PCB Footprint Generator - Implementation Context Document

This document contains all context, requirements, technical specifications, and ground truth data needed to implement the AI PCB Footprint Generator. This project is intended as a demo/MVP for Arena AI.

---

1. PROJECT OVERVIEW

1.1 Problem Statement

Hardware engineers frequently encounter components without existing PCB footprints. Creating footprints manually from datasheets is time-consuming (15-60 min per footprint), error-prone, and tedious. Errors lead to costly board re-spins ($5K-50K+).

1.2 Proposed Solution

A web application where engineers upload datasheet images/PDFs containing land pattern drawings. AI extracts dimensions and generates Altium Designer-compatible footprint files.

1.3 MVP Scope

- User uploads cropped datasheet image (land pattern section)
- AI extracts pad dimensions, positions, shapes
- User confirms/edits extracted values
- System generates Altium 26 compatible footprint file
- User downloads and imports to Altium

1.4 Tech Stack

- **Frontend:** React + Tailwind CSS
- **Backend:** Python FastAPI
- **AI:** Anthropic Claude API (start with Haiku for cost, upgrade path to Sonnet/Opus)
- **Hosting:** Railway

---

2. SUPPORTED PAD TYPES (MVP)

2.1 Pad Technologies

- **SMD (Surface Mount):** Top Layer only, no drill hole

- **Through-Hole (TH):** MultiLayer, circular drill hole
- **Slotted Through-Hole:** MultiLayer, oval/slot drill hole
- **Mixed:** Footprints containing both SMD and TH pads

2.2 Pad Shapes

- **Round:** Circular pads
- **Rectangular:** Square or rectangular pads
- **Rounded Rectangle:** Rectangular with rounded corners (Pin 1 indicator)
- **Oval:** Elongated pads (often for shield mounts)

2.3 Additional Elements

- **Thermal/Exposed Pads:** Large central pads for heat dissipation (e.g., QFN, PowerPAD)
- **Thermal Vias:** Small vias within exposed pad area
- **Mounting Holes:** Non-plated through holes (NPTH) for mechanical mounting
- **Silkscreen Outline:** Component boundary on TopOverlay layer

---

3. ALTIUM FILE FORMAT SPECIFICATION

3.1 Target Version

Altium Designer 26

3.2 Output Format

Altium ASCII format - tab-delimited text that can be imported or pasted into Altium's PcbLib editor.

3.3 Internal Units

Altium uses **mils (1/1000 inch)** internally in many exports, but also supports **mm**. The ground truth data shows coordinates in mm for some footprints and mils for others. The generator should output in mm for consistency with datasheet dimensions.

3.4 Key Fields for Pad Records

Based on parsed ground truth, the Altium pad format includes these key fields (tab-delimited):

```
Type  Layer  Net  Designator  X  Y  Rotation  HoleSize  [Shape fields...]
```

Critical fields:

- **Type:** Pad, Via, Track, Arc, Region, 3D Body, Text
- **Layer:** MultiLayer (TH), Top Layer (SMD), TopOverlay (silkscreen)
- **X, Y:** Pad center coordinates
- **Rotation:** Degrees (0, 90, 180, 270)
- **Pin Designator:** Pad name/number (e.g., "1", "A1", "SH1", "Un1")
- **HoleSize:** Drill diameter for TH pads (0 for SMD)
- **Shape:** Round, Rectangular, Rounded Rectangle
- **XSize, YSize:** Pad dimensions

3.5 Example Pad Records (from ground truth)

Through-hole round pad:

```
Pad  MultiLayer  No Net  Free -5.715  8.89  False  False  False  1  0.000  0.9
... Simple Rounded Rectangle 1.5  1.5  ... Drilled Round 0  0.000  ...
```

SMD rectangular pad:

```
Pad  Top Layer  No Net  Free -2.498  1.905  False  False  1  90.000 0  N/A N/A 0  ... Simple
Rectangular 0.802  1.505  ... Drilled Round 0  0.000  ...
```

Slotted hole pad:

```
Pad  MultiLayer  No Net  Free -6.4  0  False  False  False  SH1 90.000  0.65 ...
Simple Round 3.05 1.25 ... Drilled Slot 2.45 0.000  ...
```

---

4. GROUND TRUTH EXAMPLES

4.1 Example 1: RJ45 Connector (LPJG0926HENL)

Type: Through-hole connector with mixed pad sizes **Manufacturer:** LINK-PP **Total Pads:** 22

Pad	X (mm)	Y (mm)	Pad Size (mm)	Drill (mm)	Shape	Notes
1	-5.715	8.89	1.5 x 1.5	0.9	Rounded Rectangle	Pin 1 marker
2	-4.445	6.35	1.5 x 1.5	0.9	Round	
3	-3.175	8.89	1.5 x 1.5	0.9	Round	
4	-1.905	6.35	1.5 x 1.5	0.9	Round	
5	-0.635	8.89	1.5 x 1.5	0.9	Round	
6	0.635	6.35	1.5 x 1.5	0.9	Round	90° rotation
7	1.905	8.89	1.5 x 1.5	0.9	Round	
8	3.175	6.35	1.5 x 1.5	0.9	Round	
9	4.445	8.89	1.5 x 1.5	0.9	Round	
10	5.715	6.35	1.5 x 1.5	0.9	Round	
11	-5.715	3.83	1.5 x 1.5	0.9	Round	
12	-3.175	2.56	1.5 x 1.5	0.9	Round	
13	3.175	2.56	1.5 x 1.5	0.9	Round	
14	5.715	3.83	1.5 x 1.5	0.9	Round	
15	-6.63	-4.06	1.5 x 1.5	1.02	Round	LED pins
16	-4.09	-4.06	1.5 x 1.5	1.02	Round	LED pins
17	4.09	-4.06	1.5 x 1.5	1.02	Round	LED pins
18	6.63	-4.06	1.5 x 1.5	1.02	Round	LED pins
19	-7.875	3.05	2.5 x 2.5	1.7	Round	Shield/mount
20	7.875	3.05	2.5 x 2.5	1.7	Round	Shield/mount
Un1	-5.715	0	3.2 x 3.2	3.2	Round	Mounting hole
Un2	5.715	0	3.2 x 3.2	3.2	Round	Mounting hole

Characteristics:

- All through-hole pads
- Three different pad/drill size classes
- Pin 1 identified by Rounded Rectangle shape
- Complex non-grid arrangement

4.2 Example 2: USB 3.0 Stacked Connector (GSB3115XXXXF1HR)

Type: Through-hole connector with slotted holes **Manufacturer:** Amphenol **Total Pads:** 22

Pad	X (mm)	Y (mm)	Pad Size (mm)	Drill (mm)	Shape	Notes
1	-3.5	-0.57	1.25 x 1.25	0.75	Rectangular	Pin 1 (square)
2	-1	-0.57	1.25 x 1.25	0.75	Round	
3	1	-0.57	1.25 x 1.25	0.75	Round	
4	3.5	-0.57	1.25 x 1.25	0.75	Round	
5	4	0.93	1.25 x 1.25	0.75	Round	
6	2	0.93	1.25 x 1.25	0.75	Round	
7	0	0.93	1.25 x 1.25	0.75	Round	
8	-2	0.93	1.25 x 1.25	0.75	Round	
9	-4	0.93	1.25 x 1.25	0.75	Round	
10	-3.5	2.63	1.25 x 1.25	0.75	Round	
11	-1	2.63	1.25 x 1.25	0.75	Round	
12	1	2.63	1.25 x 1.25	0.75	Round	
13	3.5	2.63	1.25 x 1.25	0.75	Round	
14	4	4.13	1.25 x 1.25	0.75	Round	
15	2	4.13	1.25 x 1.25	0.75	Round	
16	0	4.13	1.25 x 1.25	0.75	Round	
17	-2	4.13	1.25 x 1.25	0.75	Round	
18	-4	4.13	1.25 x 1.25	0.75	Round	
SH1	-6.4	0	3.05 x 1.25	0.65 (slot 2.45)	Oval	90° rot, shield
SH2	6.4	0	3.05 x 1.25	0.65 (slot 2.45)	Oval	90° rot, shield
SH3	-7.8	-7.5	2.15 x 1.075	0.65 (slot 1.65)	Oval	270° rot, shield
SH4	7.8	-7.5	2.15 x 1.075	0.65 (slot 1.65)	Oval	270° rot, shield

Characteristics:

- Slotted drill holes (SH1-SH4)
- Oval/elongated pads for shield mounts
- Pin 1 uses Rectangular shape (different from Example 1)
- 4 rows of signal pins with varying X positions

4.3 Example 3: M.2 Mini PCIe Connector (MDT-XXX-X-XX-001)

Type: SMD edge connector **Manufacturer:** Amphenol **Total Pads:** 79

Pad Type	Count	X Range (mm)	Y (mm)	Size (mm)	Shape	Layer
Signal (odd: 1,3...75)	38	-9.25 to 9.25	5.275	0.3 x 1.55	Rectangular	Top Layer
Signal (even: 2,4...74)	37	-9.0 to 9.0	-2.275	0.3 x 1.55	Rectangular	Top Layer
S1 (shield)	1	-10.35	4.5	1.2 x 2.75	Rectangular	Top Layer
S2 (shield)	1	10.35	4.5	1.2 x 2.75	Rectangular	Top Layer
Mounting (left)	1	-10.0	0	1.15 dia	Round	MultiLayer (NPTH)
Mounting (right)	1	10.0	0	1.65 dia	Round	MultiLayer (NPTH)

Characteristics:

- Pure SMD (except mounting holes)
- High pin count (75 signal pins)
- 0.5mm pitch, staggered dual-row
- Asymmetric mounting holes (keying)
- Raw data was in mils, converted to mm above

4.4 Example 4: Samtec HLE Socket (HLE-110-02-F-DV-BE-A)

Type: Mixed SMD + TH dual-row socket **Manufacturer:** Samtec **Total Pads:** 42

Pad Type	Count	X Range (mm)	Y (mm)	Size (mm)	Drill (mm)	Layer
Signal SMD (01-20)	20	-11.43 to 11.43	±2.72	1.27 x 1.68	—	Top Layer
TH mounting (grid)	20	-11.43 to 11.43	±1.27	0.97 dia	0.97	MultiLayer
Large mount holes	2	±10.16	0	1.78 dia	1.78	MultiLayer

Characteristics:

- True mixed technology (SMD + TH in same footprint)
- Unnumbered TH mounting pads
- 2.54mm (100 mil) pitch
- Pin 1 marker via Arc element

4.5 Example 5: SO-8EP (SOIC-8 with Exposed Pad)

Type: SMD IC package with thermal pad **Package:** SO-8EP / SOIC-8 PowerPAD **Total Pads:** 9 + 6 vias

Pad	X (mm)	Y (mm)	Size (mm)	Rotation	Shape	Notes
1	-2.498	1.905	0.802 x 1.505	90°	Rectangular	Pin 1
2	-2.498	0.635	0.802 x 1.505	90°	Rectangular	
3	-2.498	-0.635	0.802 x 1.505	90°	Rectangular	
4	-2.498	-1.905	0.802 x 1.505	90°	Rectangular	
5	2.497	-1.905	0.802 x 1.505	90°	Rectangular	
6	2.497	-0.635	0.802 x 1.505	90°	Rectangular	
7	2.497	0.635	0.802 x 1.505	90°	Rectangular	
8	2.497	1.905	0.802 x 1.505	90°	Rectangular	
9	0	0	2.613 x 3.502	0°	Rectangular	Exposed pad

Thermal Vias (6 total):

X (mm)	Y (mm)	Pad (mm)	Drill (mm)
±0.55	-1.1	0.5	0.2
±0.55	0	0.5	0.2
±0.55	1.1	0.5	0.2

Characteristics:

- Standard SOIC-8 with exposed thermal pad
- Thermal vias in footprint
- 90° rotated signal pads
- 1.27mm pitch
- Table-variable datasheet format (X, Y, X1, Y1, etc.)

---

5. DATASHEET DRAWING PATTERNS

5.1 Common Dimension Formats

Direct dimensions: Each pad position explicitly labeled **Dimension chaining:** Total width + gap + pad width (requires calculation) **Table-variable:** Generic drawing with A, B, C variables + lookup table **Center-to-center vs Edge-to-center:** Reference point varies

5.2 Extraction Challenges

1. **Dimension leader lines** may point to wrong features
2. **Tolerance notations** (±0.1mm) — extract nominal value
3. **Mixed units** — mm and mils in same drawing
4. **Pin 1 identification** — dot, chamfer, notch, or numbering
5. **Multiple variants** — same drawing for different package sizes

5.3 Standard Package Detection

Detect standard IPC-7351 package codes via regex:

- `QFN-XX`, `QFP-XX`, `SOIC-XX`, `TSSOP-XX`
- `BGA-XXX`, `0402`, `0603`, `0805`

- SOT-23, SOT-223, TO-252

If detected, offer user option to use Altium's built-in IPC Footprint Wizard instead of AI extraction.

---

6. USER WORKFLOW

```
1. User uploads datasheet image/PDF (cropped to land pattern section)
   ↓
2. System checks for standard package code
   ↓
3. [If standard] → Prompt: "Detected QFN-48. Use IPC calculator or continue?"
   ↓
4. [If continue or non-standard] → AI extracts dimensions
   ↓
5. Display extracted values in editable table + visual preview
   ↓
6. User confirms/edits values, specifies pin names
   ↓
7. System generates Altium ASCII footprint file
   ↓
8. User downloads file, imports to Altium 26
```

---

7. API ENDPOINTS

```
POST /api/upload
  - Accepts: image (PNG/JPG) or PDF (single page)
  - Returns: job_id

GET /api/extract/{job_id}
  - Returns: extracted dimensions (JSON), confidence scores

POST /api/confirm
  - Accepts: confirmed/edited dimensions, pin mapping
  - Returns: —

GET /api/generate/{job_id}
  - Returns: Altium footprint file (download)

GET /api/detect-standard
```

- Accepts: extracted text/geometry
- Returns: package classification, IPC parameters if standard

---

8. EXTRACTION OUTPUT SCHEMA

json

```
{
 "package_type": "custom | QFN | SOIC | ...",
 "standard_detected": "QFN-48" | null,
 "units": "mm",
 "component_outline": {
  "width": 5.0,
  "height": 4.0
 },
 "pads": [
  {
   "designator": "1",
   "x": -2.498,
   "y": 1.905,
   "width": 0.802,
   "height": 1.505,
   "rotation": 90,
   "shape": "rectangular",
   "type": "smd",
   "drill": null
  },
  {
   "designator": "SH1",
   "x": -6.4,
   "y": 0,
   "width": 3.05,
   "height": 1.25,
   "rotation": 90,
   "shape": "oval",
   "type": "th",
   "drill": {
    "diameter": 0.65,
    "slot_length": 2.45,
    "type": "slot"
   }
  }
 ],
 "vias": [
  {
   "x": 0.55,
   "y": -1.1,
   "pad_diameter": 0.5,
   "drill_diameter": 0.2
  }
```

```
  ],
  "confidence": {
    "overall": 0.85,
    "pad_dimensions": 0.90,
    "pad_positions": 0.80,
    "pin_count": 1.0
  }
}
```

---

9. SUCCESS CRITERIA

9.1 Demo Success

- End-to-end flow working: upload → extract → confirm → download → import to Altium 26
- At least one non-standard package extracts correctly >70% of time
- Generated footprint imports into Altium without errors
- Total processing time <60 seconds

9.2 Accuracy Targets

- Dimension extraction within 0.05mm of ground truth for >70% of values
- Pad count correct 100% (user can verify)
- User confirmation step catches remaining errors

---

10. IMPLEMENTATION NOTES

10.1 Technical Spikes (Do First)

1. **Altium 26 ASCII format validation** — Create test footprints manually, verify import
2. **Vision model accuracy test** — Run sample datasheets through Claude Haiku
3. **Prompt engineering** — Develop structured extraction prompt

10.2 Model Selection

Start with **Claude Haiku** for cost efficiency. Architecture should allow swapping to Sonnet/Opus if accuracy insufficient.

10.3 Error Handling

- "Could not detect dimensions" → Manual entry fallback
- "Multiple land patterns detected" → User selects
- "Units ambiguous" → User confirms
- "Pin 1 not found" → User designates

10.4 Pin 1 Indicators (from examples)

- Rounded Rectangle shape (Example 1)
- Rectangular/Square shape (Example 2)
- Arc element on TopOverlay (Examples 4, 5)
- Position-based inference (corner position)

---

11. RAW GROUND TRUTH DATA

The raw tab-delimited Altium export data for all 5 examples is available in the project knowledge. Key observations from parsing:

11.1 Field Positions (approximate, tab-delimited)

- Field 1: Type (Pad, Track, Arc, Via, etc.)
- Field 2: Layer
- Field 5-6: X, Y coordinates
- Field 14: Pin designator
- Field 15: Rotation (degrees)
- Field 21: Hole size
- Field ~35-40: Shape, XSize, YSize
- Field ~50: Drill type (Round, Slot)

11.2 Layer Names

- `MultiLayer` — Through-hole pads
- `Top Layer` — SMD pads
- `TopOverlay` — Silkscreen
- `TopPaste` — Solder paste

- `Mechanical 1`, `Mechanical 13`, `Mechanical 15` — Mechanical/documentation layers

---

12. FILES TO CREATE

1. **Backend (Python/FastAPI)**
   - `main.py` — API endpoints
   - `extraction.py` — Claude API integration, dimension extraction
   - `generator.py` — Altium ASCII file generation
   - `models.py` — Pydantic models for data structures
   - `utils.py` — Unit conversion, validation
2. **Frontend (React)**
   - Upload component
   - Extraction results display/editor
   - Visual footprint preview (2D canvas)
   - Download component
3. **Configuration**
   - `requirements.txt`
   - `package.json`
   - Railway deployment config

---

13. REFERENCE LINKS

- Altium Creating PCB Footprints: https://www.altium.com/documentation/altium-designer/components-libraries/creating-pcb-footprint
- Altium SDK Overview: https://www.altium.com/documentation/altium-dxp-developer/an-overview-of-the-altium-sdk

---

Document Version: 1.0 Created: February 2026 For: Arena AI Demo Project