

REQUIREMENTS DOCUMENT

# QRコード 画像作成ツール

Python × Tkinter Desktop Application



# 目次

## CONTENTS

### 01 プロジェクト概要

Project Overview

### 03 技術スタック

Technology Stack

### 05 入出力仕様

I/O Specifications

### 07 エラーハンドリング

Error Handling

### 09 拡張案

Future Extensions

### 02 最終目標

Final Goal

### 04 機能一覧

Feature List

### 06 ユーザーインターフェース

User Interface

### 08 非機能要件

Non-Functional Requirements

### 10 受け入れ基準

Acceptance Criteria



## 01 PROJECT OVERVIEW

# プロジェクト概要



QRコード画像作成ツール



ユーザーが選択した**画像ファイル**または**手動入力テキスト**（例: URL）をエンコードしてQRコードを**生成・プレビュー**し、**画像として保存**できるデスクトップツールを提供する。



OS  
Windows



Runtime  
Python 3



- デスクトップアプリケーション
- GUIベースの直感的操作
- リアルタイムプレビュー機能
- 柔軟な出力形式対応



QR CODE GENERATION

02 / FINAL GOAL

## 最終目標

作成後も使える実用性の高いデスクトップアプリを開発



### 持続可能

開発完了後も継続的に使用できる実用的なツールとして設計



### 画像対応

ユーザーが画像ファイルをアップロードしてQRコード化できる機能

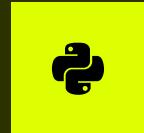


### 常時利用

いつでも使えるデスクトップツールとしての利便性を追求

Building a practical tool for everyday use

# 技術スタック



## メイン言語: Python 3

シンプルで強力なプログラミング言語。豊富なライブラリエコシステムを活用。

3.x

Version



## GUIフレームワーク

### Tkinter

Python標準のGUIライブラリ。クロスプラットフォーム対応で学習コストが低い。



## 主要ライブラリ

### qrcode

QRコード生成用

### pathlib

ファイルパス操作用

### Pillow (PIL)

画像処理・保存・プレビュー用



requirements.txt: qrcode[pil], Pillow

pip install

ready to use

# 機能一覧（必須機能）



## 画像ファイル選択機能

ファイル選択ダイアログから画像ファイル (.jpg, .png等) を選択。  
選択されたファイルのパスを内部データとして保持。



## 手動入力フィールド

任意のURLやテキストを直接入力できるテキストフィールドを提供。  
画像ファイル選択以外の入力方法として利用可能。



## QRコード生成

選択・入力されたデータをエンコードしてQRコードを生成。ファイルパスまたはテキスト文字列をQRコード化。



## プレビュー表示

生成されたQRコードを画面上でリアルタイムに確認。縮尺、解像度の選択オプションを検討。



## QRコード画像保存

生成したQRコードをPNG/JPEG等の画像形式で保存。ユーザー指定の形式とパスに対応。



## 保存ダイアログ

保存時にファイル名・保存先を選べるダイアログを表示。ユーザーの希望に応じた柔軟な保存が可能。

# 05 / I/O SPECIFICATIONS | 入出力仕様

## → 入力 (Input)

### ■ 画像ファイル

ファイル選択ダイアログから選択されるQRコードデータとしてエンコードする対象の画像ファイル。

.jpg   .png   etc.

### A 手動データ

ユーザーが直接入力する任意のURLまたはテキスト。画像データを直接貼り付ける場合は、ファイルとして扱う仕様にするか、Base64等で取り扱う案を検討。

## ⚙️ 処理 (Process)

### エンコードデータ:

選択されたファイルの絶対パス、または手動入力されたテキストを文字列としてQRコードにエンコードする。

※ ファイルそのもののバイナリをエンコードする場合は、容量上の注意が必要

## → 出力 (Output)

### ⌚ プレビュー

生成されたQRコードのビットマップ画像（画面上での確認用）。縮尺、解像度の選択オプションを付けることを検討。

### 💾 保存ファイル

ユーザーが指定したファイル名と場所に保存されるPNG/JPEG形式のQRコード画像ファイル。

## 06 / USER INTERFACE

# ユーザーインターフェース

### ■ メイン画面構成



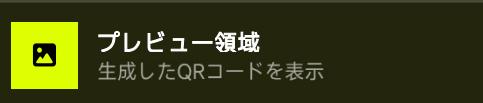
ファイル選択ボタン  
「ファイルを開く」



手動入力フィールド  
URLを入力



「生成」ボタン  
QRコードを作成



プレビュー領域  
生成したQRコードを表示



「保存」ボタン  
保存ダイアログを表示

### 操作フロー

- 1 ユーザーがファイルを選択する、またはテキストを入力する
- 2 「生成」ボタンでQRコードを生成し、プレビューに表示する
- 3 プレビューで確認後、「保存」ボタンでPNG/JPEGを出力する

### ■ UIスクリーンショット



実際のUIイメージ  
シンプルで直感的なデザイン

Input Mode  
Text / URL or File Path



# エラー処理



## 入力なしで「生成」した場合

ユーザーが何も入力せずに生成ボタンを押した場合、適切な警告を表示。

エラーメッセージ: 「入力がありません」



## サポート外のファイル形式

対応していないファイル形式が選択された場合、エラーメッセージを表示。

対応形式を案内

.jpg, .png 等の対応形式を明示



## 保存失敗時

アクセス権限等の理由で保存に失敗した場合、ユーザーに通知。

保存失敗ダイアログ

別パスの選択を促す



すべてのエラーはユーザーフレンドリーなメッセージで通知し、次のアクションを明確に示す



# 非機能要件

## 可用性

シンプルなUIで直感的に操作できること。初めて使うユーザーでも迷わず操作できる設計を目指す。

## 性能

QRコードの生成は瞬時～数秒以内に完了すること（大容量データを除く）。ストレスのない快適な操作体験を提供。

## 互換性

Windows環境で動作すること。Python 3.xの標準環境で問題なく動作するよう設計・実装する。

## 保守性

依存ライブラリは最小限に抑え、拡張しやすい設計にする。将来の機能追加や改修を見据えた構造化を行う。

## セキュリティ

手動入力されたURLやデータは外部送信を行わず、ローカルで処理する。外部通信の必要が出る場合は明示的にユーザーに通知し、同意を得る設計とする。

# 拡張案（将来検討）



## バッチ処理機能

複数ファイルを一括でQRコード生成する機能。複数の画像やURLをまとめて処理し、時間を節約。

複数ファイル

一括生成



## サイズ・誤り訂正レベル選択

QRコードのサイズと誤り訂正レベル（L/M/Q/H）をユーザーが選択できる機能。用途に応じた最適な設定が可能に。

L

M

Q

H



## 透かし・ラベル機能

エクスポート時に透かしやラベル（元ファイル名）を付ける機能。生成したQRコードの識別やブランディングに活用。

透かし

ラベル

ファイル名



## クロスプラットフォーム対応

Mac / Linux環境への対応拡張。Tkinterが動作する環境での動作検証を行い、マルチプラットフォーム化を実現。

Mac

Linux

Cross-Platform



これらの拡張機能は、**基本機能の完成後に段階的に実装を検討**



# 受け入れ基準

以下の**条件**をすべて満たすこと



## QRコード生成

画像ファイル (.jpg, .png) または手動入力テキストからQRコードが**正常に生成**できること



## プレビュー表示

生成結果が**プレビューで確認**できること。リアルタイムで生成されたQRコードを画面上で確認可能



## 画像保存

ユーザーが**任意のファイル名・保存先**でPNG/JPEGとして保存できること



## エラーハンドリング

基本的なエラー（未入力、非対応形式、保存失敗）に対する**メッセージ**が適切に**表示**されること



PROJECT COMPLETION

すべての基準を満たした時点で開発完了



11 / QR code screenshot

## QRコードスクショ

実際のスクリーンショット画像



JAXA GCOM-C SATELLITE DATA

# 地表面温度 データ分析

JAXA-MCP

SGIセンサー観測データの  
可視化と応用

01



JAXA Earth Observation

2026.02.18

# 02

GCOM-C  
「しきさい」

## SATELLITE OVERVIEW

# 気候変動観測衛星

2017年12月に打ち上げられたJAXAの気候変動観測衛星。  
SGLIセンサーを搭載し、19チャンネル・250m解像度で  
約2日に1回の頻度で全球観測を実施。



### ミッション目的

放射収支と炭素循環の変動メカニズム解明



### 観測能力

地球全体を約2日でカバー

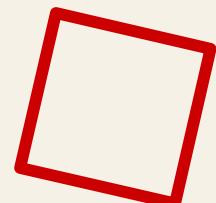


### データ精度

250m高解像度、19スペクトルバンド



© JAXA



03

SGLI SENSOR

## 技術仕様

### Second-generation Global Imager

ADEOS-II/GLIの後継センサー。380nm～12μmの広い波長範囲を19チャンネルで観測。

**波長範囲** 380nm～12μm

**観測幅** 1,150km (VNR)～1,400km (IRS)

**解像度** 250m (陸域・沿岸域)

**特殊機能** 偏光・多角度観測

SGLI スペクトルバンド



# 04

LST DATA

## 地表面温度データ

### 熱赤外線観測

10.8μmと12.0μmの熱赤外線バンドを使用し、地表から放射される熱エネルギーを観測

### 物理量変換

$$\text{物理量} = \text{DN} \times \text{slope} + \text{offset}$$

slope = 0.02, offset = 0

### 単位

ケルビン (K)

### データ利点

- ✓ 日周変動の把握
- ✓ 季節変動の追跡
- ✓ 長期トレンド分析
- ✓ 都市ヒートアイランド監視

### 温度範囲例

250K ————— 316K

極寒地

砂漠地帯

Land Surface  
Temperature

## 05 データ取得システム



### JAXA G-Portal

Mission Control Portal

JAXAが提供する衛星データポータル。GCOM-C/SGLIのL3データを検索  
・ダウンロード可能。

- Webインターフェース対応
- 多様なデータフォーマット
- 詳細なメタデータ



### JAXA Earth API

2022年6月公開

開発者向けの衛星データ取得API。PythonとJavaScriptに対応。

- COG形式対応
- STACカタログ
- 無償・商用利用可



Cloud Optimized  
GeoTIFF形式



STAC対応  
時空間カタログ



data.earth.jaxa.jp  
統合ポータル



IT開発者向け  
民主化推進

06

## OBSERVATION DATA

# 福岡県 地表面温度

2025年12月 月間データ

L3-LST daytime v3 global monthly

緯度範囲

33.01° ~ 33.99°

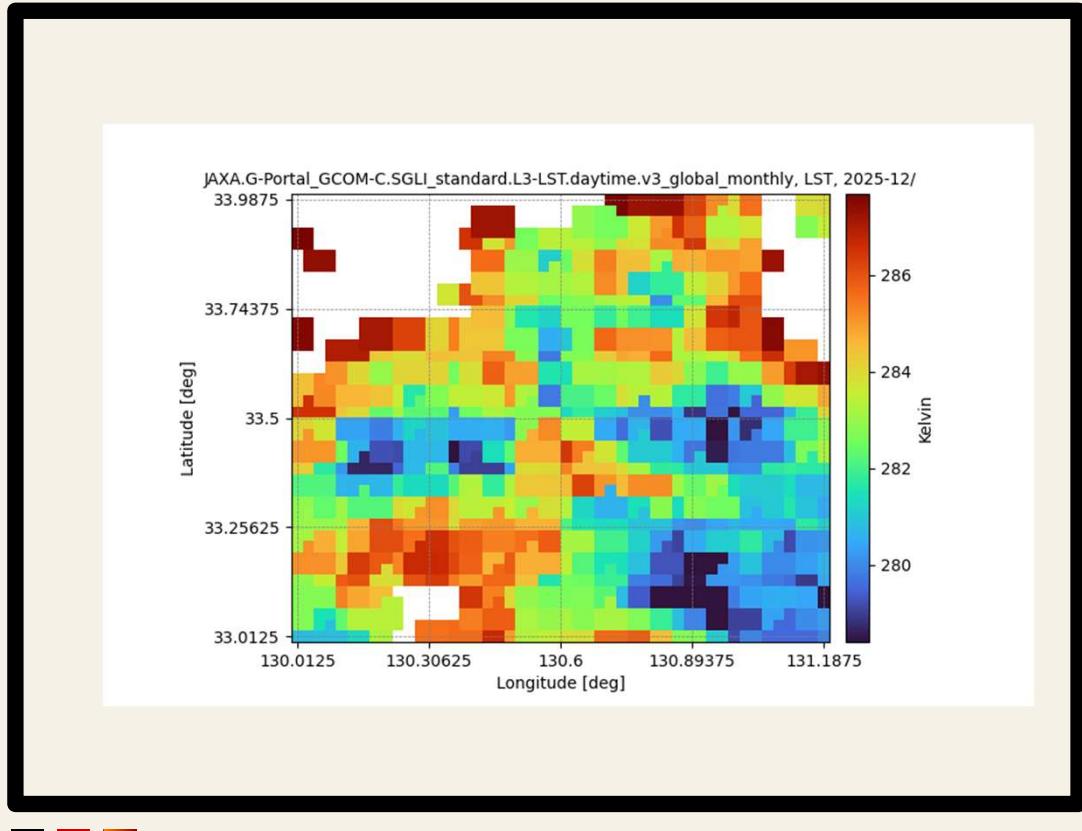
経度範囲

130.01° ~ 131.19°

温度範囲

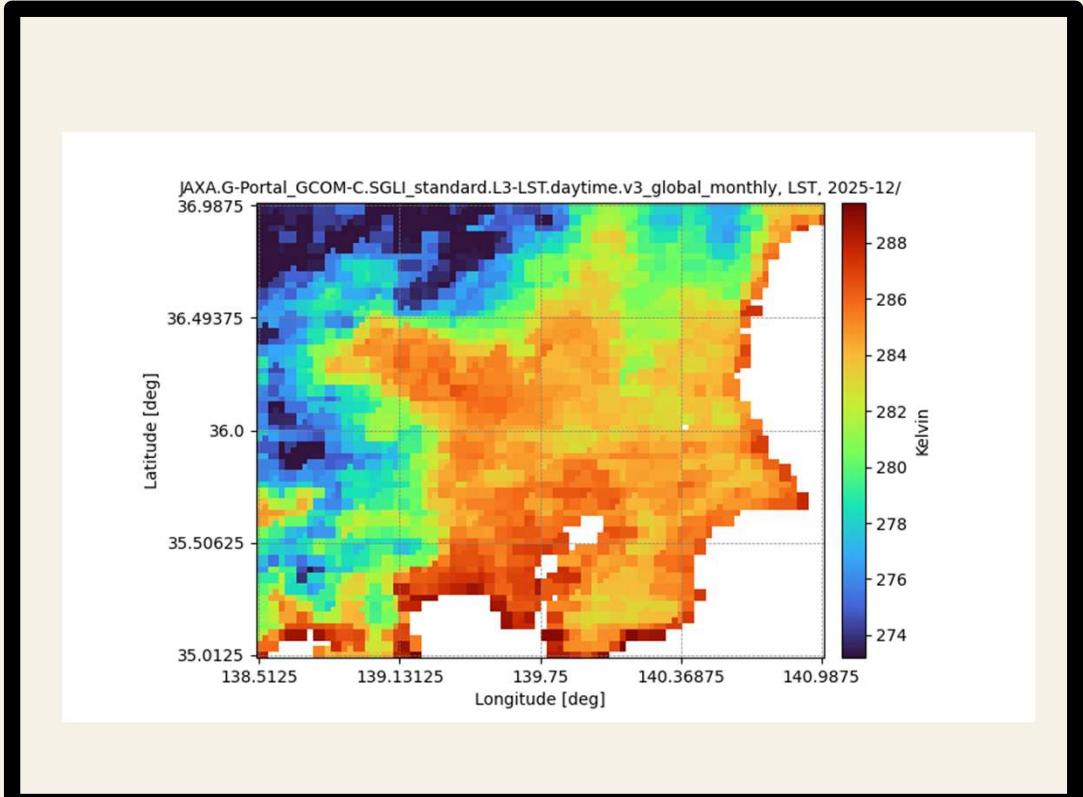
280K ~ 286K

● 都市部と周辺部の温度差が明確に観測される



● Kelvin (280K → 286K)





## OBSERVATION DATA

# 関東地方 地表面温度

2025年12月 月間データ

L3-LST daytime v3 global monthly

緯度範囲

35.01° ~ 36.99°

経度範囲

138.51° ~ 140.99°

温度範囲

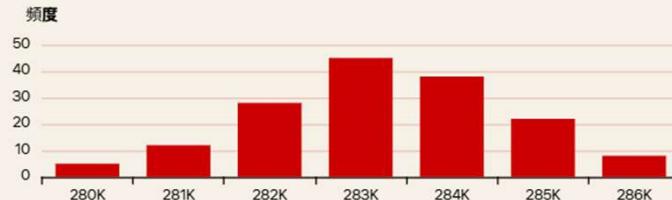
274K ~ 288K

⚠ 東京湾周辺の都市ヒートアイランド現象が顕著

# 温度分布パターン分析

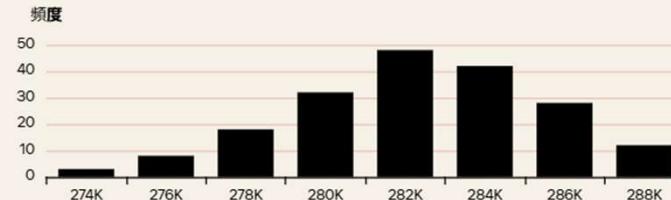
福岡県

温度分布



関東地方

温度分布



- 福岡市中心部で相対的高温
- 筑後平野の広域分布
- 山地部での低温化

- 東京・横浜の顕著なヒートアイランド
- 内陸部での極端な温度上昇
- 山梨・長野県境の低温帯

福岡県平均  
283K | 関東平均  
281K | 温度差  
+2K

福岡県は関東よりも平均温度が高い。これは冬季の気候特性と地形要因によるもの。

## 09 応用分野：気候変動研究



### 地球システムモデル

LSTデータを気候モデルへの入力データとして活用。地表面温度の正確な再現により、気候予測の精度向上に貢献。

- 放射収支計算
- エネルギー循環解析



### 温暖化予測

長期的な地表面温度トレンドの分析により、地球温暖化の進行状況を把握。予測モデルの検証に使用。

- IPCC報告書への貢献
- COPでの政策反映



### SENTANプログラム

気候変動予測先端研究プログラム。衛星データを活用した火災モデル開発により、地球システムモデルの予測精度向上を目指す。

- 林野火災過程のモデル化
- 物質放出量推定



### 国際的な気候変動対策への貢献

JAXAの衛星データは、IPCC（気候変動に関する政府間パネル）の報告書やCOP（気候変動枠組み条約締約国会議）での温暖化政策策定に活用される。地球規模の気候変動予測精度向上に向け、国際的な研究機関と連携し、データの精度検証や品質向上を進めている。

# 10

防災・環境  
モニタリング

## APPLICATION

### 多様な社会課題解決へ



#### 都市ヒートアイランド

都市部の地表面温度を継続的に監視し、ヒートアイランド現象の実態把握と対策立案に活用。

✿ 热中症リスク評価



#### 農業管理

作物の水ストレス状態を把握し、灌漑計画の最適化や収量予測に利用。水資源の効率的な活用を支援。

✿ 蒸発散量推定



#### 林野火災検出

地表面温度の異常上昇を検知し、火災発生の早期発見と拡大防止に貢献。燃焼面積の推定にも活用。

✿ リアルタイム監視

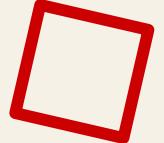


#### 災害・インフラ監視

災害時の状況把握やインフラ施設の温度監視に活用。地盤沈下や構造物の異常検知にも利用可能。

✿ 広域観測

# JAXA衛星データの社会実装



## 農林水産業

作物生育監視、森林状況把握、漁場環境分析。広範囲の土地状況を効率的に把握し、生産性向上に貢献。



## 土木・インフラ

都市の地盤沈下監視、インフラ施設の老朽化把握。SARセンサーと組み合わせた広域観測で維持管理コスト削減。



## 海洋状況把握（MDA）

Maritime Domain Awareness。船舶追跡、海洋環境監視。政府機関への情報提供を通じて海洋安全保障に貢献。



## Tellusプラットフォーム

JAXAが推進する衛星データプラットフォーム。クラウド上で衛星データの検索、分析、可視化が可能。宇宙戦略基金事業の代表機関として、衛星データの民主化を推進。

- API提供
- クラウド解析環境
- 無償利用



## 宇宙戦略基金事業

衛星データ利用システム実装加速化

持続可能で競争力のある産業構造確立を目指し、衛星データを社会インフラとして位置づける

**C O N C L U S I O N**

# 衛星データで 地球を見る

JAXA GCOM-C/SGLIの地表面温度データは、  
気候変動予測から防災・環境モニタリングまで、  
**多様な社会課題解決に貢献する。**

❖ 宇宙からの視点が、地球の未来を照らす。

12



**JAXA Earth Observation**  
GCOM-C / SGLI / LST