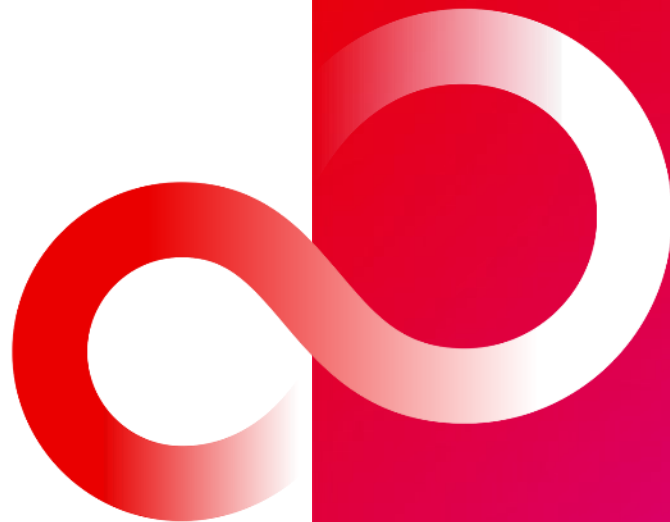


Supporting Hobbyist Friendly OpenSource Hardware in Zephyr

TOKITA Hiroshi

Linux Appliance Engineering Department
Data Center Division
Infrastructure System Business Unit

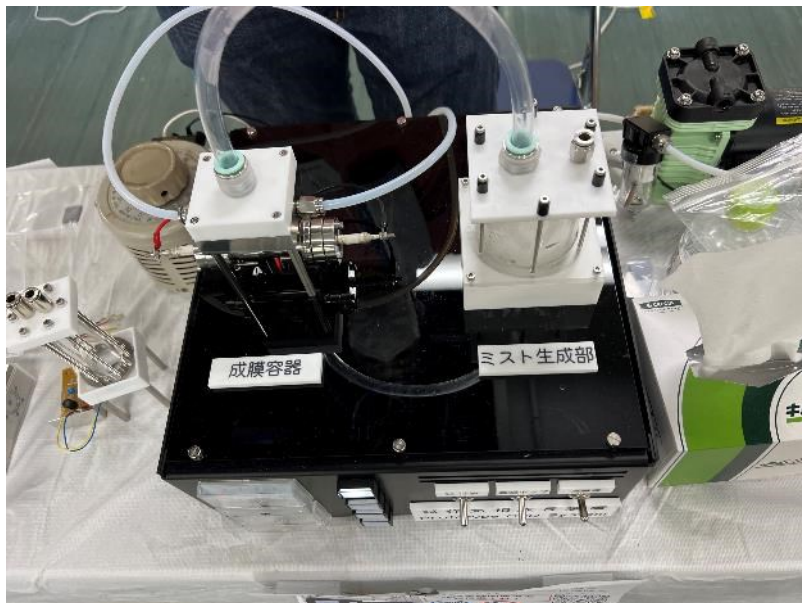
Fujitsu Ltd.



- I'm TOKITA Hiroshi, [@soburi](#) on GitHub
- I'm working as an embedded software programmer at Fujitsu for 20years.
- I'm also a collaborator (co-maintainer) of the Raspberry Pi Pico and GD32 in the Zephyr Project.

- Short introduction about the electronics hobbyist community in Japan
- Supporting Raspberry Pi Pico and Arduino UNO R4 in the Zephyr.
- "ArduinoCore" library for the Zephyr
- Cheap RISC-V board and their insights

Short introduction about the electronics hobbyist community in Japan



Semiconductor film deposition equipment
(Mist CVD)
By 自宅半導体製造プロジェクト
(Homemade Semiconductor Project)

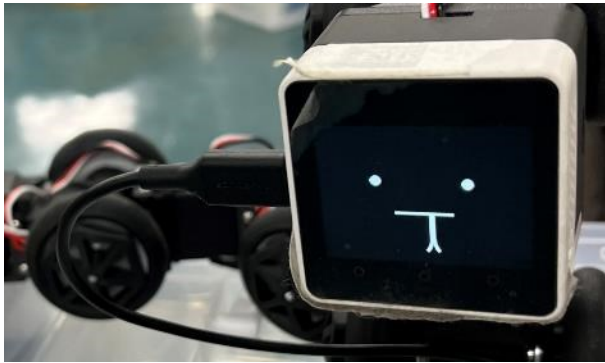
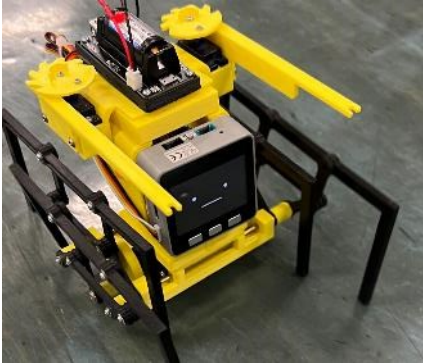


CHIRO (Desktop Robot)
By KOBAYASHI Ryota
<https://twitter.com/CH1H160>

Explanation about page 5

- The electronics hobbyist communities are slightly different from Open Source Software communities, and also they vary by regions.
- NT Tokyo, the event was held near the Imperial Palace, is a grassroots and intimate event.
- We can find marvels in this event, such as personally-made semiconductor manufacturing machines.
- Making humanoid robots, charming robots, and funny gadgets are popular in this community.

Stack-chan: The pop star of the community



Explanation about page 7

- Stack-chan is a pop star in this community. It brought to life with M5Stack and open-source software and published mechanical designs.
- Stack-chan is not commercially available. Every enthusiast constructs and arranges their own unique .
- It is heavily influenced by manga and anime culture.

Previous pages works:

Left Upper/Lower: Stack-chan with eyeglass/hood

By もくはり工房 (Mokuhari-kobo) https://twitter.com/hedgehog_noodl

Middle Upper/Lower Stack Chan built into a quadrupedal robot/LED cat ear with Stack-chan

By Namako <https://twitter.com/KantenNamako>

Upper Right: Sneak-chan (Stack-chan on head of a Serpent form robot)

By B-SKY Lab <https://twitter.com/eternalfriend17>

Lower Right: My own plain version Stack-chan.

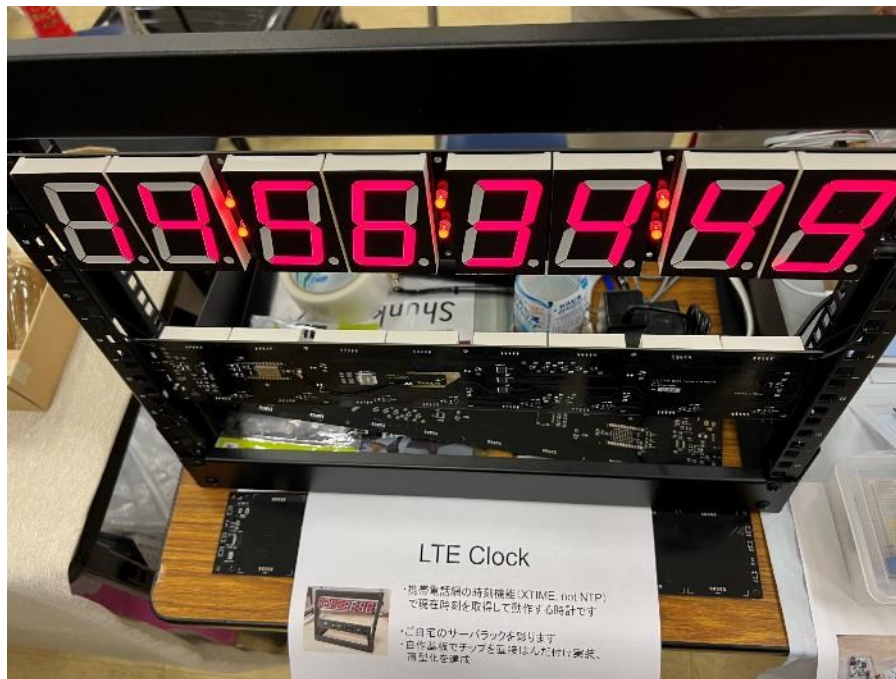
(Designed by Takao <https://twitter.com/mongonta555>)

The “Long Table”



Similar types of this table are used at various exhibition halls in Japan. Electronics hobbyists often participate in anime/manga communities at the same time, resulting in the spread of know-how. In Japan, there seems to be a tendency to decorate this small space as a personal space.

The work that used the Zephyr



LTE Clock (LED clock that can synchronize the time via cellphone network)

By 調布技研 (Chofu tech)

<https://chofu.tech/>

Explanation about page 9-10

- The exhibition table (long table) is commonly used by both the electronics hobbyist community and manga and anime fan communities.
- The table is a common basis for expressing their personal preferences.
- The big LED clock is nRF9160, and the Zephyr uses it to synchronize with the cellphone network.
- These technical demonstrations are also enjoyed as works of artistic expression, as the table provides a common platform for expressing individual preferences.

- Architectural historian MORIKAWA Kaichiro was commissioner of the Japanese pavilion at the 2004 Venice Biennale, taking over from his predecessor ISOZAKI Arata. His exhibition takes up the space at the comic market (the booths separated by long desks featured this presentation) to transform the city by expanding the private space to the public.
<https://venezia-biennale-japan.jpf.go.jp/e/architecture/2004>
- The Japan Media Arts Festival, sponsored by Japan's Agency for Cultural Affairs and held from 1997 to 2022, presented awards and exhibitions in four categories: entertainment (mainly games), manga, anime, and art (technology art). This classification has been criticized as an arbitrary framework based on the convenience of Japan's cultural administration, but by exhibiting technology art in the same frame as categories that have many consumers, such as manga, anime, and games, it has created a positive impact on technology. I think I developed an appreciative attitude.
<https://j-mediaarts.jp/en/>



津山権 Chamaecyparis obtusa 111cm 判型鞍馬石 東京都港区 古野善一

This picture quote from 95th Kokufu (National style) Bonsai exhibition illustrated catalog page 130
第95回 国風盆栽展 (一般社団法人 日本盆栽協会) 図録 130頁より引用

The practice of Bonsai is understood as a means of discovering a wild nature within a pot.

Ko-chu-ten(Universe in a pot)/壺中天



Ko-chu-ten is a concept in Taoism that “The universe is in a pot”.



Bottle circuit By KOSAKA Kimio
https://twitter.com/kimio_kosaka

Explanation about page 13-15

- An electronics exhibition expresses a microcosm within a small space formed by an exhibition table such like as a Bonsai.
- These values are not exclusive to Japan. Their origin is back to Chinese Taoism.
- The concept of "ko-chu-ten" encapsulates the idea of finding a universe within a pot, representing the East Asian inclination to uncover the profound in the small things.
- The works of circuits built within bottles demonstrate this philosophy.
- “Ko-chu-ten” also means utopia, which you can only go to when drinking heavily

- Bonsai is also a cultural tradition that originated in China and underwent transformation and refinement in Japan, evolving into a unique cultural style. The photograph quoted on page 13 features a bonsai crafted from Tsuyama cypress, which was selected for an exhibition. While the typical image of bonsai often conjures elegantly contorted pine trees, this particular type deviates from the norm. When viewed from below, as depicted in the photo on the right, it unfolds a dense, forest-like scene. Tsuyama cypress, with its small leaves, is favored as a bonsai material due to its suitability for achieving the proper scale for miniature representations of large trees.
- Located about an hour north of Tokyo by train, Omiya Bonsai Village has been renowned as a bonsai production hub for over a century. The staff at the bonsai gardener, with its long history in this region, express that, "Generally, Japanese bonsai tends to meticulously recreate nature on a small scale, resembling miniature landscapes. To complement this, there are miniature turtle figurines and buildings, akin to dollhouse elements. Visitors from China often prefer more dynamic and larger bonsai, something that becomes part of the garden."
- The MakerFaire also seems to have interesting trends depending on the country and venue. I think cultural differences between countries also influence electronic work.

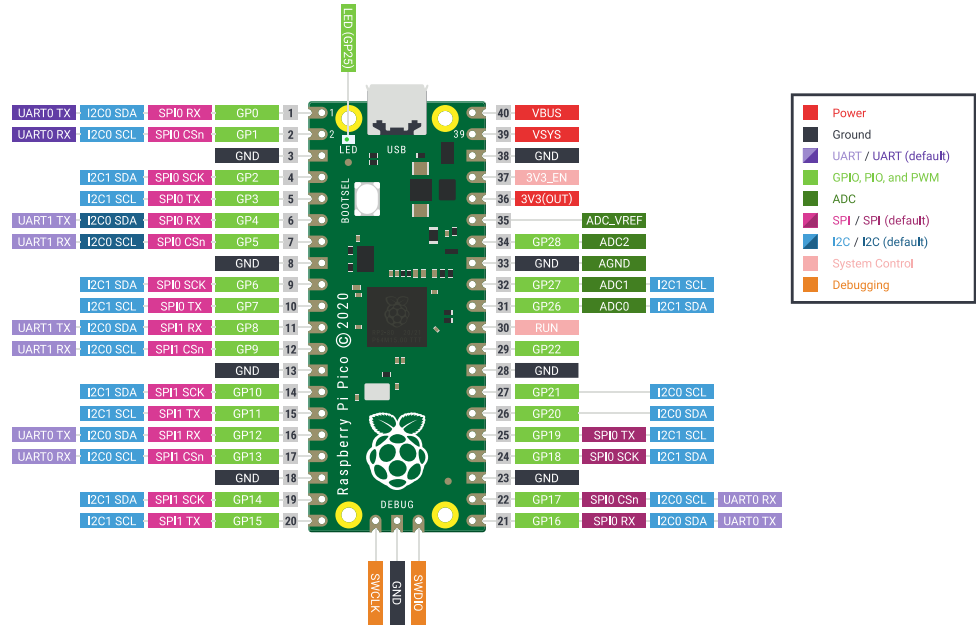


This is NOT a real forest, but a fantastical scene created by bonsai. If you look up at a bonsai like the photo on page 13 from below, you will see something like this.

Supporting Raspberry Pi Pico and Arduino UNO R4

Raspberry Pi Pico

- The board develops and distributes from the Raspberry Pi Foundation.
- \$4. Very cheap and easy to get at parts shop.
- The Pico has basic peripherals GPIO/ADC/PWM/SPI/I2C and **PIO**.

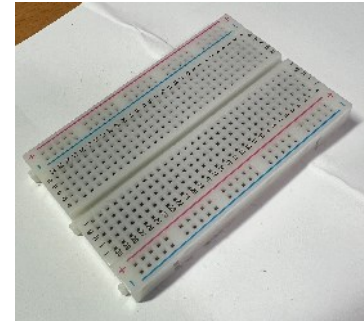
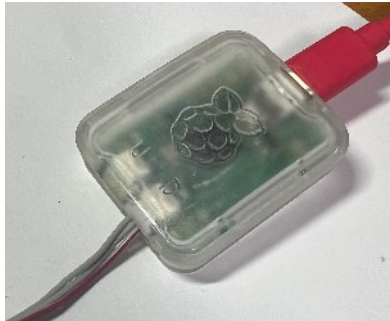
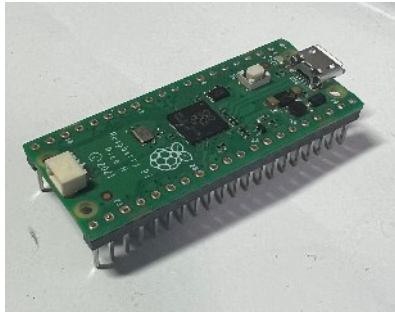


This figure is quoted from Raspberry Pi Pico documentation.

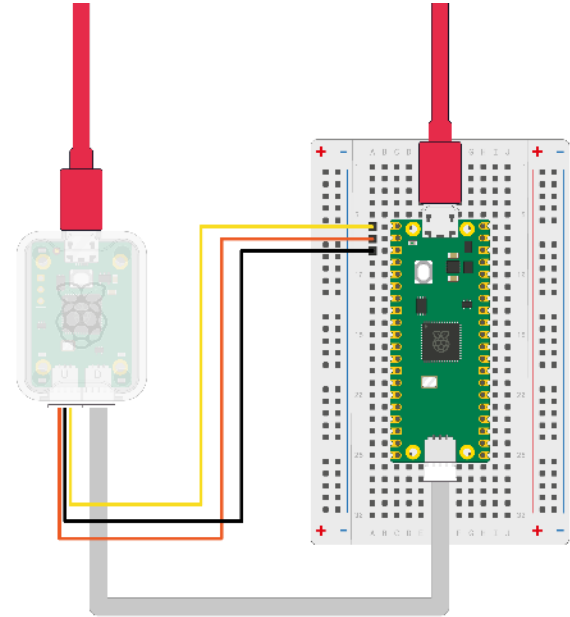
raw.githubusercontent.com/raspberrypi/documentation/develop/documentation/asciidoc/microcontrollers/raspberry-pi-pico/images/Pico-R3-SDK11-Pinout.svg

● Preparation

- [Raspberry Pi Pico](#) x 1 (We recommend the **H** version.)
- [Raspberry Pi Debug Probe](#) x 1
- Breadboard and Jump wires
- ...And your Laptop PC



- Connect the Debug Probe to the connector at the edge.
- Colored wires are optional. It makes it possible to communicate with UART.



This figure is quoted from the Raspberry Pi Debug Probe product explanation.
<https://www.raspberrypi.com/documentation/microcontrollers/images/wiring.png>

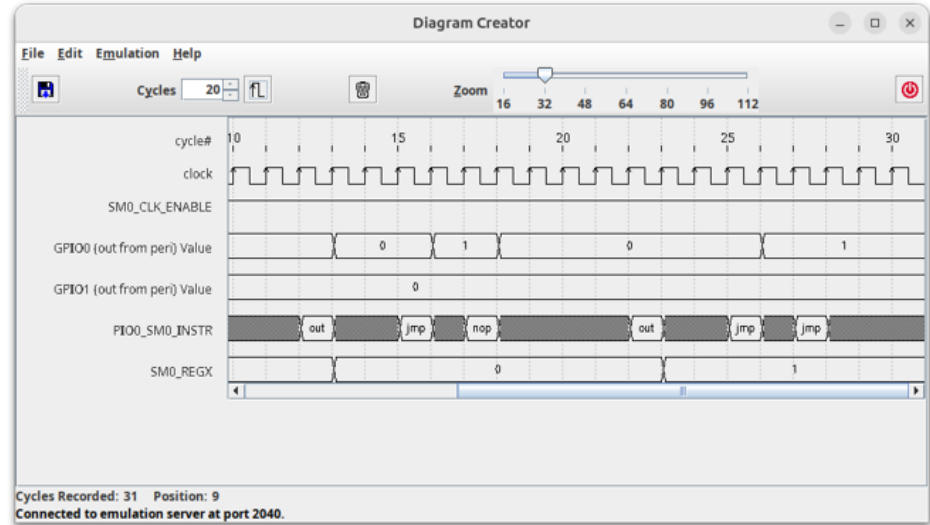
- Type the following commands to run blinky example

```
$ west build -b rpi_pico samples/basic/blinky  
$ west flash --runner pyocd
```

- We can also use samples in the Zephyr repository
 - hello_world, blinky, button, etc...

- The Pico is the **easiest** choice to start Zephyr.
- You need three things to begin
 - Raspberry Pi Pico (I recommend an H-type that has a debug connector)
 - Raspberry Pi Debug Probe.
 - Breadboard
- And your laptop PC
- Wire the Pico and Debug Probe with the cable.
- Run the “west build” and “west flash” commands to make blinking.

- The PIO is a unique feature of the RP2040
- It is a configurable device for implementing various peripherals (even UART, SPI, VGA, etc...) with a tiny assembler language



RP2040 PIO Simulator <https://github.com/soundpaint/rp2040pio> is a helpful tool for verifying how behave the PIO. It shows GPIO's state changes synchronizdl with PIO instruction and clocks.

WS2812 LED

- WS2812 protocol is real-timely controlled high and off to send the bit.
- I'm fixing issues that were found in review by [Simon Guinot @simonguinot](#). (Thanks!)
- It will need a little more time to merge.

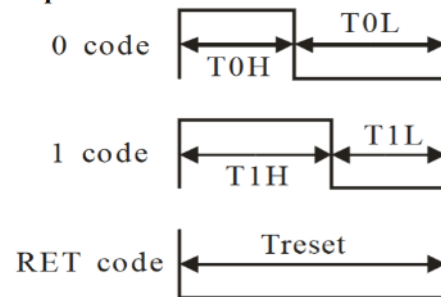


WS2812 is often used as a strip that connects multiple LEDs.

Data transfer time($T_H+T_L=1.25\mu s\pm 600ns$)

T0H	0 code ,high voltage time	0.35us	$\pm 150ns$
T1H	1 code ,high voltage time	0.7us	$\pm 150ns$
T0L	0 code , low voltage time	0.8us	$\pm 150ns$
T1L	1 code ,low voltage time	0.6us	$\pm 150ns$
RES	low voltage time	Above 50us	

Sequence chart:



The timing definition of the WS2812 protocol from the WS2812 datasheet.

<https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>

Explanation of page 24-25

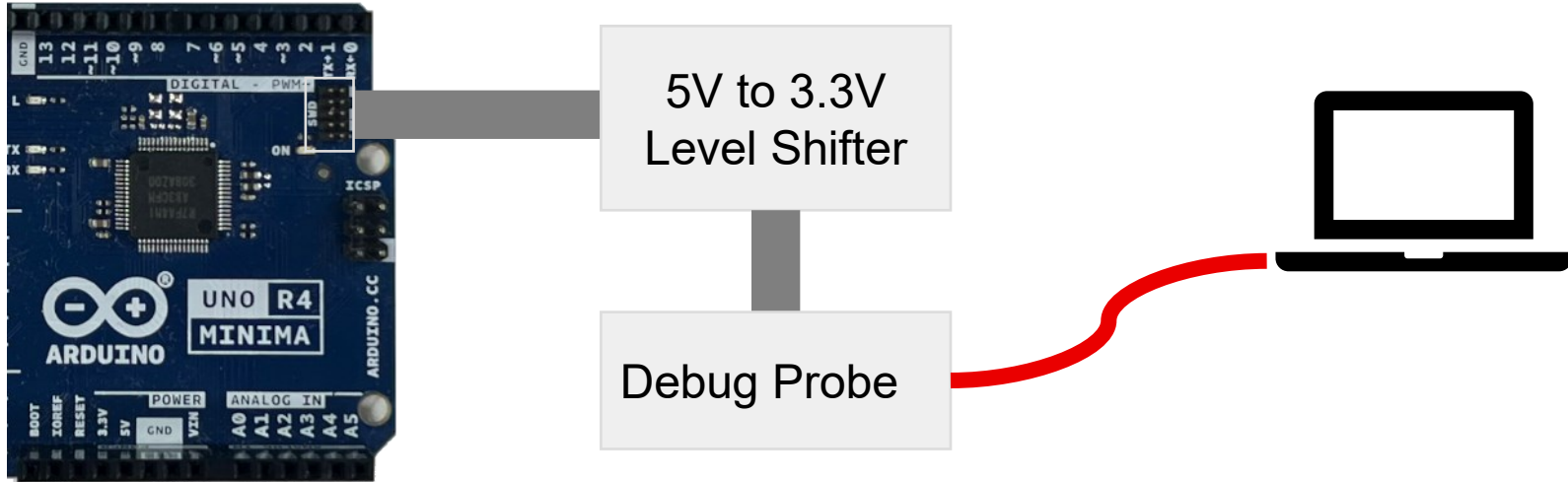
- The PIO can control waveforms synchronized with the system clock in real time.
- It means the PIO can implement several types of peripherals.
- WS2812 can be controlled with the real-time controlled pulse.
- It requires nervous control if implemented with GPIO.
- Using PIO can easily implement the controlling of the WS2812.

- Arduino UNO R4 released Jul 23!
- Supporting UNO R4 has been merged into the mainline on Nov 1. ([#60760](#))
- Only GPIO/UART/Interrupts are supported now.

The screenshot shows a GitHub pull request titled "Add support for Arduino UNO R4 #60760". The pull request is marked as "Merged" and was created by "fabiobaltieri" who merged 8 commits into the "zephyrproject-rtos:main" branch from the "soburi:arduino_uno_r4" branch, completed "last month". Below the title, there are statistics for the pull request: 79 Conversations, 8 Commits, 21 Checks, and 59 Files changed. A comment from the user "soburi" is visible, dated "Jul 25" and marked as "edited". The comment text reads: "Adds support for Arduino UNO R4. I have verified that hello_world, blinky, and button sample work with this PR." The user's profile picture and a "Member" badge are also visible next to the comment.

Debugging the UNO R4

- Arduino UNO R4 can develop with a CMSIS-DAP adapter with the PyOCD.
- The UNO R4 run on 5V. Usually we need a level converter to connect the Debug Probe.



Explanation of pages 26-27

- Zephyr now supports the Arduino UNO R4.
- It supports basic functionality only at this time.
- The Arduino UNO R4 can debug with the CMSIS-DAP adapter.
- But UNO R4 works in 5V. We usually need the 3.3V-5V level shifter for debugging.

- Basic the Pico usage
 - To run the west command, you need to set up Zephyr. See the [Getting Started Guide](#).
- Implementation status of PIO and other Pico functions
 - In fact, PIO's Zephyr implementation is essentially just the device-tree definition, and currently most of the actual processing uses the Pico-SDK implementation as is. Pico-W's Wi-Fi also uses a 3-wire SPI implemented in PIO, which is also a big issue.
 - There are also proposals for PIO API, but discussions are still ongoing.
 - Also, the other big challenge remains SMP support.
- Debugging Arduino UNO R4
 - Unfortunately, we cannot use most of inexpensive debuggers with UNO R4 because they do not support 5V target.
 - It's not that difficult, but it's a high hurdle for people who aren't good at building hardware. I want to think of a better way.

● J-Link

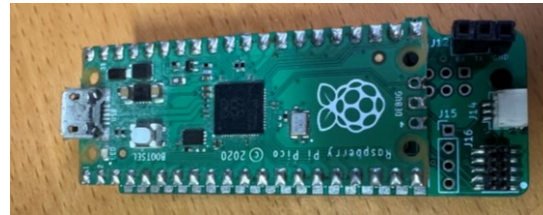
- Proprietary
- Most used in the Zephyr support boards (supporting 253 boards in 439 of all)
- Usually, this is used with a J-Link debug adapter.
- Some board has on-board version adapters. (e.g. Nordic DK)
- It's a bit expensive for hobbyists.

● OpenOCD

- Open source
- Longley is used in open source community.
- Various adapters and targets supporting.

● PyOCD

- Open Source
- ARM-specific OCD program written in Python.
- Use with a CMSIS-DAP compatible adapter(such as Pico Debug Probe)
- Mostly ARM processor supported with Open-CMSIS-pack extensions



The photo is a Raspberry Pi Debug Probe compatible debugger with a 5V-3.3V level shifter made by the author. Raspberry Pi Debug Probe is open-sourced. We can easily create compatible devices with the Pico. The PyOCD can also use <https://github.com/ARMmbed/DAPLink> based adapter. (LPC11U35 and etc.)

- Add following lines to board.cmake to support PyOCD.

```
board_runner_args(pyocd "--target=r7fa4m1ab")  
include(${ZEPHYR_BASE}/boards/common/pyocd.board.cmake)
```

- You can find the PyOCD supporting SoC with these commands.

```
pyocd pack update  
pyocd pack find '*'
```


Explanation of page 30-32

- Zephyr supports several debug programs.
- PyOCD is a specified debug program that can be configured easily.
- We can add support for the PyOCD by adding two lines to the board.cmake.

- The Raspberry Pi Pico is one excellent/cheap choice for starting development with the Zephyr.
- Arduino UNO R4 is also starting the effort to support the Zephyr. We're welcoming a patch for enhance support for the Arduino UNO R4.
- Ease of debugging is a strong point of the Zephyr.
- Supporting the PyOCD is easy.
- If the Zephyr supports your board and does not support the PyOCD, This is a chance to make your first contribution to the Zephyr!

ArduinoCore for Zephyr

- This project ports ArduinoCore APIs for the Zephyr.
- [Dhruva Gole](#) began the project as the GSoC2022 Project
- The development is still going, not yet merged.



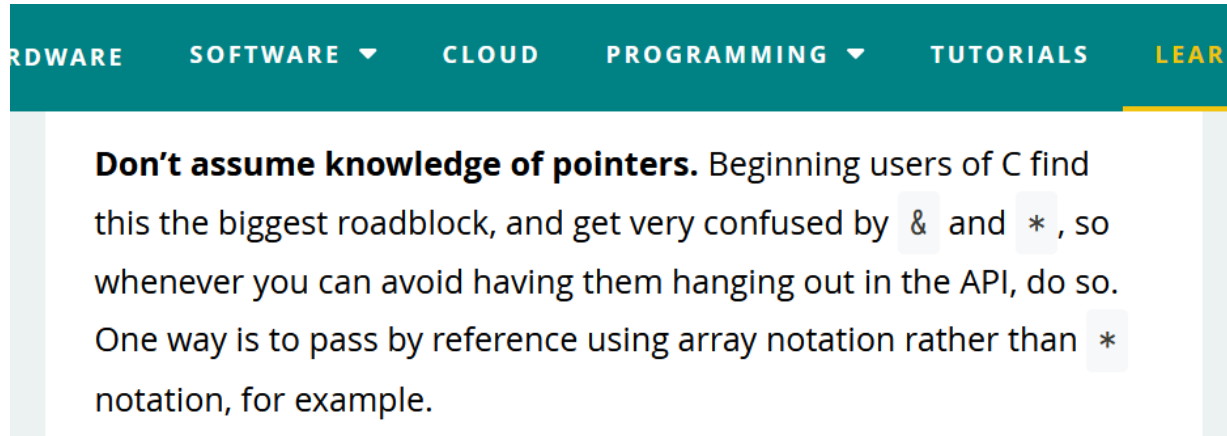
<https://github.com/zephyrproject-rtos/gsoc-2022-arduino-core>

```
#include <Arduino.h>

/* 1000 msec = 1 sec */
#define SLEEP_TIME_MS 1000
void setup() { pinMode(LED_BUILTIN, OUTPUT); }
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(SLEEP_TIME_MS);
  digitalWrite(LED_BUILTIN, LOW);
  delay(SLEEP_TIME_MS);
}
```

- We can use Arduino-API as Zephyr's API wrapper.

- **Don't assume knowledge of pointers**

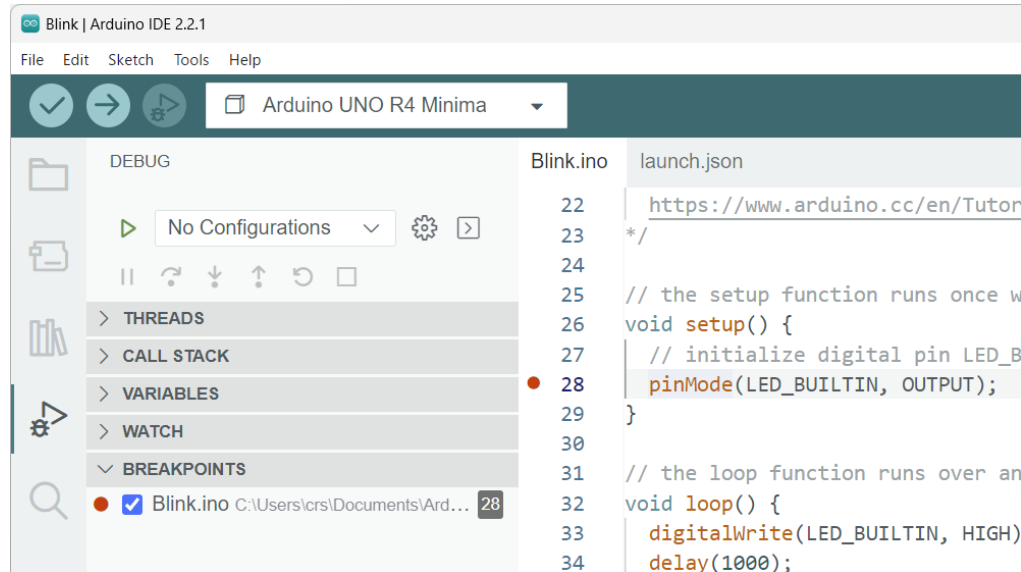


The screenshot shows a navigation bar with the following items: **RDWARE**, **SOFTWARE** (with a dropdown arrow), **CLOUD**, **PROGRAMMING** (with a dropdown arrow), **TUTORIALS**, and **LEARN** (with a dropdown arrow). Below the navigation bar is a text block with the following content:

Don't assume knowledge of pointers. Beginning users of C find this the biggest roadblock, and get very confused by `&` and `*`, so whenever you can avoid having them hanging out in the API, do so. One way is to pass by reference using array notation rather than `*` notation, for example.

Behind ease of use

- Arduino IDE 2 has supported debugging feature!
- Arduino IDE hides some of complex, but we are notice it besides with we used to Arduino.



```
#if defined(__AVR__)
static volatile uint8_t *ss_pin_reg;
static uint8_t ss_pin_mask;
inline static void initSS() {
    ss_pin_reg = portOutputRegister(digitalPinToPort(ss_pin));
    ss_pin_mask = digitalPinToBitMask(ss_pin);
    pinMode(ss_pin, OUTPUT);
}
----- SNIP -----
#elif defined(__MK20DX128__) || defined(__MK20DX256__) || defined(__MK66FX1M0__) ||
defined(__MK64FX512__)
static volatile uint8_t *ss_pin_reg;
inline static void initSS() {
    ss_pin_reg = portOutputRegister(ss_pin);
    pinMode(ss_pin, OUTPUT);
}
```


Explanation of page 35-40

- The ArduinoCore for Zephyr is a project for porting Arduino APIs for the Zephyr.
- Dhruva Gole started the project as the GSoC2022 project.
- The Arduino API is designed for beginner programmers, so the convention is unique.
- The “ease of use” hides complexity well. It is slightly different from Zephyr’s principles.

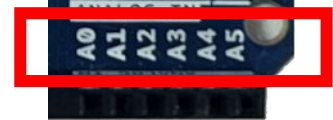
Connector; The Zephyr's solution

- The connector definition binds the Arduino Pin Name and GPIO number.

```
arduino_header: connector {
    compatible = "arduino-header-r3";
    #gpio-cells = <2>;
    gpio-map-mask = <0xffffffff 0xffffffffc0>;
    gpio-map-pass-thru = <0 0x3f>;
    gpio-map = <0 0 &ioport0 14 0>, /* A0 */
               <1 0 &ioport0 0 0>, /* A1 */
               <2 0 &ioport0 1 0>, /* A2 */
               <3 0 &ioport0 2 0>, /* A3 */
               <4 0 &ioport1 1 0>, /* A4 */
               <5 0 &ioport1 0 0>, /* A5 */
               <6 0 &ioport3 1 0>, /* D0 */
    ...
}
```



Arduino
"Pin Name" is
displayed on board.



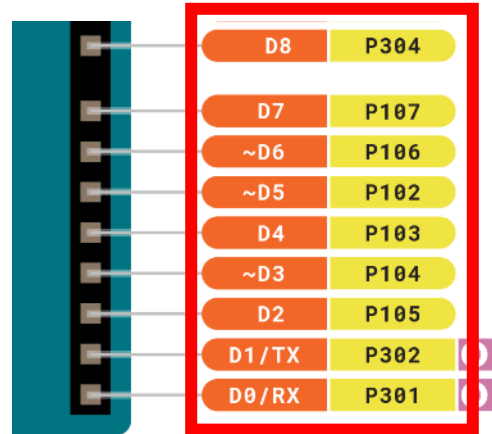
“Pin Name” as an abstraction mechanism

- Arduino “Pin Name” can be commonly used on different boards. It means some “hardware abstraction” is here.

nRF52840-DK



Arduino UNO R4 minima



The actual pin is specified by GPIO number.

The figure of Arduino UNO R4 pinout quoted from <https://docs.arduino.cc/static/7501e2ad784ed34f331afae4947e06fd/ABX00080-pinout.png>

- License incompatibilities exists.

Lesser GNU Public License

Apache2 License

ArduinoCore Library

Zephyr

Need to redistribute the code that linked LGPL2 library.

You can select to not redistribute the code you wrote even if link to the library.

Explanation of page 41-45

- Arduino basically has no hardware abstraction layer in the API, making it challenging to support much hardware.
- In the Zephyr case, we are using device-tree for hardware abstraction.
- The "connector" is definition in device-tree for supporting the Arduino connector. It is useful for the ArduinoCore for Zephyr.
- The ArduinoCore for Zephyr has an LGPL and Apache2 incompatibility problem.
- I'm sure there will be a positive outcome from the Zephyr and Arduino project collaboration.

- Zephyr usually generates a C language macro from the device-tree definition and can reference it from a C language program.
- However, C language macros are not generated for connectors.
This is because the connector node is treated as a conversion table when interpreting the device-tree, and the conversion has already been completed at the time of macro generation.
- As mentioned in the main text, connectors can also be used as a hardware abstraction mechanism, so I think it would be better to be able to reference them from the C language in some way.
We are currently considering whether and how to respond.

Conclusion of the second topic

- The ArduinoCore for the Zephyr project still needs to be mature enough.
- The ArduinoCore for Zephyr can provide comprehensive support for many hardware leveraging the device-tree hardware abstraction.
- This is not a replacement for Arduino.
- It may be helpful for migration paths for Arduino projects, which have grown and become complex.
- Still, there are license problems. However, the collaboration between the Zephyr and the Arduino projects will bring good results.
- I expect this project to encourage more use of Zephyr.

Emerging RISC-V processors

More than 4,000 RISC-V Members across 70 Countries



114 Chip

SoC, IP, FPGA

3 I/O

Memory, network, storage

21 Services

Fab, design services

58 Software

Dev tools, firmware, OS

3 Systems

ODM, OEM

13 Industry

Cloud, mobile, HPC, ML, automotive

159 Research

Universities, Labs, other alliances

3k+ Individuals

RISC-V engineers and advocates

RISC-V membership up 24% in 2023



#RISCVSummit #RISCVEverywhere

From Keynote session by Calista Redmond at RISC-V Summit 2023

https://static.sched.com/hosted_files/riscvsummit2023/33/1.%20Calista%20Redmond.pptx.pdf

- Sipeed Longan Nano is cheap development board with 1st generation of commercial RISC-V chip GigaDevice GD32V.
- I added support for this chip/board in December 2021. This is my first contribution for the Zephyr. ([#34970](#))
- SoC support is a huge patch, and if a first-time contributor were to issue such a patch, it would be understandable the team to be wary. The merge was successful mainly due to patient reviews by [Gerson Fernando Budke\(@nandojve\)](#) and [Gerald Marull-Paretas \(@gmarull\)](#). Thanks.
- The Zephyr community is very welcoming to new contributors. Please come and participate in the development.



MCAUSE register

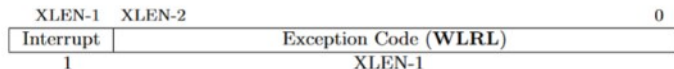


Figure 3.25: Machine Cause register mcause.

The definition of mcause register in RISC-V ISA spec Volume 2, Privileged Specification version 20211203. https://drive.google.com/file/d/1EMip5dZInypTk7pt4WWUKmtjUKTOkBgq/view?usp=drive_link

- GD32V's mcause register has its specific extensions.
- It requires exceptional support for kernel.

Table 16.3 mcause register

Field	Bit	Description	Note
INTERRUPT	31	Current trap type: <ul style="list-style-type: none">• 0: Exception or NMI• 1: Interrupt	
MINHV	30	Indicate processor is reading interrupt vector table	Note: These fields are only effect in CLIC mode. When in CLINT mode, these field is masked read as zero, write ignored.
MPP	29:28	privilege mode before interrupt	
MPIE	27	Interrupt enable before interrupt	
Reserved	26:24	Reserved 0	
MPIL	23:16	Previous interrupt level	
Reserved	15:12	Reserved 0	
EXCCODE	11:0	Exception/Interrupt Encoding	

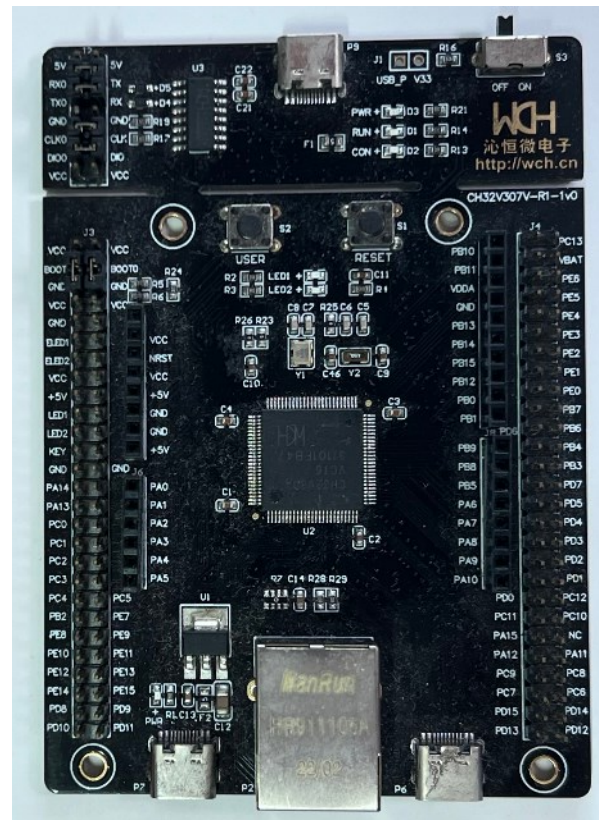
The table quoted from Nuclei ISA Spec https://doc.nucleisys.com/nuclei_spec/isa/core_csr.html?highlight=mcause#mcause

Explanation of page 48-50

- The RISC-V processor shipments are rapidly emerging.
- The GigaDevice GD32V is the first generation of commercial RISC-V products. (I added support for Zephyr.)
- The chip has some non-standard extended specs from RISC-V ISA specs.
- It requires exceptional support for the kernel.

In a case of a new generation of RISC-V chip

- WCH CH32V307 is a newer generation than GD32V.
- This chip has no unique extension for standard registers.
- I think it is the maturing of design.



- WCH CH32V003 is a highly cheap chip.
- I think this is a typical example of the low-cost RISC-V chips that will increase in the future.



Picture From <https://akizukidenshi.com/catalog/g/gI-18062/>

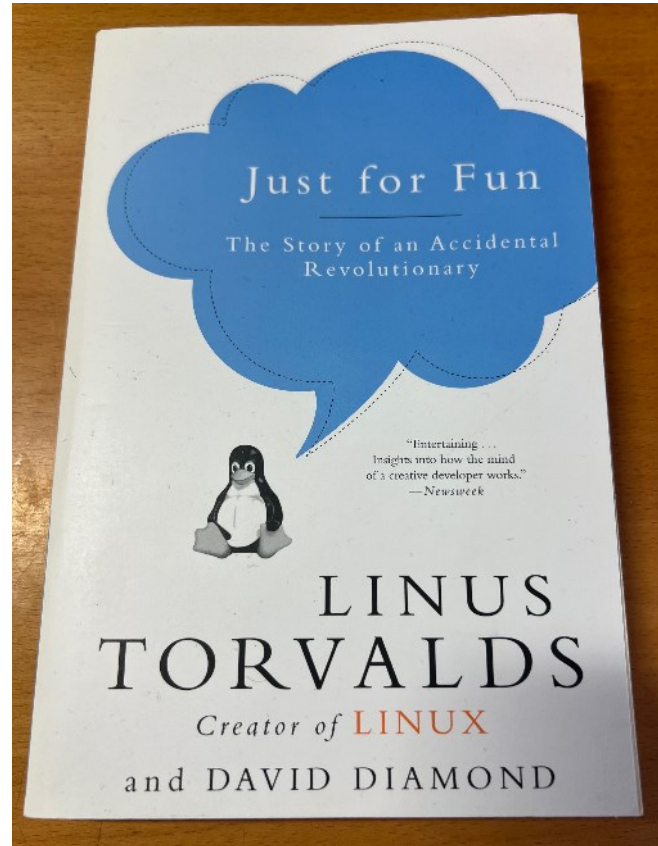
秋月電子通商(Akizuki-Denshi Electric Commerce) is one of most loved parts shop by Japanese electronics hobbyist communities. The shop import this. Quick-witted hobbyists are starting to play with this chip right away.

Explanation of page 52-53

- WCH CH32V is a newer generation RISC-V chip than the GigaDevice GD32V.
- The non-standard registers shown earlier are gone.
- With the new generation, standardization and sophistication have progressed.
- CH32V003 is a highly cheap RISC-V chip.
- It has only 2kb of memories, but the Zephyr says “at least 2kb!”.
- It may be funny if it we can be used with the Zephyr.
- We hope that hobby applications will expand with cheap and interesting chips.

“Just for Fun”

- This word embodies the spirit of community.



- Linus, the creator of Linux, expressed the spirit of community with the title of their book, **"Just for fun"**.
- Zephyr can become something fun like that, just like Linux.
- I would be very happy if many people could join Zephyr, Linux, and Open-Source communities.

Thank you

