# Pattern Testing Samples for OCR Pipeline

# Use these text samples to test if patterns.json is working correctly

## 1. NORMALIZATION PATTERNS

### remove\_extra\_spaces (Pattern: \\s+)

# Should convert multiple spaces to single space

Original: "This has multiple spaces"

Expected: "This has multiple spaces"

Original: "Text with tabs and spaces"

Expected: "Text with tabs and spaces"

### trim\_whitespace (Pattern: ^\\s+|\\s+$)

# Should remove leading and trailing whitespace

Original: " leading and trailing spaces "

Expected: "leading and trailing spaces"

Original: " tab at start and end "

Expected: "tab at start and end"

### normalize\_dashes (Pattern: [-–—])

# Should normalize en-dash (–) and em-dash (—) to hyphen (-)

Original: "Part number 77–123–456–789"

Expected: "Part number 77-123-456-789"

Original: "Range: 100—200 units"

Expected: "Range: 100-200 units"

Original: "Mixed dashes: 77-123–456—789"

Expected: "Mixed dashes: 77-123-456-789"

### normalize\_quotes (Pattern: [\\u201c\\u201d\\u2018\\u2019`\\u00b4])

# Should normalize smart quotes to standard quotes

Original: "This is "quoted text" with 'single quotes'"

Expected: "This is \"quoted text\" with \"single quotes\""

Original: "OCR often reads `backticks´ as quotes"

Expected: "OCR often reads \"backticks\" as quotes"

## 2. PREPROCESSING PATTERNS

### normalize\_part\_number\_newlines (Pattern: (?<![0-9])77[\\s\\n\\r]\*-[\\s\\n\\r]\*([0-9]+)[\\s\\n\\r]\*-[\\s\\n\\r]\*([0-9]+)[\\s\\n\\r]\*-[\\s\\n\\r]\*([0-9]+))

# Should normalize 77-\*-\*-\* with newlines to clean format

# Test Case 1: Normal format (should remain unchanged)

Original: "77-123-456-789"

Expected: "77-123-456-789"

# Test Case 2: With newlines between segments

Original: "77

-

123

-

456

-

789"

Expected: "77-123-456-789"

# Test Case 3: With spaces and newlines mixed

Original: "77

-

123

-

456

-

789"

Expected: "77-123-456-789"

# Test Case 4: In parentheses (no preceding space)

Original: "(77-123-456-789)"

Expected: "(77-123-456-789)"

# Test Case 5: In parentheses with newlines

Original: "(77

-

123

-

456

-

789)"

Expected: "(77-123-456-789)"

# Test Case 6: Adjacent to text (no preceding space)

Original: "Part77-123-456-789end"

Expected: "Part77-123-456-789end"

# Test Case 7: Should NOT match if 77 is part of larger number

Original: "12377-123-456-789"

Expected: "12377-123-456-789" (unchanged)

# Test Case 8: Multiple part numbers in same text

Original: "Parts: 77-111-222-333 and 77

-

444

-

555

-

666"

Expected: "Parts: 77-111-222-333 and 77-444-555-666"

### ocr\_noise\_removal (Pattern: [^\\w\\s\\-\\.\\,\\:;\\(\\)\\[\\]\\{\\}])

# Should remove OCR noise characters while keeping allowed punctuation

Original: "77-123-456-789@#$%^&\*+=|\\<>?/"

Expected: "77-123-456-789"

Original: "Part: 77-123-456-789, Model (ABC-123); Price: $99.99"

Expected: "Part: 77-123-456-789, Model (ABC-123); Price: 99.99"

### fix\_common\_ocr\_errors

# Should fix common OCR misrecognitions

# Test Case 1: 0 → O when before digits

Original: "Part 0123-456-789"

Expected: "Part O123-456-789"

# Test Case 2: l → 1 when before digits

Original: "Code l234-567-890"

Expected: "Code 1234-567-890"

# Test Case 3: S → 5 when before digits

Original: "Item S678-901-234"

Expected: "Item 5678-901-234"

# Test Case 4: Should NOT change when not before digits

Original: "Part 0ABC, Code lXYZ, Item SABC"

Expected: "Part 0ABC, Code lXYZ, Item SABC" (unchanged)

## 3. VALIDATION PATTERNS (for reference)

### part\_number\_format\_77 (Pattern: 77-\\d+-\\d+-\\d+)

# These should MATCH the validation pattern:

Valid: "77-123-456-789"

Valid: "77-1-2-3"

Valid: "77-999999-888888-777777"

# These should NOT match:

Invalid: "76-123-456-789" (wrong prefix)

Invalid: "77-123-456" (missing segment)

Invalid: "77-123-456-789-000" (extra segment)

### general\_part\_number\_format (Pattern: \\d{2,3}-\\d{3}-\\d{6,7}-\\d{2,3})

# These should MATCH the validation pattern:

Valid: "12-345-678901-23"

Valid: "123-456-7890123-456"

# These should NOT match:

Invalid: "1-345-678901-23" (first segment too short)

Invalid: "12-34-678901-23" (second segment too short)

## 4. COMBINED TEST CASES

# Test multiple patterns working together

### Complex OCR Text Sample

Original: " Part number: 77

-

123

-

456

-

789 is "quoted" with noise@#$ and dashes–like—this "

Expected after all patterns:

"Part number: 77-123-456-789 is \"quoted\" with noise and dashes-like-this"

### Real OCR Scenario

Original: "(77

-

l23

-

4S6

-

0789) with extra spaces"

Expected after patterns:

"(77-123-456-0789) with extra spaces"

## 5. EDGE CASES

### Empty and whitespace

Original: ""

Expected: ""

Original: " "

Expected: ""

### Numbers that should NOT be affected

Original: "1277-123-456-789" (77 is part of larger number)

Expected: "1277-123-456-789" (unchanged)

Original: "77" (incomplete part number)

Expected: "77" (unchanged)

### Mixed content

Original: "Before 77-111-222-333 middle text 77

-

444

-

555

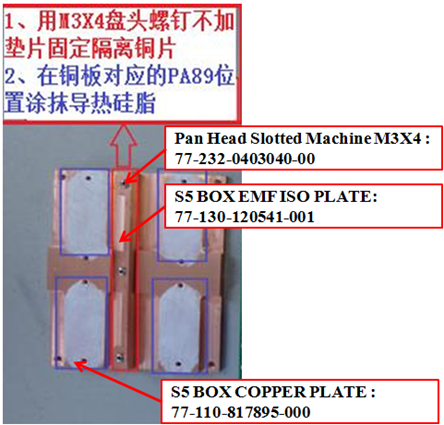
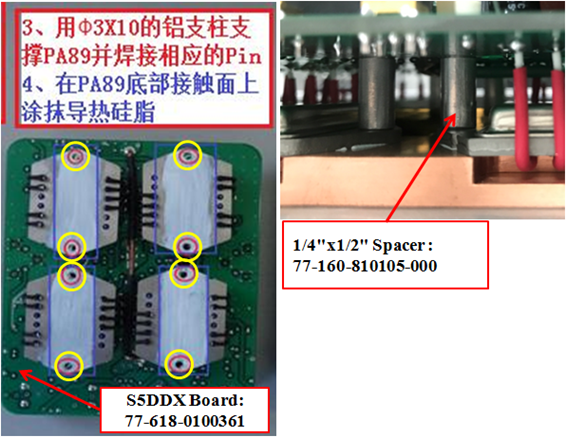
-

666 after"

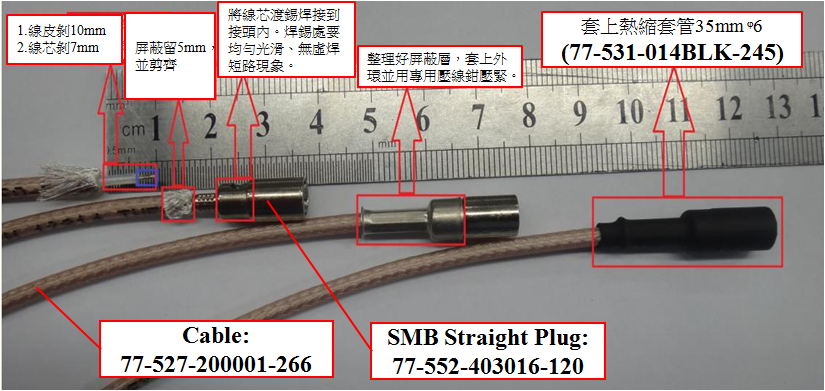
Expected: "Before 77-111-222-333 middle text 77-444-555-666 after"

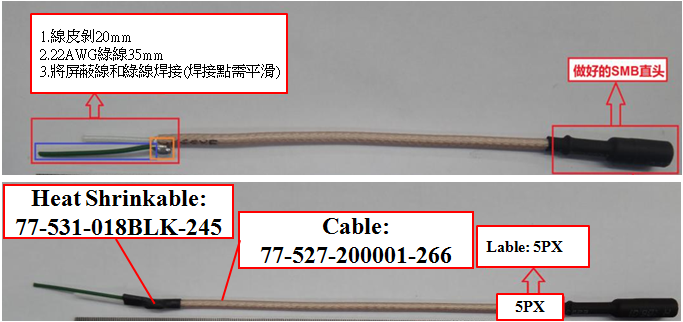
### PA89製作

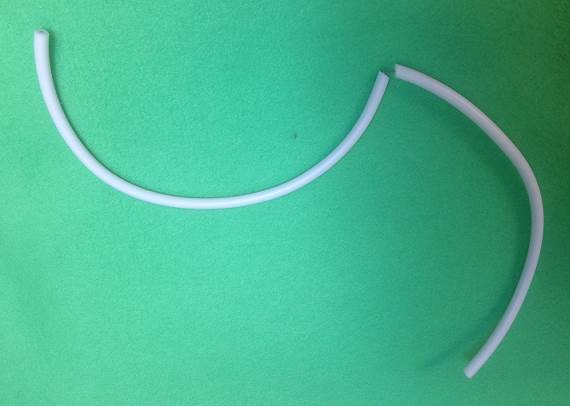




#### 







**4022-527-0000001-00 Hight Voltage Wire, White, 20AWG(420mm), Peel off from the middle(10mm)**



**77-245-0004040-00**

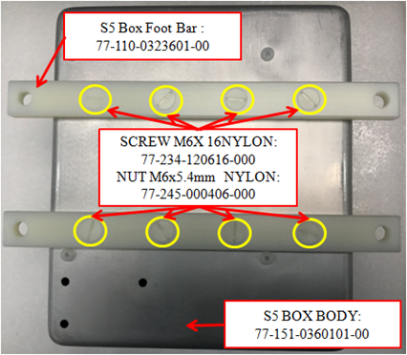
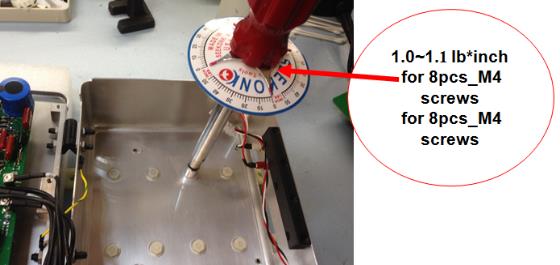
**Hex Nuts, Metric, Nylon, M4\*0.7, 6.90mm \*3.00mm**

**77-160-810470-000**

**Thread-Hex M-F Spacer M4x14 Nylon**

**77-160-810466-000**

**Thread-Hex M-F Spacer M4x6 Nylon**

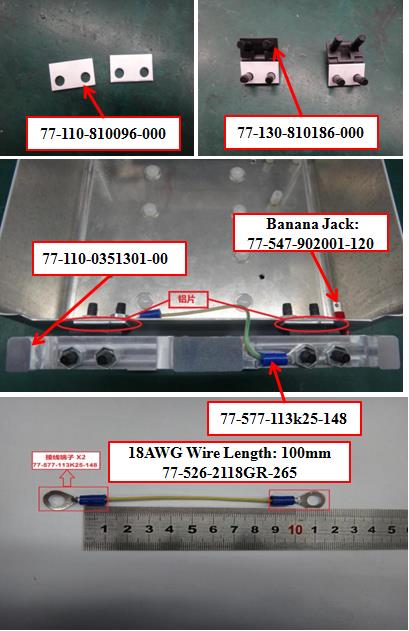


**S5BOX Body**

[**77-151-0301701-00**](https://app.bom.com/items/detail-spec?item_id=30363169&version_id=1262188677)

**2**

**1**



**4**

