ESP32 + FluidNC + Raspberry Pi 4B + CNCjs レシピ本

文責: 小木勇輝

2025 年 8 月 16 日

目 次

1　まえがき

本書は、つくば鳥人間の会（TBW）向けに筆者（24 代 小木勇輝）が制作した CNC システム電装系のレシピ本である。CNC は本質的に危険で、制御系も難解である。しかし、その構造や仕組みを理解する利点は大きく、本書がその一助となることを期待する。

本書を読むことで、シングルボードコンピュータ（Raspberry Pi など）上で CNC コントローラ UI（CNCjs）を動作させ、マイコン（ESP32 など）でステッピングモーター・ドライバを駆動できるようになる。

想定読者は高校卒業直後の初学者とする。わからない語は末尾の用語集を参照し、それでも不明な点は適宜 Google 検索や ChatGPT を用いて補うこと。

なお、本書の LaTeX ソースコードは `https://github.com/nilpe/CNCrecipe` に存在する。適宜参照のこと。

2　概念設計図

本章では、Raspberry Pi 4B 上で CNCjs を動作させ、USB 経由で ESP32（FluidNC）を直接制御する最小構成の全体像を示す。

2.1　全体構成

Raspberry Pi 4B 上で CNCjs を起動し、USB 接続された ESP32（FluidNC）に対し G-code を送信する。ESP32 はモーター制御信号（STEP ／ DIR ／ EN）をステッピングドライバへ出力し、ドライバがステッピングモーターを駆動する。（図 1）

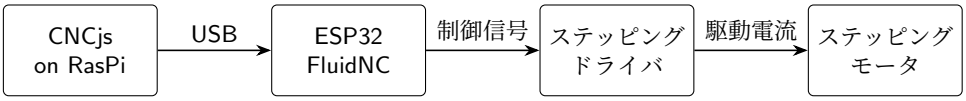

図 1: 本 CNC の概念図

2.2　必要構成要素（最小）

以下は最小構成の一例である。詳細な選定基準は末尾「おまけ＞選定基準一覧」を参照。

- シングルボードコンピュータ（例：Raspberry Pi 4B 4 GB）… CNCjs 動作用

- ESP32 ボード … FluidNC を動作
- ステッピングドライバ（例：TB6600 系、3 A 級、4 線バイポーラ対応）
- ステッピングモーター（NEMA17 ／ 23 対応品など）
- スピンドルモーター（ER25 コレット対応を推奨）
- 電源（24 V 系：ステッピング用、スピンドル用電源：DC もしくは VFD、5 V 系：Raspberry Pi ／ ESP32）
- USB ケーブル（Raspberry Pi と ESP32 の接続用）

2.3 信号と電力の流れ

- 信号経路：Raspberry Pi（CNCjs） → ESP32（FluidNC） → ステッピングドライバ → モーター
- 電力系統：24 V 系（モーター／ドライバ）、スピンドル用（独立系統）、5 V 系（Raspberry Pi ／ ESP32）

2.4 ソフトウェア構成

- Raspberry Pi OS（64-bit）：Linux ベースのデスクトップ OS。GUI 環境があり、扱いやすい。
- CNCjs：ブラウザベースの CNC 操作 UI。Raspberry Pi 上で動作。
- FluidNC：ESP32 上で動作する CNC 制御ファームウェア。G-code を解釈してモーション制御信号を生成する。

2.5 最低限調べておいてほしいこと

Linux の基礎 ファイルシステム、ユーザーと権限、パッケージ管理（`apt` など）、CLI 操作。

Raspberry Pi の基礎 ARM 系 CPU、シングルボードコンピュータの制約、GPIO ／ USB の基本。

高校電磁気の復習 オームの法則 $V = IR$、抵抗・コイル・コンデンサ、電力 $P = VI$、直列／並列、誘導起電力の概念。

ステッピングモーター ステップ角、マイクロステップ、保持トルク、脱調、STEP ／ DIR と回転量の関係。

CNC の事故例 巻き込まれ、工具破損の飛散、固定不良、誤動作衝突などの危険要因と対策。非常停止（E-Stop）、安全カバーの有効性。

2.6　安全留意事項（最重要）

- モーター・スピンドル動作中は可動部へ手や物を近づけない。長髪は束ね、ゆったりした服や手袋は着用しない。
- 工具交換時は電源断を徹底。
- 電源投入前に配線の正誤と固定状態を確認。各系統に適切なヒューズを設ける。
- **Z プローブ未搭載時**は Z=0 設定ミスによる突き刺しに注意。初動は必ず浅切込みまたは空中動作で確認する。

3　セットアップ編

3.1　ESP32 ボードの準備とファーム書き込み

CNC 用モーションコントローラとして ESP32 を用い、ファームウェアは FluidNC [1] を採用する。初期導入は PC 上の Google Chrome から FluidNC 公式 Web Installer [2] を使用（Web Serial API のため Chrome 前提 [3]）。

手順概観

1. ESP32 を USB で手元の PC に接続する（給電は PC 側／基板側の二冗長系を推奨）。
2. Chrome で FluidNC Web Installer を開き、Connect で該当シリアルポートを選択。
3. Install から Latest ビルド（Wi-Fi 有効版）を選択して書き込む。[1]
4. 書き込み完了後、**Installer の「Config」または「Files」**から `config.yaml` を新規作成。基本パラメータは実機を見ながら対話的に調整する（エディタの生 YAML 直書きより Web UI 編集・保存を推奨）。

GPIO 選定の注意点　STEP ／ DIR ／ EN のピン割付では、**出力に使ってよい GPIO** のみを選ぶ。以下は避ける [4, 5]。

- ブートストラッピング用ピン：GPIO 0, 2, 4, 5, 12, 15（特に GPIO12 はブート時電圧でフラッシュ電圧選択に関与）
- フラッシュ接続ピン：GPIO 6–11
- 入力専用ピン：GPIO 34–39

[1] 運用では Wi-Fi は使用しない。導入・更新時のみ有効化の方針。

- UART TX/RX（GPIO 1/3）は原則避ける

必ず最新の資料でピン機能を確認してから割付を決定すること。

3.2 FluidNC の基本設定（Config 欄での方針）

本書の数値を固定値として信用せず、**実機を見ながら Config 欄で対話的に調整**する [6, 7]。理由は、スクリューピッチ、マイクロステップ、剛性、負荷、ドライバ仕様といった**機械固有要素**が設定値を大きく左右するためである。

設定の流れ（対話的チューニング）

1. 軸構成（X ／ Y ／ Z ／ A、スレーブ軸の有無）を定義。
2. 各軸の**ピン割付**を決定（使用可能 GPIO 制約を満たすように）。
3. **steps_per_mm**（例：320 steps/mm）は実測で校正。測長と微調整を繰り返す。
4. **max_rate** と **acceleration** は低めから開始し、脱調・共振・発熱を観察しつつ段階的に引き上げる。
5. 方向反転やリミット極性は実機挙動に合わせて修正。

補足：Config は Web UI から保存する前提。外部エディタでの貼付は極力避ける。

3.3 Raspberry Pi 4B への CNCjs 導入（デスクトップ OS + 公式手順）

CNCjs[8] は Web ベースの G-code 送受信／ UI であり、Raspberry Pi 4B で常用する。OS は **Raspberry Pi OS（64-bit, Desktop 版）** [9] を用い、**運用は有線（USB ／有線 LAN）**とする。**突然の電源断（いわゆる「電プチ」）対策**として Overlay File System の有効化を推奨 [10, 11]。

OS インストール（Raspberry Pi Imager）

1. PC で Raspberry Pi Imager を起動し [12]、OS は **Raspberry Pi OS (64-bit) with desktop** を選ぶ。
2. ホスト名／ロケール、**SSH 有効化、Wi-Fi 設定（導入時のみ使用）**などを事前設定して書き込む [13]。

3. 初回起動後、必要なパッケージを更新する。

Listing 1: 初期パッケージ導入（APT）

```
# Raspberry Pi OS の更新
sudo apt update
sudo apt -y full-upgrade

# 導入に使うツール群
sudo apt -y install git curl build-essential

# Node.js（Bookworm 既定の nodejs でも可。必要に応じて NodeSource を利用）
sudo apt -y install nodejs npm

# Yarn（必要に応じて）
sudo npm -g install yarn
```

CNCjs の導入（公式手順を参照）

```
sudo npm install -g cncjs@latest --unsafe-perm
```

CNCjs は公式インストール手順に従う [14, 15, 16]。Node.js は OS 付属版または NodeSource 提供版を利用できる [17, 18]。常駐化には systemd サービス化を行う [19]。応用編となるので詳細は割愛。

電プチ対策（Overlay File System）

Raspberry Pi OS は `raspi-config` から **Overlay File System** [10, 11] を有効化できる。運用中は rootfs を実質 Read-Only 化し、変更は RAM 上に乗せる。設定変更や更新時のみ一時的に無効化して再起動する運用にする。

Listing 2: OverlayFS の有効化

```
# 対話 UI
sudo raspi-config
# Performance Options -> Overlay File System -> Enable

# 非対話（対応バージョンのみ）
sudo raspi-config nonint enable_overlayfs
sudo reboot
```

推奨フロー

1. 通常運用：OverlayFS 有効
2. 設定変更：OverlayFS 無効化 → 再起動 → 変更反映
3. 変更確定：OverlayFS 再有効化 → 再起動

起動・アクセス

- ターミナルから `cncjs` コマンドで起動。
- ブラウザで `http://<raspi-ip>:8080/` にアクセス（Chrome 推奨）。
- ESP32 は USB 接続し、CNCjs のポート選択で `/dev/ttyUSB*` または `/dev/ttyACM*` を選ぶ。

専用のスクリプトを組んでも良い。

4　機械構築・接続編

本章では、本 CNC 制御システムの機械構築と配線について述べる。

4.1　必要部品と仕様確認

本システムは制御系と駆動系を電気的に分離する。制御系には ESP32、ステッピングドライバ、Raspberry Pi 4B（CNCjs 実行）、制御系 5 V 電源を用いる。駆動系にはステッピングモーターと伝達機構（リードスクリュー）、12 V to 24 V のモーター電源を用いる。モーター仕様（トルク、定格電流、保持トルク）は適宜相談して決定する。安全系には非常停止スイッチとヒューズを備える。

現状、スピンドルは制御系と完全分離され、独立電源で運用されている。リミットスイッチは物理設置済みだが未接続、Z プローブは未搭載である。

4.2　配線計画と電源構成

配線計画では「系統分離」と「耐環境性」を重視する。モーター電源、制御系電源、スピンドル電源は物理分離し、ノイズの回り込みを抑える。屋外設置を想定し、耐候性のある屋外用ケーブル（VCT 等、耐 UV シースなど）を使用する。信号線とモーター線は別ルートで引き回し、筐体導入も分離する。整備性・信頼性の観点から、コネクタ点数は可能な限り削減する。

4.3　ステッピングモーターとドライバ接続

モーターは 4 線式バイポーラで、**A+/A −/B+/B − の巻線端子**をドライバに接続する。配線色はメーカーで異なるため、付属資料または導通試験で確認する。片側相の極性を反転すると回転方向が逆転する。

ドライバには電源入力（例：24 V）、モーター出力端子に加え、STEP ／ DIR ／ EN の入力端子がある。多くの TB6600 [20] ブレイクアウトボードは

5V 駆動のフォトカプラ入力であり、**ESP32 の 3.3 V 直結では確実に動作しない可能性のある個体**がある点に注意する。配線方針は必ずドライバ仕様に合わせること。

マイクロステップ設定は DIP スイッチで行い、機械分解能と加工速度に応じて選択する（例：1/32 ステップ）。電流制限も DIP で設定し、モーター定格の 80 % to 100 % 目安で開始する。

4.4 リミットスイッチ

現状は未接続のため、FluidNC ではソフトリミットまたは手動位置決めで運用する。将来導入する場合は **NC（常閉）接点**を用い、断線検知性を高める。配線は屋外用シールドケーブルとし、ノイズ対策としてツイストペアを推奨。

4.5 Z プローブ

現状は未搭載。ツール長補正は手動で行い、Z 原点の設定ミスによる事故を防ぐため初回は必ず浅切込みまたは空中動作で検証する。

4.6 スピンドル

現状は制御系から完全分離し、独立電源で運用する。将来的に制御を導入する場合、ESP32 から PWM を出力し、適切な変換回路を介して可変電源のアナログ入力へ与える。この際、**光絶縁**を推奨する。

4.7 非常停止回路

非常停止（E-Stop）は、ステッピングドライバの `EN` を用いて押下時に**即時に駆動無効**とする。加えて、**主電源遮断**（安全リレーや自己保持回路）を組み合わせると安全性が高い。スイッチは **B 接点（NC）**、**IP65 以上**を推奨。

4.8 ノイズ対策と配線整理

モーター線と信号線は距離を取り、交差は直角で行う。必要に応じてシールドケーブルやフェライトコアを追加する。配線はケーブルクランプや配線ダクトで固定し、振動断線を防止する。

4.9　接続テストと初動確認

1. CNCjs のジョグで各軸が指令方向へ動くか確認（Y アームは外した状態が安全）。
2. 回転方向が逆なら `DIR` 極性または相線の極性を修正。
3. リミット未接続前提で安全範囲内を確認。
4. スピンドルは別系統で安全に起動確認。

4.10　トラブルシュート

動かない場合は、(1) ドライバ入力電圧、(2) フォトカプラ入力の電圧レベル、(3) ESP32 の GPIO 出力状態の順で点検する。ノイズ疑いではシールド強化や配線経路見直しを行う。オシロスコープが望ましいが、電圧計でも基本確認は可能。

5　制御パラメータ

CNC の動作精度・加工効率は、制御パラメータの適切化に大きく依存する。本節では FluidNC 環境における主要パラメータの設定方針を述べる。

5.1　ステップパルスと加速度設定

ステッピングモーターはパルス幅と周期に同期して回転する。パルス幅が短すぎると取りこぼし、長すぎると最高速度低下や誤作動の恐れがある。初期はドライバ推奨値を用い、安定を確認してから調整する（多くのケースで初期値で問題ない）。

加速度は停止から目標速度までの増加率である。過大では脱調、過小では加工時間が増える。低めから開始し、問題が出ない範囲で徐々に引き上げる。

5.2　機械原点とワーク原点

機械原点（Machine Zero）は機械の絶対座標基準で、通常はリミット到達位置を原点とする。電源投入後に原点復帰（homing）して機械座標を確定させるのが基本だが、**現状はリミット未接続のため homing は行わず、ワーク原点のみで運用**する。

ワーク原点（Work Zero）は加工対象物の基準位置である。図面の基準点（左下角など）を決め、FluidNC の G54 系で設定できるが、当面は手動設定を推奨する。

5.3　バックラッシュ補正

バックラッシュ（機構的遊び）による誤差は、CAM のオフセット設定で補正する。

1. 長辺と短辺（例：30 mm × 40 mm）の長方形輪郭を切削。
2. 実測と設計値の差から各軸の寸法誤差を算出。
3. CAM に補正オフセットを設定し、再切削して最小化。

5.4　加工前の確認事項

- ワーク設置面が水平であること（水平器等で確認）。
- ワーク固定が確実であること（締結状態を再確認）。

フラットベッドは湿度変化による木材の反り・歪みや過切削で凹凸が生じる。年 1 回程度を目安に水平出しを行い、精度を維持する。

5.5　フラットベッドの水平出し

本機のフラットベッドはスタイロフォーム製であり、変形や凹凸の影響を受けやすい。標準サイズは幅 910 mm、長さ 1820 mm である。

作業目的

ベッド全面を浅く切削して平坦性を回復する。スタイロフォームでは 3 mm to 5 mm 程度の切り込みを許容できることが多い。

事前準備

- ベッド上の釘・ネジ・金属部品は**必ず撤去**（工具破損・事故防止）。
- 面取り用またはサーフェスカッター（直径 16 mm to 25 mm、2–3 枚刃超硬）。
- 集塵機やエアブローで切粉排出経路を確保。

手順

1. **原点設定**：左前端をワーク原点とし、Z 原点を適切に設定。
2. **加工範囲設定**：X=910 mm、Y=1820 mm を指定（外周に 20 mm 程度のマージン可）。
3. **切削条件例**：
   - 切り込み量（Z）：3.0 mm to 5.0 mm ／パス
   - 送り速度：500 mm min^{-1} to 800 mm min^{-1}
   - スピンドル回転数：10 000 rpm to 16 000 rpm
4. **パス生成**：全面走査パスを作成。ピッチは工具径の 80–90% 程度とし、パスをオーバーラップさせて段差を低減。X ／ Y 交互切削は効果的だが時間がかかるため任意。
5. **切削実行**：まず小領域を浅切削して高低差を把握。必要に応じて深さを足して全面切削。
6. **仕上げ確認**：凹凸が許容範囲なら完了。

注意

- 切込みが大きい場合でも一度で削り切らず、2–3 回に分けると精度が出やすい。
- 水平出し後はベッドの損傷具合を点検する。

適切に設定しないと長時間（数十時間）を要する。太径エンドミルやサーフェスカッターの導入を検討すること。

6　おまけ

用語集（機械／加工）

ステッピングモーター パルス信号で一定角度ずつ回転するモーター。位置制御に用いる。

ステッピングドライバ ステッピングモーターを駆動する回路／モジュール。電流制御やマイクロステップ生成を行う。

NEMA17・NEMA23 モーター取付寸法規格。NEMA17 は約 42 mm、NEMA23 は約 57 mm フレーム。

バイポーラ 巻線に双方向の電流を流す駆動方式。高トルクだが電流制御が必要。

リードスクリュー 回転を直線運動に変えるねじ機構。

マイクロステップ 巻線電流を細かく制御し 1 ステップを分割。滑らかさ・振動低減に有効。

保持トルク 通電して静止を保てる最大トルク。

脱調 指令ステップと実回転がずれる現象。過負荷・過大加減速・共振が原因。

共振 固有振動数と駆動周波数一致で振動が増える。速度帯回避や減衰で対策。

バックラッシュ ねじやギアの遊びによる方向反転時のガタ。オフセット補正やプリロードで低減。

リミットスイッチ 移動端検出。原点復帰や過走行防止に用いる。

NC（常閉）接点 通常導通、作動で開放。断線検知に有利。

Z プローブ 工具先端の Z=0 位置を自動検出する仕組み。

スピンドル 工具を回転させる主軸。回転数・剛性が品質に影響。

エンドミル 側面・底面で切削できる工具。

サーフェスカッター 面出しに用いる広幅工具。

送り速度 工具とワークの相対移動速度（$\mathrm{mm\,min^{-1}}$）。

ツール長補正 工具長差を Z オフセットで補正。

ツイストペア 2 本を撚ってノイズを相殺する配線。

フェライトコア ケーブルに装着して高周波ノイズを減衰。

ケーブルクランプ／配線ダクト ケーブル固定・整理用部品。

用語集（電気／信号）

STEP ステッピングモーターの回転指令パルス。

DIR 回転方向指定信号。

EN 駆動有効／無効の切替（Enable）。

GPIO 汎用入出力（General Purpose Input/Output）。

UART（TX/RX） シリアル通信方式。TX は送信、RX は受信。

フォトカプラ 光結合で信号を伝える絶縁素子。

GND 回路の基準電位（グラウンド）。

PWM パルス幅変調。

DIP スイッチ 小型レバー式切替スイッチ。

ストラッピングピン 起動設定を決める固定配線端子（GPIO 0, 2, 4, 5, 12, 15 など）。

EN ライン Enable 信号線。

用語集（CNC 固有）

G-code CNC 機械を制御する標準命令コード。

機械原点（Machine Zero） 機械座標系の原点位置。

ワーク原点（Work Zero） 加工対象物に対する基準位置。

Homing（原点復帰） リミット等を使って機械原点に戻す動作。

ジョグ操作 手動（ソフト上）で軸を少しずつ動かす操作。

ソフトリミット 制御ソフト上の移動範囲制限。

用語集（数量／単位／制御パラメータ）

steps per mm 1 mm 移動に必要なステップ数。

max rate 最大移動速度。

acceleration 加速度。

rpm 回転毎分（revolutions per minute）。

mm min^{-1} 送り速度の単位。

選定基準一覧

- シングルボードコンピュータ（例：Raspberry Pi 4B 4 GB）
  - CNCjs + ブラウザが安定稼働（クアッドコア、メモリ 4 GB 以上）
  - 複数 USB ポート（ESP32・周辺機器）
  - 有線 LAN 対応、長時間稼働の安定性

  - OS + Node.js 環境を容易に構築可能

- ESP32 ボード
  - FluidNC ／ GRBL 互換 FW が動作
  - ステッピングドライバ向け STEP ／ DIR 出力を十分数確保
  - USB シリアルが安定（Wi-Fi は導入時のみ）
  - 5 V 系から安定給電（オンボード LDO 経由）

- ステッピングドライバ（例：TB6600 系）
  - 定格電流 3 A クラス対応
  - マイクロステップ（1/8〜1/32）
  - 入力仕様を確認
  - 放熱性（ヒートシンク／ファン）に配慮

- ステッピングモーター（NEMA17 ／ 23）
  - 必要トルク・慣性に見合うサイズ。12V ／ 3A 程度で十分。
  - 定格電流がドライバ範囲内

- スピンドルモーター
  - ER25 コレット対応で工具径自由度
  - 木材・樹脂向けに 10 000 rpm 以上を確保
  - 200 W to 400 W 以上を目安

- 電源（24 V 系・5 V 系・スピンドル用）
  - 24 V：モーター本数×定格電流をカバー
  - 5 V：Raspberry Pi ／ ESP32 へ 3 A 以上を安定供給
  - モーター系と制御系は**別系統**でノイズ回避

- USB ケーブル（Raspberry Pi–ESP32）
  - シールド付きでノイズに強い
  - 長さは必要最小限（0.5 m to 1.0 m 目安）
  - コネクタ形状（Micro-B ／ Type-C）に適合

参考文献

[1] Fluidnc wiki（公式）. http://wiki.fluidnc.com/en/home.

[2] Fluidnc web installer（公式）. http://wiki.fluidnc.com/en/installation.

[3] Google chrome シリアルポートに対して読み取り / 書き込みを行う（公式）. https://developer.chrome.com/docs/capabilities/serial.

[4] Esp32 の gpio ピンのクセ（非公式）. https://qiita.com/nanbuwks/items/0c5ee8fb4cf10d9c59e1.

[5] Espressif esp32 series datasheet（公式）. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[6] Fluidnc config file overview（公式 wiki）. http://wiki.fluidnc.com/config/overview.

[7] Fluidnc motion setup（公式 wiki）. http://wiki.fluidnc.com/en/support/setup/motion.

[8] Cncjs （公式）. https://cnc.js.org/.

[9] Raspberry pi os ダウンロードページ（公式）. https://www.raspberrypi.com/software/operating-systems/.

[10] Configuration raspi-config （公式）. https://www.raspberrypi.com/documentation/computers/configuration.html.

[11] Raspberry pi whitepaper: Making a more resilient file system（公式）. https://pip.raspberrypi.com/categories/685-app-notes-guides-whitepapers#document-25.

[12] Raspberry pi imager（公式）. https://www.raspberrypi.com/software/.

[13] Getting started with your raspberry pi（公式）. https://www.raspberrypi.com/documentation/computers/getting-started.html#advanced-options-in-raspberry-pi-imager.

[14] Cncjs installation guide（公式）. https://cnc.js.org/docs/installation/.

[15] Cncjs raspberry pi setup guide（公式）. https://cnc.js.org/docs/rpi-setup-guide/.

[16] Cncjs npm パッケージページ（公式）. https://www.npmjs.com/package/cncjs.

[17] Node.js: Install via package manager（公式）. https://nodejs.org/en/download/package-manager.

[18] Nodesource distributions（公式 github）. https://github.com/nodesource/distributions.

[19] systemd.service マニュアル（公式 freedesktop.org）. https://www.freedesktop.org/software/systemd/man/systemd.service.html.

[20] Toshiba tb6600hg ステッピングモータドライバ ic（公式）. https://toshiba.semicon-storage.com/info/TB6600HG_datasheet_ja_20160610.pdf?did=13822&prodName=TB6600HG.