

¹ ATLAS ピクセル検出器電荷較正の自動補完と
² 次世代器量産管理システムの開発

³ 東京工業大学 理学院物理学系物理学コース 陣内研究室
木下怜士 (20M00395)

⁴ 2022年1月7日

¹ Abstract

² abstract

概要

世界最高エネルギーでの陽子衝突加速器 LHC で新物理の発見を目指す ATLAS 実験のピクセル検出器においては、電荷較正の結果にデータの欠損や較正の失敗が含まれると、実測およびシミュレーションに影響を及ぼすため、較正結果を評価し再較正を行う必要がある。本研究では、再較正の際に適切な欠損の補完処理を行うよう、例外をアルゴリズムとして抽出・処理する自動解析ツールを開発した。

また、LHC 高輝度化に向けた ATLAS 検出器アップグレードのため、新型ピクセル検出器の開発および量産の準備を進めている。検出器の品質管理のために、組立工程において様々な試験を行う。本研究では、効率の良い量産と統合されたモジュール選定のために、先行する読み出し試験についての管理機能に加え、外観鑑別や形状測定などの試験項目についての管理機能、モジュール登録機能、試験結果の共有機能の開発を行った。

1 目次

2 概要	i
3 第 1 章 次世代ピクセルモジュールの量産	1
4 1.1 次世代ピクセルモジュールの組み立て部品	1
5 1.2 次世代ピクセルモジュールの組み立て工程	2
6 1.3 品質試験	4
7 1.4 量産における試験結果管理	9
8 第 2 章 モジュール量産におけるデータベースシステム	10
9 2.1 量産に用いるデータベースの概要	10
10 2.2 ローカルデータベースの構造	11
11 2.3 モジュールの品質試験に必要な開発項目	14
12 第 3 章 試験結果データ管理システムの開発	17
13 3.1 品質試験結果の表示機能	17
14 3.2 品質試験結果の管理機能	21
15 3.3 中央データベースとローカルデータベースの同期機能	24
16 3.4 本章のまとめ	38
17 第 4 章 まとめ	40
18 4.1 まとめ	40
19 4.2 今後の課題	40
20 付録 A 非読み出し試験結果のブラウザ表示	41
21 A.1 非読み出し試験結果のブラウザ表示	41
22 A.2 ピクセル検出器の基本特性のブラウザ表示	41
23 参考文献	42
24 謝辞	44

1 第1章

2 次世代ピクセルモジュールの量産

3 RUN3 に向けた現行ピクセルモジュールの測定準備に加え、ATLAS では HL-LHC アップグレードに
 4 向けた内部飛跡検出器の総入れ替えのため、次世代ピクセルモジュールの開発が進められている。現在、
 5 ITk に搭載するピクセルモジュール量産の各組み立て工程における試験やそのシステム確立のため、試作
 6 器を用いたデモンストレーションが行われている。

7 日本では新型器量産の際に約 2000 個のモジュールを生産する予定である。次世代ピクセルモジュール
 8 の量産の際に、効率の良い量産と統合されたピクセルモジュール選定を行うため、品質試験結果を統合管
 9 理するシステムの開発が必要となる。

10 1.1 次世代ピクセルモジュールの組み立て部品

11 量産工程は、各組み立て機関に届いたセンサーと ASIC から作られるベアモジュールとフレキシブル基
 12 板の接着から始まる。本節では各部品の詳細について説明する。

13 1.1.1 ベアモジュール

14 ベアモジュールはセンサーと ASIC をバンプ接合することにより作られる。クアッドモジュールではセ
 15 ソンサー 1 枚に対して ASIC 1 枚、トリプレットモジュールではセンサー 1 枚に対して ASIC 1 枚から構成
 16 される。ベアモジュールは通過する粒子を検出する。センサーを通過した荷電粒子は電子・ホール対を生
 17 成し、それにより得られる信号を ASIC を用いて増幅・整形を行う。

18 1.1.2 フレキシブル基板

19 フレキシブル基板はセンサーの裏側に接着、およびワイヤー配線により ASIC と電気的に接続される。

20 フレキシブル基板の全体図を図 1.1 に示す。フレキシブル基板は、以下の 3 つの役割を持つ。

21 フレキシブル基板は、以下の 3 つの役割を持つ。

- 22 • ASIC からの信号輸送

23 センサーから得られた信号は ASIC で増幅・整形され、フレキシブル基板に送られてくる。フレキ
 24 シブル基板は送られてきた信号を後段の PC へ送る。

- 25 • 電源の供給

26 外部からの電源を、センサーと ASIC に供給する。センサーには、空乏領域を増加させるために

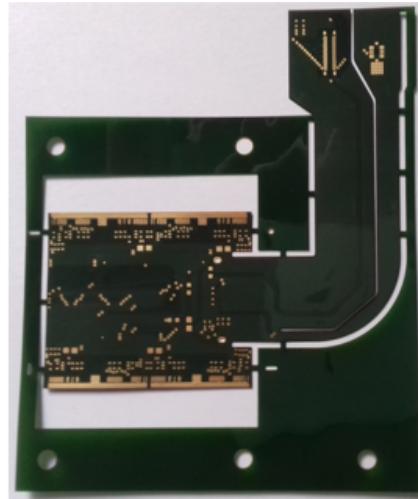


図 1.1 フレキシブル基板の全体図 [12]。

1 100 V 程度の HV(High Voltage) をかける。ASIC には、電源供給のために 5.6 V 程度の LV(Low
2 Voltage)

- 3 • モジュールの制御システム (DCS: Detector Control System)

4 モジュールの温度測定のために 2 つの NTC(Negative Temperature Coefficient) が配置されて
5 いる。

6 1.1.3 モジュールキャリア

7 モジュールキャリアはモジュールの運搬の際や品質試験を行う際に、モジュールを保護用の容器である。
8 組み立てられたモジュールは ASIC とフレックス基板を繋ぐワイヤー部やセンサーの部分等が剥き出
9 しになっているため、そのままの状態で品質試験を行うのはモジュール破損のリスクを伴う。モジュール
10 キャリアでモジュールを保護することにより、安全に品質試験を行うことができる。

11 また、モジュールキャリアの別の役割として、モジュール周囲の湿度環境を一定に保つことが挙げられる。
12 運転時に想定される最低温度は -45°C のため、品質管理試験ではペルチェ素子を用いた温度制御装
13 置^{*1}を用いて最低 -45°C までモジュールの周囲温度を下げる。その際、ピクセルモジュールに結露が
14 発生すると損傷のリスクを伴う。そのため、キャリア内に乾燥窒素ガスを流し込むことで氷点下における
15 ピクセルモジュールへの結露を防いでいる。

16 1.2 次世代ピクセルモジュールの組み立て工程

17 次世代ピクセルモジュールの組み立て工程を図 1.2 に示す。組み立て工程ではフレキシブル基板とベア
18 モジュールの接着から始まり、ワイヤー配線、パリレン高分子によるコーティング、ワイヤー保護を行い
19 ピクセルモジュールが完成する。その後、温度サイクル試験および低温耐久試験において、運転時に想定
20 される温度環境において組み立てたモジュールが運用できるかの試験を行う。本節では、組み立て工程、
21 およびモジュールの温度耐久についての試験についての説明を示す。

*1 KEK における次世代ピクセルモジュールの量産では、東工大を中心に関発している温度制御システムを用いる。

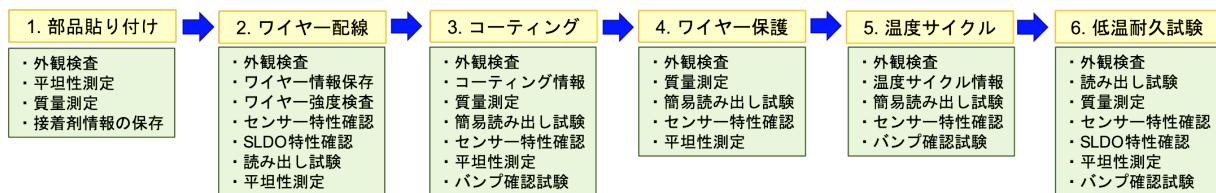


図 1.2 ピクセルモジュールの組み立て工程。

1 ベアモジュール・フレキシブル基板の接合

モジュールの組み立て工程は、組み立て機関に輸送されたベアモジュールとフレキシブル基板の接合から始まる。輸送された各部品の受け取り時の品質試験を行った後、ベアモジュールとフレキシブル基板の接合を行う。専用治具を用いて行うことにより、フレックス基板の位置の交差は $\pm 50 \mu\text{m}$ 、平面度は $25 \mu\text{m}$ の精度で接合を行うことができる。

6 ワイヤー配線

フレキシブル基板と ASIC を電気的に接合し、電源の供給や、ASIC からの信号を読み出すため、フレキシブル基板と ASIC 間をワイヤーで接続する。この組み立て工程をワイヤー配線と呼ぶ。ワイヤーは直径 $25 \mu\text{m}$ でのアルミ製であり、1 モジュールに対して約 500 本用いられる。ワイヤー配線後からは、モジュールの電気的な読み出しが行うことができるため、これ以降の全ての組み立て工程では読み出し試験を行い正常に動作するかの確認を行う。

12 パリレンコーティング

モジュールのセンサーと ASIC の端の部分での放電を防ぐこと、湿気や化学物質からの保護を目的としてパリレンコーティングを行う。パリレンはパラキシリレン系ポリマーの略である。パリレンは結晶性が高く絶縁耐力に優れ、周波数に依存せず低い誘電率・誘電正接特性を持っており、湿気や腐食性ガスへの耐性も併せ持つ。

17 ワイヤー保護

ワイヤーは直径 $25 \mu\text{m}$ と非常に細いため、力が加わると損傷してしまう可能性がある。ITk を実装する際、モジュールとワイヤーの距離は 2 mm 程度のため、モジュールのケーブルがワイヤーに触れてしまい読み出しが正常にできなくなる恐れがある。このような問題を避けるため、構造体を用いて、ワイヤーを保護する。

22 温度サイクル

組み立てたモジュールに対して、ITk 実装後にされる特異的な温度変化のサイクルを行い、その後もモジュールが正常な応答をするか試験をする。温度変化の際、モジュールの部品間の熱膨張の違いにより熱応力が生じ、それが原因でバンブ接合部に剥がれが生じてしまうことがある。このような温度サイクルによるモジュールの損傷がないことを確認する必要がある。

ITk の運転の切り替えが年間 10 回以上あるため、10 年間の運転を想定すると 100 回以上の熱サイクルにさらされる。そのため、量産における品質試験では、動作温度範囲 $-45^{\circ}\text{C} < T < 40^{\circ}\text{C}$ の温度サイクルを 10 回、 $-55^{\circ}\text{C} < T < 60^{\circ}\text{C}$ の温度サイクルを 1 回を行う。これらの温度サイクルの後にモ

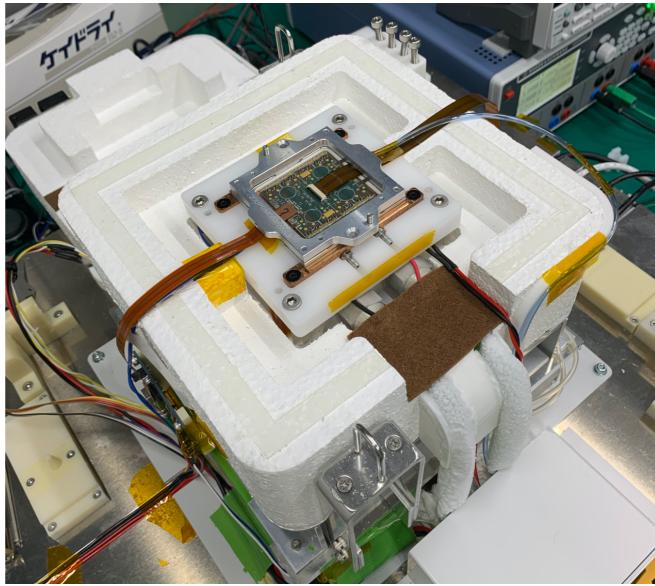


図 1.3 読み出し試験のセットアップ。この図は、温度制御筐体の蓋を開けた状態であり、測定時には蓋を閉めてモジュールの周囲を密閉し測定を行う。銀色のフレームがモジュールキャリアであり、その中にモジュールが設置されている。

- 1 モジュールが正常に動作するかを確認するため、ASIC 回路読み出し試験等を行う。モジュールの周囲温度
- 2 を変える際には、恒温槽を用いて行う予定である。

3 低温耐久試験

- 4 常温において ITk 運転におけるピクセルモジュールの周囲温度は $-15^{\circ}\text{C} < T < 0^{\circ}\text{C}$ である。組み立
- 5 てたモジュールが低温環境下において長時間正常に動作することを確認する試験が低温耐久試験である。
- 6 低温耐久試験では、温度制御筐体を用いてモジュールの周囲温度を -15°C に保ちつつ ASIC の回路読み
- 7 出し試験を行う。読み出し試験は 1 時間に 1 度行われる。
- 8 長時間放置しつつ読み出し試験を行うため、インターロックシステム、機器の遠隔制御、温度制御筐体
- 9 の遠隔監視等の技術が必要となる。

10 1.3 品質試験

- 11 モジュールの各組み立て工程の後に、モジュールが正常に動作するかを確認するために品質試験を行
- 12 う。図 1.2 に示したように、モジュールの外観検査は全ての工程で行われ、ASIC の回路読み出し試験は
- 13 ワイヤー配線後の全ての工程で行われる。本節では、各品質試験項目の詳細を以下に示す。

14 1.3.1 外観検査

- 15 モジュールの表面（フレックス基板側）をカメラを用いて撮影し、モジュールに損傷や汚れ等がないこ
- 16 とを目視で確認する。特に、ASIC とフレックス基板を電気的に接合するためのワイヤーの接着位置が正
- 17 しいか、断線がないかを確認することが重要である。目視で確認する際は、モジュール全体の高解像度画
- 18 像を 36(縦横 6×6) 分割して得られる拡大画像を用いて細かく検査を行う。また、ワイヤー部分について
- 19 は約 500 本のワイヤーを目視で漏れなく検査することは困難且つ労力を伴うため、ワイヤーの断線や接続

1 部分のずれを自動で検知するアルゴリズムの開発が進んでいる。

2 1.3.2 平坦性測定

3 モジュール上の3次元位置座標を取得することにより、歪み具合や接着剤の厚み等を計算することができます。これにより、接着時のずれや接着剤の厚みに問題やモジュール輸送時にモジュールへの損傷等を確認することができる。

6 1.3.3 質量測定

7 質量測定では、モジュール全体の質量を測定する。各工程における質量の差を計算することにより、接着剤の質量やワイヤーの合計質量等を取得することができる。

9 1.3.4 ワイヤー強度検査

10 ワイヤー配線により接続したワイヤーの強度を調べるため、専用の機械を用いてワイヤー部分に負荷を与える。

12 1.3.5 センサー IV 特性

13 センサーの電流-電圧特性を調べることにより、モジュール製造工程におけるセンサーの損傷や HV のショートを確認することができる。プラナーセンサーでは、漏れ電流が 80 V で 2 μ A、降伏電圧が 120 V、3D センサーについては漏れ電流が 25 V で 2 μ A、降伏電圧が 35 V 程度となるのが想定される結果である。また、測定は ASIC からの消費電力による発熱を避けるため、ASIC への LV を切った状態で行われる。

18 1.3.6 SLDO VI 特性

19 ITk 実装時には、モジュールを直列に並べて電源の供給を行う。そのため、各モジュールに対する電源は定電圧ではなく、供給電圧はつなげるモジュールの数に依存してしまう。ASIC 回路内部で一定の電圧を供給するために、SLDO(**S**hunt **L**ow **D**rop **O**ut) という制御回路を用いる。SLDO 制御回路が供給電流の一部を用いてデジタル回路、アナログ回路の動作電圧を生成し、余剰電流はグランドに捨てられる。ASIC には二つの SLDO 制御回路が搭載されている。1つはデジタル回路用、もう1つはアナログ回路用に用いる。図 1.4 に ASIC への電流と、入力電圧 V_{in} 、アナログ回路の出力電圧 V_{analog} およびデジタル回路の出力電圧 $V_{digital}$ の例を示す。この結果から、SLDO 回路によりアナログ回路およびデジタル回路からの出力電圧は一定であり、制御回路が正常に動作していることがわかる。

27 さらに、SLDO VI 特性についての品質試験を行う際には、ASIC が低温においても正常に動作するかの試験も行う。モジュールの周囲温度を -35°C にし、デジタル回路の読み出し試験を行いモジュールが低温環境で動作することを確認する。正常に動作しない場合は 15°C ずつ温度を上げて再び試験を行い、正常に動作を始める温度を記録する。温度を記録する際には、表 1.1 に示すように各温度に対する階級を表す数値を用いて値の入力を行う。

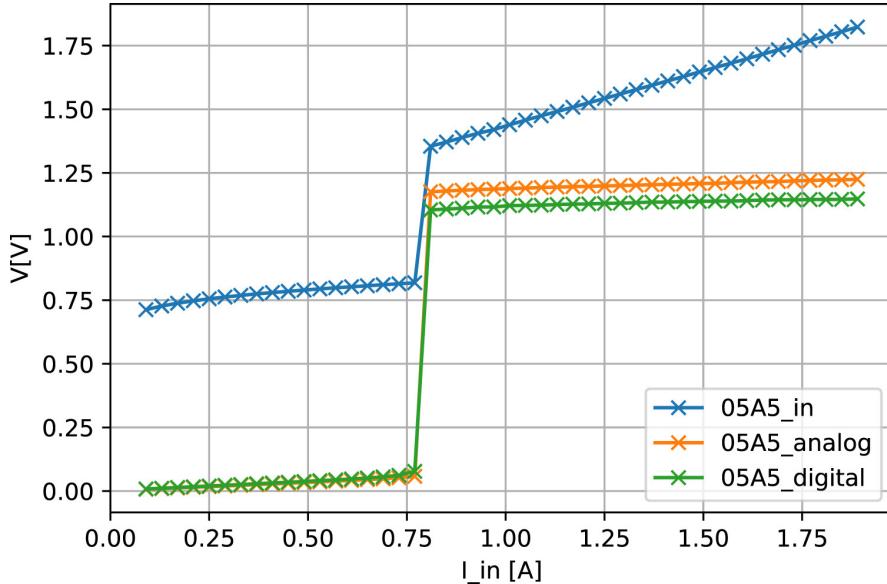


図 1.4 SLDOVI

表 1.1 モジュール起動温度に対する階級値。

温度 [°C]	階級値
-35	1
-35 < T ≤ -20	2
-20 < T ≤ -5	3
-5 < T ≤ 10	4
10 < T	5

1.3.7 読み出し試験

モジュールに通電し、正常に読み出しができるか確認する。読み出し試験は ITk 運転時の温度環境を想定し、モジュールの周囲温度を変えつつ試験を行う。そのため、読み出し試験の際には低温耐久試験と同様に図 1.3 の温度制御筐体を用いて試験を行う。設定温度は、低温におけるモジュール起動試験では -35°C (正常に起動できない場合は 15°C ずつ温度を上げて試験)、Threshold 測定や ToT 測定等の通常の読み出し試験では $-20^{\circ}\text{C}, 20^{\circ}\text{C}$ である。

読み出し試験に用いる DAQ(Data AcQuisition) として、YARR(Yet Another Rapid Readout) を用いる。YARR とは ASIC 読み出しそうに開発されている。YARR を用いて行う、モジュールの読み出し試験の項目を以下に示す。

- Digital スキャン

各ピクセルについてのデジタル回路の応答を確認する。デジタル回路に試験用パルスを入射し、信号の応答数を測定する。

- Analog スキャン

各ピクセルについてのアナログ回路の応答を確認する。アナログ回路に試験電荷を入射し、信号の応答数を測定する。

表 1.2 ピクセル解析の評価基準一覧 [23]

評価名	読み出し試験項目	評価基準
Digital Dead	Digital Scan	Occupancy < 1% of injections
Digital Bad	Digital Scan	Occupancy < 98% or Occupancy > 102% of injections
Merged Bump	Analog Scan	Occupancy < 1% of injections
	Crosstalk Scan	Occupancy < 80% of 25 ke injections
Analog Dead	Analog Scan	Occupancy < 1% of injections
Analog Bad	Analog Scan	Occupancy < 98% or Occupancy > 102% of injections
Tuning Failed	Threshold Scan	s-curve fit failed
	ToT Test	ToT response is 0 or 14 BCs
Noisy	Noise Scan	Occupancy < 10^{-6} hits per BC
Disconnected Bump	Source Scan	Occupancy < 1% of mean Occupancy
High Crosstalk	Crosstalk Scan	Occupancy > 0 with 25 ke injection

1 ● Threshold スキャン

2 各ピクセルの Threshold を確認する。試験電荷を用いて S カーブのフィッティングを行い、
3 Threshold 値やノイズを測定する。

4 ● ToT スキャン

5 一定の試験電荷を各ピクセルに 100 回入射させ、その試験電荷に対する ToT の値の測定を行う。

6 ● Noise スキャン

7 各ピクセルのノイズを確認する。試験電荷を用いず、クロックによるトリガーで測定を行い、応答
8 率を求める。

9 ● Source スキャン

10 放射線を照射し、各ピクセルの応答を確認する。これにより、シリコンセンサーやセンサーと
11 ASIC 間の接続の不具合等も確認することができる。

12 また、これらのスキャン項目に加え、ピクセルのチューニングも行うことがある。

13 ● Global threshold tuning

14 ASIC 全体のピクセルにおける Threshold を一括チューニングする。

15 ● Pixel threshold tuning

16 各ピクセル毎に Threshold 値チューニングする。

17 ● Global ToT tuning

18 ASIC 全体のピクセルにおける ToT 値を一括チューニングする。

19 ● Pixel ToT tuning

20 各ピクセル毎に ToT 値をチューニングする。

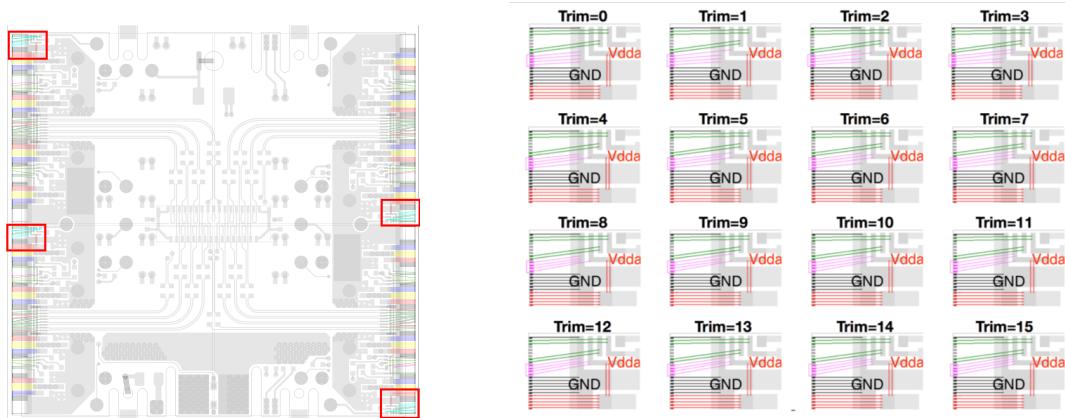


図 1.5 クアッドモジュールの Iref Irim 部分を表す部分 (左図) とワイヤーの配線と Iref 値の関係 (右図)[23]。ピンク色のワイヤーの配置により 4bit の Iref 値を表すことができる。

1.3.8 モジュール特性

モジュールの特性として、以下のようのものが定義されている。

- FE chip version

モジュールの部品となっている ASIC の種類を入力する。現在行っている試作器は RD53A であり、ITk 向けたモジュール量産では ITkpix_v2 を用いる予定である。

- Thickness

センサーの厚さの情報を入力する。この特性に入力する値としては”thin”と”thick”があり、それぞれセンサーの厚みが 150 μm と 300 μm である。

- Roof

モジュールのワイヤー部を保護する構造体の有無を記録する。

- IrefTrim

全ての DAC^{*2}(Digital Analog Converter) は”IREF”と呼ばれるグローバルなレファレンスから 4 μA の電流を生成する。Iref の値はワイヤー配線の際に 4bit の値で決められる。ワイヤー配線と Iref 値の関係を図 1.5 に示す。

- Pull-up register

プルアップ抵抗とは、電子回路の入力端子に接続される抵抗であり、スイッチがオフの状態の時に電圧が一定の値を安定して維持できるようにするためのものである。アナログ回路への電源 $VDDA$ とプルアップ抵抗値についての関係を表 1.3 に示す。

- PCB-Bare Orientation

ベアモジュールとフレックス基板の向きが正しいか確認する。

上記に示したモジュールの特性のうち、FE チップ version 等の情報はモジュールの作成と同時に決まるものである。しかし、Iref 値やプルアップ抵抗値についての特性はワイヤー配線後に初めて確認できるものであるため、品質試験を行う際に並行して確認する必要がある。

また、ベアモジュールの見た目は、180° 回転対称になっている。そのため、フレックス基板とベアモ

^{*2} D/A コンバーターとも呼ぶ。

表 1.3 プルアップ抵抗値と電源の関係。

VDDA [V]	Pull-up Resistor が必要か	Pull-up Resistor の値 [Ω]	VDDA の増加量
$VDDA \leq 1.09$	Yes	150	0.1
$1.09 < VDDA \leq 1.14$	Yes	300	0.05
$1.14 > VDDA$	No	n/a	n/a

- 1 ジュールの貼り付け工程で誤って 180° 回転した状態で接合してしまうことがある。この誤りは、ワイヤ配線後に行う読み出し試験の際に初めて確認される。その場合は、ペアモジュールとフレックス基板の向きについての特性を書き直す必要がある。

4 1.4 量産における試験結果管理

5 ITk のために世界各地の組み立て機関においてモジュールの量産およびそのための品質管理試験を行
6 う。各組み立て機関では $\mathcal{O}(100) \sim \mathcal{O}(1000)$ のモジュールの量産を行う予定であり、最も量産数が多い
7 のは日本の高エネルギー加速器研究機構 (KEK) における約 2000 個のモジュールの量産である。各モ
8 ジュールに対して、各組み立て工程および温度耐性についての品質試験を行い合計 30 個程度の試験を行
9 い、さらにワイヤ配線をした後に確認できるモジュール特性についてのデータも適切に管理する必要が
10 ある。

11 ITk の製造に関するモジュール情報や品質試験の結果は、チェコに設置されている中央のデータベー
12 スに保存する必要がある。そのため、各研究機関においてモジュール情報と品質試験結果を統一的に管理
13 する必要がある。しかし、各モジュールにおいて 30 程度の試験項目がありデータの形式が異なること、
14 さらに各研究機関で組み立てるモジュール数が多いことから、各研究機関独自のシステムを用いてデータ
15 管理すると以下のような問題が想定される。

- 16 ● データの不整合

17 各研究機関において独自のシステムを用いると、試験結果の管理方法が異なることから他の機関との
18 結果の比較が困難になる。モジュールを他研究機関に送った際に、輸送中にモジュールに損傷がないことを
19 確認するために受け取り試験（レセプション試験）を行う。輸送前後の試験結果を比較
20 を行うが、その際に試験結果の形式が異なると比較前に結果を整形する必要があり、試験機関の数
21 だけ整形しなければならないため非常に面倒である。

- 22 ● データの重複

23 試験結果やモジュール情報を共有した際、既にそのデータが存在すると使用している PC のディ
24 レクトリ構造の違いにより別のデータと認識してしまいデータの重複が発生してしまう可能性が
25 ある。そのため、別の研究機関や異なるサーバーでデータを共有する際に、適切に管理する必要が
26 ある。

27 上記の問題を解決するために、各モジュール組み立て機関において適切にデータ管理を行うことを目的
28 として、データベースシステムの開発を行っている。このシステムについて、次章において説明する。

1 第2章

2 モジュール量産におけるデータベースシ 3 ステム

4 前章に示したように、次世代モジュールの量産の時にデータを適切に管理するためにデータベースシ
5 テムを開発している。

6 本章ではデータベースシステムの全体像について説明する。

7 2.1 量産に用いるデータベースの概要

8 量産に用いるデータベースは、ITk の製造に関する情報を全てを記録するために開発されている中央
9 データベースと、各研究機関でモジュール量産を管理するためのローカルデータベースがある。中央デー
10 タベースとローカルデータベースの設置位置を図 2.1 に示す。本節ではそれについての概要を示す。

11 2.1.1 中央データベース

12 中央データベースは ITk に実装するためのピクセルモジュール・ストリップ検出器の量産についての
13 データを管理することを目的として開発されている。ユニコーン大学が中心に開発を行っており、チェコ
14 にデータベースサーバーが設置されている。



図 2.1 中央データベースとローカルデータベースの設置位置。ローカルデータベースは全てのモ
ジュール組み立て機関に設置される。

1 ITk に実装される検出器とその構成部品は 50 万個程度であり、中央データベースではそれら全ての量
2 産についてのデータを管理する。量産の組み立て工程で行われる試験結果は全てを記録するのではなく、
3 ある品質管理試験における良い結果を一つのみを保存しておく。

4 中央データベースに保存されたデータは、ITk に配置するピクセルモジュールの選別に用いられる。 $|\eta|$
5 が小さい領域は通過する粒子の密度が高くなることが想定されるため、高い放射線量が予想される。その
6 ため、ITk に搭載する際にはなるべく品質の良いモジュールを搭載する予定である。また、品質の悪いモ
7 ジュールを配置する領域が固まると、その領域を通過する飛跡の再構成が難しくなり、物理解析の際に不
8 良データと判別されてしまうことがある。そこで、品質の悪いモジュールの領域が固まらないように配置
9 する必要がある。この選別に用いる参考値として、中央データベースに保存されているデータを用いる予
10 定である。

11 また、運転前後で検出器に損傷が起きていないかを確認するためにもこれらの値を用いる。運転前後の
12 読み出し試験やその他の試験の結果の比較を行うことにより、センサーへの放射線損傷や不良ピクセルの
13 推移等を確認することができる。ITk は約 10 年間の運転が予定されており、少なくとも 20 年間は検出
14 器についてのデータが利用できるようにしておく予定である。

15 2.1.2 ローカルデータベース

16 ローカルデータベースは各研究機関に設置され、モジュールの組み立て工程における品質試験結果を管
17 理するためのシステムである。ローカルデータベースは各研究機関におけるモジュールの品質試験結果を全
18 て記録しておくという点で中央データベースとは異なる。

19 先述したように、モジュールの組み立て工程およびそれぞれの工程の際に行われる品質試験の数は非常
20 に多く、それぞれのモジュールに対して適切に試験を管理する必要がある。モジュールの量産は各研究機
21 関において $\mathcal{O}(100) \sim \mathcal{O}(1000)$ 個であり、1 つのモジュールについて 30 項目程度の品質試験を行うため、
22 $\mathcal{O}(10000)$ 個の試験結果を適切に管理する必要がある。各組み立て機関において、適切にデータを管理し、
23 かつピクセルモジュールの量産を円滑に行うことができるようサポートするためにローカルデータベー
24 スというシステムを開発している。ローカルデータベースに対する要求は以下のようなものである。

- 25 • 検出器情報や試験結果の情報を他研究機関との整合性を保ちつつ管理すること。
- 26 • ピクセルモジュールの量産を円滑に行うことができるようサポートすること。
- 27 • 中央のデータベースにモジュール量産に関わるデータを同期すること。

28 以上の目標を達成するために、東工大を中心にローカルデータベースの開発が進められている。次節に
29 おいて、ローカルデータベースの構造と、先行研究における開発項目について示す。

30 2.2 ローカルデータベースの構造

31 ローカルデータベースの構造の全体像を図 2.2 に示す。ローカルデータベースは品質試験結果を保有し
32 ておくためのデータベースである MongoDB と、データベースに保存されたデータを操作および閲覧す
33 るためのウェブアプリケーションから構成される。MongoDB への試験結果の登録は、読み出し試験につ
34 いては専用のソフトウェアである YARR を用いて行い、それ以外の品質試験については大阪大学および
35 都立大学によって開発が進められている品質試験結果登録用 GUI を用いて行う。本節では、MongoDB
36 とウェブアプリケーションについて説明する。

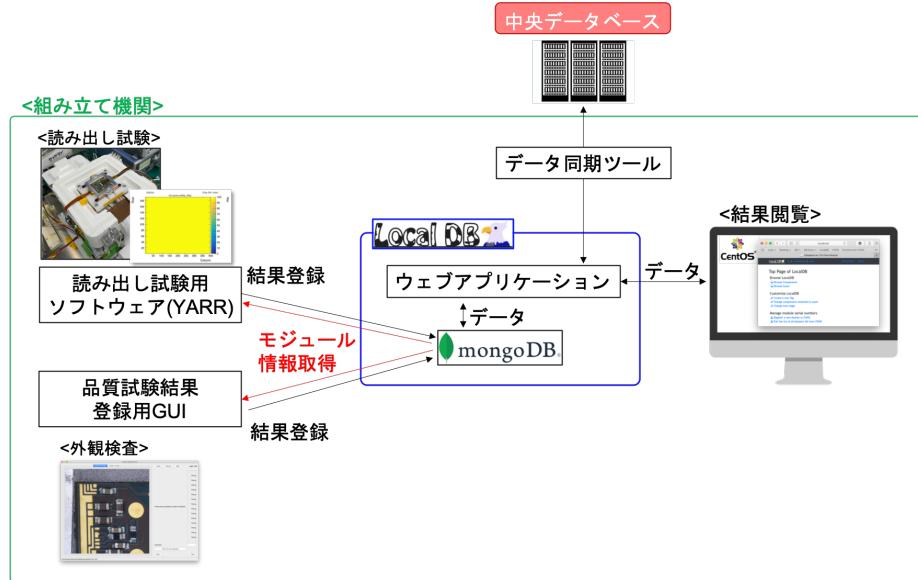


図 2.2 ローカルデータベースの全体像。ローカルデータベースは MongoDB とウェブアプリケーションから構成され、品質試験結果を管理するための複合的な機能を持つ。

2.2.1 MongoDB[25]

MongoDB はアプリケーションの開発や拡張を簡単に行うことができるよう設計されている、オープンソースのドキュメントデータベースである。ドキュメントデータベースは、非リレーションナルデータベース (NoSQL) の一種であり、1つのデータをドキュメントと呼び、具体的にはデータを JSON^{*1}のようなドキュメントとして保存する。MongoDB では、**BSON** という、JSON に非常によく似た形式のデータを扱うが、binary 表記したデータを持つため JSON よりも多くのデータの型を保存することができる。

MongoDB では、データベース、コレクション、ドキュメント（オブジェクト）の概念を用いてデータを管理する。各ドキュメントはコレクションという枠の中に格納され、さらに各コレクションはデータベースによって包括される。各コレクション間の関係を自由に定義することができ、データベースが構造化し、階層的な性質を持たせてデータを管理することができる。

各ドキュメントは**オブジェクト ID** という 12 byte の ID によって管理される。オブジェクト ID はデータ生成時の時間情報によって決まる値 (4 byte)、データベースマシンによってランダムに生成される値 (5 byte)、データベースマシンのカウンターによって生成される値 (3 byte) により自動生成される。そのため、複数のドキュメントが同一のオブジェクト ID を持つことはなく、これにより識別を行うことができる。また、オブジェクト ID を用いることにより、異なるコレクションにおけるドキュメント間の関連付けを行い、データベース内の構造を柔軟に設計することができる。

Code 2.1 MongoDB のドキュメントの例

```

18
19     1  {
20     2      {
21     3          "_id": ObjectId("6038c960b9a87924947df638"),

```

^{*1} JSON とは JavaScript Object Notation の略で、データ記述言語の一種である。ある key と value を対応させることにより、データを取り出すことができる。ウェブアプリケーションでデータを転送する場合に使われることが多い。

```
1     4      "year": 2015,
2     5      "title": "The_Big_New_Movie",
3     6      "info": {
4     7          "plot": "Nothing_happens_at_all.",
5     8          "rating": 0
6     9      }
7    10  }
8  11 }
```

10 ローカルデータベースシステムの開発において、MongoDB を使う利点を以下に示す。

- 11 ● スキーマレスで、ドキュメント構造を動的に変更することができる。

12 品質管理試験は、試験項目により保存するパラメータが異なる。格納形式に柔軟性のある NoSQL
13 の MongoDB を用いることにより、同一コレクションに異なる形式の試験結果を保管しておくこ
14 とができる。また、現在は試作器を用いて、次世代器の品質試験に向けた実験装置の準備や試験の
15 レビューを行っている。そのため、最終的にデータベースに残る結果が変わることがある。この
16 際、NoSQL の MongoDB では変更点が最小に抑えることができるので、開発スピードを早くでき
17 ることが多い。

- 18 ● ObjectId により、中央データベースとの整合性を保ちやすい。

19 中央データベースで管理するデータは 12 byte の ID を保持している。そのため、中央データベー
20 スにおける ID とローカルデータベースの MongoDB における ObjectId を関連付けて管理するこ
21 とが可能になる。

22 これらの利点から、MongoDB を用いてローカルデータベースの開発を行っている。ローカルデータ
23 ベースにおけるモジュール情報および品質管理に用いるコレクションを表 2.1 に示す。ローカルデータ
24 ベースにおいて、**localdb** と **localdbtool** の 2 つのデータベースを準備している。**localdb** はピクセル
25 モジュール情報および品質管理試験結果等の各組み立て機関から中央データベースに共有する情報を保
26 有し、**localdbtool** はユーザー情報、中央データベースからダウンロードした組み立て機関のリストやピク
27 セルモジュールの構成要素等の中央データベースに共有しない情報や組み立て機関に依存しない情報を保
28 有する。各コレクション間の関係を図 2.3 に示す。このように各コレクションに保存する情報を関連付け
29 て管理することにより、品質試験結果やそれに関する情報をデータベースから取り出しやすくなる。

30 2.2.2 ウェブアプリケーション

31 各研究機関において、ローカルデータベースを使用するために、試験者がデータベースに保存されてい
32 る品質試験結果を閲覧および管理を簡単にできる必要がある。しかし、MongoDB はデータ構造が柔軟で
33 あり、使用用途に基づき多様なデータベース構造が考えられるため、データベース内のデータを表示・処
34 理するインターフェースは提供されておらず、必要に応じて適宜インターフェースを開発する必要があ
35 る。ローカルデータベースシステムでは、多様な利用者がデータベース内のデータを閲覧、操作を実現で
36 きることを目標として、Python のウェブアプリケーションフレームワークである Flask を導入している。

37 ウェブアプリケーションの処理を図 2.4 に示す。利用者が見るウェブブラウザのインターフェイスは
38 html で書き、ボタンを押したら Python のバックエンド側にパラメータが送信される。受信したパラメー
39 タをもとに Flask で処理を返し、html 側で表示を行う。Flask が受け取った処理を行う際、MongoDB

表2.1 ローカルデータベースのコレクション一覧。

データベース名	コレクション名	保存情報
localdb	component	モジュール情報、FEチップ情報
	childParentRelation	FEチップとモジュールの関係性
	testRun	読み出し試験結果
	componentTestRun	componentとtestRunの関係性
	user	読み出し試験実施者
	institute	読み出し試験実施場所
	comments	部品、試験結果についてのコメント情報
	QC.module.status	各モジュールに対する組み立て工程及び選択された試験結果
	QC.result	品質試験結果
	QC.prop.status	ワイヤー配線の後に決まるモジュール特性の書き換え情報
localdbtools	QC.status	組み立て工程及び試験項目
	QC.module.types	モジュールの構成部品
	viewer.user	ローカルデータベースのユーザ情報
	viewer.query	読み出し結果キーワード、検索機能実行時に使用
	viewer.tag.docs	モジュールや試験結果に付けるタグの情報

- ¹ に保存するデータを操作するために、PythonのパッケージであるPyMongoを用いている。また、ウェブブラウザにおいて、動的な処理を行う際にはJavaScriptを利用している。この流れにより、ウェブブラウザから、データベースに保存されているデータを簡単に取り扱うことができるよう設計している。

⁴ 2.3 モジュールの品質試験に必要な開発項目

- ⁵ 東工大を中心開発が進められており、これまででは読み出し試験を中心に開発が進められている。モ⁶ジュールの品質管理に必要な開発項目および開発状況を以下に示す。

⁷ 1. データベース構造の設計

⁸ 図2.3に示したように、MongoDBにおけるコレクションを定義し、データを取り出しやすく工夫⁹している。読み出し試験についてのデータベース構造は先行研究[3, 4]によって定義され、非読み出¹⁰し試験およびモジュールの組み立て工程を管理するためのデータベース構造は先行研究[2]を行った奥山氏と私が開発を行った。

¹¹ 2. 試験結果管理機能

¹² 2-1. 読み出し試験

¹³ 読み出し試験の結果の閲覧および解析機能は先行研究[2, 3]によって開発が行われた。読み出¹⁴し試験はピクセルの読み出し性能の不良判定するのに重要な試験であり、各組み立て工程で品¹⁵質に変化がないこと、あるいは不良ピクセルが発生した際にどの工程で問題があったかを発見¹⁶することが重要である。本研究では、各組み立て工程間で評価できる機能を追加した。これに¹⁷

LocalDB Structure

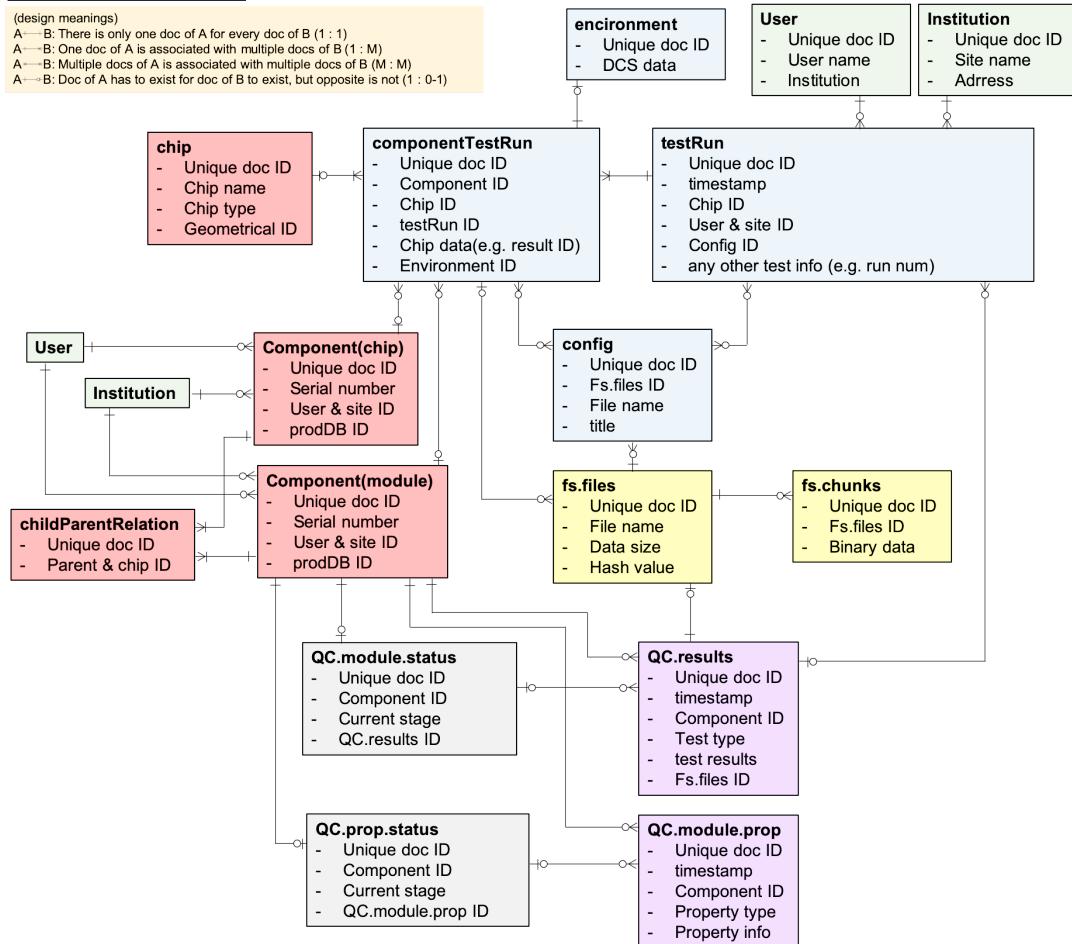


図 2.3 ローカルデータベースの構造。それぞれの四角はコレクションを表しており、緑色は使用ユーザーに関する情報、赤色は部品についての情報、青色はYARRを用いて登録した読み出し試験結果、紫は品質試験結果登録用GUIを用いて登録した結果についてのものである。また、灰色はモジュールの組み立て工程を管理するためのコレクションであり、黄色は読み出し試験データや画像データを管理するファイルシステムに関するコレクションである。直線はオブジェクトIDによるドキュメント間のリンクを示している。

<ウェブブラウザ>

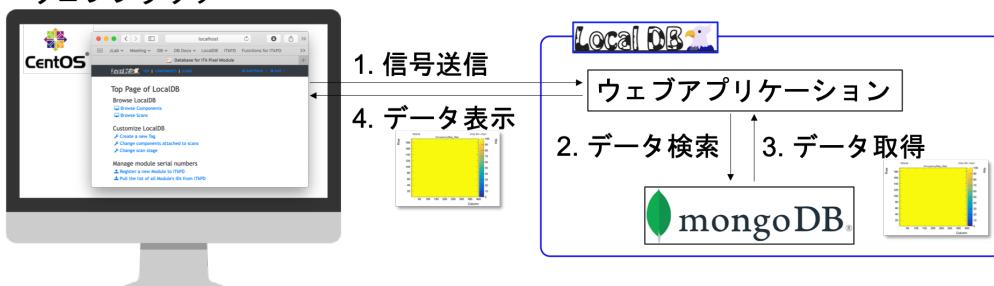


図 2.4 ウェブアプリケーションの処理の概念図。図中の数字は処理の順序を表し、ウェブブラウザから送信した信号に基づきウェブアプリケーション内で処理を行う。

について 3.1 に示す。

2-2. 非読み出し試験

非読み出し試験の閲覧機能はこれまで未開発であり、本研究においてこの機能を実装した。閲覧機能について、3.1 に示す。

3. モジュール組み立て工程管理機能

各組み立て機関で $\mathcal{O}(100) \sim \mathcal{O}(1000)$ 個のモジュールを量産するため、それぞれのモジュールについて品質試験が適切に行われたこと、各工程で全ての結果が揃っていることを担保する必要がある。この機能は先行研究 [2] によって開発が行われたが、モジュール特性の結果管理についてまだ未開発であった。これについて 3.2 に示す。

4. 中央データベースとの同期機能

これまで、モジュールを中央データベースからダウンロードする機能、および読み出し試験の結果を中央データベースと同期する機能の開発が先行研究 [2] によって行われてきた。しかし、モジュールを中央データベースへ登録、非読み出し試験の結果およびモジュールの組み立て工程を中央データベースと同期する機能については未開発であった。これについて、3.3.2 に示す。

1 第3章

2 試験結果データ管理システムの開発

3 本研究では、これまで読み出し試験用に開発されていたシステムの他に、ピクセルモジュールの品質
 4 管理の流れの全てをサポートできるように機能の作成を行った。本研究において開発した項目を以下に
 5 示す。

- 6 1. 品質試験結果の表示機能 (3.1 節)
 - 7 – 非読み出し試験結果の閲覧機能
 - 8 – ピクセルモジュール特性の閲覧機能
 - 9 – 読み出し試験結果比較機能
- 10 2. 品質試験結果の管理機能 (3.2 節)
 - 11 – ピクセルモジュール特性の選択機能
- 12 3. 中央データベースとローカルデータベースの同期機能 (3.3 節)
 - 13 – ピクセルモジュール情報の登録機能
 - 14 – 品質試験結果のアップロード機能
 - 15 – 品質試験結果のダウンロード機能

16 3.1 品質試験結果の表示機能

17 前章で示したように、多様な利用者がデータベース内のデータを閲覧、操作を実現できることを目標と
 18 してウェブアプリケーションの開発を行っている。これまで読み出し試験に特化した開発が行われてい
 19 たが、その他の品質試験結果を表示する機能は未実装であった。本研究では、品質試験結果登録用 GUI
 20 の開発者と協力し、ローカルデータベース内でのデータの保管方法と結果の閲覧機能の開発を行った。さ
 21 らに、先行研究で開発された読み出し試験結果の解析機能を拡張し、別の試験結果と比較する機能の開発
 22 を行った。

23 本節では、それぞれの機能の概要と各品質試験項目についての結果の閲覧機能について説明する。

24 3.1.1 非読み出し試験結果閲覧機能の概要

25 これまで読み出し試験結果についての開発を中心に行われており、その他の品質試験結果を表示する
 26 機能は未実装であった。ピクセルモジュール次世代器量産における品質試験を管理するためには、非
 27 読み出し試験結果を管理する機能が必要である。本研究では、非読み出し試験結果登録用 GUI である

- 1 QC-helper の開発者と協力し、ローカルデータベース内でのデータの管理方法と結果の閲覧機能の開発を行った。以下に詳細を示す。
- 2 表 2.1 に示したように、品質試験結果は MongoDB の QC.results というコレクションに保存される。
- 3 非読み出し試験結果の例を以下に示す。

Code 3.1 品質試験結果を表すドキュメントの例。以下は質量測定結果の一つを表している。

```

5   1 {
6
7     2   "_id" : ObjectId("6131a0532e16df12955d9c6d"),
8
9     3   "component" : "60b9dc51d978dc000b9232fc",
10
11    4   "user" : "kinoshita",
12
13    5   "address" : "Tokyo\u2014Institute\u2014of\u2014Technology",
14
15    6   "currentStage" : "MODULETOPCB",
16
17    7   "testType" : "MASS",
18
19    8   "sys" : {
20
21      9     "cts" : ISODate("2021-09-03T04:10:57.338Z"),
22
23      10    "mts" : ISODate("2021-09-03T04:10:57.338Z"),
24
25      11    "rev" : 0
26
27    },
28
29    13   "results" : {
30
31      14     "property" : {
32
33          15       "Scale_accuracy" : 0.5,
34
35          16       "Scale_accuracy_unit" : "g"
36
37      },
38
39      18     "mass_value" : 1,
40
41      19     "mass_unit" : "g",
42
43      20     "comment" : "hoge"
44
45    },
46
47    22   "dbVersion" : 1.01
48
49  }

```

先述したように非読み出し試験の結果は QC-helper によって MongoDB に登録される。全ての品質試験結果では上記のドキュメントの **results** を除く値を共通要素として持ち、品質試験項目により results に格納するパラメータが異なる。

画像や JSON file 等のファイルデータの保存は GridFS[25] と呼ばれるインターフェースが使用される。GridFS はサイズの大きいファイルの実体を 255 kB サイズのドキュメントに分割して保存する使用である。GridFS は、**fs.files** と **fs.chunks** というコレクションを用いてファイルデータを管理する。fs.chunks にファイルを分割して書き込み、fs.files にファイルのメタデータ（ファイル名、サイズ、fs.chunks における分割数等）が書き込まれる。非読み出し試験については、外観検査における高画素画像およびワイヤー配線のためのデータファイルを GridFS によって保存し、QC.result のドキュメントに fs.files ドキュメントファイルのオブジェクト ID を記録することによって、品質試験結果とデータの関連付けを行っている。

The figure consists of two screenshots of a web-based test result management system. The left screenshot shows a component details page for 'Component: 20UPGR10099999'. It includes sections for 'Information' (Serial Number: 20UPGR10099999, Component Type: module, FE type: RD53A, Children: 20UPFC999995, 20UPFC999996, 20UPFC999997), 'Comments' (comment: kinoshita, Institution: kinoshita, submit button), and 'Current Stage Status' (Locally sign-off QC results for this stage). Below these are tables for 'Test Type' (OPTICAL, GLUE_MODULE_FLEX_ATTACH, MASS, METROLOGY) and 'QC Test' (a large table of test results). The right screenshot shows a detailed view of an optical inspection result for 'TestType:OPTICAL'. It displays a table with keys like component_id, current_stage, test_type, institute, and user, along with a 'Pictures of Visual Inspection' section showing a picture of a chip wire and a comment 'comment only'.

図 3.1 非読み出し試験結果のブラウザ出力。

3.1.2 非読み出し試験結果閲覧機能

非読み出し試験結果のウェブブラウザ出力を図 3.1 に示す。試験結果をウェブブラウザから閲覧できるように、各ピクセルモジュールのページ（図 3.1 の左）の下部に登録した品質試験結果の一覧を表示する。この一覧は、初めは試験日時について昇順に表示しているが、必要に応じて別の項目におけるソートを可能にするため、JavaScript を利用し動的に処理できるようにした。項目名をクリックするとソートすることができるであり、クリックする毎に昇順、降順、ソート解除となる。

ブラウザから受け取った信号をもとにデータベース内データの抽出処理を行い、試験結果を適切な形に整形しブラウザに表示する。Code 3.1 に示すように、試験日時や試験場所等の試験結果の基本要素を全ての試験項目について共通要素としてもち、「results」の値のみ試験項目で個別の形式を持つ。そこで、共通項目としては前試験項目について共通の票を出力、品質試験結果については適切な形に整形しブラウザに表示するようにした。図 3.1 の右図は外観検査についてのウェブブラウザ表示であり、外観検査において問題があると判定された部分の拡大写真の GridFS から抽出し、コメントと共に表示している。他の品質試験結果の表示画面については付録 A.1 にまとめれる。

3.1.3 ピクセルモジュール特性閲覧機能

ピクセルモジュールの基本特性について、Iref 値や Pull-up 抵抗値はワイヤー配線後に初めて確認できる。そのため、組み立て工程の途中で情報を更新する必要があり、ローカルデータベースへの基本特性の更新情報は非読み出し試験結果登録用 GUI を用いて行われる。Code 3.2 に基本特性を管理するドキュメントを示す。基本特性に関する情報は MongoDB の **QC.module.prop** というコレクションに保存され

る。内容は品質試験結果を表すコレクションである QC.results と同じであるため、品質試験結果を表すウェブページと同様の処理を行い特性情報を閲覧することができる。基本特性の表示画面については付録 A.2 にまとめる。

Code 3.2 ピクセルモジュールの基本特定更新情報を表すドキュメント。

```

4   1  {
5     2   "_id" : ObjectId("613ec032f5194763373ddad1"),
6     3   "component" : "6138924058c2e3000a0db59d",
7     4   "user" : "kinoshita",
8     5   "address" : "Tokyo\u2022Institute\u2022of\u2022Technology",
9     6   "currentStage" : "MODULEWIREBONDING",
10    7   "testType" : "IREFTRIM_FE",
11    8   "sys" : {
12      9     "cts" : ISODate("2021-09-13T03:06:25.499Z"),
13     10    "mts" : ISODate("2021-09-13T03:06:25.499Z"),
14     11    "rev" : 0
15   },
16   12 },
17   13   "results" : {
18     14     "value" : {
19       15       "chip1" : "1000",
20       16       "chip2" : "1000",
21       17       "chip3" : "1000",
22       18       "chip4" : "1000"
23     },
24     19   },
25     20     "comment" : "hoge"
26   },
27   21 },
28   22   "dbVersion" : 1.01
29 }
```

3.1.4 読み出し試験結果比較機能

読み出し試験は組み立てたピクセルモジュールの性能を評価する上で重要な品質試験であり、各ピクセルが正常に動作していることを確認する必要がある。表 1.2 に基づきピクセル応答評価機能が先行研究 [2] によって開発された。ピクセル応答評価機能は以下の流れで処理が行われる。

- 33 1. データベースから解析するためのデータファイルを取得し、キャッシュディレクトリに保存
- 34 2. 作成したファイルを読み込み、ピクセル解析を実行し、結果値やプロットを after_analysis.root と
35 言う名前のファイルにまとめる
- 36 3. after_analysis.root にまとめられた結果を PNG の画像に変換し出力

ピクセル応答評価機能を応用し、二つの結果を比較できる機能の開発を行った。ピクセル解析を行うために CERN が提供している解析フレームワークである ROOT を使用している。ROOT を用いて作成した after_analysis.root においてピクセルごとに解析結果がまとめられており、このファイルを利用することにより 2 つの結果を比較することができる。ピクセル解析結果を図 3.2 に示し、各評価基準における不良ピクセルの分布を図 3.3 に示す。

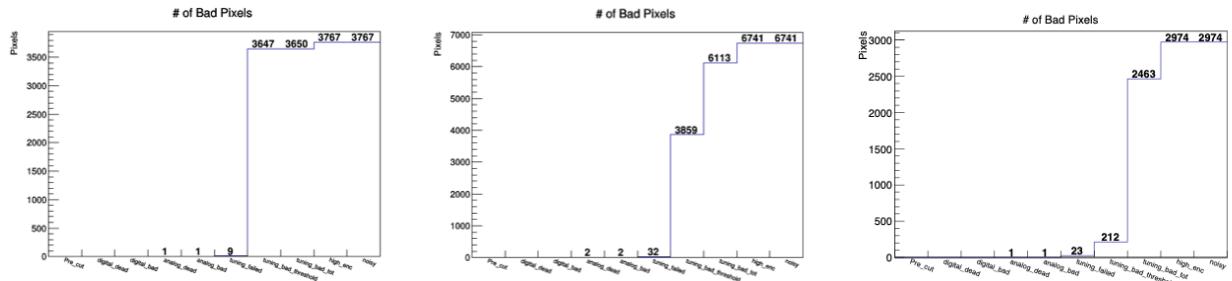


図 3.2 ピクセル応答評価機能を用いて作成したピクセル解析結果(左・中央)と2つの差分を用いた比較結果。横軸は評価基準、縦軸は該当する不良ピクセル数を表す。

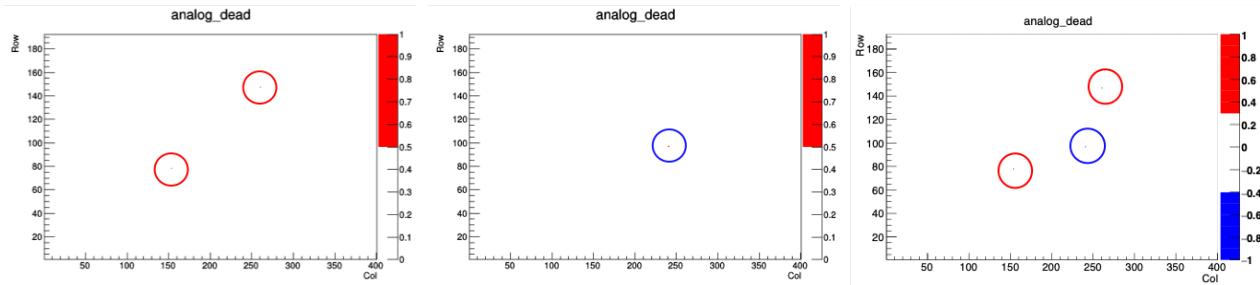


図 3.3 ピクセル解析結果における不良ピクセルの分布(左・中央)と2つの差分を用いた比較結果。各図は二次元ヒストグラムであり、横軸は ASIC における各ピクセルの列番号、縦軸は行番号を示している。

- 1 この機能を用いることにより、温度サイクル試験におけるモジュールへの熱応力により発生するバンプ
2 部の剥がれが発生した際に、1つ前の組み立て工程でバンプ剥がれが見られるかを確認すること
3 に役に立つ。バンプ剥がれが発生したピクセルモジュールについての読み出し試験結果を図 3.4 に示す。
4 図 3.4 は X 線を用いて行った Source スキャンについての分布である。この図の二次元ヒストグラムの左
5 上の領域にヒット数が無いことから、この部分のバンプが剥がれてしまっていると予想される。このよう
6 な構造が見つかった際に1つ前の組み立て工程における試験結果と比較し、温度サイクル試験によりバン
7 プ剥がれが生じたかを確認することができる。
8 しかし、先行研究において開発されたピクセル応答評価機能は Source スキャンを含むバンプ確認試験
9 に対応していない。そのため、ピクセル応答評価機能を改良し、バンプの接続確認のために必要な解析機
10 能をローカルデータベースに実装する必要がある。

11 3.2 品質試験結果の管理機能

12 ローカルデータベースの中でピクセルモジュールの組み立て工程の管理及び各工程に対応する品質試験
13 の選択機能が先行研究 [2] によって開発された。本機能は各ピクセルモジュールが中央データベースで定
14 義された枠組みに準拠して、組み立て工程および品質試験を適切に行うこと目標に開発が進められて
15 いる。さらに、測定の失敗などでローカルデータベースに保存された不要な品質試験結果を中央データベ
16 スに同期しないために、本結果を1つ選択する必要がある。このような機能も先行研究によって開発が進
17 められてきた。

18 先行研究において開発されたピクセルモジュールの組み立て工程に対して結果を選択する機能の流れを
19 図 3.5 に示す。

20 MongoDB の `QC.module.status` というコレクションを用いて、ピクセルモジュールの組み立て工

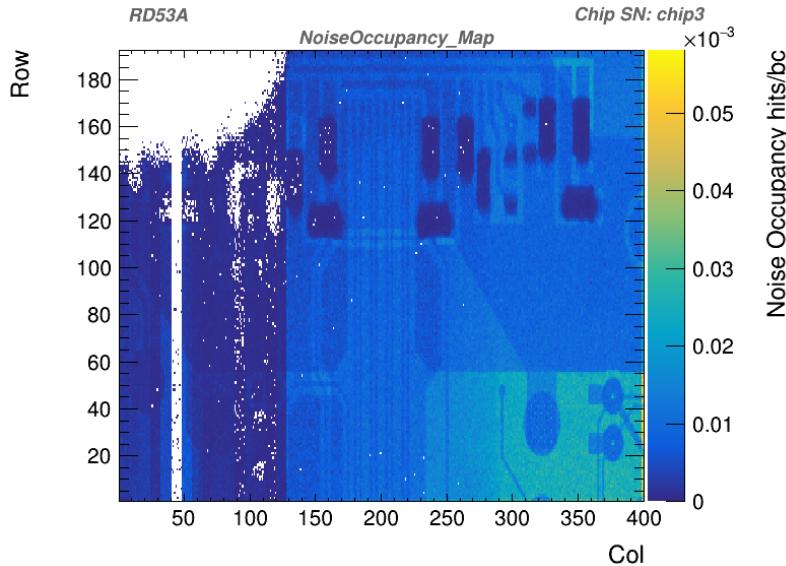


図 3.4 バンプ剥がれが発生したピクセルモジュールについての読み出し試験結果。

- 1 程の情報が管理される。あるピクセルモジュールの組み立て工程を管理するためのドキュメントの一部を
- 2 以下に示す。

Code 3.3 ピクセルモジュールの組み立て工程を管理するためのドキュメントの一部。

```

3
4   {
5     "component": "5fa79114e615fa000a1a5976",
6     "currentStage": "MODULEWIREBONDPROTECTION",
7     "QC_results": {
8       "MODULETOPCB": {
9         "OPTICAL": "6111f7b73837eb41ce260ea2",
10        "GLUE_MODULE_FLEX_ATTACH": "6111f7f63837eb41ce260ea3",
11        "MASS": "612357447d856be03b38a189",
12        "METROLOGY": "6111f8d36e74625bdf6ae78b"
13      },
14      "MODULEWIREBONDING": {
15        "WIREBONDING": "613efde160bd9bd2e7b55baf",
16        "WIREBOND": "61440cdcb8ead3bffe1eee9",
17        "OPTICAL": "6112039ac94f52b6bffe6193",
18        "SENSOR_IV_30_DEGREE": "611c8b3c6d8576d04476a345",
19        "SENSOR_IV_20_DEGREE": "61272573ff1e5897417740be",
20        "SENSOR_IV_min15_DEGREE": "6127248dc2768e0d8b9a1e62",
21        "SLDO_VI": "61120852ccc38bbac041823c",
22        "PIXEL_FAILURE_TEST_30_DEGREE": "619b4461a77fcabe31d3bbd3",
23        "PIXEL_FAILURE_TEST_20_DEGREE": "6111fe09eef9e1d67e00943a",
24        "PIXEL_FAILURE_TEST_min15_DEGREE": "61120415e8df6613c5393f0c"
25      },
26      "MODULEWIREBONDPROTECTION": {
27        "POTTING": "-1",
28        "OPTICAL": "-1",
29        "MASS": "-1",

```

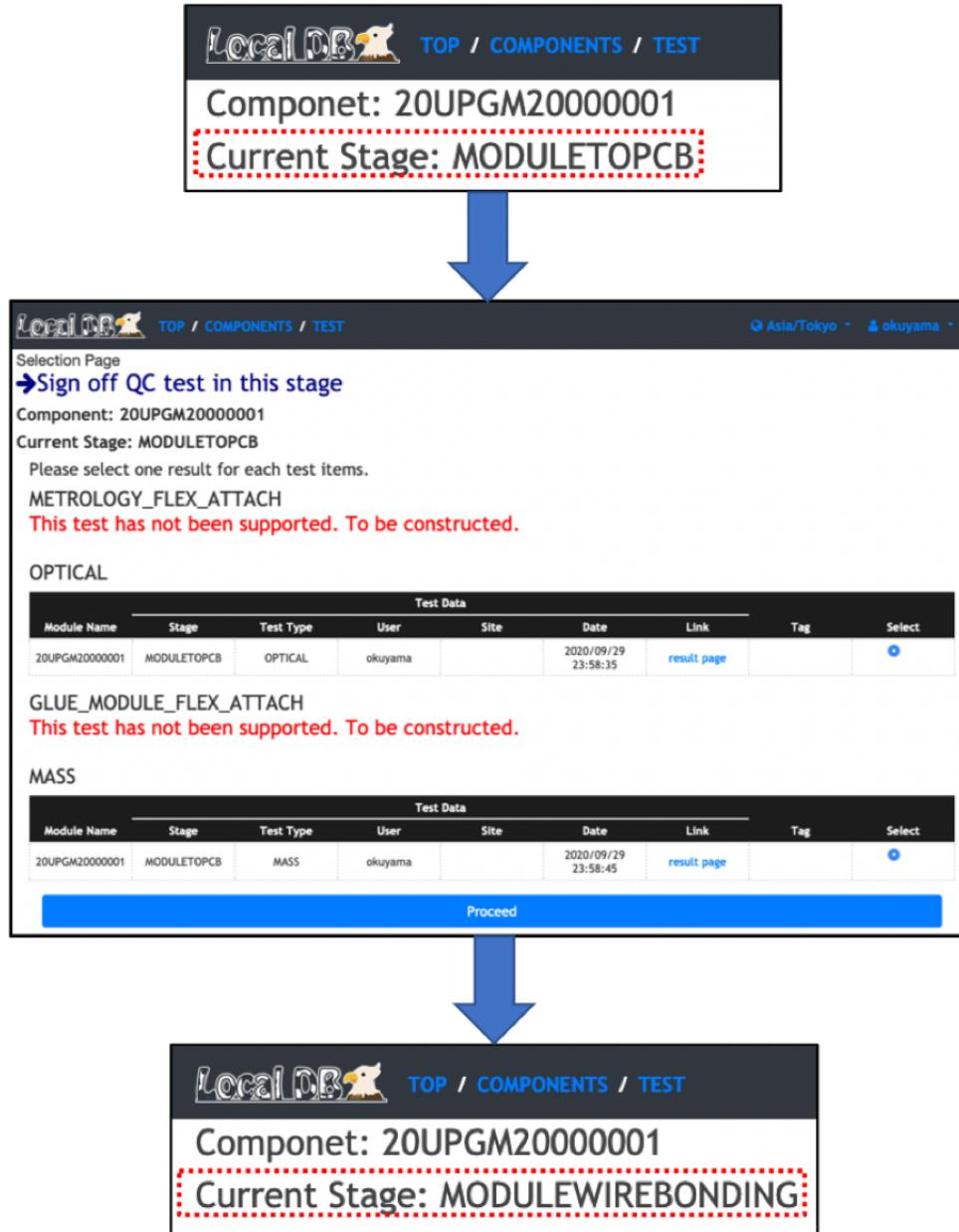


図 3.5 結果選択画面及び組み立て工程表示の例 [2]。図の上部で組み立て工程が “MODULETOPCB” である。図の中部において品質試験結果選択処理を行なっており、この図では “OPTICAL” と “MASS” の結果を選択している。この時、ローカルデータベース内部では選択された結果にタグ付けがなされる。これらの結果は中央データベースと同期される。また結果選択後は組み立て工程が自動的に更新される。図の下部では” MODULEWIREBONDING” になっていることが分かる。

```

1   27      "SENSOR_IV" : "-1",
2   28      "REGISTER_TEST" : "-1",
3   29      "READOUT_IN_BASIC_ELECTRICAL_TEST" : "-1"
4   30    }
5   31  }
6   32 }
```

8 QC.module.status のドキュメントはピクセルモジュールごとに作成される。ドキュメント内の”cur-

1 rentStage”にそのピクセルモジュールの現在の工程についての情報を保持し、図 3.5 で選択した結果
 2 は”QC_results”にそのドキュメントを表すオブジェクト ID を記録することによって、品質試験結果との
 3 関連付けを行っている。

4 この機能は品質試験結果のみに対応しており、ワイヤー配線後に更新を行うピクセルモジュール特性に
 5 ついての管理機能は未実装であった。先行研究で開発された選択機能を改良し、ピクセルモジュール特性
 6 を選択する機能の開発を行った。本研究で開発した機能を以下に示す。

7 3.2.1 ピクセルモジュール特性の選択機能

8 ワイヤー配線後に決まるピクセルモジュールの基本特性を管理するために、MongoDB 内に新たなコ
 9 レクションを定義した。基本特性の本結果を管理するコレクションとして **QC.prop.status** を定義し、
 10 Code 3.4 に示すようなドキュメントを作成、保存する。”QC_properties”にそれぞれの特性項目とその
 11 結果を表すオブジェクト ID を記録することにより、特性結果が管理されている **QC.module.prop** の
 12 ドキュメントとの関連付けを行う。

Code 3.4 ピクセルモジュールの組み立て工程を管理するためのドキュメントの一部。

```
13
14 1 {
15 2   "_id" : ObjectId("611a1c039c1b5786d950a17c"),
16 3   "proddbVersion" : 1.02,
17 4   "component" : "60d426d8b33600000af63e5b",
18 5   "currentStage" : "MODULEWIREBONDING",
19 6   "status" : "created",
20 7   "QC_properties" : {
21 8     "RD53A_PULL-UP_RESISTOR" : "61120d26ccc38bbac0418240",
22 9     "IREFTRIM_FE" : "61120c00ccc38bbac041823e",
23 10    "ORIENTATION" : "61120c0fccc38bbac041823f"
24 11  }
25 12 }
```

27 図 3.5 に示した組み立て工程に対して本結果を選択する機能に加え、組み立て工程がワイヤー配線であ
 28 れば基本特性の測定結果を選択できるようにした。基本特性を選択するウェブブラウザ上の表示を図 3.6
 29 に示す。この機能を用いて選択した基本特性結果の ID が Code 3.4 の”QC_properties”の各特性項目の
 30 欄に記録される機能となっている。ここで記録された結果が、ワイヤー配線工程における品質試験結果と
 31 同時に中央データベースへ同期される。

32 3.3 中央データベースとローカルデータベースの同期機能

33 中央データベースにおいてピクセルモジュールの構成部品情報や品質試験結果を管理するために、中央
 34 データベースにそれらについての構造を定義する必要がある。中央データベースにおける構造の定義、お
 35 よび実装の大部分は先行研究 [2] によって行われた。さらに、現在行われている試作器を用いた組み立て
 36 工程の試験を通して定義された構造の見直しが行われている。構造の再定義のために、ピクセルモジュー
 37 ル開発グループ内で国際的に議論を行いながら、構造の実装を行った。

38 ピクセルモジュールの情報や品質試験結果を中央データベースに共有するために、中央データベースと
 39 ローカルデータベース間のデータ共有機能を開発する必要がある。中央データベースとの通信のために、

RD53A_PULL-UP_RESISTOR									
Test Data									
Module Name	Stage	Test Type	User	Site	Date	Link	Tag	Select	
20UPGR10099999	MODULEWIREBONDING	RD53A_PULL-UP_RESISTOR	kinoshita	Tokyo Institute of Technology	2021/08/10 14:17:30	result page		<input type="radio"/>	
20UPGR10099999	MODULEWIREBONDING	RD53A_PULL-UP_RESISTOR	kinoshita	Tokyo Institute of Technology	2021/08/10 14:22:45	result page		<input checked="" type="radio"/>	

IREFTRIM_FE									
Test Data									
Module Name	Stage	Test Type	User	Site	Date	Link	Tag	Select	
20UPGR10099999	MODULEWIREBONDING	IREFTRIM_FE	kinoshita	Tokyo Institute of Technology	2021/09/13 12:06:25	result page		<input checked="" type="radio"/>	

ORIENTATION									
Test Data									
Module Name	Stage	Test Type	User	Site	Date	Link	Tag	Select	
20UPGR10099999	MODULEWIREBONDING	ORIENTATION	kinoshita	Tokyo Institute of Technology	2021/08/10 14:18:01	result page		<input checked="" type="radio"/>	

[Proceed](#)

図 3.6 ピクセルモジュールの基本特性選択画面。

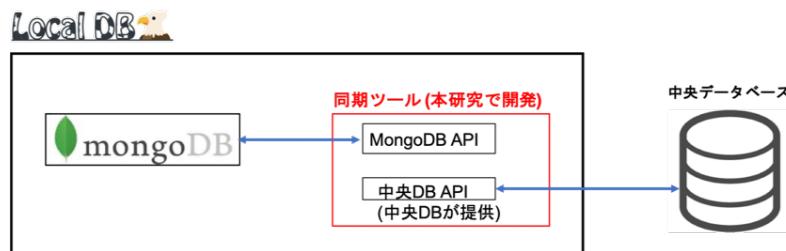


図 3.7 同期ツールの処理のイメージ

- 1 中央データベースが開発、提供している Python パッケージを用いた。
- 2 本研究では以下の機能を実装した。
 - 3 ● ピクセルモジュール情報の登録機能
 - 4 ● 品質試験結果の同期機能

5 3.3.1 ピクセルモジュール情報の登録

- 6 ピクセルモジュールの品質試験結果を管理するために、ピクセルモジュール情報を中央データベース、
- 7 ローカルデータベースに登録する必要がある。本研究では、ピクセルモジュール情報を中央データベース
- 8 に登録およびローカルデータベースに同期する機能の開発を行った。ピクセルモジュールの構成部品であるペアモジュールおよびフレキシブル基板は、それぞれについての品質管理を行う機関で登録される。
- 9 ピクセルモジュールについても各組み立て機関において、組み立てるピクセルモジュールを登録する予
- 10 定であり、登録用のシステムを開発する必要がある。本研究で開発したピクセルモジュールの登録機能の
- 11 詳細を以下に示す。
- 12

13 ピクセルモジュール情報の管理方法

- 14 ピクセルモジュールの構成部品との関係を図 3.8 に示す。ピクセルモジュールと構成部品は親子関係を定義することにより、部品構造の定義を行う。親子関係により部品構造を定義することにより、それぞれの部品の情報を関連付けて管理することができる。ピクセルモジュールの組み立てにおいて重要なのは、
- 15 読み出し試験結果は各 ASIC 每の結果が得られるため、ピクセルモジュールに対して行った試験結果を
- 16 各 ASIC と関連付けて管理する必要があるということである。親子関係を辿ることにより、ピクセルモ
- 17
- 18

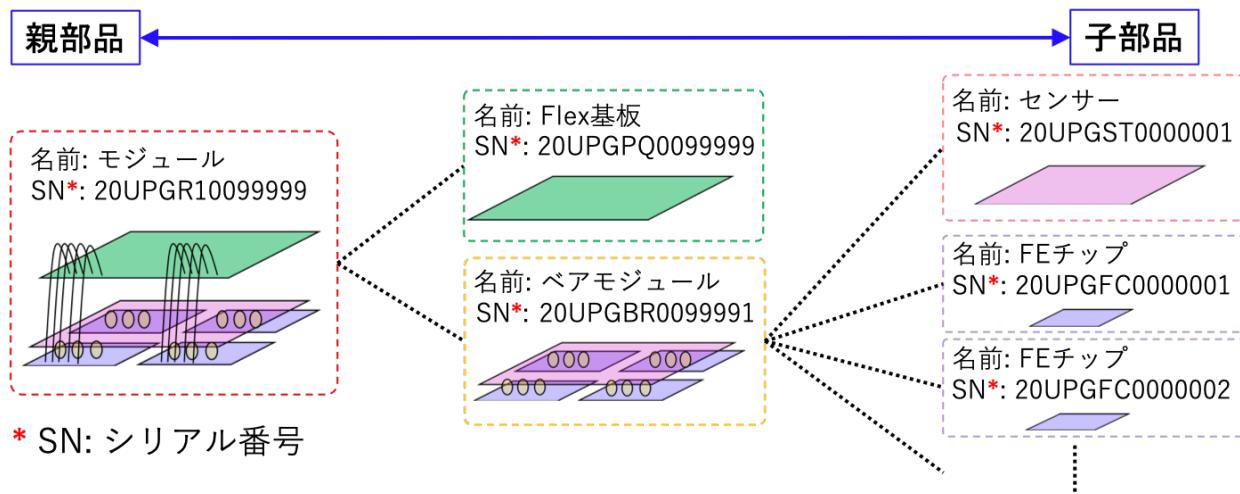


図 3.8 ピクセルモジュールの親子関係。この図では、Quad モジュールを例にピクセルモジュールの親子関係を示している。

- 1 ジュールの情報から 4 枚の ASIC の情報を漏れなく得ることができる。
- 2 また、中央データベースでは ASIC が 4 枚搭載された Quad モジュールだけではなく、3 枚の ASIC が搭載されている Triplet モジュールの情報も取り扱う。Quad モジュールに搭載されるベアモジュールは、センサー 1 枚と ASIC 4 枚から構成されるのに対して、Triplet モジュールに搭載されるベアモジュールはセンサー 1 枚と ASIC 1 枚で構成される。また、読み出し試験のテスト用に開発された、センサーが搭載されていないピクセルモジュール (Digital モジュール) の情報についても中央データベースに登録する。そのため、ピクセルモジュールはいくつかの種類を持ち、それぞれに対して構成するベアモジュールおよびフレキシブル基板の種類が異なる。親子関係を定義する際に、あるピクセルモジュールがどの種類の子部品を持つか定義することにより、データベース上で間違った組み合わせでピクセルモジュールの組み立てをすることを防ぐことができる。あるピクセルモジュールのタイプと組み立て可能なベアモジュールおよびフレキシブル基板の関係を表 3.1 に示す。
- 3 ピクセルモジュールはシリアルナンバー (製造番号) を用いて管理される。ITk に搭載されるピクセルモジュールや構成部品に用いられるシリアルナンバーは「20 U xxxx nnnnnnnn」のように 14 桁で定義される。「20」は ATLAS を構成する部品であることを表し、「U」は ITk アップグレードに関連する部品であることを表す。さらに、「xxxx」において各部品の種類の識別を行うことができ、「nnnnnnnn」は各部品を特定するための 7 桁の通し番号である。さらに、ピクセルモジュールのシリアルナンバーの 7 桁の通し番号の 1 桁目は ASIC の種類を表し、2 桁目はセンサーの厚みを表す。通し番号の 1 桁目、2 桁目の数字と ASIC の種類、センサーの厚みの対応を表 3.2 に示す。また、ピクセルモジュールとフレキシブル基板は 1 対 1 の対応をすることから、最後の 5 桁はフレキシブル基板と同じになるようにシリアルナンバーが決定される。

21 ピクセルモジュール情報の登録機能の開発

22 表 3.1 に示したように、中央データベースに定義した構造を用いて、ピクセルモジュール情報を登録する機能の開発を行った。登録の際に必要な流れは以下の通りである。

- 23 i. 表 3.1 のように定義されたピクセルモジュールの構造を中央データベースから取得
- 24 ii. ピクセルモジュールの登録に必要な情報の入力

表 3.1 あるピクセルモジュールのタイプと組み立て可能なベアモジュールおよびフレキシブル基板の関係。識別 ID の項目は、各部品についての種類を識別するためにシリアルナンバーに含まれる 4 桁の英数字である。

ベアモジュール	識別 ID	フレキシブル基板	識別 ID	ピクセルモジュール	識別 ID
Single bare module	PGB1	Triplet L0 Stave PCB	PIPT	Triplet L0 Stave module	PIMS
		Triplet L0 R0 PCB	PIP0	Triplet L0 Ring0 module	PIM0
		Triplet L0 R0.5 PCB	PIP5	Triplet L0 Ring0.5 module	PIM5
Dual bare module	PGB2	Dual PCB	PGPD	Dual chip module	PGR2
Quad bare module	PGB4	Quad PCB	PGPQ	L1 quad module	PIM1
				Outer system quad module	PGM2
Digital single bare module	PGBS	Triplet L0 Stave PCB	PIPT	Digital triplet L0 stave module	PIR6
		Triplet L0 R0 PCB	PIP0	Digital triplet L0 Ring0 module	PIR7
		Triplet L0 R0.5 PCB	PIP5	Digital triplet L0 Ring0.5 module	PIR8
Digital quad bare module	PGBQ	Quad PCB	PGPQ	Digital L1 quad module	PIR9
				Digital quad module	PGRB
Dummy single bare module	PGBT	Triplet L0 Stave PCB	PIPT	Dummy triplet L0 stave module	PIR3
		Triplet L0 R0 PCB	PIP0	Dummy triplet L0 Ring0 module	PIR4
		Triplet L0 R0.5 PCB	PIP5	Dummy triplet L0 Ring0.5 module	PIR5
Dummy quad bare module	PGBR	Quad PCB	PGPQ	Dummy L1 quad module	PIR1
				Dummy quad module	PGRA

表 3.2 モジュールのシリアルナンバーと ASIC およびセンサーの関係。

1 桁目の数字	ASIC の種類	2 桁目の数字	センサーの厚み
0	RD53A	0	Thin(150 μm)
1	ITkpix_v1	1	Thick(300 μm)
2	ITkpix_v1.1		
3	ITkpix_v2		
9	No ASIC		

- 1 iii. 中央データベースと通信し、ピクセルモジュールを登録および構成部品を登録

- 2 i. ピクセルモジュールの構造を中央データベースから取得
 - 3 ピクセルモジュールを中央データベースに登録する際、表 3.1 に示されている組み合わせに基づいて構成部品の情報を登録する必要がある。ローカルデータベースのウェブブラウザからピクセルモジュールの登録を行うために、登録するピクセルモジュールの種類に対して、適切な種類のベアモジュールおよびフレキシブル基板が使用されているかを確認する必要がある。
 - 4 ローカルデータベースにおいて、適切な種類のベアモジュールとフレキシブル基板が使用されていることを確認するために、Code3.5 に示すドキュメントを QC.module.types のコレクションに作成する。

1 Code 3.5 は中央データベースからピクセルモジュールの構造についてのデータをダウンロードし、必要な情報抽出することにより作成される。このドキュメントは全てのピクセルモジュールについて共通のドキュメントであり、他組み立て機関においても同様の構造を持つことから、QC.module.types は localdbtools のデータベースにおいて管理する。Code 3.5 は以下の情報を保持する。ここで、以下の括弧内は Code 3.5 におけるキー値を表している。

- 6 ● ドキュメント作成時のデータベースのバージョン情報 (dbVersion)
- 7 ● ピクセルモジュールを ITk に実装する際の大まかな位置情報 (subprojects)
- 8 ● ピクセルモジュールの種類と識別コード (types)
- 9 ● ピクセルモジュールの子部品情報 (children)

Code 3.5 ピクセルモジュールの組み立て工程を管理するためのドキュメントの一部。

```
10 1 {
11 2     "_id" : ObjectId("5bb1e1bef981520009c54bc5"),
12 3     "dbVersion" : 1.01,
13 4     "code" : "MODULE",
14 5     "name" : "Module",
15 6     "project" : {
16 7         "code" : "P",
17 8         "name" : "Pixels"
18 9     },
19 10    "subprojects" : [
20 11        {
21 12            "code" : "PI",
22 13            "name" : "Inner\u2225pixels"
23 14        },
24 15        {
25 16            "code" : "PB",
26 17            "name" : "Outer\u2225pixel\u2225barrel"
27 18        },
28 19        {
29 20            "code" : "PE",
23 21            "name" : "Pixel\u2225endcaps"
32 22        },
33 23        {
34 24            "code" : "PG",
35 25            "name" : "Pixel\u2225general"
36 26        }
37 27    ],
38 28    "types" : [
39 29        {
40 30            "code" : "TRIPLET_L0_STAVE_MODULE",
41 31            "name" : "Triplet\u2225L0\u2225stave\u2225module",
42 32            "subprojects" : [
43 33                {
44 34                    "code" : "PI",
```

表 3.3 ピクセルモジュール登録に必要な入力情報

ピクセルモジュール 基本情報	project	ピクセルモジュールの場合は”P”を入力
	subproject	ITk に搭載する際の大まかな位置情報
	institution	登録者の所属機関
	componentType	ピクセルモジュールの場合は”MODULE”を入力
	type	ピクセルモジュールの種類
	ATLAS Serial Number	14桁のシリアルナンバー
ピクセルモジュール 特性情報	FE chip version	搭載される ASIC の種類
	Thickness	搭載されるセンサーの厚み情報
組み立て部品 情報	ベアモジュール	14桁のシリアルナンバー
	フレキシブル基板	14桁のシリアルナンバー
	モジュールキャリア	14桁のシリアルナンバー

```

1   35           "name" : "Inner_Layer_pixels"
2   36       }
3   37     ],
4   38     "snComponentIdentifier" : "MS"
5   39   },
6   40   ...
7   41   ],
8   42   "children" : {
9   43     "TRIPLET_LO_STAVE_MODULE" : {
10  44       "BARE_MODULE" : "PGB1",
11  45       "PCB" : "PIPT"
12  46     },
13  47   ...
14  48   }
15  49 }
```

ii. ピクセルモジュールの登録に必要な情報の入力

ピクセルモジュールを中央データベースに登録する際に入力必須情報を表 3.3 に示す。ピクセルモジュールを登録する際には、これらの情報を中央データベースに送信する必要があるが、入力項目が多いことから入力ミスが発生する可能性ある。一般的に人がルーティーンタスクにおいて入力ミスをする確率は 0.3% であることが知られている [26]。各ピクセルモジュールに必要な入力項目は 11 項目であり、日本においてはこの作業を約 2200 回 (ピクセルモジュールの数) 繰り返すため、最大 60 個程度のピクセルモジュールに対して誤った情報が記録されてしまうと予想される。さらに、表 3.3 に示した入力情報は、データベースの構造を熟知した人であれば項目から入力する内容を直ちに理解することができるが、品質試験を行う試験者はデータベースについて詳しいとは限らない。

そこで、ローカルデータベースにおけるピクセルモジュールの登録機能では、入力ミスを防ぐために入力パラメータを必要最低限に削減し、入力フォーマットを簡略化するように設計した。入力パラメータを削減するために、別のパラメータから情報を抽出する手法を導入した。ピクセルモジュール登録のための

The screenshot shows a web-based application for registering a module. At the top, there's a navigation bar with 'LocalDB' logo, 'TOP | COMPONENTS | SCANS', and user information 'Asia/Tokyo' and 'kinoshita'. Below the navigation is a breadcrumb 'Top Page > Register a Module to ITkPD'. The main title is 'Register a Module to ITkPD'. A note says 'Please enter the information about the Module.' Below this, there are four required fields: 'Module Type' (a dropdown menu with placeholder '*** choose Module Type ***'), 'Bare Module serial number*' (text input field containing 'Bare Module serial number 1'), 'PCB serial number*' (text input field containing 'PCB serial number'), and 'Module Carrier serial number*' (text input field containing 'Module Carrier serial number'). At the bottom is a blue 'Confirm' button.

図 3.9 ローカルデータベースを用いたピクセルモジュール登録画面。

1 入力パラメータと削減したパラメータを以下に示す。

2 ● 入力パラメータ

- 3 – ピクセルモジュールの種類情報
- 4 – ベアモジュールのシリアルナンバー
- 5 – フレキシブル基板のシリアルナンバー
- 6 – モジュールキャリアのシリアルナンバー

7 ● 削減したパラメータ

- 8 – project: 本機能ではピクセルモジュールのみであるから”P”で固定
- 9 – subproject: ピクセルモジュールの種類情報から抽出
- 10 – institution: 中央データベースにアクセスする際のユーザー情報から抽出
- 11 – componentType: 本機能ではピクセルモジュールのみであるから”MODULE”で固定
- 12 – ATLAS Serial Number: ベアモジュールおよびフレキシブル基板のシリアルナンバーから自動生成
- 13 – FE chip version: ベアモジュール情報から抽出
- 14 – Thickness: ベアモジュール情報から抽出

16 この手法を導入することにより、ピクセルモジュールの登録の際に必要な入力パラメータを半数以下に
17 減らすことができる。ローカルデータベースのウェブページにおけるピクセルモジュールの登録画面を
18 図 3.9 に示す。入力ミスを防ぐために、ピクセルモジュールの種類はプルダウンから選択するようにし、
19 各部品のシリアルナンバーを入力する欄は、14桁の英数字のみ入力できるものとした。

20 また、Triplet モジュールはフレキシブル基板 1 枚とベアモジュール 3 つから構成される。そのため、
21 図 3.9 に示したピクセルモジュール登録画面において、Triplet モジュールである種類のモジュールが選
22 択された場合には、ベアモジュールについての入力欄を 3 つに増やすようにした。

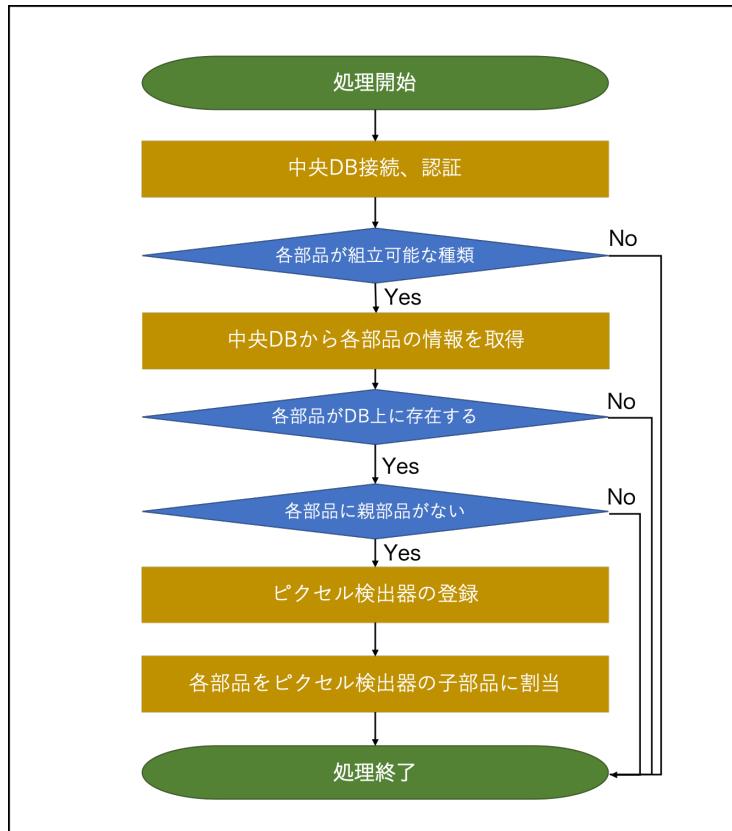


図 3.10 ピクセルモジュール登録機能による処理の流れ。

₁ iii. 中央データベースと通信し、ピクセルモジュールを登録および構成部品を登録

₂ 入力した情報を用いて中央データベースと通信し、ピクセルモジュールの登録およびデータベース上
₃ において構成部品の組み立てを行う。図 3.10 にピクセルモジュール登録機能による処理の流れを示す。
₄ 図 3.9 において入力した情報と、中央データベースへのログインパスワードを用いてピクセルモジュール
₅ の登録を行う。登録の際に、各部品が中央データベース上に存在すること、データベース上で組み立て可
₆ 能であることを確認する。これにより、本来とは異なる部品によるピクセルモジュールの登録を一部防ぐ
₇ ことができる。

₈ 3.3.2 品質試験結果の同期機能

₉ 第 2 で示したように、ピクセルモジュールを複数機関で組み立てを行う場合や、品質試験を終えて
₁₀ CERN に送った際に、ピクセルモジュールの受け取り先の機関において輸送中にピクセルモジュールに
₁₁ 損傷がないことを確認するため、品質管理試験を行う。そのために、輸送前後の品質試験結果を比較する
₁₂ ことや輸送前の設定値を用いて読み出し試験を行う必要がある。

₁₃ 本研究において、先行研究 [2] で開発された読み出し試験結果を中央データベースにアップロードする
₁₄ 機能を拡張し、全ての品質管理試験結果およびピクセルモジュールの組み立て工程を中央データベースと
₁₅ 同期する機能の開発を行った。

1 中央データベースにおける品質試験結果の管理方法

2 先行研究において、中央データベースにおけるピクセルモジュールの組み立て工程と付随する品質管理
3 試験項目の実装が行われた。各品質試験項目について、アップロードする情報を以下に示す。

- 4
- ピクセルモジュールのシリアルナンバー
 - 5 ● 試験日時
 - 6 ● 試験機関
 - 7 ● コメント
 - 8 ● ローカルデータベースを再現するためのファイル

9 ※ 品質試験結果

10 ※ 品質試験特性

11 上記において、「・」で記した情報は全ての試験項目において共通であり、「※」で記した情報は各試験項
12 目で固有の値を持つ。ローカルデータベースは NoSQL の MongoDB を用いて品質試験結果を管理して
13 いるため、品質試験結果に含まれるパラメータを定義することなく柔軟に管理できるが、中央データベー
14 スにおいては各試験項目について、アップロードするパラメータの枠を定義し、その枠に従った形の情報
15 のみを管理することができる。非読み出し試験について、中央データベースに定義したアップロードする
16 パラメータの枠を表 3.4 に示す。

17 品質試験結果のアップロード機能

18 表 3.4 のように中央データベースに定義した品質試験のパラメータを用いて、品質試験結果をアップ
19 ロードする機能の開発を行った。各品質試験結果をアップロードするための処理を以下に示す。

- 20 1. 中央データベースからアップロードするパラメータの枠を取得
- 21 2. ローカルデータベースから試験結果を抽出し、各パラメータの枠に値を埋める
- 22 3. 作成した結果を中央データベースへアップロード
- 23 4. アップロードした結果にデータファイルを添付

24 4 番目の処理の際に、読み出し試験における試験結果の JSON file や外観検査についての画像ファイル
25 の添付を行う。さらに、ローカルデータベースの各品質試験結果を表すドキュメントについても JSON
26 ファイルに変換し添付する。この JSON ファイルをダウンロードすることにより、別の組み立て機関に
27 設置したローカルデータベースにおいて試験結果の再現を行うことができるよう設計した。

28 中央データベースは各組み立て機関において保存されている全ての結果を保存するのではなく、ある組
29 み立て工程における本結果のみを中央データベースに同期する。ある組み立て工程において、本結果を選択
30 する機能において選ばれた結果を一括アップロードする機能の開発を行った。品質試験結果のアップロー
31 ド処理の流れを図 3.11 に示す。この処理の流れのループに示すように、選択機能を用いて選択された品
32 質試験結果を中央データベースへ一括アップロードを行う。さらに、アップロード処理を行っている工程
33 がワイヤー配線であれば、ピクセルモジュールの特性項目のアップロード処理も行う。ループ中の条件分
34 枝において、同一の試験結果が中央データベースにアップロードされていないかの確認を行う。これによ
35 り、中央データベースにおいて結果の重複を避けることができる。

36 ローカルデータベースにおいて、複数の組み立て工程において試験結果の選択が行われている場合は、
37 アップロード処理を繰り返す。これを繰り返すことにより、最終的にローカルデータベースにおけるピク

表 3.4 中央データベースに定義したアップロードパラメータ

品質試験項目	種類	パラメータの種類	パラメータのタイプ
質量測定	結果	質量	float
	特性	測定精度	float
質量測定	結果	全体画像	image
平坦性測定	結果	平坦性測定結果	1 次元の list
センサー IV 特性	結果	電流値	1 次元の list
		電流の分散値	1 次元の list
		電圧	1 次元の list
		温度	1 次元の list
		湿度	1 次元の list
		時間	1 次元の list
	特性	室温	float
SLDO VI 特性	結果	電流値	1 次元の list
		電圧	1 次元の list
		電圧の分散値	1 次元の list
		温度	1 次元の list
		湿度	1 次元の list
		時間	1 次元の list
	特性	室温	float
ワイヤボンド強度測定	結果	平均負荷値	float
		負荷の分散値	float
		最大負荷値	float
		最小負荷値	float
		ワイヤボンド部の損傷割合	float
	特性	使用した機械の名前	string
		試験者の名前	string
		テストのスピード	float
		ボンド部の高さ	float
		負荷のテスト値	float
ワイヤボンド情報	結果	温度	float
		湿度	float
	特性	使用した機械の名前	string
		試験者の名前	string
		ワイヤーボンドに用いたジグ	string
		ワイヤーボンドのプログラム	string
		ワイヤーボンドのバッチ数	string
ベアモジュールと フレキシブル基板の 接着情報	結果	室温	float
		湿度	float
	特性	接着剤の比率	float
		バッチナンバー	string
		試験者の名前	string
	接着方法		string

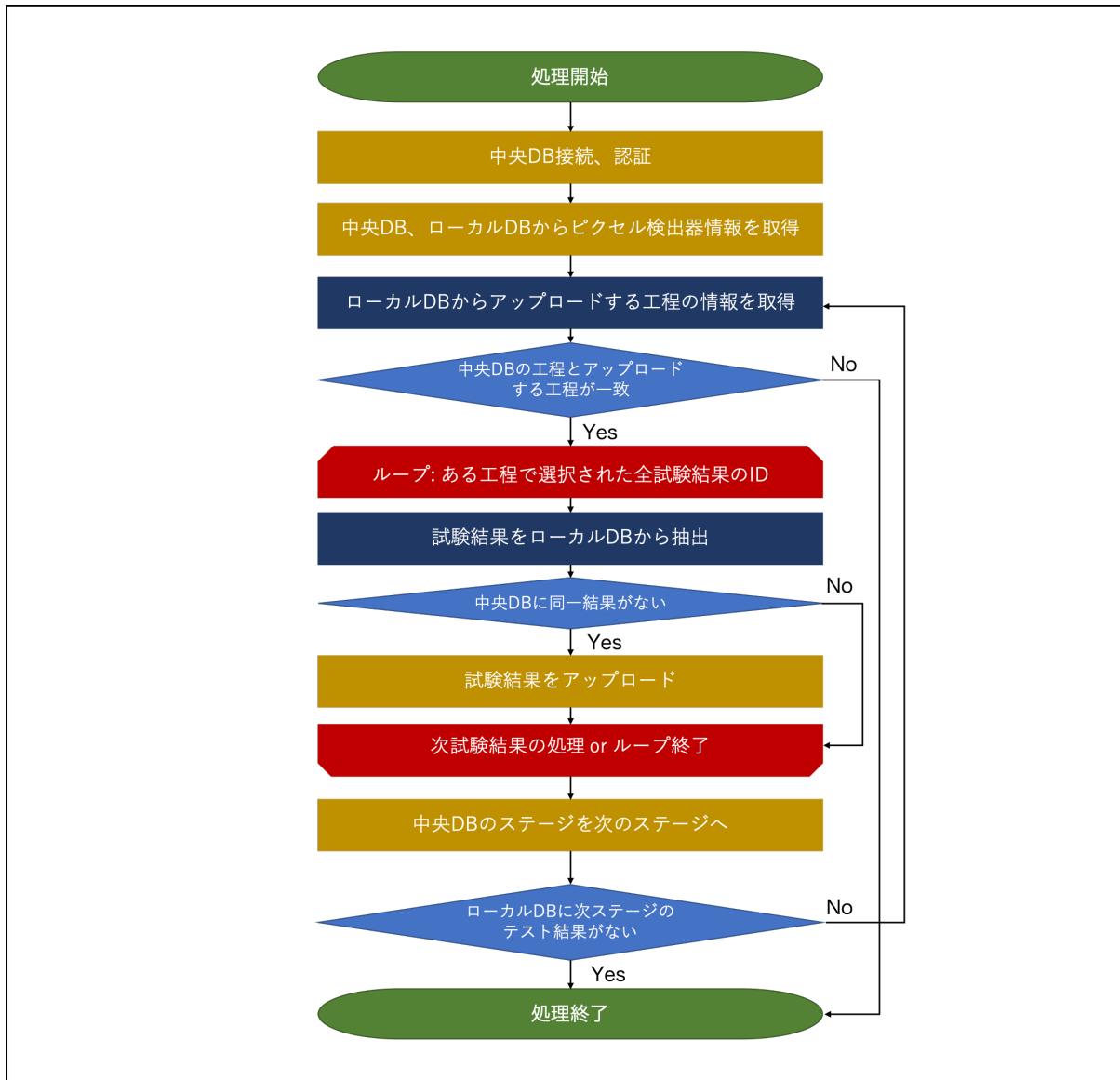


図 3.11 品質試験結果アップロード処理の流れ。

- 1 セルモジュールの組み立て工程の一つ前の工程までの品質試験結果のアップロード処理を行う。各ループ
- 2 の最後に、中央データベースの組み立て工程を一つ後の工程に変更するため、最終的にローカルデータ
- 3 ベースと中央データベースの組み立て工程が同一のものとなり、ピクセルモジュールの組み立て工程の情
- 4 報の同期を行うことができる。よって、中央データベースにおいてピクセルモジュールの量産過程が実際
- 5 に行われている各組み立て機関の進捗度と同じになり、中央データベースにおけるデータを確認すれば、
- 6 量産の進捗度を確認することができる。

7 アップロード機能の性能評価

- 8 開発した機能の実用性を検証するためにアップロード機能の性能評価を行った。実際の組み立て機関
- 9 で用いるハードウェアに近い環境で行うために、自身のラップトップ環境ではなく、陣内研究室で管理
- 10 しているサーバーを用いて性能評価を行った。サーバーのCPUはIntel Core i3 2.93GHz、通信速度は

表 3.5 アップロード機能の処理時間の測定結果

品質試験項目	データ形式	容量	処理時間
質量測定	テキスト	8 B	1.9 ± 0.7 [sec]
平坦性測定	テキスト	88 B	2.0 ± 0.5 [sec]
外観検査	画像 (png)	3.7 MB	8.8 ± 3.9 [sec]
読み出し試験	zip ファイル	3.9 MB	161.4 ± 10.0 [sec]

表 3.6 アップロード機能に使用される API 一覧

関数名	処理の内容	処理時間の平均
generateTestTypeDtoSample	表 3.4 に定義した情報を取得	0.65 ± 0.43 [sec]
uploadTestRunResults	品質試験結果を登録	0.88 ± 0.55 [sec]
createTestRunAttachment	試験結果にデータファイルを添付	0.60 ± 0.17 [sec]

¹ 47.5 MB/sec^{*1}である。

² 測定方法として、以下の処理 100 回繰り返し、その平均を処理時間とし分散を誤差導出した。

- ³ 1. 品質試験結果を表す ID を用いて試験結果をローカル DB から抽出
⁴ 2. 品質試験結果を中央データベースへアップロード

⁵ アップロード機能の性能評価結果を表 3.5 に示す。

⁶ 読み出し試験のアップロードにかかる処理時間は $\mathcal{O}(100)$ であり、読み出し試験以外の結果については
⁷ 処理時間が $\mathcal{O}(1)$ となった。1つの工程に定義される読み出し試験以外の品質試験は最大 7 項目であるた
⁸ め 20 秒程度で処理が終わり、実用可能だと考えられる。しかし、読み出し試験についてはアップロード
⁹ 処理に 5 分以上処理時間を要する。中央データベースへのアップロード処理は、ウェブブラウザのボタン
¹⁰ を押せば図 3.11 の流れに沿って自動で処理を行うため、休憩時間の前などに行えば実用可能であると考
¹¹ えられるが、アップロード処理中にネットワークが不安定になるようなことが起きると途中で処理が止ま
¹² り、不十分なデータが中央データベースに残ってしまう。処理時間が長くなる原因としては以下の 2 つが
¹³ 挙げられる。

- ¹⁴ ● アップロードの処理に用いる中央データベースとの通信 API の処理時間
- ¹⁵ ● ローカルデータベースから試験結果を抽出するのにかかる処理時間

¹⁶ 初めに中央データベースとの通信 API の処理時間について考察する。各品質試験結果をアップロードす
¹⁷ る際に使用する API の一覧を表 3.6 に示す。非読み出し試験の場合は、「generateTestTypeDtoSample」、
¹⁸ 「uploadTestRunResults」をそれぞれ 1 回行い、「createTestRunAttachment」を 1 回（ローカルデータ
¹⁹ ベースを再現するための JSON ファイル）+ 添付するデータファイルの数だけ行う。アップロードする
²⁰ データの容量は通信速度 47.5 [MB/sec] と比較して小さいため、API の使用回数に処理時間が律速され
²¹ ると考えられる。

²² 一方で、読み出し試験はピクセルモジュールのみではなく、各 ASIC についても結果が作成される。そ

*1 測定を行った 2021 年 9 月時の値

表3.7 ダウンロードツールの処理時間

スキャン項目	抽出するデータファイル数	処理時間の平均
デジタルスキャン	31 (96.8 MB)	11.36 ± 1.15 [sec]
アナログスキャン	31 (96.8 MB)	9.21 ± 1.11 [sec]
スレッショルドスキャン	204 (118.6 MB)	11.13 ± 0.91 [sec]
ToT スキャン	47 (112.9 MB)	10.12 ± 1.27 [sec]
ノイズスキャン	31 (102.8 MB)	9.05 ± 0.75 [sec]
クロストークスキャン	161 (272.8 MB)	28.08 ± 2.61 [sec]
合計	505 (800.7 MB)	78.95 ± 3.52 [sec]

1 のため、クアッドモジュールの場合は「generateTestTypeDtoSample」、「uploadTestRunResults」をそ
 2 れぞれ5回行い、「createTestRunAttachment」は34回行う。そのため、APIを用いた場合の処理時間
 3 は式(3.1)のようになる。

$$(0.65 \pm 0.43) \times 5 + (0.88 \pm 0.55) \times 5 + (0.60 \pm 0.17) \times 34 \simeq 28.05 \pm 1.81 \text{ [sec]} \quad (3.1)$$

4 ここで、アップロードするデータの容量は通信速度47.5 [MB/sec]と比較して小さく無視できるものと
 5 し、表3.6に示したものと同程度になると仮定して計算を進めた。図3.1より、中央データベースとの通
 6 信にかかる処理時間は30秒程度であるが、表3.5で求められた処理時間はこれに比べて非常に長くなっ
 7 ている。そのため、ローカルデータベースから読み出し試験結果を抽出する際の処理時間が原因でアップ
 8 ロード処理に時間を要していると考えられる。

9 ローカルデータベースからの読み出し試験結果の抽出処理は、先行研究[3]によって開発されたダウン
 10 ロードツールを用いている。ダウンロードツールを用いて読み出し試験結果を抽出する際に要する処理時
 11 間を表3.7に示す。これにより、読み出し試験結果の抽出に79秒かかることがわかる。さらに、抽出し
 12 たデータはZIPファイルに圧縮された後に、中央データベースにおける品質試験結果に添付される。圧
 13 縮処理の平均時間は0.26 ± 0.21 [sec]であり、作成するZIPファイルは34個である。そのため、ローカ
 14 ルデータベースにおいて中央データベースにアップロードする読み出し試験結果の作成のために、90秒
 15 程度の処理時間を要する。

16 アップロード処理時間の改善案

17 図3.11に示した流れでアップロード処理を行うと、読み出し試験を含む工程では2分以上かかってしまう。
 18 ブラウザの応答に時間がかかると、使用者の満足感や信頼感の低下の原因となり、ユーザーエクス
 19 ペリエンスが非常に悪くなる。そこで、本研究においてはバックグラウンドでアップロード処理を行うこ
 20 とにし、使用者のブラウザ応答待機時間を削減させた。使用者がアップロード開始のボタンを押した後、
 21 中央データベースへの接続およびユーザー認証を完了した後、バックグランド処理を行うようにした。こ
 22 れにより、使用者の待ち時間は接続およびユーザー認証にかかる3.10 ± 0.38 [sec]^{*2}となり、待ち時間を
 23 抑えることができた。

24 バックグラウンドで処理を行うと、ネットワークの接続切れやアップロード機能のバグが原因で処理が
 25 中断されても使用者が問題に気づかないということが発生する。そのような問題を防ぐため、品質試験

^{*2} 中央データベースへの接続およびユーザー認証処理を100回行い、その平均と分散を計算した。

表 3.8 アップロード機能に使用される API 一覧

まとめ方	ZIP ファイルの容量	処理時間の平均
全ての結果	10 MB	-
ピクセルモジュール	29 KB	0.88 ± 0.8 [sec]
1 つの ASIC	2.5 MB	4.30 ± 0.94 [sec]

1 結果アップロードのバックグラウンド処理が中断された時にブラウザ上に問題を表示する機能の実装を行った。

3 バックグラウンド処理により、ブラウザの応答時間は短縮できるが、実際に行う処理時間に変化はない
4 ため、今後この改善を行う必要がある。中央データベースとの通信を伴う処理は、中央データベース API
5 開発者によるものであるため、本研究の対象外である。今後改善可能な処理を以下に示す。

6 1. 添付するデータファイルの数の削減

7 2. ローカルデータベースからデータを抽出する処理時間の削減

8 項目 1 に関して、読み出し試験における品質試験に添付するデータファイルの数は 34 個である。
9 読み出し試験はピクセルモジュールのみではなく、各 ASIC 毎に作成され、結果添付ファイルは、読み出し試験における 6 項目のスキャンについて作成される。データファイルを添付するための関数
11 は”createTestRunAttachment”であり、表 3.6 から、この通信の処理に 20.4 ± 0.99 [sec] だけ必要となる。
12 そこで、圧縮するファイルの数を変更しデータファイルの添付を行い、その処理時間の変化を確認した。
13 ファイルの圧縮方法として以下の 2 つについて考える。

14 • 全てのデータファイルを一つの ZIP ファイルにまとめる

15 • ピクセルモジュールおよび各 ASIC に対するデータファイルをそれぞれ一つの ZIP ファイルにまとめる

17 これらの処理時間の測定結果を表 3.8 に示す。

18 全ての結果をまとめた場合はエラーが発生し添付に失敗した。これは、データファイルを添付するため
19 の関数”createTestRunAttachment”は 4 MB 以上のデータを扱えないことが原因である。よって、今回
20 確認した中で可能な添付の処理方法は、ピクセルモジュールと各 ASIC の結果をそれぞれまとめる方法のみ
21 であり、処理時間は式 (3.2) となる。

$$(0.88 \pm 0.8) + 4 \times (4.30 \pm 0.94) = 18.8 \pm 2.04 [\text{sec}] \quad (3.2)$$

22 式 (3.2) から得られる結果は 20.4 ± 0.99 [sec] と比較して大きな改善は期待できない。

23 項目 2 に関して、マルチスレッドを用いて、各試験項目抽出処理の並列化をすることにより抽出処理に
24 かかる時間を削減することが考えられる。表 3.7 からそれぞれのスキャン項目について、ローカルデータ
25 ベースから結果を取得することに 10 秒程度かかっていることがわかる。この抽出処理を並列化すること
26 により、処理時間が削減でき、最も時間のかかるクロストークスキャンについての時間に律速され、処理
27 時間を 50 秒程度削減できると考えられる。



図 3.12 品質試験結果ダウンロード処理の流れ。

1 品質試験結果のダウンロード機能

2 図 3.12 に品質試験結果のダウンロード機能の全体像を示す。中央データベースに添付した品質試験
3 結果のドキュメント情報を保有する JSON ファイルをダウンロードすることにより同期を行う。この
4 JSON ファイルには、品質試験結果を識別するためのオブジェクト ID が記述されている。同一オブジェ
5 クト ID の品質試験結果が QC.results のコレクション内に存在するかを確認することにより、品質結果
6 の重複を避けることができる。

7 さらに、全ての品質試験結果のダウンロード処理の後に、中央データベースにおけるピクセルモジュー
8 ルの組み立て工程をローカルデータベースへ同期する。これにより、ピクセルモジュールを輸送した後、
9 受け取り機関において正しい組み立て工程における品質管理を行うことができる。

10 3.4 本章のまとめ

11 本章では、本研究における開発項目である品質試験結果の表示機能、品質試験結果の管理機能、および
12 中央データベースとローカルデータベースの同期機能についての開発の詳細について述べた。これらの機
13 能をローカルデータベースに実装することにより、ピクセルモジュールの次世代器量産における品質試験
14 結果管理に必要な機能の基本的な部分が全て揃った。

15 品質試験結果の表示機能について、読み出し試験以外の試験項目についての結果表示機能の開発を行っ
16 た。読み出し試験結果以外の項目については、品質試験結果登録用 GUI を用いてローカルデータベース
17 に登録するため、その開発者と協力し MongoDB のドキュメントの構造の構築をし、ウェブブラウザを

1 用いた結果の表示機能の開発を行った。これにより、ピクセルモジュールの全ての試験項目についての結果を、ウェブブラウザを用いて閲覧可能になった。また、先行研究で開発された読み出し試験におけるピクセル応答評価機能を拡張し、別の読み出し試験結果と比較する機能の開発を行った。これにより各組み立て工程間における読み出し試験結果の比較を行い、不良ピクセルの変化が確認できる。ピクセル応答評価機能は、デジタル回路やアナログ回路の読み出し試験等のFEチップについてのピクセル評価が行うことができるが、バンプ接続についての評価機能が未実装である。そのため、バンプ接続についての評価機能を実装する必要がある。

8 品質試験結果の管理機能について、先行研究でピクセルモジュールの組み立て工程の管理及び各工程に
9 対応する品質試験を選択する機能が開発された。これらの機能を拡張し、ワイヤー接続工程の後に確認さ
10 れるピクセルモジュール特性の更新機能の開発を行った。この機能において選択する品質試験結果は、各
11 組み立て工程における本試験結果であり、中央データベースとの同期機能を用いて同期が行われる。

12 中央データベースとローカルデータベースの同期機能について、ピクセルモジュールの情報や品質試験
13 結果を中央データベースに共有する機能の開発を行った。この機能を開発するために、中央データベース
14 においてピクセルモジュールの構成部品構造や品質試験構造を定義する必要があり、ピクセルモジュール
15 開発グループ内で国際的に議論を行いながら、構造の実装を行った。定義した構造を用いて、ピクセルモ
16 デュール情報の登録機能や、品質試験結果のアップロード機能およびダウンロード機能の開発を行った。

17 品質試験結果のアップロード機能についての性能評価を行った。表3.5のように読み出し試験結果以外
18 の項目については数秒でアップロード処理を終えるが、読み出し試験は160秒程の処理時間が必要であ
19 り、この時間の間にネットワークが不安定になるとバグの原因になりうるため改良が必要である。本研究
20 では処理時間の改善案をいくつか考案し、それぞれについての見積もりを行った。その結果、ローカル
21 データベースからの読み出し試験結果の抽出処理を並列化することにより、処理時間が最大で50秒程度
22 削減できるという結論が得られた。

1 第4章

2 まとめ

3 4.1 まとめ

4 4.2 今後の課題

¹ 付録 A

² 非読み出し試験結果のブラウザ表示

³ A.1 非読み出し試験結果のブラウザ表示

⁴ A.1.1 外観検査

⁵ A.2 ピクセル検出器の基本特性のブラウザ表示

参考文献

- [1] ” 大学院 素粒子物理学 1 素粒子の基本的性質 ”
南部陽一郎・木下東一郎・牧二郎・中西襄・政池明、講談社サイエンティフィック
- [2] ” HL-LHC ATLAS ピクセル検出器量産時の品質試験に向けた データベースシステムの構築 ”、奥山広貴
- [3] ” HL-LHC ATLAS 実験用新型ピクセル検出器の系統評価と 量産時に向けた試験管理システムの開発 ”、窪田ありさ
- [4] ” Development of DAQ test system and database for the HL-LHC ATLAS production Pixel detector ”、Eunchong Kim
- [5] ” The CERN accelerator complex ” ,
<https://cds.cern.ch/images/OPEN-PHO-ACCEL-2013-056-1> CERN Document Server
- [6] Joao Pequenao. Computer generated image of the whole ATLAS detector. Mar 2008.
- [7] ”Study of the material of the ATLAS inner detector for Run 2 of the LHC”, <https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/PERF-2015-07/ATLASCollaboration>
- [8] ”Computer Generated image of the ATLAS calorimeter”, <https://cds.cern.ch/images/CERN-GE-0803015-01>
- [9] <https://hilumilhc.web.cern.ch/content/hl-lhc-project>
- [10] ”The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments”, <https://iopscience.iop.org/article/10.1088/1742-6596/706/2/022002/pdf>
- [11] ””, <https://www2.kek.jp/ja/news/topics/2007/crab.html>
- [12] ”ATLAS Inner Tracker Pixel Detector Technical Design Report”, <https://cds.cern.ch/record/2285585/files/ATLAS-TDR-030.pdf>
- [13] ATLAS Collaboration. “Combined measurements of Higgs boson production and decay using up to 80 fb^{-1} of protonproton collision data at $\sqrt{s} = 13 \text{ TeV}$ collected with the ATLAS experiment”. CERN Document Server. <http://cdsweb.cern.ch/record/2629412/>
- [14] <https://atlas.kek.jp/main/movie/photos/physics/index.html>
- [15] <https://home.cern/science/physics/dark-matter>
- [16] Particle Data Group, 33. Passage of particles through matter 1
<https://pdg.lbl.gov/2019/reviews/rpp2018-rev-passage-particles-matter.pdf>
- [17] M. Moll, “Displacement Damage in Silicon Detectors for High Energy Physics” , IEEE Transactions on Nuclear Science, Aug. 2018. DOI: 10.1109/TNS.2018.2819506.
- [18] ”The ATLAS Pixel Detector”, <https://pos.sissa.it/137/006/pdf>

- 1 [19] "Calibration analysis software for the ATLAS Pixel Detector", <https://cds.cern.ch/record/2027805/files/ATL-INDET-PROC-2015-003.pdf>
- 2 [20] M. Verducci, ATLAS conditions database experience with the LCG COOL conditions database
3 project, J. Phys.: Conf. Ser. 119 (2008) 042031. <https://iopscience.iop.org/article/10.1088/1742-6596/119/4/042031>
- 4 [21] ATLAS physics plots, https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/PIX-2018-009/fig_05.png
- 5 [22] "Production accompanying testing of the ATLAS Pixel module", Daniel Adam Dobos
6 <https://cds.cern.ch/record/1016933/files/thesis-2007-016.pdf>
- 7 [23] "RD53A Module Testing Document", Lingxin Meng
8 <https://cds.cern.ch/record/2702738/files/ATL-COM-ITK-2019-045.pdf>
- 9 [24] "Digital module electrical testing", Massimiliano Antonello
10 <https://cds.cern.ch/record/2723333/files/ATL-COM-ITK-2020-020.pdf>
- 11 [25] "MongoDB Documentation"
12 <https://docs.mongodb.com>
- 13 [26] "Practical Plant Failure Analysis", Neville W. Sachs, P.E.

1 謝辞

2 shaji

図目次

1.1	フレキシブル基板	2
1.2	ピクセルモジュールの組み立て工程	3
1.3	読み出し試験のセットアップ	4
1.4	SLDOVI	6
1.5	クアッドモジュールの Iref Irim 部分を表す部分(左図)とワイヤーの配線と Iref 値の関係(右図)[23]。ピンク色のワイヤーの配置により 4bit の Iref 値を表すことができる。	8
2.1	中央データベースとローカルデータベースの設置位置	10
2.2	ローカルデータベースの全体像	12
2.3	ローカルデータベースの構造	15
2.4	ウェブアプリケーションの処理の概念図	15
3.1	非読み出し試験結果のプラウザ出力。	19
3.2	ピクセル応答評価機能を用いて作成したピクセル解析結果(左・中央)と 2 つの差分を用いた比較結果。横軸は評価基準、縦軸は該当する不良ピクセル数を表す。	21
3.3	ピクセル解析結果における不良ピクセルの分布(左・中央)と 2 つの差分を用いた比較結果。各図は二次元ヒストグラムであり、横軸は ASIC における各ピクセルの列番号、縦軸は行番号を示している。	21
3.4	バンプ剥がれが発生したピクセルモジュールについての読み出し試験結果	22
3.5	結果選択画面及び組み立て工程表示の例	23
3.6	ピクセルモジュールの基本特性選択画面	25
3.7	同期ツールの処理のイメージ	25
3.8	ピクセルモジュールの親子関係	26
3.9	ローカルデータベースを用いたピクセルモジュール登録画面	30
3.10	ピクセルモジュール登録機能による処理の流れ	31
3.11	品質試験結果アップロード処理の流れ	34
3.12	品質試験結果ダウンロード処理の流れ	38

1 表目次

2	1.1	モジュール起動温度に対する階級値	6
3	1.2	ピクセル解析の評価基準一覧	7
4	1.3	プルアップ抵抗値と電源の関係	9
5	2.1	ローカルデータベースのコレクション	14
6	3.1	あるピクセルモジュールのタイプと組み立て可能なベアモジュールおよびフレキシブル 基板の関係	27
7	3.2	モジュールのシリアルナンバーと ASIC およびセンサーの関係	27
8	3.3	ピクセルモジュール登録に必要な入力情報	29
9	3.4	中央データベースに定義したパラメータ	33
10	3.5	アップロード機能の処理時間の測定結果	35
11	3.6	アップロード機能に使用される API 一覧	35
12	3.7	ダウンロードツールの処理時間	36
13	3.8	アップロード機能に使用される API 一覧	37