Cyclone IV FPGA Core Board AC415

System on Module

ALINX

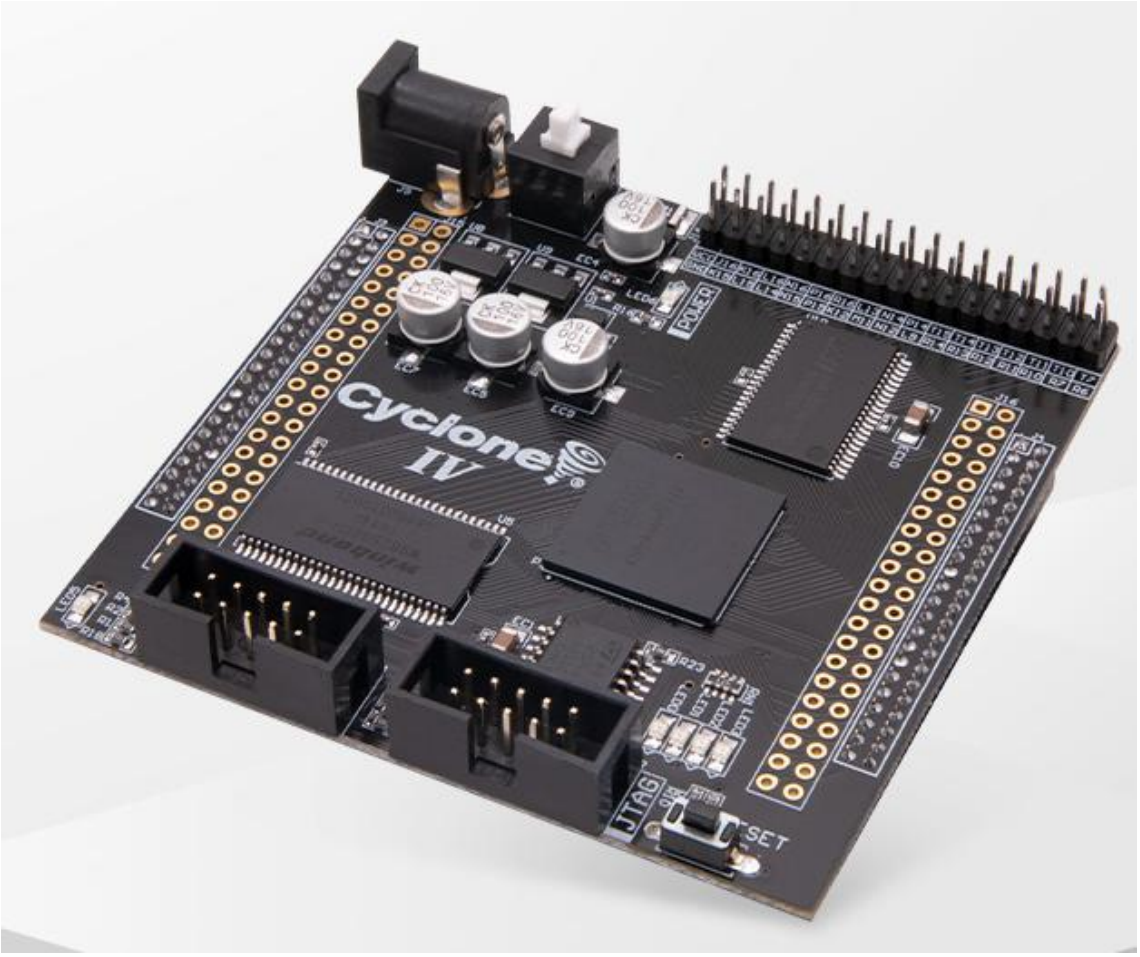
Version Record

Version	Date	Release By	Description
Rev 1.0	2020-11-20	Rachel Zhou	First Release

Table of Contents

Version Record.....2
Part 1: AC415 Core Board.....4
Part 2: Documents Introduction.....4
Part 3: Inspection.....5
Part 4: Software Installation.....7
Part 5 : Install USB download cable driver.....9
Part 6: Verilog program download test.....9
Part 7: NIOS program download test.....15
Part 8: Experimental Demos introduction.....16
Part 9: TCL script file.....16

Part 1: AC415 Core Board



Part 2: Documents Introduction

The documents come with this board saved in the dropbox. For the link and password, contact email: rachel.zhou@alinx.com to get it.

The documents list as below:

Content	Content/File	Description
01_demo_document	01_DEMO_N	NIOS Programs for Testing AC415 Core board
	02_DEMO_V	Verilog Programs forTesting AC415 Core board
	03_NIOS	NIOS Program Demos for Testing AC415 Core board
	04_VERILOG	Verilog Program Demos for Testing AC415 Core board

04_Schematic_PCB_Size Dimension	Core board interface structure diagram	Core board AC415 interface structure diagram
	PCB	Core board AC415 PCB
	Schematic	Core board AC415 Schematic
	Schematic Library	Core board AC415 Schematic Library
05_Chip Manual	/	The chip datasheet used in the FPGA Development Board
06_Factory recovery program	/	/
07_Software	Quartus_Nios_Modelsim_Software	Quartus, Nios, Modelsim Software
	Others	/
08_TCL	/	TCL script files can be opened by software such as Notepad
09_SD Card	/	SD Card Read BMP Picture Display Experiment
AC415 User Manual	/	

Part 3: Inspection

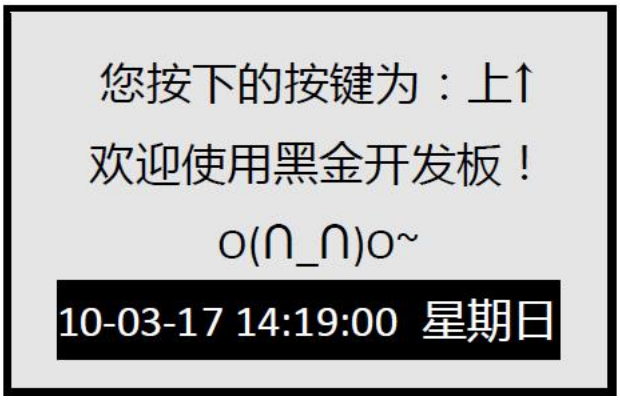
Power-on detection

Before leaving the factory, it has undergone multiple rounds of rigorous testing, and a DEMO program is downloaded to the FPGA (the downloaded program is the NIOS II test program in the DEMO_N folder).

The 5V power adaptor come with the AC415 Core board Kit.

After power on, the development board will run automatically, and you can observe the following phenomena:

- The power indicator (LED6) is always on
- The digital tube starts counting from 000000 and adds 1 to accumulate
- LED for water lamp operation
- The serial port indicator TXD keeps flashing
- The two indicators of the network port flash simultaneously
- The LCD screen has the following content display



The first line is the key test program display, the default display is up ↑, when you press other keys, the LCD screen will display the corresponding direction.

The fourth line is the real-time clock display, the default is **10-03-17**

14:19:00 星期日（Sunday）. After power on again, the time will be displayed by default, because this FPGA development board does not have a backup battery for the real-time clock. The battery model is CR1220.

After the above phenomenon occurs, the main component inspection is completed. The remaining unused parts will be verified in subsequent tests.

Part 4: Software Installation

Install Software.

The software stored in the below

\AC415_EN\07_Software\Quartus_Nios_Modelsim_Software

The ALTERA FPGA development environment is Quartus II. The version we are using is 11.0sp1. If you want to develop NIOS II, you also need to install NIOS II IDE. Its version must be consistent with the Quartus II version. This is necessary, otherwise during the compilation , NIOS II IDE will generate an error.

Note: Copy the software to a directory without Chinese characters and spaces, and then install it according to the following steps

(please note that steps 2 and 3 cannot be exchanged, otherwise an error will occur when the software is running):

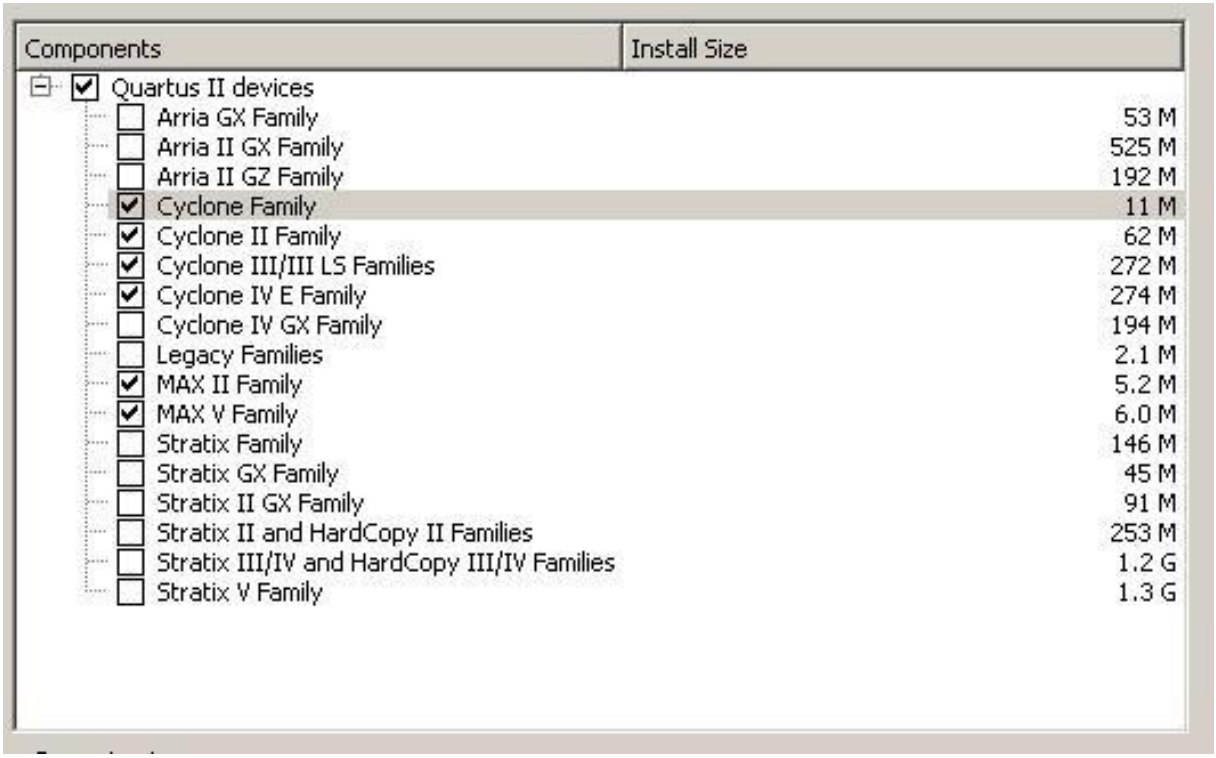
Note: Copy the software to a directory without Chinese characters and spaces, and then install it according to the following steps

(please note that steps 2 and 3 cannot be exchanged, otherwise an error

will occur when the software is running):

1. Install “11.0_quartus_windows.exe”;
2. Install "Setup.exe" in the installation folder 11.0_devices_windows;
3. Install "11.0sp1_quartus_windows.exe";
4. Use "Crack_QII_11.0_SP1_Windows.rar" to crack (32bit system uses x86 crack patch, 64bit system uses x64 crack patch);
5. Install "11.0_legacy_nios2_windows.exe";
6. Install "11.0sp1_legacy_nios2_windows.exe{;
7. Install modelsim-win32-10.0c-se.exe or modelsim-win64-10.0c-se.exe (according to the system decision, 32-bit system,
   choose the former, otherwise, choose the latter);
8. Crack according to modelsim_se_10.0c.rar;

When installing 11.0_devices_windows, please note that since we deleted the unused devices in it, some options cannot be checked when selecting, as shown in the following figure:



Part 5 : Install USB download cable driver

USB BLASTER download cable driver

The driver of USB BLASTER is provided by Quartus II software.

If you are a WIN7 system user, if you cannot install it, please turn off the firewall and anti-virus software, and then re-install it.

How to use USB download cable ?

The FPGA core board supports two download methods, one is JTAG and the other is AS. Correspondingly, there are two 10-pin interfaces on the core board, with the AS port and JTAG port on them. One end of the USB download cable is connected to the USB port of the computer, and the other end is connected to the JTAG port or AS port of the core board. We can only use one of these interfaces. The JTAG method is responsible for FPGA online simulation, and the program is downloaded to the FPGA internal SRAM through the USB download line. Since FPGA is based on SRAM technology, the program will be lost after power off. The AS method downloads the program to the configuration chip EPCSX outside the FPGA. EPCSX is essentially a serial FLASH, so the program will not be lost after the power is off. Every time the power is turned on, the program in EPCSX will be automatically loaded into the FPGA, and then start to run.

USB BLASTER itself has two functions: simulation and downloading. Therefore, USB BLASTER is needed in the FPGA development process to achieve both online simulation and program curing functions. The JTAG method is mainly used for online simulation, and the AS method is used to solidify the final program into the external configuration chip EPCSX.

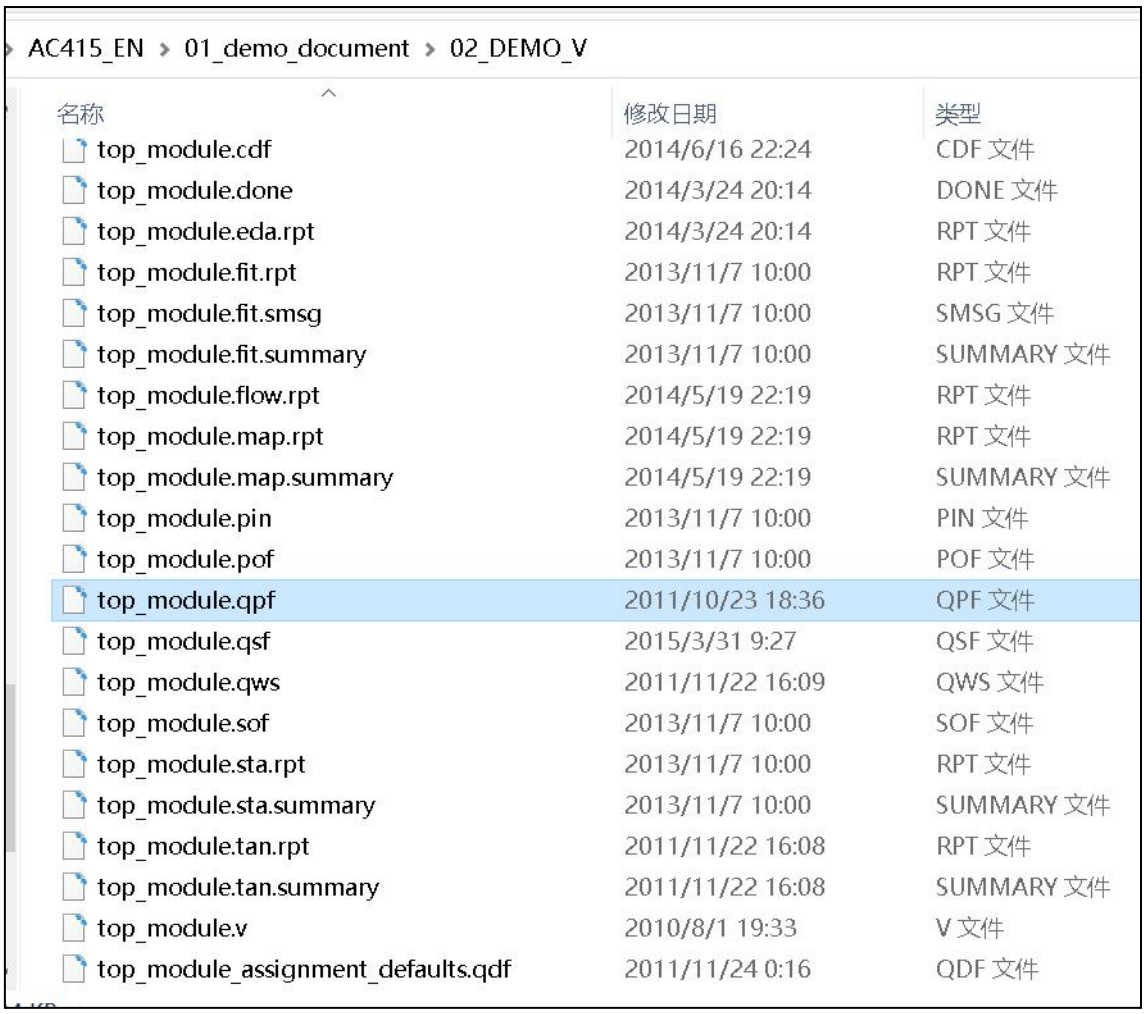
Part 6: Verilog program download test

In the folder ***AC415_EN\01_demo_document\02_DEMO_V,*** there is a

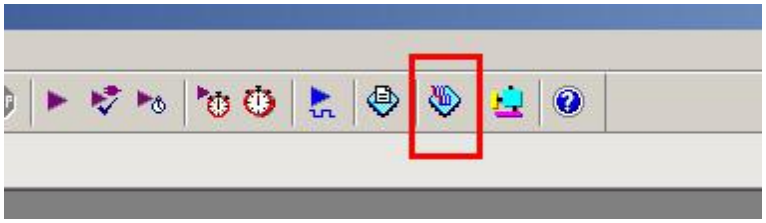
compiled Verilog test program.

Let's show you how to download and test. The test procedure is the water lamp experiment

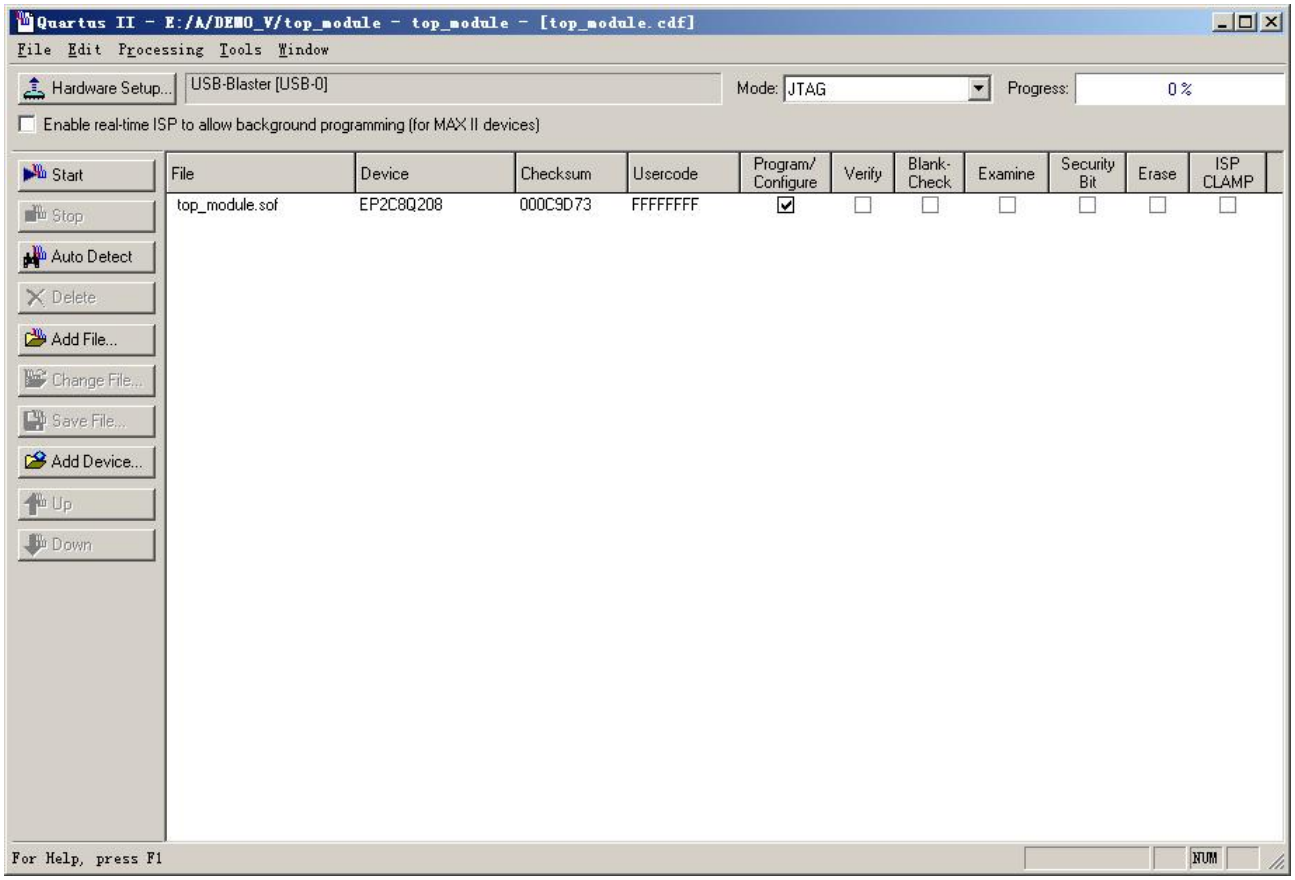
In the ***AC415_EN\01_demo_document\02_DEMO_V*** folder, find the **top_module.qpf** file, double-click it to open the Quartus project file.



After entering the Quartus II software, click on the red circle shown in the figure below to enter the program download interface.



Then, the following dialog box will pop up.

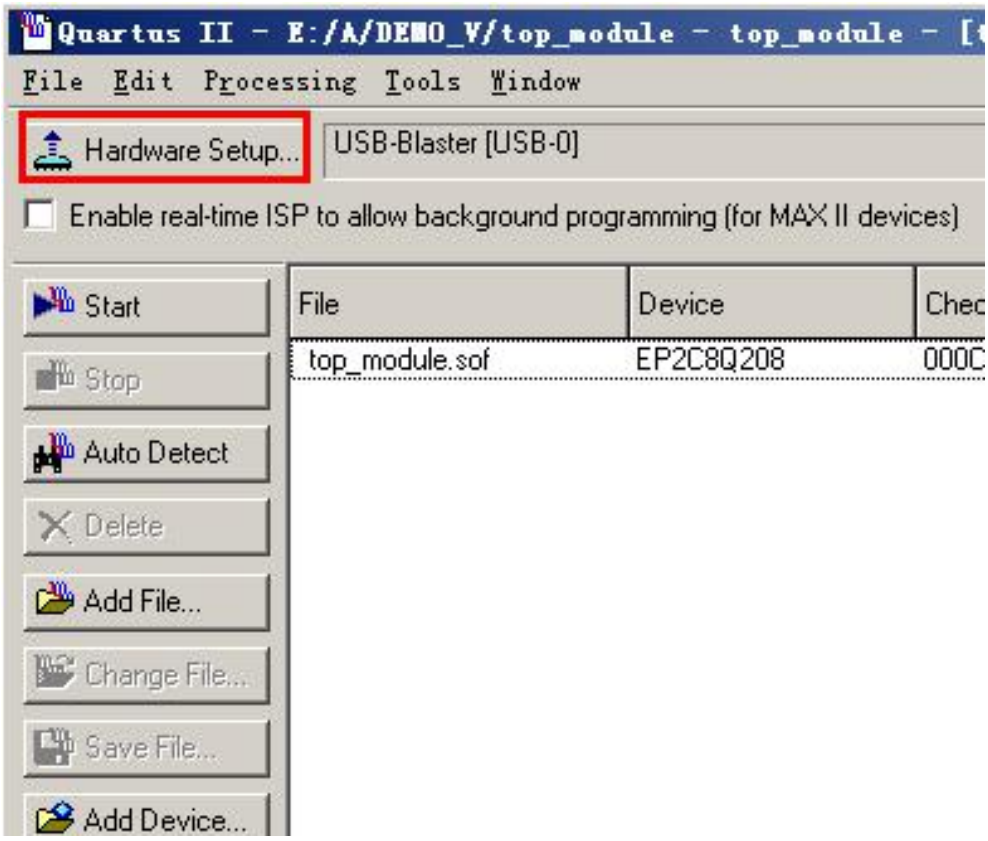


Before downloading, connect the USB download cable to the JTAG port of the FPGA development board. Now what we have to do is online simulation.

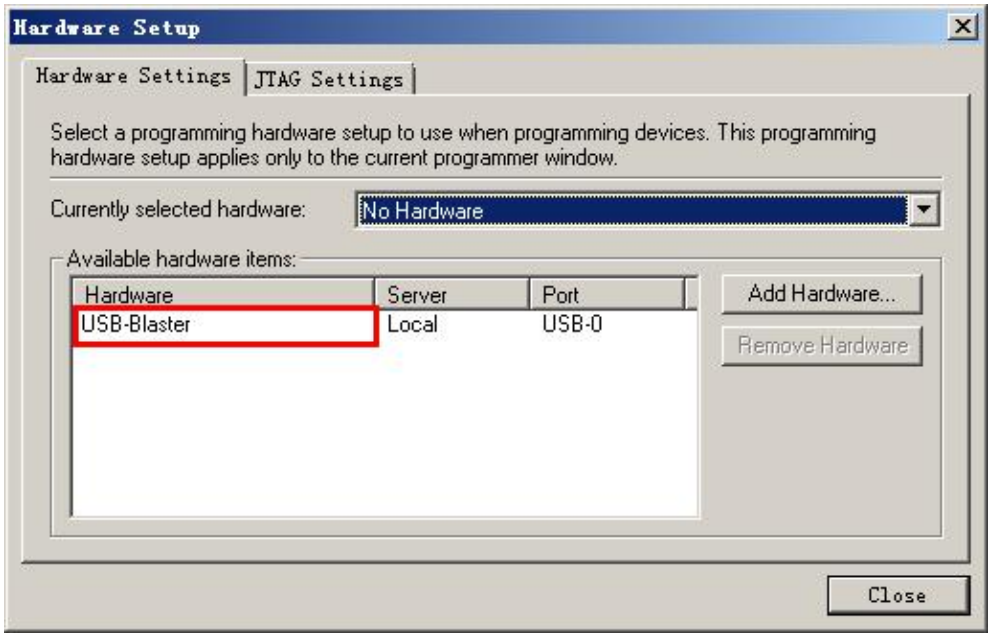
After connecting the USB download cable, plug in the power supply and press the Power switch.

Note: The USB download cable must not be plugged or unplugged when it is powered on, and the power must be turned off before plugging or unplugging. If you plug and unplug the USB download cable with power on, it will burn the JTAG port of the FPGA, so that it can no longer be restored. Without the JTAG port, this FPGA basically cannot be used anymore. The consequences are serious, please remember!

After the hardware connection is completed, the next step is to set up the USB download cable. On the download interface, click Hardware Setup

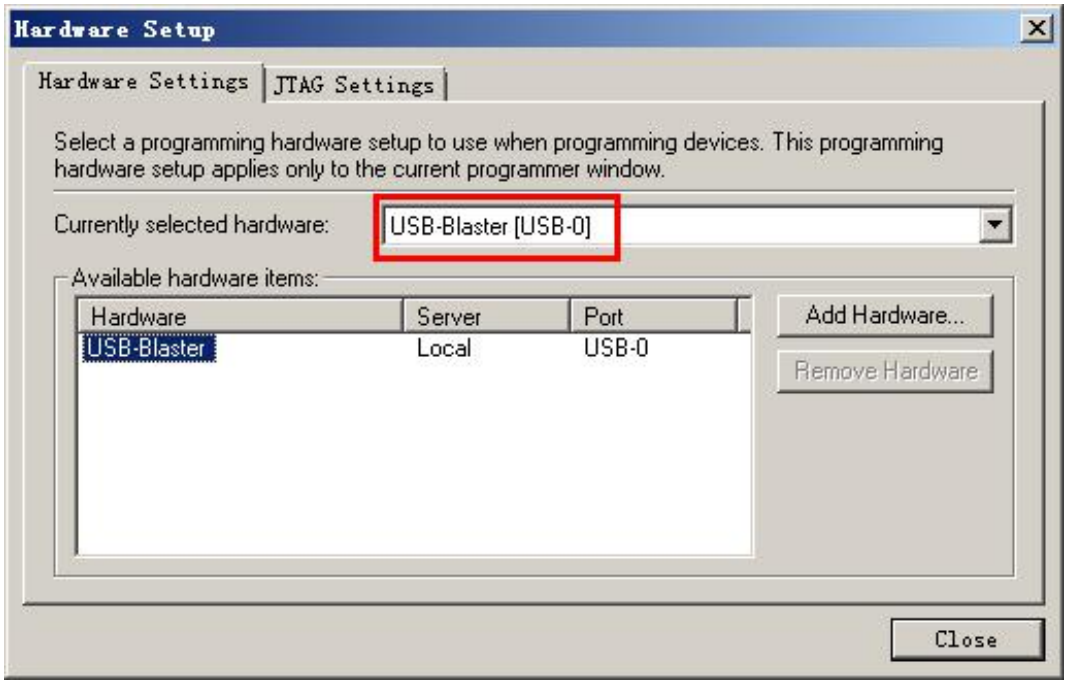


After clicking, the following window will appear

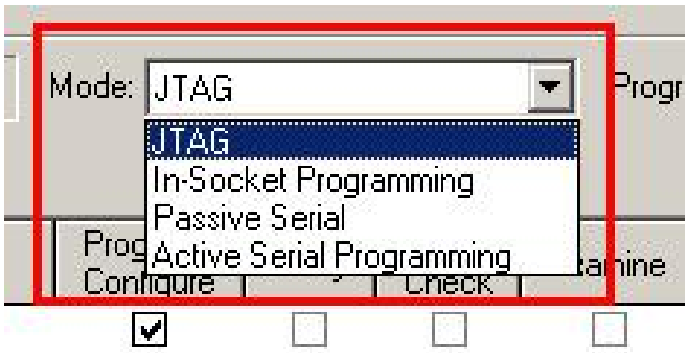


Then, double-click the USB-Blaster marked in the red circle above, and USB-Blaster[USB-0] will appear in the red circle below. If there are other

options, select other options and click Remove Hardware to remove all other options. Finally, click Close to close this window and return to the download window.



Here, we need to pay attention to the option Mode, as shown in the red circle as shown below. There are four options here. The FPGA chip we use supports JTAG and Active Serial Programming (AS). This is the JTAG and AS two download methods I mentioned above.



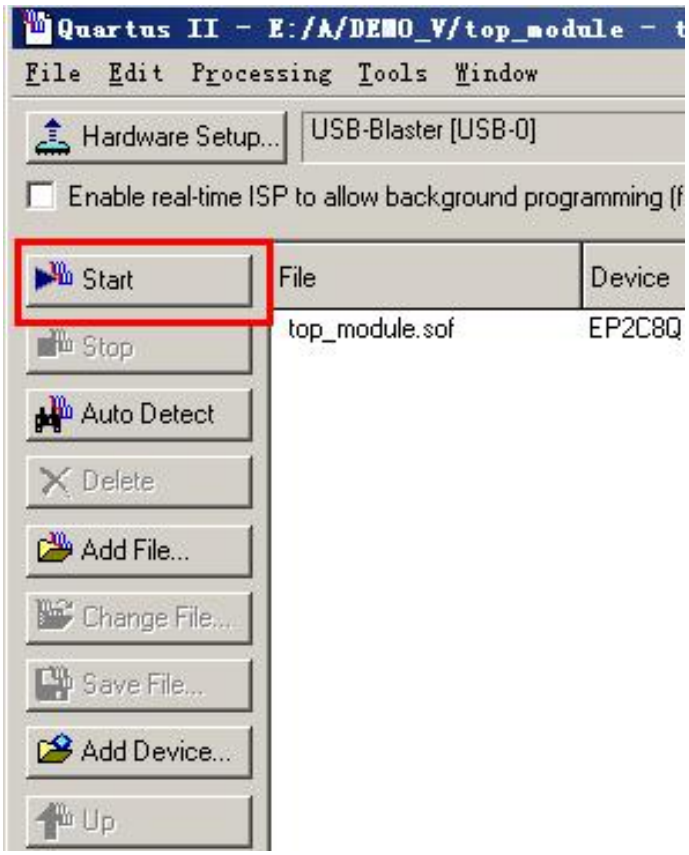
Here, we choose JTAG to implement the JTAG simulation function. If we want to solidify the final program into the external EPCSX, then we choose Active Serial Programming here, and we also need to connect the USB download cable to the AS interface of the FPGA development board.

The files corresponding to JTAG mode and AS mode are different. After

the program is compiled, two downloadable files will be generated. Among them, the file with a suffix of .sof is for downloading in JTAG mode, and the file with a suffix of .pof is for downloading in AS mode.

After entering the download interface, the default .sof file has been loaded, as shown in the top_module.sof file as shown below.

Next, start downloading, click start in the figure below, and the program will start downloading.



After the download is complete, the degree bar will display as 100%, as shown in the figure below

In this way, our program is downloaded. Please observe the four LEDs on the FPGA core board. If successful, a running light experiment will be carried out.

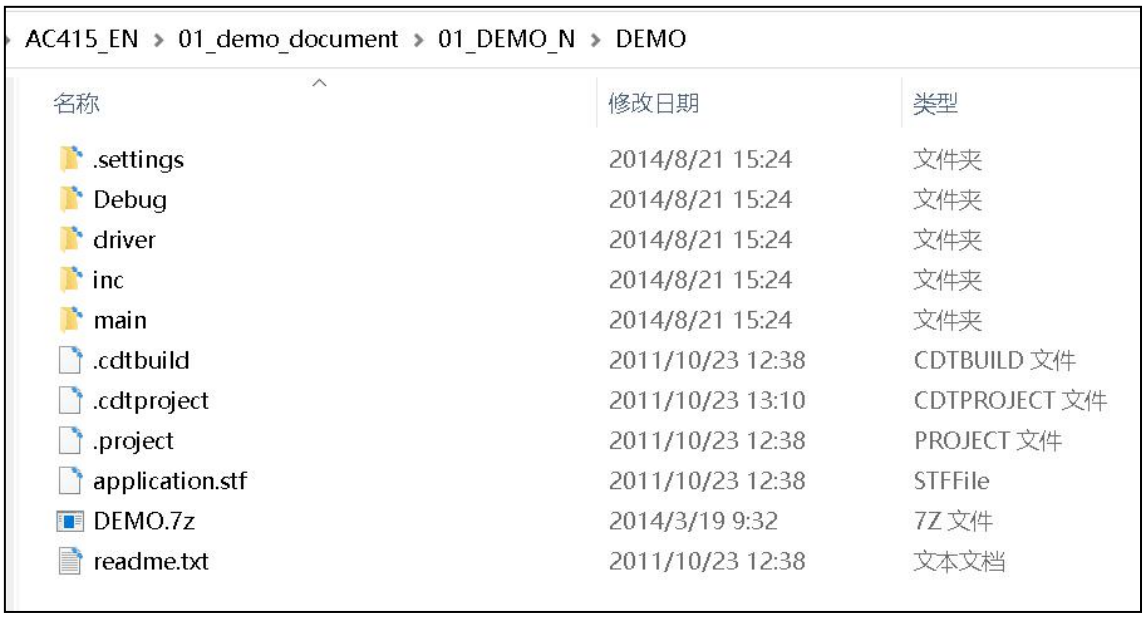
Noted: because the program has been solidified into EPCSX before

leaving the factory. We used the JTAG mode for the above-mentioned test experiments. If you re-power on, you will find that the program is still the previous factory default DEMO program, which may be confused by beginners. The problem is that FPGA is based on SRAM technology. Downloading through JTAG mode only downloads the program to the SRAM inside the FPGA. Therefore, after power off, the program we downloaded through JTAG mode will be lost. After power on again, EPCSX will automatically load the program into the FPGA.

So far, the download test function is complete.

Part 7: NIOS program download test

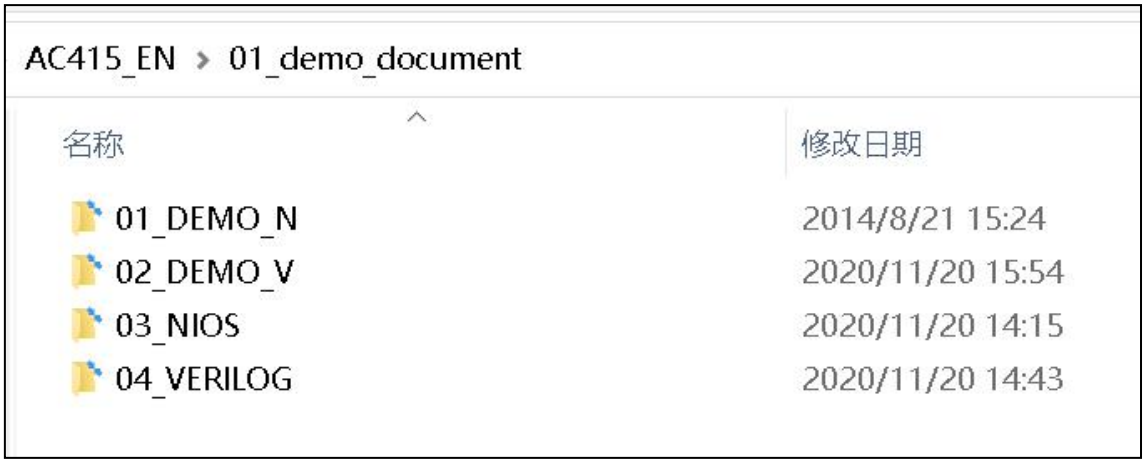
The NIOS Demo programs is under ***AC415_EN\01_demo_document\01_Demo_N\DEMO,*** which includes three folders: main, driver, and inc. The DEMO program of NIOS is also a test program that is programmed into the configuration chip at the factory



To test the NIOS program, you need to re-create the NIOS software project, and then add the above three folders to the NIOS project

Part 8: Experimental Demos introduction

The supporting Demos is divided into two parts, NIOS and Verilog.



- **NIOS**

Saved in the ***AC415_EN\01_demo_document\03_NIOS***

To learn NIOS technology, you must have the foundation of C language, which is also the most basic

- **Verilog HDL Demos**

Saved in the ***AC415_EN\01_demo_document\04_VERILOG***

Part 9: TCL script file

The TCL script file is used to allocate pins. We can use this file to view the pins of the peripherals, avoiding the trouble of viewing the pins through the schematic diagram. The TCL script file is ***AC415_EN\08_TCL*** . To briefly explain the following script file, let's take the first example, PIN_M1 is the 23rd pin of FPGA. RESET is the corresponding reset pin. The name RESET is named by ourselves. There is no special regulation. As long as the name of the script file is the same as the pin name to be assigned, other pins have the same principle.

There is one point to note in the following pins, SRAM and FLASH pins are shared, so when using FLASH, please comment out the SRAM pins with the #

symbol; for the same reason, when using SRAM, please set the FLASH pins Comment out with the # symbol.

```
#Reset Pin
set_location_assignment        PIN_M1        -to RESET

#Clock Pin
set_location_assignment        PIN_R9        -to CLOCK

#EPCS Pin
set_location_assignment        PIN_H2        -to DATA0
set_location_assignment        PIN_H1        -to DCLK
set_location_assignment        PIN_D2        -to SCE
set_location_assignment        PIN_C1        -to SDO

#SDRAM Pin
set_location_assignment        PIN_A10       -to S_DB[0]
set_location_assignment        PIN_B10       -to S_DB[1]
set_location_assignment        PIN_A11       -to S_DB[2]
set_location_assignment        PIN_B11       -to S_DB[3]
set_location_assignment        PIN_A12       -to S_DB[4]
set_location_assignment        PIN_B12       -to S_DB[5]
set_location_assignment        PIN_A13       -to S_DB[6]
set_location_assignment        PIN_B13       -to S_DB[7]
set_location_assignment        PIN_A2        -to S_DB[8]
set_location_assignment        PIN_B1        -to S_DB[9]
set_location_assignment        PIN_C2        -to S_DB[10]
set_location_assignment        PIN_D1        -to S_DB[11]
set_location_assignment        PIN_F2        -to S_DB[12]
set_location_assignment        PIN_F1        -to S_DB[13]
set_location_assignment        PIN_G2        -to S_DB[14]
set_location_assignment        PIN_G1        -to S_DB[15]

set_location_assignment        PIN_F15       -to S_A[0]
set_location_assignment        PIN_F16       -to S_A[1]
```

```
set_location_assignment PIN_G15 -to S_A[2]
 set_location_assignment PIN_G16 -to S_A[3]
 set_location_assignment PIN_C8 -to S_A[4]
 set_location_assignment PIN_A7 -to S_A[5]
 set_location_assignment PIN_B7 -to S_A[6]
 set_location_assignment PIN_A6 -to S_A[7]
 set_location_assignment PIN_B6 -to S_A[8]
 set_location_assignment PIN_A5 -to S_A[9]
 set_location_assignment PIN_D16 -to S_A[10]
 set_location_assignment PIN_B5 -to S_A[11]
 set_location_assignment PIN_A4 -to S_A[12]

 set_location_assignment PIN_A3 -to S_CLK
set_location_assignment PIN_C16 -to S_BA[0]
set_location_assignment PIN_D15 -to S_BA[1]
set_location_assignment PIN_A15 -to S_NCAS
set_location_assignment PIN_B4 -to S_CKE
set_location_assignment PIN_B16 -to S_NRAS
set_location_assignment PIN_B14 -to S_NWE
set_location_assignment PIN_C15 -to S_NCS
set_location_assignment PIN_B3 -to S_DQM[1]
set_location_assignment PIN_A14 -to S_DQM[0]

#Parallel FLASH pin
#set_location_assignment PIN_P16 -to F_DB[0]
#set_location_assignment PIN_P15 -to F_DB[1]
#set_location_assignment PIN_N16 -to F_DB[2]
#set_location_assignment PIN_N15 -to F_DB[3]
#set_location_assignment PIN_L16 -to F_DB[4]
#set_location_assignment PIN_L14 -to F_DB[5]
#set_location_assignment PIN_K16 -to F_DB[6]
#set_location_assignment PIN_L15 -to F_DB[7]
#set_location_assignment PIN_R16 -to F_A[0]
#set_location_assignment PIN_T7 -to F_A[1]
#set_location_assignment PIN_R6 -to F_A[2]
#set_location_assignment PIN_T10 -to F_A[3]
#set_location_assignment PIN_R7 -to F_A[4]
#set_location_assignment PIN_T11 -to F_A[5]
#set_location_assignment PIN_R10 -to F_A[6]
#set_location_assignment PIN_T12 -to F_A[7]
```

```
#set_location_assignment        PIN_T15         -to F_A[8]
#set_location_assignment        PIN_R14         -to F_A[9]
#set_location_assignment        PIN_P14         -to F_A[10]
#set_location_assignment        PIN_L9          -to F_A[11]
#set_location_assignment        PIN_M11         -to F_A[12]
#set_location_assignment        PIN_L13         -to F_A[13]
#set_location_assignment        PIN_N12         -to F_A[14]
#set_location_assignment        PIN_N14         -to F_A[15]
#set_location_assignment        PIN_K15         -to F_A[16]
#set_location_assignment        PIN_R11         -to F_A[17]
#set_location_assignment        PIN_T13         -to F_A[18]
#set_location_assignment        PIN_R13         -to F_A[19]
#set_location_assignment        PIN_T14         -to F_A[20]
#set_location_assignment        PIN_J16         -to F_ALSB

#set_location_assignment        PIN_R12         -to F_NWE
set_location_assignment         PIN_J13         -to F_NCE
#set_location_assignment        PIN_K12         -to F_NOE

#SRAM pin
set_location_assignment         PIN_L9          -to SRAM_A[0]
set_location_assignment         PIN_P14         -to SRAM_A[1]
set_location_assignment         PIN_N12         -to SRAM_A[2]
set_location_assignment         PIN_N14         -to SRAM_A[3]
set_location_assignment         PIN_M11         -to SRAM_A[4]
set_location_assignment         PIN_L15         -to SRAM_A[5]
set_location_assignment         PIN_K16         -to SRAM_A[6]
set_location_assignment         PIN_K15         -to SRAM_A[7]
set_location_assignment         PIN_J16         -to SRAM_A[8]
set_location_assignment         PIN_J15         -to SRAM_A[9]
set_location_assignment         PIN_T6          -to SRAM_A[10]
set_location_assignment         PIN_R5          -to SRAM_A[11]
set_location_assignment         PIN_T5          -to SRAM_A[12]
set_location_assignment         PIN_R4          -to SRAM_A[13]
set_location_assignment         PIN_R4          -to SRAM_A[14]
set_location_assignment         PIN_T14         -to SRAM_A[15]
set_location_assignment         PIN_R14         -to SRAM_A[16]
set_location_assignment         PIN_T15         -to SRAM_A[17]

set_location_assignment         PIN_L13         -to SRAM_DB[0]
```

```
set_location_assignment PIN_L13 -to SRAM_DB[1]
set_location_assignment PIN_L13 -to SRAM_DB[2]
set_location_assignment PIN_P15 -to SRAM_DB[3]
set_location_assignment PIN_P16 -to SRAM_DB[4]
set_location_assignment PIN_N15 -to SRAM_DB[5]
set_location_assignment PIN_N16 -to SRAM_DB[6]
set_location_assignment PIN_L14 -to SRAM_DB[7]
set_location_assignment PIN_R6 -to SRAM_DB[8]
set_location_assignment PIN_T7 -to SRAM_DB[9]
set_location_assignment PIN_R7 -to SRAM_DB[10]
set_location_assignment PIN_T10 -to SRAM_DB[11]
set_location_assignment PIN_R10 -to SRAM_DB[12]
set_location_assignment PIN_T11 -to SRAM_DB[13]
set_location_assignment PIN_R11 -to SRAM_DB[14]
set_location_assignment PIN_T12 -to SRAM_DB[15]

set_location_assignment PIN_R12 -to SRAM_BYTE[0]
set_location_assignment PIN_T13 -to SRAM_BYTE[1]
set_location_assignment PIN_R13 -to SRAM_OE
set_location_assignment PIN_L16 -to SRAM_WE
set_location_assignment PIN_L11 -to SRAM_CS

#USB corresponding pin
set_location_assignment PIN_E10 -to USB_DB[0]
set_location_assignment PIN_C11 -to USB_DB[1]
set_location_assignment PIN_D9 -to USB_DB[2]
set_location_assignment PIN_C9 -to USB_DB[3]
set_location_assignment PIN_E9 -to USB_DB[4]
set_location_assignment PIN_E9 -to USB_DB[5]
set_location_assignment PIN_F8 -to USB_DB[6]
set_location_assignment PIN_E8 -to USB_DB[7]

set_location_assignment PIN_D12 -to USB_A0
set_location_assignment PIN_D11 -to USB_WR
set_location_assignment PIN_B9 -to USB_NINT
set_location_assignment PIN_E11 -to USB_RD

#Network port corresponding pin
set_location_assignment PIN_E15 -to LAN_NINT
set_location_assignment PIN_E16 -to LAN_NWOL
```

```
set_location_assignment PIN_J14 -to LAN_MOSI
set_location_assignment PIN_J12 -to LAN_MISO
set_location_assignment PIN_J12 -to LAN_SCK
set_location_assignment PIN_L10 -to LAN_CS

#VGA corresponding pin
set_location_assignment PIN_E6 -to VGA_G
set_location_assignment PIN_D5 -to VGA_B
set_location_assignment PIN_C6 -to VGA_R
set_location_assignment PIN_D3 -to VGA_HS
set_location_assignment PIN_C3 -to VGA_VS

#LCD screen corresponding pin
set_location_assignment PIN_L1 -to LCD_CS
set_location_assignment PIN_N1 -to LCD_CS
set_location_assignment PIN_L2 -to LCD_SCL
set_location_assignment PIN_P1 -to LCD_SI

#LED corresponding pin
set_location_assignment PIN_J1 -to LED[0]
set_location_assignment PIN_J2 -to LED[1]
set_location_assignment PIN_K1 -to LED[2]
set_location_assignment PIN_K2 -to LED[3]

#Key corresponding pin
set_location_assignment PIN_R8 -to KEY_OK
set_location_assignment PIN_E1 -to KEY_UP
set_location_assignment PIN_T8 -to KEY_DOWN
set_location_assignment PIN_M2 -to KEY_LEFT
set_location_assignment PIN_T9 -to KEY_RIGHT

#Serial port corresponding pin
set_location_assignment PIN_A8 -to RXD
set_location_assignment PIN_G5 -to RXD

#24LC04（EEPROM）corresponding pin
set_location_assignment PIN_G11 -to I2C_SDA
set_location_assignment PIN_C14 -to I2C_SCL

#PS2 corresponding pin
```

```
set_location_assignment PIN_E7 -to PS2_DAT
set_location_assignment PIN_D8 -to PS2_CLK

#DS1302(Real Time Clock) Corresponding
Pin
set_location_assignment PIN_K9 -to RTC_SCLK
set_location_assignment PIN_F13 -to RTC_nRST
set_location_assignment PIN_D14 -to RTC_DATA

#Buzzer corresponding pin
set_location_assignment PIN_D6 -to BUZZER

#Digital tube corresponding pin
set_location_assignment PIN_M8 -to DIG[0]
set_location_assignment PIN_L7 -to DIG[1]
set_location_assignment PIN_P9 -to DIG[2]
set_location_assignment PIN_N9 -to DIG[3]
set_location_assignment PIN_M9 -to DIG[4]
set_location_assignment PIN_M10 -to DIG[5]
set_location_assignment PIN_P11 -to DIG[6]
set_location_assignment PIN_N11 -to DIG[7]
set_location_assignment PIN_N6 -to SEL[5]
set_location_assignment PIN_P6 -to SEL[4]
set_location_assignment PIN_M6 -to SEL[3]
set_location_assignment PIN_M7 -to SEL[2]
set_location_assignment PIN_P8 -to SEL[1]
set_location_assignment PIN_N8 -to SEL[0]
```