

ぼっち開発から始めるGit

MATSUOKA Takashi

@matsujirushi12



- ✓ 事前にインストールしておくソフトウェア
- ✓ PCとWi-Fi接続
- ✓ スライド

<https://speakerdeck.com/matsujirushi/botutikai-fa-karashi-merugit>





MATSUOKA Takashi

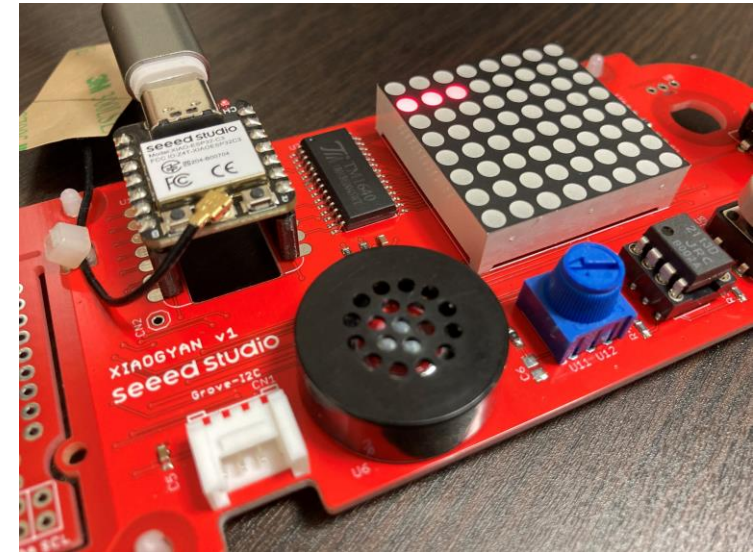
@matsujirushi12



2017～ MVP for Windows Development
2020～ for Microsoft Azure
2022～ for Internet of Things



1991年大手メーカーに入社し、
工場の自動化や情報システム
の開発、運用に従事。2017年8
月、Seeed株式会社設立時に
転職。組み込みデバイスの開
発環境整備やライブラリ開発
を担当。



↑
趣味

seeed studio



Open Hardware Modules

Single Board Computer



Microcontroller, Module & Extension Board



Wireless Node



Sensor & Actuator



Industrial IoT Solutions

Edge Computing & Private Cloud

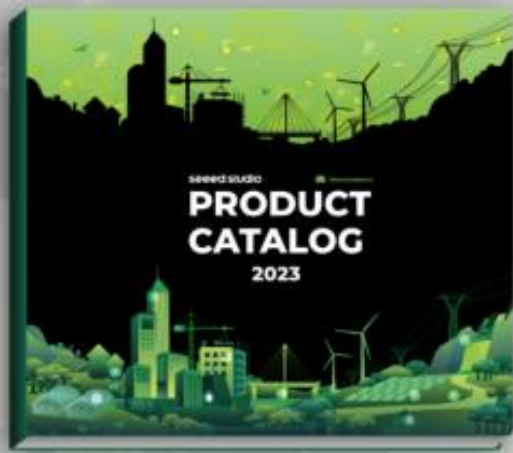


Seed Edge Computing Solution



https://www.seeedstudio.com/make_sense_from_the_true_wild.html

How To Get Involved



Download Seed Studio Catalog 2023

A super comprehensive product catalog where you can have everything you want to know about our product lineups.

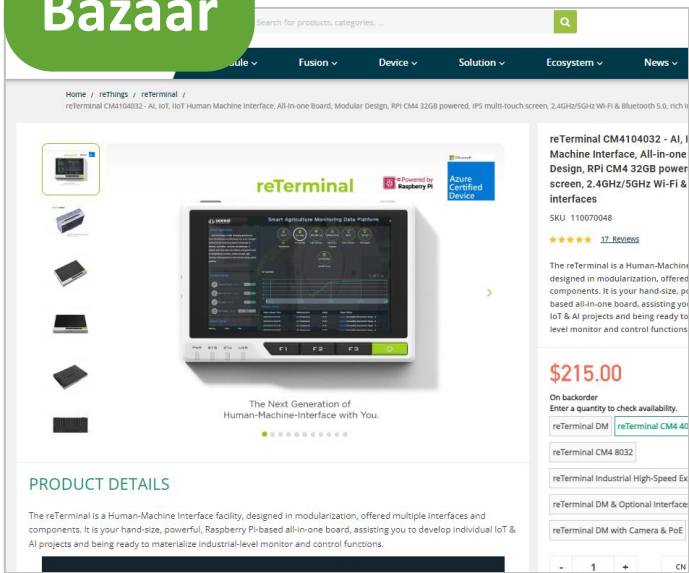
- **Product details** of specifications, features, and applications of our modules, devices, and solutions
- **Hardware services** of customization and manufacturing that we offer to tailor your products
- **Integrated software platforms** for easy hardware deployment
- **Successful cases** of our solutions in various industrial settings
- **Ways to collaborate and co-invent** with us based on your knowhow

Download the catalog to explore your possible application with our advanced perception system.

Available for Download on Mar. 14 (PT)

Bazaar

<https://www.seeedstudio.com/>



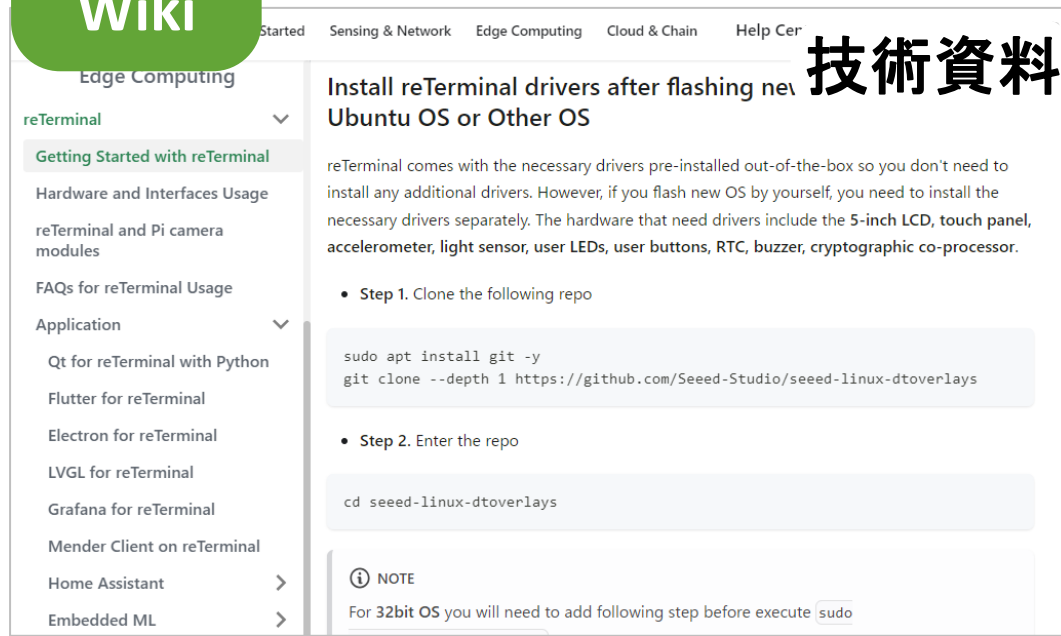
Catalog



https://www.seeedstudio.com/make_sense_from_the_true_wild.html

Wiki

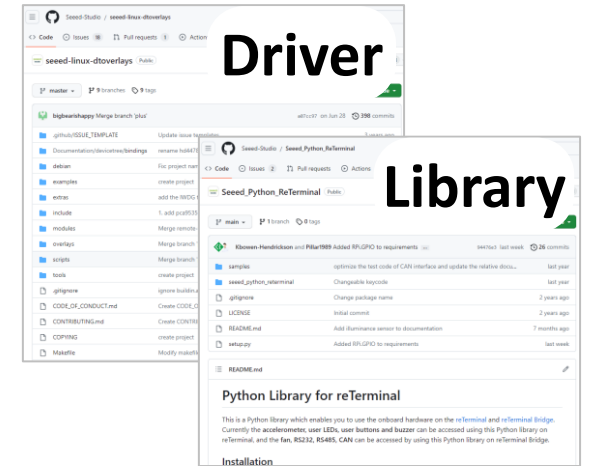
<https://wiki.seeedstudio.com/>



seeed studio



> 450



ひとり開発(ぼっち開発)でも
Gitは使える？





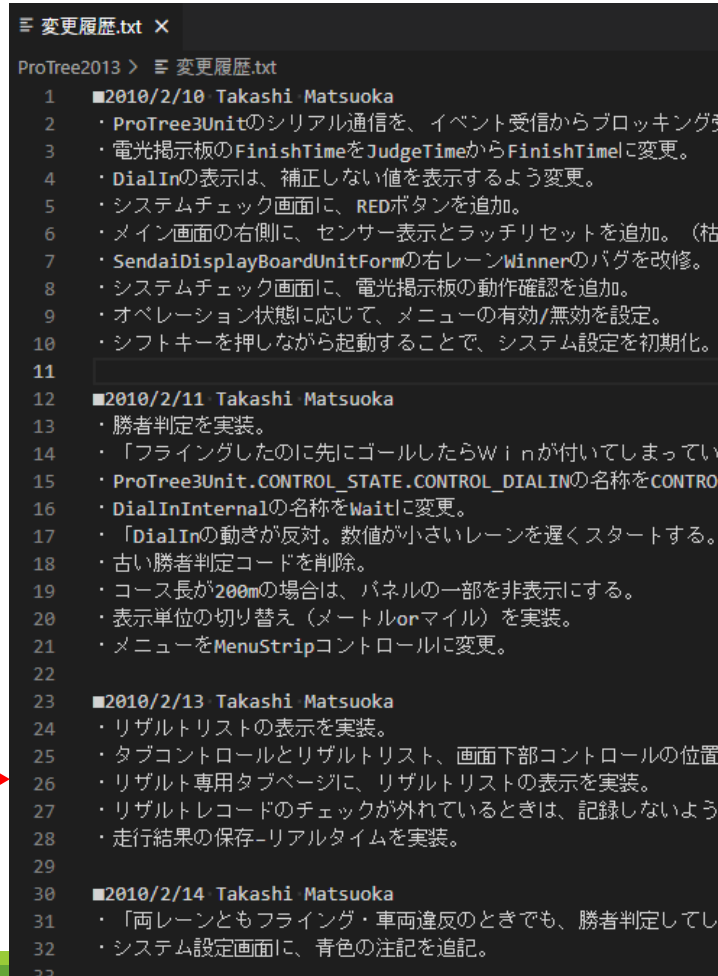
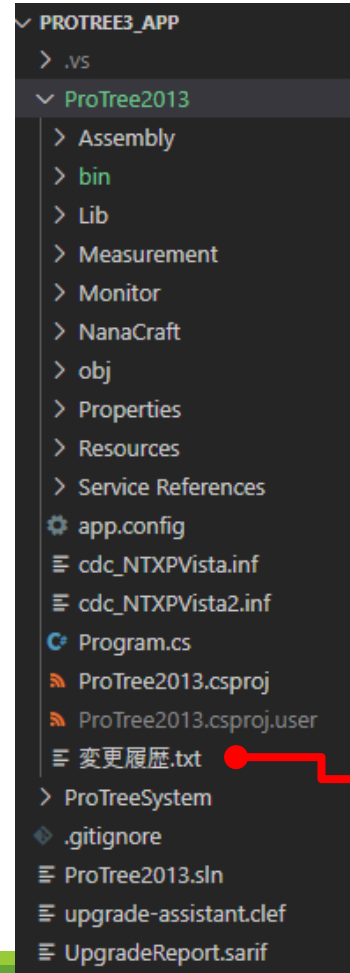
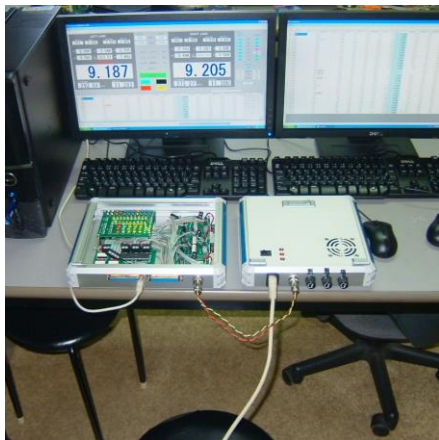
もちろん使えます。

Gitを使うと変更履歴の記録や確認、戻しがスマートに行えます。

費用はかかりません。無償です。

さらに、GitHubなどのホスティングサービスと連携すると、あなたの大切なソースコードを守ってくれます。

ドラッグレース計測システム



139ファイル

DisplayBoardTest.20100124.zip	18 KB	2010/03/13 13:36
ProTree3Monitor.20091105.zip	8 KB	2010/03/13 13:35
ProTree3Notification.20091116.zip	413 KB	2013/03/14 20:58
ProTree2013.4.20.zip	433 KB	2013/03/08 22:15
ProTree2013.20130308A.zip	409 KB	2013/03/10 12:08
ProTree2013.20130310A.zip	409 KB	2013/04/07 13:18
ProTree2013.20130407A.zip	409 KB	2013/04/14 18:27
ProTree2013.20130414A.zip	611 KB	2013/04/23 19:42
ProTree2013.20130421A.nagao.zip	409 KB	2013/04/23 21:11
ProTree2013.20130423A.zip	896 KB	2013/05/20 17:35
ProTree2013.20130520A.zip	896 KB	2013/05/20 18:08
ProTree2013.20130520B.zip	417 KB	2013/05/23 21:04
ProTree2013.20130523A.zip	410 KB	2013/05/24 21:40
ProTree2013.20130524A.zip	420 KB	2014/02/11 11:38
ProTree2013.20140211A.MOTEGI.zip	420 KB	2014/02/11 11:36
ProTree2013.20140211A.zip	422 KB	2014/04/24 22:05
ProTree2013.20140424A.zip	607 KB	2014/04/24 22:21
ProTree2013.20140424B.zip	431 KB	2015/04/20 21:06
ProTree2013.20150420A.zip	627 KB	2016/02/07 12:45
ProTree2013.20150521.zip	424 KB	2016/02/12 23:00
ProTree2013.20160212A.zip	423 KB	2016/02/13 14:42
ProTree2013.20160213A.zip	622 KB	2016/02/14 10:00
ProTree2013.20160213A.nagao.zip		

Git未使用

① 変更履歴、書き忘れ

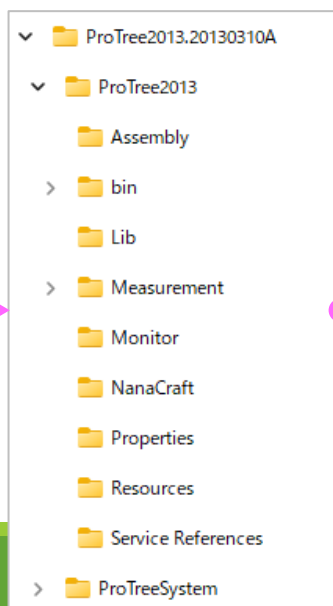
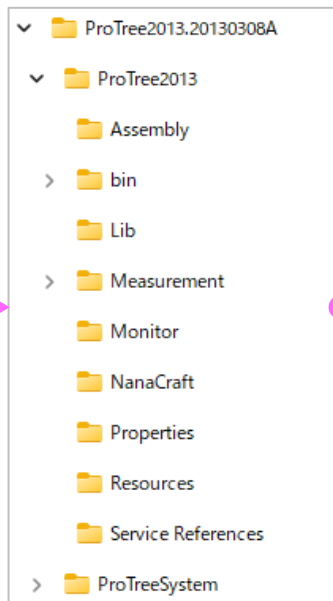
```
変更履歴.txt
1 2016/2/1
2 ProTree
3 電光掲示
4 DialInの表示は、補正しない値を表示するよう変更。
5 システムチェック画面に、REDボタンを追加。
6 メイン画面の右側に、センサー表示とラッチリセットを追加。(結果
7 SendiDisplayBoardUnitFormの右レーンInnerのバグを修正。
8 システムチェック画面に、電光掲示板の動作確認を追加。
9 オペレーション状態に応じて、メニューの有効/無効を設定。
10 シフトキーを押しながら起動することで、システム設定を初期化。
```

② zip忘れ

③ zip解凍

ファイル名	サイズ	更新日時
ProTree2013.20130308A.zip	433 KB	2013/03/08 22:15
ProTree2013.20130310A.zip	109 KB	2013/03/10 12:08
ProTree2013.20130407A.zip	409 KB	2013/04/07 13:18
ProTree2013.20130414A.zip	409 KB	2013/04/14 18:27
ProTree2013.20130421A.nagao.zip	611 KB	2013/04/23 19:42
ProTree2013.20130423A.zip	409 KB	2013/04/23 21:11
ProTree2013.20130520A.zip	896 KB	2013/05/20 17:35
ProTree2013.20130520B.zip	896 KB	2013/05/20 18:08
ProTree2013.20130523A.zip	417 KB	2013/05/23 21:04
ProTree2013.20130524A.zip	410 KB	2013/05/24 21:40
ProTree2013.20140211A.MOTEGI.zip	420 KB	2014/02/11 11:38
ProTree2013.20140211A	420 KB	2014/02/11 11:36

⑤ 特別扱い



④ フォルダ比較

フォルダ名	比較結果
ProTree2013	フォルダは異なっています
Assembly	同一
bin	フォルダは異なっています
Lib	同一
Measurement	フォルダは異なっています
Radio	フォルダは異なっています
TimeBoardUnit.cs	テキストファイルは異なります
FrmCarNoEdit.cs	テキストファイルは同一です
FrmCarNoEdit.Designer.cs	テキストファイルは同一です
FrmCarNoEdit.resx	テキストファイルは同一です
FrmMain.cs	テキストファイルは同一です
FrmMain.Designer.cs	テキストファイルは同一です
FrmMain.resx	テキストファイルは同一です
FrmResultListCreate.cs	テキストファイルは同一です
FrmResultListCreate.Designer.cs	テキストファイルは同一です
FrmResultListCreate.resx	テキストファイルは同一です
FrmStartType.cs	テキストファイルは同一です
FrmStartType.Designer.cs	テキストファイルは同一です
FrmStartType.resx	テキストファイルは同一です
FrmSystemCheck.cs	テキストファイルは同一です
FrmSystemCheck.Designer.cs	テキストファイルは同一です
FrmSystemCheck.resx	テキストファイルは同一です
FrmSystemSettings.cs	テキストファイルは同一です
FrmSystemSettings.Designer.cs	テキストファイルは同一です
FrmSystemSettings.resx	テキストファイルは同一です
FrmTimeBoardDetail.cs	テキストファイルは同一です
FrmTimeBoardDetail.Designer.cs	テキストファイルは同一です
FrmTimeBoardDetail.resx	テキストファイルは同一です
FrmTimeBoardUnit.cs	テキストファイルは同一です

```
if (!value.HasValue)
{
    str = " ";
    dot = false;
}
else
{
    // -9.999~99.999の範囲に制限。
    value = Math.Max(value.Value, -9.999);
    value = Math.Min(value.Value, 99.999);
    str = string.Format("{0,6:f3}");
    dot = true;
}
this.WriteAndRead(22, (byte)(Conv...

public void SetSegmentSpeed(double? va
{
    string str;
    if (!value.HasValue)
    {
        str = " ";
    }
    else
    {
        // -9.999~99.999の範囲に制限。
        value = Math.Max(value.Value, -9.999);
        value = Math.Min(value.Value, 99.999);
        str = string.Format("{0,6:f3}");
        dot = true;
    }
    this.WriteAndRead(22, (byte)(Conv...
```

Git使用

The screenshot shows the Git Graph interface. The top part is a commit history table with columns for Graph, Description, Date, Author, and Commit. The bottom part shows a diff view for a specific commit, listing files and their changes.

Graph	Description	Date	Author	Commit
origin/hokkaido	キーファイル確認を削除	14 Jul 2023 21:15	MATSUOKA Tak...	6b13c24c
dev/origin	タイムシートの"1/6"を"1000"に訂正	21 Jun 2023 10:...	MATSUOKA Tak...	c78ccb0e
	バックアップを有効化	21 Jun 2023 10:...	MATSUOKA Tak...	3b042fdf
	メイン - 計測画面の右下にあるテキストボックス、ボタン位置を調整	15 Jan 2023 20:...	Takashi Matsuo...	f542f69f
	画面レイアウトをざっくり改修 #8	14 Jan 2023 22:...	Takashi Matsuo...	30053da6
	ProTree2013を.NET 6.0でコンパイル	14 Jan 2023 21:...	Takashi Matsuo...	e6e29c81
	ProTree2013を.NET 6.0にアップグレード	14 Jan 2023 21:...	Takashi Matsuo...	358ac758
	ProTreeSystemを.NET 6.0でコンパイル	14 Jan 2023 20:...	Takashi Matsuo...	dd9b5557
	ProTreeSystemを.NET 6.0にアップグレード	14 Jan 2023 20:...	Takashi Matsuo...	dd9b5557
main/origin	.gitignore	14 Jan 2023 20:...	Takashi Matsuo...	dd9b5557
	不要なファイルの追跡を除外	14 Jan 2023 20:...	Takashi Matsuo...	dd9b5557
	文字コードをSJISからUTF-8に変更	14 Jan 2023 20:...	Takashi Matsuo...	dd9b5557
	20190226A_NAGAO	11 Jan 2023 20:...	Takashi Matsuo...	dd9b5557
	20190104A	27 Dec 2022 21:...	Takashi Matsuo...	caba7753
	20190103A	27 Dec 2022 21:...	Takashi Matsuo...	df29ac5a
	20180813	27 Dec 2022 21:...	Takashi Matsuo...	dd2181c4

Diff view for commit dd2181c4c692b00d35cdeeb94fe7bc6c47991106 to caba775306a132969bb671be77cd2173a10a2ac:

- ResultTable.cs (+79|-1)
- TimeSheetHks.cs (+27|-22)
- TimeSheetTarget.cs (+27|-22)
- Measurement
 - FrmMain.cs (+12|-1)
 - FrmMain.Designer.cs (+34|-21)
 - FrmSystemCheck.cs (+14|-10)
 - FrmSystemCheck.Designer.cs (+74|-48)
 - FrmSystemSettings.cs (+8|-0)
 - FrmSystemSettings.Designer.cs (+184|-...)
 - LanePanel.cs (+67|-17)
 - LanePanel.Designer.cs (+42|-1)
 - LanePanel.resx (+2|-2)
 - Parameters.cs (+4|-0)
 - ProTree3Unit.cs (+58|-38)
 - ProTreeLane.cs (+29|-0)
 - SensorPanel.cs (+5|-0)
 - SensorPanel.Designer.cs (+65|-26)
- Monitor
 - LanePanel2.Designer.cs (+6|-3)
- 変更履歴.txt (+7|-0)

変更の記録
(コミット)

変更の関係
(コミット履歴)

The screenshot shows a code diff for FrmMain.cs. The left side is the original code, and the right side is the modified code. The diff highlights changes in the Speed200LengthDiff, Speed300LengthDiff, and Speed400LengthDiff properties.

```
335 double lWait = rDialIn;
336 if (this.unitLaneR.Enabled && this.unitLaneL.Enabled)
337 {
338     double waitMin = Math.Min(rWait, lWait);
339     rWait -= waitMin;
340     lWait -= waitMin;
341 }
342 else
343 {
344     rWait = 0;
345     lWait = 0;
346 }
347
348 this.Judge = JudgeType.UNKNOWN;
349
350 this.unitLaneR.CarNo = rCarNo;
351 this.unitLaneR.DialIn = rDialIn;
352 this.unitLaneR.Wait = rWait;
353 this.unitLaneR.Stock = stock;
354 this.unitLaneR.UseGuard = Global.Parameters.UseGuard;
355 this.unitLaneR.CheckCarRegulation = Global.Parameters.CheckCarRegulation;
356 this.unitLaneR.MeasureOfDeadLeaf = Global.Parameters.MeasureOfDeadLeaf;
357 this.unitLaneR.CourseLength = courseLength;
358 this.unitLaneR.Speed200LengthDiff = Global.Parameters.Speed200LengthDiff;
359 this.unitLaneR.Speed400LengthDiff = Global.Parameters.Speed400LengthDiff;
360 this.unitLaneR.CourseLength200 = Global.Parameters.CourseLength200;
361 this.unitLaneR.CourseLength400 = Global.Parameters.CourseLength400;
362 this.unitLaneR.JudgeTypeRtEt = judgeTypeRtEt;
363 this.unitLaneR.TimeoutRt = Global.Parameters.TimeOverRt;
364 this.unitLaneR.TimeoutFinish = Global.Parameters.TimeOverGoal;
365 this.unitLaneR.RtBaseGo = Global.Parameters.RtBaseGo;
366 this.unitLaneR.BestRt = this.resultTable.FindBestRtByCarNo(rCarNo);
367 this.unitLaneR.BestTFinish = this.resultTable.FindBestTFinishByCarNo(rCarNo);
368 this.unitLaneR.RallyName = this.chkRallyNamePrintOut.Checked ? Global.Parameters.RallyName : Global.Parameters.RallyName;
369 this.unitLaneR.RallySponsor = this.chkRallyNamePrintOut.Checked ? Global.Parameters.RallySponsor : Global.Parameters.RallySponsor;
370 this.unitLaneR.TimeSheetType = Global.Parameters.TimeSheetType;
371
372 this.unitLaneL.CarNo = lCarNo;
373 this.unitLaneL.DialIn = lDialIn;
374 this.unitLaneL.Wait = lWait;
375 this.unitLaneL.Stock = stock;
376 this.unitLaneL.UseGuard = Global.Parameters.UseGuard;
377 this.unitLaneL.CheckCarRegulation = Global.Parameters.CheckCarRegulation;
378 this.unitLaneL.MeasureOfDeadLeaf = Global.Parameters.MeasureOfDeadLeaf;
379 this.unitLaneL.CourseLength = courseLength;
380 this.unitLaneL.Speed200LengthDiff = Global.Parameters.Speed200LengthDiff;
381 this.unitLaneL.Speed300LengthDiff = Global.Parameters.Speed300LengthDiff;
382 this.unitLaneL.Speed400LengthDiff = Global.Parameters.Speed400LengthDiff;
```

比較
(差分)

Lab1 インストール

Visual Studio Code

参考画面ショット: <https://github.com/algyan/XIAOGYAN/blob/main/manuals/software/screenshot/install/vscode.md>

1. ブラウザでDownload Visual Studio Codeページ (<https://code.visualstudio.com/download>)を開いてください。
2. Windowsをクリックしてください。インストールファイルがダウンロードされます。
3. インストールファイルを実行してください。Visual Studio Codeのセットアップ画面が表示されます。
4. 画面の指示にしたがってセットアップを完了してください。
 - 使用許諾契約書の同意 - 同意するを選択
 - 追加タスクの選択 - PATHへの追加(再起動後に使用可能)を選択

Git

参考画面ショット: <https://github.com/algyan/XIAOGYAN/blob/main/manuals/software/screenshot/install/git.md>

1. ブラウザでDownload Gitページ (<https://git-scm.com/downloads>)を開いてください。
2. Windowsをクリックしてください。Download for Windowsページが表示されます。
3. 64-bit Git for Windows Setupをクリックしてください。インストールファイルがダウンロードされます。
4. インストールファイルを実行してください。Gitのセットアップ画面が表示されます。
5. 画面の指示にしたがってセットアップを完了してください。
 - Select Components - Git LFSとScalarを選択
 - Choosing the default editor used by Git - Use Visual Studio Code as Git's default editorを選択
 - Adjusting your PATH environment - Git from the command line and also from 3rd-party softwareを選択

Visual Studio Code拡張

参考画面ショット: <https://github.com/algyan/XIAOGYAN/blob/main/manuals/software/screenshot/install/vscodeext.md>

※ PlatformIOの部分は不要

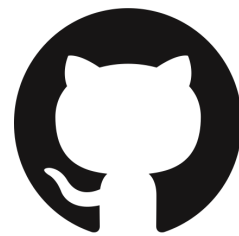
1. Visual Studio Codeを起動してください。
2. EXTENSIONS画面でMS-CEINTL.vscode-language-pack-ja (Japanese Language Pack for Visual Studio Code)を検索してインストールしてください。
3. Visual Studio Codeを再起動してください。
4. 拡張機能画面でmhutchie.git-graph (Git Graph)を検索してインストールしてください。
5. Visual Studio Codeを再起動してください。

Lab2 予備知識



git

!=



GitHub



“Git(ギット)は、プログラムのソースコードなどの**変更履歴を記録・追跡**するための**分散型バージョン管理システム**である。Linuxカーネルのソースコード管理に用いるためにリーナス・トーバルズによって開発され、それ以降ほかの多くのプロジェクトで採用されている。”

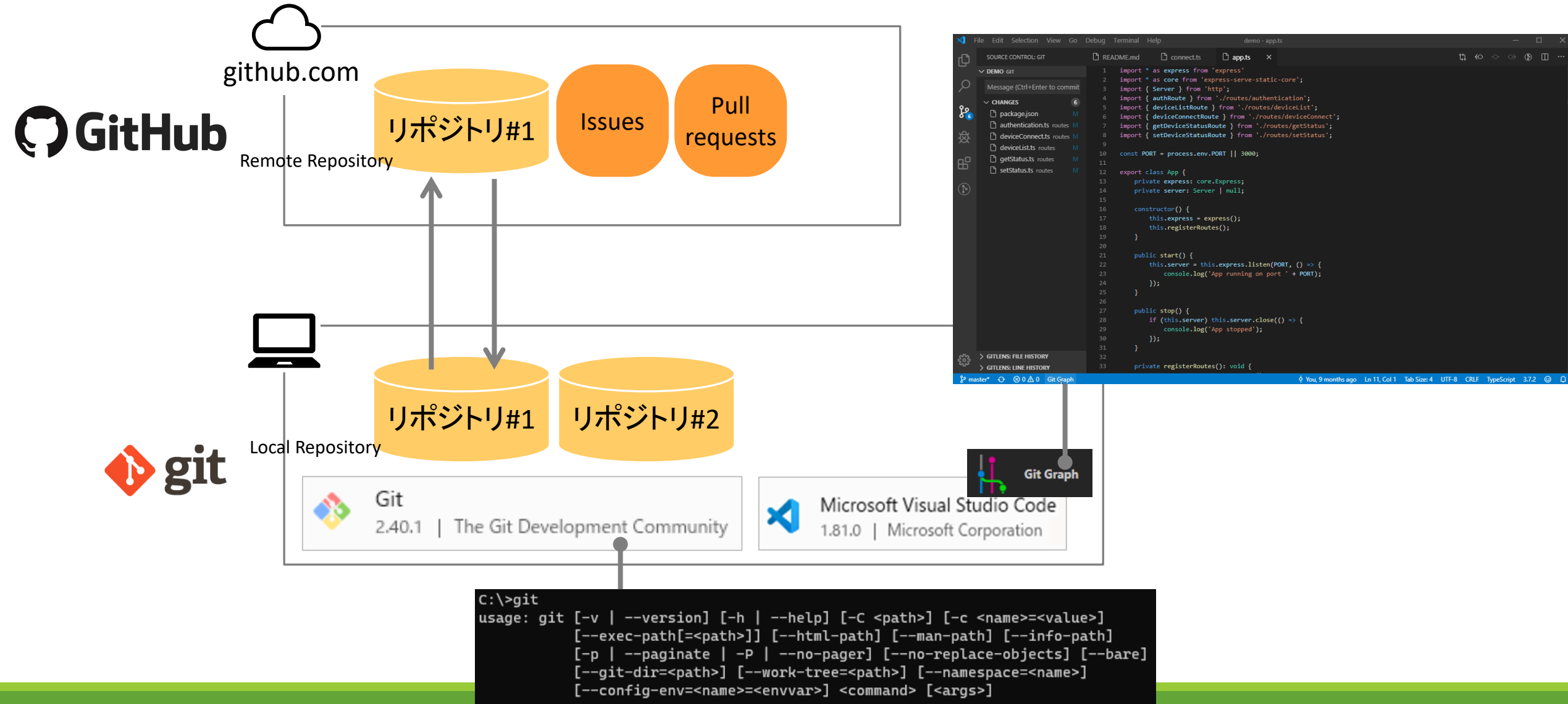
<https://ja.wikipedia.org/wiki/Git>



“GitHub(ギットハブ)は、ソフトウェア開発のプラットフォームであり、**ソースコードをホスティング**する。コードのバージョン管理システムにはGitを使用する。”

<https://ja.wikipedia.org/wiki/GitHub>

Git / GitHub



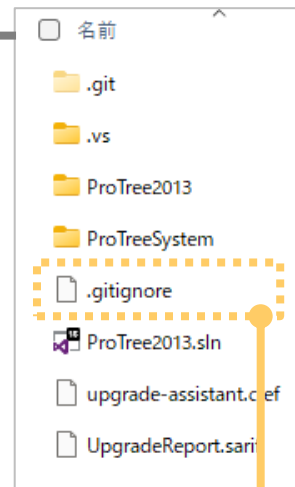
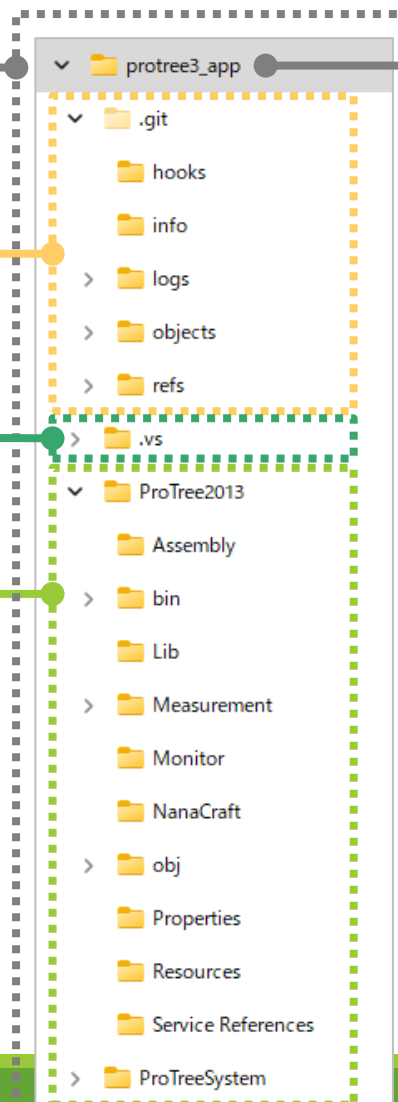
作業エリア / リポジトリ / 追跡

作業エリア
作業フォルダー
プロジェクトフォルダー

リポジトリ

追跡対象外
(Untracked)

追跡対象
(Tracked)



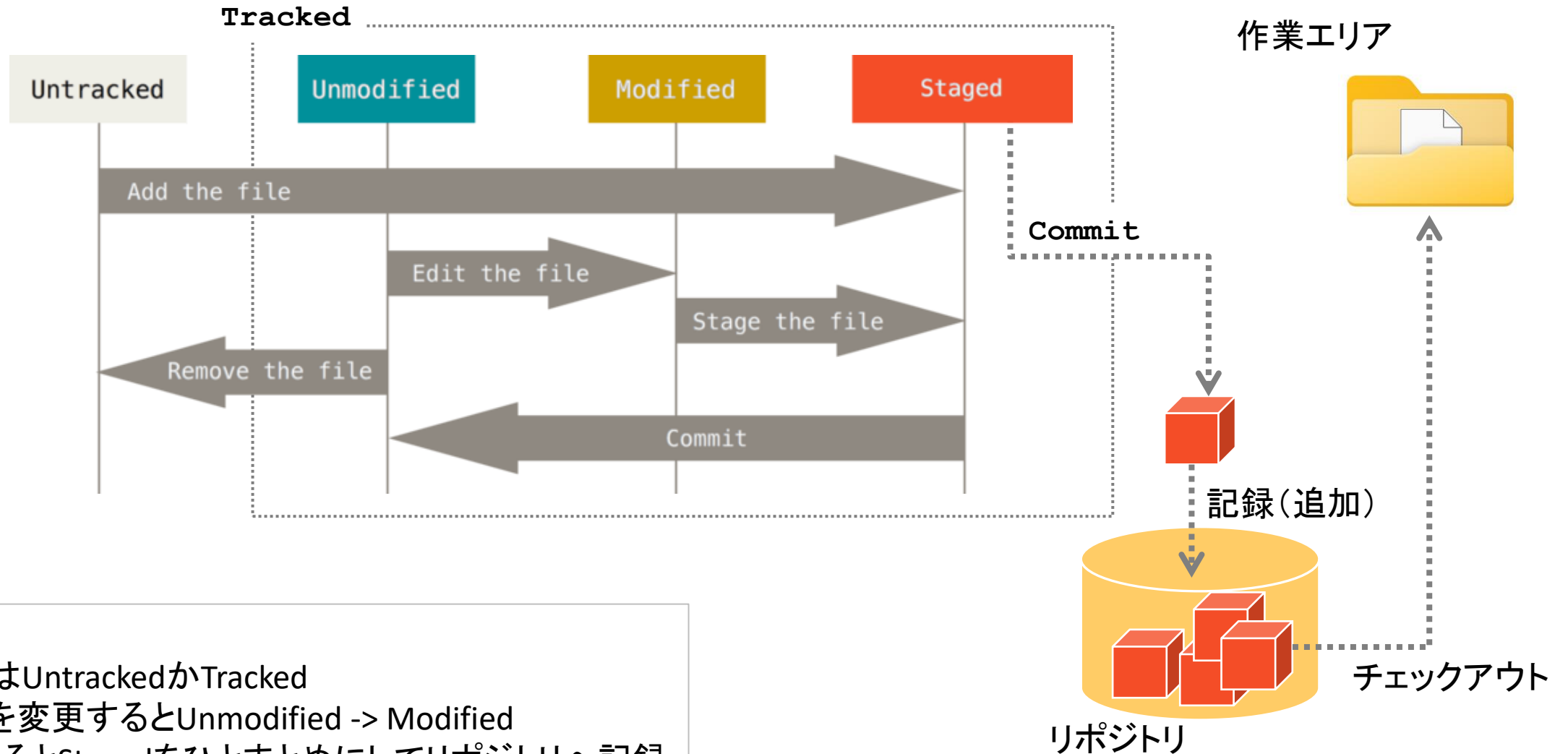
追跡無視

```
.gitignore
1 *.csproj.user
2 .vs/
3 /*/obj/
4 /*/bin/**/Storage/
5 /*/bin/**/Parameters.xml
6
```

.gitignore templates
<https://github.com/github/gitignore>

- ✓ 作業エリアのファイルを追跡してリポジトリに記録
- ✓ リポジトリは.gitフォルダー
- ✓ .gitignoreで追跡を無視
- ✓ 作業エリアの削除はフォルダー削除でOK

未変更 / 変更 / ステージとコミット、チェックアウト

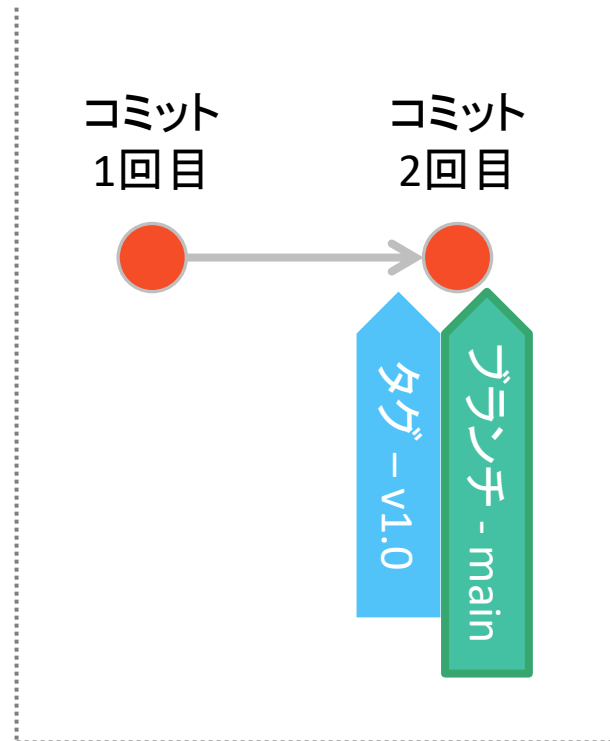


変更の記録

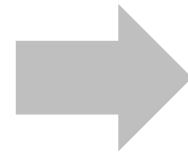
- ✓ ファイルはUntrackedかTracked
- ✓ ファイルを変更するとUnmodified -> Modified
- ✓ コミットするとStagedをひとまとめにしてリポジトリへ記録
- ✓ コミットを作業エリアへ展開するときはチェックアウト

コミット履歴 / タグ / ブランチ

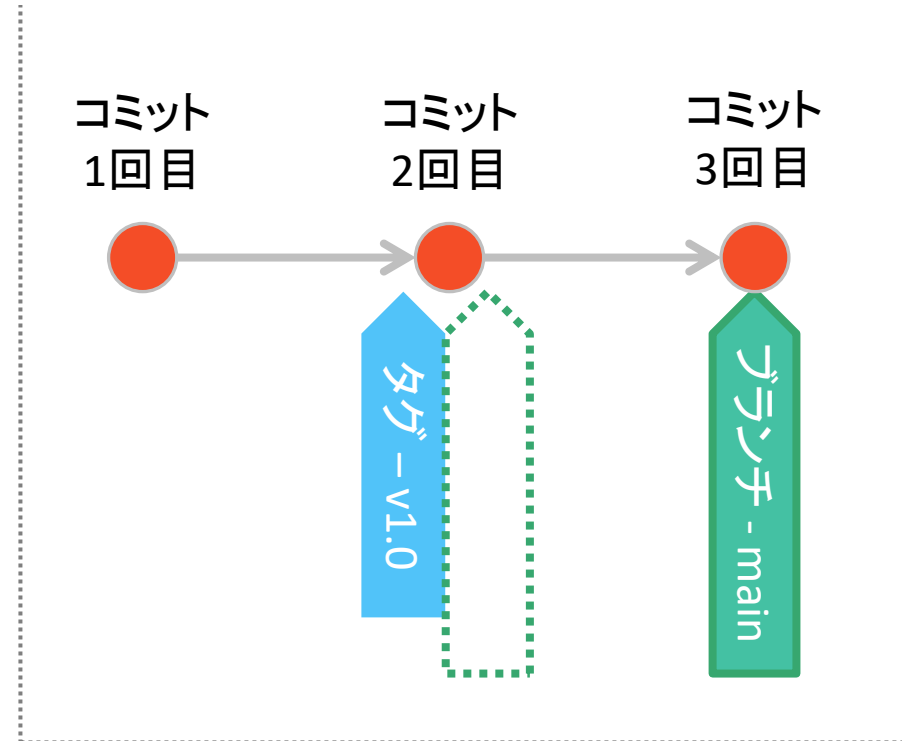
コミット履歴 #1



mainブランチで
コミットすると...



コミット履歴 #2

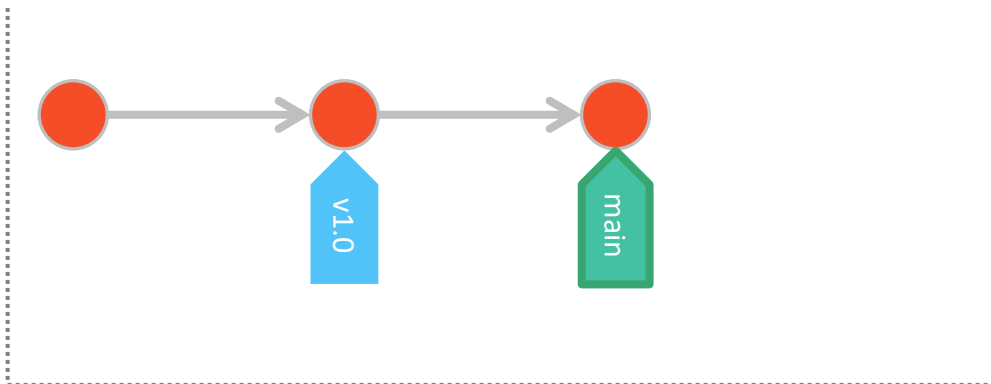


変更の関係

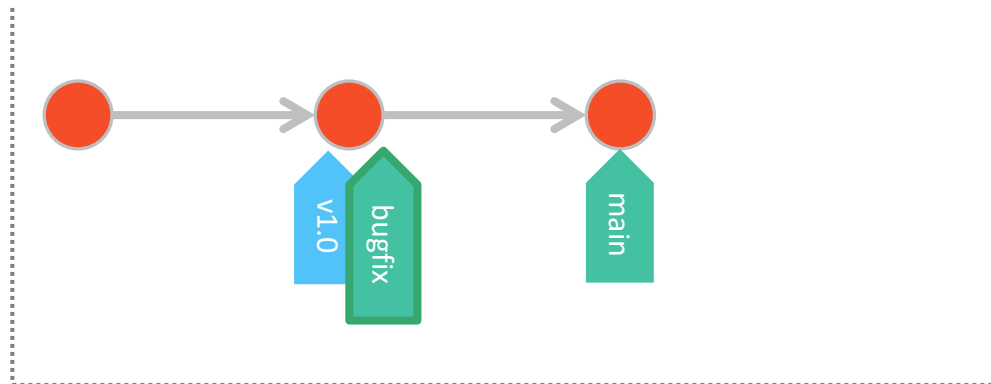
- ✓ コミットの履歴はコミット履歴
- ✓ コミットに名前を付けることができる。タグとブランチ
- ✓ タグはコミットの別名
- ✓ ブランチは作業エリアの選択に使う

分岐 / マージ

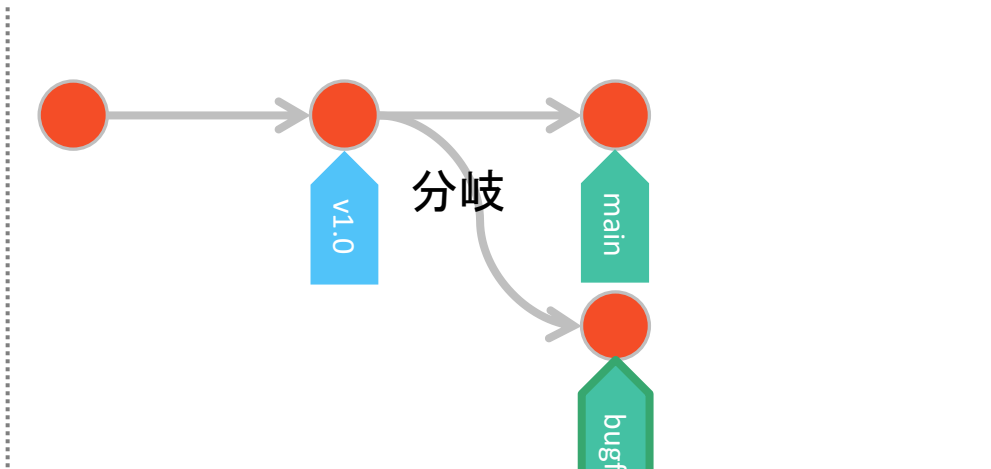
コミット履歴 #1



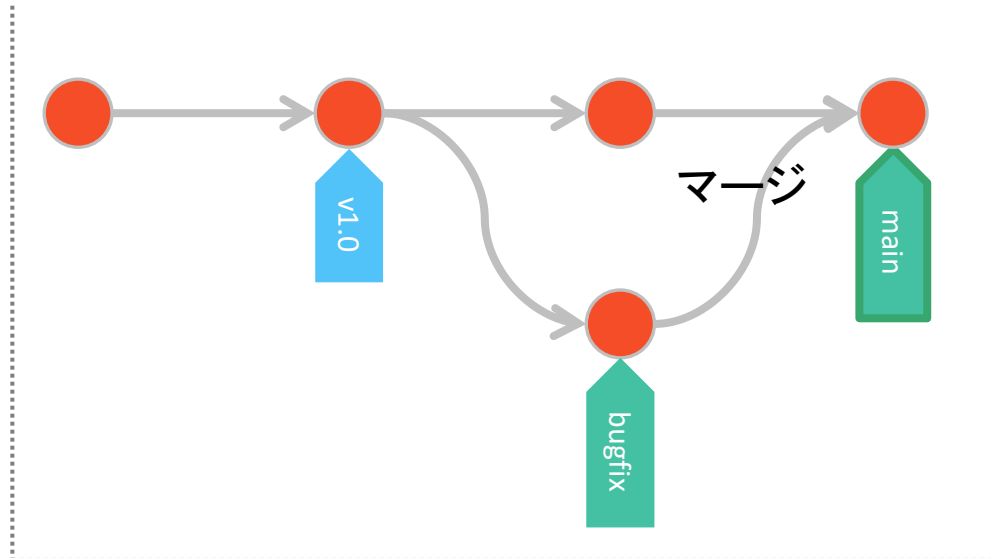
コミット履歴 #2



コミット履歴 #3



コミット履歴 #4



変更の関係

- ✓ 分岐という操作は無い(ブランチを作ってコミット)
- ✓ マージは細心の注意を

Lab3 基本操作

1. リポジトリ化

2. 変更の記録

3. 分岐

4. マージ

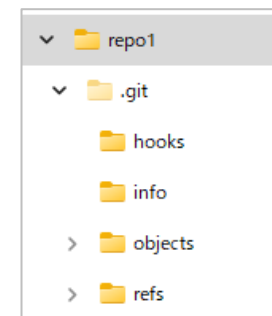
1-1. リポジトリを作成

空のリポジトリを作成します。

1. 作業エリア(ここではC:\swest25s3c\myrepo)に移動して、git initコマンドを実行

```
C:\> mkdir swest25s3c
C:\> cd swest25s3c
C:\swest25s3c> mkdir myrepo
C:\swest25s3c> cd myrepo
C:\swest25s3c\myrepo> git init
Initialized empty Git repository in C:/swest25s3c/myrepo/.git/
C:\swest25s3c\myrepo>
```

エクスプローラーのオプションで「隠しフォルダーを表示する」にすると、.gitフォルダーを見ることができます。



1-2. ReButtonAppをコピー

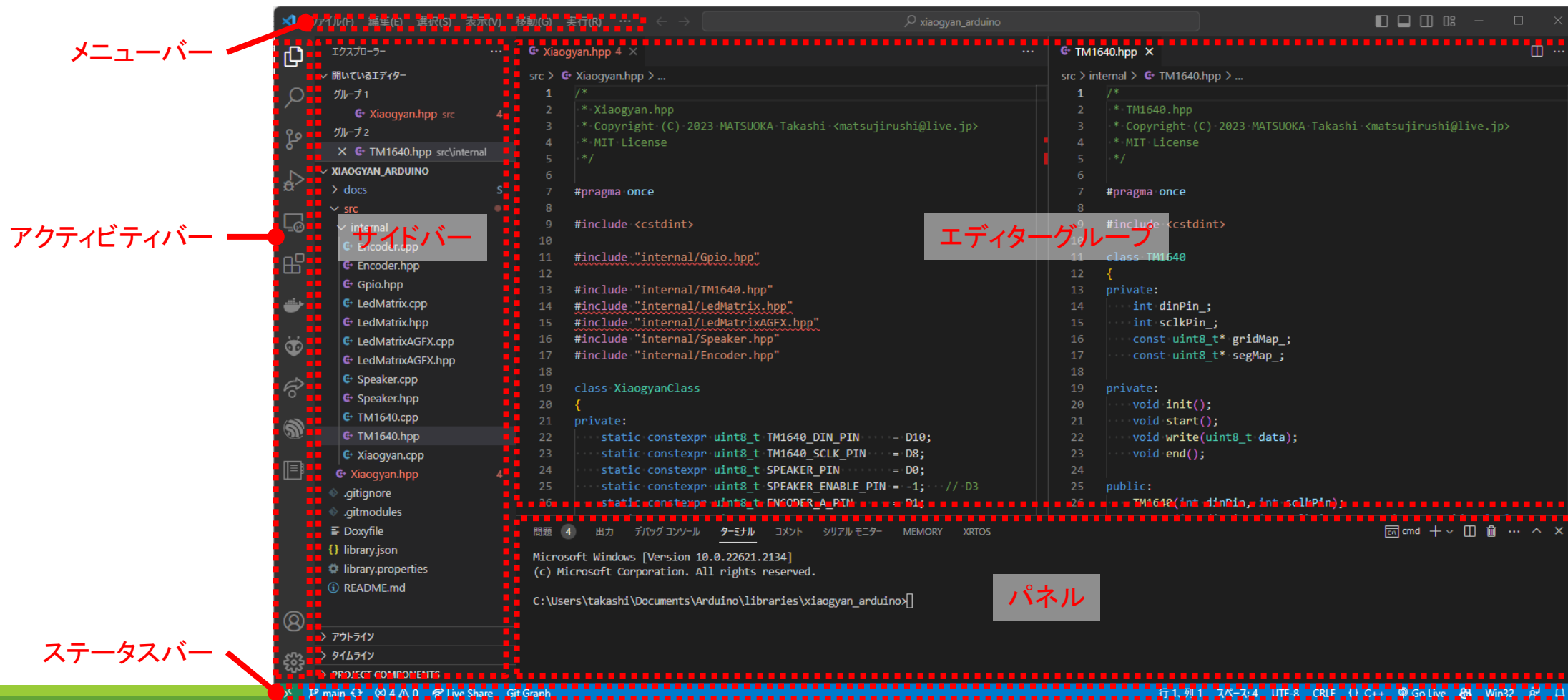
以降、履歴が入っているリポジトリを使いたいので、GitHubにあるReButtonAppをコピーします。

1. C:¥swest25s3cに移動して、git cloneコマンドを実行

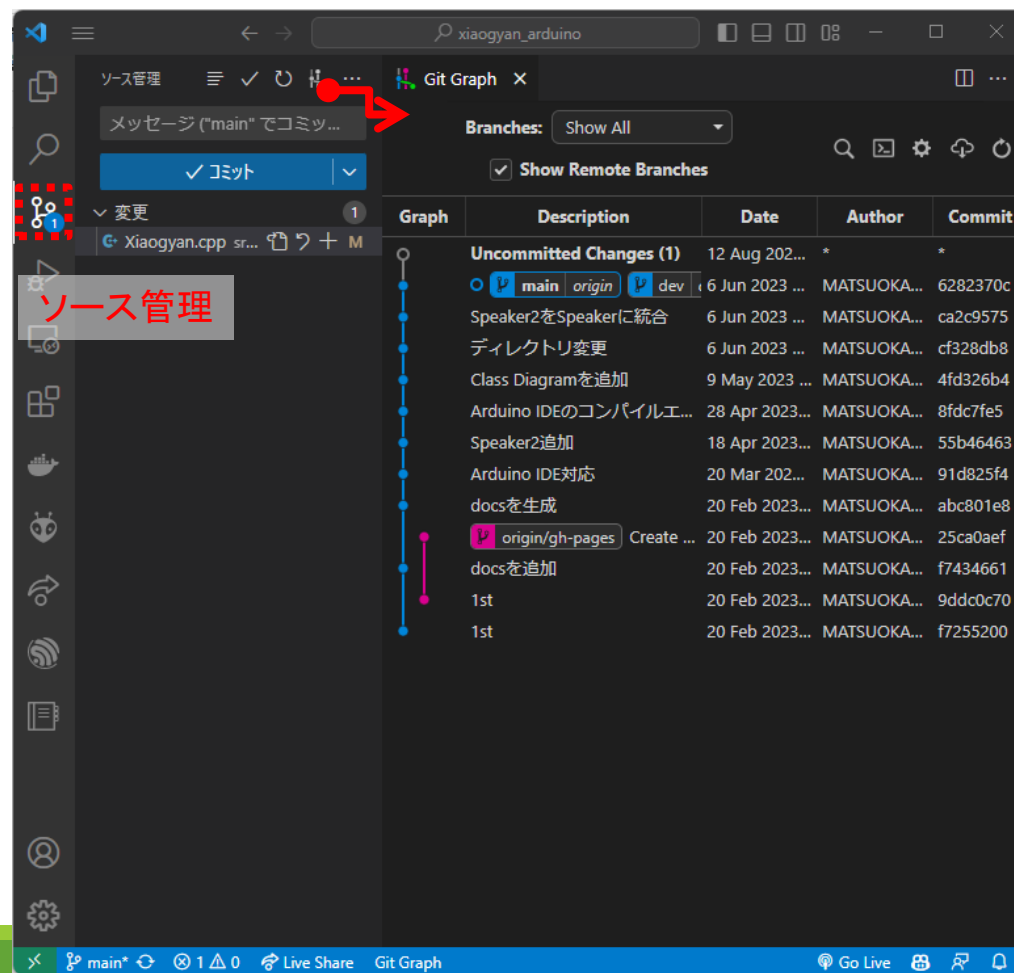
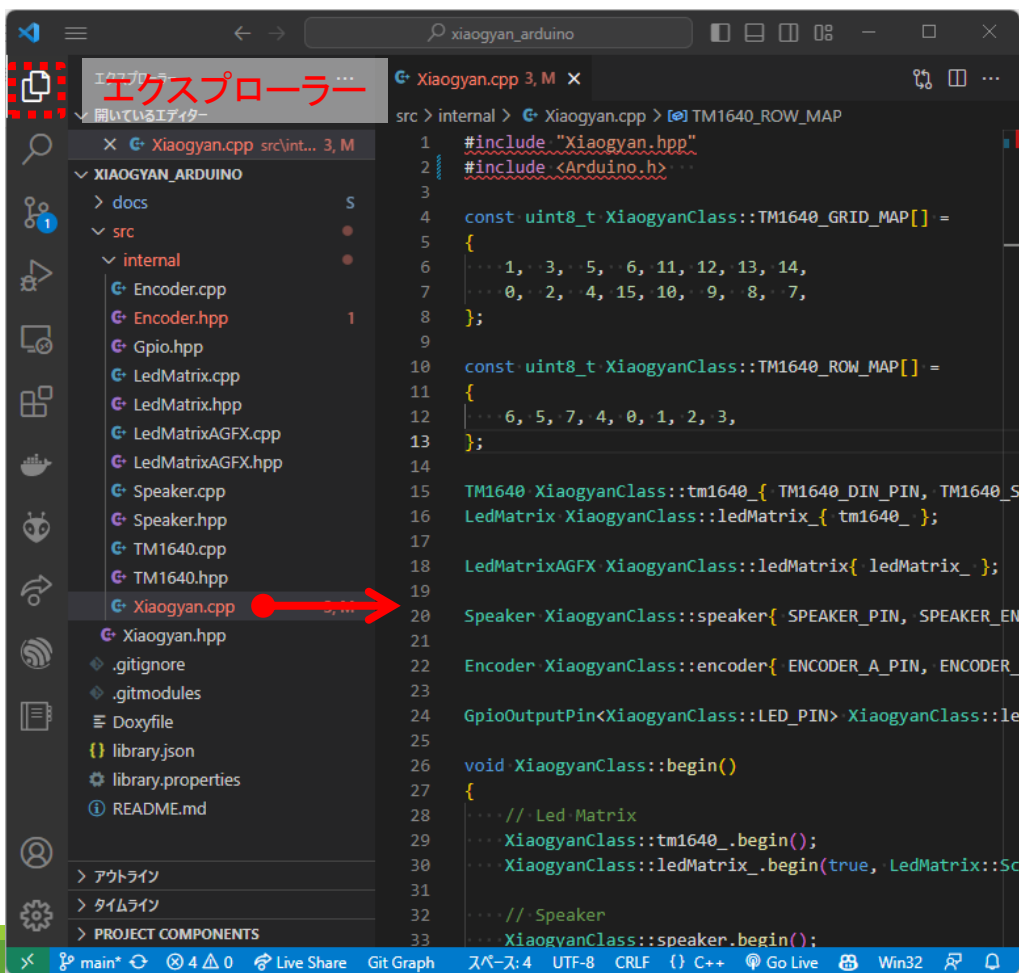
```
C:¥> cd swest25s3c
C:¥swest25s3c>git clone https://github.com/SeedJP/ReButtonApp
Cloning into 'ReButtonApp'...
remote: Enumerating objects: 489, done.
remote: Counting objects: 100% (58/58), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 489 (delta 43), reused 37 (delta 34), pack-reused 431
Receiving objects: 100% (489/489), 136.64 KiB | 3.50 MiB/s, done.
Resolving deltas: 100% (296/296), done.

C:¥swest25s3c>
```

1-3. VSCodeで開く



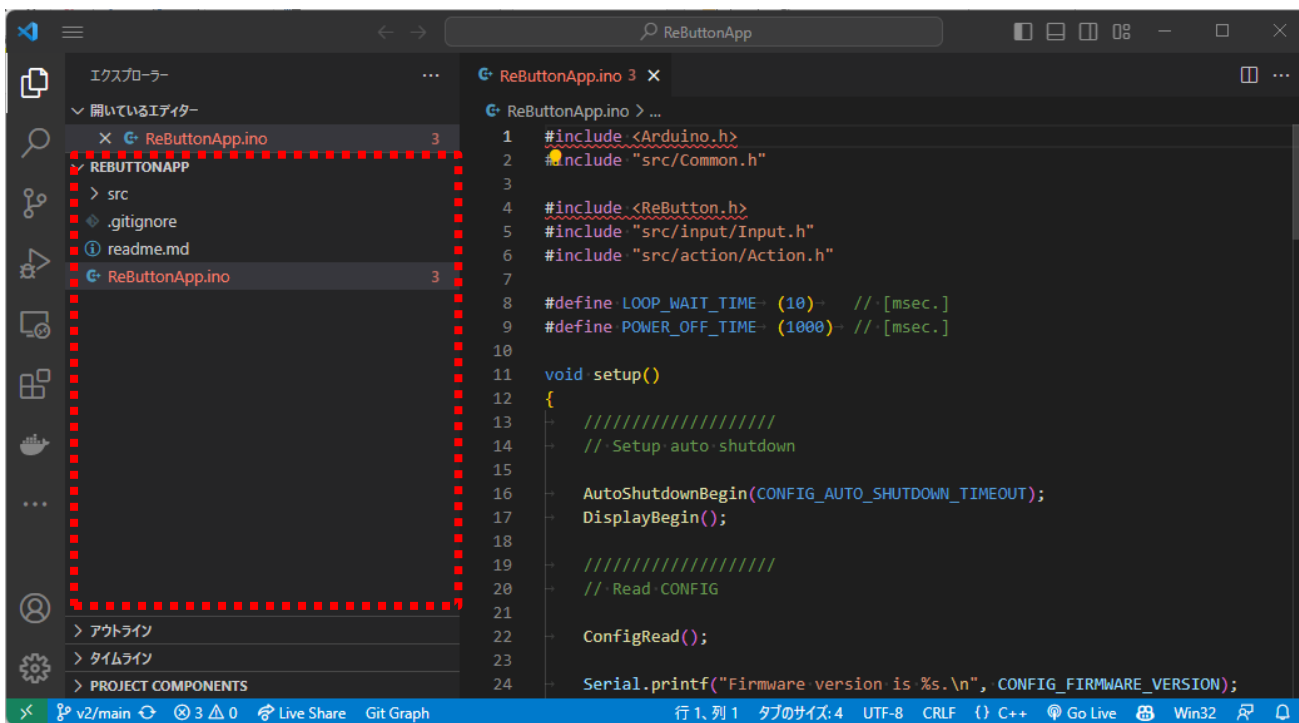
1-3. VSCodeで開く



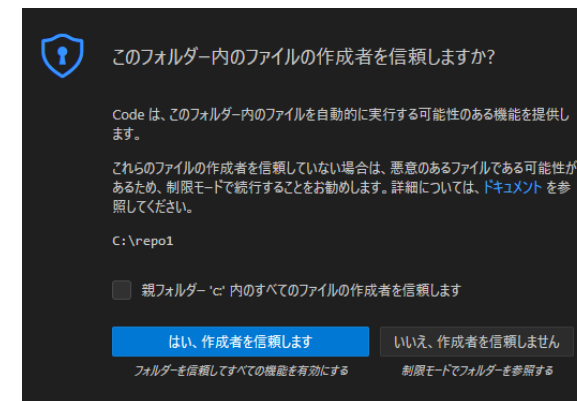
1-3. VSCodeで開く

VSCodeで作業エリアを開きます。

1. 「ファイル > フォルダを開く」を実行して、作業エリア (C:\swest25s3c\ReButtonApp) を開く



「このフォルダ内のファイルの作成者を信頼しますか？」が表示されたときは、「はい、作成者を信頼します」をクリックしてください。



1-3. VSCodeで開く

ソース管理

メッセージ ("v2/main" でコミットするため...)

✓ コミット

Git Graph

Branches: Show All Show Remote Branches

Graph	Description	Date	Author	Commit
origin/v1/main	v1.12	21 Nov 202...	Takashi Mat...	bd110d1d
	Specifying a root certificate for OTA	17 Nov 202...	Takashi Mat...	d48c2a7f
v2/main origin	Modify the mo...	1 Sep 2021 ...	Takashi Mat...	4abf29d9
	Change property name "actionCount" to "lastActionCount"	1 Sep 2021 ...	Takashi Mat...	7d40be40
	Implement IoT Plug and Play	1 Sep 2021 ...	Takashi Mat...	fc0e9ce6
	Support for platform v2.0	31 Aug 202...	Takashi Mat...	3f5e7371
v1.11		26 Dec 201...	Takashi Mat...	4eb2abba
	Update readme	26 Dec 201...	Takashi Mat...	ae130f19
	support hidden ssid	25 Dec 201...	Toshio HIG...	6afc796b
v1.10		3 Dec 2019 ...	Takashi Mat...	07fc7f85
	Change timer class (bugfix)	3 Dec 2019 ...	Takashi Mat...	f27e80a2
	Add batteryVoltage in telemetryInterval	26 Nov 201...	Takashi Mat...	a774fab
	Add Grove-BME280	26 Nov 201...	Takashi Mat...	1908cd39
v1.9		22 Nov 201...	Takashi Mat...	15bec736
	Bugfix at ACTION_CONNECTED	22 Nov 201...	Takashi Mat...	ec4b465b
v1.8		13 Nov 201...	Takashi Mat...	1c3cb726
	Add telemetryInterval.at ACTION_CONNECTED	13 Nov 201...	Takashi Mat...	2509ab19

変更履歴

v2/main 0 0 Live Share Git Graph Go Live

1-4. 追跡無視を設定

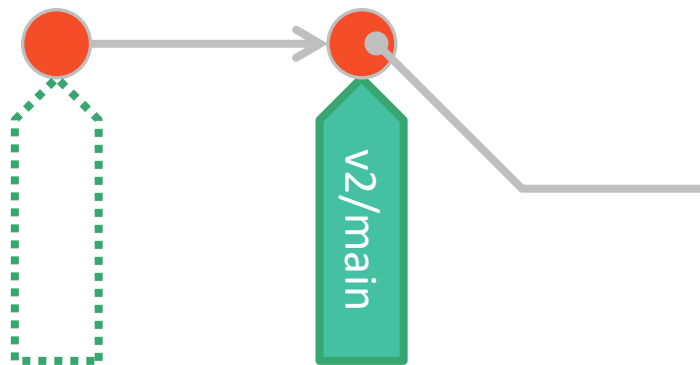
追跡無視の振る舞いを確認します。

1. ReButtonApp.objファイルを作成
2. ソース管理画面を確認
3. .gitignoreファイルに*.objを追記
4. ソース管理画面を確認



2. 変更の記録

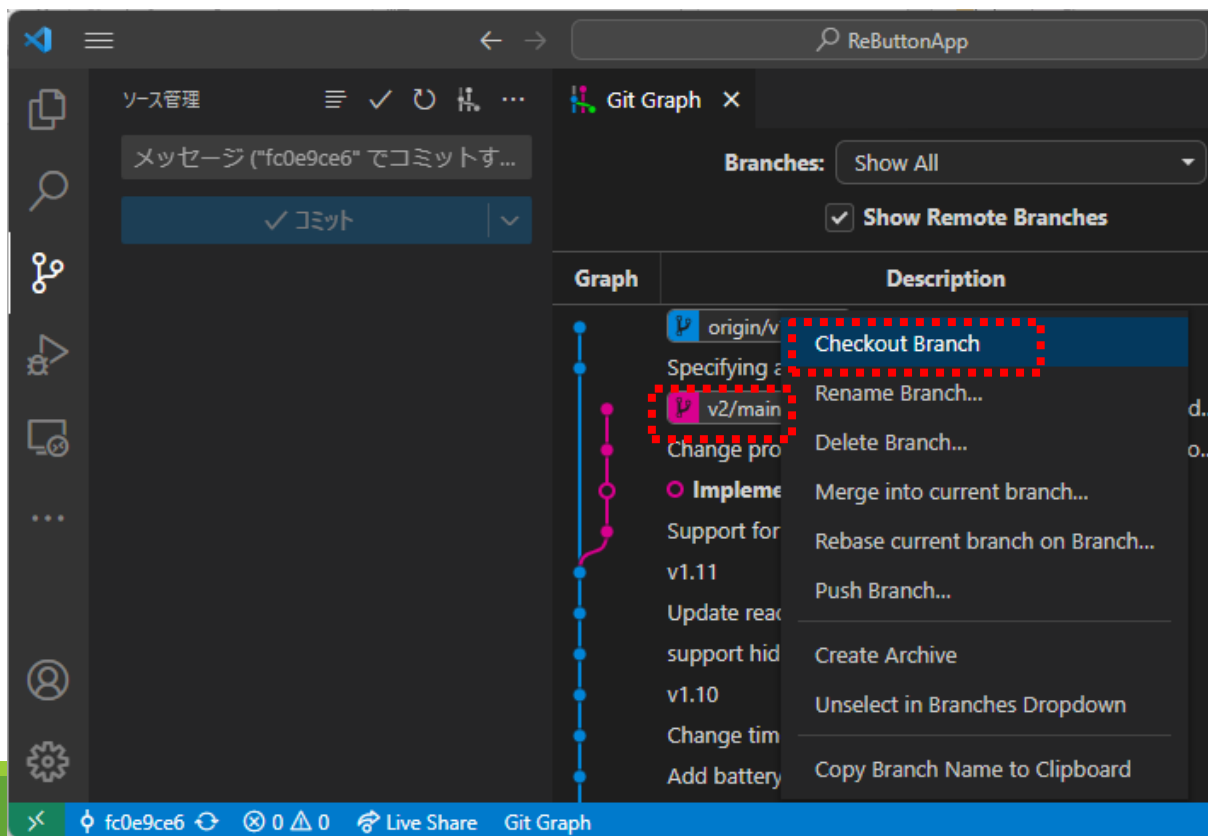
ReButtonApp.inoの変更を記録します。



```
ReButtonApp.ino 3
ReButtonApp.ino > setup()
19 //////////////////////////////////////////////////
20 // Read CONFIG
21
22 ConfigRead();
23
24 Serial.printf("SWEST25 special version is %s.\n",
25
26 Serial.println("Parameters:");
27 Serial.println("-----");
28 ConfigPrint();
29 Serial.println("-----");
30
31 if (ReButton::IsButtonPressed() && ReButton::IsJump)
32 {
33     Serial.println("Force factory reset.");
```

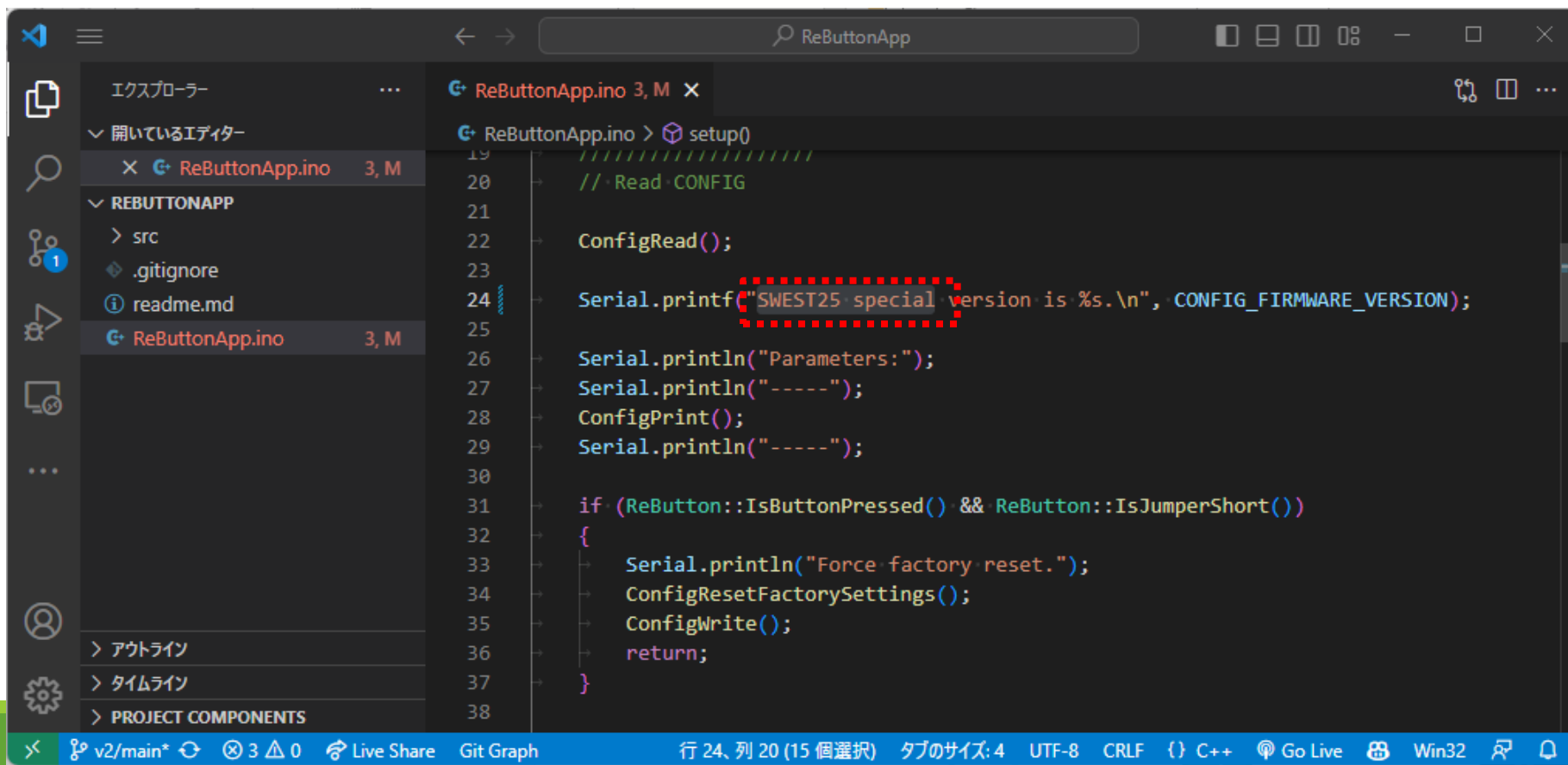
2-1. チェックアウト

1. v2/mainを右クリックして、Checkout Branchを選ぶ
(既にチェックアウトしているときは、Checkout Branchが表示されません)



2-2. 変更

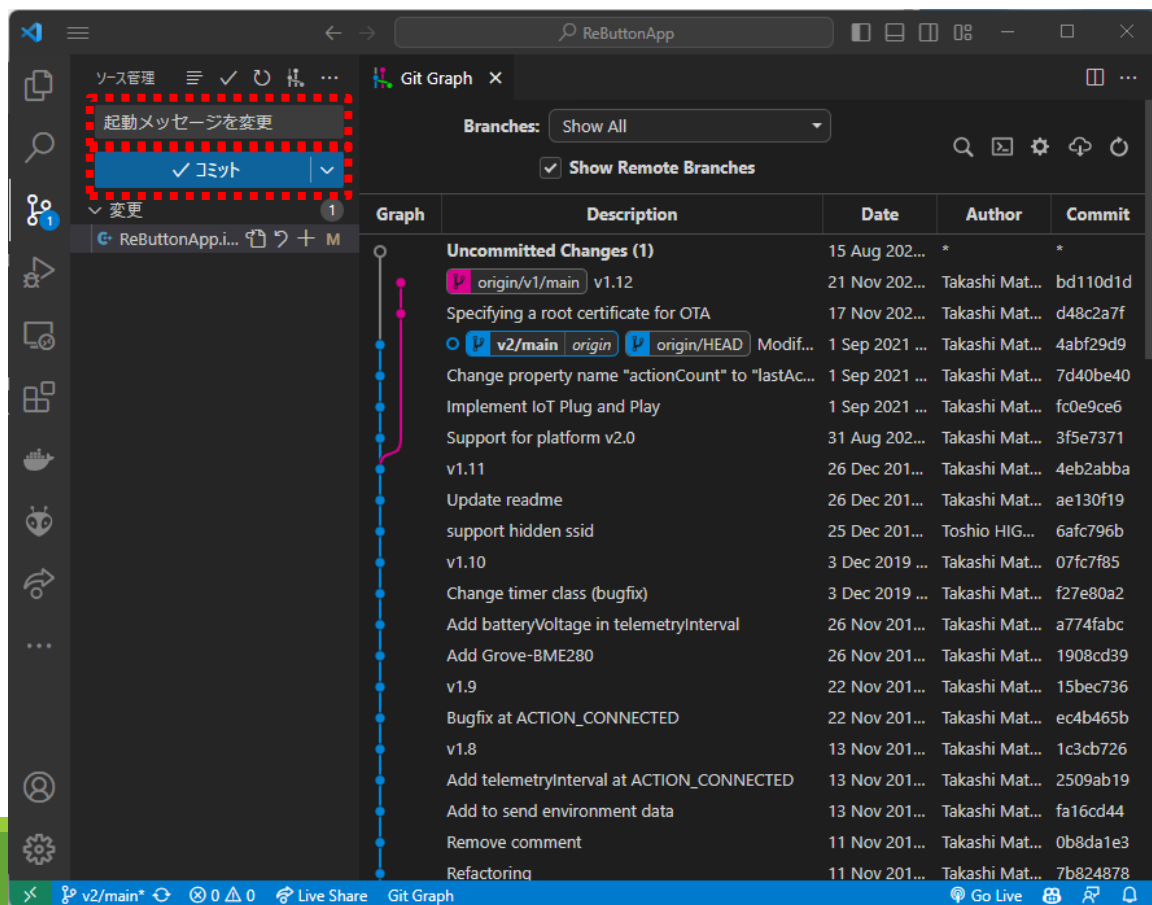
1. ReButtonApp.inoを変更



```
19 //////////////////////////////////////////////////
20 // Read CONFIG
21
22 ConfigRead();
23
24 Serial.printf("SWESt25 special version is %s.\n", CONFIG_FIRMWARE_VERSION);
25
26 Serial.println("Parameters:");
27 Serial.println("-----");
28 ConfigPrint();
29 Serial.println("-----");
30
31 if (ReButton::IsButtonPressed() && ReButton::IsJumperShort())
32 {
33     Serial.println("Force factory reset.");
34     ConfigResetFactorySettings();
35     ConfigWrite();
36     return;
37 }
38
```

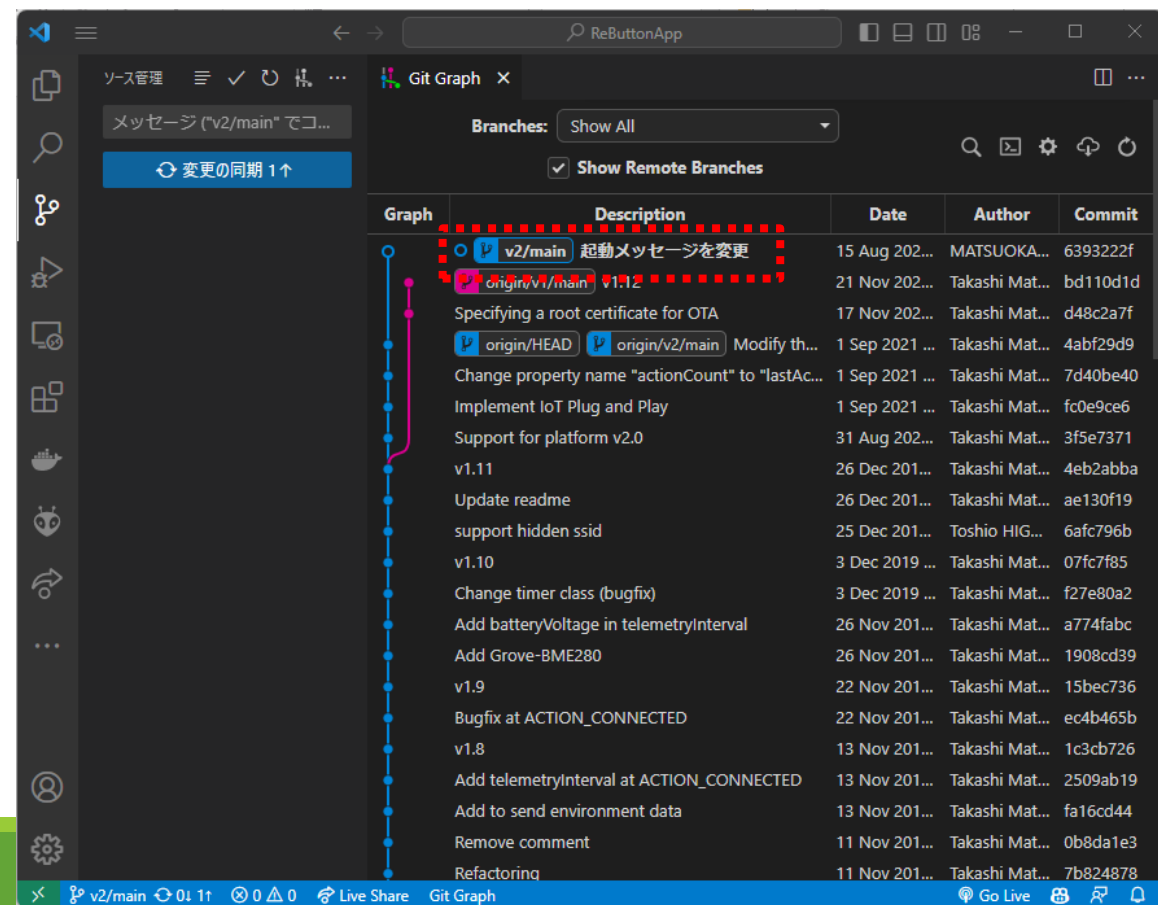
2-3. コミット

1. メッセージを記入してコミット



The screenshot shows the Git Graph interface in Visual Studio Code. The 'Commit' button is highlighted with a red dashed box. The commit message '起動メッセージを変更' is entered in the input field above the button. The commit list shows 'Uncommitted Changes (1)' and a list of previous commits.

Graph	Description	Date	Author	Commit
Uncommitted Changes (1)		15 Aug 202...	*	*
origin/v1/main	v1.12	21 Nov 202...	Takashi Mat...	bd110d1d
	Specifying a root certificate for OTA	17 Nov 202...	Takashi Mat...	d48c2a7f
v2/main	origin	1 Sep 2021 ...	Takashi Mat...	4abf29d9
origin/HEAD	Modif...	1 Sep 2021 ...	Takashi Mat...	7d40be40
	Change property name "actionCount" to "lastAc...	1 Sep 2021 ...	Takashi Mat...	7d40be40
	Implement IoT Plug and Play	1 Sep 2021 ...	Takashi Mat...	fc0e9ce6
	Support for platform v2.0	31 Aug 202...	Takashi Mat...	3f5e7371
	v1.11	26 Dec 201...	Takashi Mat...	4eb2abba
	Update readme	26 Dec 201...	Takashi Mat...	ae130f19
	support hidden ssid	25 Dec 201...	Toshio HIG...	6afc796b
	v1.10	3 Dec 2019 ...	Takashi Mat...	07fc7f85
	Change timer class (bugfix)	3 Dec 2019 ...	Takashi Mat...	f27e80a2
	Add batteryVoltage in telemetryInterval	26 Nov 201...	Takashi Mat...	a774fab
	Add Grove-BME280	26 Nov 201...	Takashi Mat...	1908cd39
	v1.9	22 Nov 201...	Takashi Mat...	15bec736
	Bugfix at ACTION_CONNECTED	22 Nov 201...	Takashi Mat...	ec4b465b
	v1.8	13 Nov 201...	Takashi Mat...	1c3cb726
	Add telemetryInterval at ACTION_CONNECTED	13 Nov 201...	Takashi Mat...	2509ab19
	Add to send environment data	13 Nov 201...	Takashi Mat...	fa16cd44
	Remove comment	11 Nov 201...	Takashi Mat...	0b8da1e3
	Refactoring	11 Nov 201...	Takashi Mat...	7b824878



The screenshot shows the Git Graph interface in Visual Studio Code. The commit message '起動メッセージを変更' is highlighted with a red dashed box. The commit list shows 'Uncommitted Changes (1)' and a list of previous commits. The status bar at the bottom shows 'v2/main' and '0:11'.

Graph	Description	Date	Author	Commit
Uncommitted Changes (1)		15 Aug 202...	MATSUOKA...	6393222f
v2/main	起動メッセージを変更	15 Aug 202...	MATSUOKA...	6393222f
origin/v1/main	v1.12	21 Nov 202...	Takashi Mat...	bd110d1d
	Specifying a root certificate for OTA	17 Nov 202...	Takashi Mat...	d48c2a7f
origin/HEAD	origin/v2/main	1 Sep 2021 ...	Takashi Mat...	4abf29d9
	Change property name "actionCount" to "lastAc...	1 Sep 2021 ...	Takashi Mat...	7d40be40
	Implement IoT Plug and Play	1 Sep 2021 ...	Takashi Mat...	fc0e9ce6
	Support for platform v2.0	31 Aug 202...	Takashi Mat...	3f5e7371
	v1.11	26 Dec 201...	Takashi Mat...	4eb2abba
	Update readme	26 Dec 201...	Takashi Mat...	ae130f19
	support hidden ssid	25 Dec 201...	Toshio HIG...	6afc796b
	v1.10	3 Dec 2019 ...	Takashi Mat...	07fc7f85
	Change timer class (bugfix)	3 Dec 2019 ...	Takashi Mat...	f27e80a2
	Add batteryVoltage in telemetryInterval	26 Nov 201...	Takashi Mat...	a774fab
	Add Grove-BME280	26 Nov 201...	Takashi Mat...	1908cd39
	v1.9	22 Nov 201...	Takashi Mat...	15bec736
	Bugfix at ACTION_CONNECTED	22 Nov 201...	Takashi Mat...	ec4b465b
	v1.8	13 Nov 201...	Takashi Mat...	1c3cb726
	Add telemetryInterval at ACTION_CONNECTED	13 Nov 201...	Takashi Mat...	2509ab19
	Add to send environment data	13 Nov 201...	Takashi Mat...	fa16cd44
	Remove comment	11 Nov 201...	Takashi Mat...	0b8da1e3
	Refactoring	11 Nov 201...	Takashi Mat...	7b824878

2-4. (ファイルをステージング)

Stagedが無いときは、Modifiedがコミットされます。(VSCodeの振る舞い)

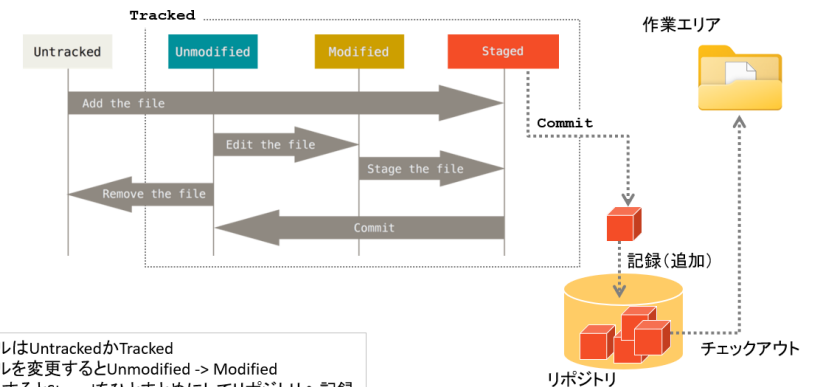
コミット対象

変更 1
ReButtonApp.i... M

ステージされている変更 1
ReButtonApp.ino M

変更 2
readme.md M
ReButtonApp.ino

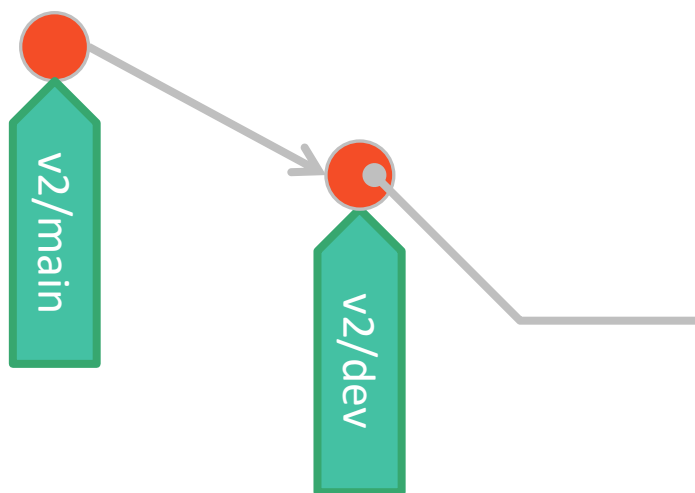
未変更 / 変更 / ステージとコミット、チェックアウト



- ✓ ファイルはUntrackedかTracked
- ✓ ファイルを変更するとUnmodified -> Modified
- ✓ コミットするとStagedをひとまとめにしてリポジトリへ記録
- ✓ コミットを作業エリアへ展開するときはチェックアウト

3. 分岐

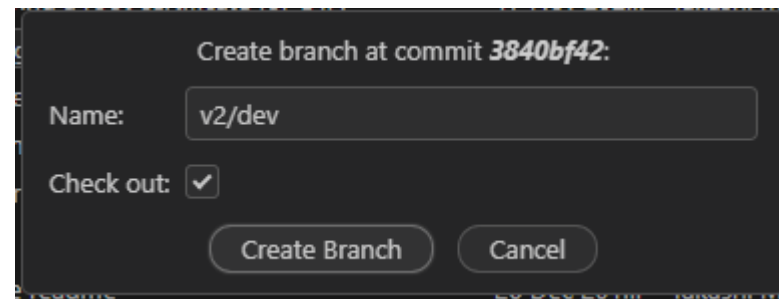
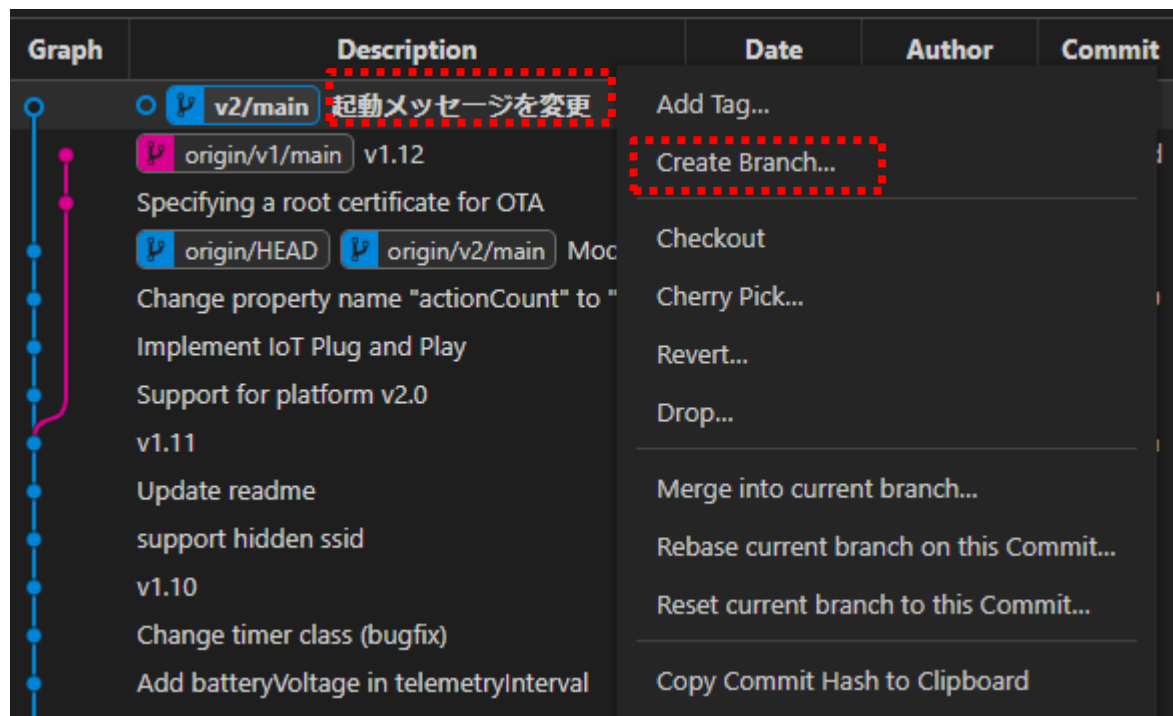
v2/mainを分岐して、v2/devへ変更を記録します。



```
ReButtonApp.ino 3, M x
ReButtonApp.ino > setup()
19 //////////////////////////////////////////////////
20 // Read CONFIG
21
22 ConfigRead();
23
24 Serial.printf("SWEST25 dev version is %s.\n", CONFIG_FIRM
25
26 Serial.println("PARAMETERS");
27 Serial.println("-----");
28 ConfigPrint();
29 Serial.println("-----");
30
31 if (ReButton::IsButtonPressed() && ReButton::IsJumperShor
32 {
33     Serial.println("Force factory reset.");
34     ConfigResetFactorySettings();
35     ConfigWrite();
36     return;
37 }
38
////////////////////////////////////
```

3-1. ブランチを作成

1. 対象のコミットを右クリックして、Create Branchを選ぶ



3-2. 変更してコミット

1. ReButtonApp.inoを変更してコミット

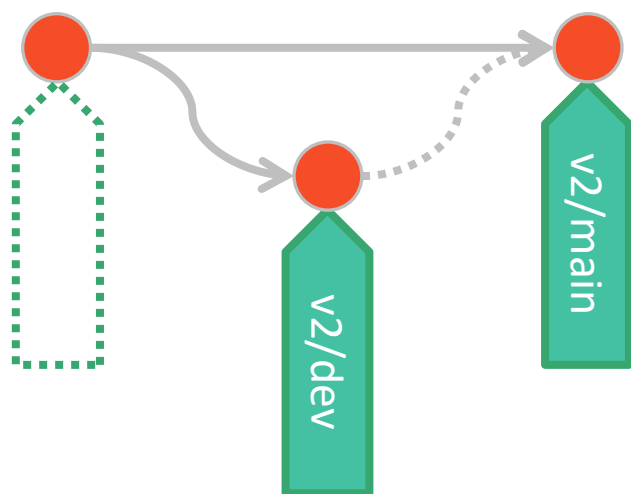
```
ReButtonApp.ino 3, M X
ReButtonApp.ino > setup()
19 //////////////////////////////////////////////////
20 // Read CONFIG
21
22 ConfigRead();
23
24 Serial.printf("SWEST25 dev version is %s.\n", CONFIG_FIRM
25
26 Serial.println("PARAMETERS");
27 Serial.println("-----");
28 ConfigPrint();
29 Serial.println("-----");
30
31 if (ReButton::IsButtonPressed() && ReButton::IsJumperShor
32 {
33     Serial.println("Force factory reset.");
34     ConfigResetFactorySettings();
35     ConfigWrite();
36     return;
37 }
38
```



Graph	Description
	v2/dev 大文字に変更
	v2/main 起動メッセージを変更
	origin/v1/main v1.12
	Specifying a root certificate for OTA
	origin/HEAD origin/v2/main Modify ...
	Change property name "actionCount" to "last...
	Implement IoT Plug and Play
	Support for platform v2.0
	v1.11
	Update readme

4. マージ

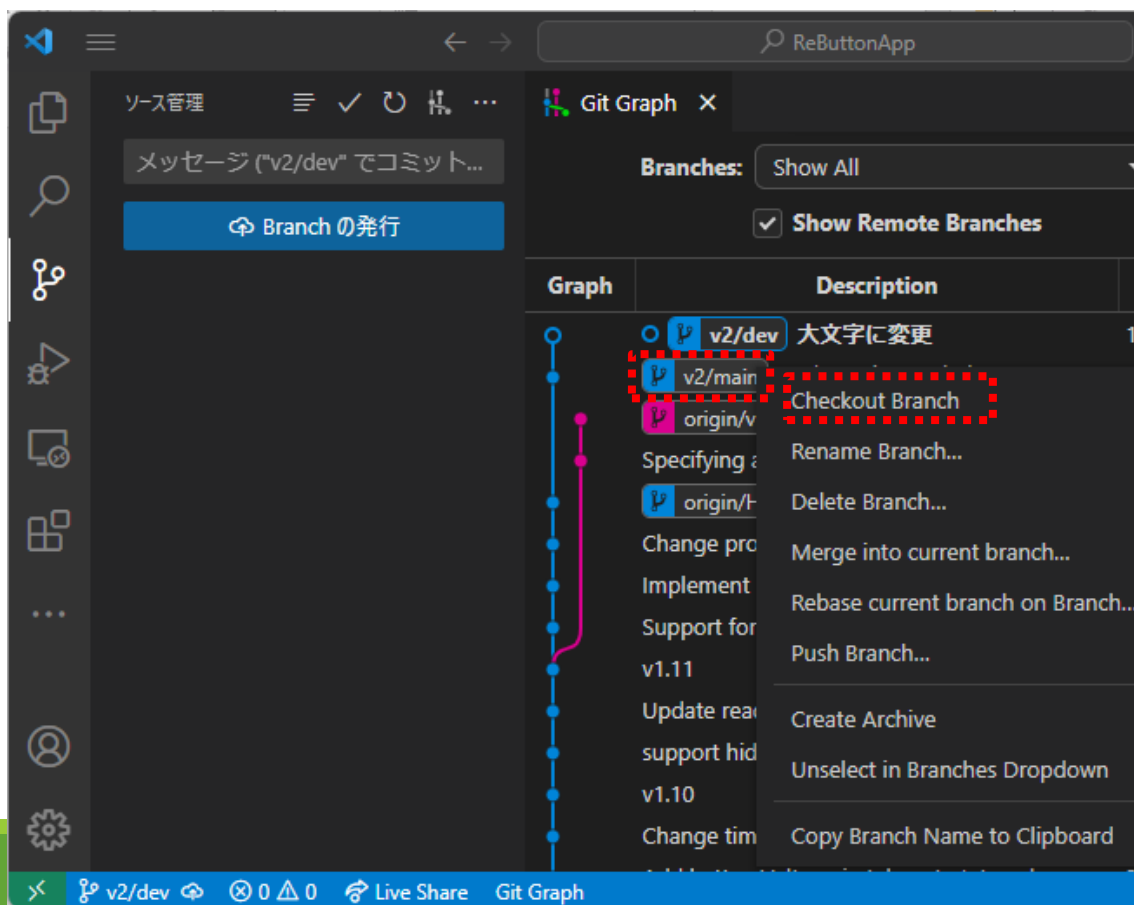
v2/devをv2/mainへマージします。



```
ReButtonApp.ino 3, M x
ReButtonApp.ino > setup()
19 //////////////////////////////////////////////////
20 // Read CONFIG
21
22 ConfigRead();
23
24 Serial.printf("SWEST25:dev version is %s.\n", CONFIG_FIRM
25
26 Serial.println("PARAMETERS:");
27 Serial.println("-----");
28 ConfigPrint();
29 Serial.println("-----");
30
31 if (ReButton::IsButtonPressed() && ReButton::IsJumperShor
32 {
33     Serial.println("Force factory reset.");
34     ConfigResetFactorySettings();
35     ConfigWrite();
36     return;
37 }
38
39 //////////////////////////////////////////////////
```

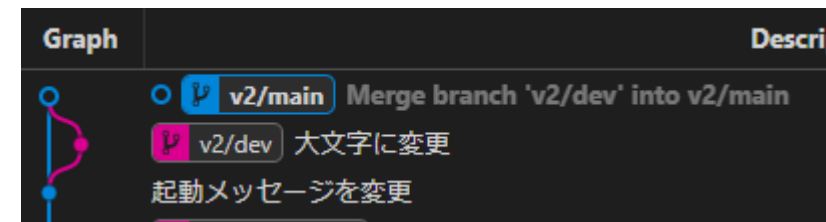
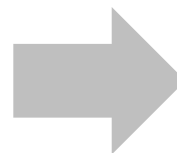
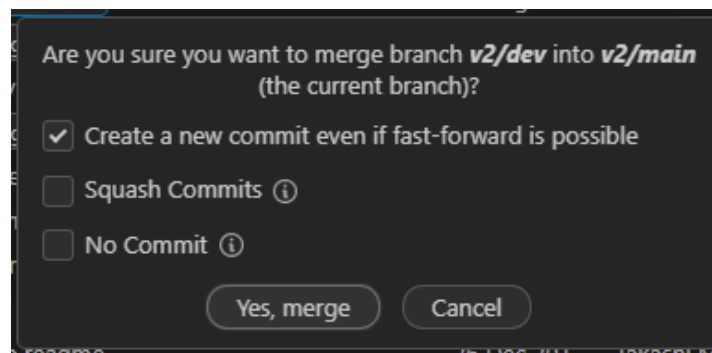
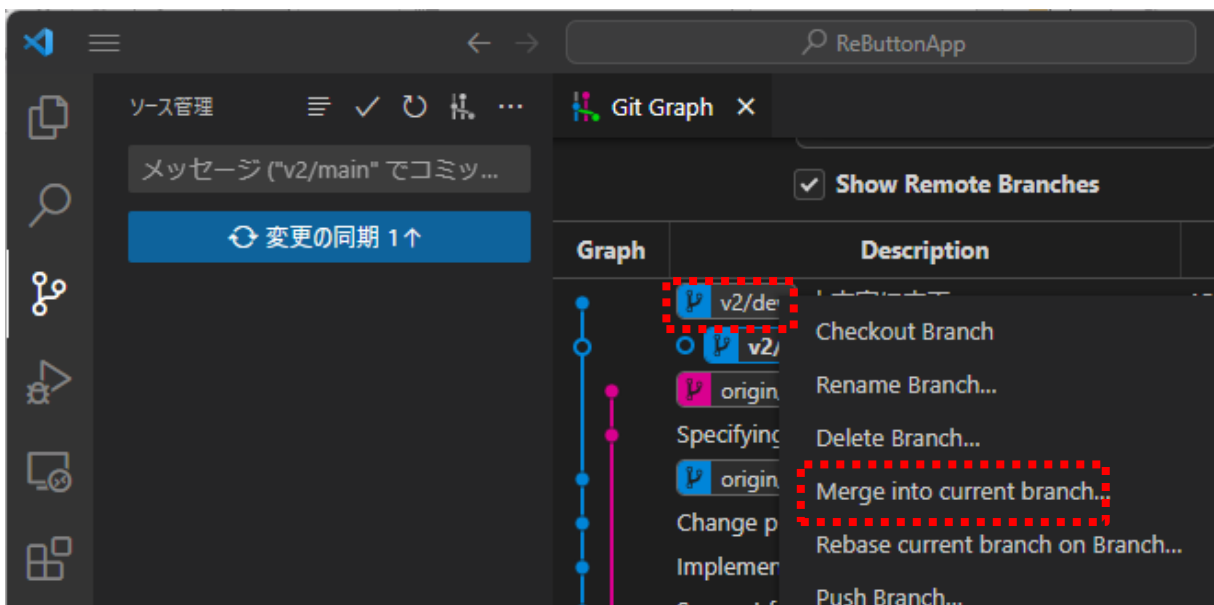
4-1. チェックアウト

1. 変更するブランチ(v2/main)をチェックアウト



4-2. マージ

1. 変更内容のブランチ(v2/dev)を右クリックして、Merge into current branchを選ぶ



Lab3 まとめ

1. 作業エリア/リポジトリ/追跡
 - ✓ mkdir
 - ✓ git init / git clone
 - ✓ .gitignore

2. 変更とコミット
 - ✓ チェックアウト
 - ✓ ステージング
 - ✓ コミット

変更の記録

3. 分岐
 - ✓ ブランチ作成
4. マージ
 - ✓ チェックアウト
 - ✓ マージ

変更の関係

Lab4 応用

1. ブランチ戦略

2. コーディングスタイル

3. ファイル分割

4. ステージングの使い方

5. コミットの頻度

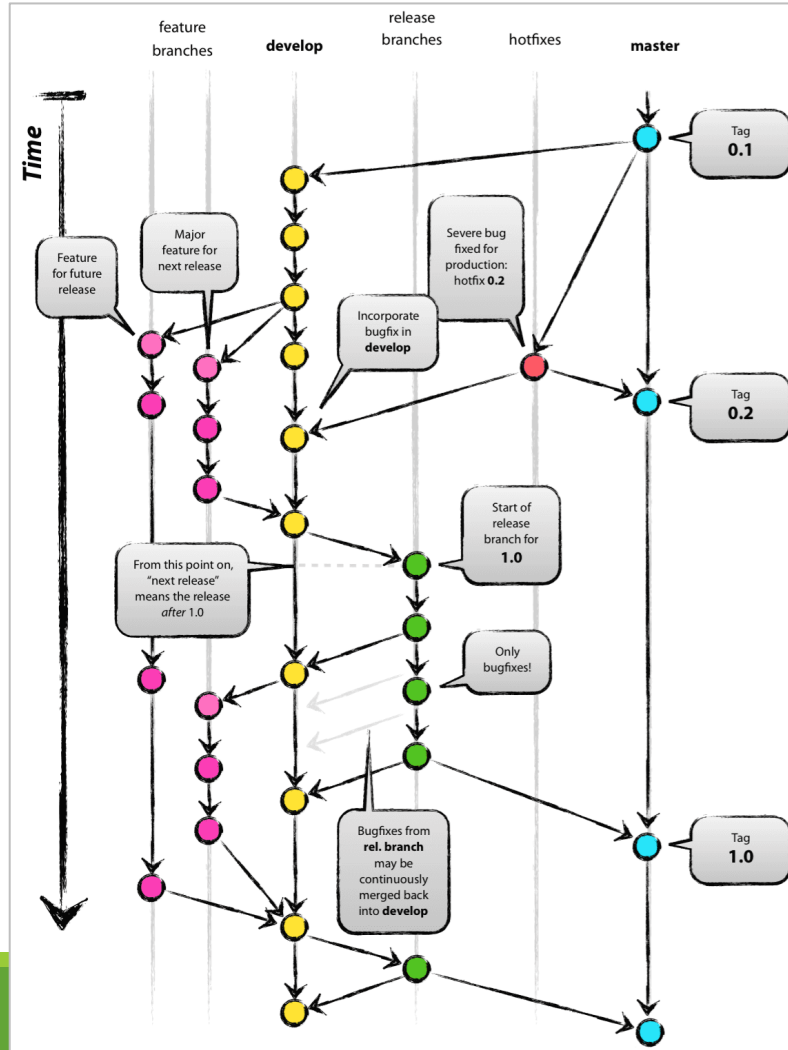
6. 変更履歴の分析

7. バックアップ

Gitに適した
運用ルール大事



1. ブランチ戦略



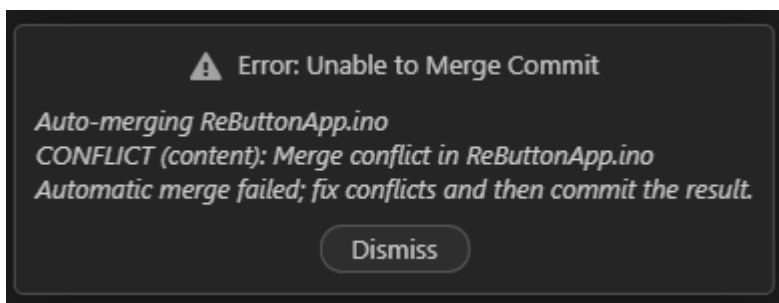
- ✓ 主軸は「main」ブランチ
- ✓ 開発中は「dev」ブランチ
- ✓ 要望/不具合毎にブランチを追加
- ✓ 早急に「main」ブランチへマージ
- ✓ (変更の関係を編集する機会を減らす)

2. コーディングスタイル

✓ 不要なコードや
ファイルは削除

```
571 ..... this.unitLaneL.Saving = this.unitLaneL.Saving || this.unitLaneL.JudgeGone
572 ..... }
573 ..... }
574 ..... else if (this.unitLaneR.Enabled)
575 ..... {
576 .....     this.unitLaneR.Saving = this.unitLaneR.Saving || this.unitLaneR.JudgeGone
577 ..... }
578 ..... else if (this.unitLaneL.Enabled)
579 ..... {
580 .....     this.unitLaneL.Saving = this.unitLaneL.Saving || this.unitLaneL.JudgeGone
581 ..... }
582 .....
583 ..... // 右レーンの保存処理
584 ..... if (this.unitLaneR.Saving && !this.unitLaneR.Saved)
585 ..... {
586 .....     DataRow dataRow = this.resultTable.NewRow(this.unitLaneR);
587 .....     this.storage.BackupAdd(dataRow);
588 .....     this.resultTableToday.Add(this.resultTableToday.NewRow(this.unitLaneR));
589 .....     if (this.chkTimeSheetPrintOut.Checked)
590 .....     {
591 .....         using (PrintDocument timeSheet = TimeSheetFactory.CreateTimeSheet(dat
592 .....         {
593 .....             timeSheet.Print();
594 .....         }
595 .....     }
596 .....     if (this.chkResultRecord.Checked)
597 .....     {
598 .....         this.resultTable.Add(dataRow);
599 .....         this.resultTableSaved = false;
600 .....         this.storage.MonitorResultTableSave(this.resultTable);
601 .....     }
602 .....
603 .....     this.unitLaneR.Saved = true;
604 ..... }
605 .....
606 ..... // 左レーンの保存処理
607 ..... if (this.unitLaneL.Saving && !this.unitLaneL.Saved)
608 ..... {
609 .....     DataRow dataRow = this.resultTable.NewRow(this.unitLaneL);
610 .....     this.storage.BackupAdd(dataRow);
611 .....     this.resultTableToday.Add(this.resultTableToday.NewRow(this.unitLaneL));
612 .....     if (this.chkTimeSheetPrintOut.Checked)
613 .....     {
614 .....         using (PrintDocument timeSheet = TimeSheetFactory.CreateTimeSheet(dat
```

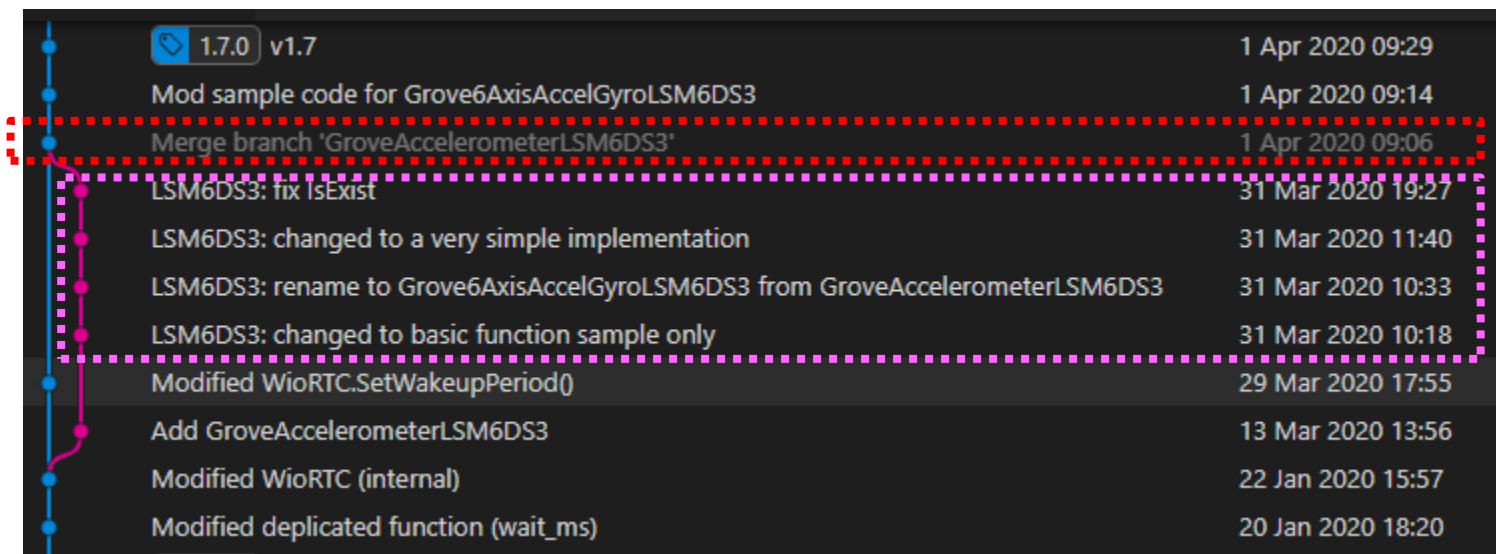

3. ファイル分割



✓ ファイルを分割

受信中 ϕ b3ab341 \cdot v2/dev	現在のマシ ϕ 8f7dbfe \cdot HEAD
22 \rightarrow ConfigRead(); 23 24 \rightarrow Serial.printf("SWEST25 dev version is %s.\n", CONFIG_FIRMWARE_VERSION); 25 26 \rightarrow Serial.println("PARAMETERS:"); 27 \rightarrow Serial.println("-----"); 28 \rightarrow ConfigPrint(); 29 \rightarrow Serial.println("-----"); 30 31 \rightarrow if (ReButton::IsButtonPressed() && ReButton::IsJumperShort()) 32 {	22 \rightarrow ConfigRead(); 23 24 \rightarrow Serial.printf("Firmware version is %s.\n", CONFIG_FIRMWARE_VERSION); 25 26 \rightarrow Serial.println("Parameters:"); 27 \rightarrow Serial.println("====="); 28 \rightarrow ConfigPrint(); 29 \rightarrow Serial.println("====="); 30 31 \rightarrow if (ReButton::IsButtonPressed() && ReButton::IsJumperShort()) 32 {
Result ReButtonApp.ino	1 Conflict Remaini
22 \rightarrow ConfigRead(); 23 24 \rightarrow Serial.printf("SWEST25 dev version is %s.\n", CONFIG_FIRMWARE_VERSION); 25 26 \rightarrow Serial.println("Parameters:"); 27 \rightarrow Serial.println("-----"); 28 \rightarrow ConfigPrint(); 29 \rightarrow Serial.println("====="); 30	

4. コミットの頻度



- ✓ devブランチは細かく
- ✓ mainブランチはまとめて

5. ステージングの使い方

- ✓ コンパイルが通ったらStaged
- ✓ 動作確認したらコミット

The screenshot displays the Visual Studio Code source control interface for a project named 'ReButtonApp'. The left sidebar shows the 'SOURCE CONTROL' view with a list of files: 'readme.md' (M), 'ReButtonApp.ino' (3, M), and another 'ReButtonApp.ino' (3, M). The main editor area is split into two panels. The top panel, labeled 'Staged', shows the code for 'ReButtonApp.ino' with a pink dashed box highlighting the line 'Serial.println("Modify 1:");'. The bottom panel, labeled 'Modified', shows the same code with a red dashed box highlighting the same line. The code in both panels is identical, showing a configuration read function and a button press handler.

6. 変更履歴の分析

- ✓ ツール多数(この例は、GitLens)
- ✓ 行ごとの履歴

The screenshot displays the Visual Studio Code interface with the GitLens extension. The main editor shows the code for `ReButtonApp.ino` with a diff view highlighting changes. The left sidebar contains several panels:

- GITLENS INSPECT**: Shows commit details for commit `b3ab341` by `You`, committed 3 hours ago. The commit message is `大文字に変更`.
- FILES CHANGED**: Shows `ReButtonApp.ino` as the only file changed.
- LINE HISTORY**: Shows a list of changes to line 24 of `ReButtonApp.ino`, including a commit by `You` 3 hours ago and a commit by `Takashi Matsuoka` 5 years ago.
- COMMIT GRAPH: REBUTTONAPP**: Shows a commit graph with branches `v2/main`, `v2/dev`, `v1/main`, and `v2/main`. The current commit is highlighted in red.
- COMMIT GRAPH DETAILS**: Shows details for the current commit, including the commit message `大文字に変更` and the author `You`.

The main editor shows the following code snippet with a diff view:

```
17  
18  
19  
20  
21  
22  
23  
24 大文字に変更 3 時間前  
25 v1.0 5 年前  
26 大文字に変更 3 時間前  
27 v1.0 5 年前  
28  
29  
30  
31 Change step to begin factory reset 4 年前  
32  
33  
34  
35  
36  
DisplayBegin();  
////////////////////////////////////  
// Read CONFIG  
ConfigRead();  
Serial.printf("SWEST25 dev version is %s.\n", CONFIG_FIRMWARE_VERSION);  
Serial.println("PARAMETERS:");  
Serial.println("-----");  
ConfigPrint();  
Serial.println("-----");  
if (ReButton::IsButtonPressed() && ReButton::IsJumperShort())  
{  
    Serial.println("Force factory reset.");  
    ConfigResetFactorySettings();  
    ConfigWrite();  
    return;  
}
```


7. バックアップ

- ✓ GitHubのプライベートリポジトリにバックアップ
- ✓ コード漏洩のリスクあり？ <https://zenn.dev/miyajan/scraps/3567cee380280c>

Free
個人向けの基本プラン

\$0 年間費用は永久に無料

Freeプランを始める

- > 無制限のパブリック/プライベートリポジトリ
- > セキュリティとバージョンの自動更新
- > CI/CD時間：2,000分/月
パブリックリポジトリでの利用は無料
- > 500MBのGitHub Packagesストレージ
パブリックリポジトリでの利用は無料
- > 新しいIssueとプロジェクト (限定ベータ版)
- > GitHubコミュニティによるサポート

オプション

- > GitHub Copilot Access

<https://github.co.jp/pricing>

