

NX-N947

サンプルプログラム解説

1版 2025年05月16日

1. 概要	2
1.1 概要	2
1.2 接続概要	2
1.3 本サンプルプログラムについて	5
1.4 開発環境について	5
1.5 ワークスペースについて	6
2. サンプルプログラムの構成	7
2.1 フォルダ構成	7
2.2 ファイルの構成	10
3. NX-N947 サンプルプログラム	13
3.1 Debug Console 機能を用いたシリアルターミナルの入出力	13
3.2 動作説明	15
3.2.1 CAN サンプルプログラムの動作説明	15
3.2.2 Ethernet サンプルプログラムの動作説明	16
3.2.3 UART サンプルプログラムの動作説明	18
3.2.4 USB ホストサンプルプログラムの動作説明	19
3.2.5 USB ファンクションサンプルプログラムの動作説明	20
3.2.6 audio サンプルプログラムの動作説明	21
3.3 PSRAM の使い方と注意事項	22
3.4 MCUXpresso IDE を用いたプロジェクトのビルド・デバッグと外部 ROM への書込み	23

1. 概要

1.1 概要

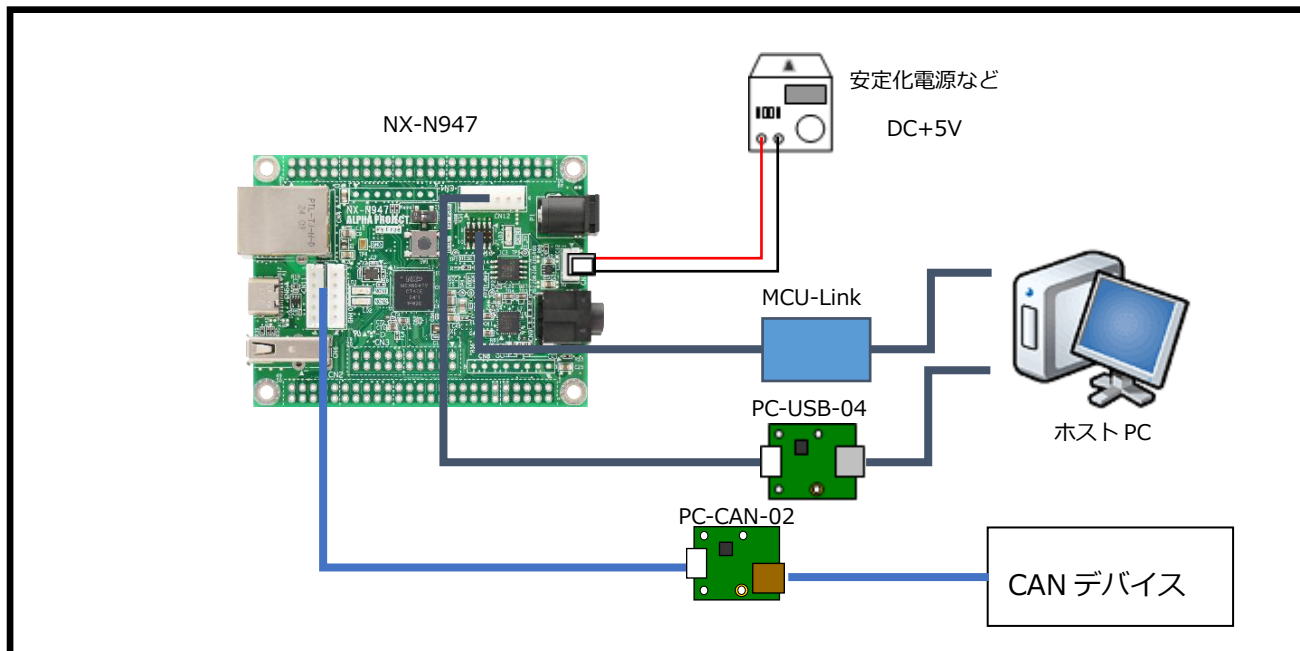
本アプリケーションノートでは、NX-N947 に付属するサンプルプログラムについて解説します。
 解説するサンプルプログラムは下記のものになります。

デバイス	機能
NX-N947	<ul style="list-style-type: none"> ・ CAN 通信 ・ ネットワーク通信 ・ UART 通信 ・ USB ホスト メモリ読み書き ・ USB ファンクション 仮想 COM 通信 ・ audio 入出力

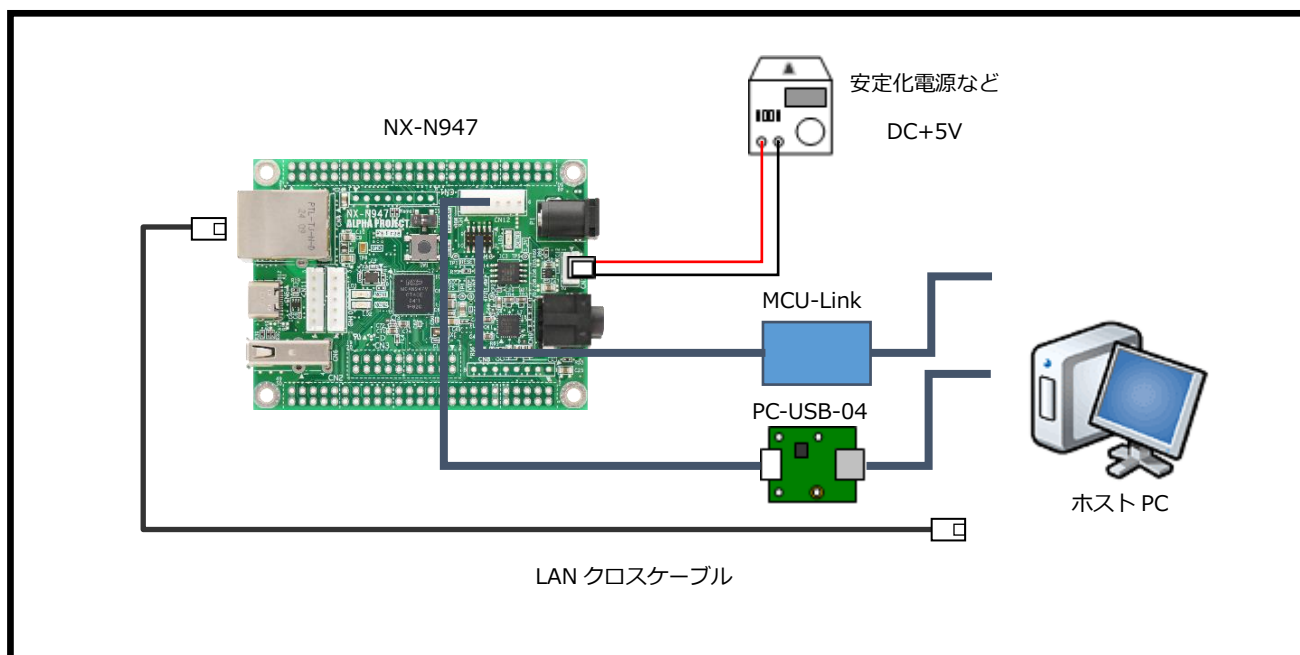
1.2 接続概要

本サンプルプログラムの動作を確認する上で必要な CPU ボードの接続例を以下に示します。
 詳細な接続に関しては後述の「3.2 動作説明」を参照してください。

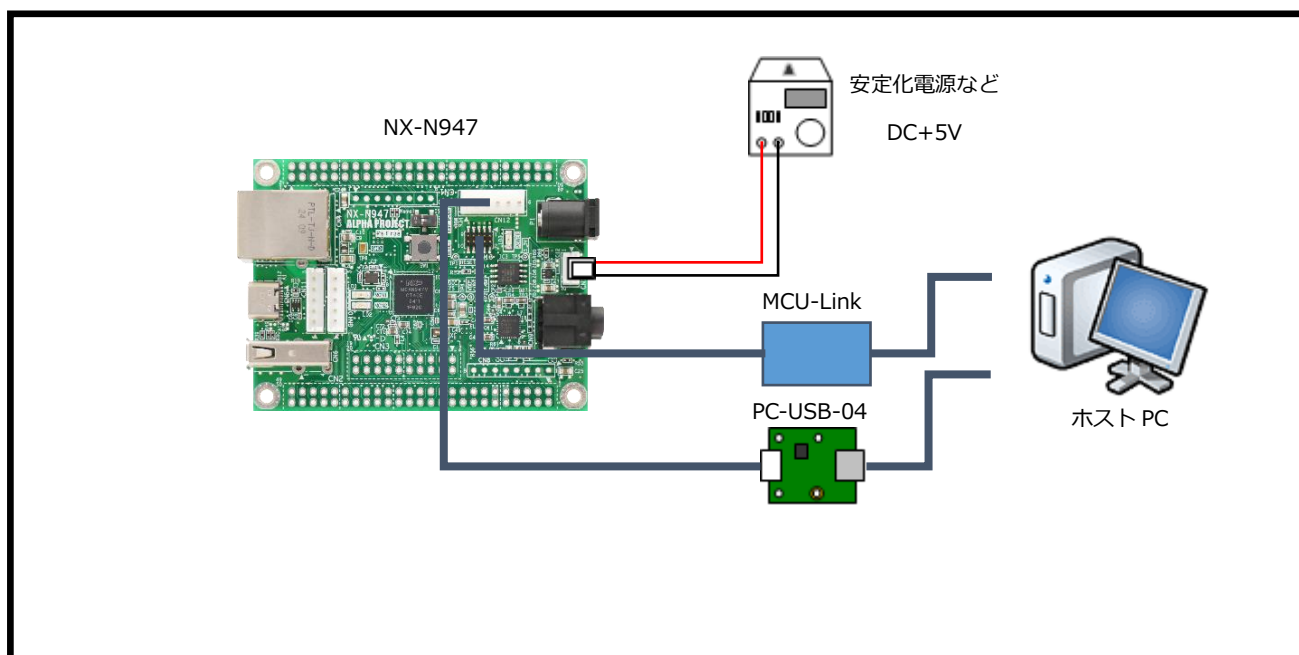
- ・ CAN サンプルプログラム動作時の接続例



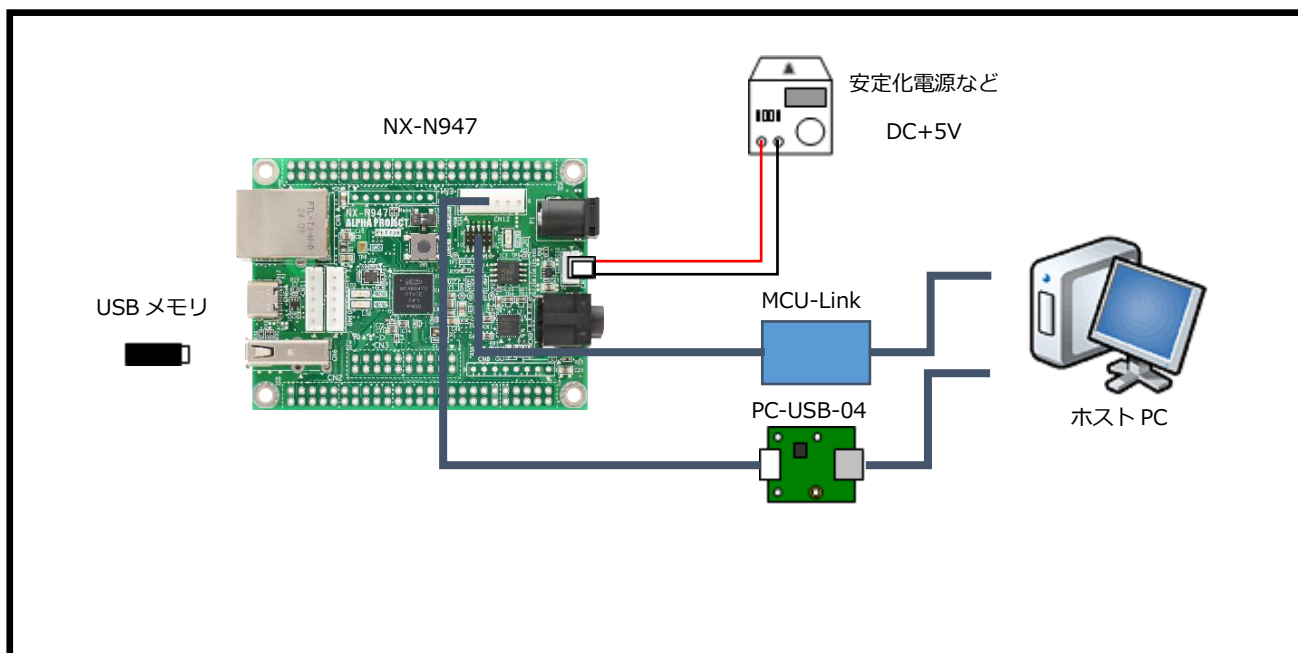
・ Ethernet サンプルプログラム動作時の接続例



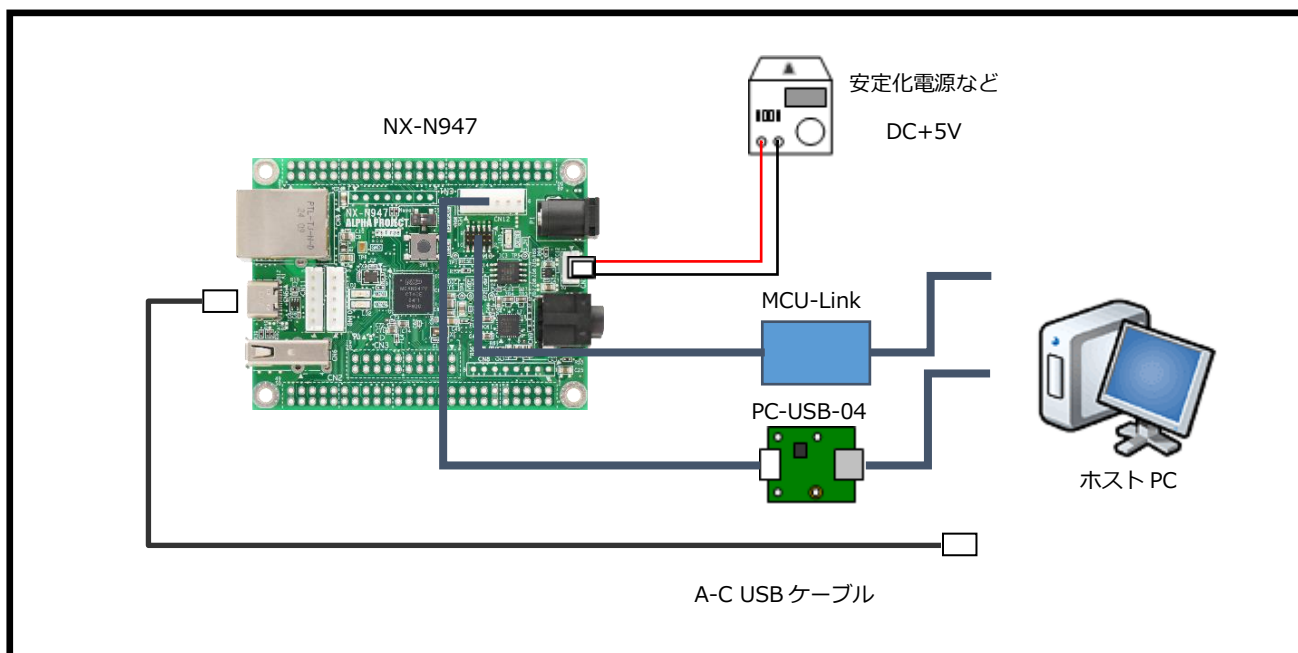
・ UART サンプルプログラム動作時の接続例



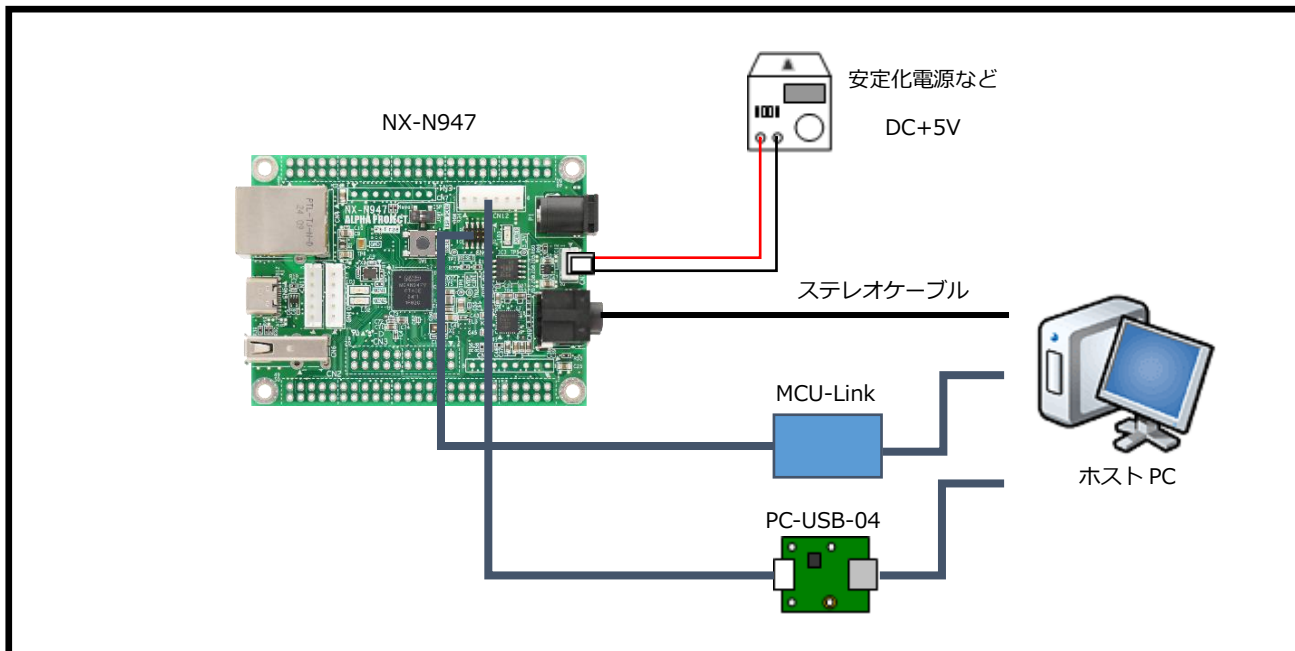
・ USB ホストサンプルプログラム動作時の接続例



・ USB ファンクションサンプルプログラム動作時の接続例



- ・ audio サンプルプログラム動作時の接続例



1.3 本サンプルプログラムについて

本サンプルプログラムおよび本書含むアプリケーションノートは、弊社 Web サイトのボード紹介ページで公開されています。

株式会社アルファプロジェクト

NX-N947 製品ページ <https://www.apnet.co.jp/product/nx/nx-n947.html>

1.4 開発環境について

本サンプルプログラムは統合開発環境「MCUXpresso IDE」と「MCUXpresso SDK」を用いて開発されています。本サンプルプログラムに対応する開発環境、FSP、コンパイラ、デバッガのバージョンは次の通りです。

ソフトウェア	バージョン	備考
MCUXpresso IDE	V24.12[Build 148]	–
MCUXpresso SDK	V24.12.00	SDK_2.x_MCXN947を使用

デバッガ	ハードウェアバージョン	備考
MCU-Link	V3.146	–



サンプルプログラムは、XIPモードでの動作ができる形にしています。
 この場合、PLL1周波数は192MHzに固定されてしまうため、SAI用クロックは12MHzとなり、サンプルプログラムは15.625Kサンプリングになります。
 XIPモードを使用しない場合、PLL1周波数は任意に設定できますので、SAI用クロックに12.285MHz等を設定することができます。

1.5 ワークスペースについて

本サンプルプログラムのプロジェクトファイルは次のフォルダに格納されています。

サンプルプログラム	フォルダ
CAN サンプルプログラム	¥sample¥nx_n947_can_sample
Ethernet サンプルプログラム	¥sample¥nx_n947_ether_sample
UART/PSRAM サンプルプログラム	¥sample¥nx_n947_uart_sample
USB ホストサンプルプログラム	¥sample¥nx_n947_usb_host_sample
USB ファンクションサンプルプログラム	¥sample¥nx_n947_usb_pcdc_sample
Audio サンプルプログラム	¥sample¥nx_n947_audio_sample

2. サンプルプログラムの構成

2.1 フォルダ構成

サンプルプログラムは下記のようなフォルダ構成になっています。



(前ページより)

<ul style="list-style-type: none"> — nx_n947_uart_sample <ul style="list-style-type: none"> — .settings — source — Debug — Release — board — CMSIS — component — device — doc — drivers — linkscript — startup — utilities — nx_n947_usb_hmsc_sample <ul style="list-style-type: none"> — .settings — source — Debug — Release — board — CMSIS — component — device — doc — drivers — fatfs — frdmmcxn947 — freertos — linkscript — startup — usb — utilities 	<ul style="list-style-type: none"> UART サンプルプログラム 設定ファイルフォルダ ソースフォルダ ワークフォルダ (RAM ダウンロード用) ワークフォルダ (Flash ダウンロード用) Config ツール出力ファイル Manage SDK components (CPU 周り) Manage SDK components (コンポーネント) Manage SDK components (デバイス関連) Manage SDK components (ドキュメント) Manage SDK components (ドライバ関連) リンクスクリプト修正部分フォルダ Manage SDK components (スタートアップ) Manage SDK components (ユーティリティ) USB ホストサンプルプログラム 設定ファイルフォルダ ソースフォルダ ワークフォルダ (RAM ダウンロード用) ワークフォルダ (Flash ダウンロード用) Config ツール出力ファイル Manage SDK components (CPU 周り) Manage SDK components (コンポーネント) Manage SDK components (デバイス関連) Manage SDK components (ドキュメント) Manage SDK components (ドライバ関連) Manage SDK components (FatFS) frdmmcxn947 用ファイル (ファイルなし) Manage SDK components (FreeRTOS) リンクスクリプト修正部分フォルダ Manage SDK components (スタートアップ) Manage SDK components (USB ドライバ) Manage SDK components (ユーティリティ)
---	---

(次ページへ)

(前ページより)

<ul style="list-style-type: none"> — nx_n947_usb_pcdc_sample <ul style="list-style-type: none"> — .settings — source — Debug — Release — board — CMSIS — component — device — doc — drivers — frdmmcxn947 — freertos — linkscript — startup — usb — utilities — nx_n947_audio_sample <ul style="list-style-type: none"> — .settings — source — Debug — Release — board — CMSIS — component — device — doc — drivers — frdmmcxn947 — linkscript — middleware — startup — utilities — mex <ul style="list-style-type: none"> — nx_n947.mex — XIPxip <ul style="list-style-type: none"> — MCXN9xx_SFDP_FlexSPI_NX_N947.cfx 	<ul style="list-style-type: none"> USB ファンクションサンプルプログラム 設定ファイルフォルダ ソースフォルダ ワークフォルダ (RAM ダウンロード用) ワークフォルダ (Flash ダウンロード用) Config ツール出力ファイル Manage SDK components (CPU 周り) Manage SDK components (コンポーネント) Manage SDK components (デバイス関連) Manage SDK components (ドキュメント) Manage SDK components (ドライバ関連) frdmmcxn947 用ファイル (ファイルなし) Manage SDK components (FreeRTOS) リンクスクリプト修正部分フォルダ Manage SDK components (スタートアップ) Manage SDK components (USB ドライバ) Manage SDK components (ユーティリティ) Audio サンプルプログラム 設定ファイルフォルダ ソースフォルダ ワークフォルダ (RAM ダウンロード用) ワークフォルダ (Flash ダウンロード用) Config ツール出力ファイル Manage SDK components (CPU 周り) Manage SDK components (コンポーネント) Manage SDK components (デバイス関連) Manage SDK components (ドキュメント) Manage SDK components (ドライバ関連) frdmmcxn947 用ファイル (ファイルなし) リンクスクリプト修正部分フォルダ Manage SDK components (美ドルウェア) Manage SDK components (スタートアップ) Manage SDK components (ユーティリティ) MCUXpresso Config Tools Settings File 弊社製 CPU ボード用に雛形としてクロックのみを設定 MCUXpresso IDE Flash 書き込み用ドライバファイル
--	---

2.2 ファイルの構成

サンプルプログラムは以下のファイルで構成されています。

本節では、サンプルプログラムの作成にあたって追加したファイルについてのみ記述し、自動生成ファイルなどに関しては説明を省略します。

・CAN サンプルプログラム

<¥sample¥nx_n947_can_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_can_sample.mex	...	NX-N947 CAN サンプルプログラム カスタムボードファイル

<¥sample¥nx_n947_can_sample¥source フォルダ内>

flexcan_interrupt_transfer.c	...	flexcan_interrupt_transfer 関数ソースファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル

・Ethernet サンプルプログラム

<¥sample¥nx_n947_ether_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_ether_sample.mex	...	NX-N947 Ethernet サンプルプログラム カスタムボード ファイル

<¥sample¥nx_n947_ether_sample¥source フォルダ内>

FreeRTOSConfig.h	...	FreeRTOSConfig ヘッダファイル
lwip_ping_freertos.c	...	メインルーチンソースファイル
lwipopts.h	...	lwip_config ヘッダファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル

・ UART サンプルプログラム

<¥sample¥nx_n947_uart_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_uart_sample.mex	...	NX-N947 UART サンプルプログラム カスタムボードファイル

<¥sample¥nx_n947_uart_sample¥source フォルダ内>

app.h	...	アプリケーションヘッダファイル
checkblank.s	...	CheckBlank アセンブリソース
flexspi_octal_flash_ops.c	...	PSRAM ドライバソースファイル
flexspi_octal_polling_transfer.c	...	PSRAM アプリケーションソースファイル
lpuart_interrupt.c	...	UART アプリケーションソースファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル
semihost_hardfault.c	...	HardFault ハンドラソースファイル

・ USB Host サンプルプログラム

<¥sample¥nx_n947_hmsc_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_usb_hmsc_sample.mex	...	NX-N947 USB Host サンプルプログラム カスタムボードファイル

<¥sample¥nx_n947_hmsc_sample¥source フォルダ内>

app.c	...	アプリケーションソースファイル
app.h	...	アプリケーションヘッダファイル
ffconf.h	...	FatFs コンフィグファイル
FreeRTOSConfig.h	...	FreeRTOSConfig ヘッダファイル
host_msdfatfs.c	...	マスタストレージドライバソースファイル
host_msdfatfs.h	...	マスタストレージドライバヘッダファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル
usb_host_config.h	...	USB Host コンフィグヘッダファイル

<¥sample¥nx_n947_hmsc_sample¥source¥template¥usb フォルダ内>

ffconf_gen.h	...	FatFs コンフィグファイル
--------------	-----	-----------------

・ USB Function サンプルプログラム

<¥sample¥nx_n947_usb_pcdc_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_usb_pcdc_sample.	...	NX-N947 USB Function サンプルプログラム カスタムボード
mex		ファイル

<¥sample¥nx_n947_usb_pcdc_sample ¥source フォルダ内>

FreeRTOSConfig.h	...	FreeRTOSConfig ヘッダファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル
usb_device_config.h	...	USD デバイス設定ヘッダファイル
usb_device_descriptor.c	...	USB ディスクリプタソースファイル
usb_device_descriptor.h	...	USB ディスクリプタヘッダファイル
virtual_com.c	...	仮想 COM ソースファイル
virtual_com.h	...	仮想 COM ヘッダファイル

・ Audio サンプルプログラム

<¥sample¥nx_n947_audio_sample フォルダ内>

.cproject	...	CPROJECT ファイル
.project	...	PROJECT ファイル
nx_n947_audio_sample.mex	...	NX-N947 Audio サンプルプログラム カスタムボードファイル

<¥sample¥nx_n947_audio_sample ¥source フォルダ内>

main.c	...	メインルーチンソースファイル
mcxn_flexspi_nor_config.c	...	XIP コンフィグレーションソースファイル
mcxn_flexspi_nor_config.h	...	XIP コンフィグレーションヘッダファイル
playbackSineWave.c	...	サイン波再生処理ソースファイル
recordPlayback.c	...	音声エコーバック処理ソースファイル
sai.h	...	SAI 処理ヘッダファイル

注意

サンプルプログラムをインポートした後で、クロックを変更するなどをしてコードを再出力する際、操作を誤るとサンプルプログラムの「Config ツール出力ファイル」フォルダ以下および「Manage SDK components」が含まれる複数のファイルが上書き(サンプルプログラム向けの修正箇所が変更前の状態に戻されてしまう)されてしまいます。

この結果、プログラムが動作しなくなることがありますので、取り扱いには注意してください。

別書 AN2301 NX シリーズ 開発チュートリアル 3.2.項を参照ください。

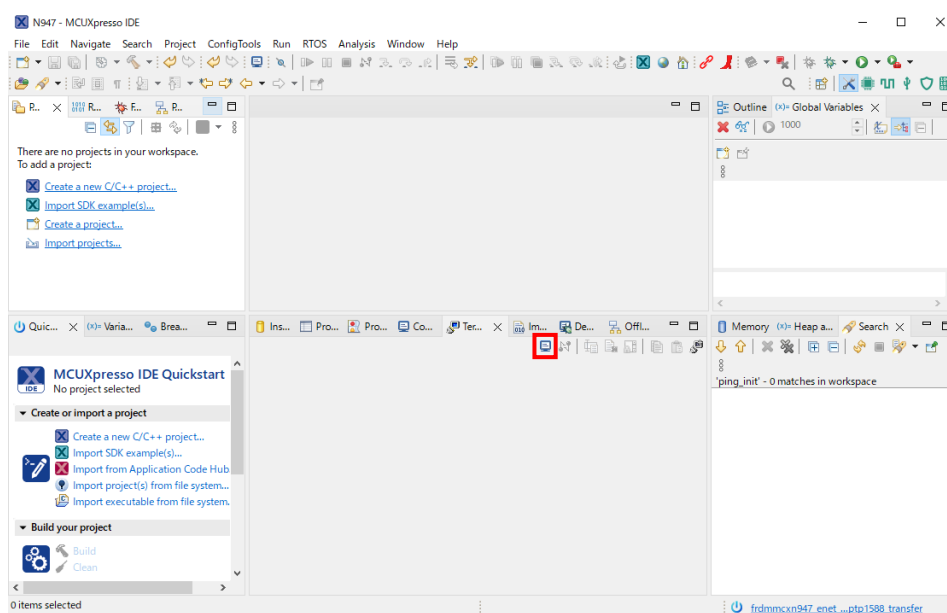
3. NX-N947 サンプルプログラム

3.1 Debug Console 機能を用いたシリアルターミナルの入出力

サンプルプログラムは、MCUXpresso IDE が提供している Debug Console 機能を使用し、動作状況やプログラムの情報が表示される動作があります。

Debug Console 機能を使用した接続は以下の手順に従ってください。

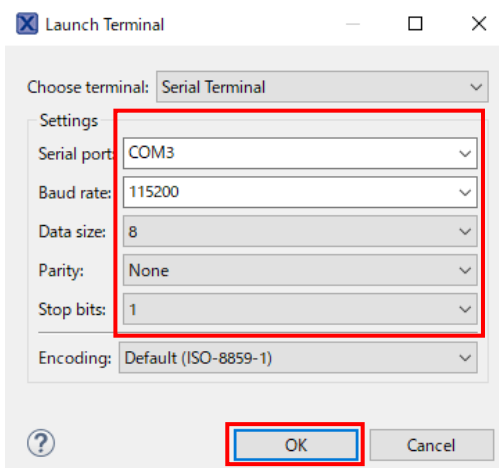
- ① CPU ボードとホスト PC を、UART コネクタ経由で接続します。
- ② ホスト PC 上で MCUXpresso IDE のターミナルソフトの設定画面を起動します。



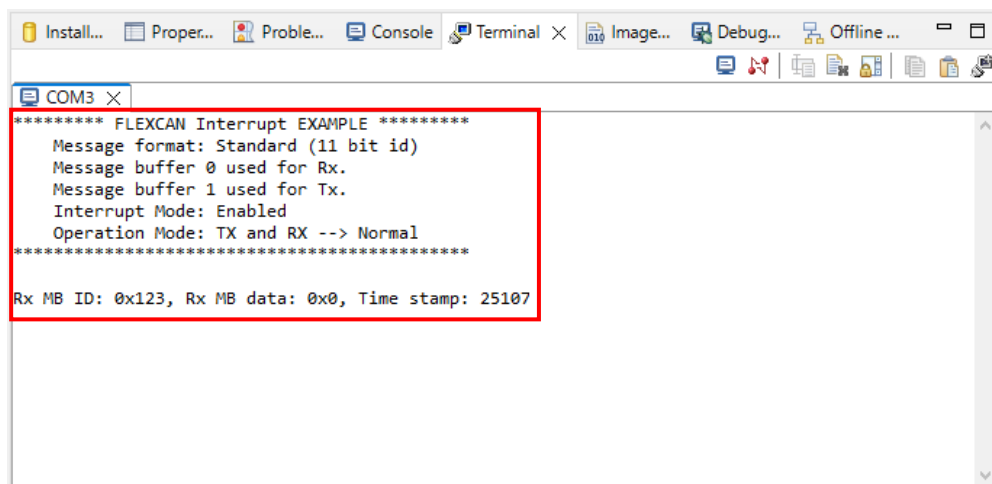
- ③ UART の設定を、115200bps、ビット長 8、パリティなし、ストップビット 1 に設定し、[OK]ボタンを押下し、ターミナルを起動してください。

※ただし[Serial port]に関しては環境によって異なります。

デバイスマネージャー等で使用されている PORT を確認の上、設定してください

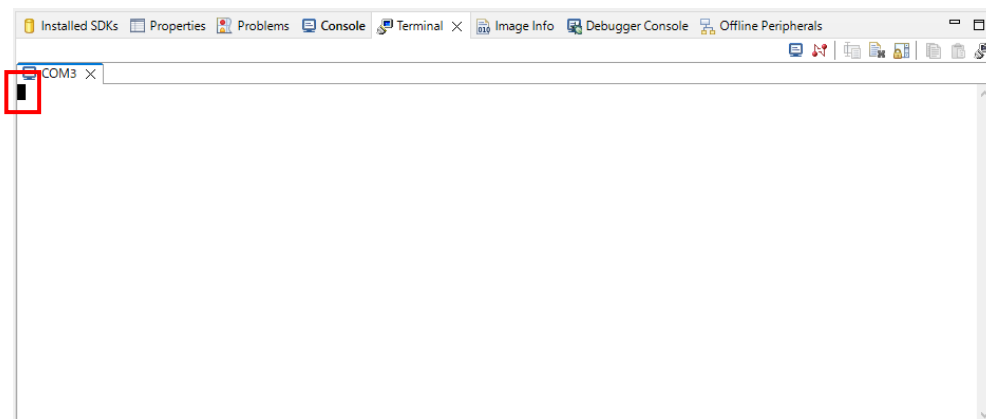


- ④ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ⑤ 接続が完了しますと、CPU ボードからの出力内容が表示されます。



```
***** FLEXCAN Interrupt EXAMPLE *****  
Message format: Standard (11 bit id)  
Message buffer 0 used for Rx.  
Message buffer 1 used for Tx.  
Interrupt Mode: Enabled  
Operation Mode: TX and RX --> Normal  
*****  
Rx MB ID: 0x123, Rx MB data: 0x0, Time stamp: 25107
```

- ⑥ CPU ボードに入力する場合はカーソルが表示されていればいつでも入力することができます。



```
***** FLEXCAN Interrupt EXAMPLE *****  
Message format: Standard (11 bit id)  
Message buffer 0 used for Rx.  
Message buffer 1 used for Tx.  
Interrupt Mode: Enabled  
Operation Mode: TX and RX --> Normal  
*****  
Rx MB ID: 0x123, Rx MB data: 0x0, Time stamp: 25107  
  
|
```

- ⑦ 動作確認が完了したら、ターミナルソフトを終了し、CPU ボードの電源を落とします。

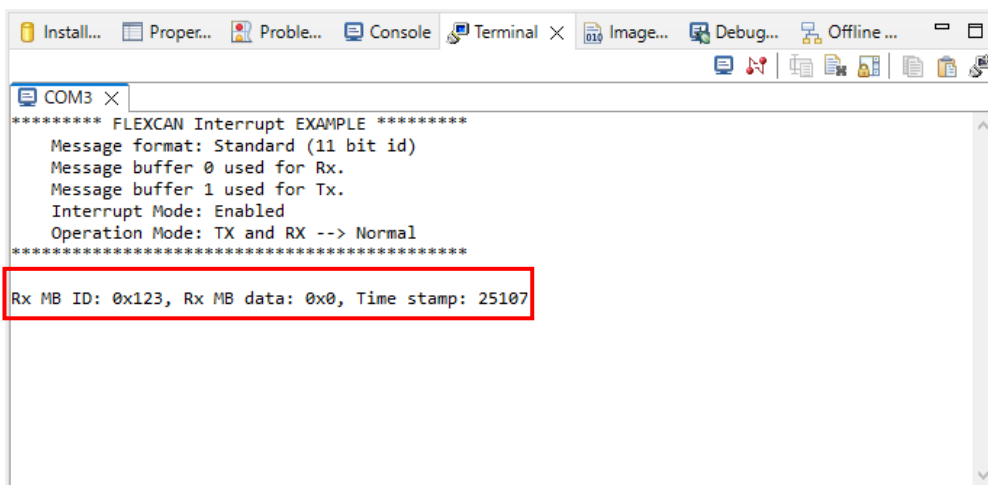
3.2 動作説明

3.2.1 CAN サンプルプログラムの動作説明

本サンプルプログラムでは CAN0・CAN1 を用いてエコーバック通信を行います。
 使用する CAN チャンネルの設定は、flexcan_interrupt_transfer.c の EXAMPLE_CAN の宣言で行います。
 CAN の通信設定は、以下のように行われます。
 なおどの CAN チャンネルに接続した場合も、動作は共通です。

CPU ボードの設定		
ID	受信 Mailbox ID	B'01100100001 (0x321)
	送信 Mailbox ID	B'00100100011 (0x123)
フォーマット		スタンダードフォーマット、データフレーム、データ長 8 バイト
通信速度		500kbps

またホスト PC から入力されたデータに対して、そのデータの Mailbox ID、入力データの 1bit 目の値、入力された際の timestamp がターミナルソフトにて表示されます。



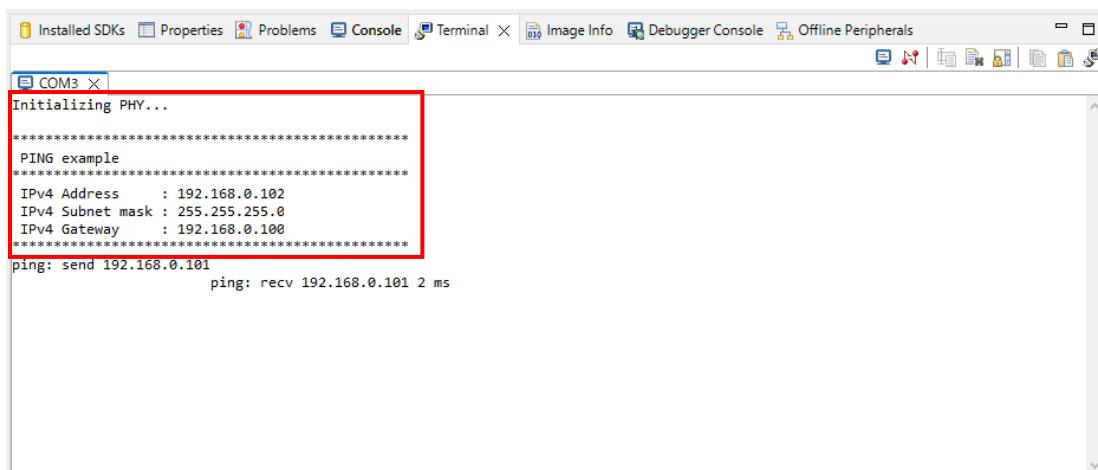
3.2.2 Ethernet サンプルプログラムの動作説明

本サンプルプログラムでは、CPU ボードから固定の IP アドレスに Ping 通信を行います。
 ホスト PC を下記のネットワーク設定で動作させ、ボードへ接続した後、通信動作を行ってください。
 ネットワーク動作の確認は、以下の手順に従い行ってください。
 またプログラムを動作させる前に、ファイアウォールは無効にしてください。

- ① LAN クロスケーブルを用い、CPU ボードの LAN コネクタ (CN4) とホスト PC を接続します。
- ② ホスト PC 上でネットワークの設定を行います。
 CPU ボードの設定に合わせるため、ホスト PC のネットワーク設定を下記の内容に変更してください。

IP アドレス	192.168.0.102
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.0.100

- ③ CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ④ サンプルプログラムが正常に動作した場合は、以下のように、ターミナル画面にネットワーク設定が表示され、その後、IP アドレス「192.168.0.101」に対し Ping 通信が開始されます。



3.2.2.1 ネットワーク推奨環境

本サンプルプログラムに実装されたネットワーク通信の確認に必要な推奨環境は以下の通りです。

ホスト PC	PC/AT 互換機
OS	Windows 10/11
LAN ポート	10/100BASE-TX 以上対応の LAN ポート
LAN ケーブル	クロスケーブル

3.2.2.2 ネットワーク設定

本 CPU ボードのネットワーク設定は以下の通りです。

IP アドレス	192.168.0.102
サブネットマスク	255.255.255.0
ゲートウェイ	192.168.0.100
MAC アドレス	00-0C-7B-60-XX-XX ※ XX-XX の値は製品基板ごとに異なります。
Ping 通信先 IP アドレス	192.168.0.101

上記設定のうち、IP アドレス・サブネットマスク・ゲートウェイの設定は、サンプルプログラム内で定義しています。各設定の定義は以下の通りです。

<¥sample¥n947_ether_sample¥source¥lwip_ping_freertos.c>

設定	CPU ボードの設定
IP アドレス	configIP_ADDR
サブネットマスク	configNET_MASK
ゲートウェイ	configGW_ADDR

<¥sample¥n947_ether_sample¥source¥lwip_ping_freertos.c>

設定	CPU ボードの設定
Ping 通信先 IP アドレス	configUIP_ADDR

また、MAC アドレスは EEPROM の先頭 6Byte に格納されています。

アドレス (CH0)	格納値
先頭アドレス + 0x00	0x00
+ 0x01	0x0C
+ 0x02	0x7B
+ 0x03	0x60
+ 0x04	0xXX
+ 0x05	0xXX

※ 0xXX の値は製品ごとに異なります

本製品の MAC アドレスは、弊社が米国電気電子学会 (IEEE) より取得したアドレスとなります。MAC アドレスを変更される際は、お客様にて IEEE より MAC アドレスを取得し、設定してください。

3.2.3 UART サンプルプログラムの動作説明

本サンプルプログラムでは、UART の動作と PSRAM の初期化を行います。

- ・ UART の動作

LPUART4 にて、通信を行います。

起動後、ボード側が PC に対して

「Lpuart functional API interrupt example

Board receives characters then sends them out

Now please input:」

を送信します。その後、エコーバックを行います。

UART の設定は、115200bps、ビット長 8、パリティなし、ストップビット 1、フロー制御なしです。

動作の確認は、ホスト PC 上のターミナルソフト（ハイパーターミナル等）を使用してください。

- ・ PSRAM の動作

ボードに実装されている PSRAM の初期化のみを行います(UART 動作には影響しません)。

PSRAM の使用方法につきましては、3.3 項を参照ください。

3.2.4 USB ホストサンプルプログラムの動作説明

本サンプルプログラムでは、USB メモリへのフォルダの生成からデータ読み書きを行うことができます。USB ホストの動作は、以下の手順に従って行ってください。

- ① UART 経由で、CPU ボードとホスト PC を接続します。
- ② ホスト PC 上でターミナルソフトを起動します。
- ③ CPU ボードの USB ホストポート(CN6)に USB メモリを挿入して電源を投入し、サンプルプログラムを動作させます。
- ④ 以下の出力とともに、「DIR1」および「DIR2」フォルダが生成されるとともに、「A～S」の文字列が入力された「F_2.DAT」ファイルが生成されます。

```

Install... Proper... Proble... Console Terminal X Image... Debug... Offline...
COM3 X
host init done
mass storage device attached:pid=0x1ddvid=0x411 address=1
.....fatfs test.....
fatfs mount as logiacal driver 1.....success
test f_mkfs.....success
test f_getfree:
  FAT type = FAT32
  bytes per cluster = 32768; number of clusters=235618
  The free size: 7539744KB, the total size:7539776KB
directory operation:
list root directory:

create directory "dir_1".....success
create directory "dir_2".....success
create sub directory "dir_2/sub_1".....success
list root directory:
  dir - ___ - DIR_1 - 0Bytes - 2018-1-1 0:0:0
  dir - ___ - DIR_2 - 0Bytes - 2018-1-1 0:0:0
list directory "dir_1":
  dir - ___ - SUB_1 - 0Bytes - 2018-1-1 0:0:0
rename directory "dir_1/sub_1" to "dir_1/sub_2".....success
delete directory "dir_1/sub_2".....success
get directory "dir_1" information:
  dir - ___ - DIR_1 - 0Bytes - 2018-1-1 0:0:0
change "dir_1" timestamp to 2015.10.1, 12:30:0.....success
get directory "dir_1" information:
  dir - ___ - DIR_1 - 0Bytes - 2015-10-1 12:30:0
file operation:
create file "f_1.dat".....success
test f_write.....success
test f_printf.....success
test f_puts.....success
test f_putc.....success
test f_seek.....success
test f_gets.....ABCDEFGHI
test f_read.....JKLMNOPQRS
test f_truncate.....success
test f_close.....success
get file "f_1.dat" information:
  fil - ___ - F_1.DAT - 19Bytes - 2018-1-1 0:0:0
change "f_1.dat" timestamp to 2015.10.1, 12:30:0.....success
change "f_1.dat" to readonly.....success
get file "f_1.dat" information:
  fil - R__ - F_1.DAT - 19Bytes - 2015-10-1 12:30:0
remove "f_1.dat" readonly attribute.....success
get file "f_1.dat" information:
  fil - ___ - F_1.DAT - 19Bytes - 2015-10-1 12:30:0
rename "f_1.dat" to "f_2.dat".....success
.....test done.....

```

3.2.5 USB ファンクションサンプルプログラムの動作説明

USB ファンクション動作の確認は、以下の手順に従って行ってください。

- ① USB ケーブルを使い CPU ボードの USB ファンクションポート(CN5)とホスト PC の USB ポートを接続します。
- ② CPU ボードに電源を投入し、サンプルプログラムを動作させます。
- ③ ホスト PC 上でターミナルソフト（ハイパーターミナルなど）を起動し、COM ポートの設定を行います。
COM ポートを以下の設定に変更します。

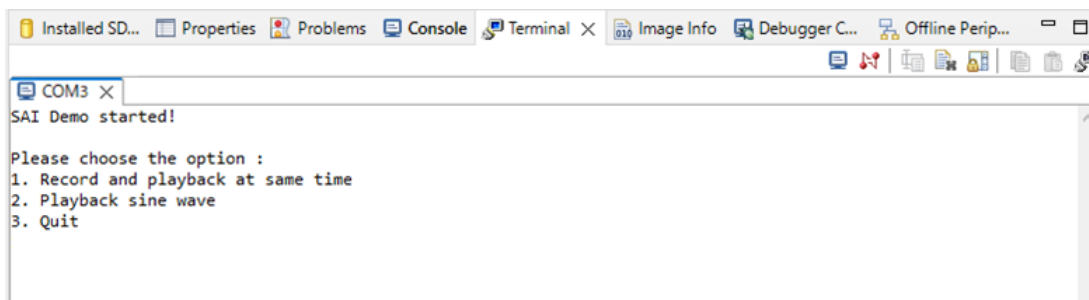
ボーレート	115200bps
ビット長	8bit
パリティ	なし
ストップビット	1bit
フロー制御	なし

- ④ ターミナルソフトから任意のデータを送信しますと、送信した内容がエコーバックで受信されます。

3.2.6 audio サンプルプログラムの動作説明

本サンプルプログラムでは、音声の入出力を行うことができます。
audio の動作は、以下の手順に従って行ってください。

- ① UART 経由で、CPU ボードとホスト PC を接続します。
- ② ホスト PC 上でターミナルソフトを起動します。
- ③ CPU ボード上のステレオミニジャック(CN9)の上段にイヤホン、下段にマイクを接続し電源を入れ、サンプルプログラムを動作させてください。
- ④ サンプルプログラムが正常に動作した場合は、以下のように、ターミナル画面に audio 動作ログが表示され、その後コマンド受信待ち状態が表示されます。



- ⑤ 下記コマンド一覧を参考に、ターミナルにコマンドを入力します。
入力されたコマンドに合わせて、下記の動作が行われることを確認してください。

コマンド	説明
1	入力された音声を出力します。
2	正弦波を音声として出力します。
3	Audio サンプルプログラムを終了します。

3.3 PSRAM の使い方と注意事項

PSRAM へのアクセスには、下記の 2 種類の方法があります。

- ・ IP コマンド : SPI のコマンドによるアクセス
- ・ AHB コマンド : XIP を使用したバスアクセス

本サンプルプログラムでは、AHB コマンドで、アクセスを行うよう初期化を行っています。

RAM 上で動作するサンプルプログラム (RAM ダウンロード用) では、IP コマンド、AHB コマンドどちらでもアクセスも行えますが、FlashROM で動作するサンプルプログラム (Flash ダウンロード用) では、コード領域が XIP モードを使用した QSPI Flash となりますので、AHB コマンドのみとなります。

また、制限事項として、下記があります。

- ① QSPI Flash を PSRAM のクロック周波数は共通となります。
- ② XIP 動作時に FlexSPI_A ポートのクロックは、96MHz 固定となり、変更できません。
- ③ 本サンプルプログラムは全て XIP 動作ができるようにしてあり、Audio 機能に制限があります。
XIP 動作時、SAI 用クロック源 (PLL1) が 12MHz に固定となってしまうため、サンプルプログラムは 15.625k サンプル動作になります。
※XIP 動作しない場合は、PLL1 を 12.285MHz などに設定することが可能です。

3.4 MCUXpresso IDE を用いたプロジェクトのビルド・デバッグと外部 ROM への書込み

サンプルプログラムを CPU 上で実行するためには、MCUXpresso IDE 上にサンプルプログラムをインポートし、ビルドを行う必要があります。

MCUXpresso IDE へのサンプルプログラムのインポート、ビルド、デバッグ方法につきましては、

別書 **AN2301 NX シリーズ 開発チュートリアル** に以下の詳細な手順が記されています。

- ・ 3.1.2 インポート方法
- ・ 3.1.3 ビルド構成の変更方法
- ・ 3.1.4 ビルド方法
- ・ 3.1.5 内蔵 RAM を用いたデバッグ、ダウンロード方法
- ・ 3.1.6 外部 ROM を用いたデバッグ、ダウンロード方法
- ・ 5.1 USB インターフェースを使用した書込み方法

ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
- ・本サンプルプログラムで使用されているミドルウェアおよびドライバの著作権は NXP セミコンダクターズ社が保有します。
- ・本文書に記載されている内容およびサンプルプログラムについてのサポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしました。が、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡ください。
- ・本文書の内容は、将来予告なしに変更されることがあります。

商標について

- ・ MCX N947 は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・ MCUXpresso IDE は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・ MCUXpresso SDK は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・ MCUXpresso Secure Provisioning は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- ・ Windows®の正式名称は Microsoft®Windows®Operating System です。
- ・ Microsoft、Windows は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
- ・ Windows®10、Windows®11 は、米国 Microsoft Corporation.の商品名称です。
本文書では下記のように省略して記載している場合がございます。ご了承ください。
Windows®10 は Windows 10 もしくは Win10
Windows®11 は Windows 11 もしくは Win11
- ・ その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市中央区積志町 834
<https://www.apnet.co.jp>
E-Mail: query@apnet.co.jp