Spresense マイコンで ELTRES 通信を試す

ELTRES 通信の評価・PoC などにご利用いただける送信モジュールボードに、高機能・低消費電力を特徴とするマイコンボード Spresense と組み合わせる製品が用意されています。これらのボードと Spresense SDK(開発キット)とを使用することで簡単に ELTRES の通信を試すことができますのでその手順をご紹介いたします。

Spresense 対応 ELTRES 評価用モジュールボードのご紹介

2023 年 8 月現在、Spresense に対応した ELTRES 評価用モジュールボードとして以下の二機種が商品化されています。

ELTRES SPEXEL for Spresense

株式会社ライズナー 製 (https://device.risner.jp/products/detail/37)
Spresense と同サイズの基板で、Spresense の裏面 Board-to-Board 拡張コネクタに装着します。マイクロ SD カードコネクタや操作用スイッチも実装されており、容易に ELTRES 送信端末を構成することができます。ELTRES 送信モジュールとして CXM1501AGR を搭載しています。

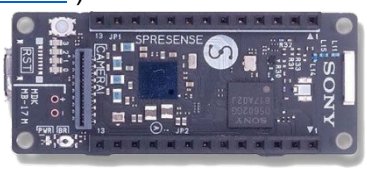
ELTRES アドオン IoT 開発キット

株式会社クレスコ 製(https://wakuwaku.cresco.co.jp/solution/eltres)
Spresense の表面 I/O ソケットに装着します。ELTRES 送信モジュールと周辺回路のみのシンプルな構成となっており、Spresense 以外のマイコンへの応用も考えられます。ELTRES 送信モジュールとして CXM1501GR または CXM1501AGR を搭載しています。
Spresense 拡張ボード(CXD5602PWBEXT1)を併用してマイクロ SD カード等を使用することができます。

上記のモジュールボードは、各々のボードメーカーが提供するライブラリで使用することもできますが、本記事においてはソニーセミコンダクタソリューションズで開発した共通のプログラムでの動作例をご紹介いたします。

Spresense について

ソニーセミコンダクタソリューションズ株式会社 製
(https://www.sony-semicon.com/ja/products/spresense/index.html)
6 コアの ARM Cortex M4F CPU を搭載した、コンパクトで強力かつ低消費電力のマイコンボードです。カメラを利用した組み込み AI などにも十分に対応できる能力を持っていますので、ELTRES モジュールボードと組み合わせる



ことで高度な IoT Edge 端末への応用も考えられます。

Spresense SDK

Spresense には、C/C++言語で本格的な開発を行う Spresense SDK と、Arduino IDE を用いた簡便な開発環境とが用意されています。そのほかにも Circuit Python 環境なども利用できますが、今回は、Spresense SDK を使用した開発をご紹介します。開発のフロントエンドとして Microsoft 社の Visual Studio Code(以下 VS Code と表記)を使用します。

Spresense SDK を使用するには、Web サイト Spresense SDK スタートガイド(IDE 版) に沿って Spresense SDK および VS Code のセットアップを行ってください。2023 年 8 月時点で SDK のバージョンは 3.1.0 です。

以降の説明は、上記サイトで説明されている環境のうち Windows 11 + WSL2(Ubuntu)環境を前提とします。
利用する環境が異なる場合は上記サイトに沿って適宜項目を読み替えてご利用ください。

ELTRES サンプルについて

Spresense SDK には 3 種類の ELTRES 用サンプルアプリケーションが含まれています。全て C 言語で記述されています。

SDK に含まれる ELTRES サンプルアプリケーション

- **ELTRES LPWA sample application**

ELTRES による基本的な通信を行うサンプルアプリケーションです。Spresense 側でデータを収集、ペイロードデータを生成して送信モジュールに設定し、そのデータを送信モジュールが受信局に向けて送信します。

- **ELTRES standalone sample application**

ELTRES 送信モジュールの 2 世代目 CXM1501AGR では、ELTRES 送信モジュール単独、あるいは外部にホストマイコンを使用する場合も簡単な制御で端末を構成できるように、ELTRES 通信を自律的に制御する Standalone 機能が追加されました。
eltres_standalone はこの機能を使用するサンプルアプリケーションです。ELTRES 送信モジュールは、GNSS からの位置情報などを自律的に収集し、ペイロードデータを生成して Spresense に通知します。Spresense 側では、得られたデータを一部編集して送信モジュールに書き戻し、モジュールはそのデータを送信します。このサンプルの実行には ELTRES 送信モジュールが CXM1501AGR であることが必要です。

- **ELTRES EEPROM read/write sample application**

ELTRES 送信モジュールの各種設定用の EEPROM に書き込み、読み出しを行うためのサンプルアプリケーションです。

注: サンプルアプリケーションで設定するペイロードのフォーマットは、クラウドサービスなどで決められているデータフォーマットとは異なる場合があります。必要に応じてペイロードのフォーマットを変更・編集してご利用ください。

サンプルのビルド、書き込み、起動

サンプルアプリケーションのビルド、Spresense への書き込み、起動を行う手順をご紹介いたします。

ワークスペースの作成

Spresense SDK スタートガイド (IDE 版)3.1. ワークスペースの作成 を参照して、ワークスペースセットアップウィザードを起動します。ワークスペースおよびプロジェクトフォルダーを作成して、保存してください。以降、myproject プロジェクトフォルダーを作成したものとして説明いたします。

コンフィグレーション

コンフィグレーションによって Spresense で使用するアプリケーションの構成を決定します。

ELTRES のサンプルアプリケーションをビルド、実行するにはそれぞれのアプリケーションに対して以下の定義済みコンフィグレーションを使用します。

[ELTRES LPWA sample application]

[ELTRES standalone sample application]

[ELTRES EEPROM read/write sample application]

定義済みコンフィグレーションの使用は、以下の手順によります。

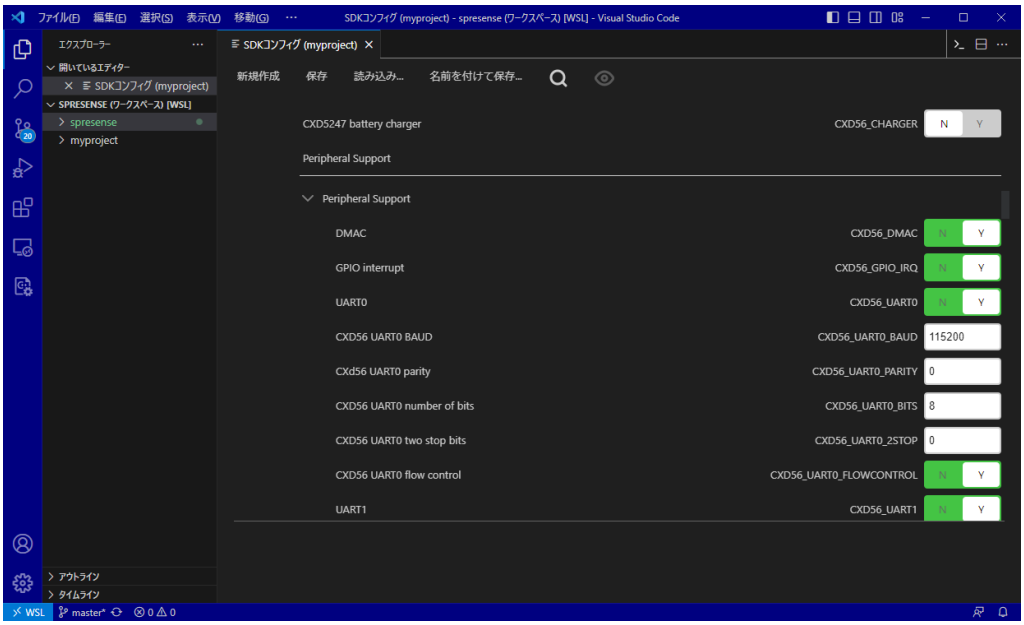
(参考:以下の説明において SDK コンフィグのメニュー項目を設定する際、SDK コンフィグタブの検索機能(虫眼鏡アイコン)を使用して項目名を入力すると素早く各メニュー項目へ移動することができます)

- 以前に行われたビルドを削除するために、Spresense プロジェクトメニュー(以下プロジェクトメニューと表記します)より[Spresense: アプリケーションのクリーン]を行ってください。
  (プロジェクトメニューは、VS Code のエクスプローラペインでプロジェクトのファイルやフォルダを右クリックして出るコンテキストメニューに含まれています。
  Spresense SDK スタートガイド (IDE 版)3.3 プロジェクトメニューについて を参照してください)

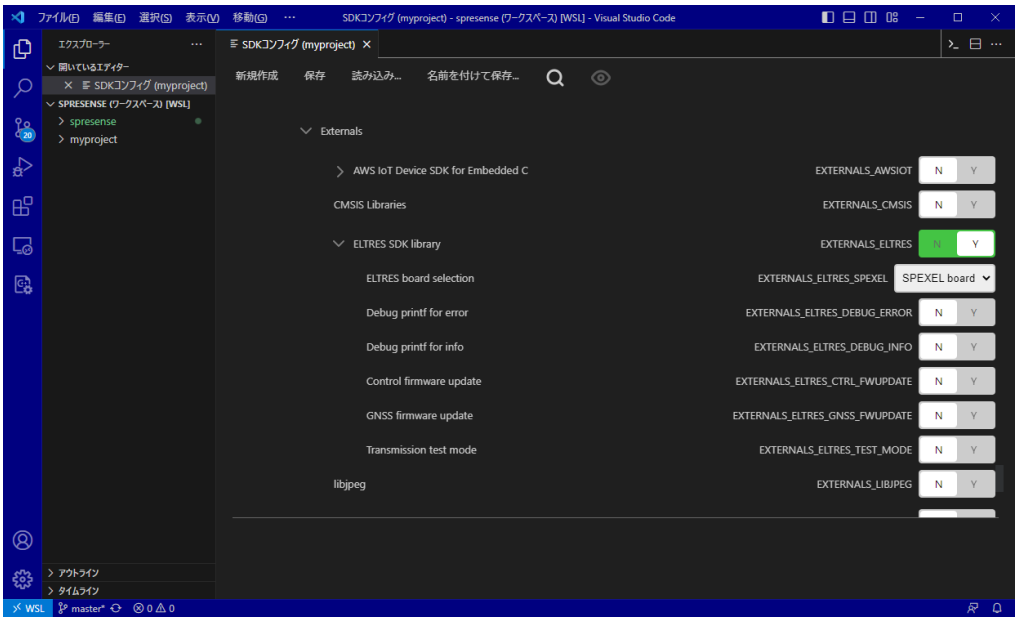
- プロジェクトメニューから[Spresense: SDK コンフィグ]を選択します。SDK コンフィグのタブが開くまで多少時間がかかることがあります。

- SDK コンフィグのタブが開いたら、タブ内の[新規作成]をクリックして、表示される[New Configuration]ダイアログの内容を変更せずに[OK]を押して、コンフィグレーションの初期化を行います。

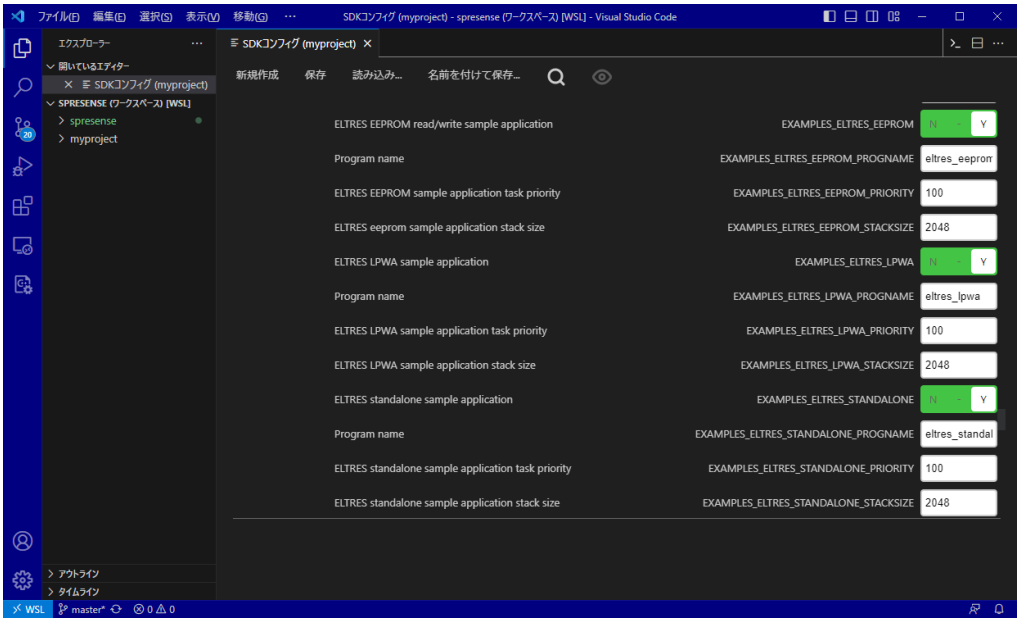
- 使用するボードが SPEXEL の場合、SDK コンフィグの項目表示の[> System Type]の'>'をクリックして下位メニューを表示し、[> CXD56xx Package Configuration] - [> Peripheral Support] - [UART0]を[Y]に、[CXD56 UART0 BAUD]を[115200]に、[CXD56 UART0 flow control]を[Y]に設定します。



- 全てのボードにおいて、SDK コンフィグの[> Application Configuration] - [> Spresense SDK] - [> Externals] - [> ELTRES SDK library]を[Y]に設定して展開し、[ELTRES board selection]で使用するボードを選択します(このとき、先に UART0 の設定がされていないと SPEXEL は選択できません)。



- [> Application Configuration] - [> Spresense SDK] - [>Examples]以下の
  [ELTRES EEPROM read/write sample application]
  [ELTRES LPWA sample application]
  [ELTRES standalone sample application]
  から組み込むものを[Y]にします。



- 以上を設定したら、SDK コンフィグの[保存]をクリックしてから、SDK コンフィグタブを閉じます。

ビルド、フラッシュ書き込み

Spresense SDK スタートガイド (IDE 版)4.5 Spresense ボードへの書き込みと動作確認に従って、シリアルポートの設定、ビルド、書き込みを行ってください。

プログラムの実行

ELTRES ボードを装着し、プログラムを書き込んだ Spresense と PC とを USB で接続した状態で、ターミナルプログラムでポートを開きます。VS Code でプロジェクトメニューから[Spresense: ビルドと書き込み]を使用して書き込んだ場合は自動的に VS Code 組み込みのシリアルターミナルが起動します。nuttx (Spresense の OS)のシェル上で help と入力すると使えるコマンド一覧が表示されます。help コマンドに対する応答例を以下に示します。

```
nsh> help
help usage:  help [-v] [<cmd>]

  .          cmp        exit       kill       mv         rmdir      uname
  [          dirname    false      ls         nslookup   set        umount
  ?          date       free       mkdir      poweroff   sleep      unset
  basename   dd         help       mkfatfs    printf     source     usleep
  break      df         hexdump    mkfifo     ps         test       xd
  cat        dmesg      ifconfig   mkrd       pwd        time
  cd         echo       ifdown     mksmartfs  reboot     true
```

```
  cp           exec         ifup         mount        rm           truncate

Builtin Apps:
  eltres_eeprom  eltres_lpwa     nsh              sh
nsh>
```

上記例はコンフィグレーションで[ELTRES EEPROM read/write sample application]および[ELTRES LPWA sample application]を組み込んだ場合です。アプリケーションコマンド(Builtin Apps)として eltres_eeprom と eltres_lpwa が組み込まれていることがわかります。これらのコマンド名をシェルのプロンプトに入力するとプログラムが起動します。
なお、送信動作を行うプログラムの実行前に、後述の ELTRES 送信モジュールの設定の項を参照して ELTRES 送信モジュールの設定を行ってください。

プロジェクトの派生

サンプルアプリケーションのソースコードは、それぞれサンプルに対応するディレクトリに格納されています。
[ELTRES LPWA sample application] ~/spresense/examples/eltres_lpwa
[ELTRES standalone sample application] ~/spresense/examples/eltres_standalone
[ELTRES EEPROM read/write sample application] ~/spresense/examples/eltres_eeprom
それぞれの場所にあるソースコードを直接編集することもできますが、変更や機能追加を行う場合はこれらを複製して独立したプロジェクトとしたほうが管理しやすいでしょう。

以下、例として ELTRES LPWA sample application から派生プロジェクトを作る方法を説明します。
以下のようなディレクトリ構成になることを想定しています。

```
spresense
|-- sdk
|-- examples
      |-- eltres_lpwa (もとのサンプルが置かれたディレクトリ)
|-- eltres_mod
      |-- eltres_lpwa (このディレクトリにソースをコピーし、変更して開発する)
```

- VS Code のエクスプローラペインのコンテキストメニュー[新しいフォルダ...]等を使用して、~/spresense/eltres_mod ディレクトリを作成します。
- ワークスペースセットアップウィザードを起動します。表示されたダイアログで、Spresense SDK パスには Spresense SDK の存在するパスを、作成した ~/spresense/eltres_mod をプロジェクトフォルダーパスに指定して、[作成]をクリックします。
- ワークスペースを作成したら、VS Code の[ファイル]メニューから[名前をつけてワークスペースを保存...]でワークスペースを保存します。保存名称、保存場所は任意です。

- VS Code の[新規ファイル...]などを使用して、プロジェクトディレクトリ ~/spresense/eltres_mod 直下に .sdksubdir ファイルを作成します。.sdksubdir ファイルは空の(内容が無い)ファイルです。
- プロジェクトフォルダー直下の~/spresense/eltres_mod/Makefile ファイルを編集します。ファイル中の
  MENUDESC = “User application”
  の行を、2 行前つまり
  ifeq ($(SDK_VERSION_MAJ),1)
  の前の行にコピー/ペーストし、保存します。
- プロジェクトメニューから[Spresense: アプリケーションのクリーン]を実行します。
- プロジェクトフォルダ~spresense/eltres_mod/以下に ~/spresense/examples/eltres_lpwa フォルダをコピーします。
- コピーした~spresense/eltres_mod/eltres_lpwa ディレクトリ以下のファイルのうち、Makefile, Make.defs, Kconfig 中の以下のシンボルを変更し、保存します。
  EXAMPLES_ELTRES_xxx ⇒ EXAMPLES_ELTRES_xxx_MOD
  CONFIG_EXAMPLES_ELTRES_xxx ⇒ CONFIG_EXAMPLES_ELTRES_xxx_MOD
  (2023 年 8 月の git リポジトリでは、Makefile 中 4 箇所、 Make.defs 中 1 箇所, Kconfig 中 5 箇所が変更になります)。
- プロジェクトメニューから[Spresense: SDK コンフィグ]を選択します。SDK コンフィグのタブが開くまで多少時間がかかることがあります。
- SDK コンフィグのタブが開いたら、タブ内の[新規作成]をクリックして、表示される[New Configuration]ダイアログの内容を変更せずに[OK]を押して、コンフィグレーションの初期化を行います。
- サンプルのコンフィグレーションの際と同様に[> System Type] - [> CXD56xx Package Configuration] - [> Peripheral Support] - [UART0] (SPEXEL を使用する場合)、[> Application Configuration] - [> Spresense SDK] - [> Externals] - [> ELTRES SDK library]および[ELTRES board selection]の設定を行います。
- [> Application Configuration] - [> Spresense SDK] - [> User application] - [ELTRES LPWA sample application]を On にして、SDK コンフィグを保存してください。

以降、プログラムのビルド、Spresense ボードへの書き込み、プログラムの実行についてはサンプルプログラムの場合と同様です。

ELTRES 送信モジュールの設定

ELTRES 送信モジュールの各種設定は、モジュール内蔵の EEPROM にデータを書き込むことによって行います。書き込みは eltres_eeprom アプリケーションによって行います。

eltres_eeprom アプリケーションの使用法は、アプリケーションをパラメータ無しで起動すると表示されます。

送信間隔

ELTRES の通信は、予め設定された間隔で行われます。受信局が送信を期待していないタイミングで送信が行われてもその送信は受信されません。このため、送信端末の通信間隔設定(BurstInterval)は原則として受信局と同一に設定(回線契約で指定されている値)でご使用ください。

ネットワーク設定

CXM150x Configuration Manual において、ネットワーク依存とされている項目についてはネットワークプロバイダ(回線契約者)の指定した値であることが必要です。設定が異なる場合、通信ができないことがあります。回線契約した設定値にてご使用ください。

standalone 設定 (CXM1501AGR のみ)

CXM1501AGR では、GNSS から得られた位置や速度情報、温度や内蔵の AD コンバータへの入力電圧などを送信前に自動的に取得してペイロードに設定することができます(AutoPayload 機能)。これらはユーザーが設定を変更することが可能です。
eltres_standalone サンプルアプリケーションでは、AutoPayload 機能によってペイロード設定を行います。デフォルトのペイロード設定での送信を行う場合、出荷時設定は変更せず、AUTOPLD_COLLECT(0x07A4)に 0 以外の値を書き込みます。使用したい状況にもよりますが、ここでは 5 を設定するものとします。
また、eltres_lpwa サンプルアプリケーションで使用するときには、AutoPayload 機能が有効になっているとアプリケーションが設定するペイロードと競合して期待したペイロードが送出されないことがありますので、AUTOPLD_COLLECT に 0 を書き込んで AutoPayload 機能は無効にしてください。
standalone モードでは、送信間隔が一定以上空く場合、自動的に電力消費を極限まで抑えた DeepSleep モードに入ることもできます。
これらの機能の詳細については CXM150x Configuration Manual および Application Note AN009 を参照してください。

その他、サンプルアプリケーションで使用する機能の設定

ELTRES LPWA sample application 動作に関係する EEPROM 設定は以下のものです。

EEPROM 設定	説明
POW_ENABLE_REMAIN_OFFSET	次回電源 On までの残時間計算のためのオフセット値 CXM150x の電源をオンにするまでの時間計算に使われます
p1Enabled	Periodic1 プロファイルの有効/無効 p1Enable か p2Enable の一方または両方を有効にして

	ください。
p2Enabled	Periodic2 プロファイルの有効/無効 p1Enable か p2Enable の一方または両方を有効にしてください。
evEnabled	Event プロファイルの有効/無効 イベント送信モードを使用する場合は 1 に設定してください。
INT_OUT1 p1INT_OUT1 p2INT_OUT1 evINT_OUT1	LPWA 送信データの更新期限の前、指定時間に INT_OUT1 端子による通知を行います。INT_OUT1 端子は送信が完了すると L になります。設定の詳細は CXM150x Configuration Manual を参照してください。 ELTRES LPWA sample application においては、使用するプロファイルにおいて INT_OUT1 による通知機能が有効になっている必要があります。5 秒以上の設定時間を推奨します。
AUTO_PERIODIC_SELECT	Periodic1/Periodic2 プロファイルをそれぞれの StartTime/EndTime に応じて自動的に切り替えます。 1(自動切り替え)に設定してください。

ELTRES standalone sample application 動作に関係する EEPROM 設定は、INT_OUT1 設定以外の eltres_lpwa のものに加えて以下のものです。

EEPROM 設定	説明
INT_OUT2	CXM150x が UART での送信開始する指定時間(ミリ秒単位)だけ前に通知する信号を発生します。INT_OUT2 端子は一行のデータ送信が完了すると L になります。 ELTRES standalone sample application の動作には INT_OUT2 が 0 以外に設定されている必要があります。 設定値 10 を推奨します。
SM_TOUT	CXM150x が起動後自動的に通常動作モードに遷移するまでの時間を指定します。 外部にマイコンを接続する場合は 0(自動遷移無効)に設定してください。
AUTOPLD_COLLECT	オートペイロードのデータを取得する際のデータ収集開始時間を、データ送信の何秒前に開始するかで設定します。 ELTRES Standalone sample application を使用する場合は 0 以外に設定されている必要があります。 設定値 5 を推奨します。 ELTRES LPWA sample application を使用する場合は 0 に設定してください。

WAKEUP_CTRL	UART インタフェース回路の自動電源オフの有無を設定します 外部にマイコンを接続する場合は 0(自動オフしない)に設定してください。
AUTO_PERIODIC_SELECT PROFILE_SELECT MIN_DSLP_TIME DSLP_BUP	送信プロファイルにあわせて設定してください。 設定については CXM150x Configuration Manual CXM150x Application Note AN009 を参照してください。
AUTOPLD_SRC_SELECT AUTOPLD_LAT_BASE AUTOPLD_LON_BASE AUTOPLD_LAT_RANGE AUTOPLD_LON_RANGE AUTOPLD_LAT_RES AUTOPLD_LON_RES AUTOPLD_HEIGHT_OFFSET AUTOPLD_SRC1_BIT_POS AUTOPLD_SRC2_BIT_POS AUTOPLD_SRC3_BIT_POS AUTOPLD_SRC4_BIT_POS AUTOPLD_SRC5_BIT_WIDTH AUTOPLD_SRC5_BIT_POS AUTOPLD_SRC6_BIT_WIDTH AUTOPLD_SRC6_BIT_POS AUTOPLD_SRC7_BIT_WIDTH AUTOPLD_SRC7_BIT_POS AUTOPLD_SRC8_BIT_POS AUTOPLD_SRC9_BIT_WIDTH AUTOPLD_SRC9_BIT_POS	ペイロードに設定するデータに合わせて設定してください。 設定については CXM150x Configuration Manual, CXM150x Application Note AN009 を参照してください。

便利な追加機能

Spresense SDK には ELTRES サンプルアプリケーション以外にも便利な機能が含まれているのでいくつかご紹介します。機能を追加するには、サンプルやアプリケーションの設定にさらにコンフィグレーションを加えます。

自動起動

通常、プログラムの起動には Spresense にターミナルを接続し、nuttx のプロンプトからアプリケーション名を入力します。しかし、これではプログラムの起動のためには端末の接続が必要になってしまいます。自動起動を使用すると、端末を接続しなくても設定により Spresense の起動後自動的にコマンドを実行することが可能です。
この手順については、Spresense SDK チュートリアル内のアプリケーションの自動起動方法に記載があります。

- プロジェクトメニューから[Spresense: SDK コンフィグ]を起動し、SDK コンフィグで[> Application Configuration] - [> Spresense SDK] - [> System tools] - [Spresense Startup Script]を[Y]にします。

- 自動起動のファイルは romfs 上に配置するので、romfs(spifs)を設定します。SDK コンフィグで[> Application Configuration] - [> NSH Library] - [> Scripting support] - [Custom ROMFS header file path]に ../../system/startup_script/nsh_romfsimg.h を設定します。
- このとき、SDK コンフィグの[> Application Configuration] - [> System Libraries and NSH Add Ons] - [VI Work-Alike Text Editor]を[Y]に設定すると、unix 等でポピュラーな vi エディタ)に似たテキストエディタを nuttx 上で使用することができます。
- SDK コンフィグの保存を行ったら、アプリケーションの作成の場合と同様にビルド、書き込みを行います。
- Spresense の/mnt/spif/以下に自動実行されるスクリプトファイル init.rc を配置します。例えば、起動後 eltres_lpwa を起動する場合は、nuttx シェルに以下のコマンドをあたえることでスクリプトファイルが作成できます。”nsh> “は nuttx シェルのプロンプトです(前述の vi 相当のエディタを組み込んでいるならそれを使用してもファイルを作成できます)。

```
nsh> echo eltres_lpwa > /mnt/spif/init.rc
```

以上の設定によって、システム起動時に eltres_lpwa アプリケーションが起動するようになります。
なお、なんらかの原因で/mnt/spif が存在しない場合は Spresense SDK チュートリアルページの SmartFS フォーマット手順 を参照の上 spif デバイスを再構築してください。先行して Flash 消去 が必要になる場合もあります。

マイクロ SD カードの利用

SPEXEL や Spresense 拡張ボード上のマイクロ SD カードソケットに挿入したマイクロ SD カードを使用することができます。

- SDK コンフィグの[> System Type] - [> CXD56xx Package Configuration] - [> Storage Options] - [SDIO SD Card]を[Y]にします。
- SPEXEL 上のマイクロ SD カードソケットを使用する場合は、SDK コンフィグで[> Application Configuration] - [> Spresense SDK] - [> System tools] - [PMIC Command]も[Y]にしてください。
- SDK コンフィグを保存してアプリケーションをビルド、書込みして Spresense を起動します。拡張ボードにマイクロ SD カードを挿入している場合は SD カードを認識しています。
  SPEXEL のカードソケットを使用する場合は nuttx シェルに以下のコマンドを入力後に SD カードを挿入します。

```
pmic -e GPO5
```

以上で、/mnt/sd0/に SD カードがマウントされ、プログラムからのアクセスが可能になります。
なお、プログラムから SD カードへの書き込みを行った場合、ファイルのクローズ処理を行わない状態でカードの抜き取り、プログラムの強制終了、システムのリセットや電源の切断が行われると書き込んだデータやファイルが失われる場合がありますのでご注意ください。

まとめ

Spresense SDK による ELTRES モジュールボードのサンプルアプリケーションについて説明いたしました。簡単な操作でサンプルアプリケーションを生成できますので、お客様の ELTRES 端末開発に役立てていただければと思います。

参考資料

ELTRES 送信モジュール CXM1501AGR ドキュメント類取得について

CXM1501AGR のデータシート他各種ドキュメントはソニーセミコンダクタソリューションズ(株)の ELTRES 資料ダウンロードページ よりご請求ください。
本ドキュメントでは、
CXM150x Configuration Manual
Application Note AN009
を参照しています。