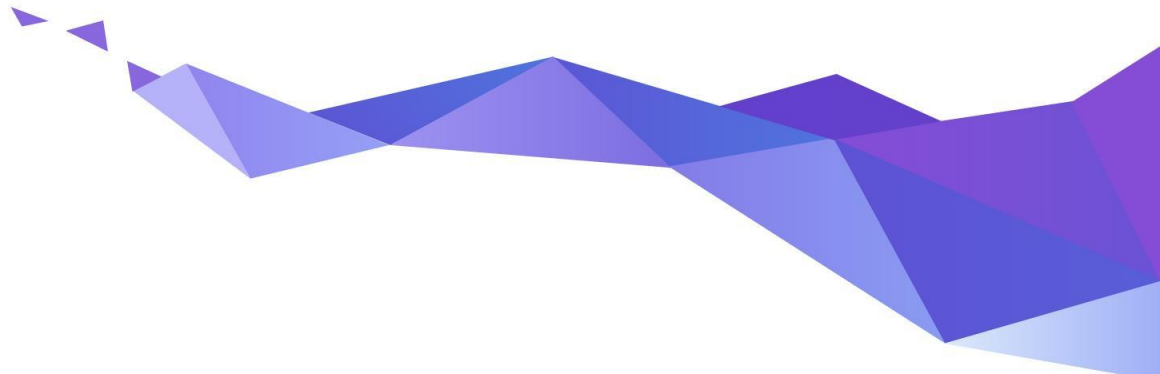


最近のArduinoとZephyr

TOKITA Hiroshi

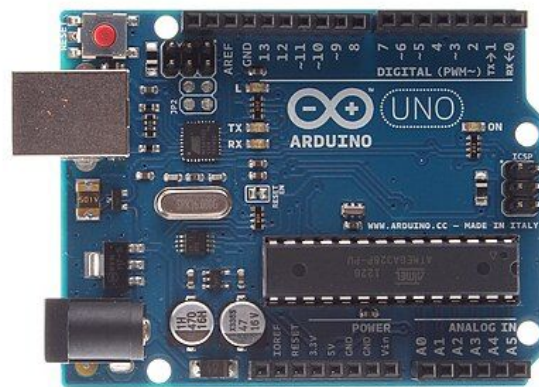


Arduino

電子工作向け(元々はインタラクティブ・アートの教材)のマイコンボード。

初代はAtmel AVRマイコンで、様々なバリエーションがある。

2021年にArduino UNOが1000万台出荷。とてもユーザーの多いプラットフォーム。



Arduino と Zephyr

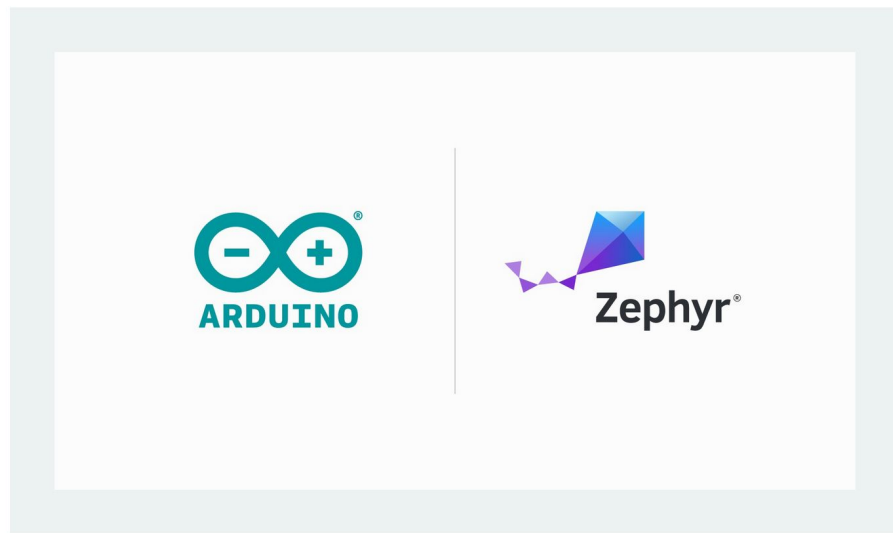
Arduinoの一部機種で使っていたMbed (ARMによる教育用RTOS)の終息に伴い、Zephyrベースの開発がアナウンスされた。

(2024年のニュース)



The end of Mbed marks a new beginning for Arduino

Posted by: ARDUINO TEAM — July 24th, 2024



Arduino UNO Q



2025/10のQualcommによる買収に合わせて発表された
Arduino UNO QでZephyrベースのArduinoが正式にリリースに。

ARDUINO
UNO Q

From blink to think.

Arduino UNO Q bridges **high-performance computing with real-time control.**

Get power and ease of use – all wrapped up into UNO.

[BUY NOW](#)

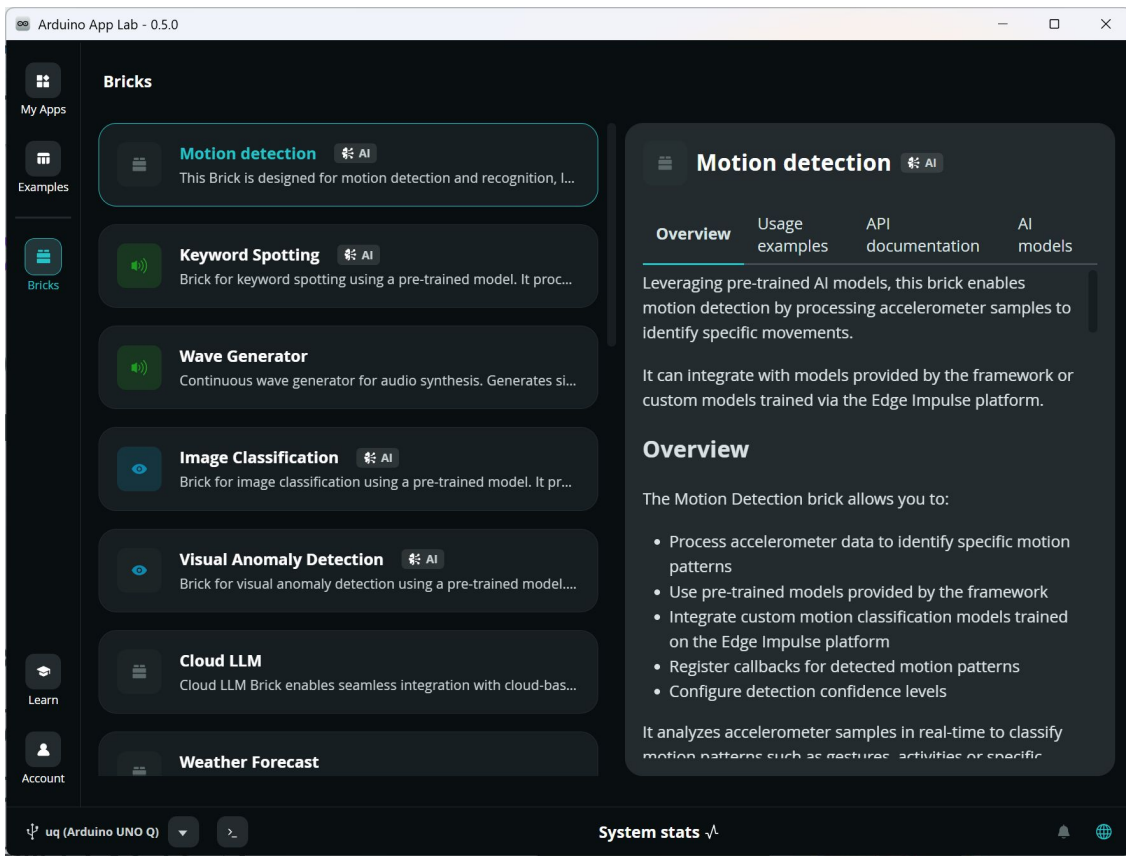


CPU2台持ち

- Qualcomm Dragonwing™ QRB2210
 - Quad-core ARM Cortex-A53 CPU (up to 2.0 GHz)
 - Adreno 702 GPU
 - Dual 13 MP Image Signal Processors (ISPs)
 - Inferencing - AI models can run on the GPU and CPU
 - Always-on Qualcomm® Hexagon™ DSP
 - Debian Linux
- STM32U585 Arm® Cortex®-M33 32-bit MCU
 - 2 MB flash memory
 - 786 kB SRAM
 - Zephyr-based ArduinoCore
- 10年前のスマホ+マイコンという構成

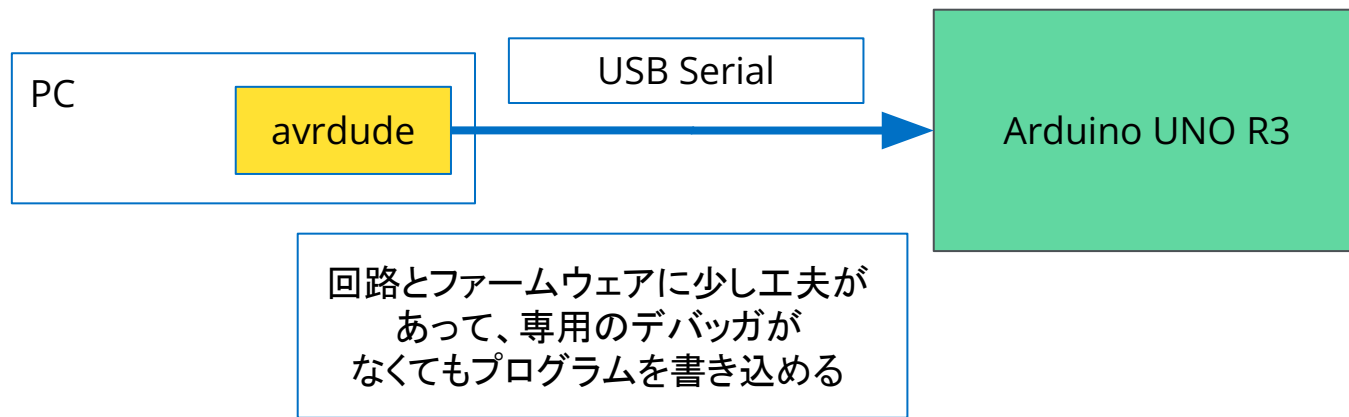
Arduino App Lab

UNO Qに合わせて新しい
アプリ開発環境のArduino
App Labも提供。
Linux/Arduino/Webの連
携したexamplesが入って
いる。
AI機能のライブラリが特徴
的。



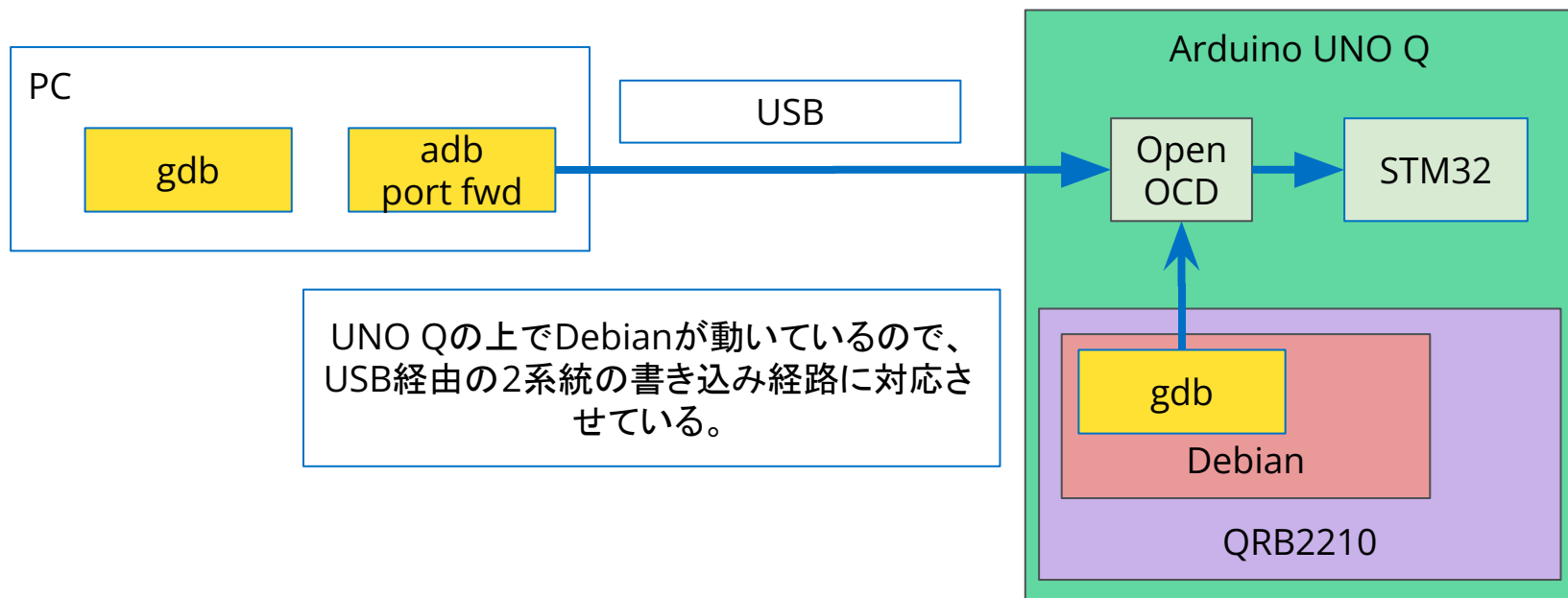
プログラムの書き込み (Traditional)

USBをつないで「書き込み」ボタンを押すだけでプログラムが動くのが特徴。Non-technicalユーザーに優しい。



UNO Qのプログラム書き込み

いろいろ複雑になったが、USBでつないでボタン一つ、は維持。



UNO Qの上でDebianが動いているので、USB経由の2系統の書き込み経路に対応させている。

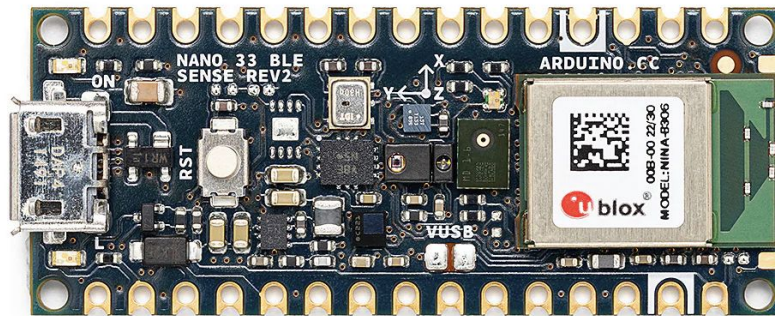
ArduinoCore-API

- GPIO操作など、基本的なマイコンの操作のAPI群。
 - ポインタ操作を排除したNon-technicalユーザーに優しい設計。
- これが動けばだいたいArduino。エコシステムの起点。
 - みんな使ってるので下手にいじれなくなった部分を集めたデファクトスタンダード的な安定API。
- 新しいマイコンを採用するたびに作り直されてきた。
 - AVR, MegaAVR: Atmel(現MicroChip) のAVRシリーズ。原点。
 - SAM, SAMD: MicroChipのSAMシリーズ(ARMcore)向け。
 - Renesas: Renesas RAシリーズ(ARMcore)向け。
 - Mbed: ARMのRTOSのMbedを使ったもの。STM, Nordicなど。

NordicさんのSoCも

Arduino Nano 33 BLE

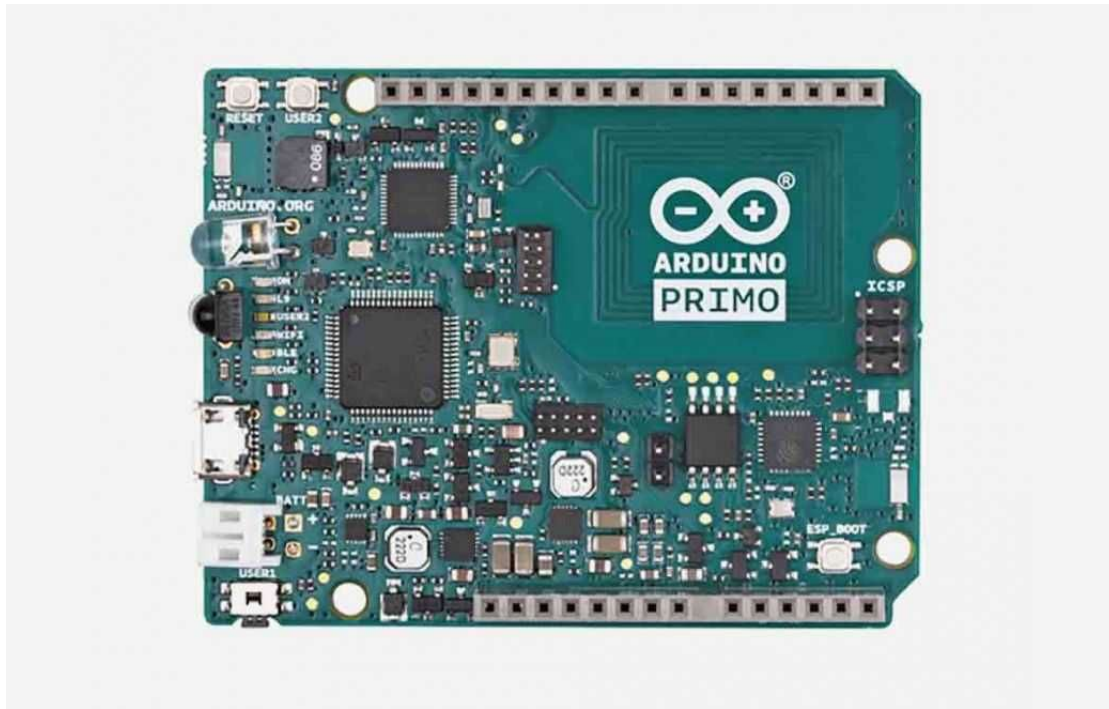
(およびそのバリエーション)



Arduino Primo

Nrf52832

(とSTM32とESP8266)



ArduinoCore-zephyr



- 車輪の再発明の最新版。
- Short history
 - 2019: PoCとして Arduino-on-zephyr を作る
 - 2022: GSoC2022で Dhruva Gole(現TI India)が現行実装のベースを作成
 - 2024: Arduino ProjectがFork作成、製品向けの開発が始まる
 - 2025: UNO Q向けに(BETAがついてない)リリース

現在のリスト



arduino_giga_r1
arduino_nano_33_ble
arduino_nano_33_ble_nrf52840_sense
arduino_nano_matter
arduino_nicla_sense
arduino_opta
arduino_portenta
arduino_portenta
arduino_uno_q
ek_ra8d1
frdm_mcxn947
frdm_rw612

beagleconnect
cc3220sf
nrf52840dk
nrf9160dk
rpi_pico
arduino_mkrzero
arduino_nano_33_iot

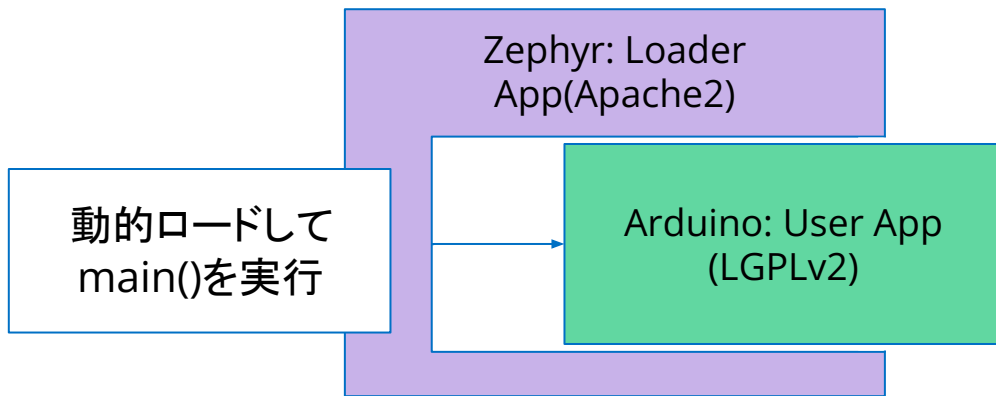
Arduinoでのモジュール構成



- ZephyrのLLEXT(DLLのような動的ロード)機能を使っている。
- このため、DLL向けにExportしている機能しか使えない。
- なぜこんな面倒なことを？

LGPLv2 vs Apache2

- ArduinoCore-APIは LGPLv2ライセンス
- ZephyrはApache2ライセンス
- 相互のライセンス尊重のため
LGPLv2の部分を動的ロードモジュールに閉じ込めている。



現状での課題

- Zephyr本体にArduinoの構成定義に使える定義あるのに使えていない。
- 設定追加がとても面倒。→対応ボード増やすのが億劫。
- デフォルト状態でも使えるようにする必要がある。

- LGPLv2との非互換 → 再実装するしかない？

構想



```
service HardwareSerial {  
    rpc BeginBaud(google.protobuf.UInt64Value) returns (google.protobuf.Empty) {  
        option (arduino.cpp_name) = "begin";  
        option (arduino.cpp_arg_types) = "unsigned long";  
    }  
    ...  
}
```

ProtoBufの定義からヘッダを生成して互換実装を作成。

DeviceTree: Nexus

DeviceTreeにはNexusノードというAlias作成のための機能がある。

2.5. Nexus Nodes and Specifier Mapping

<https://devicetree-specification.readthedocs.io/en/latest/chapter2-device-tree-basics.html#nexus-nodes-and-specifier-mapping>

Nexusノードの例



```
soc {
    soc_gpio1: gpio-controller1 {
        #gpio-cells = <2>;
    };

    soc_gpio2: gpio-controller2 {
        #gpio-cells = <2>;
    };
};
```

```
connector: connector {
    #gpio-cells = <2>;
    gpio-map = <0 0 &soc_gpio1 1 0>,
               <1 0 &soc_gpio2 4 0>,
               <2 0 &soc_gpio1 3 0>,
               <3 0 &soc_gpio2 2 0>;
    gpio-map-mask = <0xf 0x0>;
    gpio-map-pass-thru = <0x0 0x1>;
};
```

```
expansion_device {
    reset-gpios = <&connector 2 GPIO_ACTIVE_LOW>;
};
```

connector:0	soc_gpio1:1
connector:1	soc_gpio2:4
connector:2	soc_gpio2:4
connector:3	soc_gpio2:2

のような変換テーブルになる

Zephyrでの扱い

- ZephyrではDeviceTreeからそれに相当するCのマクロ定義を生成する(承前)
- Nexusノードについては、変換ルールとして適用して、Nexusノード自体は出力されていなかった。
- つまり、コード上からはNexusノードの情報は参照できない状態だった。

- 気づいたのは6年前。その間誰も必要としなかったw

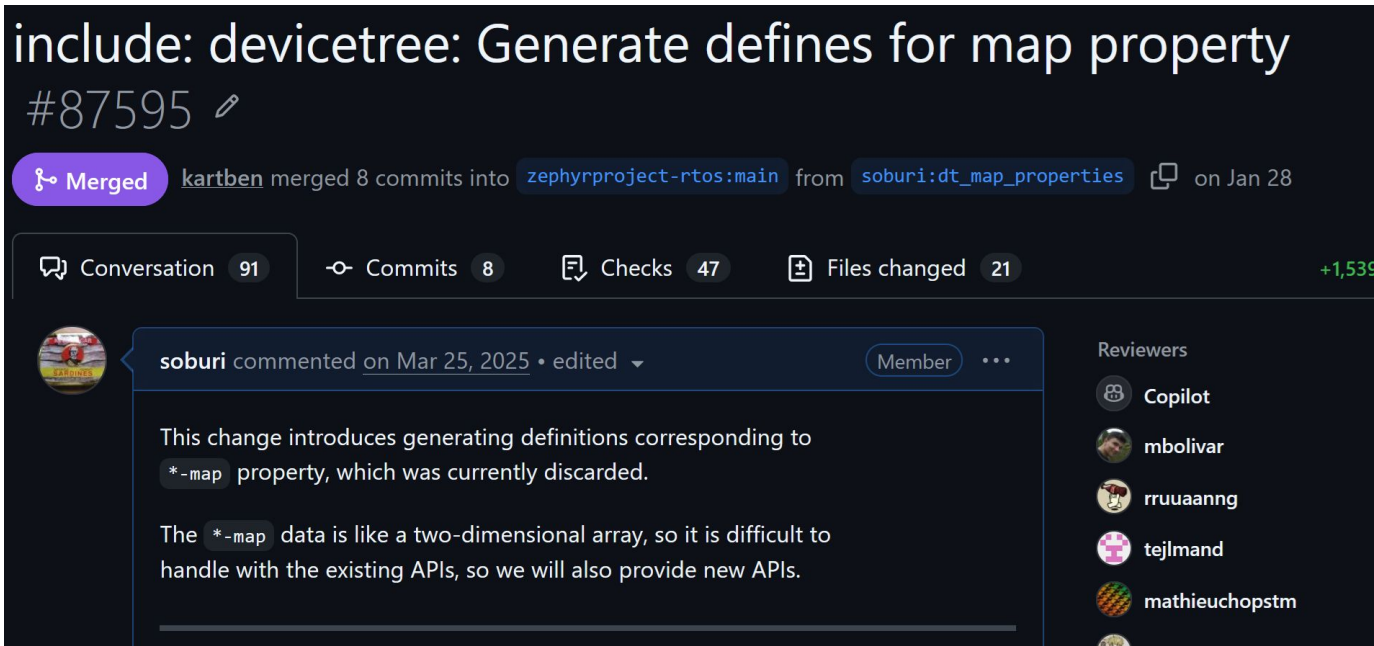
実例






```
arduino_header: connector {  
    compatible = "arduino-header-r3";  
    #gpio-cells = <2>;  
    gpio-map-mask = <0xffffffff 0xffffffffc0>;  
    gpio-map-pass-thru = <0 0x3f>;  
    gpio-map = <ARDUINO_HEADER_R3_A0 0 &ioport0 14 0>,  
              <ARDUINO_HEADER_R3_A1 0 &ioport0 0 0>,  
              <ARDUINO_HEADER_R3_A2 0 &ioport0 1 0>,  
              /***** snip *****/,  
              <ARDUINO_HEADER_R3_D15 0 &ioport1 0 0>;  
};
```





ZephyrではArduino形状のコネクタの定義が標準化されている
読めればこれからピン名の定義(D0, D1, ...)を決定できる。




PR提出してからマージされるまで8か月かかりました。
2次元表をプリプロセッサで処理する「キモい」マクロ



include: devicetree: Generate defines for map property
#87595 

 Merged [kartben](#) merged 8 commits into `zephyrproject-rtos:main` from `soburi:dt_map_properties`  on Jan 28






 Conversation 91  Commits 8  Checks 47  Files changed 21 +1,539

 **soburi** commented on [Mar 25, 2025](#) • edited  Member 

This change introduces generating definitions corresponding to `*-map` property, which was currently discarded.

The `*-map` data is like a two-dimensional array, so it is difficult to handle with the existing APIs, so we will also provide new APIs.

Reviewers

-  Copilot
-  mbolivar
-  rruaanng
-  tejlmand
-  mathieuchopstm

結果

- Arduino_headerが定義されているボードは現在236。
- これがサポートできれば今までのArduinoCoreでは最多のサポート数で、**Everything became Arduino**が視野に入る。
- あとはライセンス問題。

Thank you

