



# 12ステップで作る 組み込みOS自作入門 KOZOS用ユティリティ kz\_h8write

## kz\_h8write

kz\_h8write は、12ステップで作る組み込みOS自作入門(KOZOS)で使用している H8 ボード用に作られた H8 フラッシュ書き込みプログラムです。著名な既存書き込みプログラムにおいて、書き込みに失敗する現象が多数報告されています。この書き込みプログラムの不具合を調査した上で新たに作られたのが kz\_h8write です。

## kz\_h8write の特徴

- 著名な既存ツールの書き込み失敗問題を解消した KOZOS 専用 H8 フラッシュ書き込みプログラム。
- Linux、Windows、Mac OS に対応。
- 書き込みステップが見える動作表示出力機能付き。
- MIT ライセンスを採用し、商用、非商用を問わず自由に再利用可能。

## 著名な既存書き込みプログラムの不具合

著名な既存書き込みプログラムの不具合の一つは、ターゲットデバイスにビットレートを自動検出させる過程にあります。ホストから送信したコードに対して、ビットレートの検出が完了した時点でターゲットからある返答が返ってきます。本来であれば、この返答を検出した時点でビットレートを自動検出させるためのコードの送出手を止めなければなりません。



著名な既存書き込みプログラムの場合、自動検出用のコードを送りすぎているために、ターゲットデバイスが期待する次の動作と異なる入力を与えてしまっているのです。上記動作だけを修正する事も考えましたが、KOZOS はフルスクラッチです。だったら書き込みプログラムも自作してしまえ！という事でフルスクラッチで実装しました。

## プロジェクトページ

プロジェクトは sourceforge で管理しています。

<http://sourceforge.jp/projects/kz-h8write/>



# 12 ステップで作る 組み込み OS 自作入門 KOZOS 用ユーティリティ kz\_xmodem

## kz\_xmodem

---

12 ステップで作る組み込み OS 自作入門(KOZOS)の第 8 ステップ以降では、ROM にブートローダ(kzload)を書き込み、ブートローダ経由で OS を RAM に流し込んで動作させるようになっています。転送プロトコルには XMODEM を採用していますが、複数のプラットフォーム上に存在する汎用ツールの多くが、何ステップかの操作をした後でようやく転送が開始されるような仕組みになっています。また、環境やタイミングに依存して、汎用ツールと KOZOS ブートローダの組み合わせでうまく転送できない事もあるようです。

kz\_xmodem は、上記の「複数の操作を段階的に行なう事の煩雑さ」や「転送に失敗する事の面倒さ」などを取り除く事を目的に、設計実装された KOZOS 専用 XMODEM プログラムです。

## kz\_xmodem の特徴

---

- KOZOS ブートローダの動作に合わせて設計した KOZOS 専用 XMODEM プログラム。
- Linux、Windows、Mac OS に対応。
- KOZOS ブートローダに対して内部で load 状態に自動遷移。
- ターゲットをリセットして kz\_xmodem を実行するだけで転送完了。
- コマンドプログラムなので Makefile からの自動実行なども可能。
- MIT ライセンスを採用し、商用、非商用を問わず自由に再利用可能。

## 汎用ツールと kz\_xmodem の比較

---

### 汎用ツールの場合

---

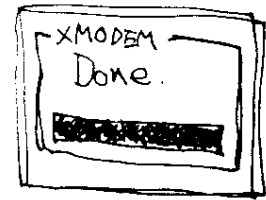
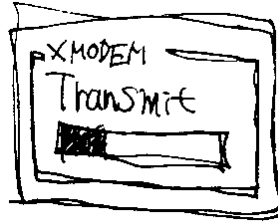
汎用ツールの場合、操作は大まかにわけて 3 段階必要です。

まず、KOZOS ブートローダを load 状態にします。

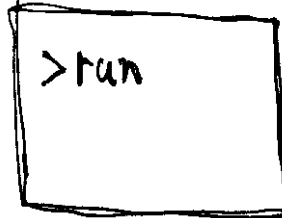
```
>load
```

次に XMODEM 転送の為にファイルを指定して転送を開始します。

ツールによっては、ファイル名の入力などに手間取ると期待したような動作にならない事があるようです。うーん。ここは気合いを入れて操作したいところではありません。



転送が正常に終了したら K0Z0S ブートローダのプロンプトに戻って run コマンドを実行します。

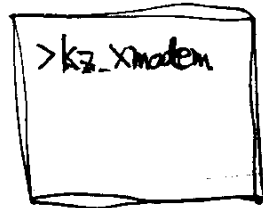


上記のように汎用ツールの場合、ツールの中の画面をいちいち行ったり来たりしなければなりません。OSの動作確認をしたくてうずうずしている時に、色々な操作が伴いととても面倒です。

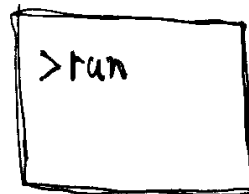
### kz\_xmodem の場合

kz\_xmodem を使うと、シンプルな2つのステップでOSの動作確認まで進めます。

kz\_xmodem に、書き込み対象ファイル名と使用するシリアルポート名を与えて実行します。



書き込みが完了したら K0Z0S ブートローダにシリアルコンソールを使って接続します。後は run コマンドを実行するだけ。



kz\_xmodem はコマンドツールですから、Makefile でビルド後に呼び出すようにすれば自動ダウンロードが可能です。要するに、ビルドした後でOSの動作確認に至るまでが make 一発で一気にできてしまうのです。従来のようにわざわざ別のツールを立ち上げてからアレコレ操作する必要はありません。

### プロジェクトページ

プロジェクトは sourceforge で管理しています。

<http://sourceforge.jp/projects/kz-xmodem/>



**CuBeatSystems**