UART ISP Using Python

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	UART ISP Using Python
BSP Version	NUC123 Series BSP CMSIS V3.01.001
Hardware	NuTiny-EVB-NUC123-LQFP64 v1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

1.1 Introduction

This example implements an ISP (In System Programming) program in Python3 on PC side. It needs ISP firmware provided by Nuvoton for ISP operation.

1.2 Principle

ISP is a technique for firmware updating without hardware modification. The target device mentioned in this example is with two banks of flash memory, LDROM and APROM, for program storage. The size of LDROM is smaller and usually for ISP firmware storage. Otherwise, the size of APROM is larger and is for user application. ISP firmware must be downloaded to LDROM before ISP operation. It acts as bridge between PC and target device for firmware updating in APROM.

The example implements ISP command listed in Table 1 for .bin file downloading. The ISP firmware must be downloaded to LDROM and setting the boot section to boot from LDROM before downloading program to APROM. The section 1.3 will detail the steps for boot setting and downloading ISP firmware to LDROM.

Command	Command Code	Description
CMD_UPDATE_APROM	0x000000A0	Within the range 0~MAX_APROM_SIZE, program APROM from address user specified.
CMD_READ_CONFIG	0x000000A2	Get Config0 and Config1.
CMD_SYNC_PACKNO	0x000000A4	Synchronize packet number with ISP device before a valid command send.
CMD_GET_FWVER	0x000000A6	Get version information of ISP firmware.
CMD_GET_DEVICEID	0x000000B1	Get chip product ID.
CMD_CONNECT	0x000000AE	Connect to ISP device.

Table 1: ISP command used by the example code.

The Figure 1 shows the steps 1 ~ 7 of program downloading. The steps 4 ~ 5 for showing user configuration information are optional. The other steps are required.

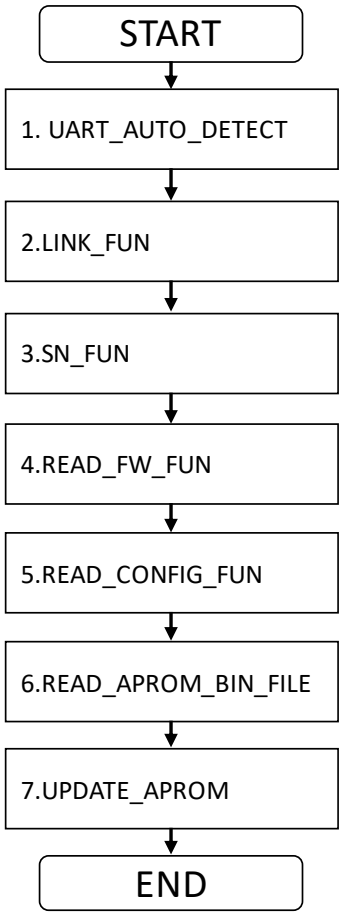


Figure 1: Downloading procedure

1.3 Download ISP firmware through ICP tool

Table 2 lists two required tool for this example.

ICP Tool	Link
ISP Tool	Link
ISP Firmware Source	Included in ISP Tool or BSP. UART pins may differ.

Table 2: Download Link of Tools

The ICP Tool is for downloading ISP firmware to LDROM. After ICP tool opened and connected to target device, please follow the steps shown in Figure 2 to download ISP firmware to LDROM.

1. Specify the location of ISP firmware for LDROM. The binary file for the example target device is “NUC123_ISP_UART1_PB4_PB5.bin” which can be found in the directory in ISP tool. And check the checkbox, “LDROM”, in “Programing” section.
2. Select “LDROM” in “Boot Options” in the section, “Config Bits”. And check the checkbox, “Config”, in “Programing” section.
3. Press the button, “Start”, to download ISP firmware.

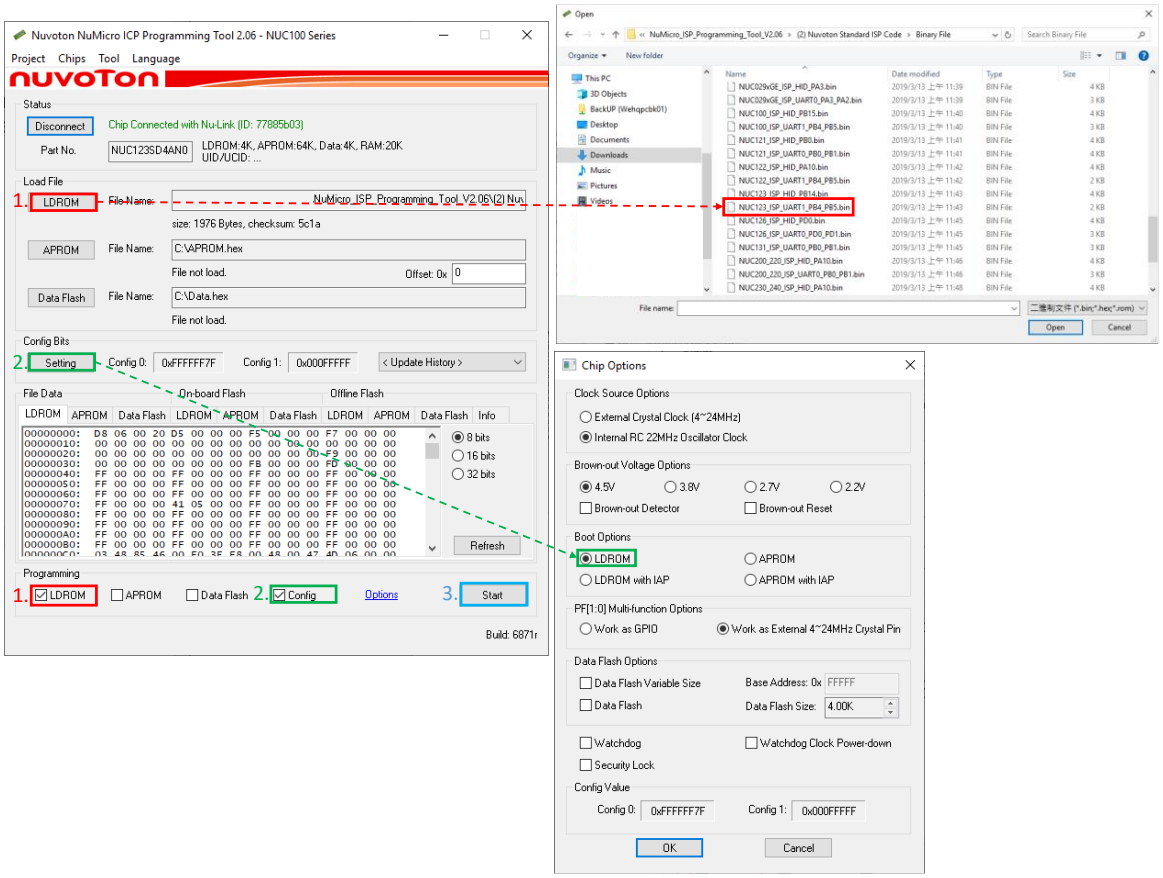


Figure 2: ICP tool setting for downloading ISP firmware to LDROM

1.4 Demo Result

After download ISP firmware to LDROM, we can execute the ISP program by command line window. The baud rate of ISP firmware is 115200bps so the python ISP program uses the same baud rate doing data transferring.

```
python .\py3isp.py [COM port] [.bin file]
```

Figure 3 shows the example that calling the ISP program by above command to specify the serial port and the binary file to download.

Figure 3: call the ISP program and parameter setting.

When the message shown in Figure 4 shown, press the button RESET on evaluation board, and then the download procedure will begin.

Figure 4: Message in ISP program

After “Download complete” message shown(Figure 5), it indicates the binary file has been downloaded to APROM. Press the button RESET, the program in APROM will start.

Figure 5: Download complete

2 Code Description

Function, UART_TRANSFER, implements the data transfer. Each transfer is 64 bytes, the 0 ~ 3 bytes is the ISP command code. The 4 ~ 7 bytes is the packet number for this transfer. When device acknowledge, the packet number will increase 1 for data synchronization. The 8 ~ 63 bytes is the payload.

```
def UART_TRANSFER( thelist, PN):
 #fill the packet number
 thelist[4]=PN&0xff
 thelist[5]=PN>>8&0xff
 thelist[6]=PN>>16&0xff
 thelist[7]=PN>>24&0xff

 com.write(thelist)

 #read ACK packet
 return_str=com.read(64) #return by string
 return_buffer=bytearray(return_str)

 #compute & check checksum
 checksum=0
 for i in range(64):
  checksum=checksum+thelist[i]

 packege_checksum=0
 packege_checksum=return_buffer[0]
 packege_checksum=(return_buffer[1]<<8)|packege_checksum

 if checksum!=packege_checksum:
    print("checksum error")
    error_return()

 #check ACK packet number
 RPN=0
 RPN=return_buffer[4]
 RPN=(return_buffer[5]<<8)|RPN
 RPN=(return_buffer[6]<<16)|RPN
 RPN=(return_buffer[7]<<24)|RPN

 if RPN!=(PN+1):
    print("package number error")
    error_return()

 return return_buffer
```

Function, UPDATE_APROM, is for updating program in APROM. It calls the function, UART_TRANSFER, to send the packet to ISP device after the payload setting is ready.

```
def UPDATE_APROM():
 global AP_FILE
 global PacketNumber
 PacketNumber=PacketNumber+2
 AP_ADDRESS=0;
 AP_SIZE=len(AP_FILE) #get .bin file size
 PAP_COMMNAD = [0 for i in range(64)]
 PAP_COMMNAD[0]=0xa0
```

```
#Set APROM START ADDRESS
PAP_COMMNAD[8]=AP_ADDRESS&0xff
PAP_COMMNAD[9]=AP_ADDRESS>>8&0xff
PAP_COMMNAD[10]=AP_ADDRESS>>16&0xff
PAP_COMMNAD[11]=AP_ADDRESS>>24&0xff

#Set APROM SIZE
PAP_COMMNAD[12]=AP_SIZE&0xff
PAP_COMMNAD[13]=AP_SIZE>>8&0xff
PAP_COMMNAD[14]=AP_SIZE>>16&0xff
PAP_COMMNAD[15]=AP_SIZE>>24&0xff
PAP_COMMNAD[16:64]=AP_FILE[0:48] #first data fragment to copy
UART_TRANSFER(PAP_COMMNAD,PacketNumber) #start first packet transfer

#continue to transfer remaining data fragment
for i in range(48,AP_SIZE,56):
  PacketNumber=PacketNumber+2
  PAP1_COMMNAD = [0 for j in range(64)]
  PAP1_COMMNAD[8:64]=AP_FILE[i:(i+56)]

  #short or last data fragment handling
  if len(PAP1_COMMNAD) < 64:
     for k in range(64-len(PAP1_COMMNAD)):
       PAP1_COMMNAD.append(0xFF)

  if (((AP_SIZE-i)<56) or ((AP_SIZE-i)==56)):
   buf=UART_TRANSFER(PAP1_COMMNAD,PacketNumber)
   d_checksum=buf[8]|buf[9]<<8

   if(d_checksum==(AP_CHECKSUM&0xffff)):
       print("checksum pass")
  else:
   UART_TRANSFER(PAP1_COMMNAD,PacketNumber)
```

3 Software and Hardware Environment

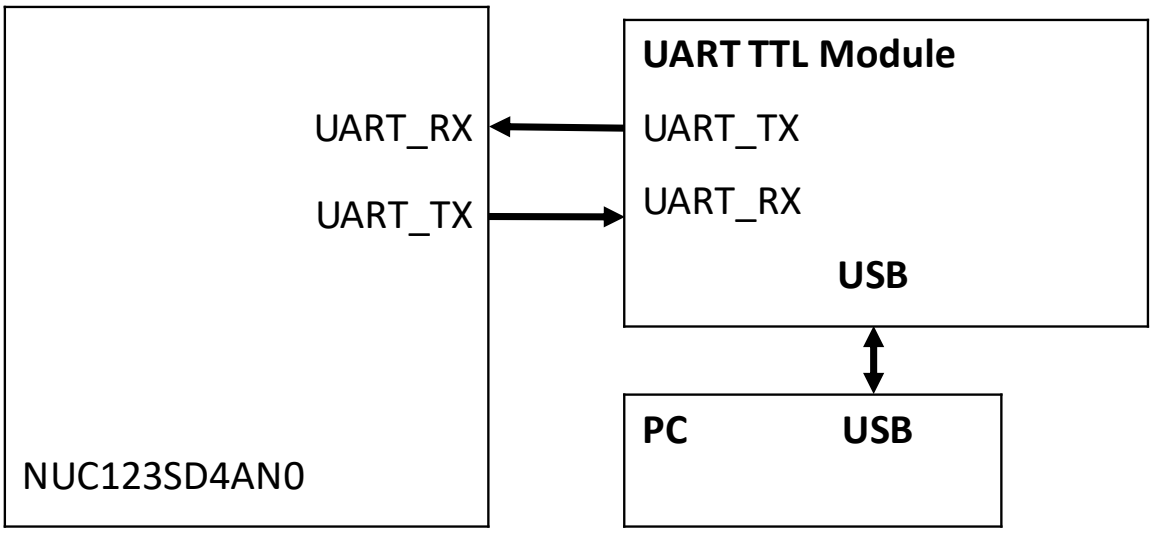
Software Environment

- BSP version
  - NUC123 Series BSP CMSIS V3.01.001
    - Source of ISP Firmware(ISP_UART)
- Python3
  - Pyserial
- ISP/ICP Tool(https://www.nuvoton.com/hq/support/tool-and-software/software/programmer/?__locale=zh_TW)
  - Source code of ISP firmware
  - ICP application on PC

Hardware Environment

- Circuit components
  - NuTiny-EVB-NUC123-LQFP64 v1.0
  - USB-UART TTL(Optional)

- Diagram



4 Directory Information

- EC_NUC123_ISP_UART_Python_V1.00
  - SampleCode
    - ExampleCode Source file of example code

5 How to Execute Example Code

1. Please refer to 1.4.

6 Revision History

Date	Revision	Description
July.01 , 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.