



技術者向け説明書

UDE Memtool

FLASH/OTP書込みツール — 機能・自動化・AURIX TC3xx UCB書込み

ポジティブワン株式会社

本資料は PLS Development Tools 社の製品資料・アプリケーションノートを基に作成しています。

本書の構成



01

製品概要

UDE Memtoolの位置付け、主な機能、対応デバイス、通信インタフェース、GangProgrammer、ライセンス



02

自動化（バッチ／COM）

バッチモードのコマンドと記述例、COMオートメーションの処理フロー、APIと言語別サンプル



03

AURIX TC3xx UCB書込み

UCBの概要と注意点、UCB設定機能（GUI・API）、ダイレクトUCB書込み（DF_UCBS）



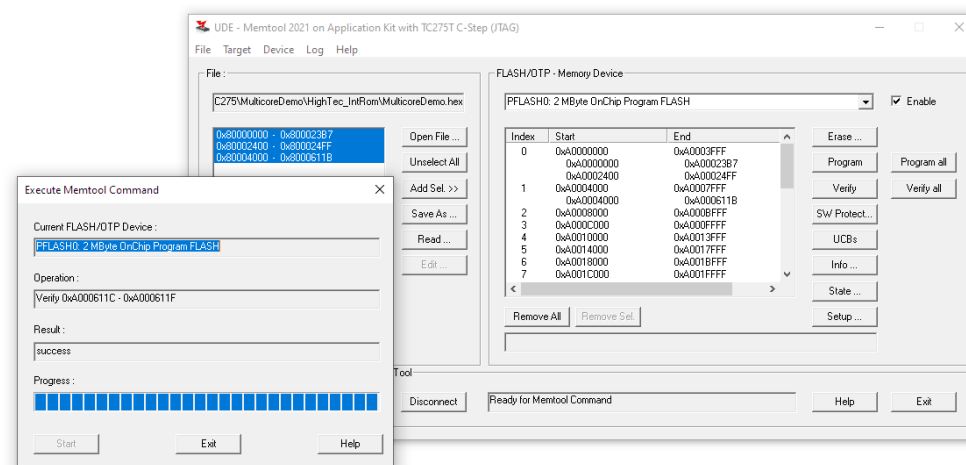
04

運用上の注意

書込み作業のチェックリスト、参照資料、お問い合わせ

UDE Memtoolとは

- オンチップFLASH/OTP、外付けFLASH・EPROMを書き込むためのツール
- UDE（Universal Debug Engine）のアドインとして動作。開発中にIDE内から書き込みでき、メモリマップを認識して必要な時に書き込みアルゴリズムを実行
- 独立したフロントエンド（UDEMemtool.exe）として単体でも使用可能
- 全書き込み機能を標準COMオートメーションで公開。生産ライン・検査システムやスクリプトに組み込み可能
- バッチモードとコマンドラインインタフェースを提供
- 書き込み処理にはターゲットのオンチップRAMを使用。複数のメモリモジュールを扱える



UDE Memtoolのメイン画面と書き込み実行ダイアログ（PLS社資料より）

UDEアドイン

単体GUI

COM API

バッチ/CLI

4つの利用形態を用途に応じて使い分け可能

主な機能（メモリ種別により異なる）



消去

メモリモジュール全体、または選択セクタを消去



ファイル読込

Intel HEX / Motorola S-Record形式に対応



書込み

ファイルの全体または選択部分をメモリへ書込み



ベリファイ

全体または選択部分を現在のメモリ内容と比較



読出し・保存

現在のメモリ内容をリードバックしディスクへ保存



プロテクト

チップ／セクタ保護の設定・解除（オンチップのみ）



UCB / ABM / BMI

UCB、ABM（Alternate Boot Mode）、BMI（Boot Mode Index）ヘッダの処理（一部デバイス）



複数モジュール

1つのターゲットで複数のメモリモジュールを扱える

対応デバイス



車載向けマイコン（オンチップ）

- AURIX / TriCore
- Stellar SR6P・SR6G・SR5E
- S32 Automotive Platform
- RH850
- Power Architecture / PowerPC
- TRAVEO T2G



汎用マイコン・Arm（オンチップ）

- STM32xx
- Arm Cortex-M0 / M3 / M4 / M7
- Cortex-R4
- Cortex-A8 / A9 / A53
- XMC7000 / XMC1000 / XMC4500
- XC2000 / XE166 / XC800



外付けFLASH

- パラレルNOR-FLASH
- シリアルEEPROM（I2C / SPI / SPIFI）
- NAND-FLASH（要相談）

最新の対応デバイス一覧は <https://www.pls-mc.com> を参照してください。開発中のデバイスやその他のデバイスについてはお問い合わせください。

通信インタフェースと製品構成

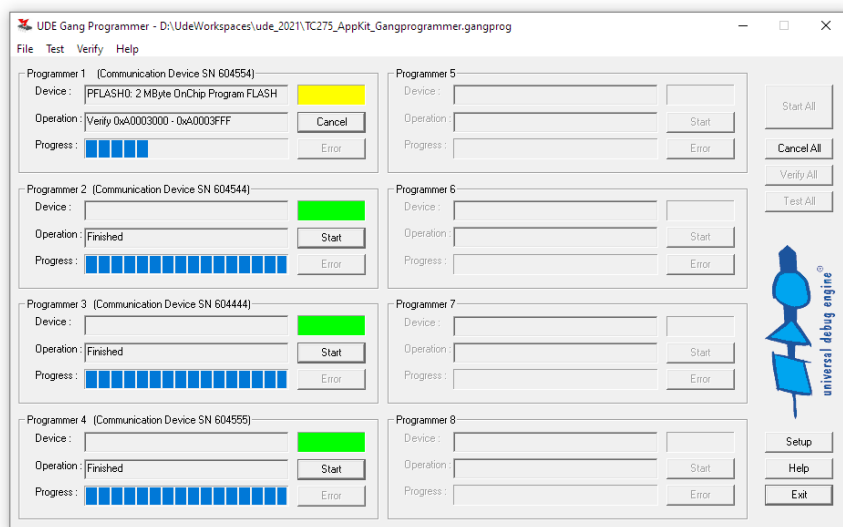
通信インタフェース	対象マイコン	Memtool/UA D2pro	Memtool/UA D2next	Memtool/UA D3+	Memtool/AS C
ASCブートストラップローダ / ASC	AURIX, TriCore, PowerPC, XMC1000/4500, STM32, XE166, XC2000	✓	✓	—	✓ *1
CANブートストラップローダ	AURIX, TriCore, PowerPC, XMC1000/4500, STM32, XE166, XC2000	✓	✓	—	—
K-Line	AURIX, TriCore, PowerPC, XE166, XC2000	✓ *2	✓ *2	—	✓ *2
JTAG / DAP / SWD	AURIX, TriCore, S32, Stellar, PowerPC, Arm, Cortex, RH850, STM32, XE166, XC2000	✓ *3	✓ *3	✓ *3	—
DXCPL (DAP over CAN物理層)	AURIX, TriCore	✓ *3	✓ *3	—	—

アクセスデバイス : UAD2pro / UAD2next / UAD3+ = Universal Access Device、ASC = ホスト側シリアル (RS232)

*1 追加ハードウェア不要 *2 外付けK-Lineコンバータが必要 *3 対応デリバティブは要確認 (<https://www.pls-mc.com>)

アクセスデバイスとホストPC間はUSBまたはEthernetで接続。通信チャネルとしてASC/CANブートストラップローダ、CAN+ASC併用、JTAGのほか、ホストのRS232 (COMx) やK-Lineも使用できます。

GangProgrammerとライセンス体系



GangProgrammer画面（PLS社資料より）

GangProgrammer（量産向け同時書込み）

- 最大8台の書込みステーションを同時に管理し、書込み時間を短縮
- アクセスデバイスはUSB / IEEE1394 / Ethernet でハブ経由接続
- ターゲットごとに個別のターゲット構成で管理

ライセンス

1シート1ライセンス

複数のマイコンアーキテクチャにはアーキテクチャアップグレードが必要

1年間の保守・サポート

ライセンスに含まれ、毎年更新可能

アダプテーション1件込み

Memtool構成フォームで顧客ターゲット向け設定を定義。以降は顧客自身で追加作成可能

その他のモデル

部門／全社ライセンス、OEMライセンス、フルカスタム版は要相談

自動化の3つの手段



バッチモード

- コマンドを並べたテキストファイルを実行
- GUIメニュー「File」→「Run Batch」
- 起動時引数で指定：Memtool[.exe] file
- 定型的な書込み手順の再現に最適



コマンドライン（ MemtoolCli.exe）

- コンソールから書込みを実行
- ダイレクトUCB書込みにも対応
- 生産ラインのスク립トからの呼出しに



COMオートメーション

- ProgID "UDE.Memtool" のCOMオブジェクト
- JScript / VBScript / Perl / Python / C# 等から制御
- ベリファイ差分数などの結果を取得でき、UCB設定APIも利用可能
- 生産・検査システムへの統合に

※ MemtoolCli.exe の詳細なオプションは、ご使用のUDEバージョンのマニュアルを参照してください。

バッチモード：コマンド一覧

コマンド	説明
<code>connect</code>	ターゲットシステムへ接続する。既に接続中の場合は、先に切断しておく必要がある
<code>disconnect</code>	ターゲットシステムとの接続を切断する
<code>open_file file [, [+ -]<offset>]</code>	HEXファイルを開き、全セクションを左側リストに表示する。fileは絶対パスで指定。offsetを指定するとHEXファイル内の全アドレスに加算される
<code>select_all_sections</code>	左側リストの全セクションを選択する
<code>add_selected_sections</code>	左側リストで選択したセクションを右側リスト（書込み対象）へ追加する
<code>program</code>	FLASHデバイスへの書込みを開始する



実行方法：GUIの「File」→「Run Batch」でバッチファイルを選択、またはコマンドラインで Memtool[.exe] の起動引数にバッチファイルを指定

バッチモード：記述例（4ファイルを一括書込み）

```
connect
open_file c:¥Data¥starter.hex
select_all_sections
add_selected_sections
open_file c:¥Data¥firmware1.hex
select_all_sections
add_selected_sections
open_file c:¥Data¥firmware2.hex , +40000h
select_all_sections
add_selected_sections
open_file c:¥Data¥sernum.hex
select_all_sections
add_selected_sections
program
```

1

接続

connect でターゲットに接続

2

セクション蓄積

open_file → select_all_sections → add_selected_sections
を繰り返し、書込み対象を右側リストに積み上げる

3

オフセット指定

firmware2.hex は全アドレスに +40000h を加算して配置

4

一括書込み

最後に program を1回実行し、蓄積した全セクションを書込み

ポイント：ファイルは必ず絶対パスで指定します。

COMオートメーション：基本処理フロー

UDEはMicrosoft COM技術に基づいており、COM（ActiveX）を利用できるクライアントやスクリプトホストからUDE Memtoolを遠隔操作できます。



ベリファイの戻り値は差分数です。0 以外の場合は書込み異常として扱ってください。

主なAPI（UDE.Memtool オブジェクト）

メソッド / プロパティ	引数	機能
Version	—（プロパティ）	Memtoolのバージョンを取得
CreateLogFile(path, level)	ログファイルのパス、ログレベル（サンプルでは 50）	ファイルへのログ出力を有効化。問題の解析に使用
Init(cfgFile)	ターゲット構成ファイル（.cfg）	ターゲット構成ファイルでコンポーネントを初期化
ConnectTarget()	—	ターゲットシステムに接続
OpenFile(hexFile, flag)	HEXファイルのパス、ブール値（サンプルでは true / 1）	ターゲットのアプリケーションファイルを読み
ProgramSections()	—	有効な全FLASHデバイスへ、読み込み済みの全セクションを書込み
VerifySections(arg)	文字列（サンプルでは ""）	書き込み済みセクションを検証し、差分数を返す
DisconnectTarget()	—	ターゲットとの接続を切断
Cleanup()	—	コンポーネントの後処理（リソース解放）
GetFlashMod(name)	モジュール名（例 "PFLASH0"）	FLASHモジュールオブジェクトを取得（UCB設定などで使用）

※ 本表はPLS社アプリケーションノートのサンプルに登場するAPIをまとめたものです。全APIの仕様はUDEのマニュアルを参照してください。

サンプル：Pythonによる書込み・ベリファイ

```
import os
import win32com.client

WorkDir = os.path.abspath(".")
CfgFile = WorkDir + "¥¥TriBoard_TC27x_dap.cfg"
HexFile = WorkDir + "¥¥MulticoreDemo.hex"
LogFile = WorkDir + "¥¥Memtool.log"

Memtool = win32com.client.Dispatch("UDE.Memtool")
print("Version: " + Memtool.Version)

Memtool.CreateLogFile(LogFile, 50)
Memtool.Init(CfgFile)
Memtool.ConnectTarget()
Memtool.OpenFile(HexFile, True)
Memtool.ProgramSections()
Diffs = Memtool.VerifySections("")
print(str(Diffs) + " differences found")

Memtool.DisconnectTarget()
Memtool.Cleanup()
```

動作環境と使い方

- Windows環境、UDE Memtoolがインストール済みであること
- Pythonの win32com (pywin32パッケージ) を使用
- 実行 : python MemtoolAutomationDemo.py
- .cfg はUDEで作成したターゲット構成ファイル。自社ボードの構成に置き換える
- ログレベル 50 でログファイルを出力。トラブル時の解析に活用

量産で使う場合のヒント

Diffs が 0 以外なら異常終了コードを返し、ライン側で不合格判定できるようにすると運用しやすくなります。

言語別サンプルの違い

言語	実行方法	オブジェクト生成	OpenFile 第2引数
JScript	<code>cscript MemtoolAutomationDemo.js</code>	<code>new ActiveXObject("UDE.Memtool")</code>	<code>true</code>
VBScript	<code>cscript MemtoolAutomationDemo.vbs</code>	<code>CreateObject("UDE.Memtool")</code>	<code>true</code>
Perl	<code>perl MemtoolAutomationDemo.pl</code>	<code>new Win32::OLE(\$ProgId)</code>	<code>1</code>
Python	<code>python MemtoolAutomationDemo.py</code>	<code>win32com.client.Dispatch("UDE.Memtool")</code>	<code>True</code>
C#	ビルドして実行	<code>new UDELib.UDEMemtool()</code>	<code>1</code>



C#

プロジェクトの参照に UDENetCOMLib.UDELib を追加し、`using UDELib;` を宣言。VerifySections は int を返す



VBScript

戻り値を使わない呼出しは括弧なしで記述（例：
`Memtool.CreateLogFile LogFile, 50`）



Perl

`use Win32::OLE;` を使用。生成失敗時は `die` でエラー終了させる

UCB (User Configuration Block) とは

- AURIX TC3xxファミリが持つ、構成情報を格納する専用データFLASHセクタ
- オンチップFLASHのリード/ライトプロテクト設定、起動動作、デバッグインタフェースのロックなどを設定
- 品種ごとのUCBの種類と内容は、各デバイスのユーザズマニュアルを参照

UCBの例 (TC38xのUCBハンドラ画面のタブ)

UCB_BMHD0	UCB_BMHD1	UCB_BMHD2	UCB_BMHD3
UCB_PFLASH	UCB_DFLASH	UCB_DBG	UCB_HSM
UCB_HSMCOTP0	UCB_HSMCOTP1	UCB_OTP	UCB_SWAP



重要な警告

UCBに不正な情報を書き込んだり、UCBのFLASHセクタを完全に消去したりすると、マイコンがオンチップのスタートアップソフトウェアを実行できなくなる場合があります。この状態になるとUDEで接続できず、UCB構成を修正することもできません。必ず正しいUCBデータを使用してください。



ヒント：反映タイミング

UCBの変更はリセット後に有効になります。Memtoolでは通常、ターゲットを切断・再接続してから結果を確認します。

UCB書込みの2つの方法



① UCB設定機能

開発者向け

- 設定ダイアログでUCB内容を作成して書込み
- UCBが許す範囲で既存構成の変更・削除が可能
- 不正な構成を防ぐチェック機能を内蔵
- 検証済みの構成をHEXファイルへエクスポート
- COMオブジェクトモデルからも利用可能（スクリプト対応）



② ダイレクトUCB書込み

量産向け

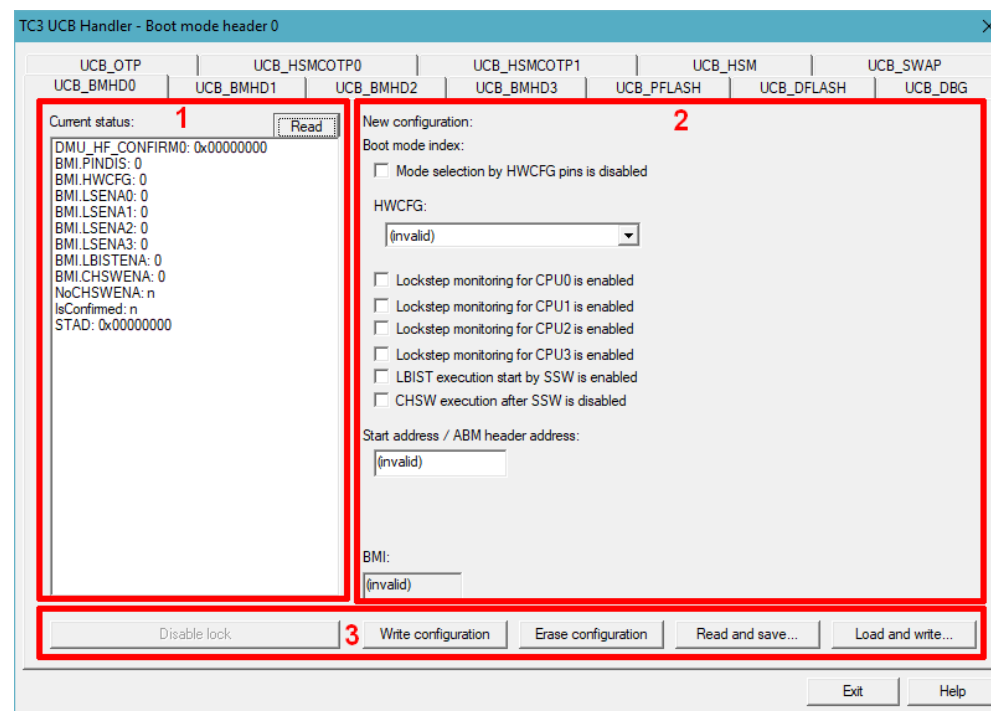
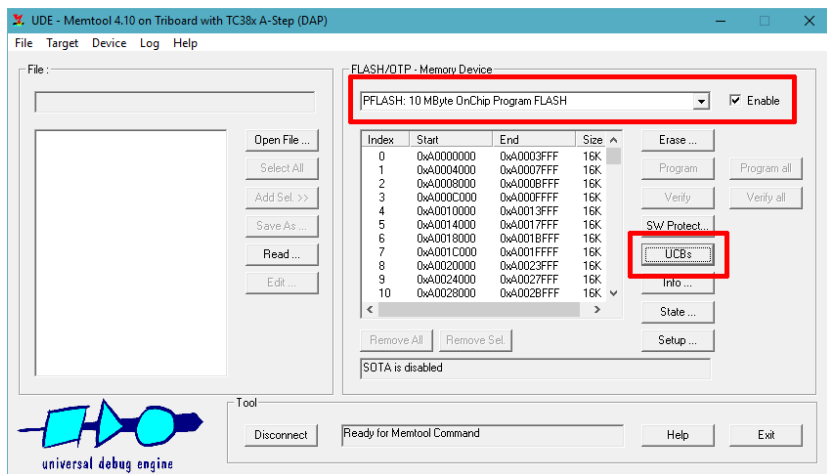
- 検証済みHEXを通常のFLASH（PFLASH等）と同様に読込・書込み
- コマンドラインツール MemtoolCli.exe が対応
- 専用メモリデバイス DF_UCBS はデフォルト無効。使用前に有効化が必要
- UCB全体をカバーするデータのみ受付（部分的なHEXは拒否）
- 無効な確認コードなど不正データの書込みを防ぐチェックあり

推奨フロー：開発段階で①により構成を作成・検証 → HEXへエクスポート → 量産では②で同一HEXを書込み

UCB設定機能（GUI）の使い方

UCB書込み機能はPFLASHメモリデバイスに紐付いています。

- 1 「FLASH/OTP - Memory Device」で PFLASH を選択し、Enable をチェック
- 2 UCBs ボタンを押して TC3 UCB Handler ダイアログを開く
- 3 対象UCBのタブを選択し、構成を準備・実行



1 現在の状態

UCBが影響するレジスタ・ビットフィールドの現状

2 新しい構成

新しいUCB構成を準備

3 操作ボタン

UCB処理を実行

UCBハンドラの操作ボタン



Disable lock

パスワード保護されたUCBのロックを解除。
256bit (32bit×8) のパスワードを指定しない限り、内容の参照も変更もできない



Write configuration

新しい構成をUCBへ書込み。パスワード保護付きのUCBでは新しい256bitパスワードの指定を求められる (UCBを確定状態にするオプションあり)



Erase configuration

実際には消去せず、空の構成で上書きする。
OTPのUCBは消去できないため、この機能は使用不可



Read and save...

UCBの現在の内容を読み出し、HEXファイルに保存



Load and write...

HEXファイルを読み込み、その内容をUCBへ書込み

パスワード入力形式：「1,2,3,4,5,6,7,8」のように8個の32bit値をカンマ区切りで入力 (PW0=0x00000001 … PW7=0x00000008)。Check ボタンで展開結果を確認できます。

UCB設定の自動化 (C# API例)

```
// access FlashModExt object
UDELib.DIUDEFlashMod FlashMod =
    m_Memtool.GetFlashMod("PFLASH0");
UDELib.DIUDEFlashModExt FlashModExt =
    FlashMod.GetExtension("UCB_DFLASH");

if (!FlashModExt.ReadStatus()) return -1;

if (FlashModExt.Locked) {
    if (!FlashModExt.DisableLock("1,2,3,4,5,6,7,8"))
        return -1;
}

FlashModExt.PrepareNewConfig();
var Config = FlashModExt.Config;
Config.SetItem("PROCONRAM.RAMINSEL1", 1);
Config.SetItem("PROCONRAM.RAMINSEL2", 1);

if (!FlashModExt.WriteConfig()) return -1;
```

1 GetFlashMod / GetExtension

PFLASH0モジュールから対象UCB（例 UCB_DFLASH）の拡張オブジェクトを取得

2 ReadStatus()

ターゲットから現在の状態を読み出し

3 Locked / DisableLock()

ロック中なら256bitパスワードで解除

4 PrepareNewConfig()

現在の状態を基に新しい構成を準備

5 Config.SetItem()

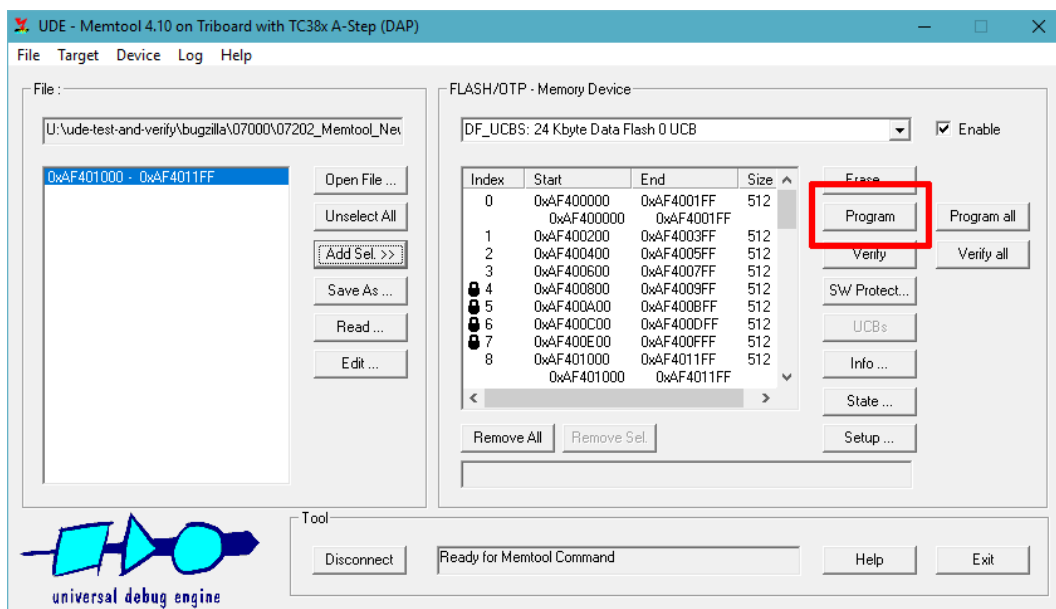
"レジスタ.ビットフィールド" 名で項目を変更

6 WriteConfig()

新しい構成をターゲットへ書込み

ReadStatus / DisableLock / WriteConfig は成否を返すため、失敗時は処理を中断します。

ダイレクトUCB書込み（DF_UCBS）



DF_UCBS を有効化し、HEXを追加して Program（PLS社資料より）

MemtoolCli.exe でも同じ方式で書込みできます。

- 1 メモリデバイスで DF_UCBS（例：24 Kbyte Data Flash 0 UCB）を選択し Enable をチェック
- 2 各UCBは個別のFLASHセクタ（例：0xAF400000～、512バイト単位）として表示される
- 3 Open File でHEXを開き、Add Sel. で書込み対象へ追加（複数ファイル可）
- 4 Program で書込み、Verify で確認

注意点

- 鍵アイコンのUCBはユーザーが書き込めません
- Erase機能はありません。消去したい場合は空のUCB構成を含むHEXを書き込みます
- UCB全体をカバーしないHEXは拒否されます

書込み作業チェックリスト



作業前

- ターゲット構成ファイル (.cfg) が対象ボード・通信IFに合っているか
- HEX/S-Recordのアドレス配置とオフセット指定を確認
- バッチのファイルパスは絶対パスか
- UCBデータはUCB設定機能で作成・検証済みのものか



作業中

- ログファイル出力 (CreateLogFile) を有効にしておく
- 接続済みの状態で connect を重ねて実行しない
- DF_UCBS は必要な時だけ有効化する
- UCBの完全消去や部分書込みを行わない



作業後

- ベリファイの差分数が 0 であることを確認
- UCB変更後は切断・再接続（リセット）して反映を確認
- DisconnectTarget() と Cleanup() で確実に終了
- 書込み結果とログを記録・保管

参照資料とお問い合わせ

本書の基になったPLS社資料

- UDE FLASH Programming データシート (2025_1126)
- Application Note : UDE Memtool Programming in Batch-Mode (2023_1114)
- Application Note : UDEMemtool Automation Basics (2023_1114)
- Application Note : Programming AURIX TC3xx UCBs with UDE Memtool (2025_0512)



ポジティブワン株式会社

UDE Memtoolの導入、ライセンス、ターゲット構成、自動化やUCB書込みに関するご相談は当社までお問い合わせください。

製品元 : PLS Programmierbare Logik & Systeme GmbH (<https://www.pls-mc.com>)

UDE、Universal Debug Engine はPLS社の登録商標です。AURIX はInfineon Technologies AGの商標です。その他の製品名は各社の商標です。本資料の内容は予告なく変更される場合があります。