

Chart Builder Structure

このモジュールは、Kairoscopeシステムの中核部分であるチャート生成処理（chart_builder.py）を担当する。天文演算・ゲート変換・プロファイル・権威・チャンネル活性化・センター定義を統合し、個別のHDチャートを生成・出力する。

ファイル名

chart_builder.py

構造概要（予定機能ブロック）

```
# 外部モジュール (Kairoscope Core)
from core.astro.astro_position import get_planet_positions
from core.gate_mapper import map_gates
from core.profile_logic import determine_profile
from core.authority_logic import determine_authority
from core.channel_center import analyze_channels_and_centers

# 出力用 (将来的にChronogramと連携)
import json
```

関数設計（今後の追加予定）

1. build_chart(input_data)

- 入力：生年月日・出生時間・出生地（緯度経度 or 都市名）
- 出力：HDチャートのJSONオブジェクト
- 機能：
 - 天体位置の取得（黄経）
 - ゲート+ラインへの変換
 - プロファイル決定（太陽+地球）
 - 権威（Authority）の算出
 - アクティブチャンネルと定義センターの推定

2. format_chart(chart_data)

- Kairoscopeの内部形式 → Chronogramで使える形式に整形
- 出力項目例：
 - `gates` : { gate_id: line, planet, etc. }
 - `channels` : [active channel list]

- `centers` : [defined / undefined]
-

外部参照ファイル

- `/core/definitions/gate-definitions.json`
 - `/core/definitions/channel-definitions.json`
 - `/spec/spec.json` (出力仕様)
-

今後の拡張 (Chronogram接続)

- 詳細なプロファイル情報・ホロジェネティクス・PHSなどとの接続
 - GUIやAPI出力形式に対応 (HTML / JSON / PDF)
 - ChatGPT / AIによる意味生成・性格文生成の基盤へ接続
-

タケ、次は `build_chart()` 関数の内部処理を1ステップずつ書いていこうか？ それとも先に入力データの形式 (spec) を定義する？ 