

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

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 2 年 20844407
大島英里香

2022 年 1 月 7 日

概要

素粒子は物質を構成する最小の単位であり、その種類を示す標準模型は未だ完成していない。標準模型を超える新物理の探索を目標とする実験の1つに衝突型加速器実験がある。

スイスにある LHC (Large Hadron Colider) は CERN(欧州原子核研究機構) の運営する大型ハドロン加速器である。世界最大のエネルギーを誇り、エネルギーフロンティアで素粒子物理学において新物理の探索を目的とする。2013年には LHC の4つの検出器のうち CMS と ATLAS の2つの実験グループがヒッグス粒子の発見に成功した。LHC はヒッグス粒子のさらなる検証等を目的として、より高輝度での実験を行うため HL (High Luminosity)-LHC へのアップグレードを予定している。積分ルミノシティは約 10 倍となり、検出器性能の飛躍的な向上が期待されるが、それに伴い高い放射線耐性や位置分解能、高速な読み出しが求められる。LHC の4つの検出器の中の1つである ATLAS 検出器では内部飛跡検出器が ITk (Inner Tracker) に置き換わる予定である。ITk はシリコンピクセル検出器と半導体ストリップ検出器から成る。ITk で用いられるモジュールの主な構成は FE チップ、センサー、フレキシブル基板、バンプ接続である。このモジュールは世界の各機関でおよそ 10,000 個量産される予定であり、また組み立てに当たってさまざまな品質試験が行われている。品質試験の結果はデータベースに保存される。このデータベースへアップロードするための GUI が先行研究で開発されていた。GUI の開発において、品質試験の項目の追加や変更への対応に加え、よりユーザー目線で使いやすく改良していく必要がある。

本研究では、品質試験の1つである平坦性測定に関する GUI の改良についてと、GUI のバージョン情報を登録する機能の追加を行った。

まず、平坦性測定に関する GUI の改良である。平坦性測定をアップロードする GUI は今までおよそ 23 項目を手打ちする仕様であった。そこで平坦性測定を実際に行った際にアウトプットされるデータをユーザーに JSON 形式で編集してもらい、そのファイルを直接入力できるように改良した。また、入力ファイルにデータベースに保存すべき項目が揃っているか確認できる機能や、確認画面でどの項目がモジュールのどの部分に当たるか一目でわかるよう ITk モジュールマップの追加も行った。さらに、Quad と Triplet の2種類あるモジュールに対して Quad のみの対応であったのに対し、Triplet への対応についても開発を行った。改良した GUI について、実際にデータベースへのアップロードを行い問題なく動作することを確認した。

次に、日々開発されアップデートされる GUI において、データの整合性を保つためにどのバージョンの GUI を用いてテスト結果をアップロードしたのかという情報は重要である。そこで、バージョン管理システムである git の仕組みを用いて、ユーザーがテスト結果をアップロードする際、同時にデータベースへその GUI のバージョン情報もアップロードできるような機能の開発も行った。ユーザーはテスト結果をアップロードする際に確認画面でバージョン情報を確認できる他、アップロードした後もテストの選択画面から詳細情報でバージョン情報を確認できる。

本研究より、より効率的にかつデータベースへ試験結果を誤りなくアップロードすることが可能となった。

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	1
1.1.3 新物理探索	4
1.2 ATLAS 実験	5
1.3 ATLAS 検出器	6
1.3.1 内部飛跡検出器	8
1.3.2 カロリメータ	10
1.3.3 ミューオン検出器	11
1.4 HL-LHC 計画	11
1.4.1 素粒子物理に対する目標	13
1.4.2 内部飛跡検出器のアップグレード	13
第 2 章 ITk シリコンピクセル検出器	16
2.1 ITk シリコンピクセル検出器の構成	16
2.1.1 シリコンピクセルセンサー	17
2.1.2 読み出し集積回路 (FE チップ)	18
2.1.3 フレキシブル基板 (FPC)	19
2.2 ITk シリコンピクセルモジュール	19
2.2.1 モジュールの種類	19
2.2.2 モジュールの量産	20
第 3 章 品質試験	21
3.1 ピクセルモジュール組み立て手順	21
3.2 品質試験の種類	22
3.2.1 目視検査	22
3.2.2 質量測定	22
3.2.3 平坦性測定	23
3.2.4 センサー電流-電圧特性測定	24
3.2.5 FE チップ電流-電圧特性測定	24

第 4 章 品質管理データベース	25
4.1 品質試験におけるデータ管理	25
4.2 データベースと関連のソフトウェア	25
4.2.1 中央データベース	26
4.2.2 ローカルデータベース	26
4.2.3 YARR(Yet Another Rapid Readout)	27
4.3 QChelper	28
4.3.1 テスト結果登録	30
4.3.2 ベアモジュールの登録	36
第 5 章 QChelper の改良	37
5.1 平坦性測定項目入力 GUI の改良	37
5.2 バージョン情報のアップロード	44
第 6 章 まとめ	46
付 録 A Databasetools review 資料	48
付 録 B 平坦性測定で使われる JSON ファイル	51
付 録 C git	52
参考文献	53

目次

1.1	素粒子標準模型	2
1.2	ヒッグスポテンシャル	2
1.3	ヒッグス粒子とみられる粒子が発見された時の ATLAS による $H \rightarrow \gamma\gamma$ の不変質量分布図	3
1.4	LHC の全体像	5
1.5	ATLAS 検出器の全体像	6
1.6	ATLAS 実験で用いられる座標系のイメージ	6
1.7	内部飛跡検出器の概要図	8
1.8	ピクセル検出器の全体図	9
1.9	ピクセルモジュールの構成	9
1.10	カロリメータの全体図	10
1.11	ミューオン検出器の全体図	11
1.12	HL-LHC 計画のスケジュール	12
1.13	SM ヒッグス粒子について粒子の質量とウィークボソン、フェルミオンの結合定数の統計誤差の予測	14
1.14	アップグレード後の ITk の全体イメージ図	14
1.15	ITk の断面図	15
2.1	インナーシステム	16
2.2	エンドキャップ	16
2.3	アウターバレル	16
2.4	ピクセルモジュールのイメージ図	17
2.5	ITkPixV1 全体図	18
2.6	ピクセルモジュールにおけるデータの信号伝達の概念図。	19
2.7	ITk におけるモジュールの種類	19
2.8	モジュールの各配置	20
2.9	LHC の全体像	20
3.1	モジュール組立工程	21
3.2	目視検査用に撮影されたモジュール写真	22
3.3	平坦性測定の装置	23
3.4	平坦性測定における測定項目	23

4.1	データベースの概念図	25
4.2	データベースと関連のソフトウェアの関係	26
4.3	LocalDB のブラウザ	27
4.4	QChelper の構造	28
4.5	QChelper の流れ	29
4.6	テスト結果のアップロード工程	30
4.7	テストの登録状況が確認できる GUI	30
4.8	質量測定の入力画面 GUI	31
4.9	モジュール情報登録 GUI。	31
4.10	外観検査の GUI	32
4.11	外観検査における画像のマッピング	32
4.12	ワイヤー接続情報 GUI。	33
4.13	平坦性測定項目入力画面 GUI	34
4.14	センサー電流電圧特性 GUI。	35
4.15	FE チップ電流電圧特性 GUI。	35
4.16	ベアモジュール登録の工程。	36
5.1	改良した平坦性測定項目の入力画面	37
5.2	平坦性測定項目のソースコードの構造	38
5.3	平坦性測定項目のソースコードの流れ	38
5.4	Quad モジュールマップを加えた確認画面	41
5.5	Triplet モジュールのマップ	42
5.6	git におけるバージョン情報の概念図	44
5.7	QChelper のバージョン情報記載場所	45
5.8	QChelper のバージョン情報記載場所	45
C.1	git の概念図	52
C.2	コミットのイメージ	53

表 目 次

1.1	現在の LHC と HL-LHC の比較	13
1.2	現在の内部飛跡検出器と ITk の比較	15
2.1	モジュール詳細 [1]	17
2.2	ITkPix の性能 [2]	18
4.1	QChelper がサポートしている試験項目一覧	28

第1章 序論

素粒子はこの世界のあらゆる物質を構成する最小の単位であり、またその粒子の性質や粒子間の相互作用を解明する学問が素粒子物理学である。素粒子の性質を実験的に探るために最適化された方法が素粒子同士を衝突させてその相互作用を解析することである。これは衝突型加速器実験と呼ばれ、スイスにある欧州原子核研究機構 (CERN) が運営する大型ハドロン衝突型加速器 (LHC) は現在世界で一番大きな加速器である。本章では素粒子物理学と LHC の中の 1 つの検出器で行われている ATLAS 実験について記す。

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いて基本的な枠組みのことを標準模型と呼ぶ。粒子間には重力、弱い力、電磁力、強い力の 4 種類の力があることが知られている。図 1.1 に示すように、ともに 6 種類存在するクォークとレプトン、重力を除く 3 つの相互作用から分けられるゲージ粒子、ヒッグス粒子から成る。クォークがそれぞれ 3 種類あるのはクォークがカラーと呼ばれる自由度を持つためである。ゲージ粒子は力を伝える粒子であり、フォトンが電磁力、ウィークボソンが弱い力、グルーオンが強い力を伝える。現在までの実験的事実はこの標準模型に矛盾することなく一致してきた。しかし、宇宙に存在する物質を標準模型で説明できる粒子は全物質の 15% ほどと言われている。この標準模型は完全なものではなく近似的に成り立つものと考えられ、標準模型を超える新物理の発見が待たれる。

1.1.2 ヒッグス機構

標準模型では、ヒッグス粒子は中性スカラー粒子であり、そのラグランジアンは以下のように書ける。

$$L = (D_\mu \phi)^\dagger (D^\mu \phi) - V(\phi^\dagger \phi) \quad (1.1)$$

スカラー ϕ に対するポテンシャルエネルギーを

$$V = \mu^2 \phi^\dagger \phi + \lambda (\phi^\dagger \phi)^2 \quad (1.2)$$

と仮定する。 V はヒッグスポテンシャルと呼ばれる。 $\mu^2 < 0$ 、 $\lambda > 0$ とすると V は

$$\sqrt{\phi^\dagger \phi} = \sqrt{\frac{-\mu^2}{\lambda}} = \frac{v}{\sqrt{2}} \quad (1.3)$$

で極小となり、この最低エネルギーの点を真空状態と呼び、 $\frac{v}{\sqrt{2}}$ は場 ϕ の真空期待値である。図 1.2 のように最低エネルギーの点は一点には定まっておらず、真の真空がどこになるかは偶然で決まる。これを自発的対称性の破れという。

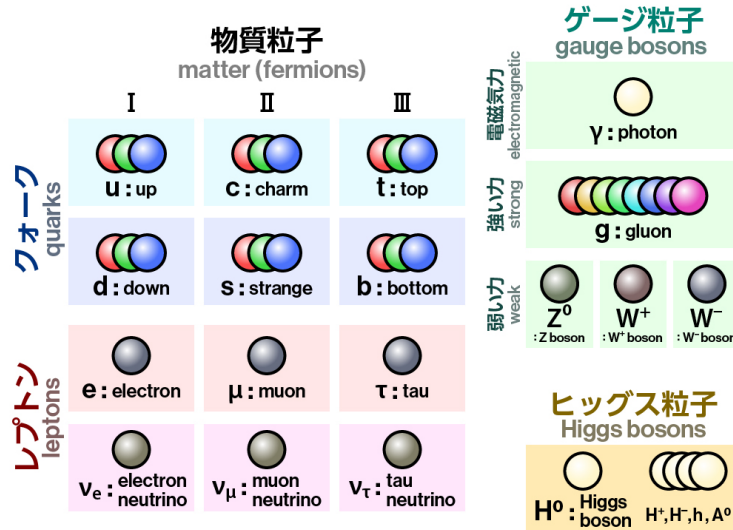


図 1.1: 素粒子標準模型

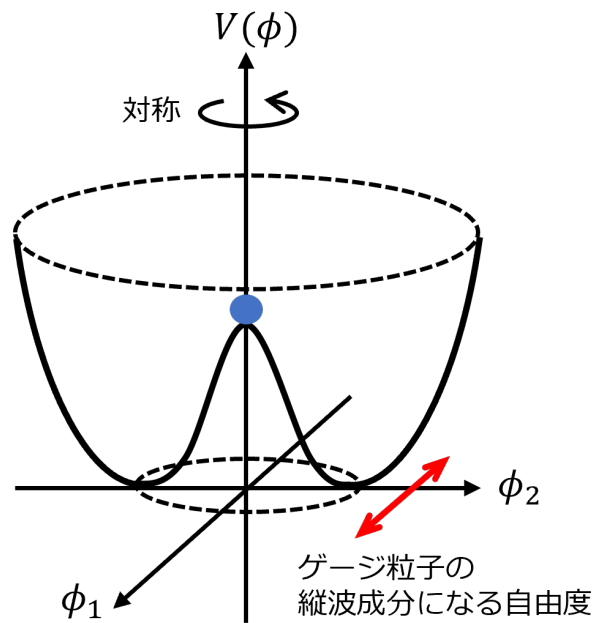


図 1.2: ヒッグスポテンシャル

さらに、SU(2) ゲージ変換で

$$\phi = \begin{pmatrix} 0 \\ \frac{v+\eta}{\sqrt{2}} \end{pmatrix} \quad (1.4)$$

としてラグランジアン密度に導入すると粒子 η の質量 $m_\eta = \sqrt{-2\eta^2}$ が得られる。この電氣的に中性な粒子 η をヒッグス粒子と呼ぶ。

また、Weinberg 角

$$\tan\theta_W = \frac{g'}{g} \quad (1.5)$$

を用いて新しいゲージ場を考える。すると、本来質量をもたないゲージ粒子 W と Z について

$$m_W = \frac{1}{2}vg, m_Z = \frac{1}{2}v\sqrt{g^2 + g'^2} \quad (1.6)$$

となる。ゆえに

$$\frac{m_W}{m_Z} = \cos\theta_W \quad (1.7)$$

と表せる。W[±]、Z は質量が 0 の時横波で 2 つの自由度しか持たない。しかし、質量を持つと 3 つ目の自由度である縦波を考えなければならない。このように、スカラー場に真空期待値を持たせ、ゲージ粒子に質量を与え、ヒッグス場の自由度をゲージ場の縦波成分としてしまうことをヒッグス機構と呼ぶ。

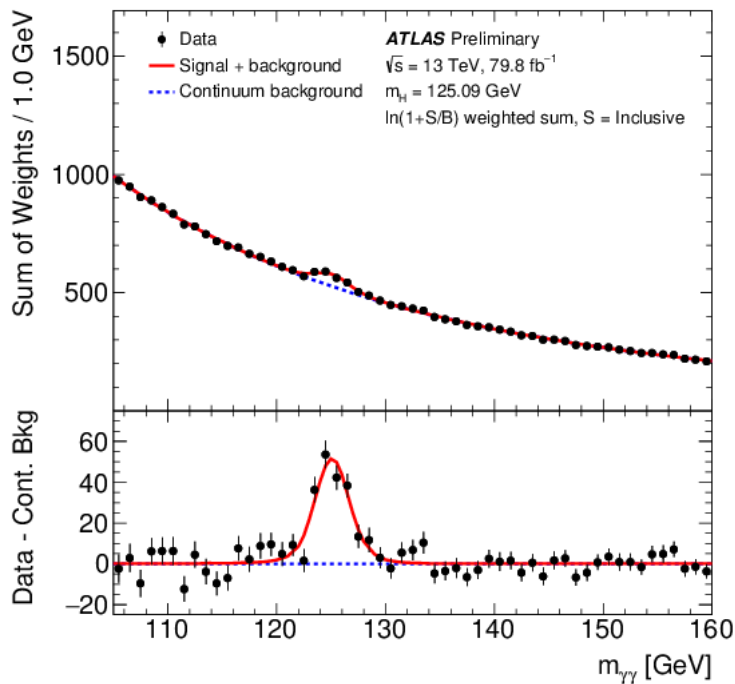


図 1.3: ヒッグス粒子とみられる粒子が発見された時の ATLAS による $H \rightarrow \gamma\gamma$ の不変質量分布図。ヒッグス粒子はある確率で 2 つの γ 線に崩壊する。横軸は観測された運動量から再構成された質量である。検出された 2 つの γ 線が 1 つのヒッグス粒子の崩壊から来るものであれば、その足し合わせは元のヒッグス粒子 1 つの質量である。一方、全く関係ない γ 線同士の足し合わせ (偽の事象) であればランダムな分布のバックグラウンドとなる。図の赤点線はそのバックグラウンドの予想分布である。ところが実際の信号 (赤線) は 125 GeV にピークを示し、ヒッグス粒子の存在を示唆した。優位性はおよそ 5σ である。

標準模型では電氣的に中性なスカラー粒子としてヒッグス粒子が 1 種類だけ存在するが、他の模型だと異なる可能性があるため更なる探索や精密測定が重要となる。

1.1.3 新物理探索

現在の標準理論では説明できない事柄が多くある。ここではそのいくつかを簡単に紹介する。

- 力の統一

標準理論は弱い力と電磁力を統一する電弱統一理論を含んでいる。しかしながら、力の統一は未完成である。大統一理論は、弱い力、電磁力、強い力を1つに統一する理論である。さらに3つの力に重力を加えようとする理論が超大統一理論である。

- ヒッグス粒子の正体

実験で観測されたヒッグス粒子の質量は約 125 GeV であったが、大統一理論などのエネルギースケールに比べはるかに軽いという謎がある。また、なぜ空っぽであると考えられてきた真空にヒッグス粒子が存在しているのか、そもそもヒッグス粒子は単一の素粒子であるのか未知の素粒子が結合してできた複合素粒子であるのかも分かっていない。

- 重力を伝える粒子の存在

存在する4つの力(重力、弱い力、電磁力、強い力)のうち標準理論で用いられるのは重力以外の3つである。つまり標準理論で重力を扱うことはできない。質量をもつものを近づけようとする、重力相互作用を伝える素粒子は未発見である。

- 暗黒エネルギー

1.1.1 項で触れたように、宇宙を構成するエネルギーや物質のうち、標準理論ではわずか5%しか説明することができず、大部分は未だ謎のままである。宇宙を占めるエネルギーや物質のおよそ3割弱が光を発しない暗黒物質、7割が宇宙を加速膨張させる暗黒エネルギーとよばれるものである。

1.2 ATLAS 実験

スイスにある大型ハドロン衝突型加速器 (LHC) は欧州合同原子核研究機関 (CERN) の運営する、周約 27 km の大型の陽子陽子衝突型加速器である。衝突エネルギーは約 14 TeV と世界最大を誇る。ATLAS は LHC の有する 4 つの検出器 (CMS, LHCb, ALICE, ATLAS) のうちの 1 つである。最大 7 TeV まで加速された 2 本の陽子ビームを検出器の中心で衝突させ、それによって発生した粒子を衝突点付近の様々な検出器で測定し、取得したデータを物理解析する。

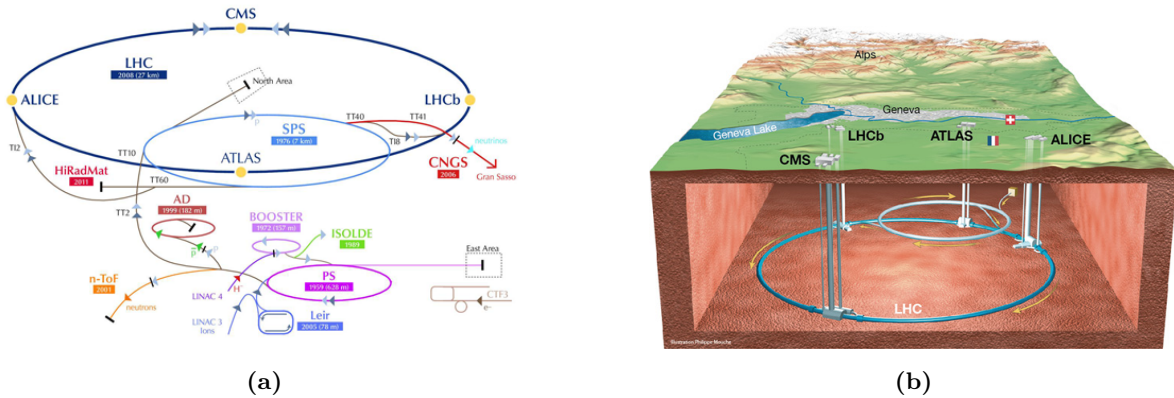


図 1.4: (a)LHC の全体像。4 つの衝突点 ALICE (A Large Ion Colider), LHCb (Large Hadron Collider beauty), CMS (The Compact Muon Solenoid), ATLAS (A Troidal LHC ApparatuS) で実験が行われている。(b)LHC はおよそ地下 100 m に位置する。

また、LHC 含む加速器実験において検出器の大事なパラメータとして陽子ビーム強度を表すのがルミノシティである。ルミノシティは粒子の毎秒あたりの衝突頻度を表す。粒子が衝突する反応断面積を σ 、その反応のイベント数を N 、時間を t 、ルミノシティを L とすると以下の式で与えられえる。

$$\frac{dN}{dt} = \sigma \times L \quad (1.8)$$

積分ルミノシティは時間に対するルミノシティの積分で与えられ、 $\int L dt$ と書く。ゆえに加速器実験における全実験期間での物理現象の全反応数はこの積分ルミノシティで決まる。そのため、ルミノシティは加速器の性能の指標の 1 つとなる。

1.3 ATLAS 検出器

ATLAS (A Toroidal LHC ApparatuS) は LHC の持つ検出器 4 つ (図 1.4a) の内の 1 つで行われている実験である。検出器で生成される粒子の衝突点の周りには 6 つの同心円状の層があり、それぞれ異なった検出器のシステムを担う。それによって粒子の軌道、運動量などを測定する。ATLAS 検出器全体を図 1.5 に示す。衝突点内側から内部飛跡検出器、ソレノイド磁石、カロリメータ、トロイド磁石、ミューオン検出器が置かれている。

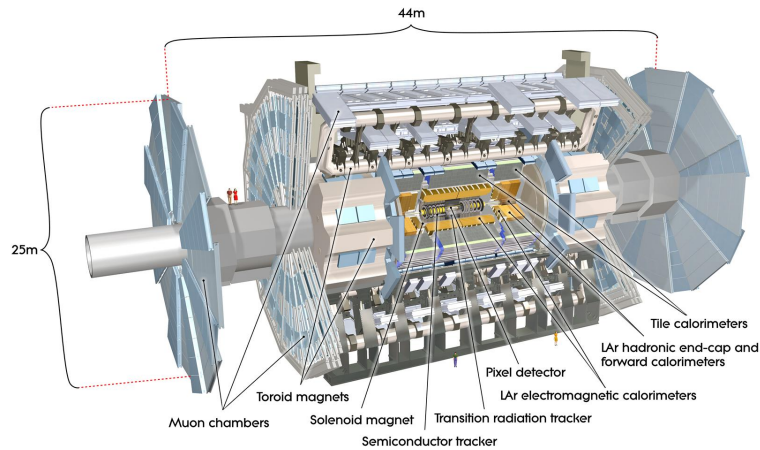


図 1.5: ATLAS 検出器の全体像

ここで ATLAS 実験で用いられる座標系について記す。座標系のイメージを図 1.6 に示す。衝突点を

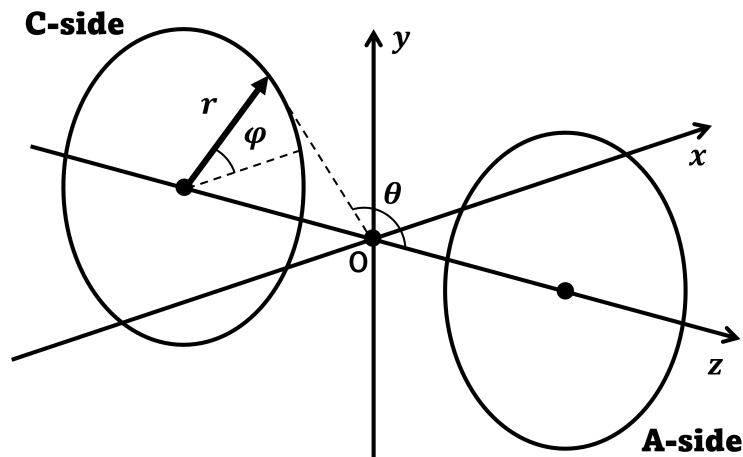


図 1.6: ATLAS 実験で用いられる座標系のイメージ

原点とし、 $x-y$ 平面はビームラインに垂直に、 z 軸はビームラインに沿って定義される。 x 軸の正は衝突点から LHC リングの中心方向を、 y 軸の正は地表上向きを指す。 z 軸の正方向を検出器の A-side、負方向を C-side と呼ぶ。 $x-y$ 平面においては $r-\phi$ 座標系が用いられ、半径方向 r はビームラインからの距離、方位角 ϕ は x 軸からの角度、極角 θ は正の z 軸からの角度である。また、ATLAS 実験を含む

一般的な加速器実験ではこの極角 θ を用いて擬ラピディティは以下のように表される。

$$\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right] \quad (1.9)$$

1.3.1 内部飛跡検出器

ATLAS 検出器の最も内側にあり、陽子ビームが衝突して生成された荷電粒子が初めに検出されるのが内部飛跡検出器である。ここでは粒子の飛跡を検出する役割を担い、粒子の方向や運動量、電荷を測定する。内部飛跡検出器は主にピクセル検出器、半導体ストリップ検出器、遷移輻射検出器の3つから成り、ソレノイド磁石によって作られた2 Tの磁場で動作する。これにより磁場中における粒子のらせん運動を測定できる分解能を持つ。図 1.7a、1.7b に内部飛跡検出器の全体図、断面図を示す。また、3

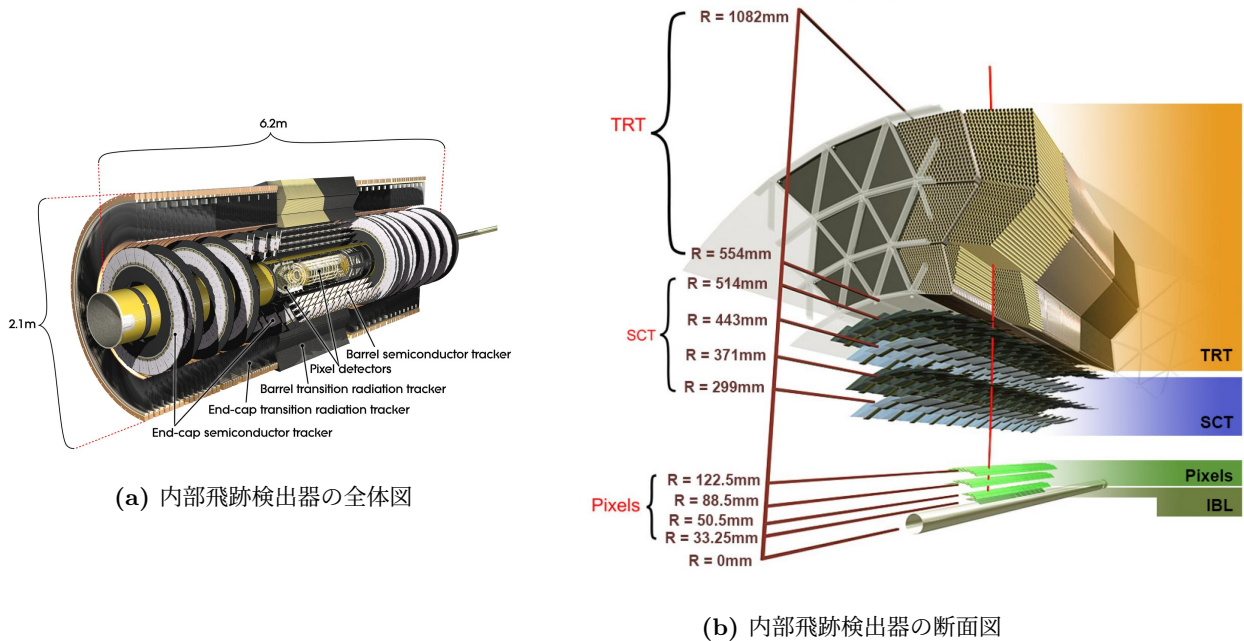


図 1.7: 内部飛跡検出器の概要図

つの検出器にはシリコンセンサー、ストローチューブという2つの技術が用いられている。前者では粒子がシリコンセンサーを通過するとき、電子と正孔のペアを作る。電場をかけることで電荷を集め粒子の位置を特定することができる。後者では粒子がドリフト領域を通過すると管内にあるガスがイオン化され、自由になった電子が管内中心にあるワイヤーにドリフトされ記録される。シリコンピクセルはピクセル検出器、シリコンストリップは半導体ストリップ検出器、ストローチューブは遷移輻射検出器でそれぞれ使われている。

ピクセル検出器

ピクセル検出器は衝突点に最も近い検出器であり、高いチャンネル密度分布をもつ。8000万個以上のチャンネルから構成され、粒子1個当たり平均3回の測定を行い、位置分解能は $r-\phi$ 平面で $10\ \mu\text{m}$ 、 z 方向で $115\ \mu\text{m}$ である。約 $2\ \text{cm} \times 6\ \text{cm}$ の長方形のモジュールが基本的な構成単位であり、 $50\ \mu\text{m} \times 400\ \mu\text{m}$ の範囲に渡って設置される。ピクセル検出器とピクセルモジュールのイメージを図1.8、1.9に示す。

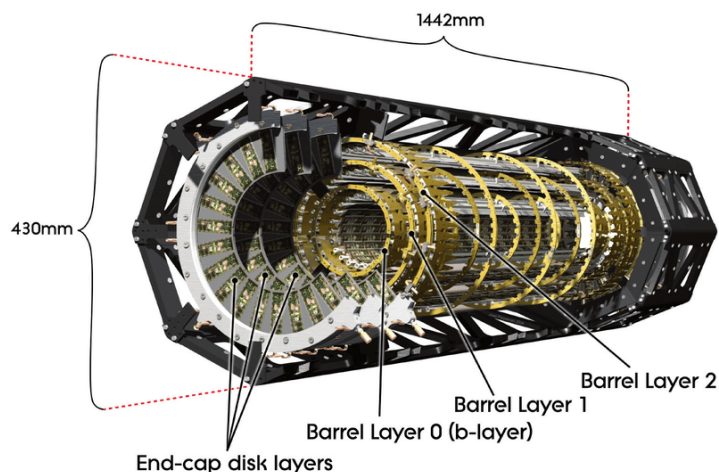


図 1.8: ピクセル検出器の全体図

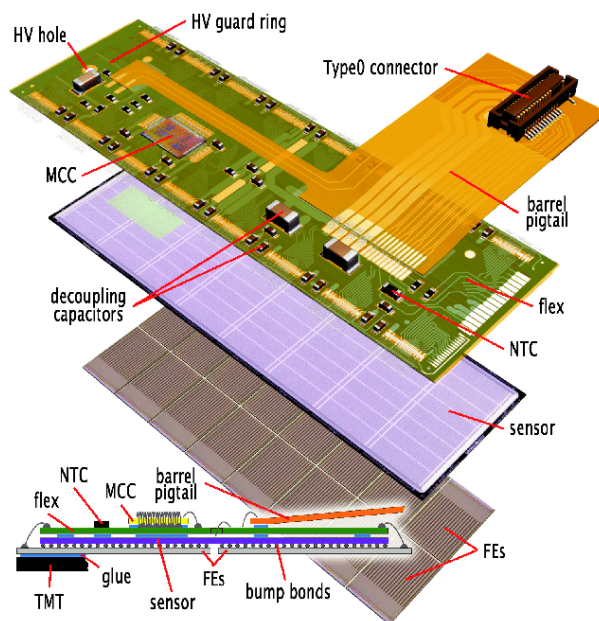


図 1.9: ピクセルモジュールの構成

バレル部は4層あり、IBL、Pixel B Layer、Pixel Layer 1、Pixel Layer 2 から成る。

半導体ストリップ検出器

半導体ストリップ検出器はピクセル検出器を覆うように設置され、バレル部が4層、エンドキャップが18層で構成される。各層シリコンストリップセンサーの2層構造になっている。粒子の位置を $r-\phi$ 面で $17\ \mu\text{m}$ 、 z 方向で $580\ \mu\text{m}$ の精度で測定する。

遷移輻射検出器

内部飛跡検出器の中で最も外側にあり最も大きいのが遷移輻射検出器である。遷移輻射検出器は約30万本のストローチューブで構成され、バレル部の $r-\phi$ 面、エンドキャップ部の $z-\phi$ 面において約 $130\ \mu\text{m}$ の精度でドリフトした電子から位置を測定する。粒子1個あたり36回のヒット数で測ることができる。また、位置測定だけでなく遷移輻射によって粒子の識別も行うことができる。粒子がこの検出器を通過する際に遷移輻射を起こし、ストローチューブ内の混合ガスの分子がイオン化され、その遷移輻射によって粒子を識別する。

1.3.2 カロリメータ

図1.10にカロリメータの全体図を示す。内部飛跡検出器のカロリメータは電磁カロリメータとハドロンカロリメータの2つから成り、ハドロン、電子、光子のエネルギーを測定する。入射した粒子はカロ

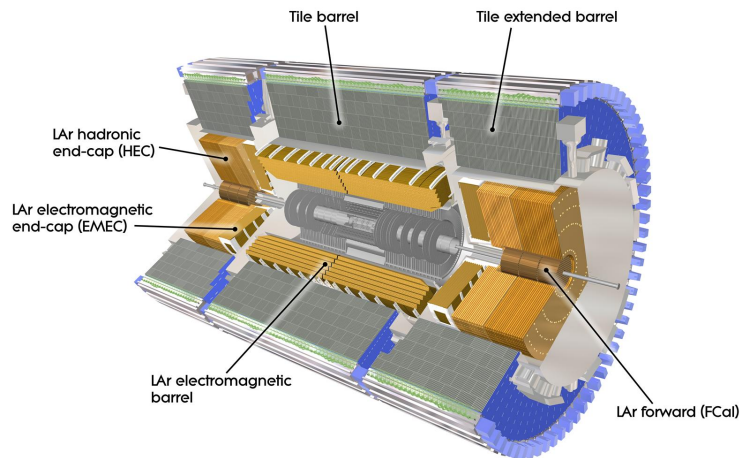


図 1.10: カロリメータの全体図

リメータの中で電磁シャワーと呼ばれる粒子群を発生させる。検出した電磁シャワーの拡がりから元の全体のエネルギーを測定することができる。電子と光子のエネルギーは、液体アルゴンで満たされたアコーディオン構造を持つ電磁カロリメータで測定される。ハドロンカロリメータはハドロンエネルギーを測定する。光子は元の粒子のエネルギーに比例する電気信号に変換される。

1.3.3 ミューオン検出器

ミューオン検出器はThin Gap Chambers (TGC)、Resistive Plate Chambers (RPC)、Monitored Drift Tubes (MDT)、Cathode Strip Chambers (CSC) の4つの検出器で構成される。ATLAS 検出器の最外層に位置し、数千個のチャンバーから成る。ごく小さな規模において、このチャンバーはわずかな温度変化に伴い形状を変化させる。ゆえにミューオン検出器には光学的なアライメントシステムが搭載されており、チャンバー同士の相対的な位置関係やチャンバーの変形などはリアルタイムで常に把握されている。 η 領域における大部分では MDT によって衝突点から飛び出すミューオンの位置が測定され、 $2 < |\eta| < 2.7$ の範囲では CSC がその役割を担う。分解能はそれぞれ $80\mu\text{m}$ 、 $60\mu\text{m}$ である。また、バレル部では RPC、エンドキャップ部では TGC がそれぞれトリガーチャンバーとして使用される。

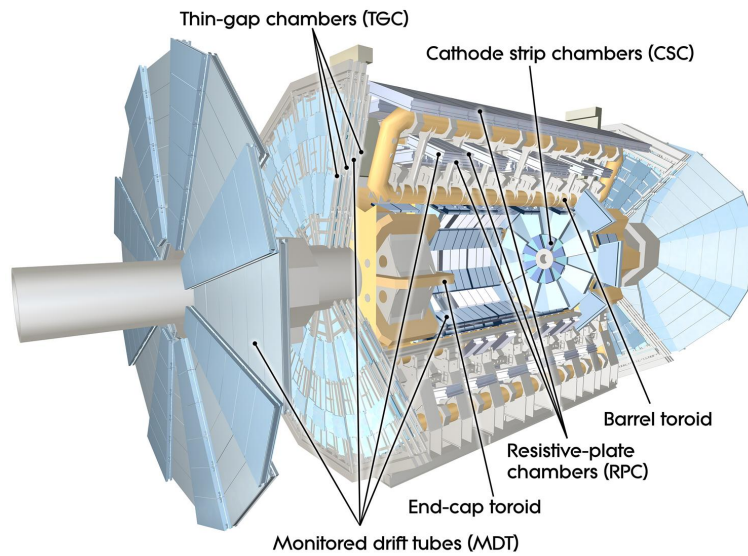


図 1.11: ミューオン検出器の全体図

1.4 HL-LHC 計画

LHC はより高輝度での実験である、HL (High Luminosity)-LHC へのアップグレードを予定している。これを HL-LHC 計画と呼び、表 1.1 にあるようにより多くのデータを得るためルミノシティを 3.5 倍程度に増加し、積分ルミノシティを最終的には 10 倍以上に増やす予定である。例えば、2017 年時点の LHC のヒッグス粒子生成量が年間約 300 万個であるのと比べ、HL-LHC は年間約 1500 万個を作ることができる。現在 (2021 年) におけるスケジュールを図 1.12 に示す。

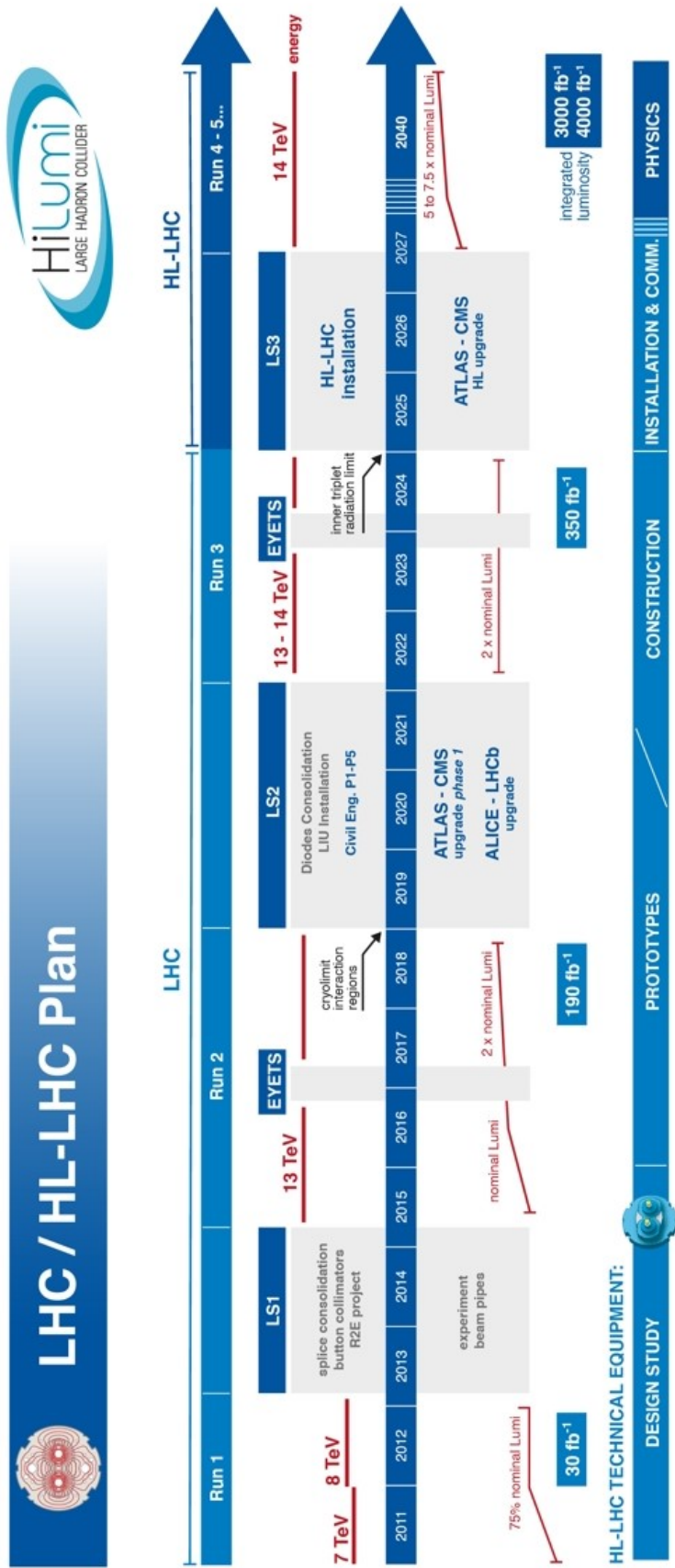


図 1.12: HL-LHC 計画のスケジュール

現在の LHC は LS2 (Long shutdown) と呼ばれる期間にあり、運転はしておらず HL-LHC に向けて 1 段階目としてとして主にカロリメータやミューオン検出器、トリガー、DAQ システムのアップグレードが行われている。現在の LS2 を終わると、13-14TeV で約 3 年間運転 (Run3) を再開し、さらに 2 年半の HL-LHC に向けてのインストール期間として LS3 期間が設けられ、およそ 2027 年ころから HL-LHC の運用が始まる。HL-LHC の運転は 10 年を予定している。

表 1.1: 現在の 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

1.4.1 素粒子物理に対する目標

HL-LHC において重要な物理探索の 1 つは、2013 年発見があった質量 125 GeV のヒッグス粒子のさらなる検証である。HL-LHC における ATLAS 実験と CMS 実験では、ヒッグス粒子と標準モデルのフェルミオンやボソンとの結合についておよそ 2~5% の精度で測定できることが期待されている。図 1.13 から 3000fb^{-1} での実験データから期待されるウィークボソンとフェルミオンの結合の測定が向上する見込みであることがわかる。

1.4.2 内部飛跡検出器のアップグレード

より高輝度での実験に伴い検出器もアップグレードしなければならない。HL-LHC での実験において放射線量は $2 \times 10^{16} \text{n}_{\text{eq}}/\text{cm}^2$ (1MeV 中性子換算) と予想され、高い放射線耐久性が求められる上高速な読み出しと高い位置分解能も重要となる。それに向けて現在 ATLAS 検出器で使われている内部飛跡検出器は LS3 の間にシリコンをベースとした ITk (Inner Tracker) に置き換わる。ITk は 5 つのピクセル検出器の層と 4 つの半導体ストリップ検出器の層から成る。ITk の全体図を図 1.14 に示す。

現在の内部飛跡検出器と比べ、チャンネル数は約 10^3 倍、半導体ストリップ検出器とピクセル検出器の表面積はそれぞれ 3 倍と 7 倍になり大幅に大きくなる。現行の内部飛跡検出器と ITk のモジュール数などの比較を表に示す。

ここで ITk の断面図を図 1.15 に示す。

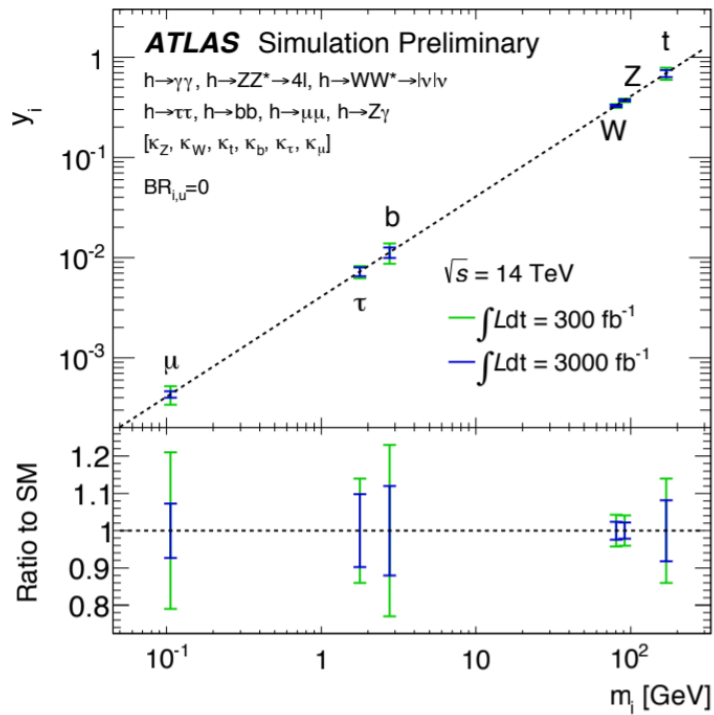


図 1.13: ATLAS 実験において 300fb^{-1} と 3000fb^{-1} における、質量 125 GeV を仮定した場合の SM ヒッグス粒子について粒子の質量とウィークボソン、フェルミオンの結合定数の統計誤差の予測。緑線が LHC、青線が HL-LHC を示す。

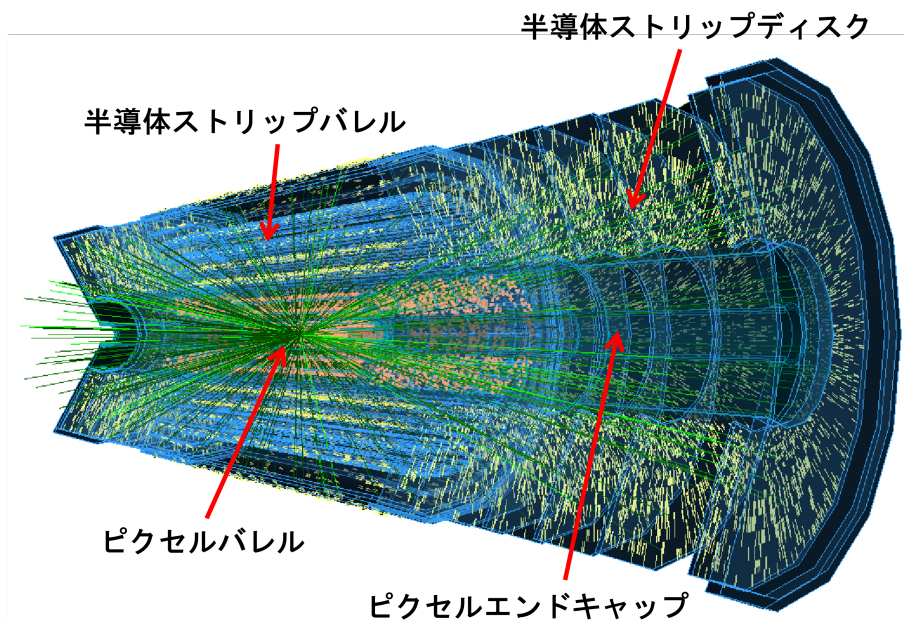


図 1.14: アップグレード後の ITk の全体イメージ図

表 1.2: 現在の内部飛跡検出器と ITk の比較

	現行の内部飛跡検出器	HL-LHC の ITk
シリコン占有率 [m^2] (ピクセル検出器)	1.9	13
シリコン占有率 [m^2] (半導体ストリップ検出器)[fb^{-1}]	60	160
チャンネル数 (ピクセル検出器)	$\approx 9.2 \times 10^6$	$\approx 5 \times 10^9$
チャンネル数 (半導体ストリップ検出器)	$\approx 6 \times 10^6$	$\approx 50 \times 10^6$
$ \eta $ カバー範囲	2.5	4.0
ピクセルサイズ [μm^2]	50×400 と 50×250	50×50 もしくは 25×100

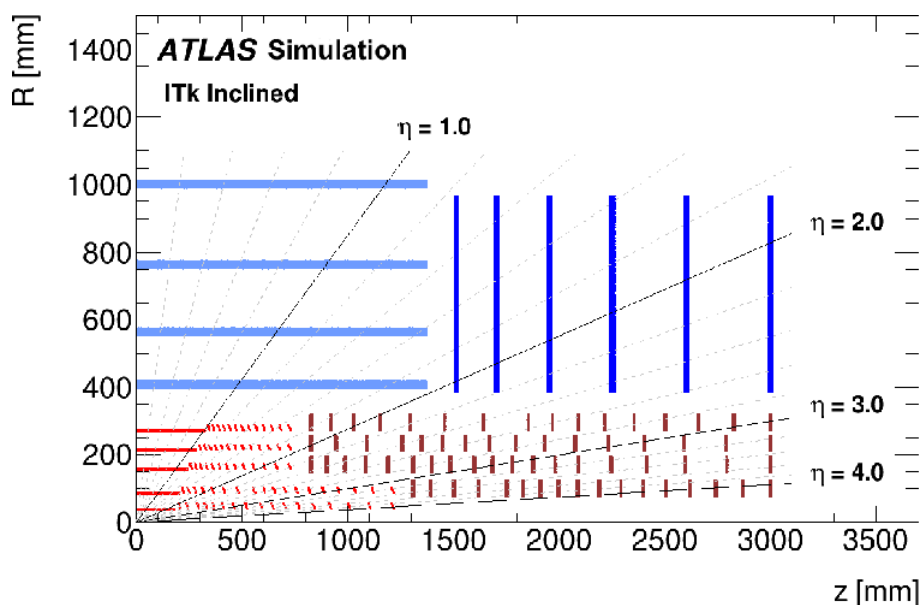


図 1.15: ITk の断面図。横軸はビームラインに沿い、原点は衝突点、縦軸は衝突点からの半径である。青は半導体ストリップ検出器の能動素子、赤はピクセル検出器の能動素子を示す。5つのバレル層とエンドキャップ部で構成されたピクセル検出器で $|\eta| < 4$ をカバーし、最大半径 1 m まで伸びる半導体ストリップ検出器は 4つのバレル層と 6つのエンドキャップによって構成され、 $|\eta| < 2.7$ をカバーする。

第2章 ITk シリコンピクセル検出器

本章では ITk に用いられる新型のシリコンピクセル検出器について詳細に記す。

2.1 ITk シリコンピクセル検出器の構成

ITk シリコンピクセル検出器は図 2.1、2.2、2.3 のようにインナーシステム、エンドキャップ、アウトバーレルの3つの構成部から成る。

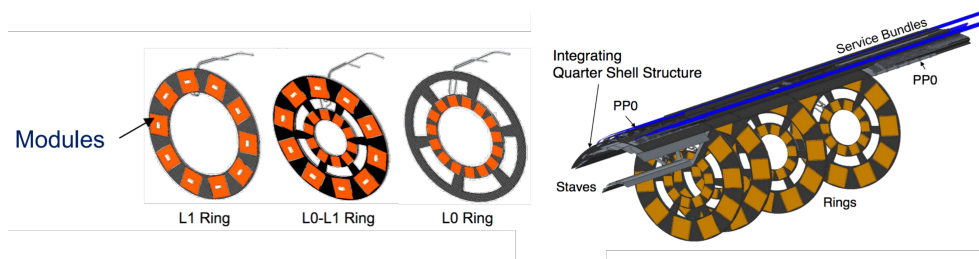


図 2.1: インナーシステム

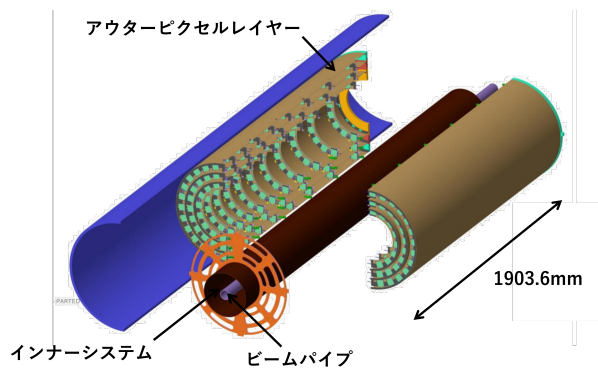


図 2.2: エンドキャップ

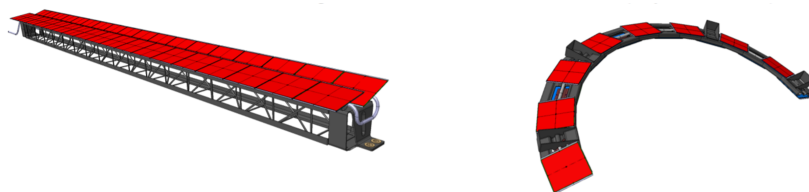


図 2.3: アウターバレル

ITk シリコンピクセル検出器モジュール

ITk シリコンピクセル検出器は約 1 万個のモジュールで構成され、搭載されるモジュール (以下ピクセルモジュール) は FE チップ (読み出し集積回路) とシリコンセンサーを bumps ボンディングでフレキシブル基板にワイヤ接続したものである。フレキシブル基板は FE チップに電圧やデータを送ったり、センサーの pn 接合部に逆バイアスを与える機能を持つ。モジュールの概略図を図 2.4 に示す。

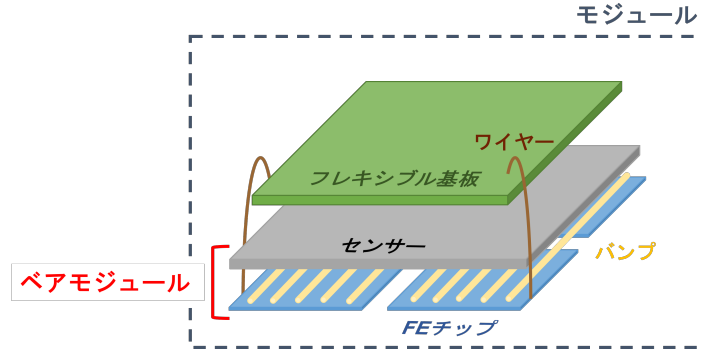


図 2.4: ピクセルモジュールのイメージ図。ここでは FE チップが 4 枚であるクアドモジュールをモデルとした。モジュールは FE チップとセンサーから成るベアモジュールと、それにフレキシブル基板を取り付け完成品となるモジュールがある。

モジュールは FE チップの数によって種類が異なり、それに伴い品質試験も異なる。モジュールの種類については後述する。

2.1.1 シリコンピクセルセンサー

ITk には 3D とプラナーの 2 種のセンサーが使われる。L2~L4 には厚さ $150\mu\text{m}$ のプラナー型センサー、L1 には厚さ $100\mu\text{m}$ のプラナー型センサーが配置されている。最内層には最も放射線耐性に優れた 3D 型センサーが置かれる。表 2.1 に詳細を記す。

表 2.1: モジュール詳細 [1]

層	センサータイプ	ピクセルサイズ μm^2	モジュールタイプ
L0 バレル部	3D n-in-p	25×100	Triplet
L0 リング	3D n-in-p	50×100	Triplet
L1	planar n-in-p	50×50	Quad
L2~4	planar n-in-p	50×50	Quad

センサーはバルク部に p 型半導体、電極部に n^+ 半導体を用いた $n^+ - in - p$ 型である。in は絶縁体 (insulator) のことである。逆バイアス電圧をかけ、完全空乏層領域を荷電粒子が通過すると、粒子はエネルギーを失い電子・ホール対が生成され、 n^+ 電極から信号が伝送される仕組みである。

2.1.2 読み出し集積回路 (FE チップ)

ピクセルモジュールの読み出し集積回路は RD53 コラボレーションによって開発された。RD53 コラボレーションは HL-LHC における、ATLAS と CMS 実験の検出器読み出しチップの開発のために組織された [3]。3 種類の前エンドから成る RD53A のプロトタイプから ITkPixV1 として開発され、最終版である ITkPixV2 が作られる予定である。現在の ITkPixV1 の全体図を図 2.5 に示す。

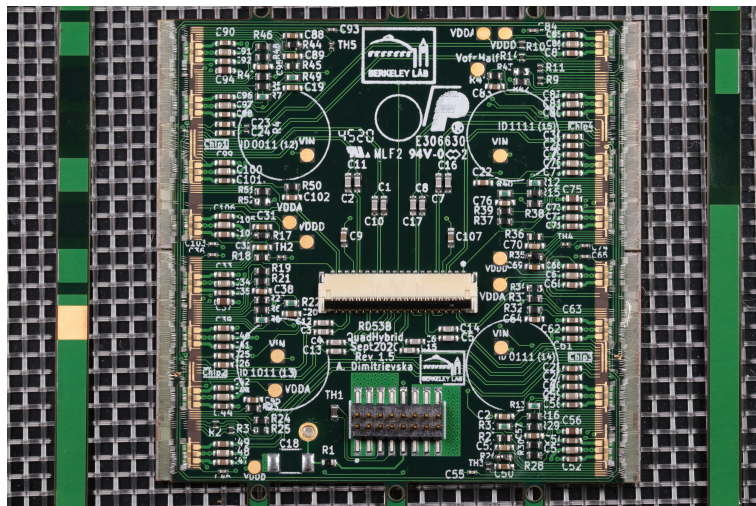


図 2.5: ITkPixV1 全体図。読み出しピクセル数は 400×384 。RD53A[4] は Synchronous FE、Linear FE、Differential FE と 3 つの領域に分けられていた。ITkPixV1 はこの Differential FE をベースとした ASIC である。

表 2.2 に主な性能を示す。

表 2.2: ITkPix の性能 [2]

チップサイズ	$20 \times 21 \text{ mm}^2$
ピクセルサイズ	$50 \times 50 \mu\text{m}^2$
ヒットレート	3 GHz/cm^2
トリガーレート	1 MHz
latency	$12.8 \mu\text{s}$
データ速度	5.12 Gbit/s
放射線耐性	500 Mrad

図 2.6 にピクセルモジュールでの信号伝達の概念図を示す。

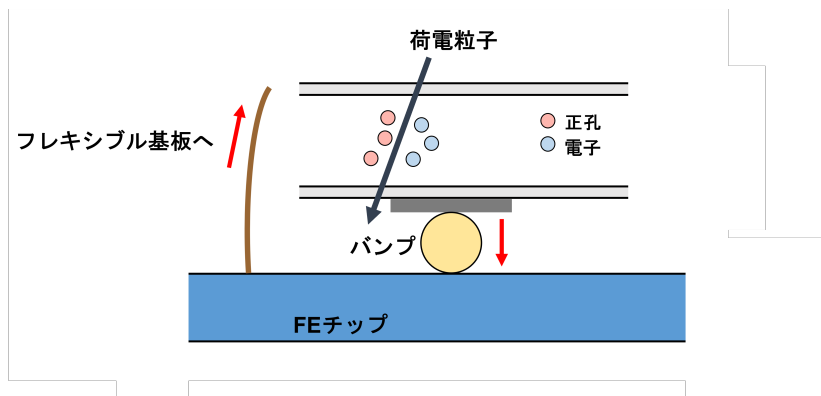


図 2.6: ピクセルモジュールにおけるデータの信号伝達の概念図。

2.1.3 フレキシブル基板 (FPC)

フレキシブル基板は表面実装など導電性の金属を取り付けた機材に電気回路を形成した基板のことである。ここではFEチップからのデジタル信号をDAQシステムに転送する他、電源供給や温度監視などの役割を担う。

2.2 ITk シリコンピクセルモジュール

2.2.1 モジュールの種類

ITk に使用されるモジュールは図 2.7 のように以下の 3 種である。

- Quad モジュール

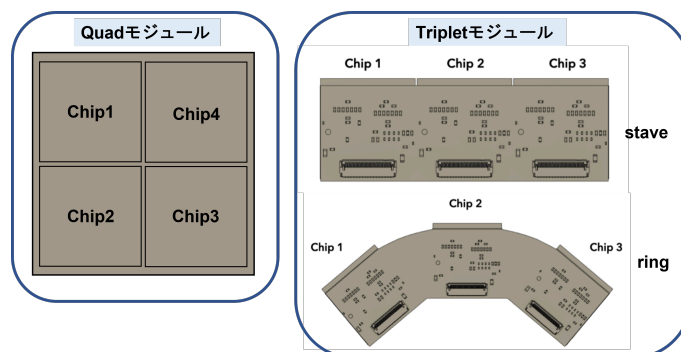


図 2.7: ITk におけるモジュールの種類

- Triplet stave モジュール
- Triplet ring モジュール

Quad モジュールはFEチップが4枚とセンサーが1枚、triplet モジュールはFEチップが3枚とセンサー3枚から成る。最内層 (L0) には Triplet stave, ring が、その他の層である検出器の大部分 (L1~L4) には Quad モジュールが搭載される。

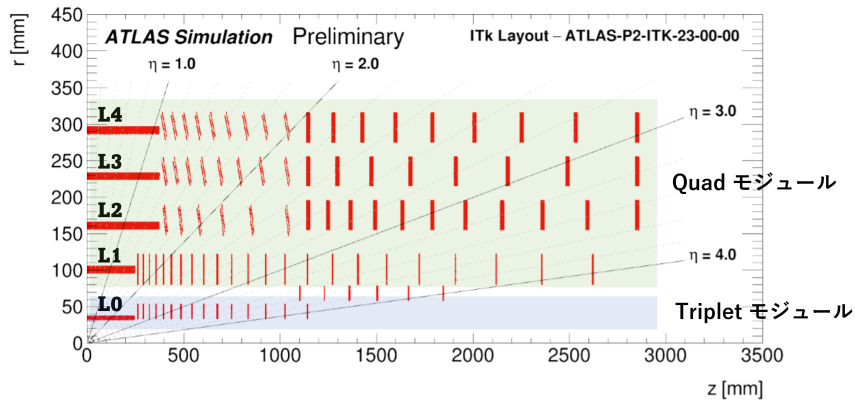
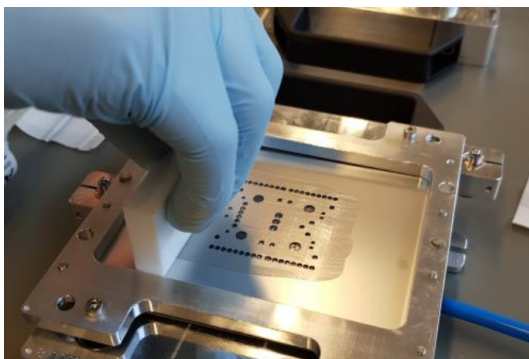


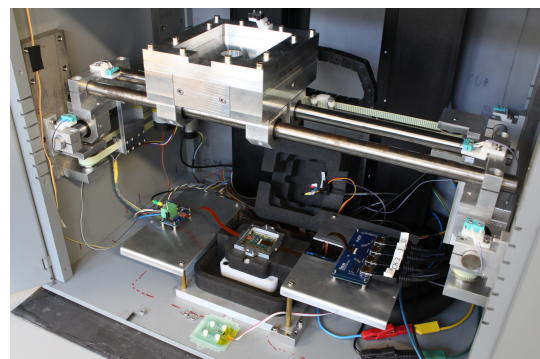
図 2.8: モジュールの各配置。Triplet の stave, ring モジュールは最内層の直線部分と円盤部分それぞれに、Quad モジュールはそれ以外の領域に使われる。

2.2.2 モジュールの量産

ITk へのアップグレードに向けて、世界各国の機関で約 10,000 個のモジュールが作られる予定であり、日本では約 2,000 個の Quad モジュールの生産を予定している。モジュールの量産に当たって、組み立てとそれに伴う品質試験が行われる。組み立ては何段階かに分けて行われ、その段階ごとに何項目かの品質試験があり、試験結果は膨大なものとなる。測定された品質試験のデータはデータベースで管理さ



(a)



(b)

図 2.9: モジュールの組み立てと品質試験を行っている様子 [5]。(a) フレキシブル基板に接着剤を塗布しているところ。誰が行っても接着剤が同じ量や形で均一になるようステンシルを介して塗布される。(b) フレキシブル基板と読み出しチップが組み立てされたものを、専用の顕微鏡で機械的特性を測っているところ。

れ、また試験は世界各国の機関で行われるためデータは正しく適切に登録・保存される必要がある。品質試験について詳細を 4.1 章で述べる。本研究では、膨大なテストデータを誰でも簡単にデータベースへ登録できるよう、ユーザビリティを高めることを目指しアプリケーションの改良を行った。このデータベースについて 4 章に記し、アプリケーションのアップデートについて 5 章で説明する。

第3章 品質試験

3.1 ピクセルモジュール組み立て手順

2.2.2 項より、ピクセルモジュールを量産されるにあたって ATLAS に参加している世界各国の機関で同じ工程によって組み立てられる。図 3.1 にピクセルモジュールの組立の工程と、ステージごとに行われる品質試験を表した。まずフレキシブル基板をベアモジュールに取り付けるところから始まる。組立

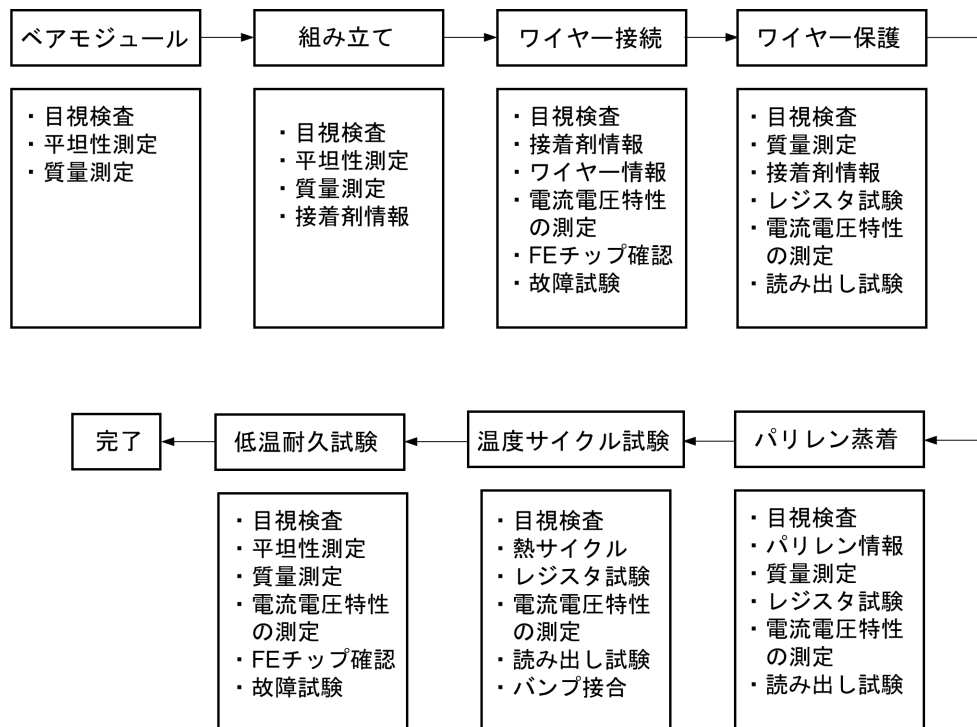


図 3.1: モジュール組立工程

後、ワイヤー配線が行われ、FEチップとフレキシブル基板が電氣的に接続される。断線などを防ぐため配線したワイヤーを保護し、モジュール自体にもパリレン高分子を蒸着させ保護する。組立としてはここで完了し、後に熱サイクル試験と低温耐久試験が行われる。HL-LHCにおける ATLAS 実験では検出器の環境温度が -55°C ~ 60°C まで変化するとされるためである。いくつかの例外を除き、基本的に電氣的な品質試験についてはワイヤー接続されるまで行われず、異なる温度での電氣的な評価試験は、すべての工程を終えた後、設置される前に再度行われる。

3.2 品質試験の種類

主な品質試験について以下に示す。

3.2.1 目視検査

図 3.2 のようなモジュールの写真を撮影し、表面実装など損傷やワイヤーの断線がないか目視で確認する。

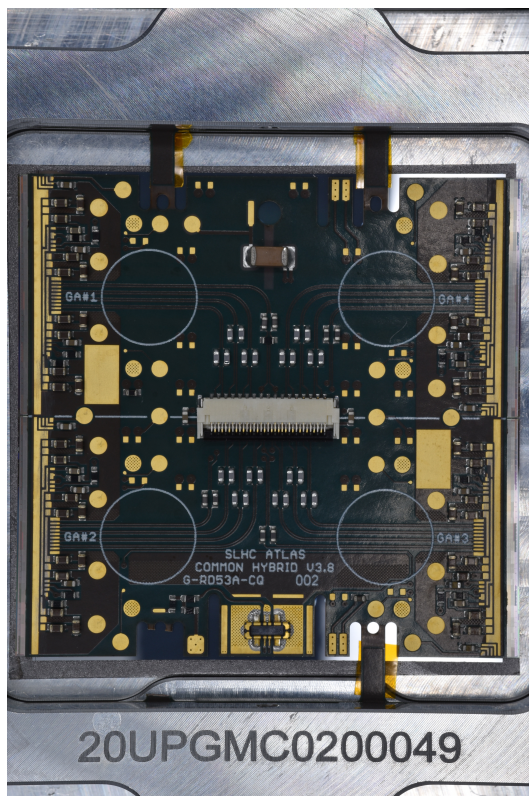


図 3.2: 目視検査用に撮影されたモジュール写真

3.2.2 質量測定

モジュールの質量を測定する。

3.2.3 平坦性測定

顕微鏡を用いた専用の装置 (図 3.3) を用いてモジュールの位置座標を何点か測定し、そこから平坦度を測り、厚さや歪みなどを確認する。

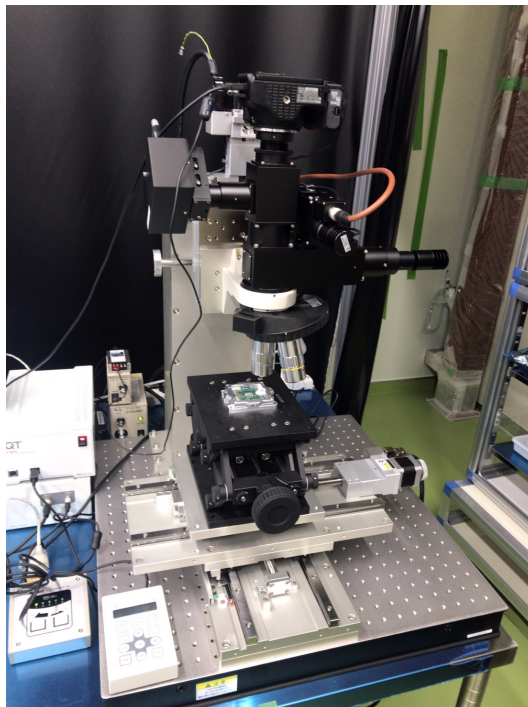


図 3.3: 平坦性測定の装置

平坦性測定における主な測定項目は次のとおりである。

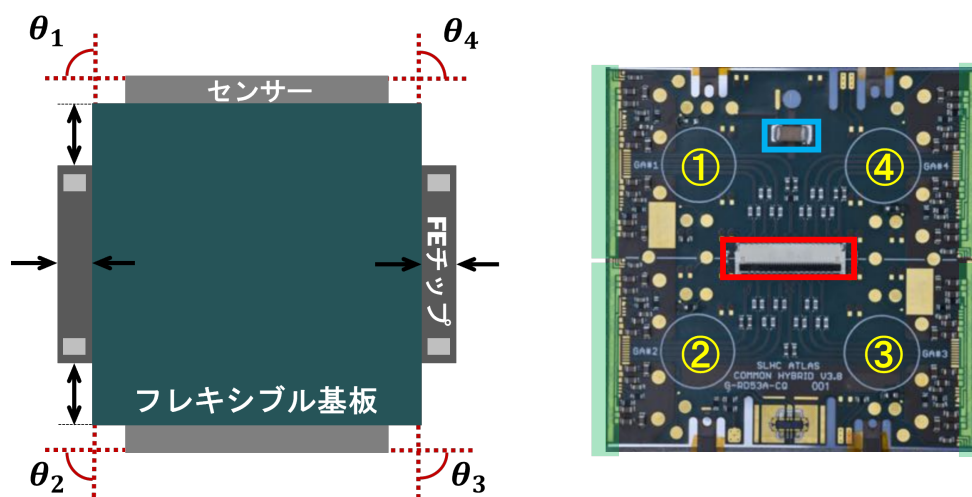


図 3.4: 平坦性測定における測定項目

- ・フレキシブル基板上辺から FE チップまでの距離
- ・フレキシブル基板左辺から FE チップまでの距離
- ・フレキシブル基板右辺から FE チップまでの距離
- ・フレキシブル基板下辺から FE チップまでの距離
- ・モジュール上部左側の角度 θ_1
- ・モジュール下部左側の角度 θ_2
- ・モジュール下部右側の角度 θ_3
- ・モジュール上部右側の角度 θ_4
- ・各 FE チップのピックアップポイント①~④の高さ
- ・フレキシブル基板のエッジとのりの厚さ (緑領域)
- ・HV capacitor の厚さ (青枠)
- ・Data connector の厚さ (赤枠)
- ・モジュール全体のたわみ (真空 on/off)

3.2.4 センサー電流-電圧特性測定

逆バイアス電圧をかけ、センサーの損傷や電流電圧特性を確認する。-15 °C、20 °C、30 °Cでそれぞれ行う。

3.2.5 FE チップ電流-電圧特性測定

FE チップが動作し始める温度を記録したのち、30 °Cで電流電圧特性を測定する。

第4章 品質管理データベース

4.1 品質試験におけるデータ管理

ITk に向けて世界各国の機関でおよそ 10,000 個のモジュールが作られる予定であり、1つのモジュールに対して章のような品質試験 [6] が行われる。それらの試験結果は正しく管理されなければならない。ATLAS 日本グループにおいては LocalDB [7] というデータベースが研究開発されており、本研究ではその LocalDB に試験結果を登録するための QChelper という GUI アプリケーションの開発状況について記述する。

4.2 データベースと関連のソフトウェア

図 4.1 にデータベースの構成概念図を示す。

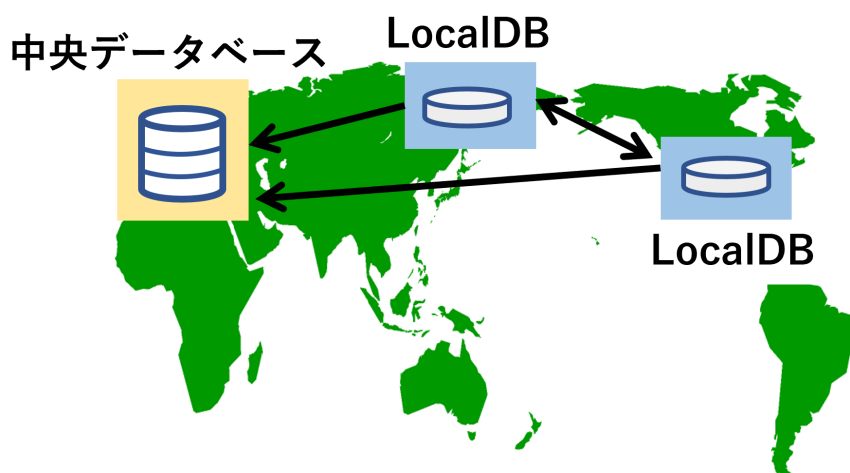


図 4.1: データベースの概念図。世界各国の機関それぞれ LocalDB においてデータが管理され、最終的に中央データベースでデータが管理される。

図 4.2 に各データベースとソフトウェアの関係を示す。

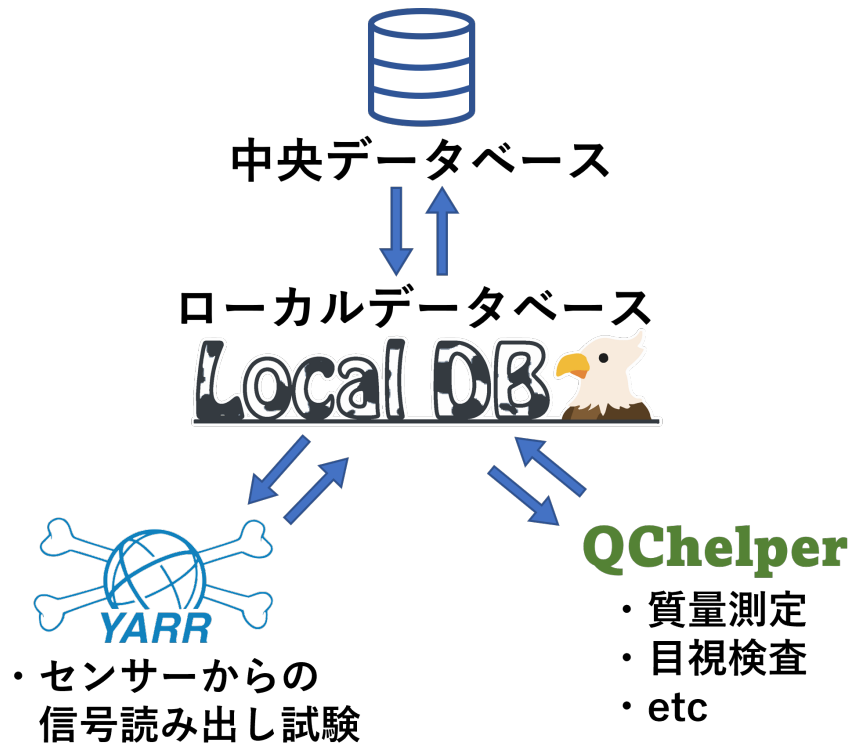


図 4.2: データベースと関連のソフトウェアの関係。品質試験を行った後、YARR や QChelper を用いて LocalDB に登録する。LocalDB に登録された結果は中央データベースに全て記録されなければならないため、中央データベースと LocalDB で同期することができる。

4.2.1 中央データベース

中央データベースはチェコに設置され、ユニコーン大学が研究開発を行っている。ITk を開発するうえで、上記のような品質試験の結果などはすべてここに保存・管理される。中央データベースに集められた結果からモジュールの解析を行い、選別することでモジュールの品質を保つ。

4.2.2 ローカルデータベース

ローカルデータベースは東京工業大学が LocalDB[7] と呼ばれるものを研究開発している。上述したように、品質試験は各組み立て工程で複数存在し、また各国の組み立て機関によって行われる。そのデータ管理は正しくかつ円滑になされるべきであり、それをサポートしているのがローカルデータベースである。オープンソースサービスの MongoDB[8] を用いたウェブアプリケーションとして開発されている。登録した試験結果を図 4.3 のようにブラウザから確認することができる。

Top Page > Component List > Component > QC
Result

Component: 20UPGM20021013 Component page of ITkPD

Current Stage: MODULETOPCB

Information

Item	Value
Serial Number	
Component Type	module
FE type	RDS3A
Children	

» TestType:MASS

Result of Mass Measurement

Scan

Key	Data
component id	
current Stage	MODULETOPCB
test Type	MASS
Institute	
user	

Result

Key	Data
Scale_accuracy	0.004
mass_value	2.94
mass_unit	g
comment	

図 4.3: LocalDB のブラウザ

4.2.3 YARR(Yet Another Rapid Readout)

モジュール用の読み出しDAQソフトウェアシステム[9]である。YARRを用いて各ピクセルのThresholdを測定するThreshold測定や、ノイズの頻度を見るノイズ占有率測定などを行う。

4.3 QChelper

QChelper は前述した LocalDB に QC test の非電氣的な試験結果をアップロードするための、python と PyQt[10] という GUI ツールキットを用いて作られたアプリケーションである。大阪大学によって開発されていた [11][12]。非電氣的な試験とは、具体的には YARR を用いない (データ読み出しに関係のない) 試験のことである。基本的にこれを使わない試験以外のデータベースへのアップロードについては QChelper が担い、アップローダーとしての役割を持つ。現在サポートしている試験項目は表 4.1 のとおりである。試験結果をアップロードする他、ベアモジュールの情報も登録することができる。

表 4.1: QChelper がサポートしている試験項目一覧

質量測定	パリレン情報
外観検査	センサー電流電圧特性
平坦性測定	FE チップ電流電圧特性
ワイヤー接続情報	RD53A チップ情報
ワイヤー接続試験	Ireftrim
接着剤情報	PCBbare orientation
熱サイクル試験	

QChelper の構造は図 4.4、4.5 のようになっている。

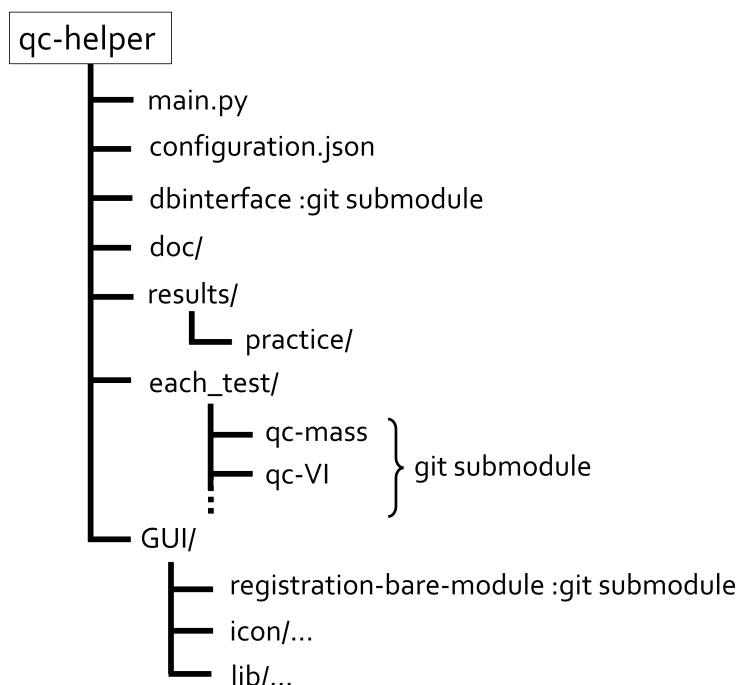


図 4.4: 先行研究によって設計された QChelper の構造。main.py がメインのコードである。configuration.json にテスト項目の一覧が設定され、each_test フォルダの中には各テストの GUI が git の submodule として存在する。

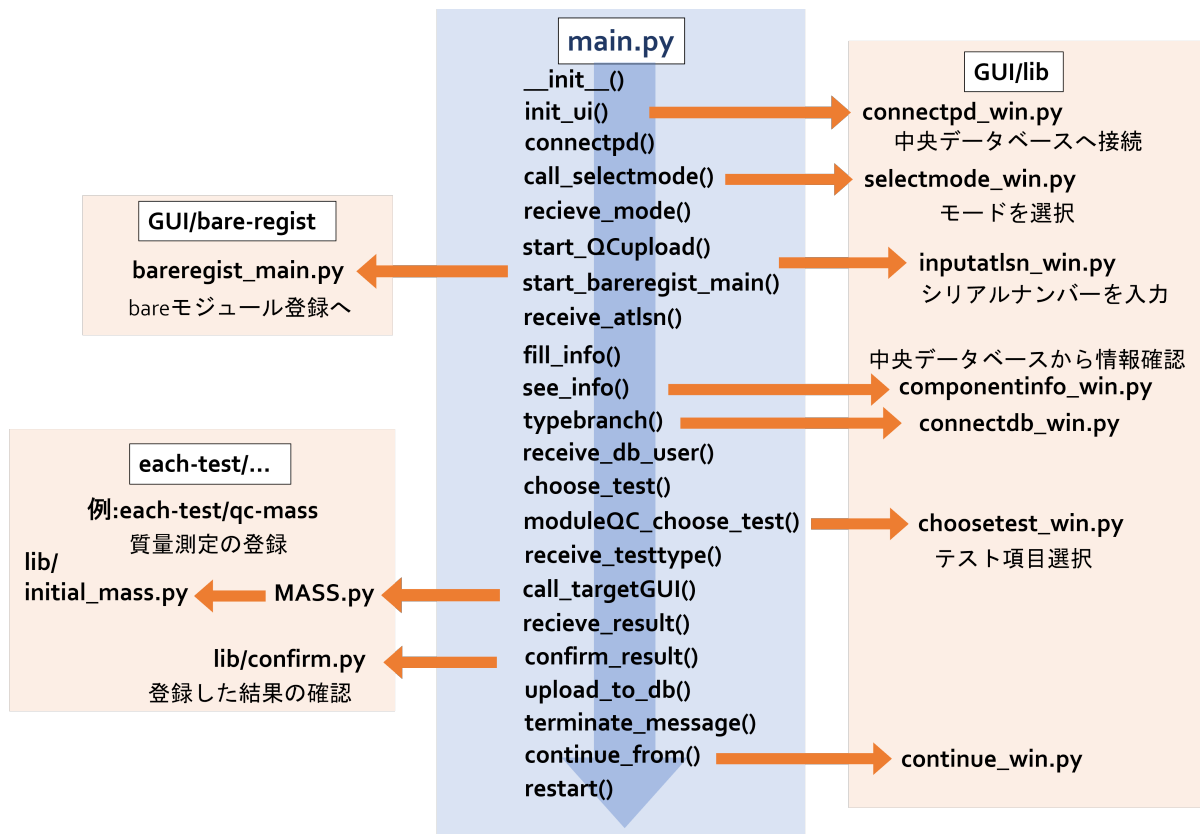


図 4.5: 先行研究によって設計された QChelper の流れ。main.py を軸として各モジュールへ接続される。

4.3.1 テスト結果登録

まず主機能である試験結果の登録について記す。最初に QChelper を立ち上げると、Production DB へのログイン画面が開かれる。そこで試験結果のアップロードか、ベアモジュールの登録の2つの選択肢が現れ、アップロードを選択する。次に登録したいモジュールのシリアルナンバーを打ち込むと、そのモジュール情報を LocalDB から参照し確認することができる。モジュールがどのテストステージでデータをアップロードしなければならないかによって、テストの項目は異なる。各テストのステージであげるべきテスト項目については、図 3.1 と同じである。そこから各テスト項目を選択し、データを打ち込み、もしくはファイルをアップすると入力した結果の確認画面が現れる。右下の upload ボタンを押せば完了である。LocalDB に結果がアップロードされる。

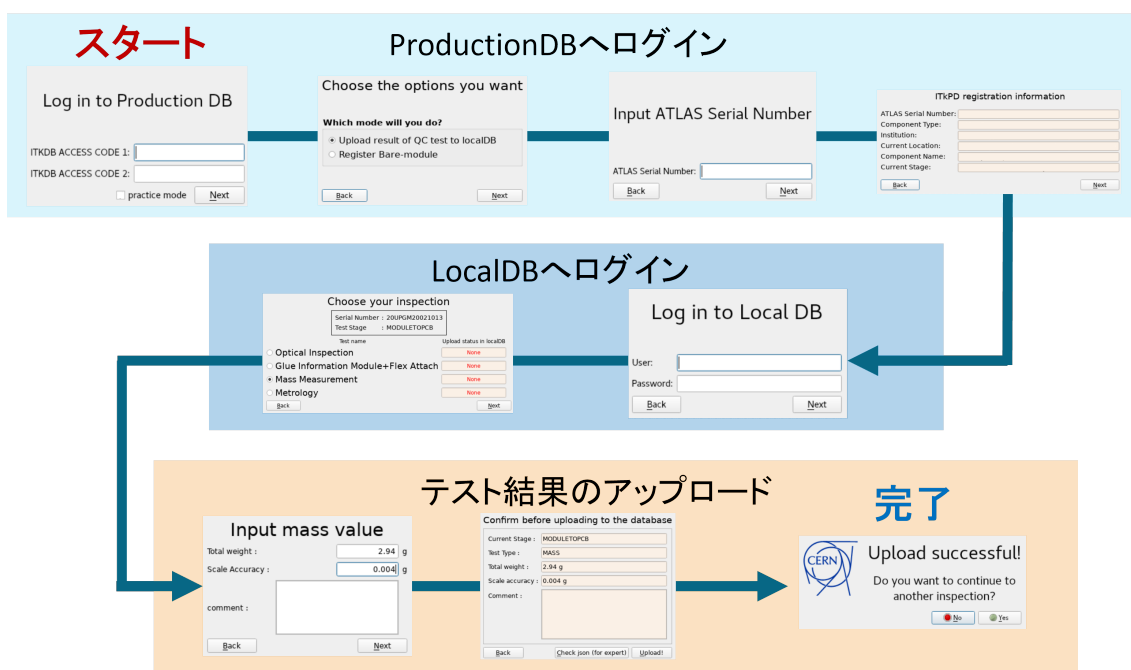


図 4.6: テスト結果のアップロード工程

また、図 4.7 に示すように、各ステージにおいて各テスト項目の LocalDB への登録状況を確認することができる。

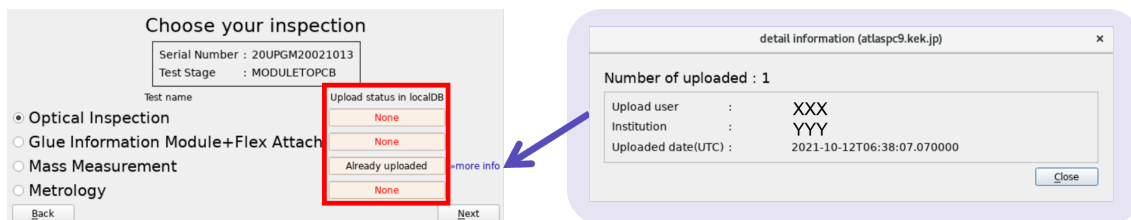


図 4.7: テストの登録状況が確認できる GUI。赤枠内においてテストの LocalDB への登録状況がわかる。登録済みのものについては Already uploaded, 未登録のものについては None と表示される。また、登録済みのものについては横の more info ボタンから登録したユーザー、場所、日時を確認できる。

実際に KEK で品質試験の結果をいくつかアップロードした。QChelper の各試験結果アップロード機能とともに、その報告を以下に記す。アップロードしたデータは図 3.1 における組み立てとワイヤー接続の各項目である。

- ・質量測定

質量測定の試験結果の数値を入力する。

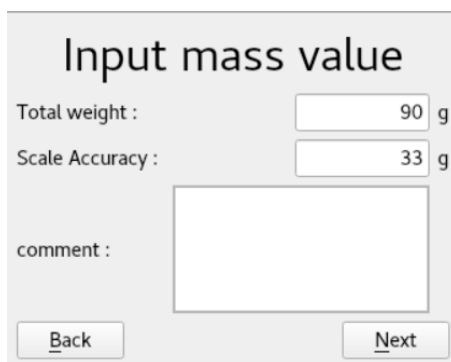


図 4.8: 質量測定の入力画面 GUI

- ・モジュール情報

各 FE チップの抵抗値を選択する。

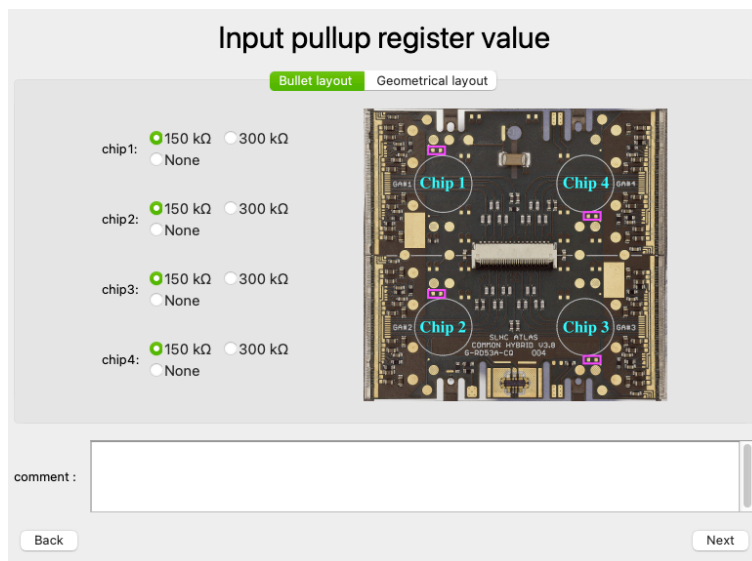


図 4.9: モジュール情報登録 GUI。

- ・目視検査

外観検査は GUI 上で行い、その結果を登録する。まずユーザーが試験結果画像を入れると、図 4.11 のように画像が 36 分割される。この 36 分割された画像について 1 つずつ異常がないか見ていく。

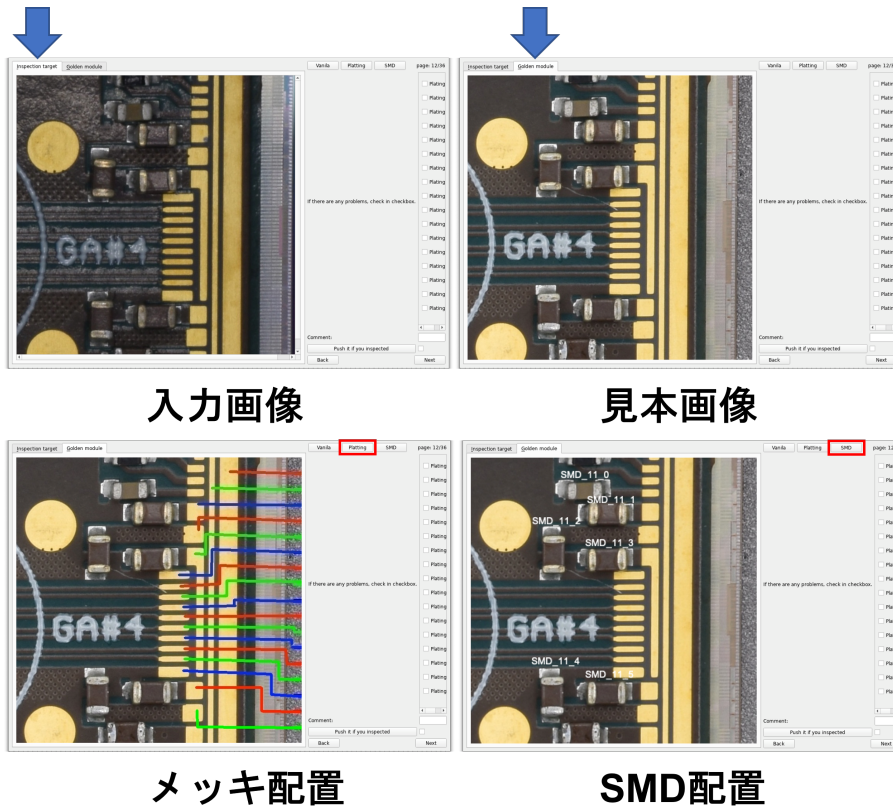


図 4.10: 外観検査の GUI。入力したファイルと見本のモジュール (golden module) がタブで切り替えられる。また、右上にある SMD、Plating などで各配置を確認でき、以上があれば配置場所の番号にチェックを入れる。

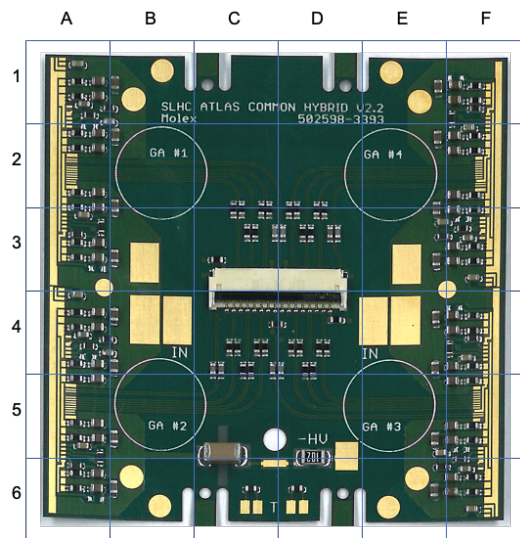


図 4.11: 外観検査における画像のマッピング

各ページの GUI は図 4.10 のようになっている。見本のモジュールと見比べ、損傷や断線、汚れなどがあつた場合は右のチェックボックスに該当の場所のチェックボックスにチェックを入れていく。

- ・ワイヤー接続情報
- ・ワイヤー接続試験
- ・接着剤情報
- ・熱サイクル試験
- ・パリレン情報

上記の項目は図 4.12 のように必要な情報を入力していく。

Wirebonding information

Input Wirebonding information

Machine used :

Operator Name :

Institution of Operator :

Bond wire batch :

Bonding jig :

Bond program :

Room temperature : degC

Humidity : %RH

comment :

図 4.12: ワイヤー接続情報 GUI。

・平坦性測定

4.13 のような必要な項目を入力していく。

The screenshot shows a window titled "Metrology" with a sub-header "Input measurement values". The interface contains a list of input fields for various parameters:

- Distance top:
- Distance left:
- Distance right:
- Distance bottom:
- Angle top-left:
- Angle top-right:
- Angle bottom-left:
- Angle bottom-right:
- Thickness pickup area chip 1:
- Thickness pickup area chip 2:
- Thickness pickup area chip 3:
- Thickness pickup area chip 4:
- Thickness edge chip 1:
- Thickness edge chip 2:
- Thickness edge chip 3:
- Thickness edge chip 4:
- Thickness HV capacitor:
- Thickness Data connector:
- Planarity vacuum on:
- Planarity vacuum on std. dev.:
- Planarity vacuum off:
- Planarity vacuum off std. dev.:

Below the input fields is a "comment :" label followed by a text area. At the bottom, there are three buttons: "Back", "Input File", and "Next". The status bar at the very bottom shows "Current user : practice" and "practice mode".

図 4.13: 平坦性測定項目入力画面 GUI

・センサー電流電圧特性

ファイルを入力し進むとプロットが確認できる。

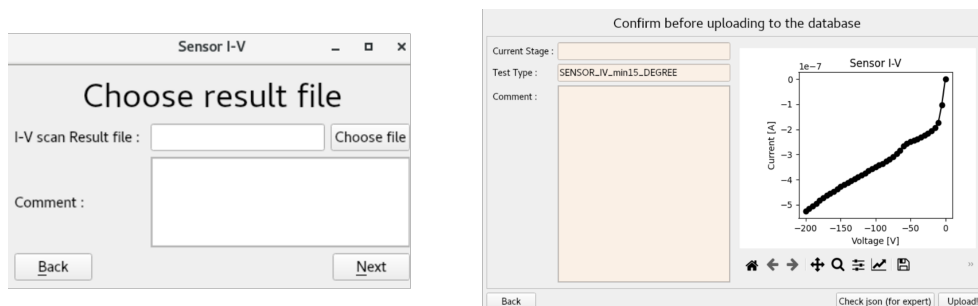


図 4.14: センサー電流電圧特性 GUI。

・FE チップ電流電圧特性

FE チップが動作した温度とともにプロットデータを登録する。

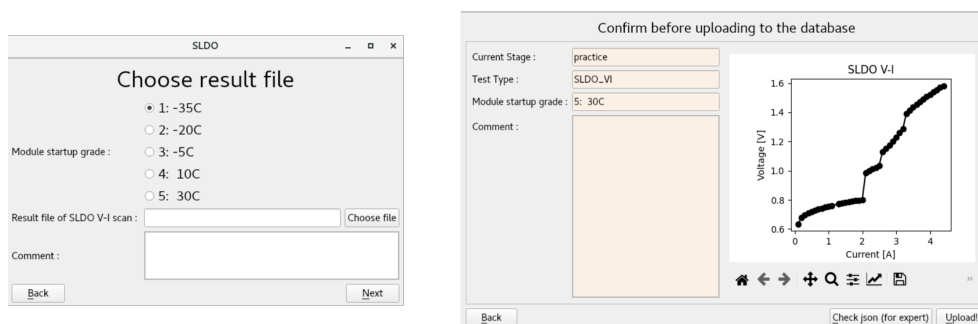


図 4.15: FE チップ電流電圧特性 GUI。

4.3.2 ベアモジュールの登録

図 2.4 より、ベアモジュールはセンサーと FE チップをバンプ接続したものを指す。LocalDB で新しいベアモジュールを登録した後、QChelper で FE チップやセンサーの種類を選択し組み立て、ベアモジュールの詳細を中央データベースに登録することができる。ベアモジュールの登録は図 4.16 の手順で行う。

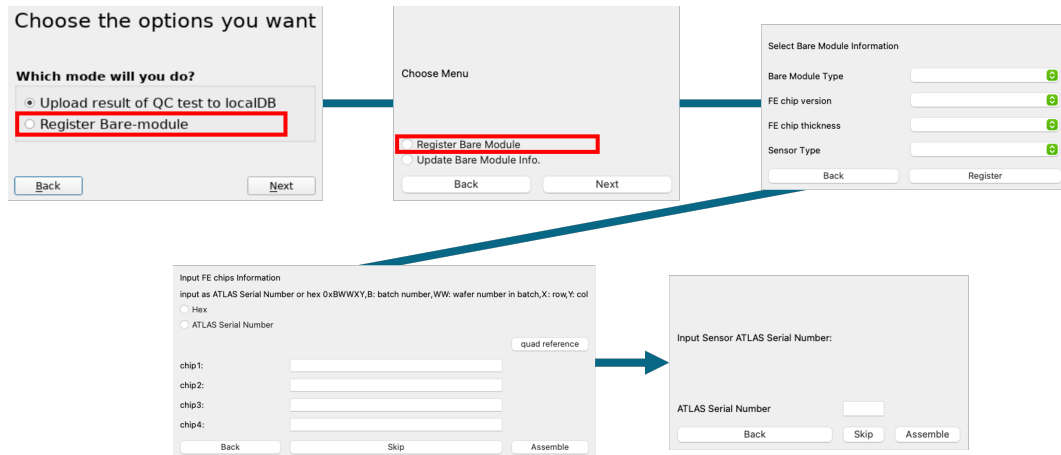


図 4.16: ベアモジュール登録の工程。

register bare module を選択した後、ベアモジュールの種類や FE チップのバージョンなど情報を追加する。その後、FE チップやセンサーをベアモジュールとして組み立てて完了である。

第5章 QChelperの改良

5.1 平坦性測定項目入力 GUI の改良

平坦性測定項目を入力する際、今までは図 4.13 のように 1 つ 1 つ入力していく仕様であった。測定項目は前述にあったように専用の顕微鏡で測定したのち、結果は JSON (JavaScript Object Notation) ファイルとしてユーザーによって編集してもらおう。JSON ファイルとは JavaScript で定義されるオブジェクト表記法の 1 つである。JavaScript に限らず主要な言語で書くことができ、5.1 のように名前とその値 (テキストや数字等) で簡単に表記できる。

Listing 5.1: JSON ファイル

```
1  {
2    "key": "value",
3    "key_num": 123
4  }
```

この JSON ファイルをそのまま入力できるよう、図 5.1 のような形でアップデートした。

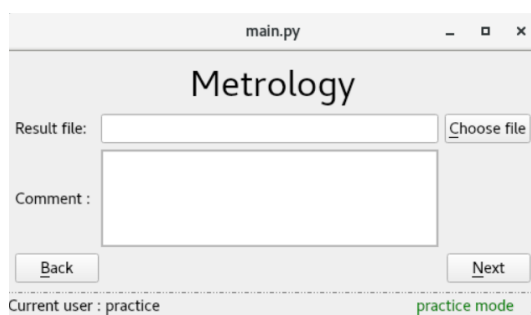


図 5.1: 改良した平坦性測定項目の入力画面

平坦性測定項目のソースコードの構造を図 5.2 に示す。今回のアップデートするにあたり各モジュールについてコードの追加や書き換えを行った。

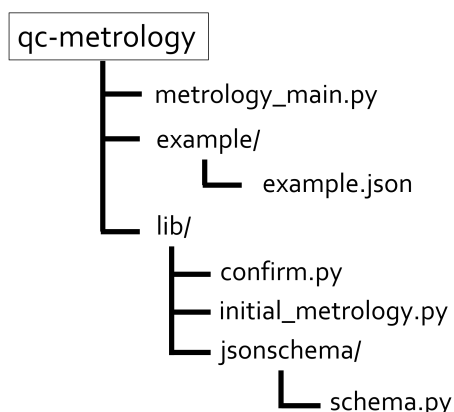


図 5.2: 平坦性測定項目のソースコードの構造

またソースコードの流れは図 5.3 の通りである。

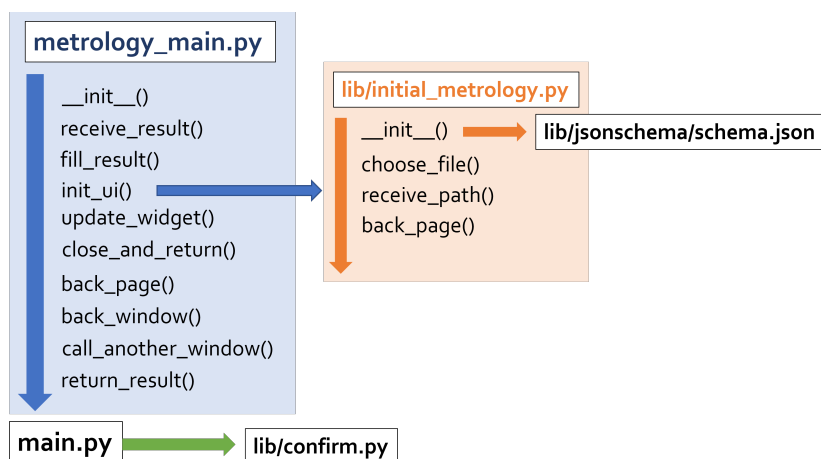


図 5.3: 平坦性測定項目のソースコードの流れ

平坦性測定項目が選択された後、まず metrology_main.py が開かれる。__init__() においては平坦性測定の各測定項目を辞書として定義する。次に initial_metrology.py へ移る。__init__() では GUI のレイアウトと jsonschema のファイルが設定してある。ここで jsonschema について簡潔に記す。jsonschema とは、入力した JSON ファイルの内部構造を定義し、検証してくれるものである。key と value で定義される JSON ファイルについて、同じように key と value をそれぞれ定義したファイルをあらかじめ作成し、jsonschema のファイルを参照して入力したファイルの中身の確認を行ってくれるものである。choose_file() で入力するファイルの設定をし、受け付けたファイル名をテキストとして記録し GUI 上に表示させる。ファイルが入力されずに次へ進もうとした場合などは警告表示 'Please choose correct file' が表示される。入力されたファイルについて receive_path() で jsonschema を用いて検証が行われる。この検証で jsonschema による必須項目がファイルに無かった場合は 'JSON File format not correct. Please check the message on CUI.' と警告が出る。例えば項目の 1 つ 'PCB to sensor edge distance top' の値が数字ではなくテキストだった場合、5.2 のようなエラーが jsonschema の持つシステムとして CUI 上に表

示される。

Listing 5.2: jsonschema によるエラーの例

```
1  Gtk-Message: 18:17:45.859: GtkDialog mapped without a transient parent. This is
    discouraged.
2  Traceback (most recent call last):
3    File "./each_test/qc-metrology/./lib/initial_metrology.py", line 109, in
      recieve_path
4      jsonschema.validate(result_dict,jsonschema_dict)
5    File "/home/oshima/.local/lib/python3.6/site-packages/jsonschema/validators.py
      ", line 934, in validate
6      raise error
7    jsonschema.exceptions.ValidationError: 'XXXXX' is not of type 'number'
8
9    Failed validating 'type' in schema['properties']['PCB to sensor edge distance
      top']:
10     {'type': 'number'}
11
12    On instance['PCB to sensor edge distance top']:
13     'XXXXX'
```

設定した jsonschema においてファイルの中身の ['PCB to sensor edge distance top'] の 'type' は 'number' でなければならないが、実際には 'XXXXX' とテキストの値が入っていることが指摘される。存在するファイルしか選べないシステムであり、考えられるエラーは jsonschema に引っかかった場合のみであるが、他何かしら不具合があった場合などは 'Unexpected Error' と出る。

また、JSON ファイルは各国形式が違うものであることを踏まえ、QChelper を使ってアップロードするにあたっては gitlab に example.json ファイルを置き、それを参照してユーザー側に編集してもらう仕様とした。JSON ファイルに必要な値が入っているよう jsonschema についても設定した。jsonschema とは入力したファイルに必要な項目が入っていないかデータ構造を検証できる JSON ファイル用のコードである。上記のファイルから設定した jsonschema は以下のとおりである。

Listing 5.3: 平坦性測定の jsonschema

```
1  {
2    "type": "object",
3    "additionalProperties": true,
4    "required": [
5      "PCB to sensor edge distance top",
6      "PCB to sensor edge distance bottom",
7      "PCB to ASIC edge distance left",
8      "PCB to ASIC edge distance right",
9      "Angle between PCB edge and sensor edge (top/left)",
10     "Angle between PCB edge and sensor edge (top/right)",
11     "Angle between PCB edge and sensor edge (bottom/left)",
12     "Angle between PCB edge and sensor edge (bottom/right)",
13     "Thickness pickup area chip1~4",
14     "Thickness edge chip1~4",
15     "Thickness HV capacitor",
```

```

16 "Thickness data connector",
17 "Planarity vacuum on/off",
18 "Planarity vacuum on/off std. dev."
19     ],
20     "properties" :{
21         "PCB to sensor edge distance top":{
22             "type":"number"
23         },
24         "PCB to sensor edge distance bottom":{
25             "type":"number"
26         },
27         "PCB to ASIC edge distance left":{
28             "type":"number"
29         },
30         "PCB to ASIC edge distance right":{
31             "type":"number"
32         },
33         "Angle between PCB edge and sensor edge (top/left)":{
34             "type":"number"
35         },
36         "Angle between PCB edge and sensor edge (top/right)":{
37             "type":"number"
38         },
39         "Angle between PCB edge and sensor edge (bottom/left)" :{
40             "type":"number"
41         },
42         "Angle between PCB edge and sensor edge (bottom/right)":{
43             "type":"number"
44         },
45         "Thickness pickup area chip1~4":{
46             "type":"array"
47         },
48         "Thickness edge chip1~4":{
49             "type":"array"
50         },
51         "Thickness HV capacitor":{
52             "type":"number"
53         },
54         "Thickness data connector":{
55             "type":"number"
56         },
57         "Planarity vacuum on/off":{
58             "type":"number"
59         },
60         "Planarity vacuum on/off std. dev." :{
61             "type":"number"
62         }
63     }
64 }

```

”required”で必須項目を定義し、”properties”で各項目の中身において数字を要求するものとした。
jsonschema については平坦性測定項目のほか、センサー電流電圧特性や FE チップ電流電圧特性においても JSON ファイルを扱っているため設定してある。

さらに、確認画面では各測定項目がどこに当たるのかというモジュールマップを加えた。

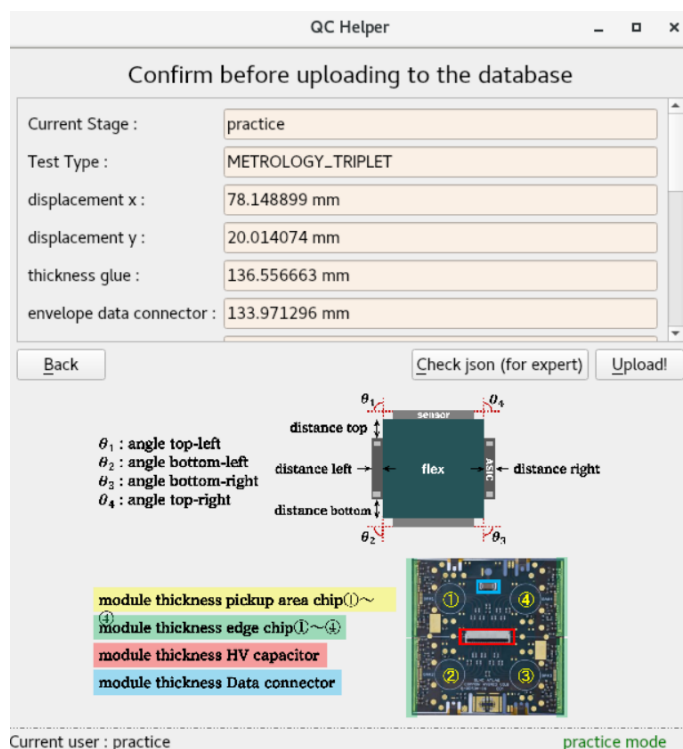
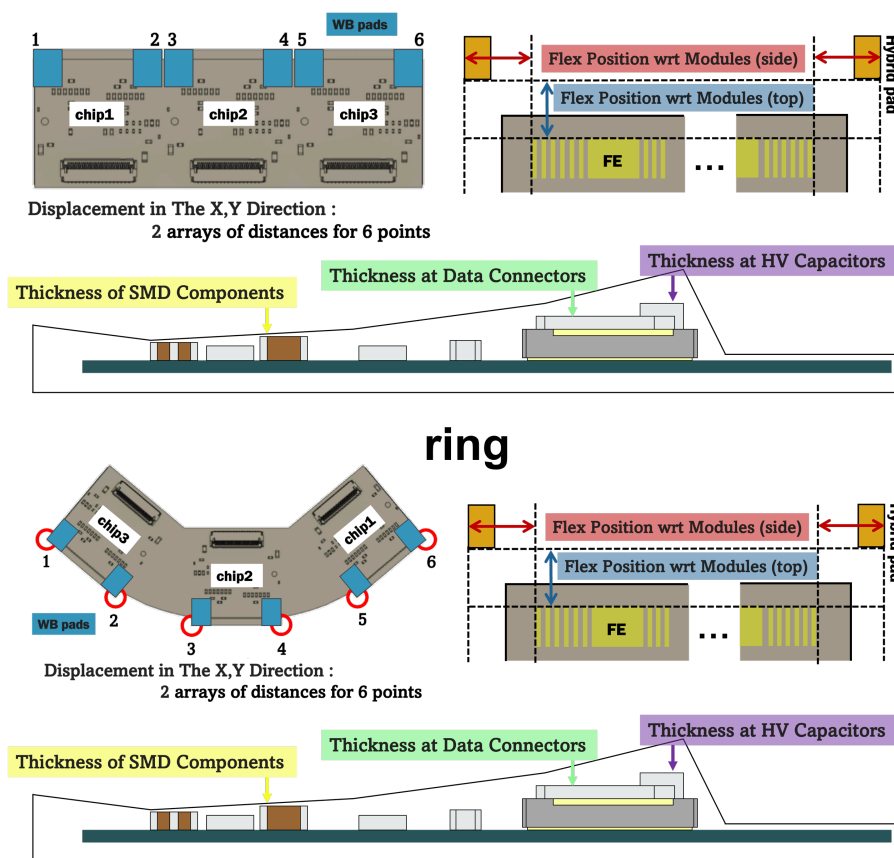


図 5.4: Quad モジュールマップを加えた確認画面

上記の平坦性測定は Quad モジュールのものであるが、新たに Triplet モジュールの平坦性測定のテストが追加されることを受け、同様に JSON ファイルから以下の項目を取り出すようにし、確認画面では Triplet モジュールのマップを作成しアップローダー機能を追加した。

stave



ring

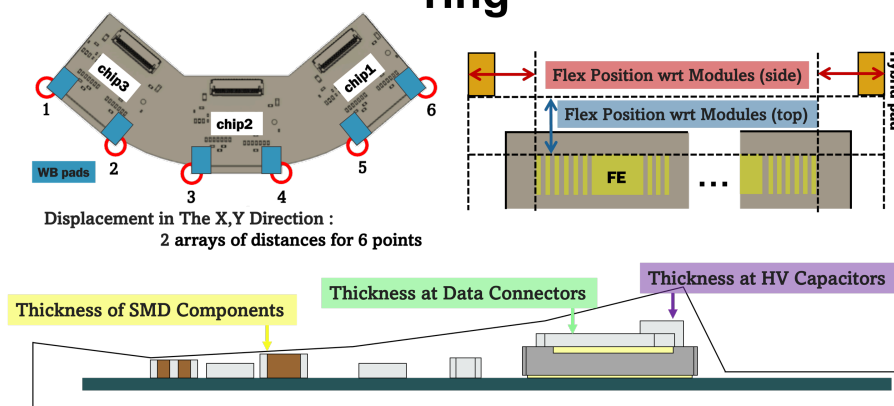


図 5.5: Triplet モジュールのマップ

Quad モジュールと Triplet モジュールは現在シリアルナンバーによって区別することができる。よって試験結果をアップロードする際の入力したシリアルナンバー [13] によって Quad か Triplet かを判断し、各平坦性測定 GUI に遷移するよう以下の関数を加え分岐するようにした。

Listing 5.4: 分岐させるため加えた関数のソースコード

```

1 def module_type(self):
2     #####
3     ##### configuration2 #####
4     SN = self.atlsn
5     ##### triplet stave #####
6     if 'MS' in SN or 'R6' in SN or 'R3' in SN:
7         print(SN + ": triplet stave module")
8         QMessageBox.warning(None, 'Error', 'UNDER DEVELOPMENT', QMessageBox.Ok)
9         self.inputatlsn_wid
10
11     ##### triplet ring #####
12     elif 'M0' in SN or 'M5' in SN or 'R7' in SN or 'R8' in SN or 'R4' in SN or
13         'R5' in SN:
14         print(SN + ": triplet ring module")
15         QMessageBox.warning(None, 'Error', 'UNDER DEVELOPMENT', QMessageBox.Ok)

```

```
15         self.inputatlsn_wid
16
17         ##### quad #####
18     else:
19         print(SN + ": quad module")
20         with open('./configuration/configuration.json') as f1:
21             self.configuration = json.load(f1)
22         self.fill_info()
23         #####
```

シリアルナンバーによってモジュールのタイプを判別した後、テスト項目一覧が書かれている configuration.json について対応するものをそれぞれ判別する。

5.2 バージョン情報のアップロード

QChelper は品質試験の項目やフォーマットの等の変化に対応していく必要があり、日々アップデートがなされる。それに伴い、データベースに試験結果をあげるアップロードする際、いつのバージョンのものでアップロードをしたのか明確であったほうが良い。バージョンによっては jsonschema のフォーマットが違い、アップロードした項目などが異なるためである。QChelper は gitlab で管理されており、バージョンの取得にあたって git のシステムを利用した。バージョン取得のために追加したコードは 5.5 の通りである。

Listing 5.5: QChelper のローカルでのバージョンを取得するソースコード

```
1 import git
2 path = './'
3 repo = git.Repo(path)
4 commit_id = repo.git.show('-s', '--format=%H')
5 tag_all = repo.git.tag('-l', '--contains', f'{commit_id}')
6 tag_list = tag_all.split()
7 tag = tag_list[0]
8 self.info_dict['version_info'] = tag
```

ここでバージョン情報の取得方法について図 5.6 を用いて説明する。

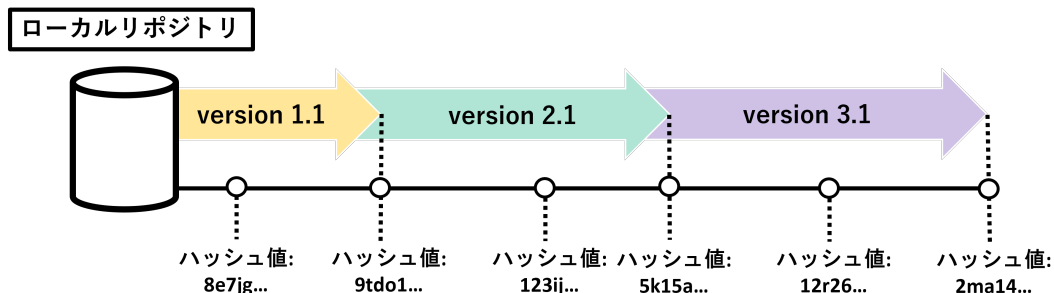


図 5.6: git におけるバージョン情報の概念図

まずユーザーの使っているローカルでの QChelper の最新のハッシュ値を取得する。図 5.6 において、例えば 9tdo1... が最新のハッシュ値だった場合を考える。次にリモートリポジトリ、つまり git で管理されている QChelper からそのハッシュ値を含むバージョンを全て取得する。ハッシュ値 9tdo1... が含まれる QChelper のバージョンは version 1.1、2.1、3.1 である。よって、複数あるバージョンの中で少なくともそれ以降のバージョンの QChelper であるということを示すため、取得したバージョンにおける一番古いものをここで用いるバージョン情報とした。例の場合、1.1 が格納されるバージョンとなる。

試験結果をデータベースに登録する際は、図 5.7 のように確認画面の JSON 形式のファイルからバージョン情報を見ることができる。バージョン情報は JSON ファイルの”dbVersion”に格納され LocalDB に保存されることとなる。

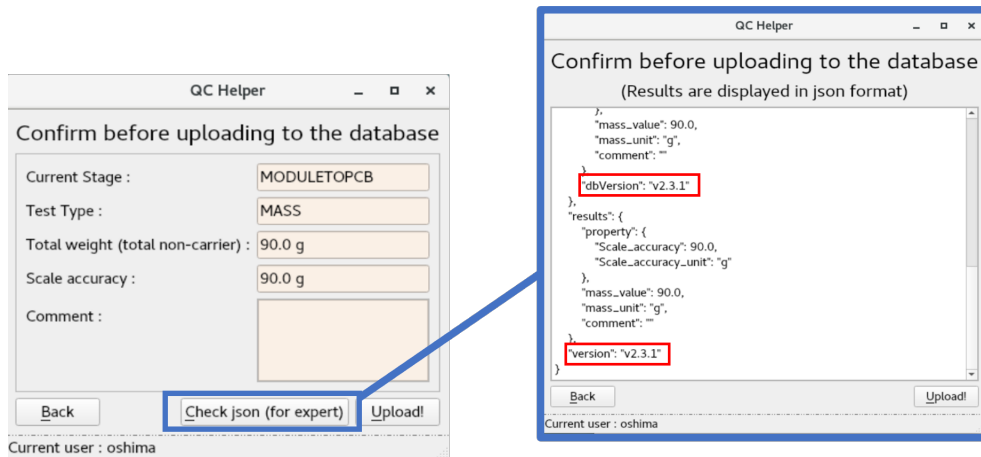


図 5.7: QC Helper のバージョン情報記載場所

保存された情報は、試験結果を登録する際のテスト項目選択画面の more info ボタンから確認できる詳細情報一覧から確認できるよう加えた。また、前述した JSON ファイルの”dbVersion”に以前は 1.01 という数値が格納されていた。ゆえに、今回のバージョン情報を登録できる以前の QC Helper で試験結果を登録したものについては”Unknown”と表示するようにした。

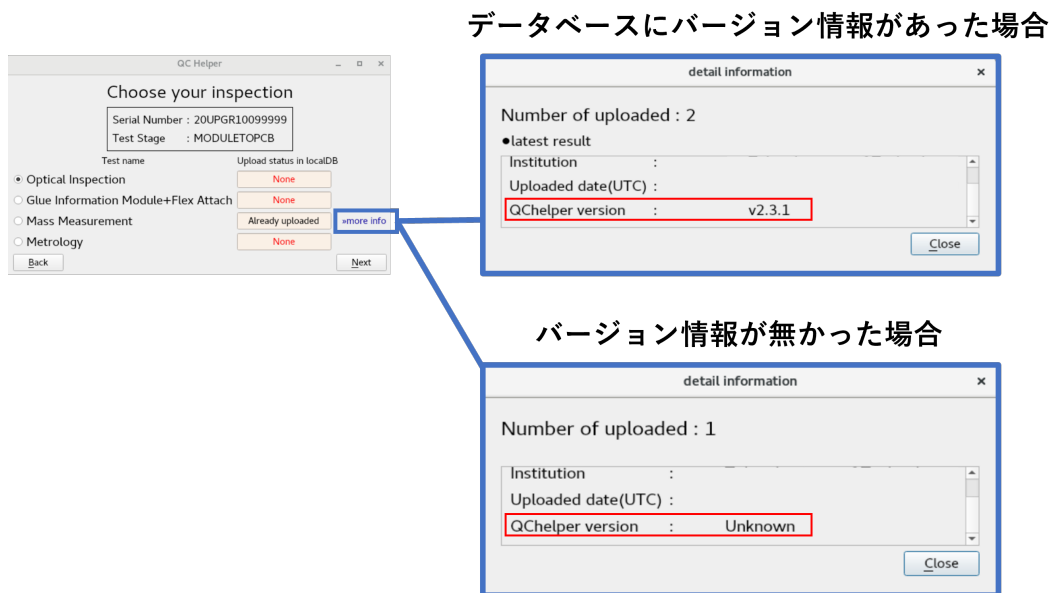


図 5.8: QC Helper のバージョン情報記載場所

第6章 まとめ

CERN の運営する LHC 中の 1 つである ATLAS 検出器は、HL-LHC へのアップグレードに向けて内部飛跡検出器を ITk へ総入れ替えを予定している。そのアップグレードに向けて世界各国の機関でおよそ 10,000 個のモジュールが量産される。モジュールの組み立ては何段階かのステージに分け細分化され、そのステージごとで品質試験が行われる。品質試験のデータ結果は各国それぞれで適切に登録・管理されなければならない。

データベースへ試験結果をアップロードする GUI が先行研究で開発されており、本研究ではさらに品質試験項目の変更やユーザビリティを高めるため GUI の改良を行った。

まず、品質試験の 1 つである平坦性測定の GUI の改良である。以前までは平坦性測定項目におけるデータベースへアップロードする項目 23 個を手動で入力する仕様であった。データが膨大になることや使いやすさを考慮し、品質試験を行った際に出てくるデータをユーザーに JSON 形式で編集してもらい、そのファイルをそのまま入力できるよう改良を行った。平坦性測定の試験の結果のフォーマットは各国機関で異なることから、サンプルのファイルを作り git へ公開するようにした。入力するデータファイルにデータベースへ保存すべき必要な項目があるか確認できる jsonschema を設定した。これにより、問題が無ければそのままアップロードできるが、ファイルに必要項目が欠けているとエラー表示によってユーザーに知らせることができる。データをアップロードする際の確認画面では、各項目がモジュールのどの部分であるか分かるモジュールマップを作成し加えた。また、平坦性項目は Quad モジュールと Triplet モジュールで測定項目が異なるが、今まで Quad のみの対応だったため Triplet のアップローダー機能も開発を行った。

次に、GUI のバージョン情報をデータベースへアップロードする機能を開発した。GUI はバージョンによって仕様が異なり、また jsonschema の内容なども異なる。ゆえに、どのバージョンの GUI を用いて試験結果がアップロードされたのかという情報が重要となる。GUI は git で管理されているため、git のシステムを利用してバージョン情報を取得し、その取得情報をデータベースへアップロードできるよう開発を行った。バージョン情報は試験結果をアップロードする際の確認画面と、アップロード後のテストの選択画面の詳細情報の 2 点で確認ができる仕様とした。

以上のことから、本研究により QChelper を用いてより効率的に使いやすく、かつ誤りなくデータベースへの試験結果のアップロードが可能になった。

現状と課題

平坦性測定項目における Quad モジュールについての GUI は実装済みであり、共に開発を行っていた大坂大学の学生と実際にローカルデータベースへアップロードし動作の確認を行った。しかし、Triplet

の品質試験はまだ行われておらず、実際のデータを用いての動作確認はまだ行うことが出来ていない。したがって今後、試験結果のデータフォーマットが変更されることも考えられ、それに対応する必要がある。実際に試験が行われた後もデータを用いてデータベースへアップロードを行い GUI の動作確認を行わなければならない。GUI のバージョン情報についても、git のシステムを用いてユーザーの用いている GUI のバージョン情報を取得するよう開発を行ったが、ユーザーがローカルで GUI への変更を行った場合を想定しておらず、今後このような場合も考えて変更を加える必要がある。また、2021 年 11 月頃 ATLAS で用いられているソフトウェアについて紹介と評価を行う批評会 (Datebasetools review) がオンラインで行われ、QChelper について発表した。そこで頂いたコメントを参考にバージョン情報のアップロード機能を開発した。今後もモジュールの組み立てと品質試験が行われていくにあたり、様々な人に利用されることが予想され、よりユーザビリティを高めていく必要がある。

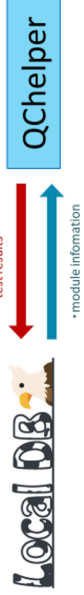
付録A Datebasetools review資料

What's QChelper

QChelper is a GUI to upload the results of non-electrical test to the LocalDB.

About [git link](#)

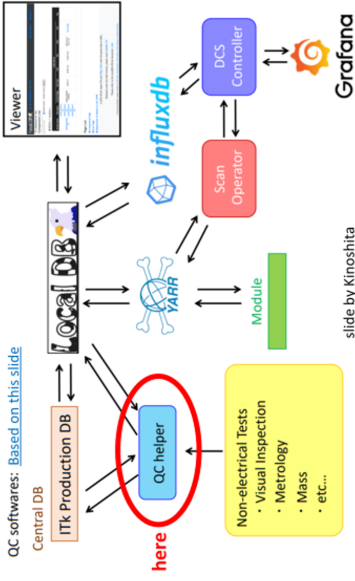
- supports module and bare module
- communicates with the Production DB and the Local DB
- registers bare modules in the Production DB
- features practice mode



2021/12/23

3

overview of software for module QC



2021/12/23

2

What QChelper can do

• uploads QC test results

QChelper is specifically the tool to upload what you need in test results other than tests using YARR. So you can upload some properties and IV curves (is electrical, however). Supported items are as follows ->

Mass	Polylene property
Optical inspection	Sensor IV
Metrology	SUDO VI
Wirebond info	RD53A pull-up register (module property)
Wirebond pull test	IrefTrim (module property)
Glue info	PCB-Bare Orientation (module property)
Thermal cycling	

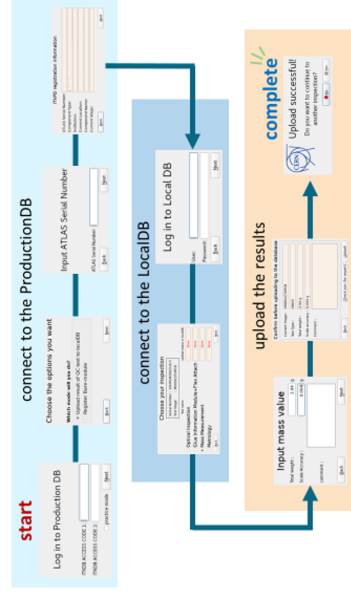
• registers bare modules in the Production DB

It was working fine, but there is currently (probably) a problem with ITKPD that is preventing registration, but we are in the process of trying to resolve it.

2021/12/23

4

Upload flow ex)mass test



2021/12/23

5

After uploaded the results

- when you upload results of other tests
- check the results with LocalDB viewer

see the upload status of the test results in LocalDB at that test stage

For uploaded results, you can see information about the user who uploaded the test, the location and the date

2021/12/23

6

GUI for QC tests

Visual Inspection

reference module result

- contents in the check list for each image tile should be discussed
- compare with the model!

Metrology

- input the measurement files **JSON as input**
- validate input files with jsonschema

SIBO.VI

- input the measurement files **JSON as input**
- validate input files with jsonschema

Sensor IV

- input the measurement files **JSON as input**
- validate input files with jsonschema
- and so on...

2021/12/23

7

GUI for QC tests

The type of file input is determined by the format of the file. Please refer to the example file in the link.

Metrology

- input the measurement files **JSON as input**
- validate input files with jsonschema

SIBO.VI

- input the measurement files **JSON as input**
- validate input files with jsonschema

Sensor IV

- input the measurement files **JSON as input**
- validate input files with jsonschema
- and so on...

2021/12/23

8

practice mode

start

in practice mode

data are not saved

2021/12/23

9

付録B 平坦性測定で使われるJSONファイル

JSON(JavaScript Object Notation) は本文でも述べたように、拡張子.json とされるテキスト形式のデータフォーマットである。JavaScript とあるが、使うことのできる言語は JavaScript に限らず様々な言語で利用される。JSON を構成するデータ型は文字列、数値、ブール、null、オブジェクト、配列と非常にシンプルで使いやすく軽量で処理も高速であり、開発者に好まれるデータファイルである。基本的に”key”とそれに対する”value”で構成され、誰でも簡単に作ることができる。

B.1 に平坦性測定試験で用いられている JSON ファイルを示す。ファイルの中身の形式はお茶の水大学の河野能知先生によって作られたものである。

Listing B.1: 平坦性測定 example.json ファイル

```
1 {
2   "moduleName": "MODULE NAME",
3   "PCB to sensor edge distance top": 0.376,
4   "PCB to sensor edge distance bottom": 0.382,
5   "PCB to ASIC edge distance left": 0.861,
6   "PCB to ASIC edge distance right": 1.121,
7   "Angle between PCB edge and sensor edge (top/left)": 89.986,
8   "Angle between PCB edge and sensor edge (top/right)": 90.023,
9   "Angle between PCB edge and sensor edge (bottom/left)": 89.984,
10  "Angle between PCB edge and sensor edge (bottom/right)": 90.020,
11  "Thickness pickup area chip1~4": [
12    0.6069499999999994,
13    0.5936499999999967,
14    0.5987899999999984,
15    0.6150099999999963
16  ],
17  "Thickness edge chip1~4": [
18    0.3558500000000002,
19    0.3753500000000001,
20    0.2761499999999959,
21    0.36254999999999704
22  ],
23  "Thickness HV capacitor": 2.2645999999999997,
24  "Thickness data connector": 1.8654499999999983,
25  "Planarity vacuum on/off": 0.015,
26  "Planarity vacuum on/off std. dev.": 0.022,
27  "Raw data (file)": "XXX.txt"
28 }
```

付録C git

git[14]とはプログラム全般におけるソースコードなどの分散型バージョン管理システムである。gitの概念図を図C.1に示す。

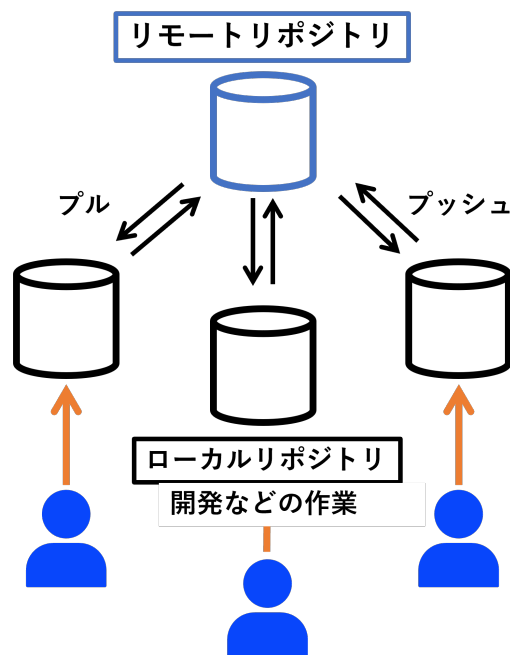


図 C.1: git の概念図

ユーザーがローカルで行った作業はローカルリポジトリで管理され、必要があればその作業内容をオリジナルであるリモートリポジトリに上げる (プッシュ) ことができる。リモートリポジトリから他の作業者が行った作業内容を自分のリポジトリに反映させたり (プル)、ソースコードをコピーしてローカルリポジトリに保存する (クローン) こともできる。また、自分の行った作業ローカルリポジトリに記録する操作をコミットと呼ぶ。コミットを実行すると作業内容や日時を記録したファイルが生成され、時系列に沿って格納されていく。コミットされるごとに図C.2のようにハッシュ値と呼ばれる16進数のIDが生成され、これによって管理される。

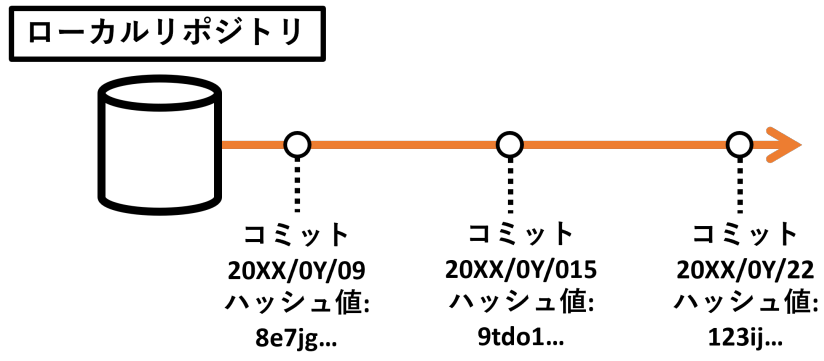


図 C.2: コミットのイメージ

参考文献

- [1] Jörn Lange. Module development for the atlas itk pixel detector, 2020.
- [2] Lingxin Meng. Atlas itk pixel detector overview, 2021.
- [3] RD53 collaboration. <https://rd53.web.cern.ch/>.
- [4] Maurice Garcia-Sciveres. The RD53A Integrated Circuit. Technical report, CERN, Geneva, Oct 2017.
- [5] Yusong Tian, Ali Skaf, Silke Mobius, Sascha Mario Bohlken, Tim Kanngiesser, Ruediger Manfred Widera, and Joern Lange. ITk Pixel Module Assembly and Testing — ATLAS Phase-2 Upgrade. General Photo, Jun 2021.
- [6] Lingxin Meng. RD53A Module Testing Document. Technical report, CERN, Geneva, Nov 2019. Latest version on gitlab: <https://gitlab.cern.ch/lmeng/rd53amoduletestingdoc>.
- [7] 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/IBO6BptySxPwD6L>, 2020 年.
- [8] MongoDB: the application data platform — MongoDB. <https://www.mongodb.com/>.
- [9] Timon Heim. YARR - a PCIe based readout concept for current and future ATLAS pixel modules. *Journal of Physics: Conference Series*, Vol. 898, p. 032053, oct 2017.
- [10] PyQt. <https://riverbankcomputing.com/software/pyqt/>.
- [11] 大西裕二. HL-LHC ATLAS 実験ピクセル検出器量産時における機械的品質試験等のためのデータベース登録システムの開発. 日本物理学会 2020 年秋季大会.
- [12] git QChelper. <https://gitlab.cern.ch/atlas-itk/sw/db/pixels/qc-viz-tools-dev/qc-helper>.
- [13] R Bates Monika Wielers. Serial Number Specification for ITk pixel modules. Technical report, CERN, Geneva, Sep 2019.
- [14] git. <https://git-scm.com/>.
- [15] 戸塚洋二. 素粒子物理. 岩波書店, 2016.

- [16] 浅井祥仁. ヒッグス粒子発見とその後の展開. *RADIOISOTOPES*, Vol. 66, No. 1, pp. 43–54, 2017.
- [17] Burkhard Schmidt. The high-luminosity upgrade of the LHC: Physics and technology challenges for the accelerator and the experiments. *Journal of Physics: Conference Series*, Vol. 706, p. 022002, apr 2016.
- [18] Claudia Gemme. *The ATLAS Tracker Detector for HL-LHC*.
- [19] Technical Design Report for the ATLAS Inner Tracker Pixel Detector. Technical report, CERN, Geneva, Sep 2017.
- [20] Localdb docs. <https://localdb-docs.readthedocs.io/en/top/>.
- [21] Francisca Munoz Sanchez. ATLAS ITk Pixel Detector Overview. Technical report, CERN, Geneva, Jun 2021.
- [22] private communication.

謝辞

コロナ禍の外部進学ということもあり、2年間で振り返っても正直つらいことのほうが多かったかと思いますが、本研究を進める上で多くの方に支えられご指導を頂いてここまでやってこれたことに深く感謝申し上げます。初めに、角野秀一先生には研究に関するアドバイスに関するほか、日常生活で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 KEK への出張から発表会、論文の添削や日頃の研究室生活でも何から何までお世話になりました。研究室で居心地よく生活でき研究を進められたのは先生方のおかげです。

ATLAS の DB 関連の研究を進めるにあたり、機会を与えてくださった廣瀬穰さん、研究だけでなく普段の研究生活にもアドバイスをくださった生田秀行さん、研究のほかスライドを作るうえでアドバイスをくださった陣内修さんにも御礼申し上げます。ソフトウェアに不慣れで他大学の学生である私にも優しく丁寧にご指導いただきありがとうございました。大変お世話になりました。また、KEK の出張の際は外川学さんにお世話になりました。

研究室では先輩方である在原さん、鶴藤さん、三宅さんにお世話になりました。三宅さんには研究の他に就活時にもお世話になりました。特に博士後期課程に進まれた在原さんとは研究の話から他愛のない話までたくさんして頂き、楽しく研究室生活を送ることができました。Thomas さん、後輩の岩城さん、古藤くん、本橋くん、北村くん、鮫島くん、澤くん、竹淵くん、楽しくお話して頂きありがとうございました。同期の安藤くん、朴くん、渡辺くんにも感謝申し上げます。皆違うグループでしたが、時には弱音を吐きあったりリラックスすることができました。

ATLAS グループでは木下くん、藤田さん、荒久田くん、いつも Skype のチャットでの質問やお願いなど快く受けてくださりありがとうございました。特に同期の木下くんにはお世話になりました。分かりやすく丁寧な説明でいつも助かっていました。また、先輩方である大西さん、調さんにもお世話になりました。違う大学の後輩であるうえ、ATLAS を去られているにもかかわらず Skype で度々分からないことや困ったことへ助言をくださった大西さん、スイスから通話をつなげてソフトウェア初心者の私に丁寧に快く色々教えてくださった調さん、感謝申し上げます。

最後に、浪人を経てから今まで、遠方から支えてくれた家族には深く感謝申し上げます。心配ばかりかけたかと思いますが、私のやりたいことをいつも応援してくれてありがとうございました。