

2023年度 修士論文

高輝度LHC加速器実験用のATLAS新型ピクセル
検出器モジュール量産に向けた品質管理用ソフト
ウェアの開発

東京都立大学 理学研究科 物理学専攻

高エネルギー物理実験研究室

博士前期課程 22844411

鮫島大輝

2024年1月9日

概要

CERN に設置されている大型ハドロン加速器 LHC は世界最高エネルギーである 14TeV の重心系エネルギーで陽子同士を衝突させている。その衝突点の一つに置かれた ATLAS 検出器は、陽子の衝突からにより生じる粒子の生成・崩壊過程を測定し、新物理の発見を目指している。2013 年には LHC 内の 4 つの衝突点のうちの 2 つの衝突点にある検出器 (ATLAS, CMS) においてヒッグス粒子の発見に成功した。さらに統計量をあげ、ヒッグス粒子の性質解明や新物理の発見を目指すため、LHC を HL-LHC へとアップグレードする計画がある。このアップグレードにより HL-LHC は瞬間ルミノシティが 3.5 倍、積分ルミノシティは 10 倍となる。このアップグレードに伴いデータレート増加やヒットレートの占有率増加のため、高い放射線耐性や検出器の高い位置分解能、読み出し速度向上が求められる。また HL-LHC に伴い ATLAS 検出器は内部飛跡検出器を総入れ替えすることになった。新しく導入する内部飛跡検出器を ITk と呼び、ITk は約 10000 個のピクセルモジュールから構成されるため、それらの量産と品質試験を行う必要がある。ITk に搭載するモジュール選定のため、モジュールの特性をまとめたデータベースが必要であり、各研究機関におかれたデータベースを localDB、およびチェコに置かれた ITk に関連する全部品に関するデータをまとめたデータベースを中央データベースと呼ぶ。また 10000 個のモジュール量産は 2024 年 4 月～6 月頃を目安に始められる予定だが、その前段階として規模を縮小したプリプロダクションを実施している。本研究ではプリプロダクション及び本量産において localDB に品質試験結果をアップロードする module-qc-nonelec-gui の開発と測定データの解析を行う解析スクリプトの開発を行った。

本研究により本量産に向けたデータベース関係の整備が進み、多くのユーザが利用するソフトウェアである module-qc-nonelec-gui のユーザビリティ向上及び品質試験結果判別の手助けとなった。その一方、モジュールにおける品質試験は全てサポートできたが、センサーやフレキシブル基板、ベアモジュールでの品質試験についてはサポートしきれていない試験があるため、整備を急ぐ必要がある。また解析スクリプトも全ての試験をサポートできているわけではないため、今後の整備が求められている。

目次

第 1 章	序論	1
1.1	素粒子物理学	1
1.1.1	素粒子標準模型	1
1.1.2	ヒッグス機構	2
1.1.3	新物理探索	3
1.2	LHC ATLAS 実験	4
1.2.1	ATLAS 検出器	7
1.2.2	内部飛跡検出器	8
1.2.3	ピクセル検出器	10
1.3	HL-LHC 計画	11
1.3.1	新型ピクセル検出器	12
第 2 章	ピクセルモジュールの量産と品質試験	15
2.1	ピクセルモジュール	15
2.2	モジュールの量産	17
2.3	モジュール組み立て手順	17
2.4	品質試験	18
2.4.1	質量測定	18
2.4.2	外観検査	19
2.4.3	寸法測定	20
2.4.4	平坦性測定	20
2.4.5	ワイヤー強度測定	20
2.4.6	I-V 特性試験	21
2.4.7	温度サイクル試験	22
第 3 章	品質試験結果管理用データベース	23
3.1	データ管理用ソフトウェア構造図	23
3.2	中央データベース	23
3.3	LocalDB	24
3.4	YARR	27
3.5	module-qc-nonelec-gui	30
第 4 章	プリプロダクション（生産状況とソフトウェアの運用）	34
4.1	館山ハヤシレピックにおける生産	34
4.2	ハヤシレピックにおける設備と体制	35
4.3	プリプロダクションの経過報告	41

第 5 章	データベースの開発	45
5.1	module-qc-nonelec-gui における機能開発	45
5.1.1	roof-envelope	45
5.1.2	source scan	47
5.1.3	Baremodule_I-V_measure	48
5.2	localDB の改良	49
5.2.1	BAREMODULE_I-V におけるリンクの表示	50
5.2.2	BAREMODULE_I-V と SENSOR_I-V の比較	52
5.3	解析用スクリプトの開発	52
5.3.1	BAREMODULE_I-V における解析スクリプトの開発	53
5.3.2	ワイヤー強度測定の解析スクリプトの開発と評価	60
第 6 章	まとめ	65
付 録 A	BAREMODULE_I-V アップロードのためのサブツール	66
	謝辞	70
	参考文献	70

目次

1.1	素粒子標準模型	1
1.2	ヒッグスポテンシャル	2
1.3	LHC 全体図	5
1.4	LHC の粒子衝突点における検出器の配置	5
1.5	Higgs 粒子発見の際のデータ	6
1.6	run2 全データでの解析	7
1.7	ATLAS 検出器の構成図	8
1.8	内部飛跡検出器の構成図	9
1.9	内部飛跡検出器の断面図	9
1.10	ピクセル検出器の構成図	10
1.11	HL-LHC へのアップグレード予定	11
1.12	新型飛跡検出器	12
1.13	新型ピクセル検出器の断面図	13
2.1	インナーシステムにおけるモジュールの配置	15
2.2	現行のピクセルモジュール	15
2.3	triplet module	16
2.4	モジュールの構成図	16
2.5	パリレンコーティング後のモジュール	17
2.6	組み立て項目と品質試験項目	18
2.7	外観測定の様子	19
2.8	外観測定用装置	20
2.9	ワイヤーの強度 (横軸 (グラム重)) と頻度 (縦軸)	21
3.1	モジュール情報管理用データベースとその関係	23
3.2	LocalDB の web ページ	26
3.3	LocalDB のテスト済み項目一覧	27
3.4	LocalDB における YARR テスト表示	29
3.5	digital scan の結果確認ページ	30
3.6	module-qc-nonelec-gui の内部構造	31
3.7	module-qc-nonelec-gui の手順	32
3.8	module-qc-nonelec-gui におけるファイルの選択	33
4.1	館山ハヤシレピックにおける作業環境	35
4.2	読み出し品質試験用装置	36
4.3	読み出し品質試験用装置内側	37
4.4	Grafana による時系列データの監視	38

4.5	X 線照射結果	38
4.6	X 線照射装置	39
4.7	温度サイクル用恒温槽	40
4.8	モジュール保管用デシケーター	41
4.9	電圧立ち上げ速度によるモジュールとの通信可能な領域の比較	42
4.10	モジュールと通信を行う際に二つの FE チップから応答がない様子	43
4.11	あるコラムを隠して通信が復活した様子	44
5.1	ワイヤーボンダプロテクション	45
5.2	roof envelope	46
5.3	source scan	47
5.4	YARR から登録された試験のタグ表示	48
5.5	module-qc-nonelec-gui における BAREMODULE_I-V 試験の概要	49
5.6	localDB 上での BAREMODULE_I-V の結果	50
5.7	module-qc-nonelec-gui から localDB に渡されたデータ	51
5.8	module-qc-nonelec-gui から localDB にデータを渡す際のイメージ図	53
5.9	センサー単体での漏れ電流とベアモジュール段階での漏れ電流を比較したプロット	55
5.10	降伏電圧の表示 (モジュールでの I-V 測定)	56
5.11	温度設定によるセンサーチップ単体とベアモジュールの漏れ電流の比較	57
5.12	センサー単体の漏れ電流の測定に $0.04\mu A$ のオフセットが乗っていると仮定した場合の温度補正前後の比較結果	58
5.13	センサー単体の漏れ電流の測定に $0.05\mu A$ のオフセットが乗っていると仮定した場合の温度補正前後の比較結果	59
5.14	センサー単体の漏れ電流の測定に $0.06\mu A$ のオフセットが乗っていると仮定した場合の温度補正前後の比較結果	60
5.15	module-qc-nonelec-gui におけるワイヤー強度測定確認画面	62
5.16	ワイヤー強度測定における midspan break の強度 (横軸、単位: グラム重) と頻度 (縦軸) のヒストグラム	63
5.17	ワイヤー強度測定における heel break on hybrid の強度 (横軸、単位: グラム重) と頻度 (縦軸) のヒストグラム	63
5.18	ワイヤー強度測定における heel break on chip の強度 (横軸、単位: グラム重) と頻度 (縦軸) のヒストグラム	64

表 目 次

1.1	LHC と HL-LHC の比較	12
1.2	現行のピクセル検出器と新型ピクセル検出器の比較	13
3.1	中央データベースにアクセスできる主な API	24
3.2	localDB のコレクション名と詳細	25
5.1	ワイヤーの壊れ方	62

第1章 序論

欧州原子核研究機構 (CERN=Centre européen pour la recherche nucléaire) に設置されている大型ハドロン衝突型加速器 (LHC=Large Hadron Collider) の4つの衝突点のうちの一つに ATLAS 検出器が設置されており、これを用いて素粒子物理学の研究が行われている。LHC では加速器のアップグレード (HL-LHC) を予定しており、これに伴い ATLAS 検出器のアップグレードが予定されており、本章では素粒子物理学、LHC-ATLAS 実験とそのアップグレード計画について説明する。

1.1 素粒子物理学

1.1.1 素粒子標準模型

現代の素粒子物理学における基本的な枠組みを素粒子標準模型と呼んでいる。1970年代半ばに体系化されたこの理論には17の素粒子とその反粒子が登場しており、6種類のクォークと6種類のレプトン、4種類のゲージ粒子と1種類のヒッグス粒子から構成されている。クォークとレプトンは物質を構成する素粒子であり、ゲージ粒子は力を伝える粒子である。ゲージ粒子のうち、光子は電磁力、ウィークボソンは弱い力、グルーオンは強い力を伝える。またヒッグス粒子は素粒子に質量を与える働きをし、2012年にCERNでLHC加速器と、本研究と大きく関連するATLASとCMSという検出器を用いてヒッグス粒子の存在が突き止められた。



図 1.1: 素粒子標準模型

1.1.2 ヒッグス機構

1.1.1 節で記述したヒッグス粒子は素粒子に質量を与える機構 (ヒッグス機構) の証拠となる粒子であり、ヒッグス機構はゲージ粒子へ質量を与えることができる。ゲージ粒子は本来質量をもたない粒子であるべきで、ゲージ粒子が質量を持つと素粒子物理の根本原理である局所ゲージ対称性を破ることになる。対称性を破らないためには、真空中で有限の期待値を持ち、各粒子と質量に比例した結合定数で結合するスカラー場が存在するとうまく理論を説明することができる。このスカラー場の4つの自由度のうち3つはゲージ粒子に質量を与え、残る1つの自由度はヒッグス粒子となり、質量獲得プロセスをヒッグス機構と呼ぶ。以下にヒッグス機構におけるゲージ粒子の質量獲得の仕組みを説明する。

複素スカラー場 $\phi (= (\phi_1 + i\phi_2)/\sqrt{2})$ に対しポテンシャル $V(\phi) = \mu^2\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2$ を持つラグランジアンは

$$\mathcal{L} = (\partial_\mu\phi)^\dagger(\partial^\mu\phi) - \mu^2\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2 \quad (1.1)$$

と表される。 $\lambda > 0$ 、 $\mu^2 < 0$ の場合を考える。式 (1.1) は以下のように書き換えられる。

$$\mathcal{L} = \frac{1}{2}(\partial_\mu\phi_1)^2 + \frac{1}{2}(\partial_\mu\phi_2)^2 - \frac{1}{2}\mu^2(\phi_1^2 + \phi_2^2) - \frac{1}{4}\lambda(\phi_1^2 + \phi_2^2)^2 \quad (1.2)$$

ポテンシャル $V(\phi)$ は以下のようになる。

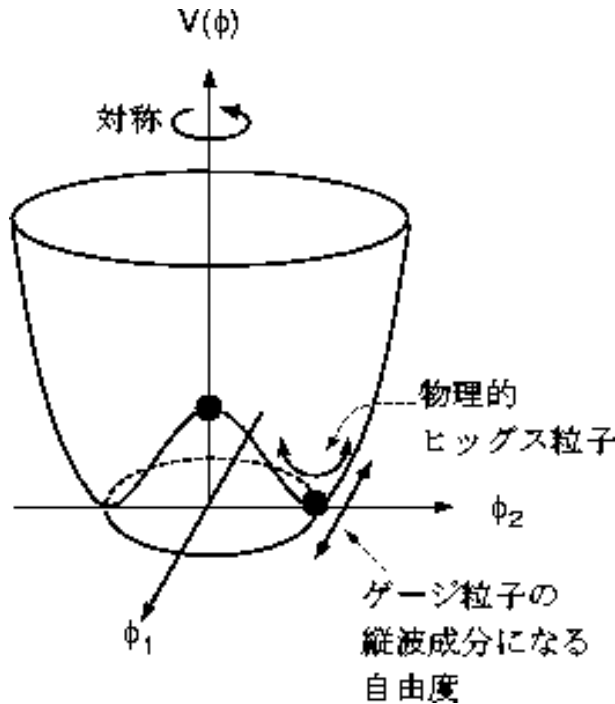


図 1.2: ヒッグスポテンシャル

このポテンシャル $V(\phi)$ を最小にする ϕ は、 $\phi_1 - \phi_2$ 平面で

$$\phi_1^2 + \phi_2^2 = -\mu^2/\lambda \equiv v^2 \quad (1.3)$$

を満たす半径 v の円周上に存在する。ポテンシャルが最小となるのは原点ではなく図のリング状の領域であり、これらを真空状態と呼ぶ。実際に実現する真空はそれらの中から一つが選ば

れた状態であり、それを選んだ瞬間に対称性が破れる。これを自発的対称性の破れという。ここで場をエネルギー最小の $\phi_1 = v, \phi_2 = 0$ としても一般性を失わないので、

$$\phi(x) = \sqrt{1/2}(v + \eta(x) + i\xi(x)) \quad (1.4)$$

U(1) 局所ゲージ変換に対し不変にするため、

$$D_\mu = \partial_\mu - ieA_\mu \quad (1.5)$$

と置き換えるとゲージ場は以下のように置き換わる。

$$A_\mu \longrightarrow A_\mu + \frac{1}{e}\partial_\mu\alpha \quad (1.6)$$

従って、ゲージ不変なラグランジアンは以下のようになる。

$$\mathcal{L} = (\partial^\mu + ieA^\mu)\phi^\dagger(\partial_\mu - ieA_\mu)\phi - \mu^2\phi^\dagger\phi - \lambda(\phi^\dagger\phi)^2 - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \quad (1.7)$$

式 1.4 を代入し、 \mathcal{L} を真空回りで展開して η と ξ であらわせば、

$$\mathcal{L}' = \frac{1}{2}(\partial_\mu\xi)^2 + \frac{1}{2}(\partial_\mu\eta)^2 - v^2\lambda\eta^2 + \frac{1}{2}e^2v^2A_\mu A^\mu - evA_\mu\partial^\mu\xi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} + (\text{相互作用項}) \quad (1.8)$$

のようになり、第三項は η 場の質量項 ($-m_\eta^2\eta^2/2$) であり、 η の質量は $m_\eta = \sqrt{2\lambda v^2}$ が生成された。ここで η の最低次で次の式が成り立つ。

$$\phi(x) = \sqrt{1/2}(v + \eta(x) + i\xi(x)) \simeq \sqrt{1/2}(v + \eta)e^{i\xi/v} \quad (1.9)$$

ここで式 1.7 の変数を、異なる一組の実数場 h, θ, A_μ で

$$\phi \longrightarrow \sqrt{1/2}(v + h(x))e^{i\theta(x)/v} \quad (1.10)$$

$$A_\mu \longrightarrow A_\mu + \frac{1}{ev}\partial_\mu\theta \quad (1.11)$$

と置き換える。これをもとに再度ラグランジアンを考えると、

$$\mathcal{L} = \frac{1}{2}(\partial_\mu h)^2 - \lambda v^2 h^2 + \frac{1}{2}e^2 v^2 A_\mu^2 - \lambda v h^3 - \frac{1}{4}\lambda h^4 + \frac{1}{2}e^2 A_\mu^2 h^2 + v e^2 A_\mu^2 h - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \quad (1.12)$$

のようになる。

この式に出てくる質量をもつ粒子 h がヒッグス粒子であり、上記のようにゲージ粒子に質量を与え、場の自由度をゲージ粒子の縦偏極としてしまう仕組みをヒッグス機構と呼んでいる。

この理論が素粒子の標準模型に適用できるものだと証明するためには、このヒッグス粒子を実験的に発見することが大きな課題であり、ATLAS 実験ではヒッグス粒子発見における大きな功績を残した。詳細は 1.2 節で説明する。

1.1.3 新物理探索

標準模型では説明できない物理現象がいくつかある。それらの物理現象を説明する理論の候補は複数あり、その代表的なものとして超対称性 (SUSY) があげられる。SUSY とはボソンとフェルミオンを入れ替える演算に対する対称性である。つまり標準模型の各粒子に対応するスピン 1/2

だけ異なる超対称性粒子 (SUSY 粒子) を予言する理論である。この SUSY は他の標準理論で説明できない物理である暗黒物質や階層性問題を説明できることを期待されている。

自然界には二つの大きく隔たったスケールが存在しており、量子重力などが無視できないプランクスケール (10^{19}GeV) と W,Z ボソンが存在する電弱スケール (100GeV) がある。これらの間にはおよそ 10^{17} もの不自然な階層性がみられ、これを階層性問題と呼んでいる。この階層性問題を解決する理論の一つとして SUSY が提唱されており、SUSY 粒子の発見が期待されている。LHC ではこの階層性問題を解決に向けて、超対称性粒子の発見を目指し研究が行われている。

素粒子実験において標準模型を超えた物理の探索には大きく分けて二つのアプローチがあり、一つはエネルギーフロンティアで、もう一つがルミノシティフロンティアである。このエネルギーフロンティアを目指す実験の一つが CERN の LHC 加速器実験である。エネルギーフロンティアを目指す実験では、その名の通りより高いエネルギーで粒子を衝突させて、発生した粒子を観測し、未だ発見されていない新物理の解明を目指している。

1.2 LHC ATLAS 実験

LHC(Large Hadron Colider) は世界最高エネルギーの陽子・陽子衝突型加速器であり、スイスにある CERN 研究所の地下およそ 100 m に設置されており、一周約 27km ある加速器である。LHC の陽子陽子衝突ではバンチと呼ばれる陽子のかたまりを加速させ正面衝突させることで、陽子内のパートンが互いに対消滅し、高エネルギーの物理現象が観測される。LHC 計画は 1994 年 12 月に承認されたものであり、日本が建設協力を発表したのは、翌年の 1995 年 5 月のことである。その後当初の建設予定よりかなり遅れることになったが、2008 年 9 月 10 日に LHC 加速器は完成し、450GeV ビーム周回を成功させた。しかし運転開始 9 日目に大きなトラブルが発生し、一時的に実験は中止となった。その後原因究明と解決を行い、2009 年 11 月に再稼働を成功させた。段階的にエネルギーをあげていき 2010 年 3 月には最終目標の半分のエネルギーである 7TeV で陽子陽子衝突実験を開始した (run1)。2012 年にはヒッグス粒子とみられる新粒子がこの LHC により示され、この発見により、はヒッグス機構を提唱した Peter W. Higgs と François Englert はノーベル賞が授与された。その後 2013 年から 2 年にわたって大規模な改良を行い、再び 2015 年から run2 として運転を再開した。このときの衝突エネルギーは 13TeV で、ヒッグス粒子の精密測定や新粒子発見に向けて実験は進められた。多量のデータ取得に成功した run2 は 2018 年 12 月に終了し、LS2(Long Shutdown) に入った。ここでは run3 に向けた準備等だけではなく、将来に向けた HL-LHC 加速器へのアップグレードも多く行われた。2022 年 7 月に run3 が再開され、13.6TeV での衝突実験に成功した。次の HL-LHC での運転に向けて様々な取り組みや準備が行われている。HL-LHC については 1.3 節で説明する。

LHC の重心系エネルギーでは、run3 の時点で 13.6TeV、HL-LHC 以降では 14TeV に向上し、新物理現象の観測を期待している。

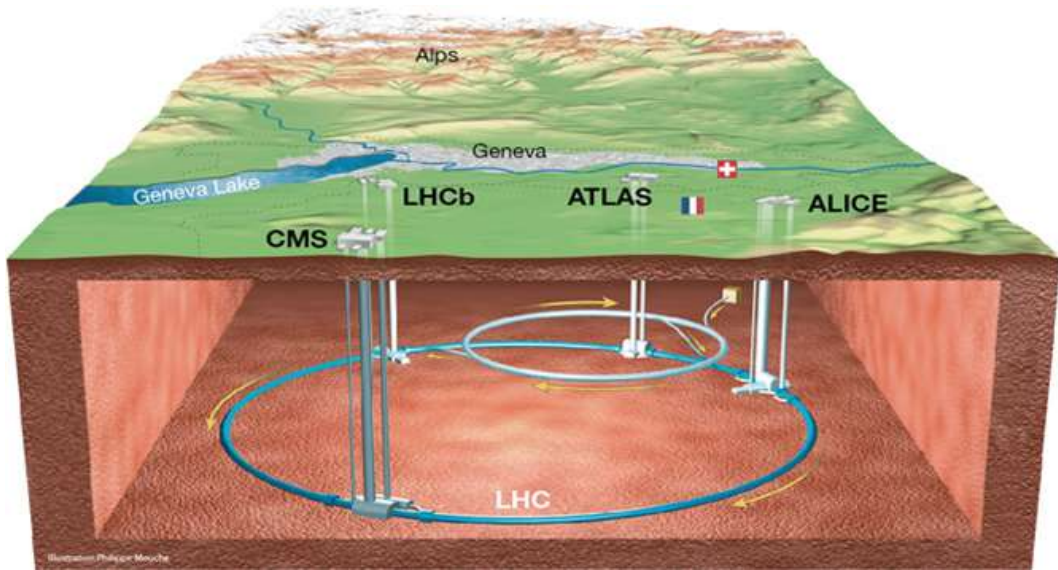


図 1.3: LHC 全体図

この LHC には衝突点が4ヶ所あり、各衝突点にはそれぞれ検出器が置かれ、それらを CMS、LHCb、ALICE、ATLAS と呼ぶ。各検出器の配置の概要を図 1.4 に示す。

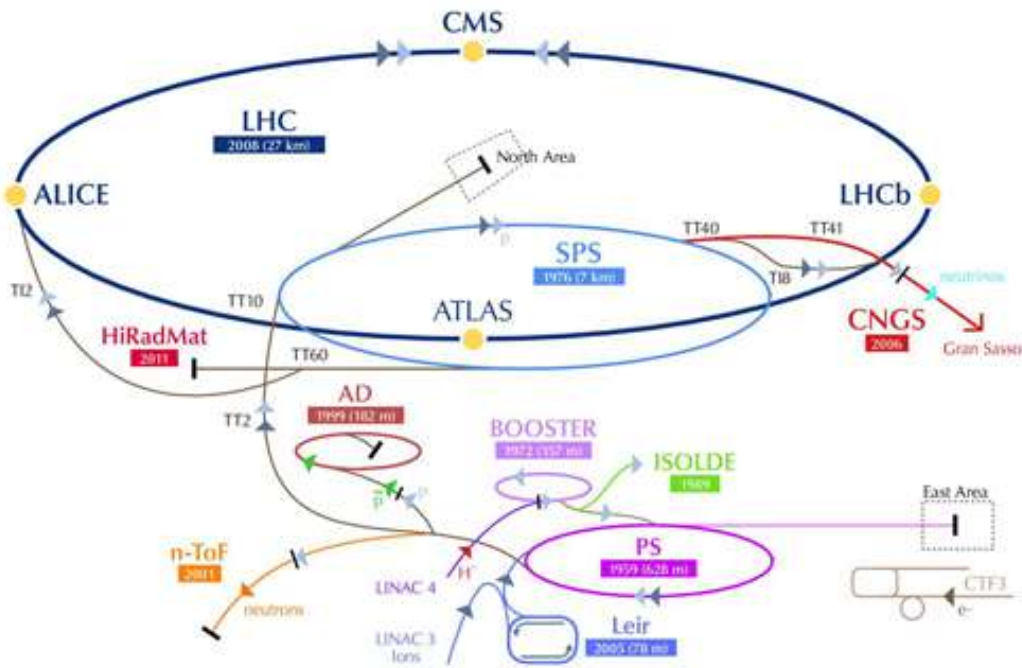


図 1.4: LHC の粒子衝突点における検出器の配置

それぞれの検出器の概略を以下に示す。

- ATLAS (A Toroidal LHC ApparatuS)
新物理の探索などを目的とした検出器。詳細は 1.2.1 節で述べる。

- CMS (Compact Muon Solenoid)
ATLAS 検出器と同様に幅広い物理を目的とした検出器。ATLAS 検出器よりサイズは小さいが強い磁場を持つ磁石を利用し運動量を測定している。
- LHCb(Large Hadron Collider beauty)
ビーム軸付近の低い運動量を持つ粒子検出に特化した検出器で、主に b,c クォーク関連の新物理の探索を目的としている。
- ALICE(A Large Ion Collider Experiment)
主に重イオン衝突を研究の対象にしている。

また 1.1.2 節で説明したヒッグス粒子について本当に存在しているかを確認するために、人工的に作り出し確認する必要があるが、LHC では run1 の 7TeV~8TeV のエネルギーでヒッグス粒子と思われるものの観測に成功した。LHC 内の衝突点のうちの一つに置かれた ATLAS 検出器によるデータ収集が行われたのは 2010 年から 2012 年間の間であり、この三年間のデータを足しあげてみることでヒッグス粒子なしでは説明できない超過事象が確認された。(図 1.5) ヒッグス粒子を仮定しない場合、図の点線のような分布を示す。ヒッグス粒子を仮定した場合は、背景事象分布 + $H \rightarrow \gamma\gamma$ となり図の赤線のような曲線に従う。下側にある図は背景事象を上から引いたもので信号をより明瞭にするようにしたものである。またこの発見があった同時期に LHC 内の別の衝突点で独立した実験を行っている CMS 実験グループも同様の超過を発表した。そこで統計的な優位性が示されたとして 2012 年 7 月に両グループは新粒子の発見を共同で宣言した。

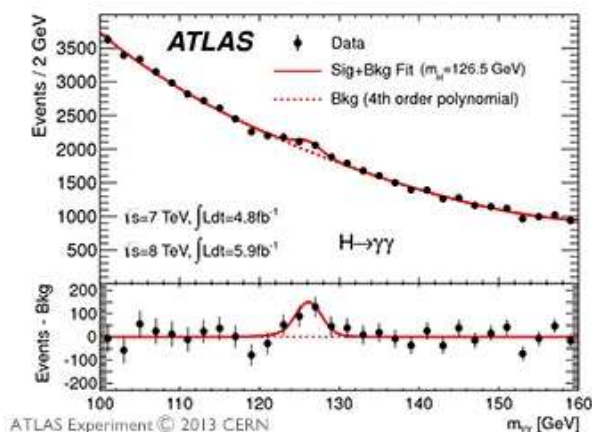


図 1.5: Higgs 粒子発見の際のデータ

2015 年から 2018 年にかけて運転した run2 におけるデータを用いたものが図 1.6 であり、よりはっきりとそのピークを見ることが可能になった。

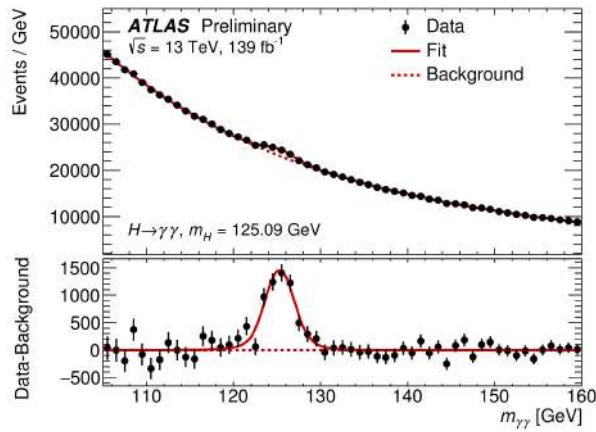


図 1.6: run2 全データでの解析

1.2.1 ATLAS 検出器

図 1.4 のうちの一つである ATLAS 検出器は、複数の検出器から構成されており、6つの同心円状の層のようになっている。最内層に内部飛跡検出器が設置され、次に超電導ソレノイド磁石、カロリメータ、トロイド磁石、ミューオン検出器の順に設置されている。全体としての大きさは高さ 25m、長さ 44m、重さ 7000 t という巨大な検出器でありながら、非常に高精度、高分解能な検出器になっている。構成図は図 1.7 のようになっている。

この検出器は物理過程を示す特徴的な実験信号の識別を行うことを目的としており、LHC のような高放射線量の環境で識別を行うためには、高速読み出しかつ放射線耐性のある検出器で構成する必要がある。ビームの衝突では多数の反応が生まれるが、その中で全ての粒子を検出し、検出器から漏れるニュートリノや新粒子を識別するには、全立体角に近い領域をカバーする必要がある。座標系等の詳細は 1.3.1 節で説明する。

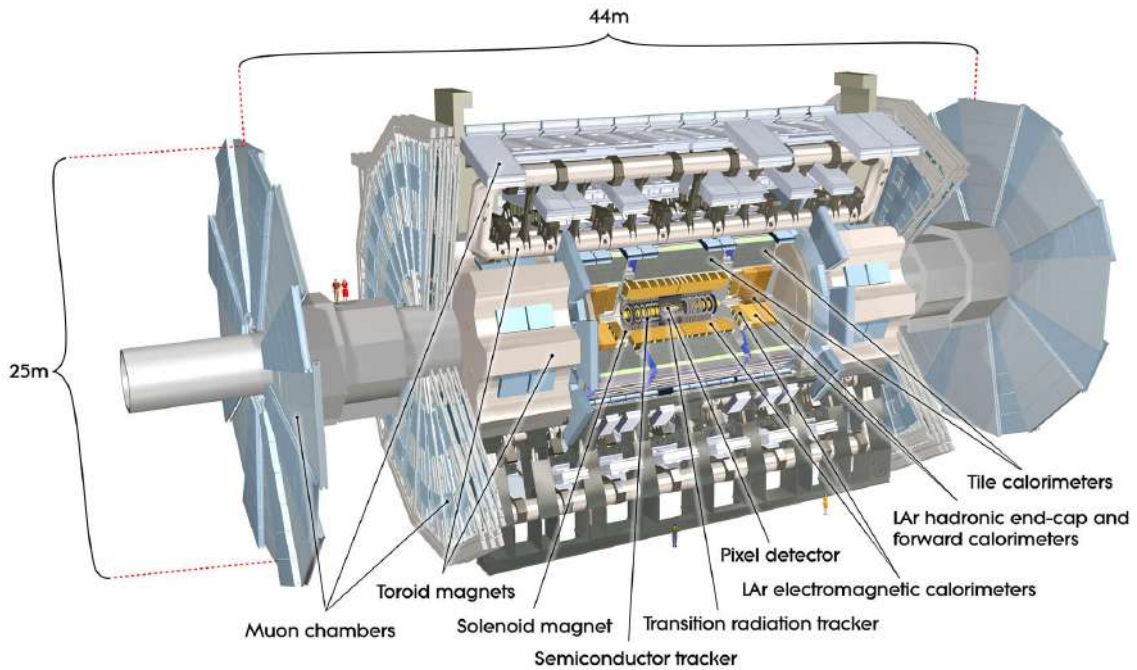


図 1.7: ATLAS 検出器の構成図

1.2.2 内部飛跡検出器

1.2.1 節で紹介した ATLAS 検出器の内部構成のうち、本研究に特に関連する内部飛跡検出器について述べる。ATLAS 検出器の最内層にある内部飛跡検出器は内側からシリコンピクセル検出器 (Pixel detectors)、シリコンストリップ検出器 (Semiconductor tracker)、遷移輻射検出器 (Transition radiation tracker) の順に配置されている。構成図を図 1.8 に示す。断面図については図 1.9 に示す。この装置は衝突点から生成された荷電粒子を検出することで飛跡の再構成を行う。この内部飛跡検出器の外側に配置されているソレノイド磁石により、2T の磁場がビーム軸に平行な方向にかけられている。

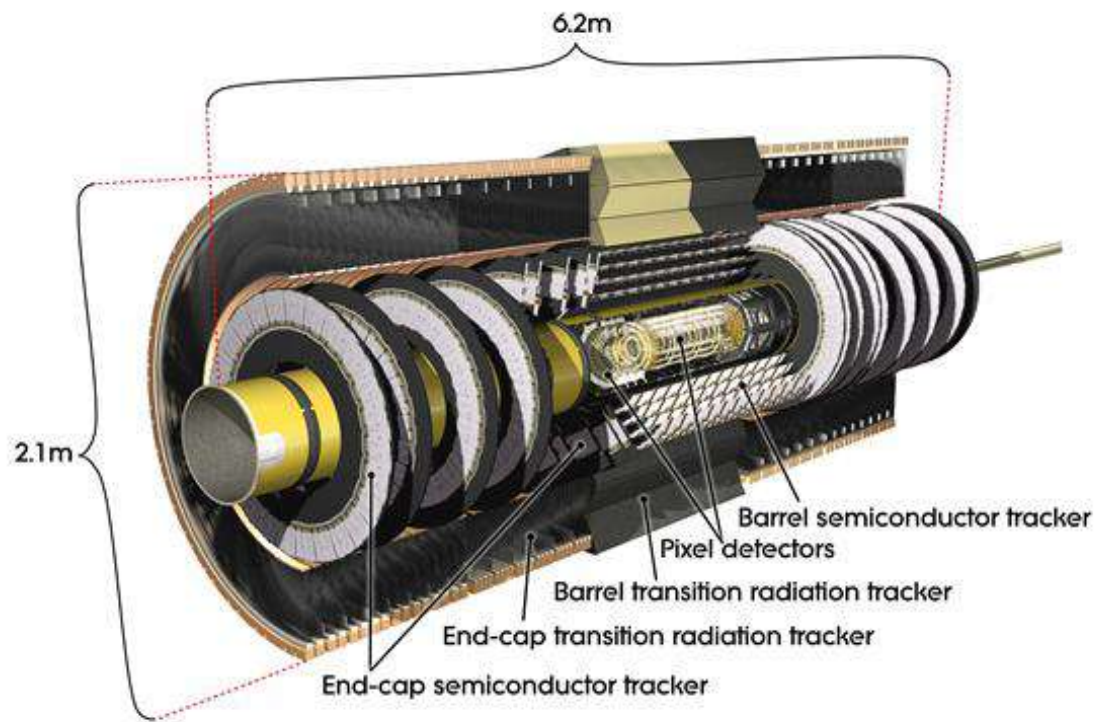


図 1.8: 内部飛跡検出器の構成図

ピクセル検出器とストリップ検出器で荷電粒子の飛跡を決定し、遷移輻射検出器で荷電粒子の飛跡検出と粒子識別を行っている。

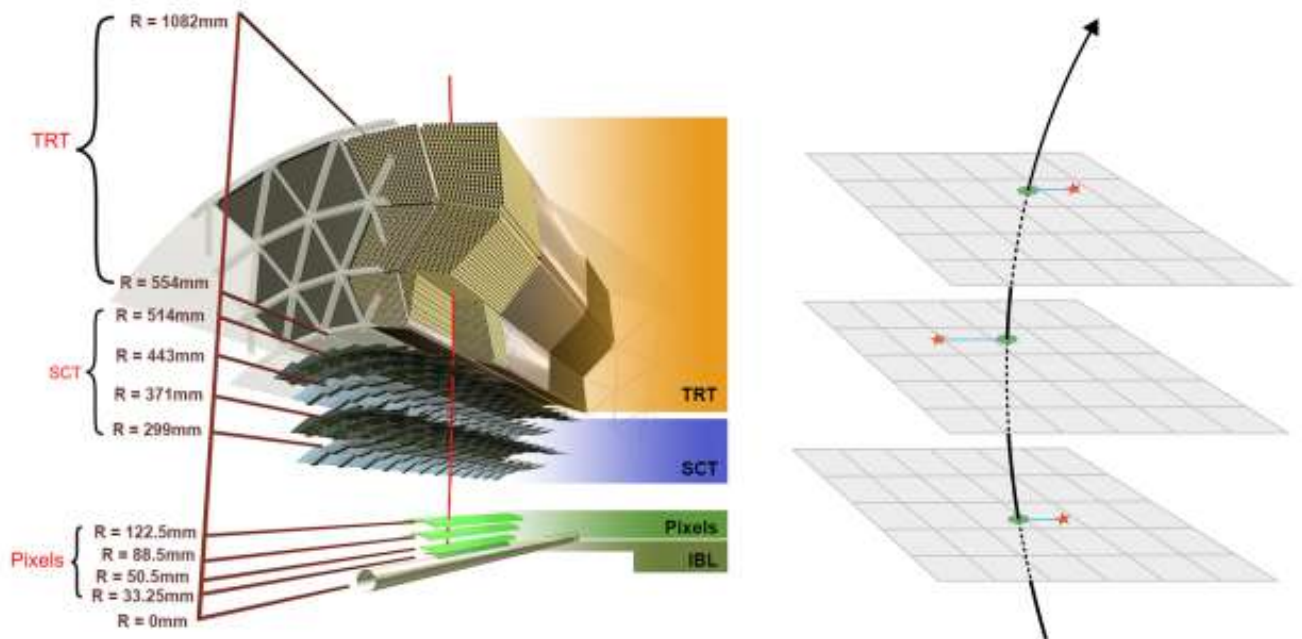


図 1.9: 内部飛跡検出器の断面図

1.2.3 ピクセル検出器

ピクセル検出器の構成図を図 1.10 に示す。ピクセル検出器は衝突点に最も近い検出器であり、バレル部が4層、エンドキャップ部が6層で構成される。ピクセル検出器も内部飛跡検出器と同様に図 1.8 で示す円筒状のような部分をバレル部と呼び、ディスク上の部分をエンドキャップ部と呼んでいる。バレル部の最内層から順に IBL, B-Layer, Layer-1, Layer-2 と呼ぶ。特にこの IBL(Insertable B-Layer) について、ピクセル検出器は LHC の運転開始時に $1 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$ のルミノシティに対応するよう設計されていたため、LHC の run3 期間におけるルミノシティ $2 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$ になると設計上性能の保証外となる。特に最内層である B-Layer では、FE チップを制御するためのデータ転送帯域が不足することから、検出効率を維持することができなくなってしまう。これらの理由から 2012 年から 2014 年にかけて行ったシャットダウン (LS-1) の期間中にこの IBL がインストールされ、run2 から使用された。図 1.10 における赤で書かれた部分が IBL に当たる部分である。

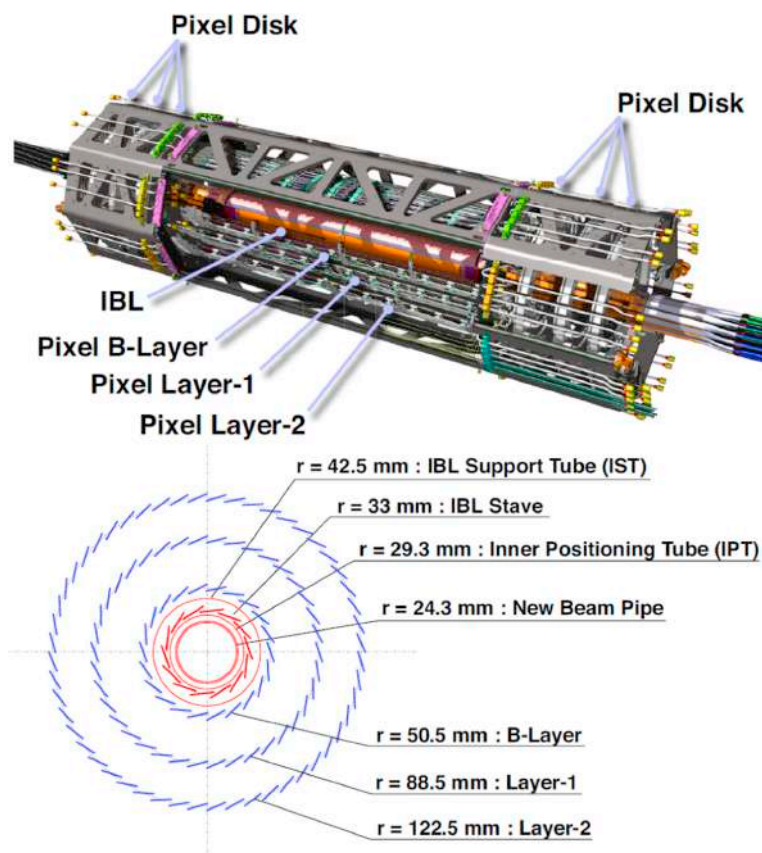


図 1.10: ピクセル検出器の構成図

1.3 HL-LHC 計画

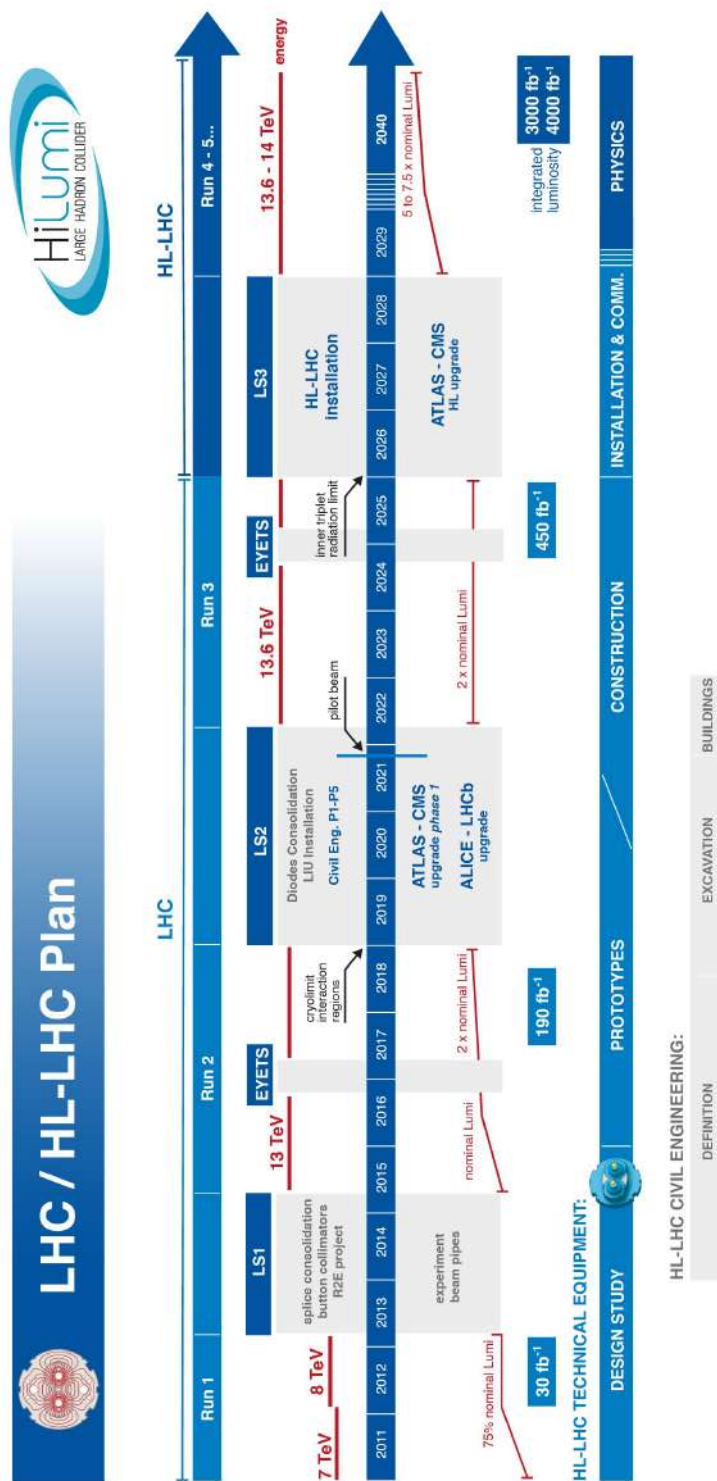


図 1.11: HL-LHC へのアップグレード予定

LHC の運転スケジュールを図 1.11 に示す。LHC では 2026 年から運転を止めて陽子同士の衝突頻度を増やす高輝度化のためのアップグレードが始まる。その後 2029 年から高輝度 LHC(High Luminosity LHC, HL-LHC) として運転を再開する予定である。このアップグレードに伴う変更を簡単にまとめたものが表 1.1 である。

表 1.1: LHC と HL-LHC の比較

	LHC	HL-LHC
重心系エネルギー [TeV]	13~14	14
瞬間ルミノシティ [fb^{-1}]	2×10^{34}	7.5×10^{34}
積分ルミノシティ [$\text{cm}^{-2} \text{s}^{-1}$]	300	3000
事象数/バンチ衝突	40	200

ルミノシティを増やし統計量が増えることで、より精密な測定が可能になり、ヒッグス粒子のさらなる性質や未知の粒子の発見を目指して標準模型を超える物理の探索を行う。一回のバンチ衝突における陽子の衝突回数が表 1.1 より、5 倍になることが想定されており、放射線による検出器へのダメージやトラック数の増加、トリガー数の増加など、検出器についても対応が求められ、アップグレードを行う必要がある。そのため LHC の衝突地点のうちの一つに置かれた ATLAS 検出器についてもアップグレードが予定されており、特に内部飛跡検出器を全て新型検出器に入れ替え、放射線耐性や読み出し速度の向上を目指す。この新型検出器を ITk(Inner Tracker) と呼ぶ。この新型検出器の断面図を図 1.12 に示す。

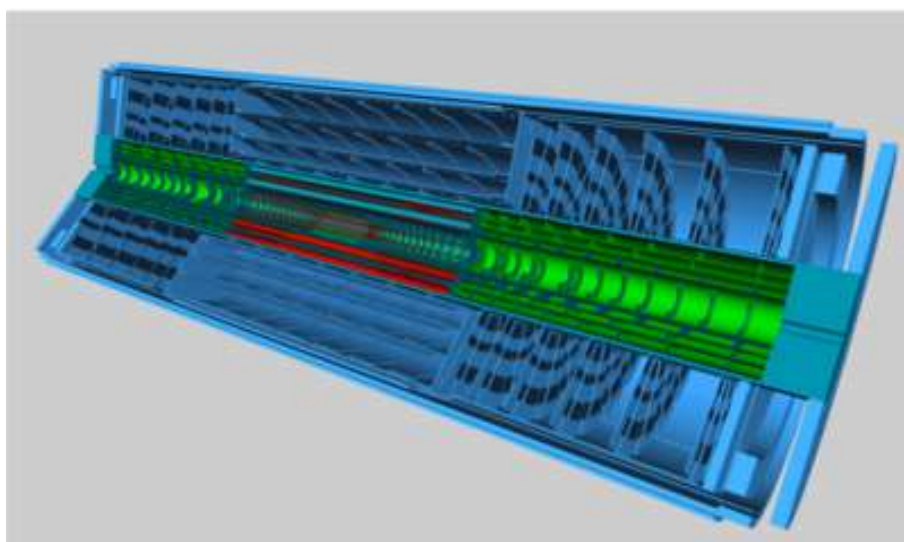


図 1.12: 新型飛跡検出器

1.3.1 新型ピクセル検出器

ATLAS 実験に用いる座標系について簡単に説明する。衝突点を原点としてビーム軸を z 軸、これに対して垂直に x - y 平面とする。 x 軸方向は原点から見て LHC リングに向かう向きで、 y 軸は

地上向きである。x-y 平面においては r - ϕ 座標系を用い、ビームラインからの距離を半径 r 、x 軸からの角度を方位角 ϕ とする。z 軸からの角度 θ を極角と呼び、特に ATLAS 実験では擬ラピディティ η を用いて表現する。以下が θ と η の関係式である。

$$\eta = -\ln(\tan(\theta/2)) \quad (1.13)$$

擬ラピディティ η の差はローレンツ不変であるため、ATLAS 実験において粒子の位置や検出器の配置を示す際に頻繁に用いられる。 $|\eta|$ が 0 付近の円筒の側面に対応する領域を領域をバレル部と呼び、 $|\eta|$ が円筒の底面に当たる領域をエンドキャップ部と呼ぶ。

図 1.12 のうち、緑色で書かれた領域がピクセル検出器であり、青色で書かれた部分がストリップ検出器である。図 1.13 に ITk のビーム軸方向の断面図を示す。原点は衝突点であり、横軸はビームラインに沿った z 軸方向、縦軸は衝突点からの半径 r である。赤で書かれた領域はピクセル検出器を表し、青はストリップ検出器である。現在 ITk 用の新型ピクセル検出器の開発に日本グループが関わっており、本論文では新型ピクセル検出器開発に関連するピクセルモジュールや、データベース関係について解説する。

またこの新型ピクセル検出器について現行のものとの変更点をまとめたものが表 1.2 である。表を見て分かるように必要となるモジュール数が大幅に増加する。そのためこのモジュールについて量産が必要となり、日本グループではおよそ 2800 個の生産を担当することになっている。

表 1.2: 現行のピクセル検出器と新型ピクセル検出器の比較

ピクセル検出器	現行	ITk
モジュール数	1744	9164
動作領域 [m ²]	1.6	13
チャンネル数	92M	5083M

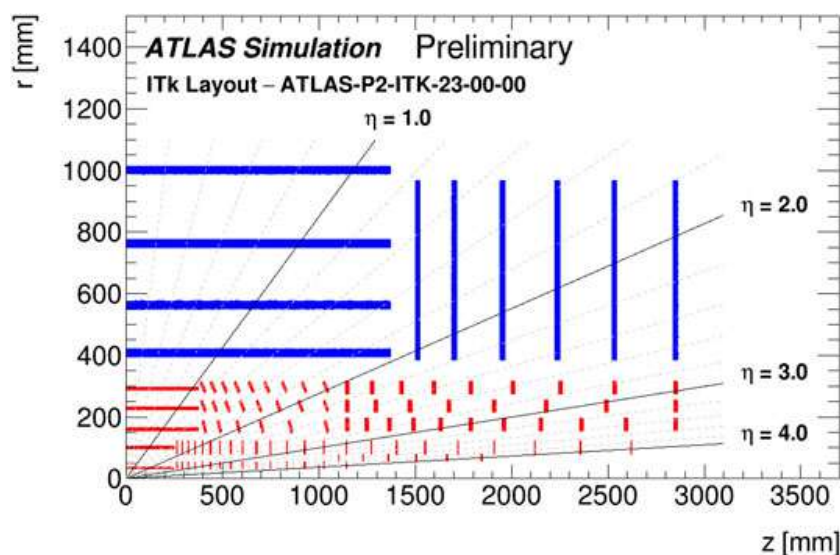


図 1.13: 新型ピクセル検出器の断面図

また ITk に搭載するセンサーについて紹介する。搭載される多くのセンサーはプラナーセンサーと呼ばれる構造をしている。プラナーセンサーはバルク部の表面に n^+ 型半導体と p^+ 型半導体を埋め込んだ構造をしている。現行のピクセル検出器に搭載されているセンサーはバルク部に n 型半導体を用いた n^+ -in- n 型のセンサーが搭載されている。しかし n 型半導体は放射線損傷を受けると、 p 型不純物としてふるまい格子欠陥が生じ、これが蓄積するとバルク部が p 型になる型反転を起こし、空乏化の挙動が変化する。HL-LHC では厳しい放射線環境が予想されるため、それを防ぐために ITk には n^+ -in- p 型のセンサーを搭載する。 p 型半導体は放射線損傷により全空乏化電圧が大きくなるが、全空乏化電圧が耐電圧を超えると部分空乏化による運用が可能である。加えて 3D センサーと呼ばれる構造をしたセンサーも存在し、現在は IBL の一部のみに搭載されており、ITk の最内層にも搭載される予定である。このセンサーは p 型のバルクに対してグラウンド電極のために n^+ を、HV 電極のためにバルク面に垂直にインプラント電極を配置している。電極間の距離が近いことからプラナーセンサーに対してはるかに低い電圧で全空乏化できることが大きな特徴である。さらに電子正孔対が電極に到達するまでの距離も近いいため格子欠損により生じたホールに電子が捉えられる可能性も低くなっている。そのため 3D センサーは放射線耐性の高いセンサーであり、放射線損傷の影響が大きい ITk の最内層に搭載する予定である。

第2章 ピクセルモジュールの量産と品質試験

1.3.1 節で説明した新型ピクセル検出器に導入するピクセルモジュールについて説明する。

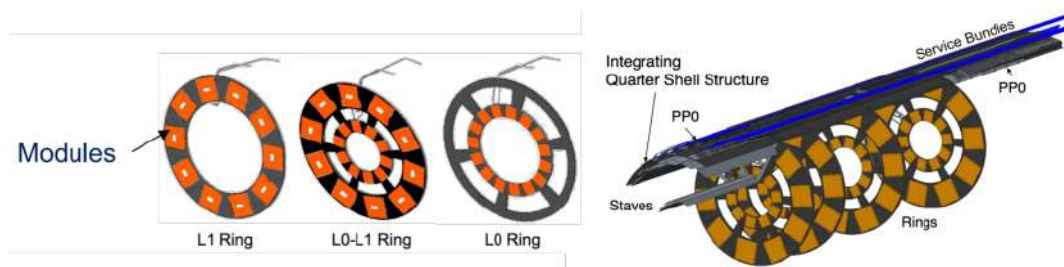


図 2.1: インナーシステムにおけるモジュールの配置

2.1 ピクセルモジュール

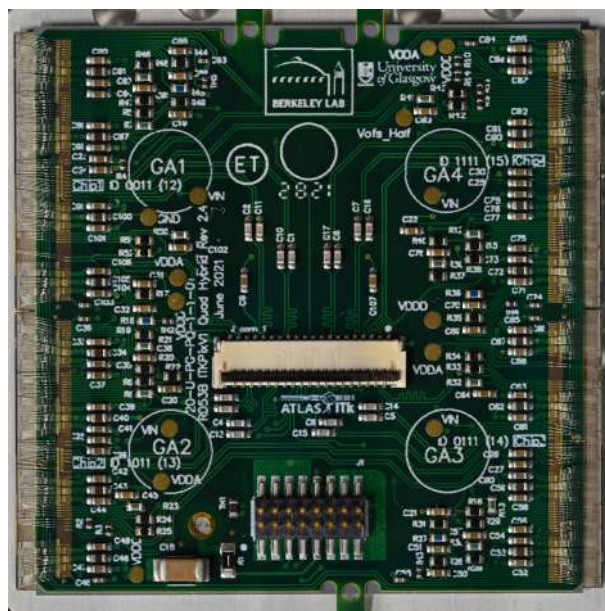


図 2.2: 現行のピクセルモジュール

ITk ピクセル検出器に搭載する ITk ピクセル検出器用モジュール (以下ピクセルモジュール) に

ついて説明する。現在 ITk に搭載するために生産が行われているモジュールを図 2.2 に表す。ピクセルモジュールは Front-End IC チップ (FE チップ) とシリコンセンサーをバンプボンディングで一つにしたベアモジュールと呼ばれるものとフレキシブル基板をワイヤーボンディングで接合したものである。つまり図 2.2 で見えている部分はフレキシブル基板である。表 1.2 で示したように現行のピクセルモジュールからアップグレードされ、FE チップあたりのチャンネル数も 18×160 (IBL に導入した FE チップは 80×336) から 384×400 となる。このピクセルモジュールについて図 2.2 の形をしており、FE チップ 4 枚に対しセンサーが 1 枚で構成されるモジュールを Quad module と呼び、FE チップ 3 枚に対しセンサーも 3 枚で構成されるモジュールを Triplet module と呼んでいる。Triplet module は図 2.3 である。モジュールの概略図は図 2.4 に示す。

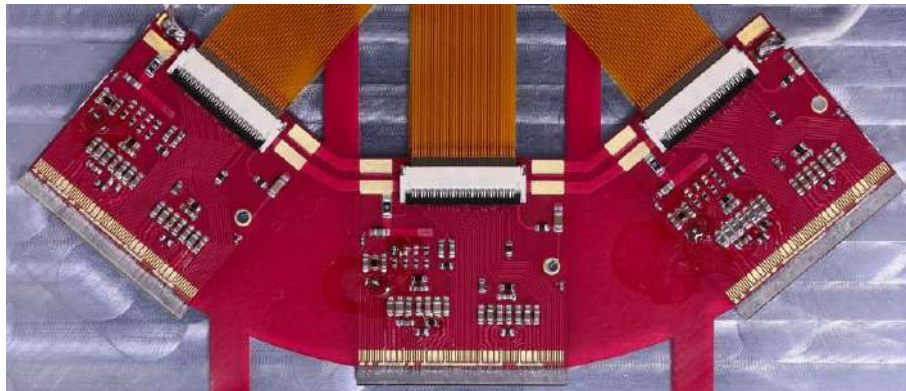


図 2.3: triplet module

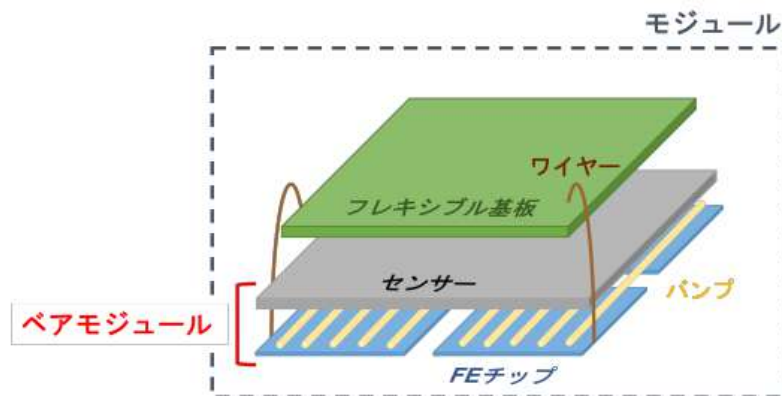


図 2.4: モジュールの構成図 (ベアモジュールとはセンサーと FE チップをバンプ接続したものである。モジュールとはベアモジュールとフレキシブル基板をワイヤーボンディング接合したもの)

主な信号の流れとして、まず荷電粒子がシリコンセンサーを通り、エネルギー損失に応じた電子ホール対を生成する。電子は電極により収集されアナログ信号としてバンプを通して FE チップに渡される。その後 FE チップが整形、増幅しデジタル信号に変えワイヤーを通してフレキシブル基板に転送し、読み出しシステムに転送される。

2.2 モジュールの量産

表 1.2 で示した通り、ピクセルモジュールについて約 10000 個の生産が必要であり、そのうち日本では約 2800 個の Quad モジュールを生産予定である。量産工程は、各組み立て期間に届いたセンサーと FE チップから構成されるベアモジュールとフレキシブル基板の接着から始まる。本論文執筆時における日本での生産としては、浜松ホトニクスでセンサー、FE チップを生産し、山下マテリアル (株) で PCB を生産している。センサー、FE チップをバンプ接続でベアモジュールとしたものを高エネルギー加速器研究機構 (KEK) で品質試験を実施し、PCB と組み合わせてモジュールにするため、千葉県館山市にあるハヤシレピック (株) 第 3 事業部において組み立てやさらなる品質試験を行っている。詳細の試験や作業内容は 2.3 節や 4.1 節で説明する。

2.3 モジュール組み立て手順

ハヤシレピックに送られたベアモジュールとフレキシブル基板はまず接着作業が行われ、その後専用のマシンでワイヤーボンディングを打つ。その状態で読み出し試験を $20\text{ }^{\circ}\text{C}/-15\text{ }^{\circ}\text{C}$ で行う工程を経て、問題なく通信ができる場合、パリレン高分子によるコーティングを行う。これはモジュールのセンサーと FE チップの端での放電を防ぐことや、湿気等からの保護が目的である。図 2.5 はパリレンコーティング済みのモジュールである。ピックアップポイント (GA1 などの記載がある丸で囲まれている場所) などを避けてパリレンコーティングされていることが確認できる。



図 2.5: パリレンコーティング後のモジュール

その後ワイヤー保護のための専用装置を取り付け、ワイヤーに直接衝撃が加わらないよう保護する。本論文執筆時では、このワイヤー保護装置は未完成であり、実際の組み立て工程においてはこの装置なしで進めている。その後、温度サイクル試験及び耐久試験を行い、このモジュールが実際に検出器に入った場合に問題なく作動するかどうか確認を行う。図 2.6 にモジュール組み立

て項目と品質試験項目を示す。このモジュール組み立て工程、品質試験項目については共通の試験が行われており、品質が保たれている。

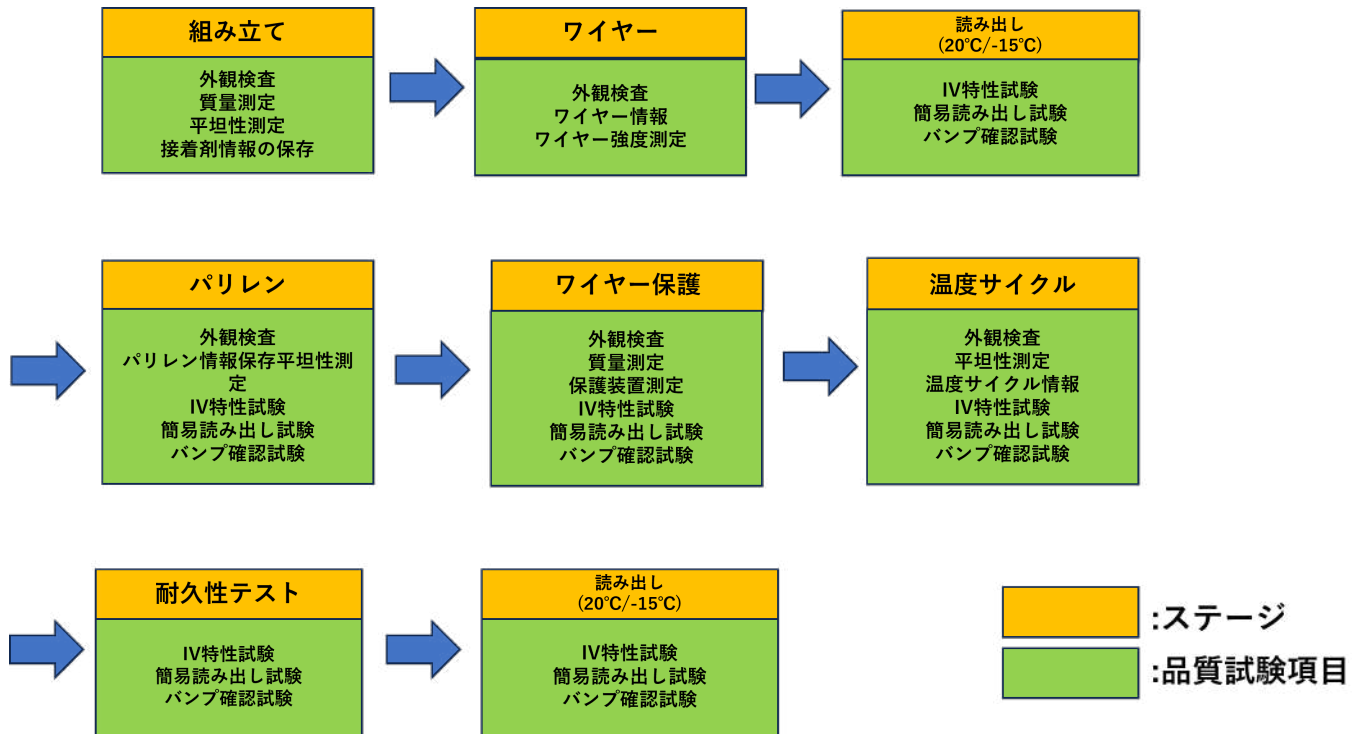


図 2.6: 組み立て項目と品質試験項目
 ステージとは生産におけるモジュール組み立て工程の区切りを表す。
 各ステージにおいて品質試験を実施する項目が定められている。

2.4 品質試験

モジュールやセンサーチップなどが思い通りの挙動をするかどうかを調べる試験であり、QC(Quality Check)と省略して表現される。主な目的は不良品を取り除くことや、モジュールごとの特性を調べることで実際に検出器に取り付けられる際に、それぞれの特性を考慮して取り付け場所が決定する。それぞれの品質試験に対して品質の基準を設けることで合格か不合格かを判断している。その判定については5.3節でまとめている。

現在行われている品質試験のうちいくつかを紹介する。

2.4.1 質量測定 (MASS MEASUREMENT)

モジュール全体の質量を0.1gの精度で測定する。各工程で質量の差を計算することで、直接測定することが困難なワイヤーの質量や接着材、ワイヤー保護装置などの質量を計算することができる。

2.4.2 外観検査 (VISUAL INSPECTION)

モジュール表面及び裏面をカメラを用いて撮影し、傷や汚れ等がないかを目視で検査する。撮影された高解像度画像はjpg形式で保存され、全体を36(6×6)分割に分けた拡大画像を用いて検査する。特に注意して検査するポイントとして、FEチップとフレキシブル基板を電気接続するワイヤーの接続場所を確認する。図2.7のように指定されたポイントにおけるワイヤーの有無をチェックしコメントに残しておくことで、今後問題が発生した場合でもどの段階で起きた問題かを探ることができ、再発防止につながる。

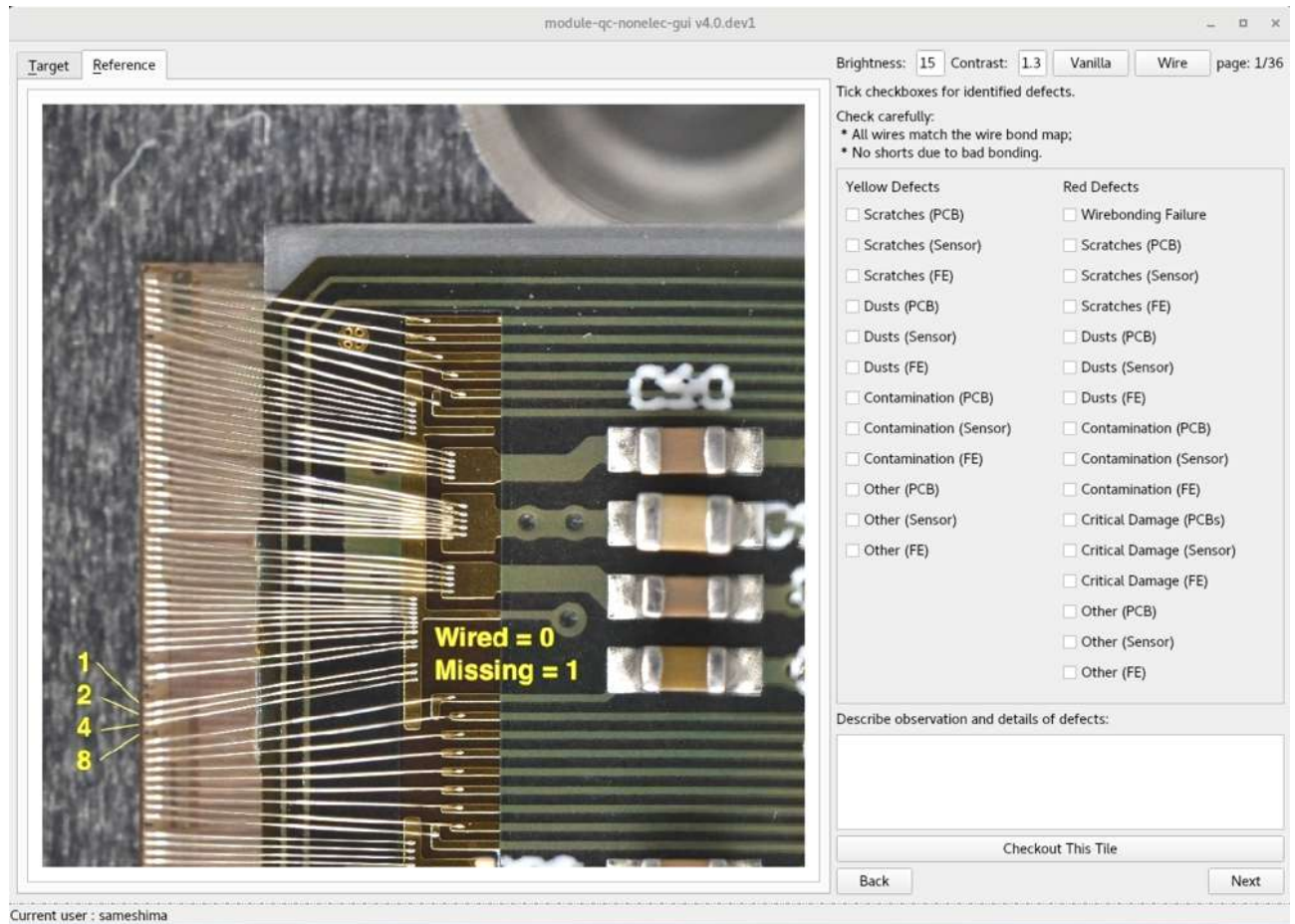


図 2.7: 外観測定の様子

また実際に使われている外観検査用の装置は図 2.8 である。



図 2.8: 外観測定用装置

2.4.3 サイズ測定 (METROLOGY)

図 2.8 を用いて撮影した画像を用いて測定を行う。モジュールの 3 次元位置座標を測定し、厚さやフレキシブル基板とセンサーチップのはみ出た部分の距離などを確認する。また測定装置は図 2.8 と同じものを使って測定を行う。

2.4.4 平坦性測定 (FLATNESS)

温度サイクルなどにより変形が発生し歪みを生む可能性があるためその前後に実施されている。また測定装置は図 2.8 と同じものを使って測定を行う。本論文執筆時では、この測定はソフトウェア・ハードウェア共に整備が不十分ということもあり、測定は行われていない。

2.4.5 ワイヤー強度測定 (WIREBOND PULL)

FE チップとフレキシブル基板をワイヤーボンディング接続するために用いるワイヤーの強度を調べるため、本来打つ必要があるワイヤー以外に、余分にワイヤーをモジュールに打ち込み、その部分に負荷を与えて切れるまでの強度を測定する。現在測定されているワイヤーの強度と頻度を図 2.9 にまとめた。

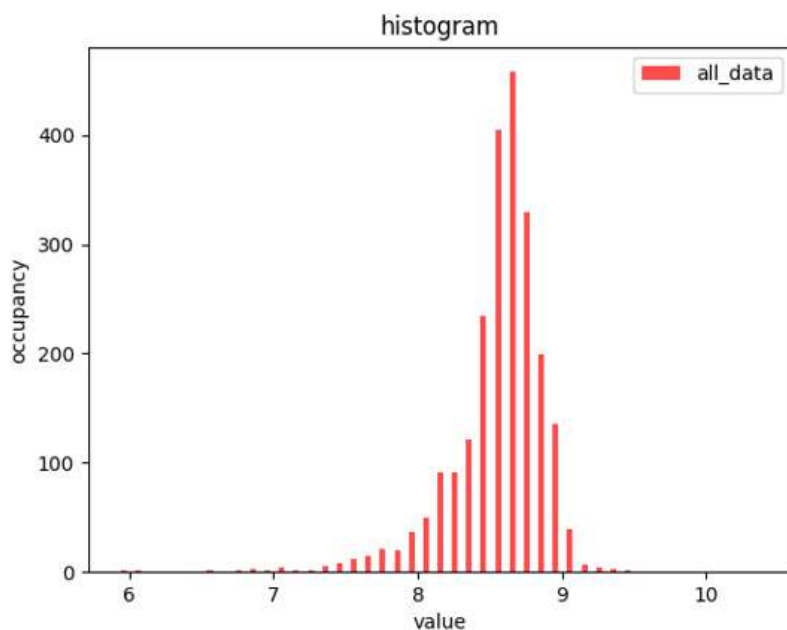


図 2.9: ワイヤの強度 (横軸) と頻度 (縦軸)

現在この試験はフレキシブル基板と FE チップにつながっているワイヤの壊れ方と、その時にかけた負荷の大きさを記録しており、それらの情報を用いてモジュールに打たれたワイヤが十分な強度で打たれているかを判断する。この試験における データの解析等は 5.3.2 節で説明する。

2.4.6 I-V 特性試験 (IV MEASURE)

センサーの電流-電圧特性を調べることで、輸送中や試験中などにおけるセンサーの故障やセンサー特有のショートなどを確認することができる。0V から 200V まで 5V ずつ段階的に印加する電圧を変えていき、その際の漏れ電流を温度、湿度と共に記録する。モジュール段階での試験としてこの I-V 特性試験を行う場合、ベアモジュールの段階での I-V 特性試験と比較して基準を超えていないか確認することも重要である。ただし漏れ電流は温度に強い依存性があるため、それを考慮して測定し比較する必要がある。一般的に漏れ電流は次の式で表される。

$$I_T = AT^2 \exp\left(-\frac{E_g}{2kT}\right) \quad (2.1)$$

ここで温度 T_1 のときに測定した漏れ電流を T_2 になおして温度を正規化することが可能である。

$$I_{norm} = I \left(\frac{T_2}{T_1}\right)^2 \exp\left(-\frac{E_g}{2k} \left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right) \quad (2.2)$$

ここで

$$E_g(T) = E_{g0} - SE_{ph} \left[\coth\left(\frac{E_{ph}}{2kT}\right) - 1 \right] \quad (2.3)$$

A 係数

I_{norm} 正規化した電流

I 測定した電流

T_1 測定温度

T_2 正規化するための温度

k ボルツマン定数

E_g バンドギャップエネルギー

E_{g0} 0Kにおけるバンドギャップエネルギー

E_{ph} 平均フォノンエネルギー

S 単位なしのパラメータ

を用いた。組み立て箇所の問題でこの漏れ電流の測定値が変わることがあるため、この温度の正規化を利用している。

また降伏電圧についても十分確認する必要がある。ATLASに搭載する大部分を占めるプラナーセンサーでは120V、放射線損傷の影響が大きい部分に搭載される予定である3Dセンサーでは25Vが基準となっており、それ以下で降伏電圧がみられる場合は基準を満たしていないという結果になる。

2.4.7 温度サイクル試験 (THERMAL CYCLE)

実際のITkでの運用において、正常でない温度変化を想定して、その後も使用できるかどうかを確認する試験である。この正常でない温度変化とは検出器内部での冷却機能が停止したり、逆に過剰に冷却されることである。この場合、前者では40℃、後者では-40℃の温度を想定している。この変化以上の温度サイクルを実際に行い、その後モジュールを測定し以上がないかを確認するものである。本論文執筆時におけるハヤシレピックでの生産における温度変化の幅は、-45℃から40℃を10サイクルし、最後だけ-55℃から60℃までのサイクルを行う。この試験において考えられるモジュールの異変として、熱膨張による平坦性の損失と、それにとまなうセンサー-FEチップ間における bumps 接続剥がれが危惧される。

第3章 品質試験結果管理用データベース

2.4 節で説明した試験項目について、それぞれのテスト結果をデータベースに保持しておくことで、何か問題が発生した場合に参照することを可能にする。また品質保証の根拠とすることもできる。3.1 節で ATLAS 実験で用いられているデータベースの紹介と役割を説明する。

3.1 データ管理用ソフトウェア構造図

ATLAS 実験で主に使われているデータベースは ITkProductionDatabase(ITkPD) と呼ばれる中央データベースと各拠点におかれる LocalDB と呼ばれローカルデータベースの二つがある。これらは二つの説明は 3.2 節と 3.3 節で行う。

この LocalDB にデータを登録するアップローダーの役割をしているものが、YARR と module-qc-nonelec-gui である。これらの説明は 3.4 節と 3.5 節で行う。図 3.1 はそれぞれのソフトウェアがやり取りする関係を図にまとめたものである。

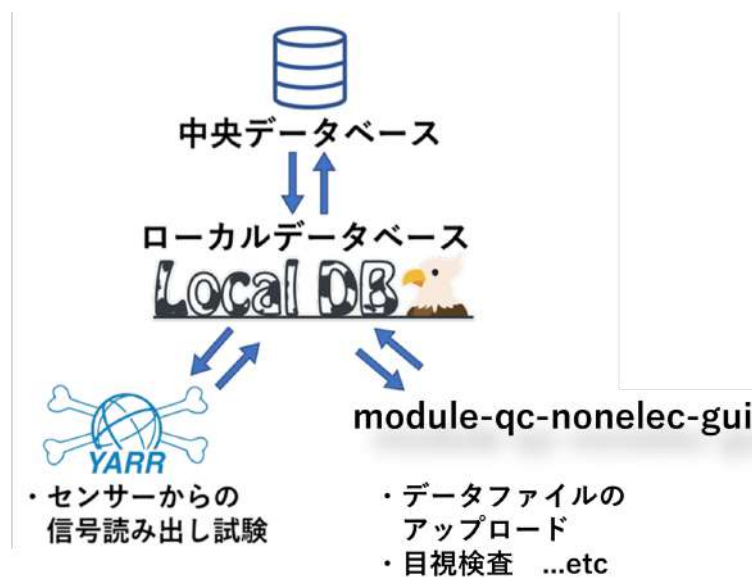


図 3.1: モジュール情報管理用データベースとその関係

3.2 中央データベース

ITkPD と呼ばれる中央データベースはチェコのユニコーン大学 (Unicorn University) が中心となって開発が進められており、チェコにデータベースサーバーが置かれている。ITk 用に作られる

ピクセルモジュールなどの基本データが格納されている。量産時に行われる品質試験については同じテストを複数回実施した場合でも、中央データベースへあげるデータは各工程で得られた結果のうち一つを選択してアップロードする。これは ITk に実装される検出器及び構成部品は約 50 万ほどになり、中央データベースに実施したテスト全てを保存すると膨大な量となるからである。またこの中央データベースにあげられたデータは実際の ITk 運用におけるモジュールの選定に使われる。擬ラピデティ $|\eta|$ が小さい領域では質量の大きい粒子の崩壊による荷電粒子が生成されやすく、物理解析に当たって興味の対象となるものは相対的に質量の大きい粒子が多いため、不良ピクセルの少ない良質なモジュールが選ばれる。また運転前後で検出器に損傷が起きていないかどうかを確認することができる。モジュールの試験項目に読み出し試験や漏れ電流の測定をおこなっているため、加速器内の強い放射線下で損傷が起きる可能性があるが、過去の測定と比較することで損傷の有無を確認することができる。

モジュールの追加や試験の登録などの方法は主に 2 つある。1 つは web ブラウザ上での操作であり、モジュールごとに用意されたページ上で一つずつ試験や情報の入力などが可能である。もう一つは API を使用して別のアプリケーションソフトウェアからアクセスする方法である。主に使用するものを以下の表に示す。

表 3.1: 中央データベースにアクセスできる主な API

getComponent	部品の情報取得
listComponent	部品の情報一覧取得
uploadTestRunResult	テスト結果をアップロード
createTestRunAttachment	テスト結果にファイルの添付

localDB ではこの API を利用してデータをモジュール単位で中央データベースに送信することが可能である。詳細は 3.3 節で解説する。

3.3 LocalDB

各研究機関に置かれるデータベースで、モジュール組み立て工程における試験結果を保存しておくツールであると同時に、中央データベースへのアップローダーとしての役割も持つ。データベース機能は MongoDB を利用しており、そこにためられたデータの操作や閲覧はウェブアプリケーションから可能である。MongoDB は MongoDB Inc. によって開発が行われ、最初の公開リリースは 2009 年に行われた。MongoDB は一般的に利用されているリレーショナルデータベースではなく、NoSQL(Not Only SQL) データベースである。この NoSQL とはデータ間の関係性の定義を行う必要がなく、データの拡張性が柔軟で分散処理を得意とするデータベースである。MongoDB は、データベース、コレクション、ドキュメントの順に細分化され、各ドキュメントはオブジェクト ID という ID で管理される。このドキュメントは JSON によく似た BSON という形式で保存されている。JSON(JavaScript Object Notation) とはデータを表現するための軽量なフォーマットであり、キーと値のペアを持つデータオブジェクトを記述するものである。BSON とは JSON によく似た形式のデータを扱い、binary 表記したデータを扱うため保存できるデータが JSON より多い。中央データベースで管理されるデータは 12byte の ID を持っており、mongoDB が生成するオブジェクト ID も 12byte であるためこの ID を関連付けて管理することができる。現在 localDB を運用する上で使用している mongoDB におけるコレクションをまとめたものを表 3.2 に示す。

表 3.2: localDB のコレクション名と詳細

データベース名	コレクション名	保存内容 (ドキュメント)
localdb	component	モジュール情報
	childParentRelation	モジュールと FE チップ、センサーの紐づけ
	QC.testRAW	module-qc-nonelec-gui からあげられるデータ
	QC.result	解析スクリプトを通した後のデータ
	institutions	読み出し試験実施場所
	pd.institution	中央データベースにおける所属機関の情報
	QC.testRuns_pdb_ldb_map	中央データベースと localDB の紐づけ
localdbtools	QC.tests	中央データベースのテストフォーマットを保存
	QC.module.types	モジュール構成部品情報
	message	localDB 上で残すコメント情報
	QC.stages	中央データベースで定められたステージ情報を保存
	viewer.query	読み出し結果キーワード、検索機能実行時に使用
	viewer.user	ローカルデータベースのユーザ情報
	componentType	各部品ごとの詳細データ
	viewer.tag.categories	試験結果につけるタグの情報

localDB は、あるモジュールを構成するベアモジュールや、さらにそのベアモジュールを構成するセンサーなどに関する情報を中央データベースからダウンロードすることができ、その LocalDB を保有する機関で品質検査を行いたいモジュールを指定してダウンロードすることができる。これらのデータは LocalDB を構築する mongoDB に蓄えられ、その情報を閲覧するためのウェブアプリケーションも開発されてある。それを表したものが図 3.2 である。

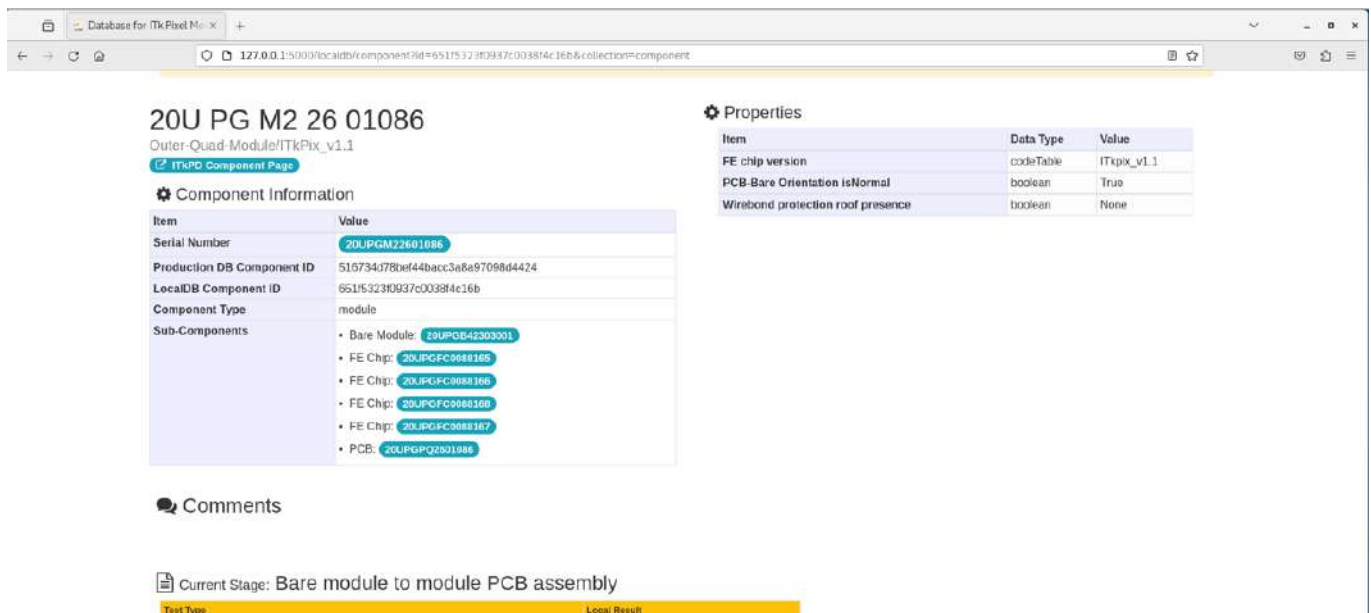


図 3.2: LocalDB の web ページ

LocalDB は各サイトで動作している mongoDB の情報を参照するため、中央データベースからダウンロードする際にモジュールのステージ情報や、品質検査に要求するデータなどを定義した情報を同時にダウンロードする。それらのデータは表 3.2 の QC.tests コレクションにおけるテストのフォーマットをモジュールごとに決めてドキュメントとして保存しているため、中央データベースが更新された場合、その更新後に中央データベースから localDB に引っ張ってきたモジュールに関してはそれらの更新が反映されるようになっている。中央データベースによる品質試験に要求する内容はリスト 3.1 のようにまとめられている。

リスト 3.1: 中央データベースが要求するフォーマットの例

```

1 {
2   "component": "...",
3   "testType": "IV_MEASURE",
4   "institution": "...",
5   "runNumber": "...",
6   "date": "2023-12-28T07:51:40.282Z",
7   "passed": true,
8   "problems": false,
9   "properties": {
10    "TEMP": 842.19,
11    "HUM": 426.11,
12    "ANALYSIS_VERSION": "HYKICQMEKZPIPOKCOCTNNXNMSHRCYGD"
13  },
14  "results": {
15    "IV_ARRAY": null,
16    "IV_IMG": null,
17    "BREAKDOWN_VOLTAGE": 528,
18    "LEAK_CURRENT": 208.92,

```

```

19     "NO_BREAKDOWN_VOLTAGE_OBSERVED": false,
20     "MAXIMUM_VOLTAGE": 900.37
21   }
22 }

```

これは I-V 特性試験におけるフォーマットを表示したものであり、key の部分が重要で value に入っている数値はあくまで参考値で実際の測定に関係があるものではない。

この LocalDB に品質試験を行ったデータをアップロードしていき、図 2.6 で示したステージを一つずつ進めていく。データのアップロードが正常に行われた場合、図 3.3 のように図 3.2 と同ページ下部に表示される。

Current Stage: Bare module to module PCB assembly

Test Type	Local Result
Mass Measurement (MASS_MEASUREMENT)	No TestRun!
Glue Information Module+Flex Attach (GLUE_MODULE_FLEX_ATTACH)	No TestRun!
Triplet Module Metrology (TRIPLET_METROLOGY)	No TestRun!
Visual Inspection (VISUAL_INSPECTION)	No TestRun!
Flatness (FLATNESS)	No TestRun!
Quad Module Metrology (QUAD_MODULE_METROLOGY)	TestRun Present

Candidate TestRuns of this Stage

LocalDB Object ID	TestRun Code	QC Passed	Inspector	Date	Analysis Version	QC Registration
656417ed22ccb0e06039ed97	QUAD_MODULE_METROLOGY	Passed	None	2023-11-27 13:15:39 JST(+0900)	1.0.11	

図 3.3: LocalDB のテスト済み項目一覧

この図のようにテスト結果がすでにアップロードされているものに関しては TestRun Present と表記されるが、まだアップロードされていないものに関しては No testRun と表示されるため、未実施のテスト項目が一目でわかるように工夫されている。

各ステージにおける必要なテストを終え先に進める場合は、基本的に中央データベースにデータを表 3.1 に書かれた API を利用してデータを送信し、localDB と中央データベースを同期する。その後どちらも新しいステージへと進めることで常に最新の情報が保たれることになる。これらの操作に関しても、LocalDB の web ページ上で操作可能であり、中央データベースにアップロードするテストを選択し、ボタンを押すことで全て内部的に処理してくれる。また、この LocalDB に品質試験を行ったデータをアップロードするアップローダは主に 2 種類存在しており、YARR と module-qc-nonelec-gui である。それらの違いや役割については 3.4 節と 3.5 節で説明する。

3.4 YARR

YARR(Yet Another Rapid Readout) はモジュール用の DAQ(Data Acquisition) ソフトウェアである。主な開発は Lawrence Berkeley National Laboratory (LBNL) によって行われている。実

際に YARR を用いて行う試験項目を以下にあげる。

- digital scan
各ピクセルについてデジタル回路の応答を確認する。
- analog scan
各ピクセルについてアナログ回路の応答を確認する。
- threshold scan
各ピクセルの threshold を確認する。
- ToT scan
ToT(Time over Threshold) と呼ばれるアナログ信号が threshold を超えた時間幅をデジタル信号に変化したものを測定する。
- noise scan
各ピクセルのノイズを確認する。
- source scan
放射線（日本での試験においては本論文執筆時は X 線）を照射し、各ピクセルの応答を確認する。

これらは ITk 運転時の温度環境を想定して、モジュール周辺の温度を変えて試験が行われる。試験を行う際の基本的な温度は 20℃と-15℃である。これらの試験が正常に終了すると、LocalDB にテストが upload される。YARR から送られた試験は LocalDB 上の YARR Scan List にまとめて表示され、時系列順に並べられている。試験を行ったモジュールのページからも確認することができる。それを表したものが図 3.4 である。

Top Page > YARR Scan List

YARR Scan List

Input keywords Partial match Perfect match

Found 15 scans in LocalDB.

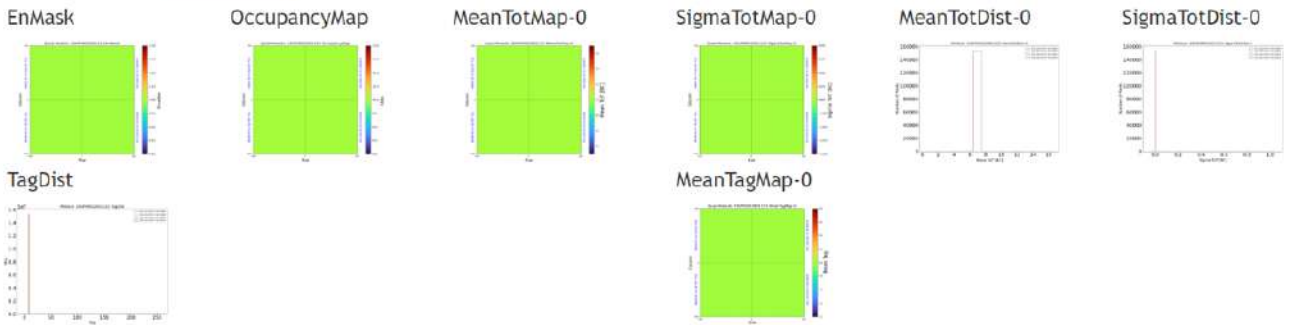
View	Module Name (ATLAS SN)	Chip Name (ATLAS SN)	Scan	User	Site	Date	Stage	Tags <input type="button" value="Create"/>
<input type="button" value="View Scan"/>	20J PG M2 26 01125	20J PG FC 00 87891 20J PG FC 00 87899 20J PG FC 00 87851 20J PG FC 00 87880	std_digitalscan	admin	hr	2023-11-29 15:58:46 JST(+0900)	MODULE/INITIAL_WARM	<input type="button" value="PFA"/>
<input type="button" value="View Scan"/>	20J PG M2 26 01125	20J PG FC 00 87891 20J PG FC 00 87899 20J PG FC 00 87851 20J PG FC 00 87880	std_totscan	admin	hr	2023-11-29 15:55:20 JST(+0900)	MODULE/INITIAL_WARM	<input type="button" value="TUN"/>
<input type="button" value="View Scan"/>	20J PG M2 26 01125	20J PG FC 00 87891 20J PG FC 00 87899 20J PG FC 00 87851 20J PG FC 00 87880	std_thresholdscan_hd	admin	hr	2023-11-29 15:51:43 JST(+0900)	MODULE/INITIAL_WARM	<input type="button" value="TUN"/>
<input type="button" value="View Scan"/>	20J PG M2 26 01125	20J PG FC 00 87891 20J PG FC 00 87899 20J PG FC 00 87851 20J PG FC 00 87880	std_retune_pixelthreshold	admin	hr	2023-11-29 15:50:32 JST(+0900)	MODULE/INITIAL_WARM	<input type="button" value="TUN"/>
<input type="button" value="View Scan"/>	20J PG M2 26 01125	20J PG FC 00 87891 20J PG FC 00 87899 20J PG FC 00 87851 20J PG FC 00 87880	std_retune_globalthreshold	admin	hr	2023-11-29 15:49:31 JST(+0900)	MODULE/INITIAL_WARM	<input type="button" value="TUN"/>
<input type="button" value="View Scan"/>	20J PG M2 26 01125	20J PG FC 00 87891 20J PG FC 00 87899 20J PG FC 00 87851 20J PG FC 00 87880	std_tune_pixelthreshold	admin	hr	2023-11-29 15:48:32 JST(+0900)	MODULE/INITIAL_WARM	<input type="button" value="TUN"/>

図 3.4: LocalDB における YARR テスト表示

個別のテスト結果が図 3.4 の中のリンクを押すことで表示され、詳細が確認できる。ここで digital scan の結果を参照した場合、図 3.5 のように表示される。YARR から送られる試験は図 3.5 のようにピクセルごとの 2 次元マップとともに結果を出力し、電気的な接続の正常性やその試験の結果を視覚的に確認し、判断することができる。また 2 次元マップによる結果だけでなく、一つ一つのパラメータについても LocalDB の中で Detail を選択すれば各パラメータの数値を一つずつ確認することができる。

Scan Result: std_digitalscan / run 8854 (Stage: MODULE/INITIAL_COLD)
 (ObjectID: 655ea751cf9f880c384c9856)

Result Plots [Regenerate Plots](#)



Details

Scan		Output Data			
Key	Data	Type	FE Chip Serial Number	Display	Download
runNumber	8854	ctrlCfg	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/>
testType	std_digitalscan	dbCfg	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/>
stage	MODULE/INITIAL_COLD	siteCfg	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/>
component	20UPQH22941123 20URFC0007890 20URFC0007898 20URFC0007899 20URFC0007879	userCfg	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/>
startTime	2023-11-29 16:56:12 .JST(+0900)	scanCfg	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/>
finishTime	2023-11-29 16:55:56 .JST(+0900)	feCfg.before	20URFC0007890 20URFC0007898 20URFC0007899	<input checked="" type="checkbox"/>	<input type="checkbox"/>
user	Ykage		20URFC0007879	<input checked="" type="checkbox"/>	<input type="checkbox"/>
site	fr	feCfg.after	20URFC0007890 20URFC0007898	<input checked="" type="checkbox"/>	<input type="checkbox"/>
targetCharge	-1			<input checked="" type="checkbox"/>	<input type="checkbox"/>
targetTot	-1			<input checked="" type="checkbox"/>	<input type="checkbox"/>

図 3.5: digital scan 結果確認ページ

3.5 module-qc-nonelec-gui

module-qc-nonelec-gui は python3.7 とその GUI 作成ツールである PyQt5 を使って実装された GUI(Graphical User Interface) である。module-qc-nonelec-gui の主な役割は、モジュール外観検査などの YARR によるデータ読み出しを用いない試験を LocalDB にアップロードすることである。先行研究 [8] によって開発された QC-helper のメンテナンス性の向上のためのリファクタリングや localDB への通信に利用する他のツールとの互換性のための名称変更を施し、module-qc-nonelec-gui となった。ここで module-qc-nonelec-gui の構造について説明する。図 3.6 は module-qc-nonelec-gui を動かすに当たって特に重要な部分を簡潔にまとめたものである。main.py が module-qc-nonelec-gui の根幹であり、これを起動することで GUI が立ち上がる。configuration/以下に書かれた json ファイルによって表示するテストを決定する。quadmodule, stave, ring 用に別々の config ファイルが用意されている。config_default.json の中に、quadmodule, PCB, BAREMODULE の内容が書かれている。加えて config_triplet_r.json と config_triplet_s.json があり、quadmodule と stave, ring で表示するテストを分けることができる。これは寸法測定試験において、quad と triplet では同じステージであるにも関わらず、中央データベースから要求されるフォーマットが異なることからである。

全てのモジュールにおいて共通する部分(図 3.7 における 1~3 ステップ)は GUI/以下に書かれたもので表示している。LocalDB と通信してデータを受け取る際には、dbinterface/の中のファイルが呼び出される。図 3.7 における 4 ステップ以降はテスト別に規定されており、それぞれのディレ

クトリを qc_tests/以下に準備している。

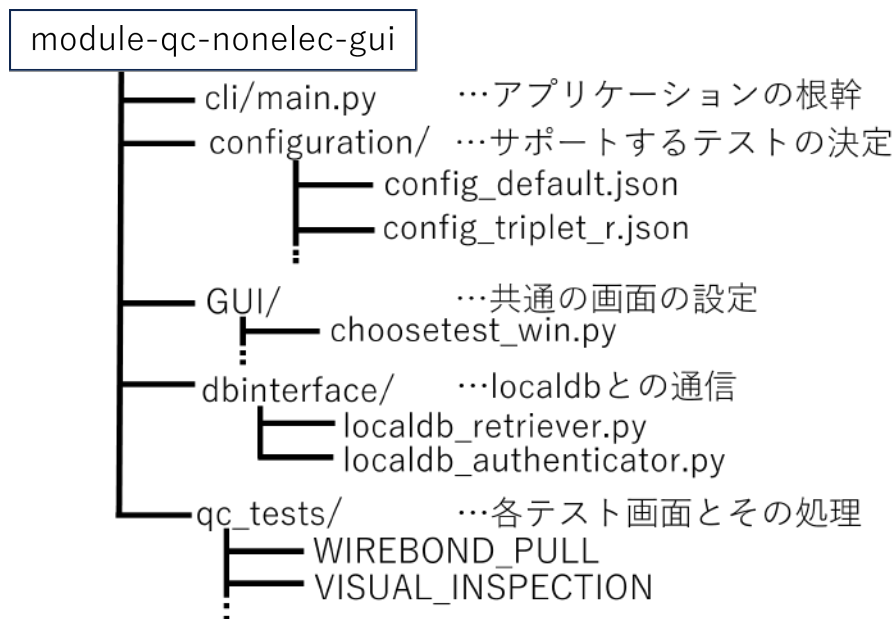


図 3.6: module-qc-nonelec-gui の内部構造

ここで現在サポートしている試験項目を以下に表す。

- 外観検査
- 平坦性測定
- ワイヤー強度測定
- 寸法測定
- I-V 測定
- BAREMODULE I-V 測定

これらの試験内容については 2.4 節で説明している。

次に module-qc-nonelec-gui の使い方及び流れについて説明する。図 3.7 が実際の操作手順を表したものである。

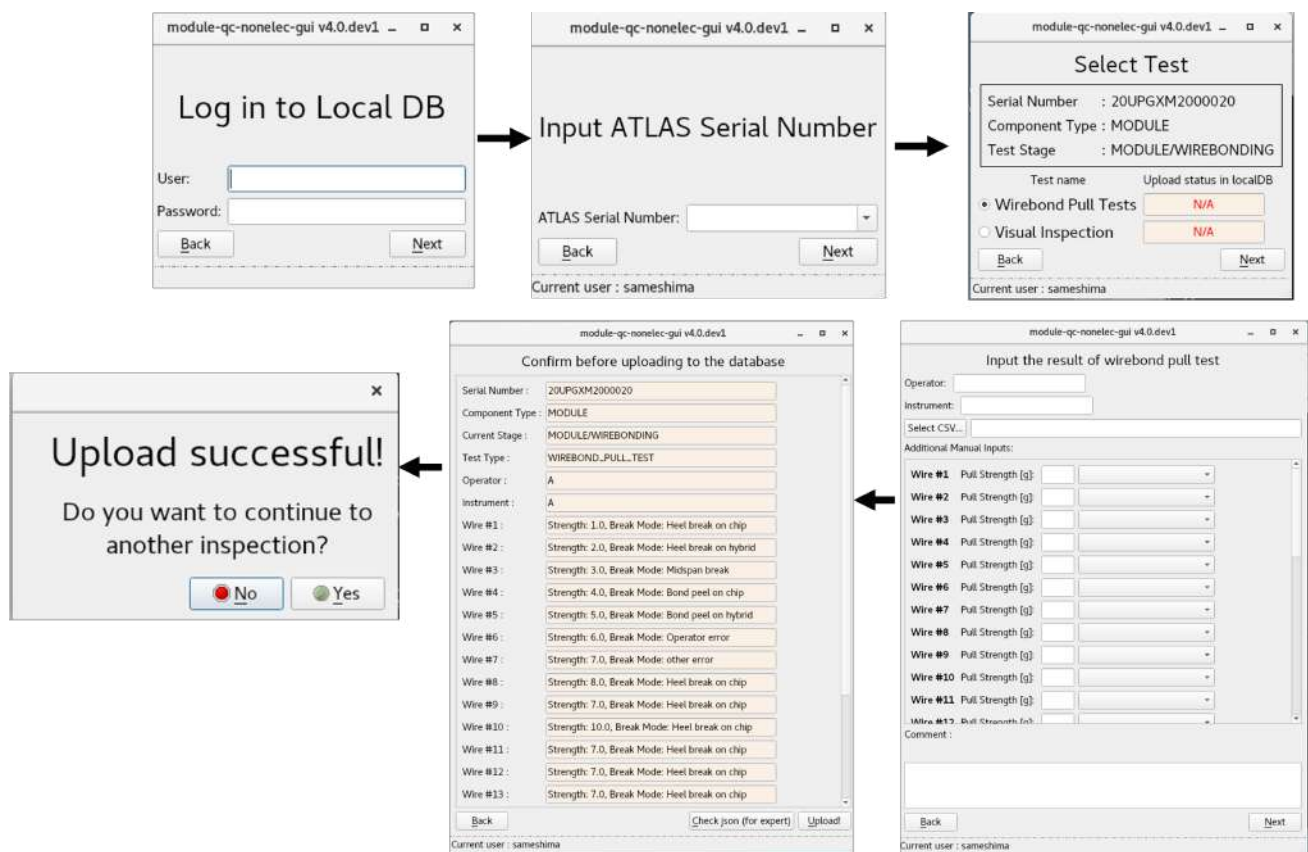


図 3.7: module-qc-nonelec-gui の手順

初めに module-qc-nonelec-gui を起動すると、LocalDB へのログインが求められる。次にその LocalDB を参照し、対応するモジュールシリアルナンバーをプルダウン方式で選択できる。またそのモジュールのテストステージを参照し、そのステージに合うテストのみが表示される。(図 3.7 の場合、WIREBONDING (ワイヤー) ステージのため、WIREBOND PULL と、VISUAL INSPECTION が表示される) その後アップロードしたいテストを選択し、テストごとに作られた入力画面に移動する。その後確認画面を経て、データが正しいことを確認し LocalDB に LocalDB の API を用いてデータを送信する。ここで nonelec-gui でサポートしている多くのテストに共通する json ファイルを選択してその中身を LocalDB に渡す処理を説明する。

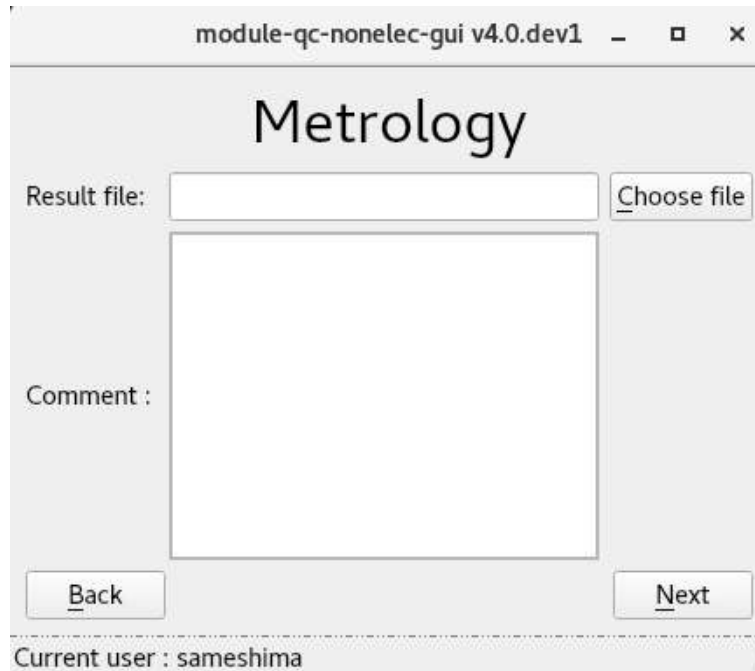


図 3.8: module-qc-nonelec-gui におけるファイルの選択

図 3.8 における `choose_file` ボタンを押すと、PyQt5 のライブラリである `QFileDialog` を使ってファイル選択ダイアログが表示される。ユーザがファイルを選択し `Next` ボタンを押すことで次の処理に移る。例として `metrlogy` という寸法を測定する試験の場合、`metrology machine` と呼ばれる測定装置が各パラメータを測定し、その結果は `db.json` という json 形式で保存されている。このファイルの中身が間違っていないことの確認や、ユーザが入力するファイル選択を間違えていないことの確認のため、それらを検証する方法として `jsonschema` による検証を採用している。`jsonschema` とはデータの形式や値の範囲など様々なルールを指定することができるスキーマ言語とよばれる言語のうちの一つである。module-qc-nonelec-gui における `jsonschema` はキーの名前と値の型を指定し、違うファイルが選択されることを排除している。

第4章 プリプロダクション（生産状況とソフトウェアの運用）

HL-LHC に向けて ATLAS 実験では多くの ITk ピクセルモジュールが必要なことを 2.2 節で述べたが、本論文執筆時では、その量産開始を目前に控え、試験工程確認等のために約 10 分の 1 の数量のプリプロダクションを実施している。このプリプロダクションは 2023 年 6 月 1 日から開始し量産に向けた準備が進められている。本量産は 2024 年 4 月～6 月頃の開始を目指している。本章では現在のプリプロダクションにおける設備や進捗を紹介する。

4.1 館山ハヤシレピックにおける生産

ハヤシレピック (株) 第 3 事業部における生産は、KEK において品質検査が済まされたベアモジュールとフレキシブル基板の接着から始まり、モジュールとなった後の品質検査を行う。実際に行う内容は図 2.6 の内容を実施する。プリプロダクション開始から約 5 か月間は 1 日 1 台のペースでモジュール作成を行い、品質検査を済ませてきた。実際のプロダクションに向けて生産ペースを高めていく必要があり、11 月以降は 1 日 3 台ペースで生産が進められている。また 3.3 節で説明した localDB サーバーを配備しており、プリプロダクションで生産したモジュールの品質管理データをその場で確認しながら試験を進めている。現地ネットワーク以外、各研究機関からのリモート環境であってもでも ssh 接続を用いて localDB へアクセス可能であり、オンラインでの作業も積極的に導入することで生産効率向上を目指している。現地ネットワークにリモート接続を行うことで localDB の操作のみならず、DAQ 機器の操作も可能であり、多くのユーザが品質試験に携わることができる。

4.2 ハヤシレピックにおける設備と体制



図 4.1: 館山ハヤシレピックにおける作業環境

品質試験等を行うエリアを簡単に表したものが図 4.1 である。ここでは組み立てそのものは行わず、読み出し試験やその操作、モジュールの管理などがメインである。図 4.1 における右側にあるクーリングボックスの説明をする。図 2.6 における読み出し試験等を行う装置になっており、モジュール設置部は断熱性の高いスタイロフォームで囲まれている。読み出し試験は常温 (20℃) と低温 (-15℃) で操作する必要がある、その点においてスタイロフォームは密閉性や保温性に優れている。しかしこの装置で読み出し試験を行い温度を下げていくと、スタイロフォーム内の空気は容易に露点に達してしまい結露を起こしてしまう。スタイロフォーム内で結露が起きると、モジュールが直接濡れることになり故障の大きな原因になりうる。そこで乾燥窒素をスタイロフォーム内に十分に充填し、露点を十分に下げている。実際の装置は図 4.2 に示す。

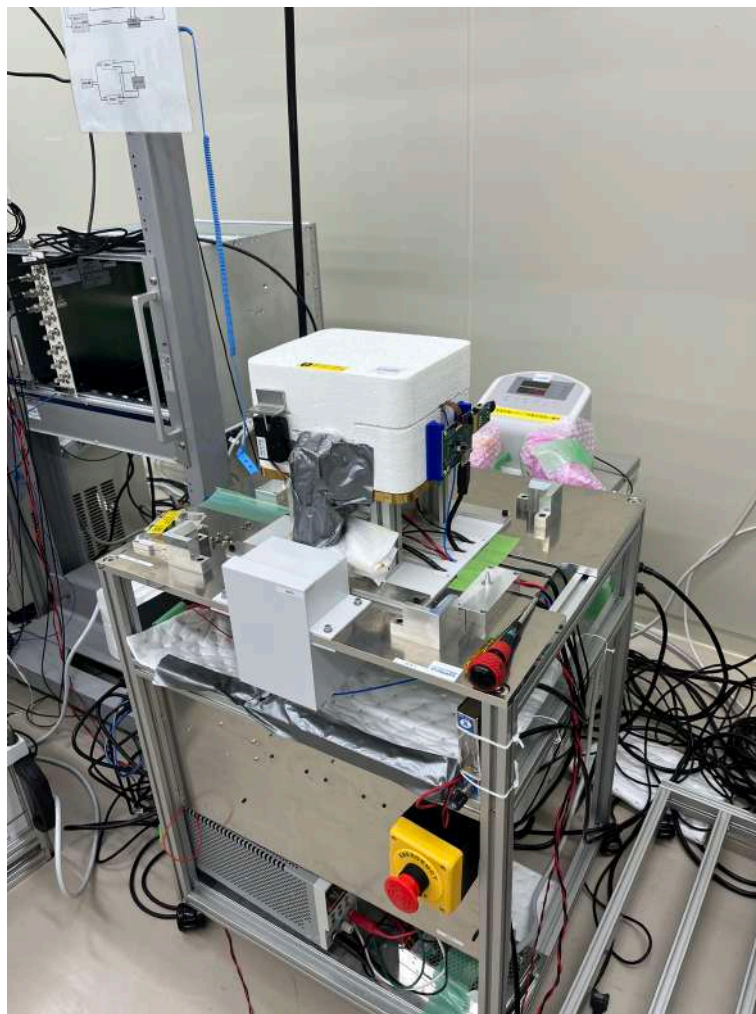


図 4.2: 読み出し品質試験用装置

図中央に写っている装置が読み出し試験用装置（クーリングボックス）である。読み出し試験が行われている間は錠がかけられ密閉性をあげる役割と何らかの要因でふたが開いてモジュールが結露することを防いでいる。解錠や設定温度の変更、電圧の on/off などは基本的にリモート操作で行うように設計され、モジュールに危険が起こる可能性のある操作（クーリングボックス内の露点と温度が近くなるなど）が行われた場合、インターロックがかかるように設計しており、モジュールにとって安全な操作が可能になっている。スタイロフォーム内側を示したものが図 4.3 である。

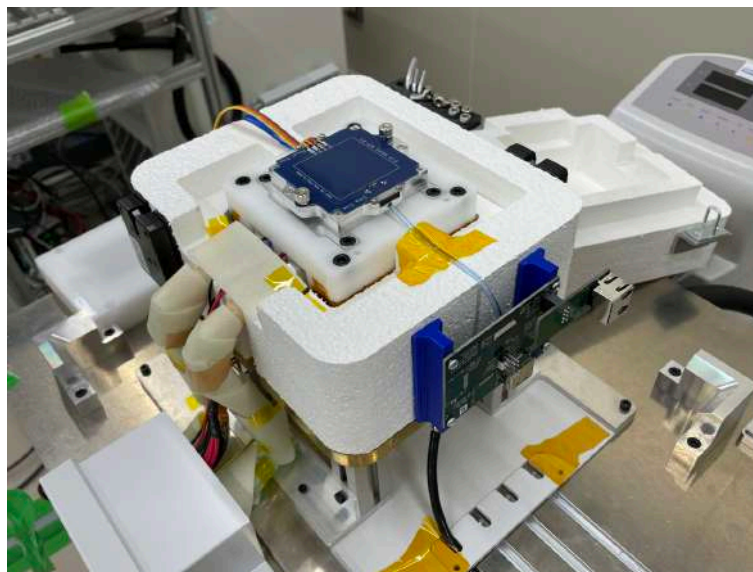


図 4.3: 読み出し品質試験用装置内側

内側はポリアセタール樹脂(デルリン)で構成されたケースになっており、その周りにはヒーターが巻きつけられている。クーリングボックス内の温度変化はペルチェ素子にかかる電流を調整して決定している。ペルチェ素子により発熱した面は、チラーポンプによる低温の不凍液の循環により吸熱を行っている。一方でモジュールを低温から戻す場合には、デルリンケース周りに巻きつけられたヒーターを利用し、目標温度に素早く達することができるように工夫されている。モジュール上にかぶせる蓋が温湿度センサー付きで、モジュールの温度を直接測定できる。モジュール温度や露点、室内の温度、モジュールにかかっている電圧などクーリングボックスや測定に関するデータは、時系列データを扱うことが得意なデータベースである InfluxDB に蓄えられており、そのデータを Grafana を用いてリアルタイムモニターをしている。この Grafana [20] とは Grafana Labs によって開発されているオープンソースツールであり、様々なデータを一つの画面上にまとめて可視化することができる。グラフの配置や大きさの変更や時間指定、リフレッシュタイムの設定など細かく表示方法を設定することができる。これにより、異常の早期発見や遠隔での状況把握が可能になっている。実際に使われている Grafana で表示している web ページが図 4.4 である。このクーリングボックスは本論文執筆時では、3 台設置してあり、この Grafana による表示も、クーリングボックス三台それぞれのページが用意されている。他には部屋内の温湿度確認ページやどのステージでの作業を行っているか確認できるページが用意されている。

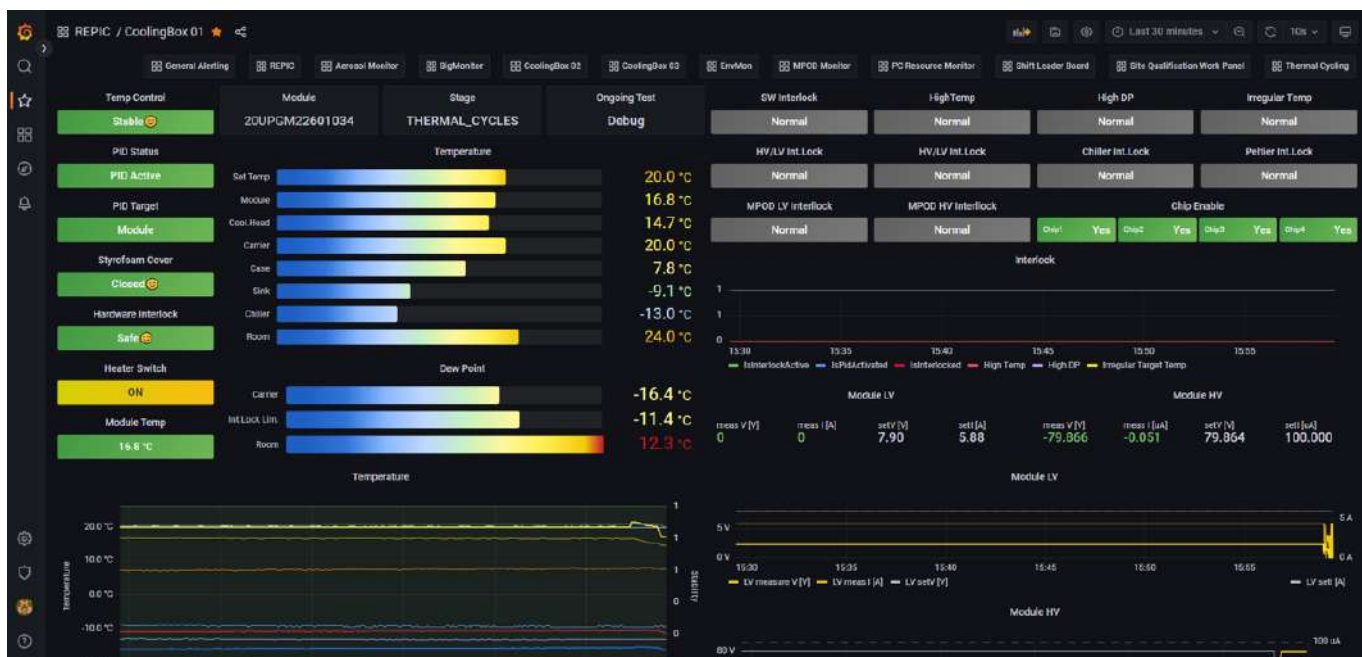


図 4.4: Grafana による時系列データの監視

続いて品質管理試験に用いられる他の装置について説明する。3.4 節で示した読み出し試験のひとつの X 線を用いてバンプ剥がれ評価を行う試験 (source scan) がある。モジュールに X 線 (50kV, 30uA) を一定時間照射し、X 線がセンサーにヒットし、ピクセルごとのヒット数を読み出すことでバンプ接続を確認することができる。測定結果は全て LocalDB に upload される。実際にアップロードされる結果を図 4.5 に示す。

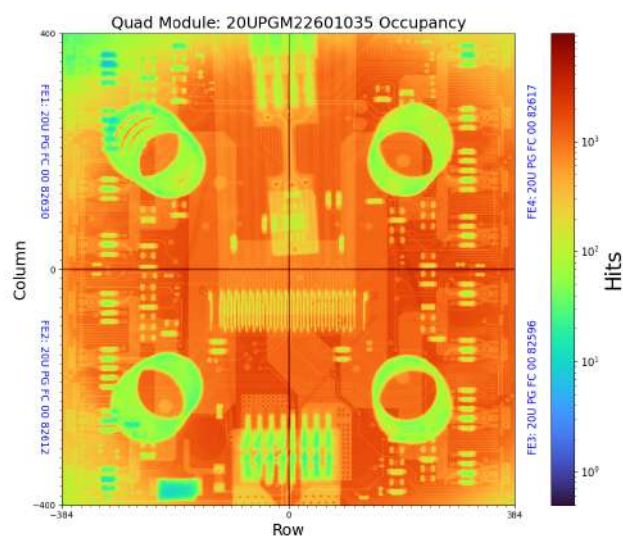


図 4.5: X 線照射結果

ヒット数が多い箇所は赤く表示され、物質量の多い箇所は比較的薄い色で表示されている。バンプ剥がれが見られる場合、多くは信号が伝わらないため、黒く表示された箇所があればバンプ剥がれを疑う必要がある。

放射線を照射する装置は4.6であり、図4.2の読み出し品質試験装置の上にかぶせるようにして設置し、セーフティーロックを外し、運転ボタンを押すとX線が照射されるため、照射する時間は周囲への安全性を考え、測定試験を行っている間のみにとどめる。



図 4.6: X線照射装置

次に温度サイクル用恒温槽の説明をする。2.4.7節で説明した試験を行う装置であり、一度の測定におよそ18時間かかる見込みの試験である。プリプロダクションの段階では実質的にこの測定スピードで問題ないが、本生産段階ではこのペースでは間に合わないため、一度に試験できるモジュールを8モジュールにすることで試験スピードをあげる予定である。



図 4.7: 温度サイクル用恒温槽

次に説明するものがデシケーターである。デシケーター内は温度湿度が均一に保たれており、組み立て終えたモジュールやパリレンコーティングのための出荷を控えたモジュールなど、現在直接試験を行う必要はないが順番待ちをしているモジュールの保管庫になっている。デシケーター内の温湿度を一定に保つためには扉が必ずしまっている必要がある。そのため扉との接触面に銅テープをはり電氣的に接触しているかどうかで扉開閉の判断を行っている。この結果は Grafana 上でリアルタイムで監視されており、一時的な操作の可能性を除くため、20分以上扉が開いた状態が続くと通知が届く仕組みができています。実際のデシケーターの写真を図 4.8 に示す。



図 4.8: モジュール保管用デシケータ

4.3 プリプロダクションの経過報告

本論文執筆時(1月7日)では102個のモジュールが新規に組み立てられ、順次試験が行われている。試験が一番進んでいるモジュールは図 2.6 における最終工程の読み出し試験 20℃まで終了しているが、-15℃での測定を残しているため、試験工程を全て終了したモジュールはまだ存在しない。品質試験の結果として10月頃までのプリプロダクションにおいては不良とみられるモジュールがいくつか確認された。不良と判断される理由に、低温でモジュールの読み出しができなくなることが多くあげられた。図 4.9 に 20℃と-35℃の場合のモジュールと通信可能なモジュール用電圧の立ち上げ速度を示す。

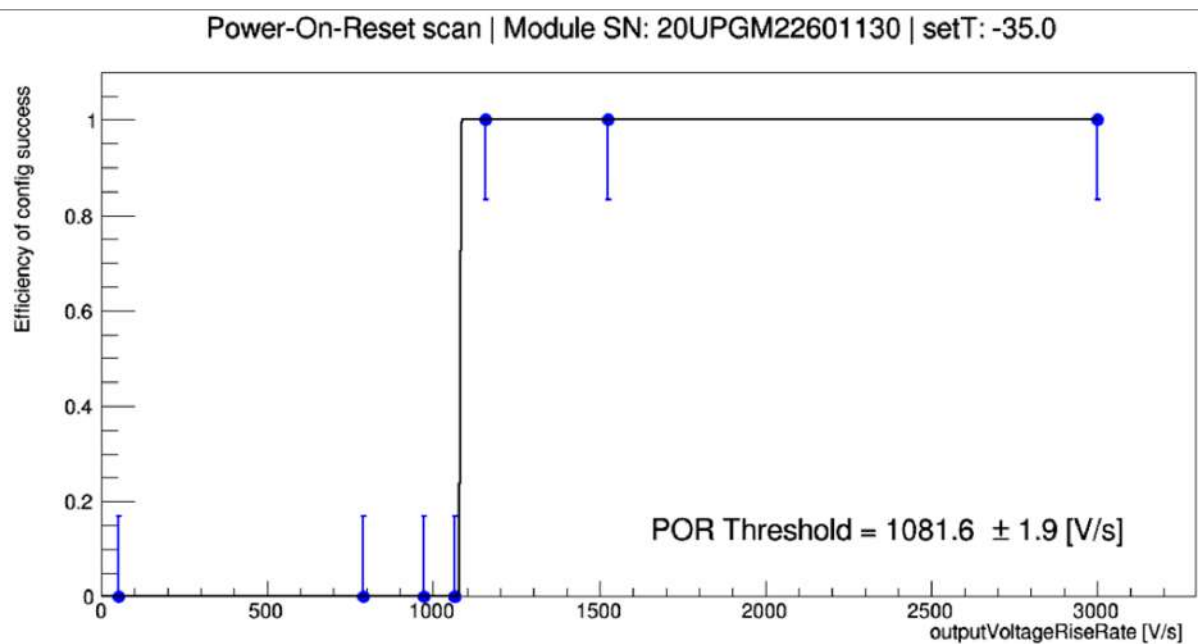
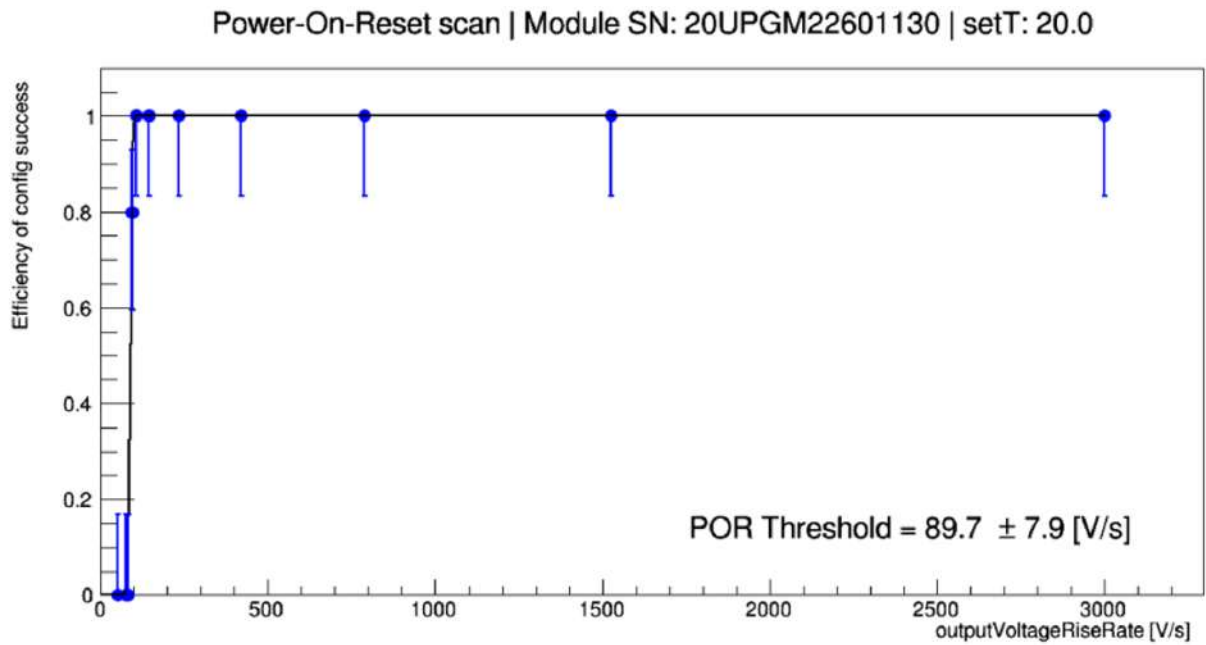


図 4.9: 電圧立ち上げ速度によるモジュールとの通信可能な領域の比較 (上:20℃, 下:-35℃)

横軸は設定した電圧の立ち上げ速度を示し、縦軸は通信がうまくいった割合を示している。二つのグラフを比較することで、モジュールと通信可能な電圧立ち上げ速度の許容範囲が、温度に依存していることがわかる。この通信可能になる立ち上げ速度はモジュールごとにバラバラで、それらの統計はとられておらず、なぜ低温で通信できないか分からず当初は混乱していた。10月頃までの品質試験においてこれらの設定値がデフォルトで100に設定されており、モジュールと通信できないと判断されるものが多く存在した。そこでこの立ち上げ速度について議論が起こり、

現時点では多くのモジュールが読み出し可能になる 3000 に設定し、低温で読み出し可能になっている。実際に 7 月 31 日～8 月 4 日、10 月 16 日～20 日の期間は現地で、11 月 6 日～10 日、11 月 20 日～24 日の期間はオンラインでプリプロダクションに参加しモジュール量産のための品質試験を行った。10 月 16 日～20 日の週にこの読み出し速度に関する話が上がり、当初は設定値を 500 に設定し、一度低温で読み出しできず不良と判断されたモジュールを再度測定したところ読み出し可能になるモジュールが現れ、この電圧立ち上げ速度が原因の一つと判断されることになった。他に不良と判断されたモジュールの特徴として、読み出し試験を行う際に、4 つある FE チップのうちの一つあるいは複数のチップが応答を返さないことが多く確認されている。以下に実際に私が経験した内容と対処法を示す。図 4.10 にこの現象を起こした際のモジュールと通信した場合の出力を示す。

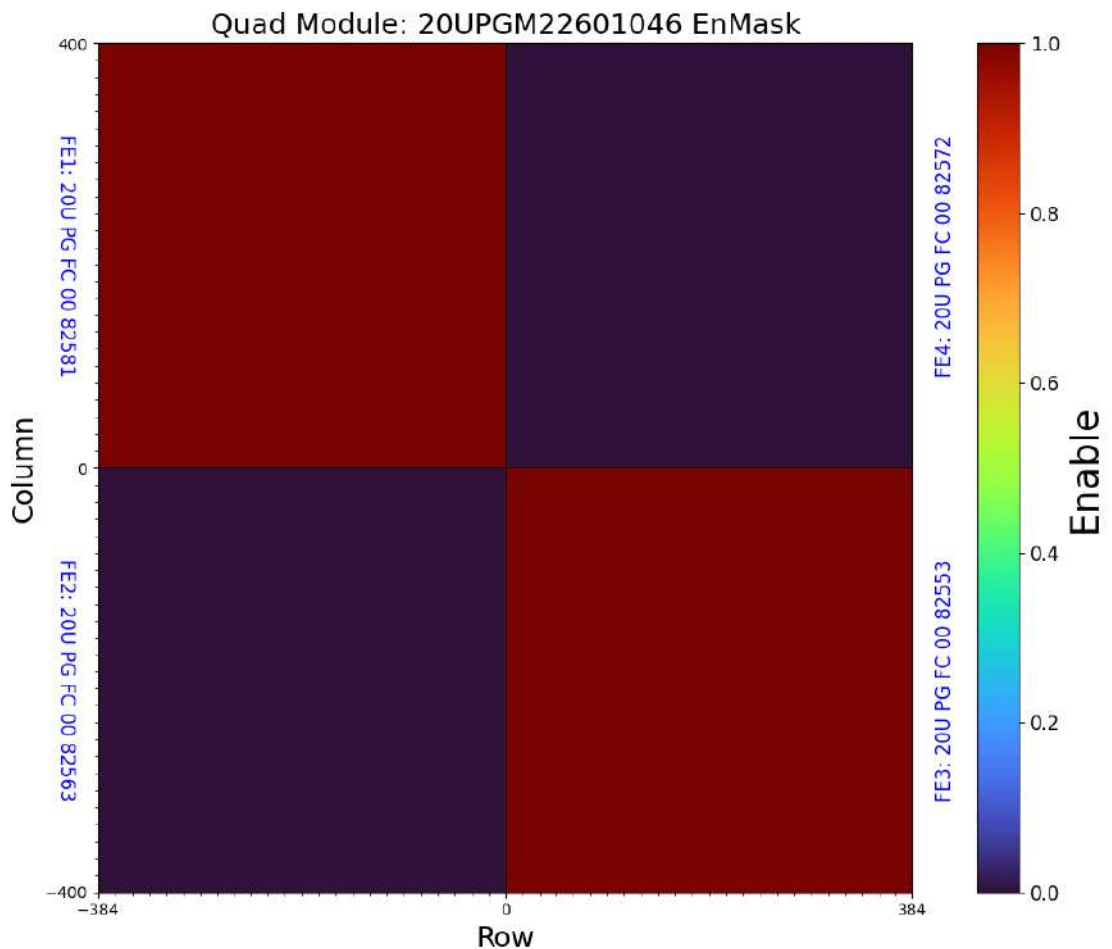


図 4.10: モジュールと通信を行う際に二つの FE チップから応答がない様子

本来ならば全て同じ色を示すことが正しい結果であるが、図のように左下と右上の FE チップからの応答がなくなっている。この現象から考えられることとして、この FE チップ内のコラムに不良なピクセルが混ざっていることである。一枚の FE チップは 54 ビット (16bit×3+6bit) で操作可能で不良なピクセルを隠して試験を進めることができる。実際に不良コラムを探し、マスクし

たものを図 4.11 に示す。

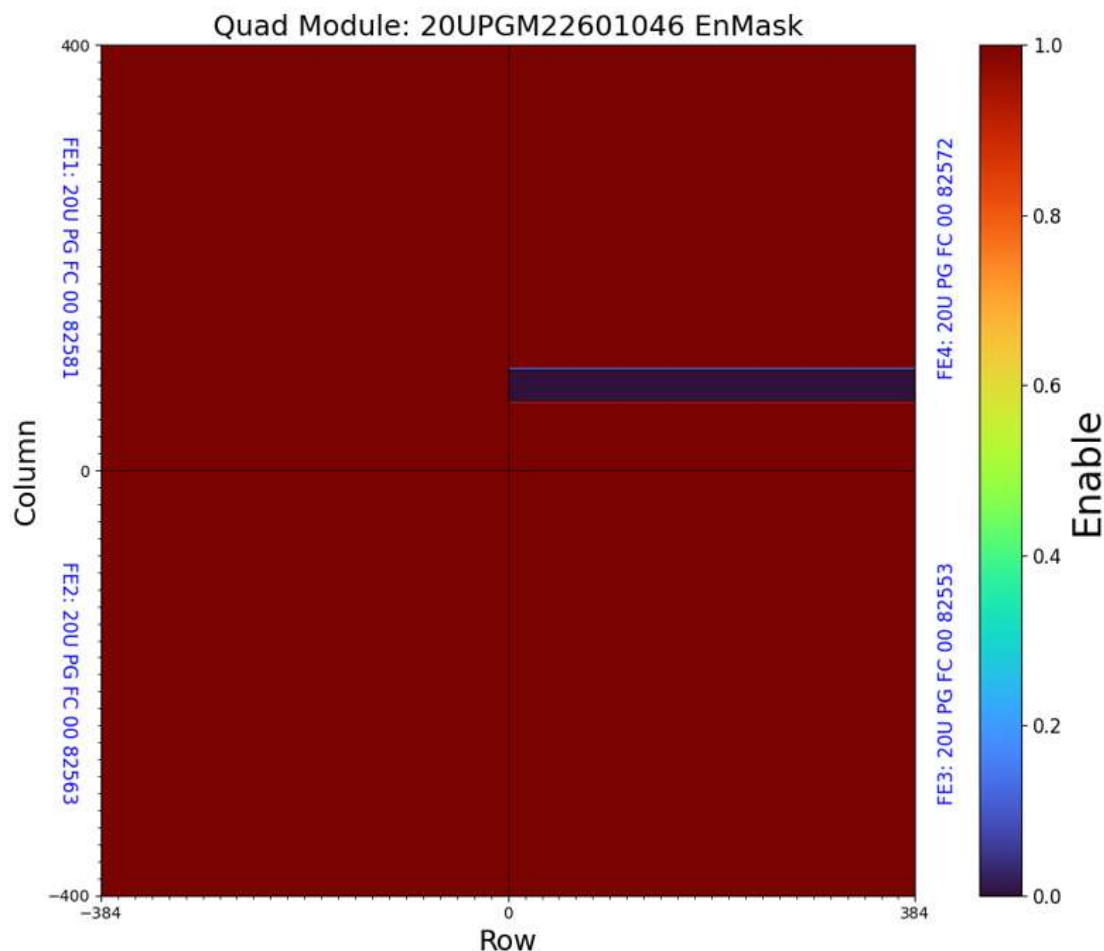


図 4.11: あるコラムを隠して通信が復活した様子

ここを隠すことで他の部分の読み出しが可能になったことから、右上の FE チップに不良ピクセルが存在していることがわかる。実際に私がこの不良ピクセルを探す際は、手作業で二分探索法を用いて不良ピクセルを絞り込んだが、現在の品質試験ではこれらを自動で行うスクリプトが書かれ、品質試験環境は日々改善している。

現時点で不良と判断されたモジュールは 14 個存在するが、それらすべてに対して電圧立ち上げ速度の変更を試してはいないため、それにより救われるモジュールがまだ存在する可能性はある。明らかに傷が原因で、降伏電圧が低い電圧になり不良と判断されるモジュールもあるが、明らかな原因が特定できていない、それ以外の通信不良モジュールの原因究明は、ITk 搭載後にモジュールが悪化する原因を探すことにもつながるため重要度が高い。

第5章 データベースの開発

LocalDB にアップロードするためのアップローダの一つである `module-qc-nonelec-gui` の機能拡張や解析用スクリプトなどの開発を行ってきた。`module-qc-nonelec-gui` が現在サポートしているテストは 3.5 節で説明している。ここからは著者自身が行ったデータベースの改良について記述する。また、開発した解析スクリプトを用いた評価結果についても述べる。

5.1 `module-qc-nonelec-gui` における機能開発

現在使われているアップローダの `module-qc-nonelec-gui` と、前身の `QC-helper` でおこなった開発をまとめる。特に `QC-helper` で開発し、`module-qc-nonelec-gui` に機能として残っていない (LocalDB 上で別の形式で実装した) ものは 5.1.2 節である。

5.1.1 `roof-envelope`

モジュールの両サイドに打たれたワイヤーボンドを物理的な接触や衝撃から守るために取り付けられる装置をワイヤーボンドプロテクションと呼ぶ。実際のワイヤーボンドプロテクションが取り付けられたモジュールを図 5.1 に示す。

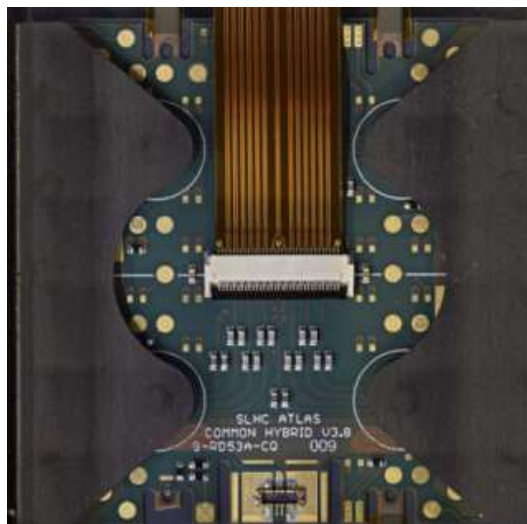


図 5.1: RD53A(ITkPix 生産のために作られたプロトタイプ) に取り付けられたワイヤーボンドプロテクション (写真で左右に見られる構造物)

ワイヤーボンドプロテクションがモジュールからはみ出しすぎていると、輸送する際に邪魔になるため評価基準を持つ必要がある。そのために、このワイヤーボンドプロテクションを装着し

その後、図 2.8 に示した外観測定装置で位置情報を測定する。それらの結果は json 形式で出力し、LocalDB にアップロード、最終的には中央データベースでこれらの情報を保存する。LocalDB にデータをアップロードする方法として図 5.2 で示した GUI を開発した。開発した機能の説明として、初めの画面でテスト結果を含んだ json ファイルを選択し、3.5 節で説明した jsonschema を介することで、ファイルの検証を行っている。想定されたファイルが選択された場合、確認画面に進み、入力が正しいものかどうかを画像を参照しながら確認する。結果が良好な場合、それらは LocalDB にアップロードされる。

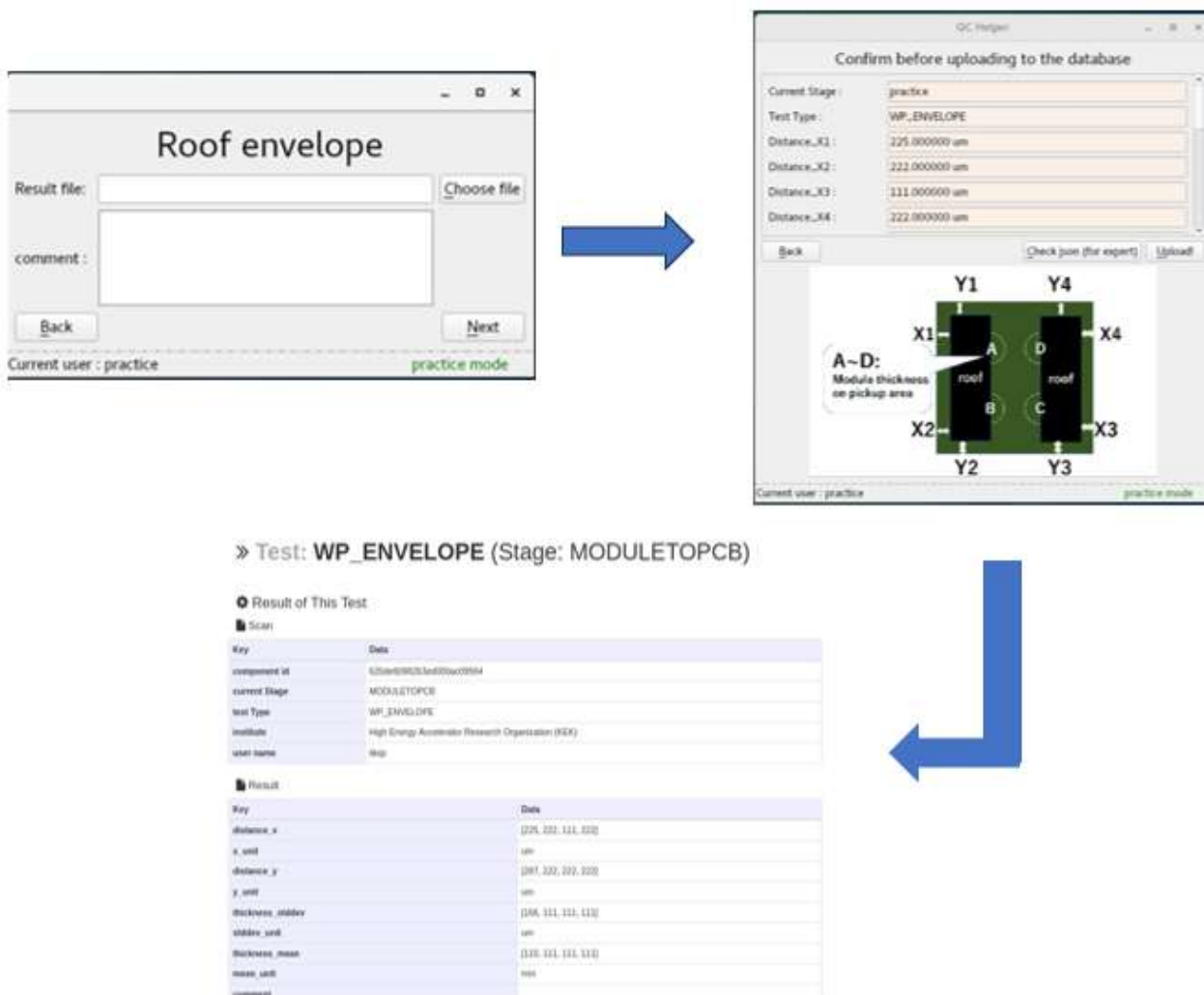


図 5.2: roof envelope

他のテスト GUI と違う点として確認画面における画像表示を取り入れている。実際のモジュールにワイヤーボンダプロテクションが搭載されたものではなく、ポンチ絵で表現している理由は、恒常的に使うモジュールとワイヤーボンダプロテクションが定まっておらず、どのバージョンにおいても対応できるものを採用した。

5.1.2 source scan

4.2 節で説明した X 線照射試験に対応したアップローダを作成した。ハヤシレピックにおける放射線照射試験では X 線を用いて試験を行うが、この線源は通常各機関で独自のものを利用して測定をすることを想定している。(そのため試験の名前が xray scan ではなく source scan)

また試験自体は YARR から QC-helper(module-qc-nonelec-gui) を介さず、直接 LocalDB にアップロードされるが、FE チップごとに試験をアップロードするため、モジュールとして試験を管理するためにはこれらを一つにまとめて QC-helper でアップロードする必要があった。図 5.3 に実際の登録するまでの流れを示したものを表示する。チップごとにスキャンを選択し、測定した時間に対応した RunNumber を選択することで結果が一つのモジュールにまとめられる。これらの RunNumber や時間に関する情報は Choose runID ボタンをクリックすると mongoDB にアクセスし、モジュールのシリアルナンバーから 3.2 に示した childParentRelation を利用してそのモジュールを構成している FE チップの情報を取得する。その FE チップに含まれている YARR で行われた試験名 (xrayscan など) を選択して該当するテストを探して登録するように実装した。

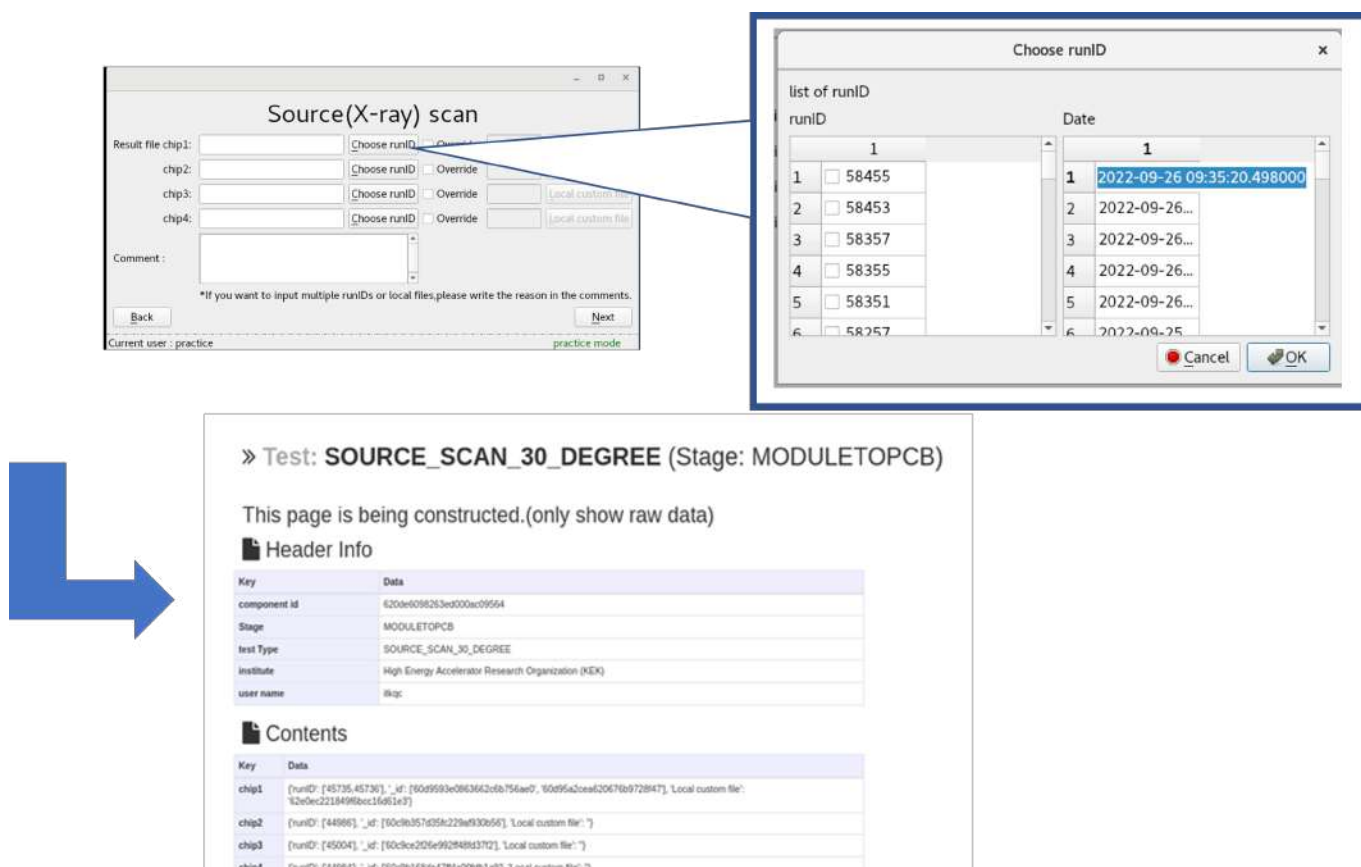


図 5.3: source scan

これらは QC-helper に実装した source scan の登録方法であり、現行の module-qc-nonelec-gui ではこれらの登録方法は廃止された。代わりに localDB ブラウザ上で手動で表 3.2 にある viewer.tag.categories により表示されるタグ (図 5.4) を参考にして試験を 1 つ選択し、登録する方式をとっている。

YARR Scan List

Input keywords Partial match Perfect match

Found 15 scans in LocalDB.


View	Module Name (ATLAS SN)	Chip Name (ATLAS SN)	Scan	User	Site	Date	Stage	Tags 
<input type="button" value="Q View Scan"/>	20U PG M2 26 01067	20U PG FC 00 99706 20U PG FC 00 99686 20U PG FC 00 99675 20U PG FC 00 99668	std_digitalscan	itkqc	hr	2024-01-05 16:26:48 JST(+0900)	MODULE/POST_PARYLENE_COLD	<input type="button" value="TEST"/>
<input type="button" value="Q View Scan"/>	20U PG M2 26 01068	20U PG FC 00 99677 20U PG FC 00 99652	std_thresholdscan_zerobias	admin	hr	2024-01-05 16:15:05 JST(+0900)	MODULE/PARYLENE_UNMASKING	<input type="button" value="PFA"/>

図 5.4: YARR から登録された試験のタグ表示 (赤い枠で囲んだ部分)

5.1.3 Baremodule I-V measure

ハヤシレピックにおけるモジュールの品質試験項目ではなく、KEK で行われるベアモジュールの品質試験である。ベアモジュールとは図 2.4 で示すように、センサーチップと FE チップをバンプ接続したものである。2.4.6 節で示したモジュール時点での I-V 測定との違いは、この試験を localDB や中央データベースに登録する際、試験データを含んだ結果はセンサーチップごとに登録し、ベアモジュールにはその試験のリンクを張る必要がある。

また試験結果が基準を満たしているかどうかの確認はセンサー単体で測定した漏れ電流と、ベアモジュールでの漏れ電流が 2 倍以下という基準がある。これらの要求をみたして、使いやすさを考慮したものを module-qc-nonelec-gui に実装した。

module-qc-nonelec-gui に実装した処理の流れを説明する。図 5.5 が module-qc-nonelec-gui における登録の進め方である。

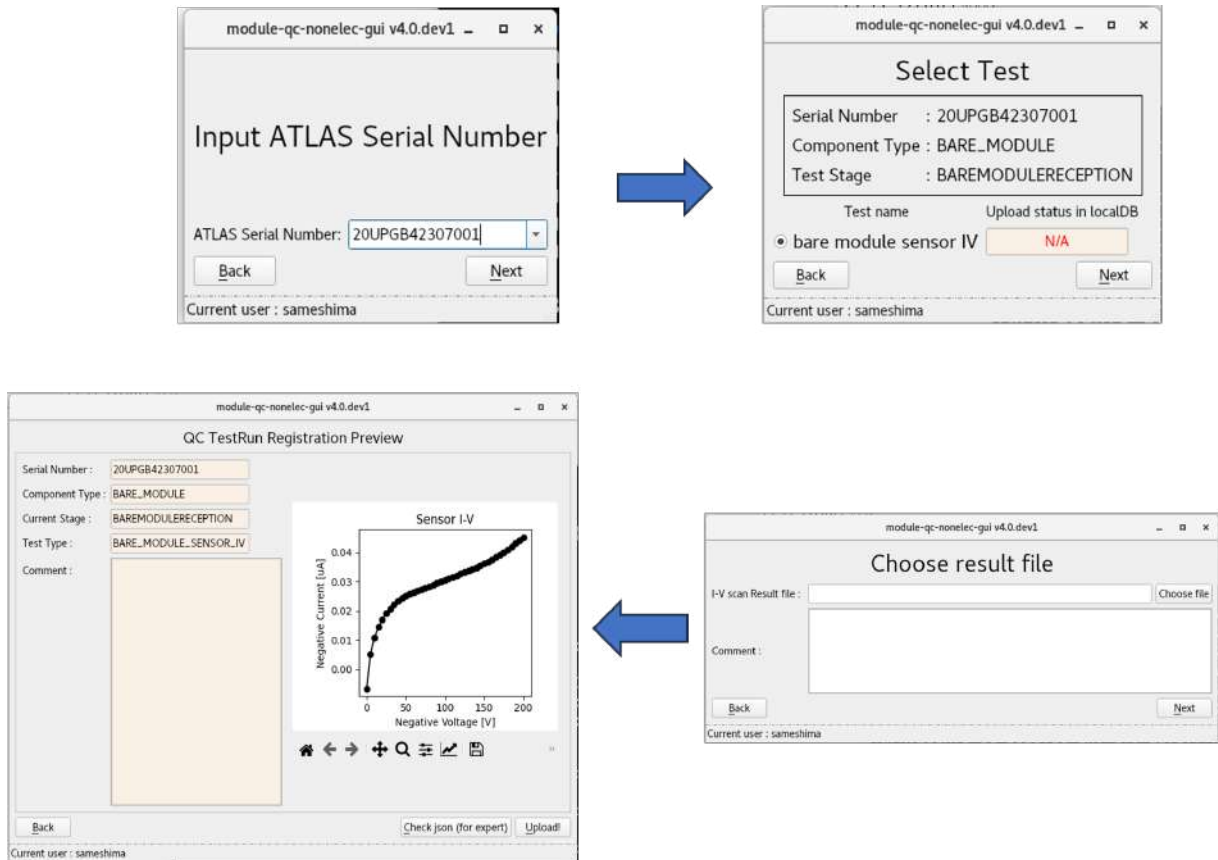


図 5.5: module-qc-nonelec-gui における BAREMODULE I-V 試験の概要

この I-V を測定する段階では、すでにセンサーと FE チップはバンプボン드로接続されたベアモジュールの状態のため、module-qc-nonelec-gui におけるシリアルナンバーの選択は、ベアモジュールのシリアルナンバーを選択する。その後、ファイル選択とプロットの確認をし、upload ボタンを押す。ユーザが担当する部分はこれだけであり、実際にはセンサーチップにテストをアップロードしているが、gui 上ではベアモジュールに記録をアップロードしているように表示している。これは中央データベースの要請に対応するため、BAREMODULE I-V が選択された時点で、内部的に mongoDB に登録されているベアモジュールのシリアルナンバーから、対応しているセンサーのシリアルナンバーを検索し、そのセンサーを対象にテストをアップロードするように変更している。この結果はセンサーに sensor-IV として登録される。localDB へアップロードが完了したことを内部的に確認し、そのセンサーのシリアルナンバーとテストの名前で localDB を検索し、そのテストに対する固有の ObjectID を入手する。この ID を BAREMODULE I-V の試験結果として渡すことで、センサーに登録された試験と紐づけができる。この ID を LocalDB の web ページ上で選択すると、該当ページへジャンプするようにした。詳細は 5.2 節で説明する。

5.2 localDB の改良

module-qc-nonelec-gui を通して localDB に結果をアップロードする際に、ユーザの利便性向上のため localDB を改良する必要があった。以下に改善したポイントを説明する。

5.2.1 BAREMODULE I-V におけるリンクの表示

図 5.5 で localDB にテストをアップロードしたものを、web ページ上で確認すると図 5.6 が表示される。

» Test: **BARE_MODULE_SENSOR_IV** **QC Passed**
(Stage: BAREMODULERECEPTION)

[Download TestRun](#) [Recycle Analysis](#)
(Server's version: 2.2.1.de43)

Basic View Only: custom view for this TestRun is not prepared.

Metadata

Key	Data
QC Passed	True
Date	2023-11-17 00:57:54 JST(+0900)
component Objectid	649290026a25bf00428b190c
Stage	BAREMODULERECEPTION
Test Type	BARE_MODULE_SENSOR_IV
Institute Objectid	
User Name	None
Objectid of this record	65563c040b059f70ecbbdeae
Objectid of RAW record	65563c020b059f70ecbbdead

Results

Key	Data
ANALYSIS_VERSION	1.0.11
LINK_TO_SENSOR_IV_TEST	26JPGS333061E - Passed (mqat v1.0.11)
NON_WORKING_FE_CHIPS	N/A

図 5.6: localDB 上での BAREMODULE I-V の結果

実際に mongoDB に登録されているデータの例を図 5.7 に示す。

id	65563c020b059f70ecbbdead				
raw	serialNumber	testType	results		
	20UPGB42307001	BARE_MODULE_SENSOR_IV	property		
			comment		
			metadata	QCHELPER_VERSION 4.0.dev1	
				MODULE SN 20UPGB42307001	
			Metadata		
			Measurements		
			LINK_TO_SENSOR_IV_TEST	65563c020b059f70ecbbdead	
			NON WORKING FE CHIPS	None	
			rawHash	9047444ae5ca1cd30ee8547497ca86eb	
stage			BAREMODULERECEPTION		

図 5.7: module-qc-nonelec-gui から localDB に渡されたデータ

図 5.7 における LINK_TO_SENSOR_IV_TEST という項目には 5.1.3 節で説明したように、センサーに登録したテストの ObjectID が記載されている。データベースとしてはこの ID があるだけで内部的に参照することができ十分だが、ユーザとしてはこの英数字が羅列しているものでは具体的なセンサーチップやモジュールと対応させることが難しい。そのため表示上はユーザに寄り添う設計にする必要がある。そのため、この LINK_TO_SENSOR_IV_TEST の中の ObjectID をもとに localDB 内を検索し、そのまま ObjectID を表示するのではなく、見た目上はテストの pass/fail の表示と対応するセンサーのシリアルナンバーにし、処理上はその試験のページへのリンクになっている。具体的な処理をリスト 5.1 に示す。

リスト 5.1: ObjectID の表示

```

1 for key,value in qctest['results']['results'].items():
2     if 'LINK_TO_SENSOR_IV' in key and value != None:
3         result = localdb.QC.result.find_one( {'_id':ObjectId(value)} )
4         component = localdb.component.find_one( {'_id':ObjectId( result
5             ['component'] ) } )
6
7         passed = 'Passed' if result['passed'] == True else 'Failed'
8
9         url = url_for('component_api.show_component', id=result['
10             component'], collection='component', test='qctest', runId=
11             value)
12
13         if passed == 'Passed':
14             value = f"<a href=\"{url}\" class=\"badge badge-success
15                 \">>{component['name']} : {passed} (mqat v{ana_version})
16                 </a>"
17
18         else:
19             value = f"<a href=\"{url}\" class=\"badge badge-dark\">{
20                 component['name']} : {passed} (mqat v{ana_version})</a
21                 >"
22
23         qctest['results']['results'][key] = value

```

5.2.2 BAREMODULE I-V と SENSOR I-V の比較

5.1.3 節で示したように、BAREMODULE I-V には QC の基準が定められており、その判断を行う必要がある。実際にはベアモジュールだけではなく、モジュールとして組みあがった時点での IV 測定においても前段階の I-V と比較する必要があり、共通の比較用フォーマットが求められる。QC 判定をおこなうスクリプトは localDB 内部ではなく、別の場所に解析スクリプトがまとめられて管理 (5.3 節で説明) している。そのスクリプトからは直接 localDB にアクセスできないため、解析したいファイルと過去の測定結果の両方を解析スクリプトに渡す必要がある。そのため localDB 内部で I-V 測定に関する試験が選ばれた場合、それより下 (モジュールが選択されている場合はベアモジュールとセンサー、ベアモジュールの場合はセンサー) の試験結果で比較したい部分のみを抽出した json ファイルフォーマットを用意した。それをリスト 5.2 に示す。

リスト 5.2: 過去データの参照

```
1 previous_data = {
2   "target_component" : component["serialNumber"],
3   "target_stage" : test[0]["stage"],
4   "reference_IVs" : [
5     { 'component' : SN_SEN,
6       'stage' : "sensor_manufacture",
7       'Vbd' : SENSOR_IV_TEST["results"]["BREAKDOWN_VOLTAGE"],
8       'Vfd' : SENSOR_CV_TEST["results"]["V_FULLDEPL"],
9       'Ilc' : SENSOR_IV_TEST["results"]["LEAK_CURRENT"],
10      'temperature' : SENSOR_IV_TEST["results"]["properties"]["TEMP"] if
          SENSOR_IV_TEST["results"].get("properties") else SENSOR_IV_TEST["results"]
          ["property"]["TEMP"],
11      'IV_curve' : {
12        "voltage" : SENSOR_IV_TEST["results"]["IV_ARRAY"]["voltage"],
13        "current" : SENSOR_IV_TEST["results"]["IV_ARRAY"]["current"]},
14      'qc_passed': SENSOR_IV_TEST["passed"]
15    }
16  ]
17 }
```

これはセンサーの試験結果のみを previous_data に代入し、もし対象がベアモジュールならばその結果もこの previous_data に代入するように処理してある。まとめたデータを json 形式のファイルで保存し、管理場所の異なるスクリプト間におけるデータ共有を可能にしてある。またデータ量の増加を防ぐため、処理が正常に終了した場合はこの json ファイルを削除するようにしてある。

5.3 解析用スクリプトの開発

5.2.2 節で説明した解析スクリプトについて詳しく説明し、加えた変更などについて解説する。まず解析スクリプトの役割について説明したものが図 5.8 である。

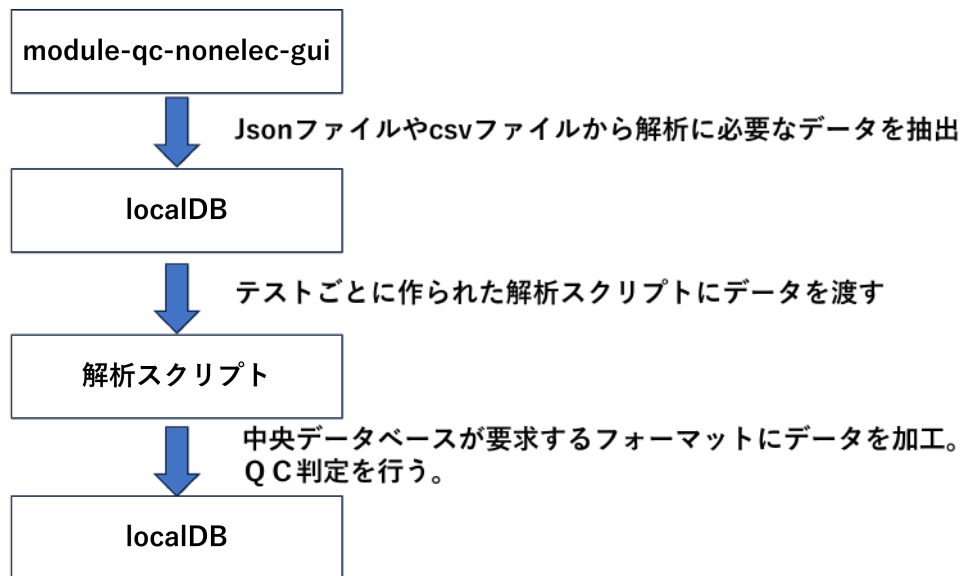


図 5.8: module-qc-nonelec-gui から localDB にデータを渡す際のイメージ図

この解析スクリプトは python を用いて開発されており、主な役割は localDB や中央データベースでデータ閲覧する際に表示される最終的なデータの整形である。加えてそれぞれの品質試験に要求される品質管理の条件を満たしているかどうかの判断を行う場所でもある。その中で新しく開発を行った部分を紹介する。

5.3.1 BAREMODULE I-V における解析スクリプトの開発

実際にベアモジュールの I-V を測定したファイルをリスト 5.3 に示す。

リスト 5.3: ベアモジュール I-V 測定に使われる dat ファイル

```

1 [Data Format]
2 voltage[V] current[A] capacitance[F] D temp[C] time(dd-MM-yyyy HH:mm:ss)
3 0 3.06087e-08 4.0652e-09 0.16373 20 13-07-2023 14:59:02
4 -5 -2.93135e-08 1.73676e-09 0.27737 20 13-07-2023 14:59:10
5 -10 -3.27581e-08 1.4343e-09 0.26113 20 13-07-2023 14:59:18
6 -15 -3.47336e-08 1.25569e-09 0.25459 20 13-07-2023 14:59:26
7 -20 -3.6278e-08 1.15054e-09 0.24623 20 13-07-2023 14:59:34
8 -25 -3.74792e-08 1.06741e-09 0.2346 20 13-07-2023 14:59:42
9 -30 -3.85172e-08 1.00403e-09 0.23528 20 13-07-2023 14:59:50
10 -35 -3.9371e-08 9.43035e-10 0.22902 20 13-07-2023 14:59:58
11 -40 -3.99988e-08 8.98915e-10 0.23146 20 13-07-2023 15:00:06
12 -45 -4.06141e-08 8.59254e-10 0.22043 20 13-07-2023 15:00:14
13 -50 -4.11373e-08 8.23379e-10 0.22518 20 13-07-2023 15:00:22
14 -55 -4.16604e-08 7.93501e-10 0.2162 20 13-07-2023 15:00:30
15 -60 -4.23678e-08 7.76113e-10 0.22301 20 13-07-2023 15:00:38
16 -65 -4.72479e-08 7.68361e-10 0.21941 20 13-07-2023 15:00:46
17 -70 -6.25582e-08 7.60217e-10 0.22545 20 13-07-2023 15:00:54
18 -75 -9.70668e-08 7.57944e-10 0.224 20 13-07-2023 15:01:02
  
```

19	-80	-1.61468e-07	7.57229e-10	0.22847	20	13-07-2023	15:01:10
20	-85	-2.61453e-07	7.60958e-10	0.22984	20	13-07-2023	15:01:18
21	-90	-3.96968e-07	7.61084e-10	0.22261	20	13-07-2023	15:01:26
22	-95	-5.61827e-07	7.63085e-10	0.22376	20	13-07-2023	15:01:34
23	-100	-7.38853e-07	7.62294e-10	0.22504	20	13-07-2023	15:01:42
24	-105	-9.31857e-07	7.6046e-10	0.23082	20	13-07-2023	15:01:50
25	-110	-1.14065e-06	7.6336e-10	0.22314	20	13-07-2023	15:01:58
26	-115	-1.34501e-06	7.59345e-10	0.22401	20	13-07-2023	15:02:06
27	-120	-1.55514e-06	7.63138e-10	0.22548	20	13-07-2023	15:02:15
28	-125	-1.77797e-06	7.59455e-10	0.22648	20	13-07-2023	15:02:22
29	-130	-1.99203e-06	7.61276e-10	0.22718	20	13-07-2023	15:02:30
30	-135	-2.19872e-06	7.57609e-10	0.22295	20	13-07-2023	15:02:39
31	-140	-2.44037e-06	7.5878e-10	0.2293	20	13-07-2023	15:02:47
32	-145	-2.65401e-06	7.59246e-10	0.22364	20	13-07-2023	15:02:55
33	-150	-2.87201e-06	7.63141e-10	0.22765	20	13-07-2023	15:03:03
34	-155	-3.10437e-06	7.62097e-10	0.22962	20	13-07-2023	15:03:10
35	-160	-3.32618e-06	7.57172e-10	0.22949	20	13-07-2023	15:03:18
36	-165	-3.58104e-06	7.65428e-10	0.22722	20	13-07-2023	15:03:26
37	-170	-3.83519e-06	7.6263e-10	0.23368	20	13-07-2023	15:03:34
38	-175	-4.06715e-06	7.65923e-10	0.22652	20	13-07-2023	15:03:42
39	-180	-4.30703e-06	7.67216e-10	0.23281	20	13-07-2023	15:03:50
40	-185	-4.58877e-06	7.66542e-10	0.23569	20	13-07-2023	15:03:58
41	-190	-4.83735e-06	7.69748e-10	0.23502	20	13-07-2023	15:04:06
42	-195	-5.09159e-06	7.70859e-10	0.23586	20	13-07-2023	15:04:14
43	-200	-5.37835e-06	7.66778e-10	0.24069	20	13-07-2023	15:04:30

この結果を解析スクリプトにより降伏電圧の有無の確認、漏れ電流の確認、センサーチップ単体で測定した I-V との比較を行い、QC 基準を満たしているかどうかを判断する。

5.2.2 節で説明した比較用のファイル (リスト 5.2) を用いてセンサーチップ単体で測定した I-V とを比較し、基準を満たしているかどうかをチェックする。漏れ電流は温度依存性が大きいいため、式 2.2 を用いて温度の補正を行う。センサーの I-V 測定は浜松ホトニクスで行われ、およそ 25 °C で測定しているとされている。ベアモジュールの I-V 測定は KEK で実施され、20 °C に管理して行われている。

センサー単体で測定した漏れ電流とベアモジュール段階での漏れ電流を解析スクリプト内で作成し、プロットしたものを比較のためにグラフを重ねて表示している。実際に localDB 上で表示しているグラフを図 5.9 に示す。

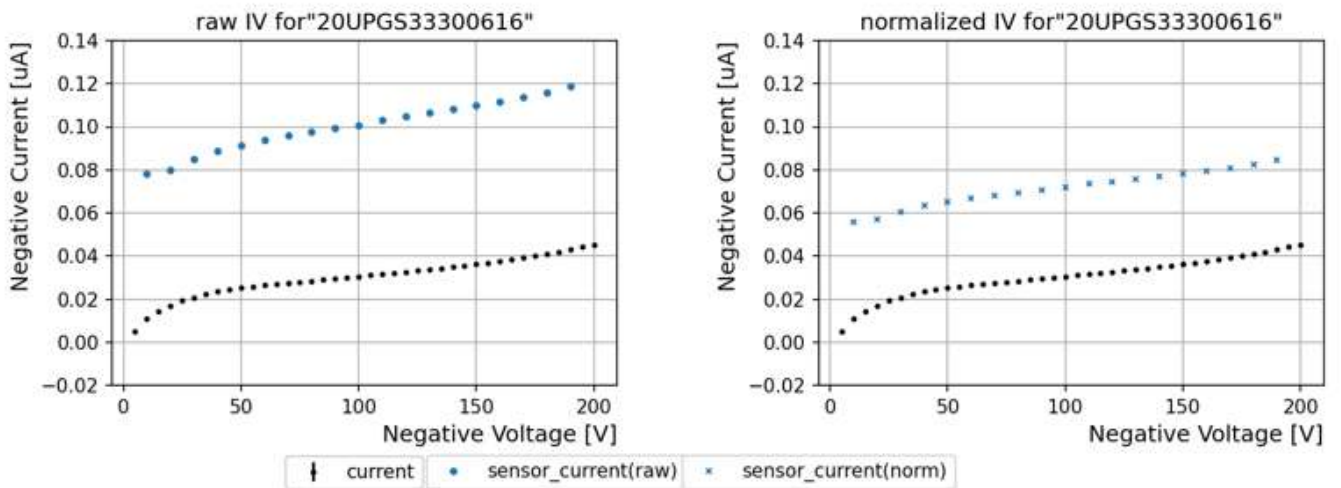


図 5.9: センサー単体での漏れ電流とベアモジュール段階での漏れ電流を比較したプロット

このプロットからセンサーとベアモジュールの降伏電圧や漏れ電流の変化を確認する。降伏電圧の判定は、印加電圧を 5V ごとに増加させながら測定している漏れ電流の測定から、ひとつ前の漏れ電流より 1.2 倍の大きさを超えた場合に降伏電圧と判定される。降伏電圧が発見された場合、ユーザが視覚的に判断しやすいよう赤い線が引かれるようになっている。これらの降伏電圧の表示方法はモジュールの I-V 測定 (図 5.10) におけるコードを利用している。

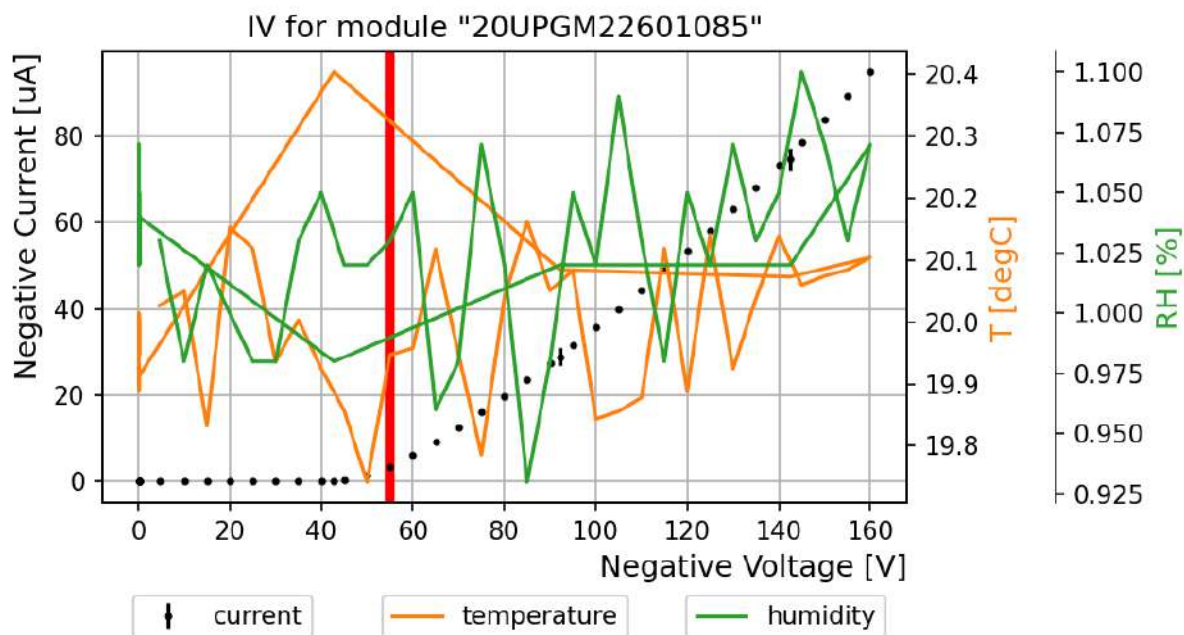


図 5.10: 降伏電圧の表示 (モジュールでの I-V 測定。モジュールの I-V 測定は温度、湿度を正確に測定しているためそれらもプロットに含めている)

また図 5.9 の左右のプロットを比較すると、温度補正後においてもセンサーチップ単体の漏れ電流がベアモジュールの漏れ電流より大きいことがわかる。温度を 25℃～30℃の範囲で設定した場合の比較をまとめた図を図 5.11 に示す。

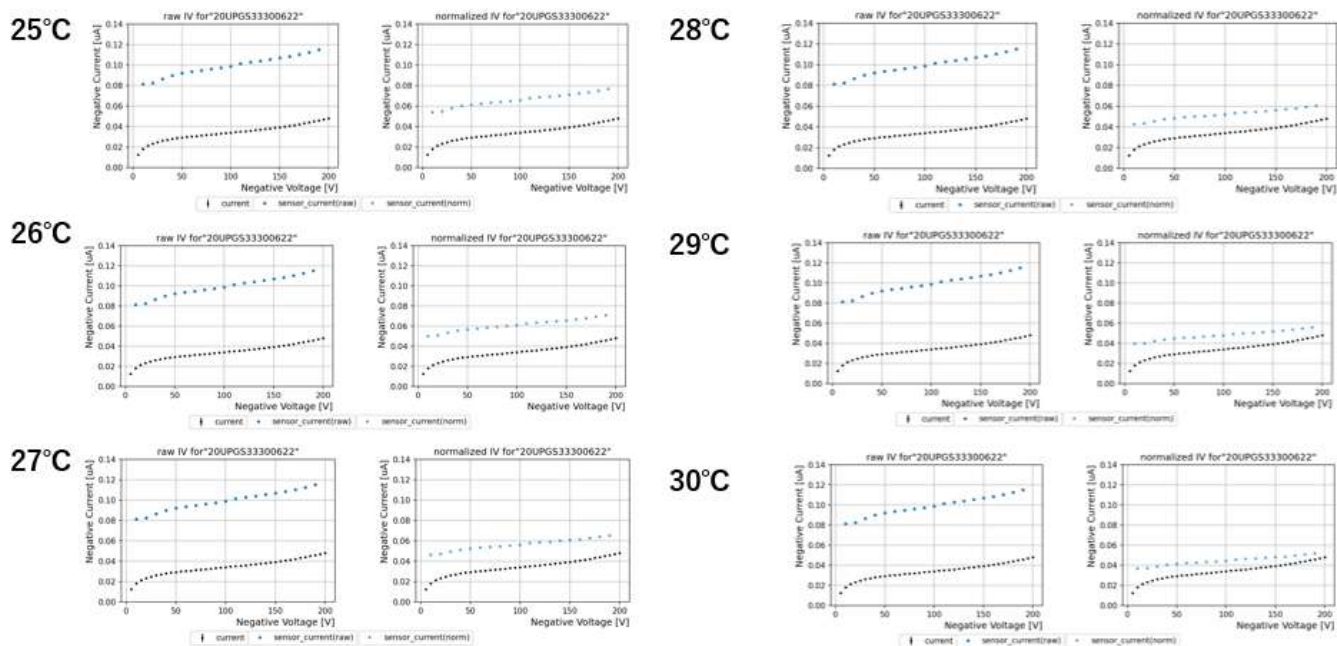


図 5.11: 温度設定によるセンサーチップ単体とベアモジュールの漏れ電流の比較

この結果から温度を 30℃まで設定した場合でもセンサーチップ単体の漏れ電流がベアモジュールの漏れ電流を下回ることが確認されなかった。次にセンサーチップ単体で測定した漏れ電流にオフセットが乗っている可能性について考察する。センサーチップの測定に $0.04\mu A \sim 0.06\mu A$ のオフセットが乗っていると仮定し比較を行った。 $0.04\mu A$ のオフセットを仮定した結果を図 5.12 に示す。 $0.05\mu A$ の場合を図 5.13 に示す。 $0.06\mu A$ の場合を図 5.14 に示す。

0.04 μ A

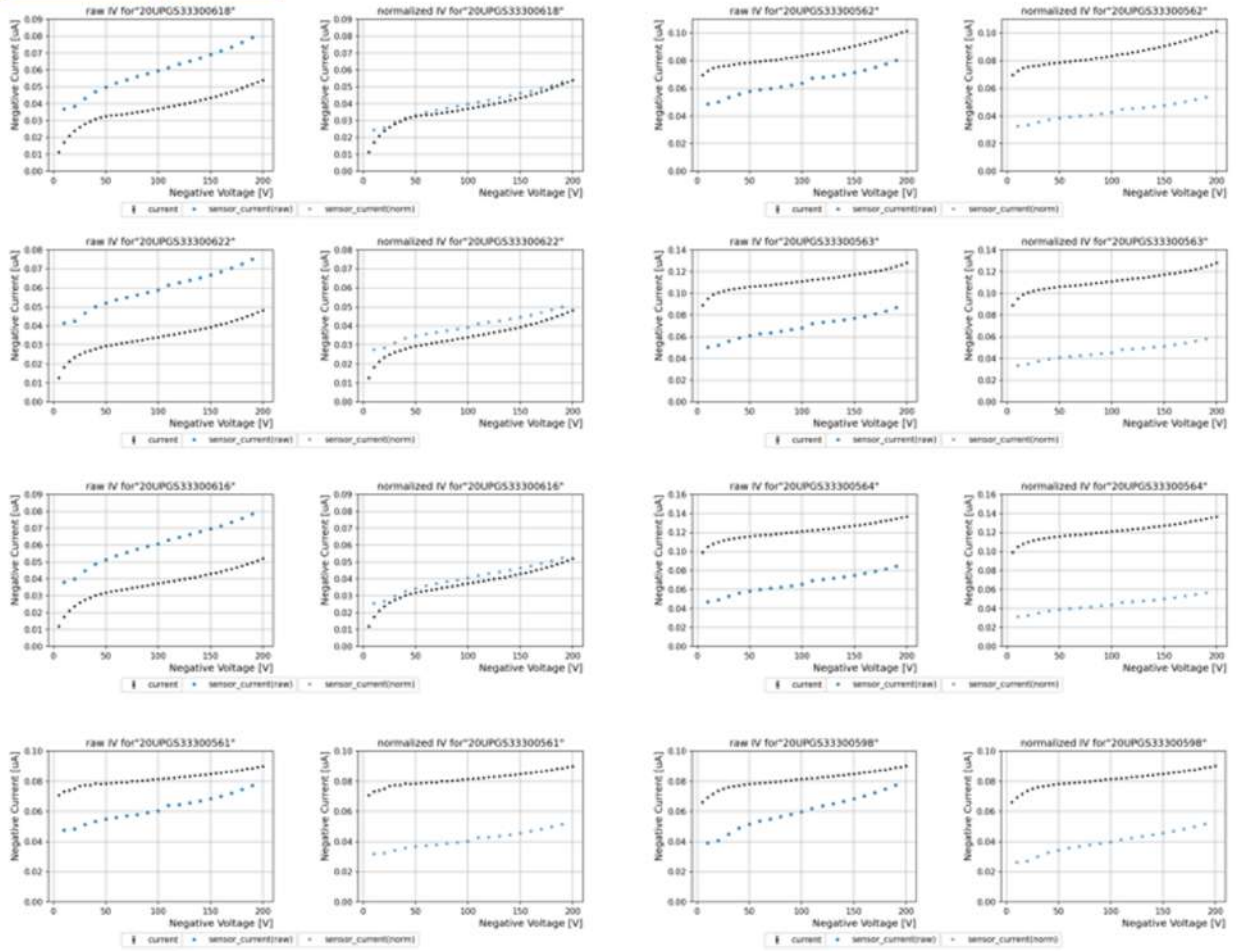


図 5.12: センサー単体の漏れ電流の測定に $0.04\mu A$ のオフセットが乗っていると仮定した場合の温度補正前後の比較結果

0.05 μA

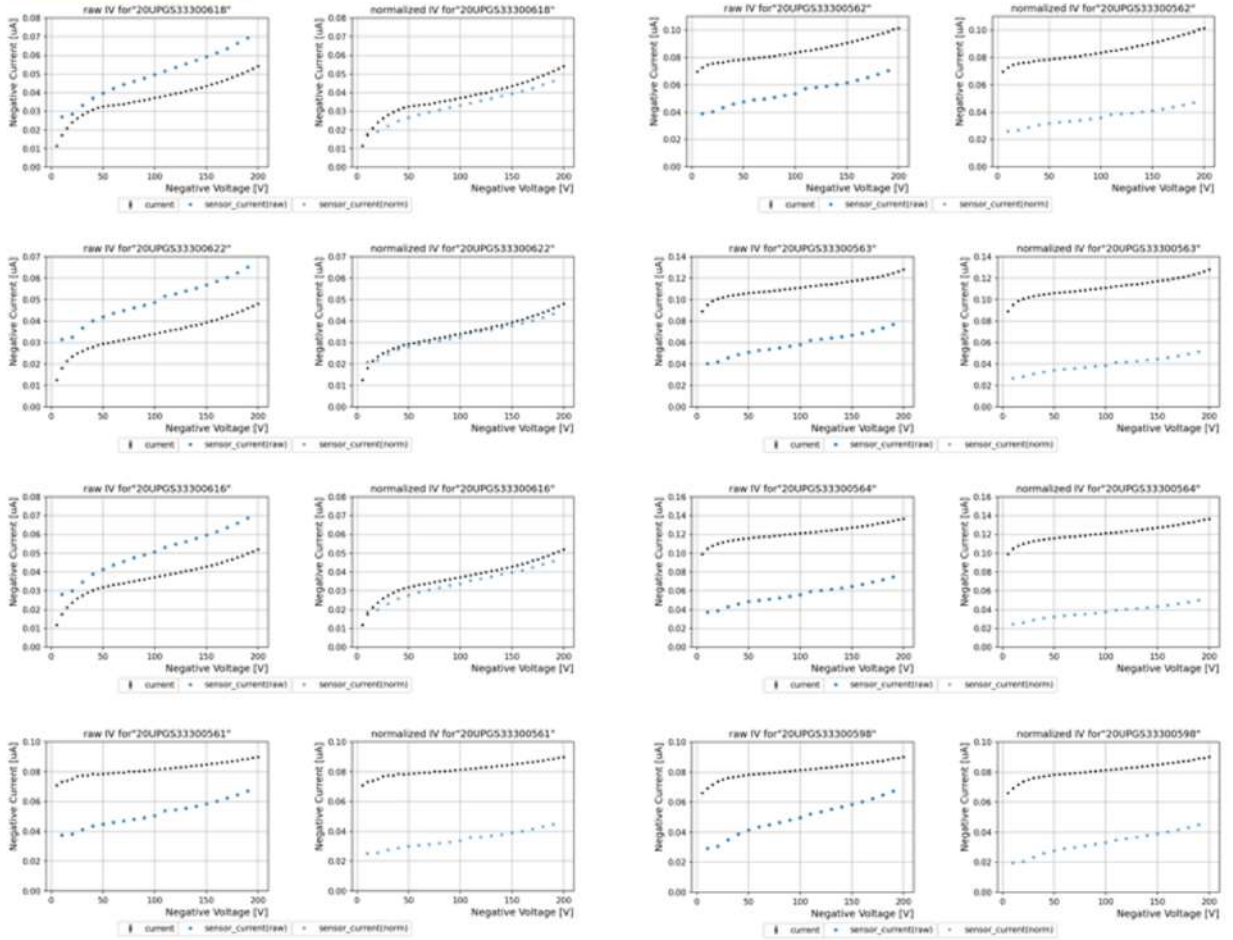


図 5.13: センサー単体の漏れ電流の測定に $0.05\mu A$ のオフセットが乗っていると仮定した場合の温度補正前後の比較結果

0.06 μ A

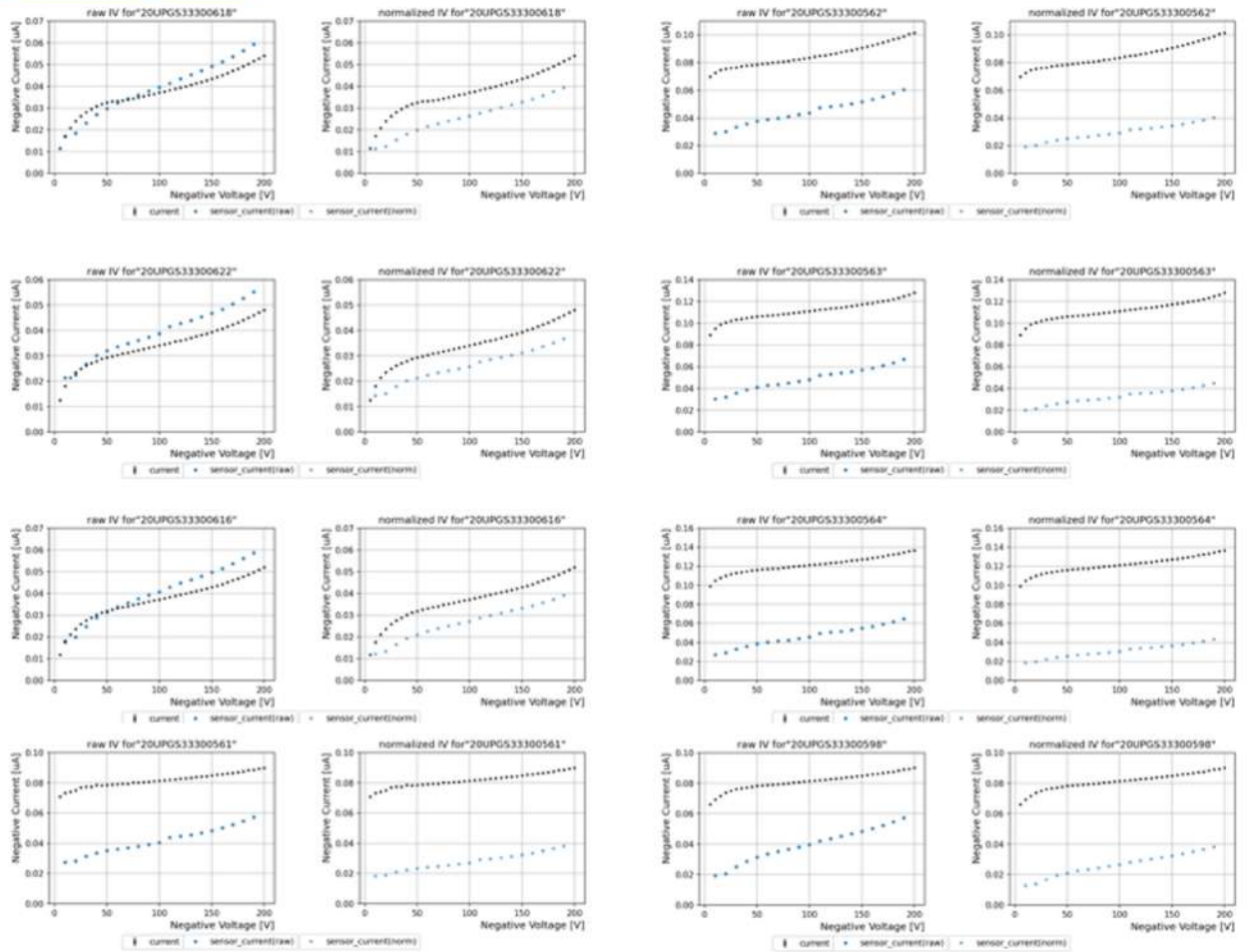


図 5.14: センサー単体の漏れ電流の測定に $0.06\mu A$ のオフセットが乗っていると仮定した場合の温度補正前後の比較結果

以上の結果より、 $0.06\mu A$ のオフセットが乗っていると仮定した場合、試験結果が矛盾しないことが分かった。

5.3.2 ワイヤ強度測定の解析スクリプトの開発と評価

2.4.5 節で説明したワイヤ強度測定試験の解析スクリプトを開発した。この試験の localDB へのアップロード方法は試験結果が保存された csv ファイルを選択し、module-qc-nonelec-gui を通して LocalDB へアップロードする。

csv ファイルはワイヤの壊れ方とその時にかけた負荷の大きさをまとめたものになっている。実際に測定し、出力されたデータの例をリスト 5.4 にまとめる。

リスト 5.4: ワイヤ強度測定に使われる csv ファイル

1 8.734,Midsparn break
2 8.785,Midsparn break
3 8.841,Midsparn break
4 8.455,Heel break on chip
5 8.806,Midsparn break
6 8.798,Heel break on chip
7 8.462,Heel break on chip
8 8.841,Midsparn break
9 7.462,Heel break on chip
10 8.893,Midsparn break
11 8.811,Heel break on chip
12 8.772,Midsparn break
13 8.854,Heel break on chip
14 8.24,Heel break on chip
15 8.069,Heel break on chip
16 8.498,Heel break on chip
17 8.633,Heel break on chip
18 8.583,Heel break on chip
19 8.582,Heel break on hybrid
20 8.409,Heel break on chip
21 8.274,Heel break on chip
22 8.606,Midsparn break
23 7.784,Heel break on chip

このデータは各ワイヤ-が切れた時の状況と張力(グラム重)が保存されている。
module-qc-nonelec-gui で、csv ファイルを選択し、localDB へのアップロードを選択する。実際の画面を図 5.15 に示す。

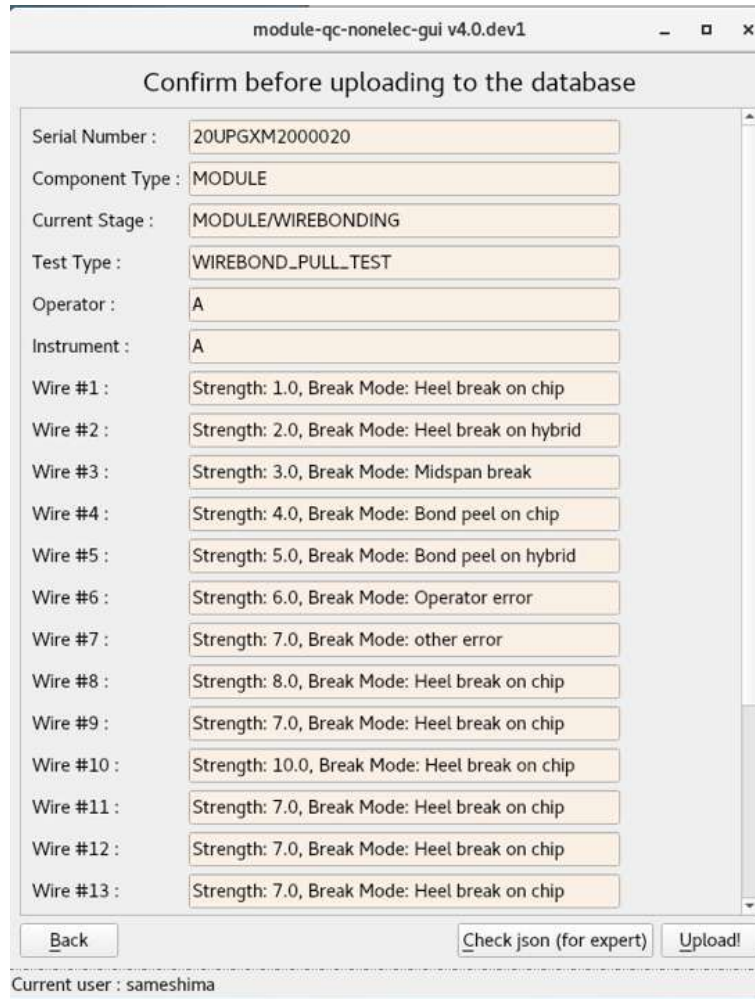


図 5.15: module-qc-nonelec-gui におけるワイヤー強度測定確認画面

アップロードボタンを押すと、これらのデータが localDB に送られ解析スクリプトを通すことで、中央データベースが要請するフォーマットに整形する。解析する内容は、5グラム重以下で壊れたワイヤーがあるかどうかをチェックすることや、平均値や最大値最小値、また壊れ方の割合を計算する必要がある。

現在の課題としてこの試験における QC 判定の基準が定まっておらず、評価が難しいことがあげられる。強度測定におけるワイヤーの壊れ方は大きく分けて 4 種類存在し、midspan break, heel break on hybrid, heel break on chip, bond peel である。それぞれの説明を 5.1 に示す。

表 5.1: ワイヤーの壊れ方

midspan break	ワイヤー中間で切れた場合
heel break on hybrid	フレキシブル基板側のボールボンド部分につながるワイヤーが切れた場合
heel break on chip	FE チップ側のボールボンド部分につながるワイヤーが切れた場合
bond peel break	ボールボンド部分が剥離した場合

これらの壊れ方について、midspan break,heel break は問題のない壊れであり、bond peel break は、通常ワイヤーボンダーのパラメータが不適切である可能性や十分な固定ができていない場合に生じることが多く、問題のある壊れ方とされている。現在のプリプロダクションで行われたワイヤー強度測定をもとに、localDB にアップロードしたデータを用いて壊れ方別のヒストグラムを作成した。

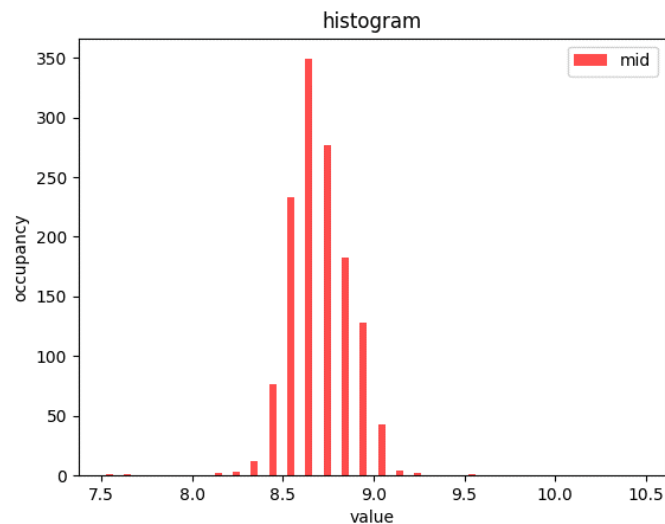


図 5.16: ワイヤー強度測定における midspan break の強度 (横軸、単位：グラム重) と頻度 (縦軸) のヒストグラム

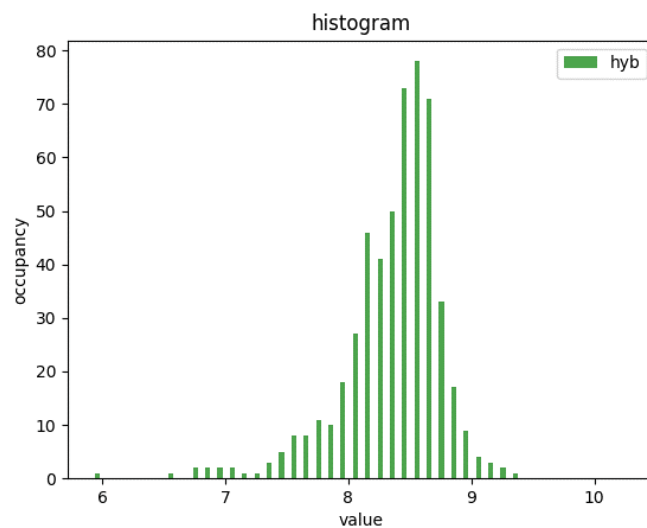


図 5.17: ワイヤー強度測定における heel break on hybrid の強度 (横軸、単位：グラム重) と頻度 (縦軸) のヒストグラム

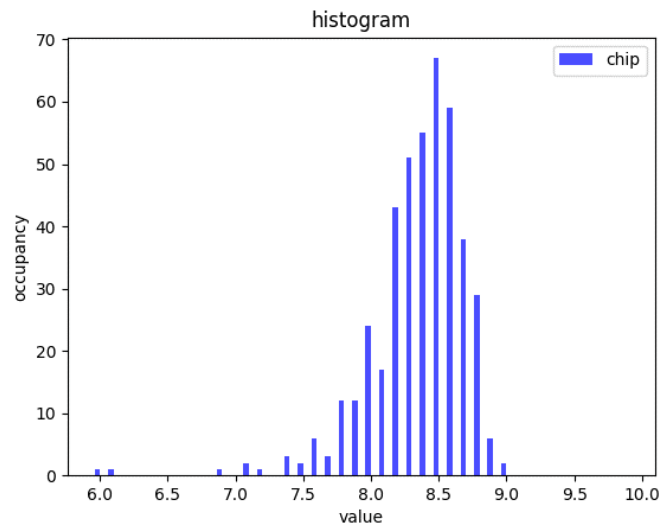


図 5.18: ワイヤー強度測定における heel break on chip の強度 (横軸、単位 : グラム重) と頻度 (縦軸) のヒストグラム

Bond peel 型の壊れ方は現状確認されていない。

これらの統計から判断できることとして、midspan break が壊れ方の多くを占めており、強度のばらつきもおおよそガウス型の分布となり目立った異常が確認されていない。bond peel 型の壊れ方は一つも確認されないことから、現在使われているワイヤーボンダーは非常に良好に機能していると考えられる。これらのプロットを参考に、QC 評価基準を決定し、解析スクリプトに取り入れていく。

第6章 まとめ

2026年からインストールが予定されている HL-LHC にむけて、ATLAS 検出器は内部飛跡検出器のアップグレードが予定されている。内部飛跡検出器は ITk に置き換えられることになり、ITk を構成する約 10000 個のモジュールについて量産することになった。モジュール量産における各組み立て工程において品質試験を実施し、それぞれの品質試験データは、各機関で行った試験全てをまとめる localDB と、全体の結果をまとめる中央データベースを通して管理される。本論文執筆時は、量産に向けた準備段階であるプリプロダクションの時期であり、現在の生産拠点であるハヤシレピックにおける品質試験状況や LocalDB 関連のアップデートを本論文でまとめた。

localDB に品質試験結果をアップロードするアップローダは YARR と module-qc-nonelec-gui の 2 つがあり、YARR は読み出し試験結果は localDB にアップロードし、module-qc-nonelec-gui は YARR で扱わないものを担当する。特に module-qc-nonelec-gui に関連するアップデートをメインに行った。module-qc-nonelec-gui に新しく機能を追加した試験として roof-envelope, source scan, Baremodule-IV, sensor-IV であり、加えて中央データベースへデータを上げるための解析スクリプトも作成した。これらはプリプロダクション及び本量産時に必要なプロセスのうちの一つであり、重要な役割を果たすものだといえる。とくに解析スクリプトにおける QC 判定は、IT k に搭載するモジュール選定の重要な判断材料となる。

現状と課題

module-qc-nonelec-gui が担当するモジュールの品質試験項目は本アップデートをもって全て対応した。しかしフレキシブル基板やセンサー、ベアモジュールにおける品質試験はまだ十分に対応しきれていない。それらのアップデートは今後の重要な課題となる。また解析スクリプトにおける QC 判定基準も十分に定まっていない、もしくは条件が厳しすぎるという課題がある。本論文執筆時におけるプリプロダクションでは約 100 台の生産を行っており、統計という点で考えると十分な量ではないがおおよその傾向を把握することができる。これらを用いて新たな判定基準を考えることも重要な課題の一つである。

付録A BAREMODULE_I-Vアップロードのためのサブツール

現状ベアモジュールのI-Vを測定したデータはlocalDB及び中央データベースに全てをあげられておらず、私が手動でいくつかアップロードしている。このデータをアップロードするにあたって注意するポイントがいくつかある。

一つはセンサーチップ単体で測定したIVが中央データベースに上がっているかどうかを確認することである。これはベアモジュールのI-V試験におけるQC判定にセンサー単体でのI-V測定を必要としているからである。中央データベースにおいて一つずつ確認すると判断に大量の時間がかかるため、簡単な検索ができるスクリプトを開発したためリストA.1に提示する。

リスト A.1: 中央データベースにセンサーチップ単体で測定したテスト判別ツール

```
1 array = []
2 array2 = []
3 array3 = []
4 with Path("PD.csv").open() as f:
5     reader = csv.reader(f)
6     for index,row in enumerate(reader):
7         try:
8             serialNumber = row[4]
9             if serialNumber == "":
10                continue
11            if serialNumber == "ATLAS SN (Bare)":
12                continue
13            #print(serialNumber)
14
15            receive = dbtools.getCompStage(pd_client,serialNumber)
16            child_data = dbtools.get_childComponent(pd_client,receive[1])
17
18            for i in child_data:
19                if i.get('code') == 'SENSOR_TILE':
20                    sensor_stage = i.get('stage')
21                    sensor_SN = i.get('serialNumber')
22                    break
23
24            sencomps=pd_client.get('getComponent',json={'component' : sensor_SN
25                })
26            for i in sencomps['tests']:
27                if i.get("code") == "IV_MEASURE":
28                    for j in i.get("testRuns"):
29                        if j.get("institution").get("code") == "HPK":
30                            array.append(serialNumber)
31                        elif j.get("institution").get("code") == "KEK":
```

```

31         array2.append(serialNumber)
32         elif j.get("institution").get("code") == "CERN":
33             array3.append(serialNumber)
34             print(serialNumber + "checked")
35     except:
36         print("unexpected occur")
37 print("uploaded by HPK")
38 print(array)
39 print("uploaded by KEK")
40 print(array2)
41 print("uploaded by CERN")
42 print(array3)
43 print("done")

```

ここに出てくる PD.csv は現時点で組み立てられているベアモジュールやモジュールのシリアルナンバーなどをまとめたデータである。

加えて BAREMODULE_I-V をベアモジュールに登録するためにはベアモジュールのステージを BAREMODULE RECEPTION というステージにする必要がある。この変更をする際、センサーチップのステージも BAREMODULE RECEPTION に合わせる必要がある。ステージをそれらにあわしたものを中央データベースからダウンロードするために、指定したモジュールのステージを変更するためのスクリプトと元のステージに戻すスクリプトを開発した。それぞれを A.2 と A.3 に示す。

リスト A.2: 中央データベースにあるベアモジュールのステージを変えるスクリプト

```

1 import itkdb
2 import json
3 import getpass
4 import sys
5 import os
6 import dbtools
7
8 code1 = getpass.getpass(prompt='access_code1: ')
9 code2 = getpass.getpass(prompt='access_code2: ')
10
11 pd_user = itkdb.core.User(access_code1=code1, access_code2=code2)
12 pd_client = itkdb.Client( user = pd_user, use_eos = True )
13
14 print("input baremodule SN")
15 serialNumber=input()
16
17 receive = dbtools.getCompStage(pd_client,serialNumber)
18 compStage = receive[0]
19 bare_data = {'serialNumber' : serialNumber,
20 'stage' : compStage,
21 'code' : 'BAREMODULE'
22 }
23
24 child_data = dbtools.get_childComponent(pd_client,receive[1])
25 child_data.append(bare_data)
26

```

```

27 uploadStage = "BAREMODULERECEPTION"
28
29 data = {"pre_stage": child_data,
30         "working_stage": uploadStage}
31
32 for i in child_data:
33     if i.get('code') == 'SENSOR_TILE':
34         sensor_stage = i.get('stage')
35         sensor_SN = i.get('serialNumber')
36         break
37
38 try:
39     os.mkdir(f"./{serialNumber}")
40     with open(f'./{serialNumber}/stage.json', 'w') as json_file:
41         json.dump(data, json_file, indent=4)
42 except(FileExistsError):
43     print("ERROR!\nBefore you change to BAREMODULERECEPTION, you need to get
44           back to the original stage.")
45     sys.exit()
46
47 for i in child_data:
48     dbtools.changestage(pd_client,i.get('stage'),uploadStage,i.get('
49         serialNumber'))
50
51 print("done")

```

リスト A.3: リスト A.2 で変更したステージを元のステージに戻すスクリプト

```

1 import itkdb
2 import json
3 import getpass
4 import shutil
5 import dbtools
6
7 code1 = getpass.getpass(prompt='access_code1: ')
8 code2 = getpass.getpass(prompt='access_code2: ')
9
10 pd_user = itkdb.core.User(access_code1=code1, access_code2=code2)
11 pd_client = itkdb.Client( user = pd_user, use_eos = True )
12
13
14 print('input baremodule SN')
15 serialNumber = input()
16
17 with open(f'{serialNumber}/stage.json','r') as file:
18     stage_data = json.load(file)
19
20 for i in stage_data['pre_stage']:
21     dbtools.changestage(pd_client,stage_data.get('working_stage'),i.get('stage
22         '),i.get('serialNumber'))
23
24 shutil.rmtree(f'{serialNumber}')

```

24 `print('done')`

ここで紹介したスクリプトはすべて python を用いて作成したものである。詳細なコードや環境は https://gitlab.cern.ch/kojin/itkpdtools/-/tree/devel_sameshima?ref_type=heads に残してある。

謝辞

本研究を進めるにあたり、多くの方々にお世話になりました。初めに角野秀一先生には研究に関するアドバイスや日常生活においても大変お世話になりました。ほかにもゼミ合宿などを考える際にも大変お世話になりました。指導教員である汲田哲郎先生には出張に関することや何回にもわたる論文指導、日々の生活面においても大変お世話になりました。ATLAS 関係の手続きや関係者への連絡など、自分の研究がスムーズに進められるよう動いてくださり大変ありがとうございました。

また ATLAS の DB 関連の開発をするにあたり、生出秀行さんには大変お世話になりました。ハヤシレピック作業における丁寧なご指導や、設備の説明や関連する物理現象の解説など大変勉強になることが多くありました。特に開発における質問を何度も丁寧に回答して下さりありがとうございました。また留目和輝さんにも大変お世話になりました。東工大に訪問するたび技術的なことから生活面までたくさんお世話になりました。Skype での連絡も度々して下さりとても心強かったです。ありがとうございました。

そして、本研究を進めるにあたって研究室の皆さんにも大変お世話になりました。先輩の在原拓司さんには異なる研究テーマであるにもかかわらず、丁寧に解決するまで相談に乗っていただき、大変お世話になりました。一昨年卒業された大島英里香さんには、本研究における環境構築など初めてだらけのことを丁寧に教えて下さり大変お世話になりました。同期である古井佑典君と北村和樹君にも大変お世話になりました。日常生活における他愛ない話からゼミなどの真剣な話まで、大変おもしろく有意義な研究室生活が送れました。ありがとうございました。また同じ研究室の Thomas さん、黒川君、関矢君、初芝君、望月君、荒井君、鎌田君、宮崎君、村田君にもお世話になりました。楽しい研究室生活を作ってくださいありがとうございます。

ほかにも多くの方に手助けいただき、本論文を完成させることができました。関わってください方々全てに感謝申し上げます。

最後に経済面や生活面など全てを支えてくださった家族に感謝申し上げます。家族のおかげでここまで自分のやりたいことをまっすぐ進めることができました。感謝してもしきれないです。ありがとうございました。

参考文献

- [1] Francis Halzen, Alan D.Martin, Quarks and Leptons, John Wiley & Sons Inc. 1984
- [2] The ATLAS Collaboration, Technical Design Report for the ATLAS Inner Tracker Pixel Detector. Technical report, CERN, (2018)
- [3] git module-qc-nonelec-gui <https://gitlab.cern.ch/atlas-itk/pixel/module/module-qc-nonelec-gui>
- [4] git module-qc-analysis-tools <https://gitlab.cern.ch/atlas-itk/pixel/module/module-qc-analysis-tools>
- [5] git localdb-tools <https://gitlab.cern.ch/YARR/localdb-tools>
- [6] localDBdocs <https://atlas-itk-pixel-localdb.web.cern.ch/>
- [7] ITkPD <https://itkpd-test.unicorncollege.cz/>
- [8] 大島英里香, 高輝度 LHC 実験 ATLAS ピクセル検出器モジュールの品質試験に向けたデータ管理用 GUI の開発, Master's thesis, 東京都立大学 <https://tokyo-metro-u.repo.nii.ac.jp/record/9891/files/T03046-001.pdf>
- [9] 木下怜士, ATLAS ピクセル検出器電荷較正の自動補完と次世代器量産管理システムの開発, Master's thesis, 東京工業大学 <https://tokyotech.box.com/s/b7x7z2pnf8ixa629gx4cacrc2gv26y0h>
- [10] 藤田侑葵子, 高輝度 LHC ATLAS 実験で使うシリコンピクセル検出器量産時の外観検査用のソフトウェアの開発, Master's thesis, 大阪大学 https://osksn2.hep.sci.osaka-u.ac.jp/theses/master/2022/MasterThesis2022_Fujita.pdf
- [11] 荒久田 周作, 高輝度 LHC ATLAS 実験で用いるシリコンピクセル検出器モジュールの閾値調整における目標閾値からの乖離, Master's thesis, 大阪大学 https://osksn2.hep.sci.osaka-u.ac.jp/theses/master/2022/MasterThesis2022_Arakuta.pdf
- [12] 窪田 ありさ, 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, Master's thesis, 東京工業大学 <https://cernbox.cern.ch/index.php/s/BdhvSTAuuE5xHXt>
- [13] 奥山 広貴, 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, Master's thesis, 東京工業大学 <https://cernbox.cern.ch/index.php/s/IB06BptySxPwD6L>
- [14] 生出秀行, 田窪洋介, 新 ATLAS ピクセル最内層「IBL」の建設 <https://www.jahep.org/hepnews/2014/14-2-1-ATLASIBL.pdf>

- [15] Brian Moser, Development and evaluation of prototypes for the ATLAS ITk pixel detector <https://cds.cern.ch/record/2837025/files/ATL-ITK-PROC-2022-020.pdf>
- [16] Unicorn, ITk Production Database Documentation <https://uos9.plus4u.net/uu-bookkitg01-main/78462435-41f76117152c4c6e947f498339998055/book>
- [17] mongoDB, <https://www.mongodb.com/>
- [18] ITkPix Module QC Doc Japan <https://atlas-jp-itk-pixel-module-qc.docs.cern.ch/>
- [19] PyQt <https://riverbankcomputing.com/software/pyqt/>
- [20] Grafana <https://grafana.com/>