C/CSDTK/cooltools:/cygdrive/C/Program\ Files/Xoreax/IncrediBuild:

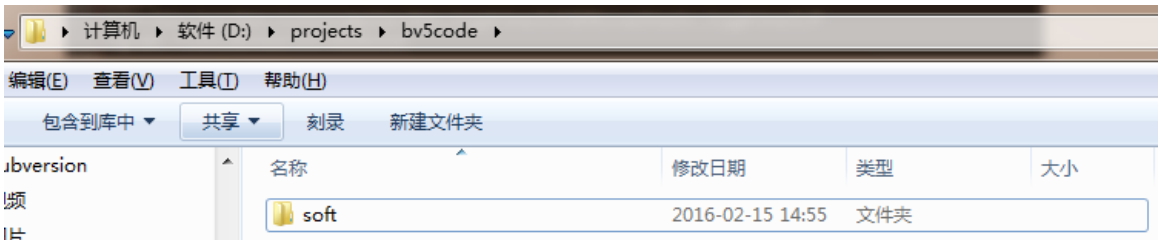
```
# set path to include the crosscompiler and the cooltools
export
PATH=/bin:/usr/bin:/crosscompiler/bin:/cygdrive/C/CSDTK/cooltools:/cygdrive/C/P
rogram\ Files/Xoreax/IncrediBuild:
```

（4） Save file.

（5） Restart the cygwin.

3. Build Project

（1） Please uncompress the demo code file “soft” to the derictory ”D:\projects\bv5code”.



（2） Switch to code work directory and input command “work bv5code”. You need to do it when cygwin starts.

（3） Build the project with the following command :

ctmake -j 16 CT_TARGET=8851cl_rdamodem CT_USER=FAE CT_RELEASE=debug WITH_SVN=0 CT_MODEM=1 CT_PRODUCT=AtCommandS at/usr/

If the building is successful, lod files will be generated.

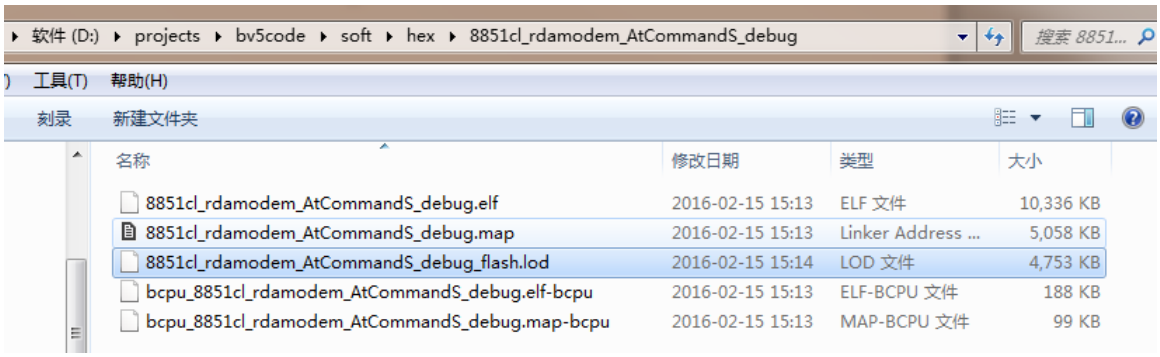
```
GEN               stripped (rm syms) elf file stripped_bcpu_elf_file.elf
LDGEN             8809.ld
LD                8851cl_rdamodem_AtCommandS_debug.elf
CP                Elf binary & map file


SREC              8851cl_rdamodem_AtCommandS_debug.srec for flash/romulator

-----------------------------------------------------
8851cl_rdamodem_AtCommandS_debug_flash.lod Generated
-----------------------------------------------------
```

The building will fail if there are compiling errors.building process will stop at where errors occurs. So you can get some compile information to help you to debug your program.

（4） You will find the lod file in the following directory.



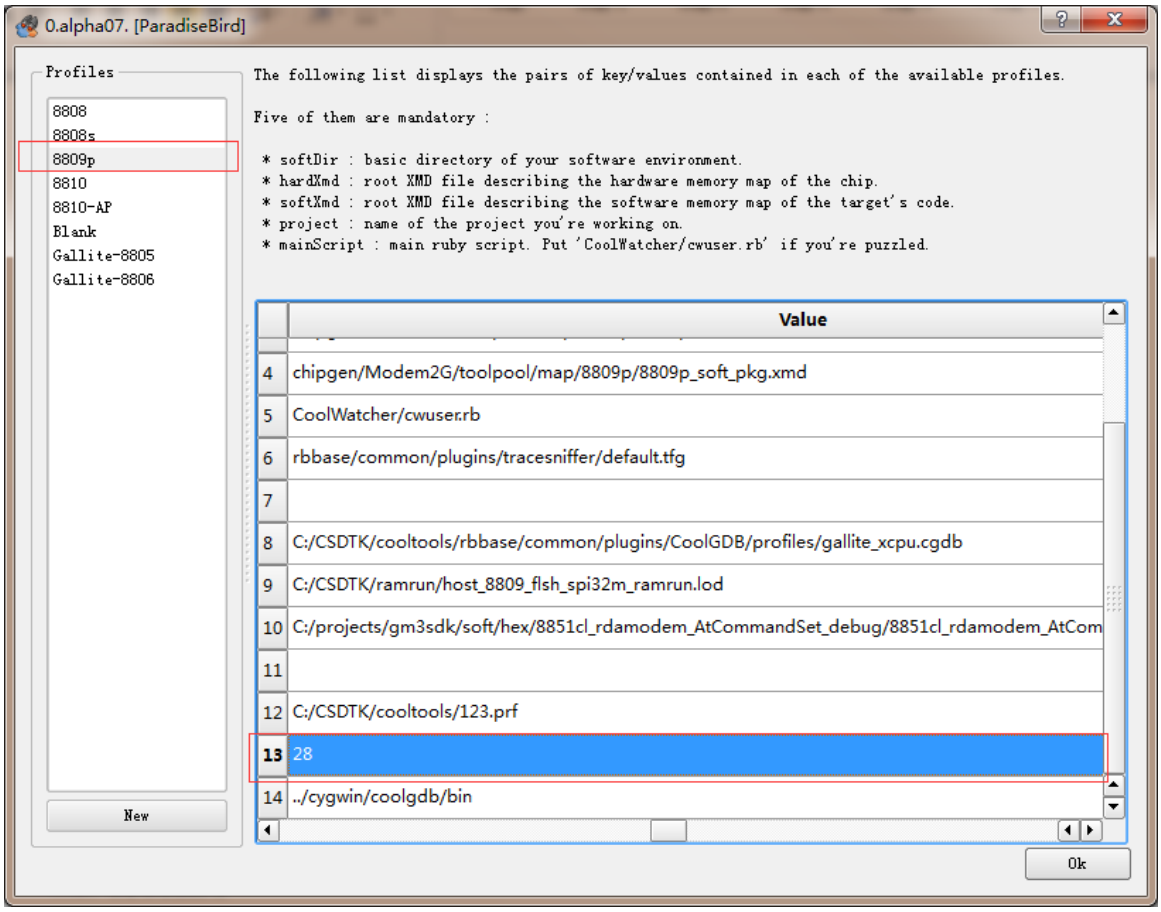
（5） The building option CT_RELEASE=debug means making a debug version. If you want to make a release version, you should use CT_RELEASE=release instead of CT_RELEASE=debug.

4. Download and Debug

4.1. Download

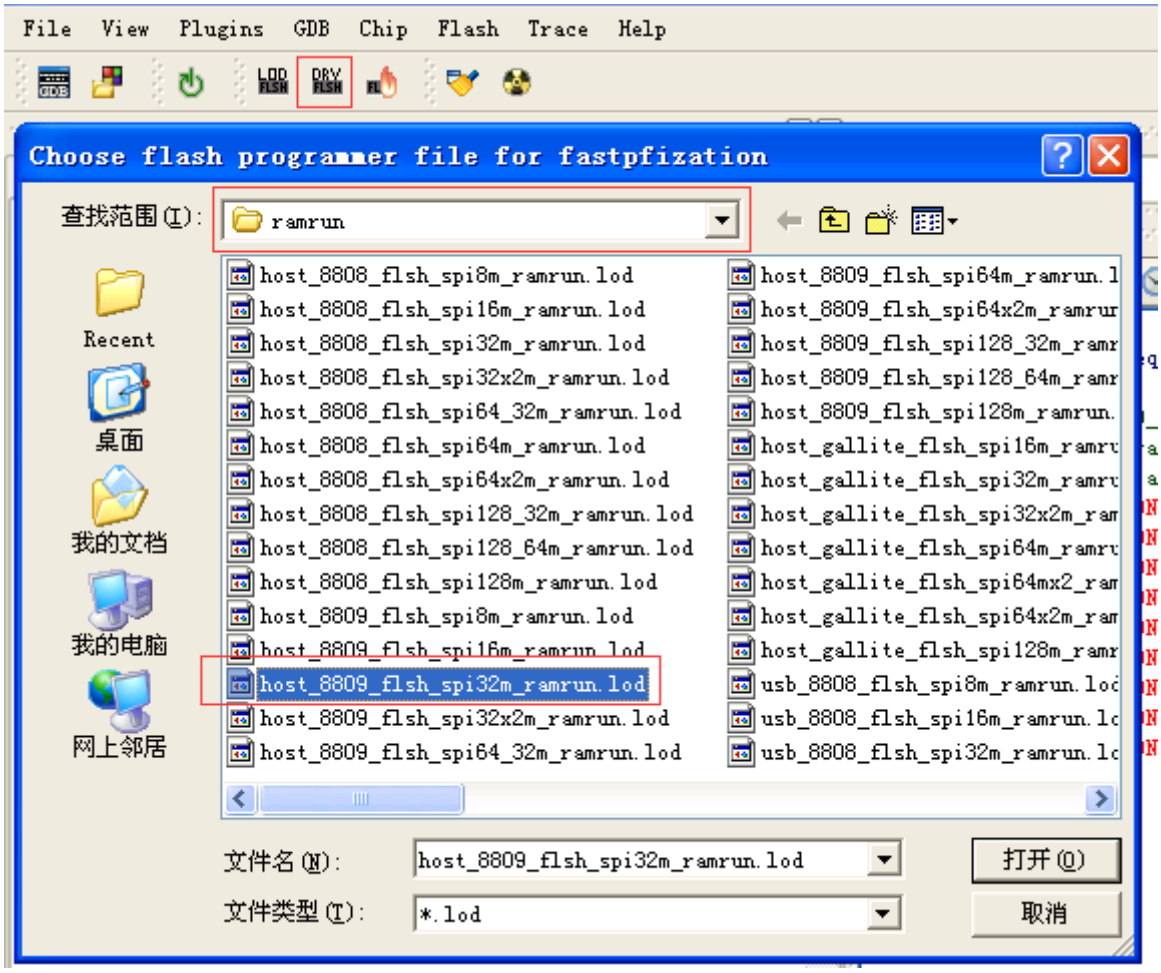
（1） Start the debug tool “coolwatcher.exe” at directory “C:\CSDTK\cooltools”.

（2） Choose 8809p in Profiles item, enter a appropriate serial number on line 13 in item value, then click OK.

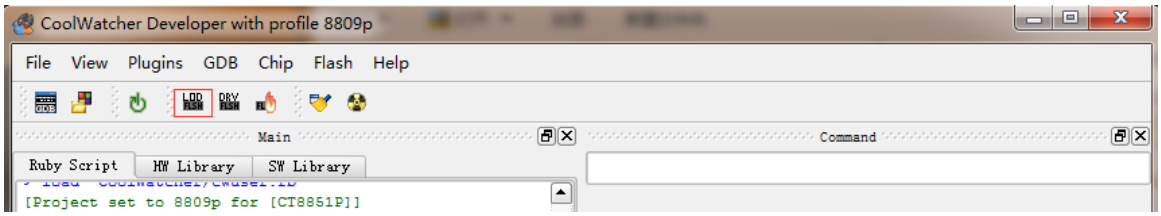


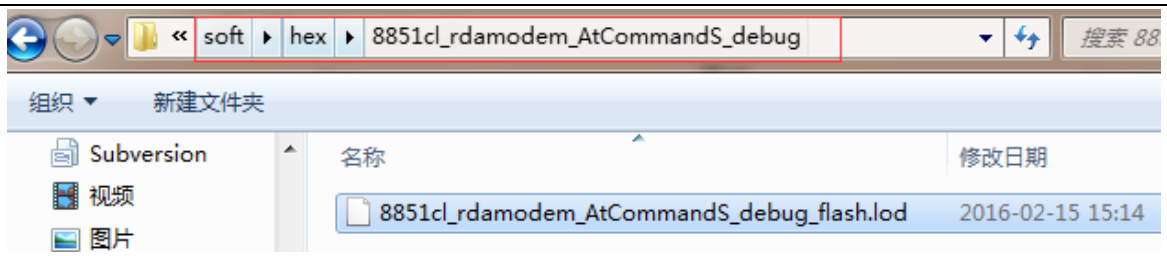
```
Loading plugin romburn...
Opening Chip on COM82...
Connection $CURRENTCONNECTION (COM82) opened (#CURRENTCONNECTION).
Connection RegWatch Connection (COM82) opened.
Connection Event Sniffer (COM82) opened.
Connection BufWatch Connection (COM82) opened.
[COM OPEN OK]
Event sniffer (re)starting. (Connection: Event Sniffer (COM82))
Loading profile script : 8809p.rb
```

（3） Uncompress ramrun into C:\CSDTK, Choose the file "host_8809_flsh_spi32m_ramrun.lod" in C:\CSDTK\ramrun. You just need to do this at the first time you run the application.

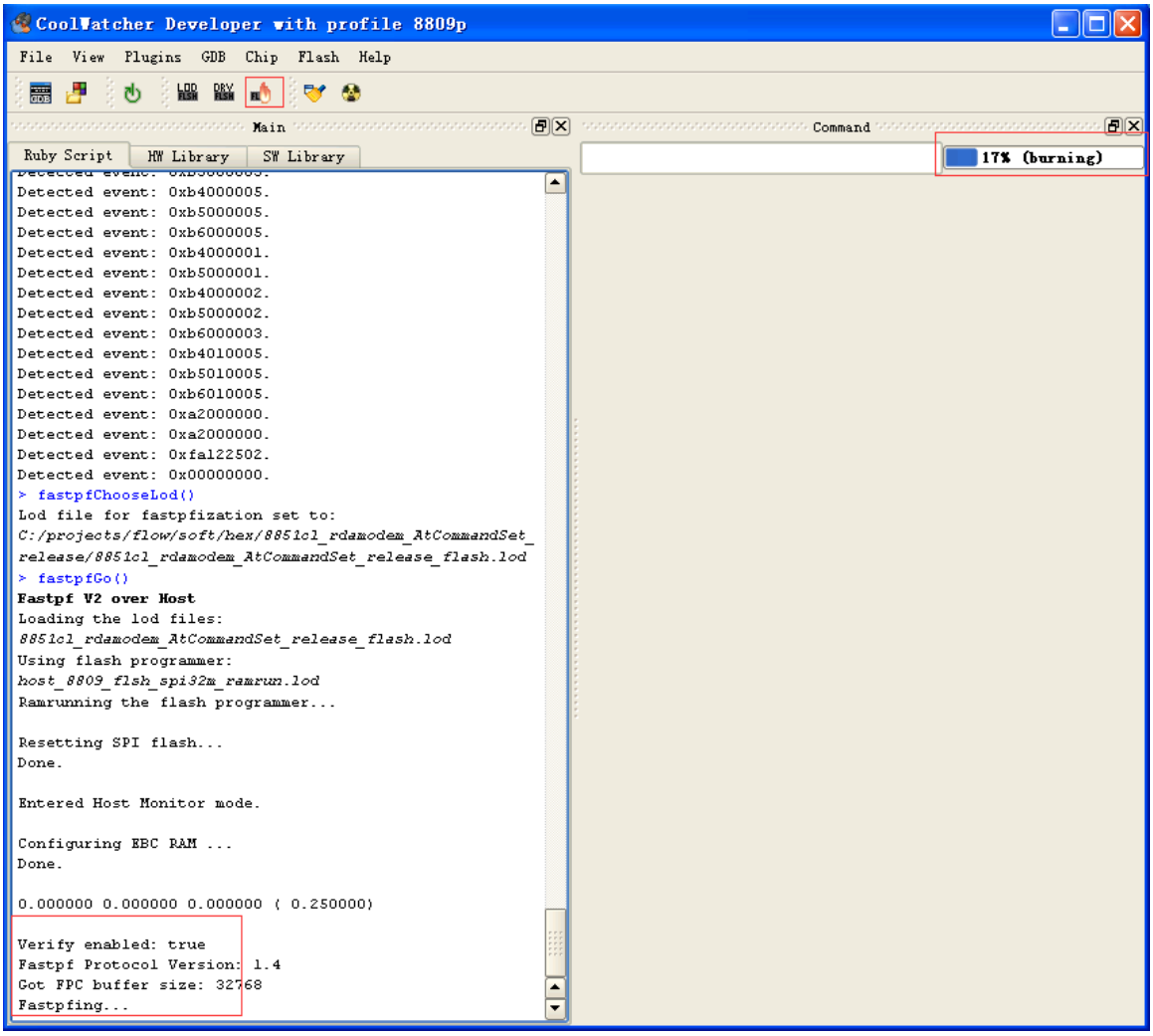


（4） Choose lod file.





（5） Download program



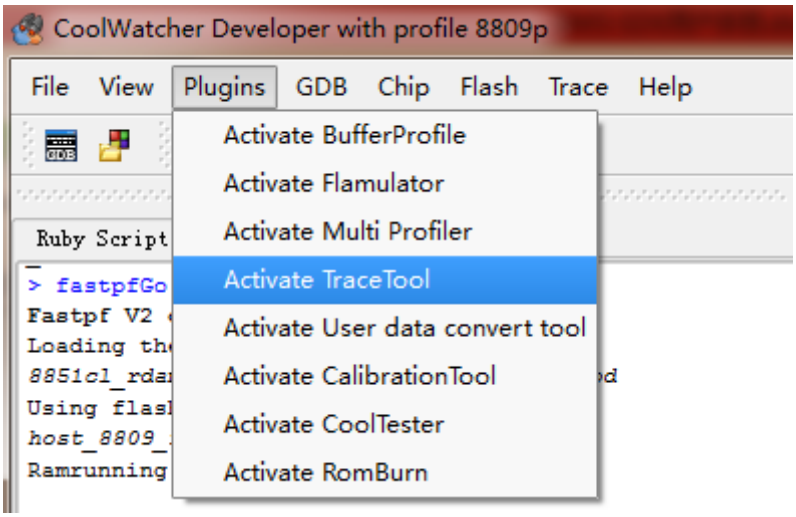
4.2. Debug

User can add the trace to print statements in the program, for example print a usable variable value in somewhere of the program

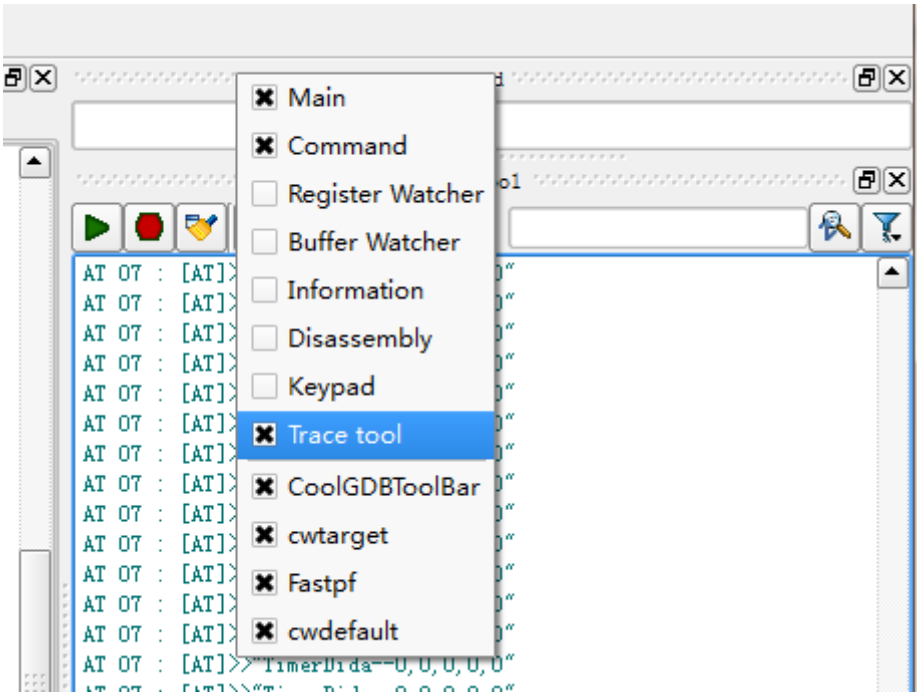
AT_TC(USR_TRACE,"data=%d",num);

After program compiled into the debug version downloaded to the module, we can use tools to GDB:

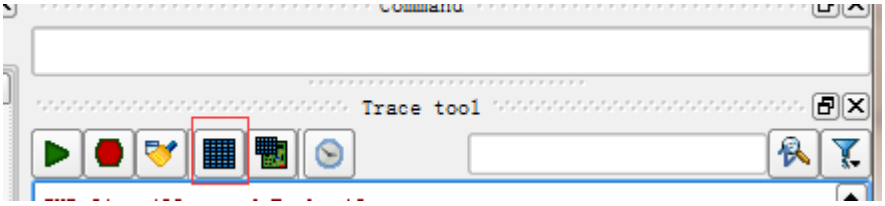
（1）First of all, run the Trace module:



（2）Right click an empty area on the desktop, then choose the Trace Tool:



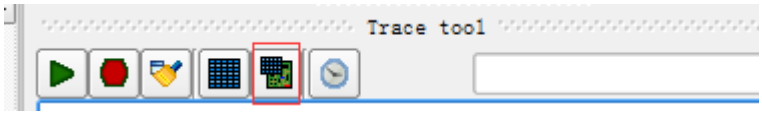
（3）Click on the select set of Trace level:



（4）The level that we trace the print is 7, select 7, and then click apply:

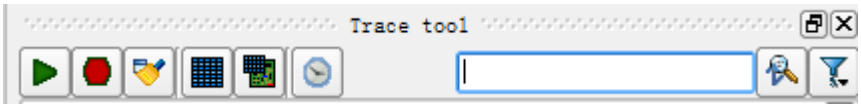
SND	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	A11	None
API	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	A11	None
MMI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	A11	None
SIM	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	A11	None
AT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	A11	None
M2A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	A11	None

（6） Click the following button :

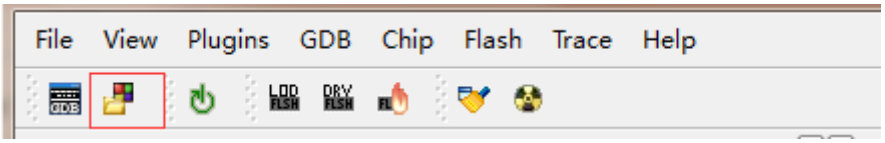


（7） If there is no trace information, click the first triangle number to start running.

（8） Search the Trace information in the input field :

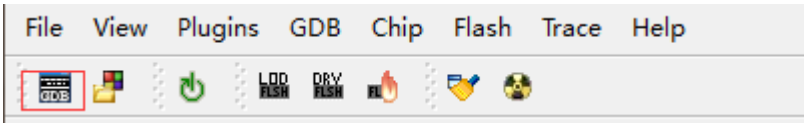


（9） If the program run into a death situation, you can conduct GDB debugging;loading the file is necessary for the first run, Click the following button,



Find “C:\CSDTK\cooltools\rbbase\common\plugins\CoolGDB\profiles\gallite_xcpu.cgd”file.

（10） Then click the following button to open the GDB debugging window,locate to the location of the crash.



5. API

5.1. Introduction

Users find folder “usr” in folder “Soft\At”. This directory is open to users for programming, where Include is the location of header files and Src is the location of source code.In the directory we have defined a C file, where there is test demo.

Usr_AppMain function can be the main function.System will call this function after the operation. USR_WAIT_TASK_INIT function is waiting for registering network;only after network registration succeeds,the system will keep on.Because there is infinite loop at the end, the content of this function will be called only one time.

All the usable interface files are listed in Usr_gm3_app. h file of Include directory.User can switch to view the files.

5.2. API LIST

5.2.1. UART Function

5.2.1.1. Usr_OpenUart

Function: Open uart

BOOL Usr_OpenUart(UART_ID id,UART_PARAM *uartCfg);

Parameter:

UART_ID Can only use UART_1

UART_PARAM

rate

HAL_UART_BAUD_RATE_2400
HAL_UART_BAUD_RATE_4800
HAL_UART_BAUD_RATE_9600
HAL_UART_BAUD_RATE_14400
HAL_UART_BAUD_RATE_19200
HAL_UART_BAUD_RATE_28800
HAL_UART_BAUD_RATE_33600
HAL_UART_BAUD_RATE_38400
HAL_UART_BAUD_RATE_57600
HAL_UART_BAUD_RATE_115200
HAL_UART_BAUD_RATE_230400
HAL_UART_BAUD_RATE_460800
HAL_UART_BAUD_RATE_921600

parity
HAL_UART_NO_PARITY
HAL_UART_ODD_PARITY
HAL_UART_EVEN_PARITY
data
HAL_UART_7_DATA_BITS
HAL_UART_8_DATA_BITS
stop
HAL_UART_1_STOP_BIT
HAL_UART_2_STOP_BITS
flow
HAL_FLOW_NONE
HAL_FLOW_RTS_CTS
HAL_FLOW_RS485

Return:
TRUE or FALSE
Version Required : V1.0

5.2.1.2. Usr_CloseUart

Function:Close Uart
BOOL Usr_CloseUart(UART_ID id);
Parameter:UART_ID Can only use UART_1
Return:
TRUE or FALSE
Version Required: V1.0

5.2.1.3. Usr_SendUartData

Function: Use serial port to send data
U16 Usr_SendUartData (UART_ID id,UINT8 *data,U16 len)
Parameter:UART_ID
*data to send data pointer
len to send data length
Return:
Send data length or 0
Version Required: V1.0

5.2.1.4. Usr_RegisterUartRxCallback

Function: Register uart,receive data and callback function

VOID Usr_RegisterUartRxCallback(UART_ID id,UART_CALLBACK_FUNC func)

Parameter:UART_ID

func callback function

Example:

a function that will process serial data:

Usr_Uart_Get (U8 * data,U16 len)

The registration method is:

Usr_RegisterUartRxCallback (UART_1 Usr_Uart_Get);

Data received by serial will switch to Usr_Uart_Get method for processing.

Return:

No Return

Version Required: V1.0

5.2.2. Socket Function

5.2.2.1. Usr_OpenSock

Function:Open Socket

VOID Usr_OpenSock(SOCK_ID id)

Parameter:

SOCK _ID Soket ID

SOCK1

SOCK2

Return:

No Return

Version Required: V1.0

5.2.2.2. Usr_CloseSock

Function:Close Socket

VOID Usr_CloseSock(SOCK_ID id)

Parameter:

SOCK _ID Soket ID

SOCK1

SOCK2

Return:

No Return

Version Required: V1.0

5.2.2.3. Usr_SetApn

Function:Set APN

BOOL Usr_SetApn(SYS_PARAM_APN *apn)

Parameter:

SYS_PARAM_APN APN

apn[50]

user[50]

psw[50]

Return:

TRUE or FALSE

Version Required: V1.0

5.2.2.4. Usr_SetSockParam

Function: Set the network parameters

BOOL Usr_SetSockParam(SOCK_ID id,GM3_SOCK_PARAM *param)

Parameter:

SOCK_ID Socket ID

GM3_SOCK_PARAM

sock_type

TCP_TYPE

UDP_TYPE

link_type

SHORT_LINK

LONG_LINK

addr[100]

server_port

Return:

TRUE or FALSE

Version Required: V1.0

5.2.2.5. Usr_SendSockData

Function: To send data by Socket

INT32 Usr_SendSockData(SOCK_ID id,U8 *data,U16 len)

Parameter:

SOCK_ID	Socket ID
*data	To send data pointer
len	To send data length

Return:

Send length or -1

Version Required: V1.0

5.2.2.6. Usr_RegisterSockRxCallback

Function: Register sockets,receive data and callback function

VOID Usr_RegisterSockRxCallback(SOCK_ID id,SOCK_CALLBACK_FUNC func)

Parameter:

SOCK_ID	Socket ID
func	Handle data receiving and function name

Return:

No Return

Version Required: V1.0

5.2.2.7. Usr_GetSockStatus

Function: Get Socket Status

BOOL Usr_GetSockStatus(SOCK_ID id,U8 *state)

Parameter:

SOCK_ID	Socket ID
*state	saving state pointers

Instructions:

AT_TCPIP_STATE_IPINITIAL	0
AT_TCPIP_STATE_IPSTART	1
AT_TCPIP_STATE_IPCONFIG	2
AT_TCPIP_STATE_IND	3
AT_TCPIP_STATE_GPRSACT	4
AT_TCPIP_STATE_IPSTATUS	5
AT_TCPIP_STATE_CONNECTING	6

AT_TCPIP_STATE_CLOSE 7
AT_TCPIP_STATE_CONNECTOK 8
AT_TCPIP_STATE_SOCKETOK 9
AT_TCPIP_STATE_UDPOK 10

Return:

TRUE or FALSE

Version Required: V1.0

5.2.3. SMS

5.2.3.1. Usr_SendSmsData

Function:Send sms function

U16 Usr_SendSmsData(UINT8 *phone,SMS_SEND_TYPE type,UINT8 *data,U16 len)

Parameter:

*phone SMS receiving phone number
type SMS sending type
ASCLL
GSM
UCS2
*data send data pointer
len send data length

Return:

Send length or 0

Version Required: V1.0

5.2.3.2. Usr_SetSmsDest

Function: Set SMS transmission phone number

BOOL Usr_SetSmsDest(char *phone,U8 len)

Parameter:

*phone Phone number pointer
len Phone number lenght

Return:

Send length or 0

Version Required: V1.0

5.2.3.3. Usr_RegisterSmsRxCallback

Function: Register SMS,receive and callback function

VOID Usr_RegisterSmsRxCallback(SMS_CALLBACK_FUNC func)

Parameter:

func receive SMS and process function name

Return:

No Return

Version Required: V1.0

5.2.4. Timer Function

5.2.4.1. Usr_StartTimer

Function:Start Timer

VOID Usr_StartTimer(TIMER_ID id)

Parameter:

TIMER_ID The timer ID, five way timer available

TIMER1

TIMER2

TIMER3

TIMER4

TIMER5

Return:

No Return

Version Required: V1.0

5.2.4.2. Usr_StopTimer

Function:Stop Timer

VOID Usr_StopTimer(TIMER_ID id)

Parameter:

TIMER_ID The timer ID, five way timer available

TIMER1

TIMER2

TIMER3

TIMER4
TIMER5

Return:

No Return

Version Required: V1.0

5.2.4.3. Usr_SetTimer

Function:Set Timer

VOID Usr_SetTimer(TIMER_ID id,U32 period,TIMER_CALLBACK_FUNC func)

Parameter:

TIMER_ID
TIMER1
TIMER2
TIMER3
TIMER4
TIMER5

period The timer triggering time, unit of seconds

func The function when the timer is triggered, recommend not to increase the time delay operation

Return:

No Return

Version Required: V1.0

5.2.5. System AT Command

5.2.5.1. Usr_SendAtCmd

Function: execute system AT command

BOOL Usr_SendAtCmd(U8 *data,U16 len)

Parameter:

*data The AT command, pay attention to conform to the AT command format,followed by "\ r \ n"

len The AT command length

Example:Usr_SendAtCmd("AT+VER?\r\n",strlen("AT+VER?\r\n"));

Return:

TRUE or FALSE

Version Required: V1.0

5.2.5.2. Usr_RegisterCmdRxCallback

Function: Register the AT command and process callback function

VOID Usr_RegisterCmdRxCallback(CMD_CALLBACK_FUNC func)

Parameter:

func The AT command processing function name

Return:

No Return

Version Required: V1.0

5.2.6. Custom AT Command

We have made the framework of the AT command processing, users can add the AT command as required.

（1） Find"usr_gm3_cmd_table.h"
{(UINT8*)"+TEST", AT_GC_USR_CmdFunc_TEST, SA_CMDCLS_TCPIP, 1, AT_IPR_PERM},
The code says adding an AT command called TEST, call AT_GC_USR_CmdFunc_TEST when receives the AT command Function.The function is in usr_gm3_sdk. C.

（2） We have made the search and setting framework in AT_GC_USR_CmdFunc_TEST;user can perfect it according to the comments in the sample code.

5.2.7. System Information

5.2.7.1. Usr_SetEcho

Function:Set Echo

VOID Usr_SetEcho(BOOL b)

Parameter:

TRUE

FALSE

Return:

No Return

Version Required: V1.0

5.2.7.2. Usr_Restart

Function: Restart the module

VOID Usr_Restart()

Parameter:

Return:

No Return

Version Required: V1.0

5.2.7.3. Usr_ReloadUserDefault

Function: Reload User Default

VOID Usr_ReloadUserDefault()

Parameter:

Return:

No Return

Version Required: V1.0

5.2.7.4. Usr_ReloadUsrFactory

Function: Reload Default Factory Settings

VOID Usr_ReloadUsrFactory()

Parameter:

Return:

No Return

Version Required: V1.0

5.2.7.5. Usr_SaveCurrentSetting

Function: Save the current parameters

VOID Usr_SaveCurrentSetting ()

Parameter:

Return:

No Return

Version Required: V1.0

5.2.7.6. Usr_SaveAsUserDefault

Function:Save the current settings as the user default settings

VOID Usr_SaveAsUserDefault ()

Parameter:

Return:

No Return

Version Required: V1.0

5.2.7.7. Usr_EnableRFC

Function:Set RFC2217

VOID Usr_EnableRFC (BOOL b)

Parameter:

TRUE

FALSE

Return:

No Return

Version Required: V1.0

5.2.7.8. Usr_EnableUartCMD

Function:Set uart AT command

VOID Usr_EnableUartCMD (BOOL b)

Parameter:

TRUE

FALSE

Return:

No Return

Version Required: V1.0

5.2.7.9. Usr_EnableNetCMD

Function:Set net AT CMD

VOID Usr_EnableNetCMD (BOOL b)

Parameter:

TRUE
FALSE

Return:

No Return

Version Required: V1.0

5.2.7.10. Usr_GetIMEI

Function:Get IMEI

VOID Usr_GetIMEI(U8 *imei)

Parameter:

*imei

Return:

No Return

Version Required: V1.0

5.2.8.GPIO

GM3 provide 6 GPIO and 4 road lamp control;users can choose some for control.

5.2.8.1. GPIO Define

GPIO Define	HAL_APO_ID_T	PIN	Original function
PIN_GPIO_1	g_gm3.gpio_1	30	UART2_RTS
PIN_GPIO_2	g_gm3.gpio_2	8	I2C_SCL
PIN_GPIO_3	g_gm3.gpio_3	31	UART2_CTS
PIN_GPIO_4	g_gm3.gpio_4	37	UART1_CTS
PIN_GPIO_5	g_gm3.gpio_5	7	485_EN
PIN_GPIO_6	g_gm3.gpio_6	26	SIM_INTN
PIN_GPIO_7	g_gm3.gpio_7	10	GPRS_ LED
PIN_GPIO_8	g_gm3.gpio_8	11	LINKA_ LED
PIN_GPIO_9	g_gm3.gpio_9	12	LINKB _LED
PIN_GPIO_10	g_gm3.gpio_10	13	DATA _LED

5.2.8.2. Usr_GpioInit

Function: Initialize GPIO

VOID Usr_GpioInit(HAL_APO_ID_T id,U8 mode)

Parameter:

id refer to HAL_APO_ID_T Struct
mode set GPIO mode; 1 for Output, 0 for Intput

Return:

No Return

Version Required: V1.0

5.2.8.3. Usr_SetGpio

Function: Set GPIO Mode

VOID Usr_SetGpio(HAL_APO_ID_T id,U8 mode)

Parameter:

id refer to HAL_APO_ID_T Struct
mode set GPIO mode; 1 for high level, 0 for low level

Return:

No Return

Version Required: V1.0

5.2.8.4. Usr_GetGpio

Function:Get GPIO Mode

VOID Usr_GetGpio(HAL_APO_ID_T id,U8 mode)

Parameter:

id refer to HAL_APO_ID_T Struct

Return:

1 or 0 (1 for high level, 0 for low level)

Version Required: V1.0

5.2.9.RTC Clock

GM3 module has built-in RTC clock.The module can realize electric timing through external RTC power supply.The external pin PIN16, called VBAT_RTC, can use 2.8 V power supply.

5.2.9.1. Usr_GetSysTime

Function:Get System Clock

BOOL Usr_GetSysTime(HAL_TIM_RTC_TIME_T* rtcTime)

Parameter:

rtcTime HAL_TIM_RTC_TIME_T Struct

UINT8 sec; Second

UINT8 min; Minute

UINT8 hour; Hour

UINT8 day; Day

UINT8 month; Month

UINT8 year; Year

UINT8 wDay; Week Day

Return:

TRUE or FALSE

Version Required: V1.0

5.2.9.2. Usr_SetSysTime

Function:Set System Clock

BOOL Usr_SetSysTime(HAL_TIM_RTC_TIME_T* rtcTime)

Parameter:

rtcTime HAL_TIM_RTC_TIME_T Struct

UINT8 sec; Second

UINT8 min; Minute

UINT8 hour; Hour

UINT8 day; Day

UINT8 month; Month

UINT8 year; Year

UINT8 wDay; Week Day

Return:

TRUE or FALSE

Version Required: V1.0

6. Contact

Company: Jinan USR IOT Technology Limited
Address: Floor 11,Building1,No.1166 Xinluo Street,Gaoxin Distric,Jinan,Shandong,250101 China
Tel: 86-531-55507297, 86-531-88826739
Web: http://www.usriot.com
Support : http://h.usriot.com
Email: sales@usr.cn

7. Update History

2016-02-15 V1.0 Created