RISC-V WGの活動とWGが進める RISC-V研究・開発の取り組み

2022年11月16日

JASA 技術本部 ハードウェア委員会 RISC-V WG
勝見 哲也※

※（株）Communication Technologies Inc. 代表取締役社長/
JASA 技術本部 副本部長

一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association

Contents

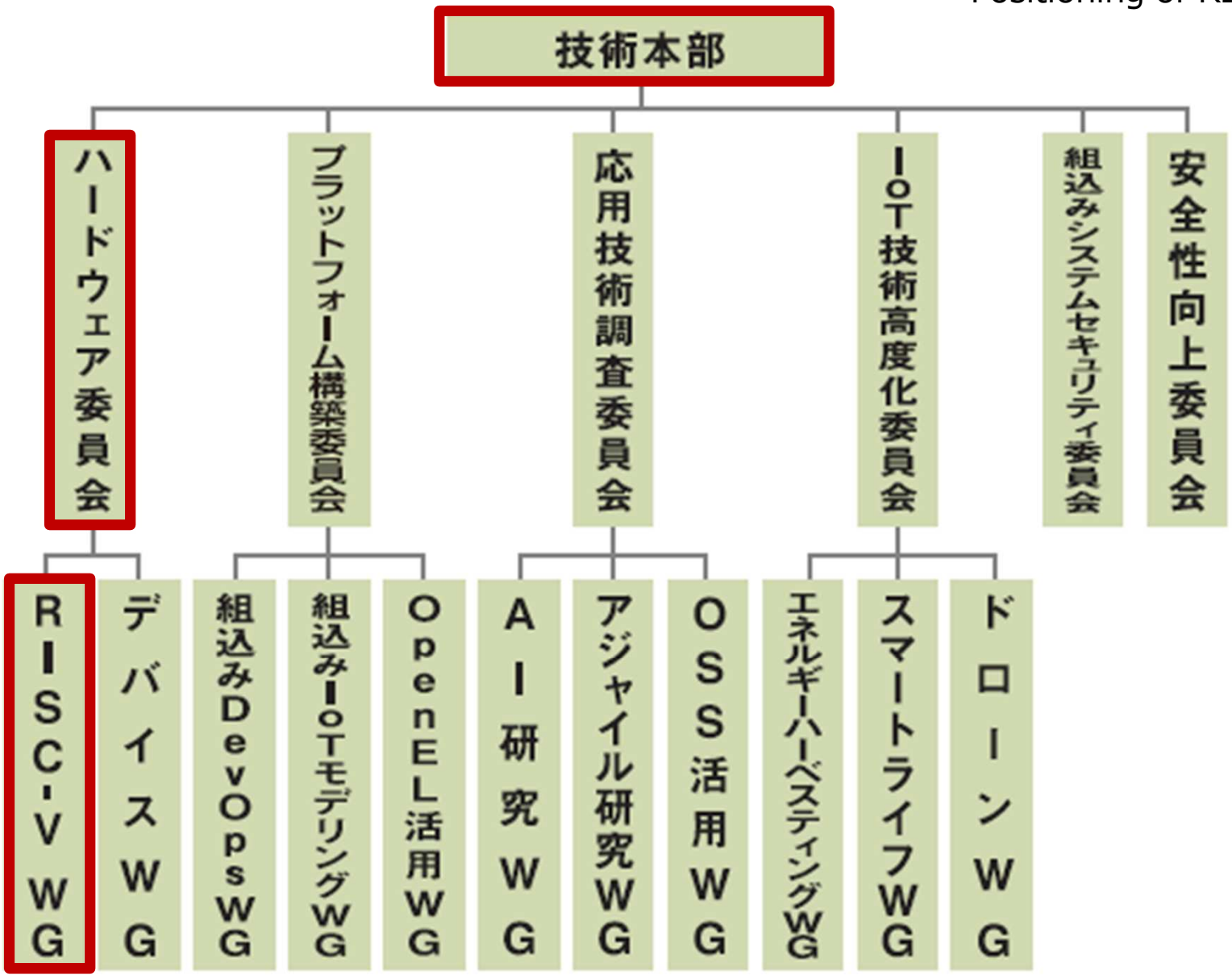
1. RISC-Vについて

About RISC-V

・オープンソースかつロイヤリティフリーのRISC(Reduced Instruction Set Computer)に基づくISA(命令セットアーキテクチャ)です。
・ RISC-V Internationalは、スイスを拠点とするグローバルな非営利団体です。 2015年に29人のメンバーで構成されるRISC-V Foundationとして設立されたRISC-Vは 現在70か国以上に2,000人以上のメンバーを擁する真のグローバル組織です。
・日本ではRISC-V協会が窓口をしています。

2-1. JASA RISC-V WGの位置づけ

Positioning of RISC-V WG



一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association

2-2. RISC-V WGの活動について

About the activities of JASA RISC-V WG

《WGの活動方針》

・オープンな仕様で会員が自由に活用できるRISC-Vプラットフォームを
　会員の協力で開発整備し、組込み分野でのRISC-V普及に努める
・関連団体とのコラボによりプラットフォームの応用範囲を広げる

《活動内容の項目》

- 月例WGの開催
- RISC-V著名人を講師にお迎えし、隔月でWebinarを開催
- 組込みに使えるRISC-Vプラットフォームの開発
- RISC-V関連団体との協創

3. 開発ロードマップ

Development schedule for this year, etc.

《開発ロードマップ》

2020年度	2021年度	2022年度
・Rocket Chipの FPGAへの実装 ・ブートローダ開発 ・Arduino環境移植	・VSCデバッグ環境構築	・64ビット版RISC-V コアのFPGA実装 ・OS移植 ・ブート環境

・市販FPGAボード上でRISC-Vを開発できるプラットフォームを開発しました
・**全体を日本語でドキュメント化**したので、初心者でも手軽に扱えます
・開発用、教育用プラットフォームとしてご活用いただくことを期待しています
・今年度は**産学連携**で同様なコンセプトのOSが走るRISC-Vプラットフォームを開発・整備しています

・現在、JASA会員は自由に使えますのでご活用ください

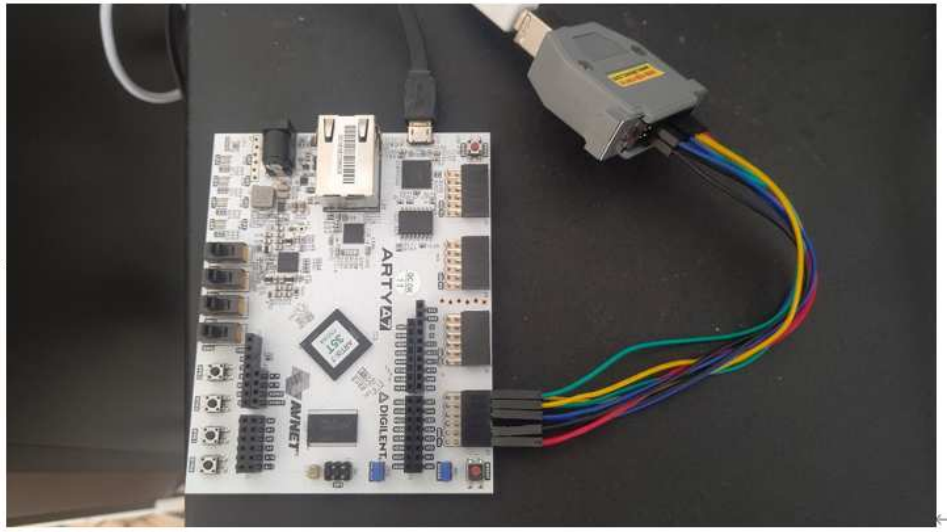
一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

4-1. 開発内容その１

Development of usable platform

《開発ロードマップ》

2020年度	2021年度	2022年度
・Rocket Chipの FPGAへの実装 ・ブートローダ開発 ・Arduino環境移植	・VSCデバッグ環境構築	・64ビット版RISC-V コアFPGA実装 ・OS移植 ・ブート環境開発



・市販FPGAボードにRISC-Vコア実装
・ブートローダを開発
・Arduino IDE環境を移植
・VSCデバッグ環境をセットアップ

上記を手順通りやれば初心者でも実現できるように手順のドキュメントを作成

4-2. 昨年度までの開発成果

《方針と目的》

FPGAやRISC-Vに触れるのが初めての人でも手順通り進めれば
一気通貫でプラットフォームを作れるようにすること
⇒ 日本語ドキュメントを作成

《成果物》

① FPGA開発環境のインストールから市販FPGAボード上への
Rocket Chipの実装
② PCのCOM(UART)経由で開発したコードをダウンロードできる
ようにするためのブートローダ開発
③ Arduino IDEのセットアップとブートローダの実装
④ Visual Studio Codeで作成したプログラムをデバックできるよう、
環境の構築とデバッグ方法のドキュメント作成

※現在、JASA会員は成果物のプラットフォームを自由に使えます

4-3. プラットフォームの概要①

Implementation on FPGA

《①FPGAへの実装》

市販のFPGAボード(Arty A7-35T)にRISC-Vコアの実装

(1) FPGAへの実装を行う仮想環境構築手順
(2) FPGA書き込みツールのインストール手順
(3) FPGAへ書き込むデータの作成手順
(4) FPGAへの書き込み・実行手順
(5) VerilogによるRocket Chipへの接続手順

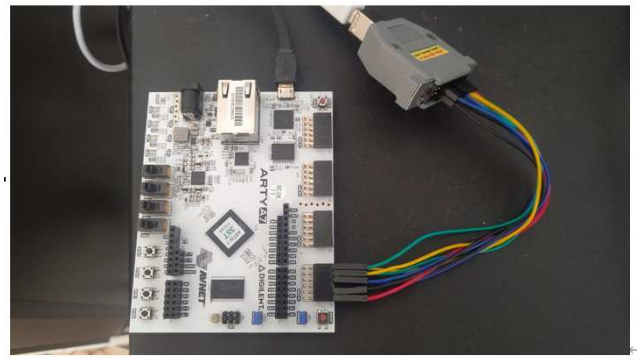
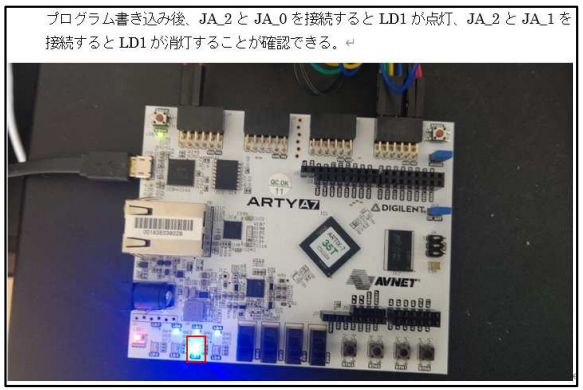


図 4-18 ARM-USB-TINY-H Debugger 接続



一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

4-3-1. FPGAへの実装

1. FPGAへの実装を行う仮想環境構築手順

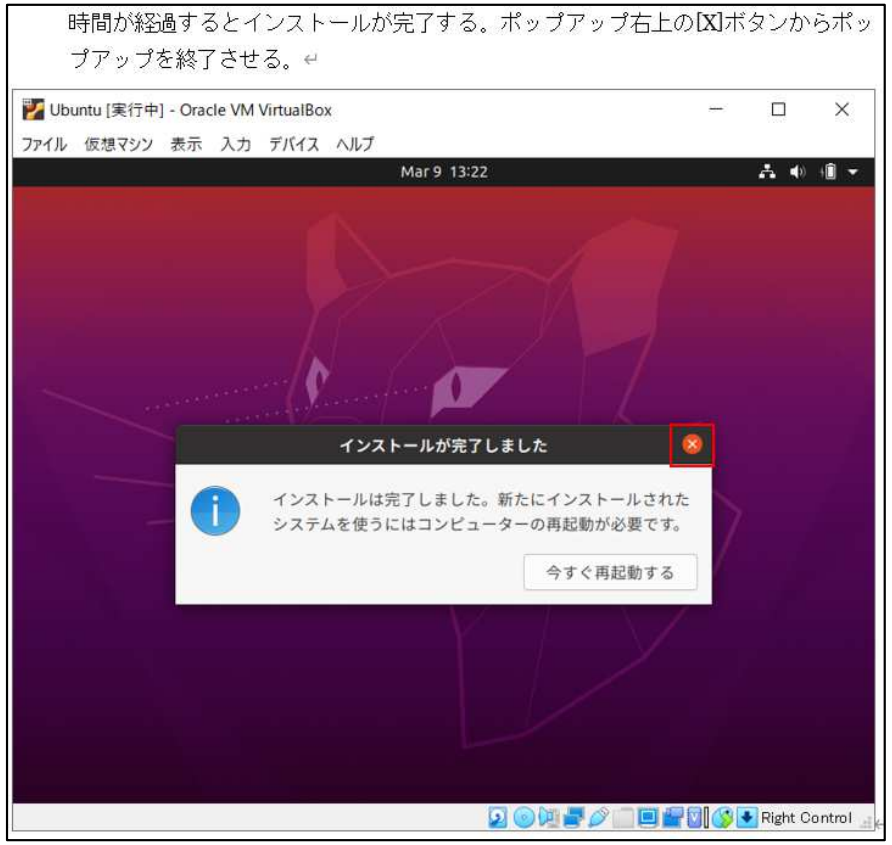
仮想化ソフトのダウンロードからOSのインストールまでの手順を説明

仮想化ソフトを使用した経験が無くても環境を構築可能

2. VirtualBox のインストール

ダウンロードしたファイルを実行する。実行すると以下のウィンドウが出現するので、Next ボタンを押下する。

時間が経過するとインストールが完了する。ポップアップ右上の[X]ボタンからポップアップを終了させる。



4-3-2. FPGAへの実装

2. FPGA書き込みツールのインストール手順

FPGA書き込みツールのダウンロードからインストールまでの手順を説明

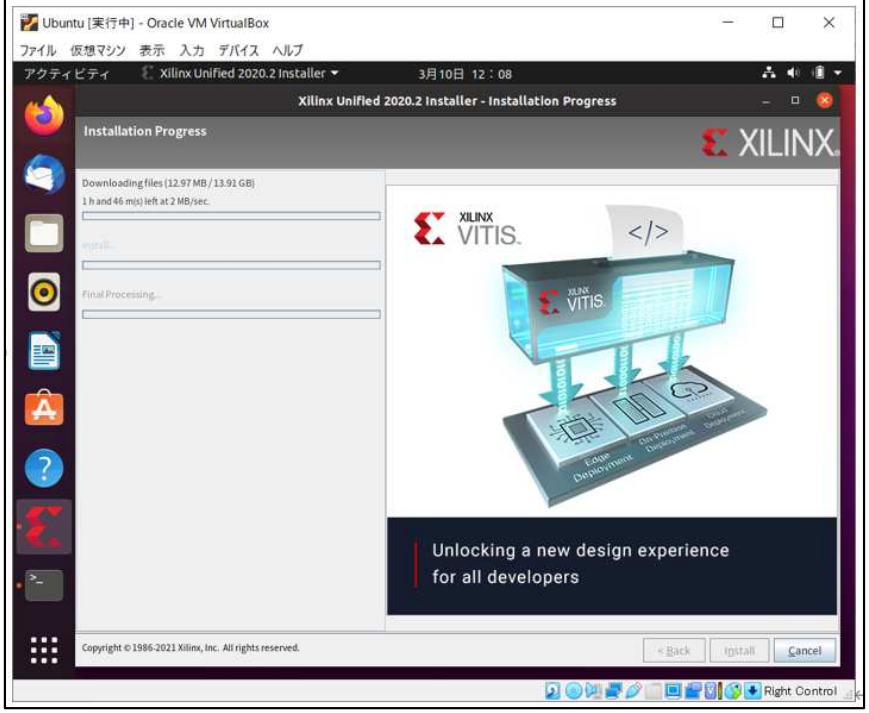
FPGA書き込みツールを使用した経験が無くても実行可能

3. Vivado インストーラーのダウンロード

Vivado のインストーラーをダウンロードする。そのために、予め Xilinx アカウントでサインインする。下記の URL にアクセスし、右上のアイコンから[ログイン/登録]を選択してクリックする。

「https://japan.xilinx.com」

ダウンロードとインストールが開始されるので、終了するまで待つ。インストール中に Xilinx Information Center のウィンドウが表示されるが、無視して構わない。インストールが完了すると"Install completed successfully."と表示されるので OK ボタンを押下する。インターネット通信速度によって変化するが、インストール完了まで2時間近く要するので注意すること。



一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

4-3-3. FPGAへの実装

3. FPGAへ書き込むデータの作成手順

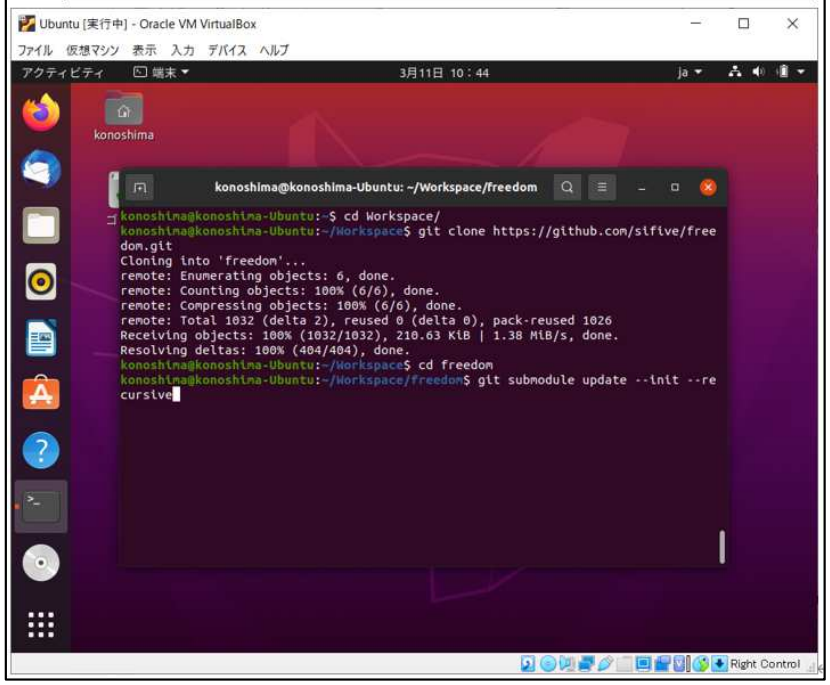
既存のプラットフォームを利用したデータの作成手順を説明

サンプルプログラムのビルド手順についても説明

7. SiFive Freedom E310 ダウンロード

SiFive Freedom E310 をダウンロードするため、以下のコマンドを実行する。本書では Workspace ディレクトリにダウンロードする。

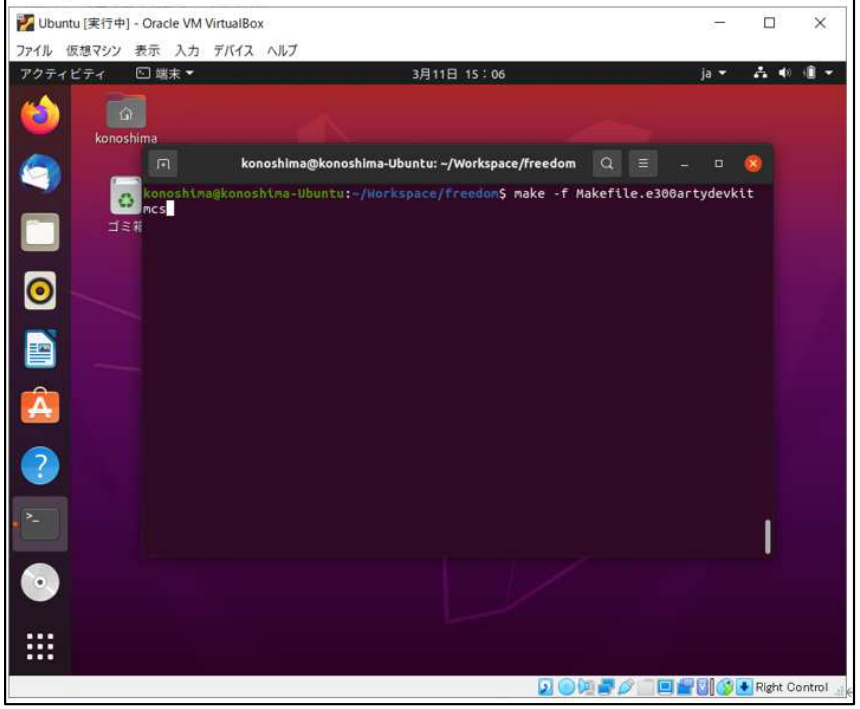
```
git clone https://github.com/sifive/freedom.git
cd freedom
git submodule update --init --recursive
```



10. mcs ファイル生成

FPGA 評価ボードの書き込みデータである mcs ファイルを生成するため、以下のコマンドを実行する。本書では生成したファイルは Workspace/freedom/builds/e300artydevkit/obj ディレクトリに格納される。

```
make -f Makefile.e300artydevkit mcs
```



4-3-4. FPGAへの実装

4. FPGAへの書き込み・実行手順

評価ボードへの接続からFPGAへ書き込んで実行するまでの手順を説明
サンプルプログラムの動作についても紹介しており、
サンプルプログラムが正常に動作しているかを確認可能

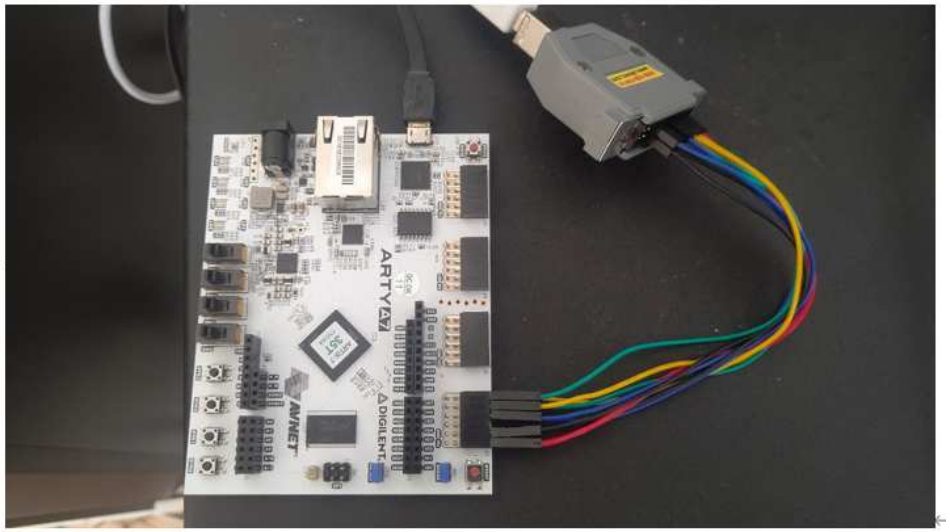
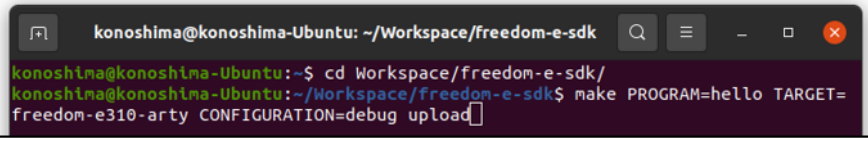


図 4-18 ARM-USB-TINY-H Debugger 接続

8. サンプルプログラム書き込み

サンプルプログラムを FPGA に書き込むため、PC と Arty A7 ボード、ARM-USB-TINY-H を接続した状態で以下のコマンドをサンプルプログラムのディレクトリで実行する。本書では Workspace/freedom-e-sdk ディレクトリからサンプルプログラムの hello を書き込む。書き込みに失敗する場合は PC への USB 接続を一旦外して、再接続して行うことで改善できる可能性がある。

```
cd Workspace/freedom-e-sdk/
make PROGRAM=hello TARGET=freedom-e310-arty
CONFIGURATION=debug upload
```



4-3-5. FPGAへの実装

5. VerilogによるRocket Chipへの接続手順

接続するデバイスやRocket Chipへの接続手順を説明
Arty A7-35TのLEDとジャンパピンの接続設定を行って
LEDを点灯・消灯させる例についても紹介

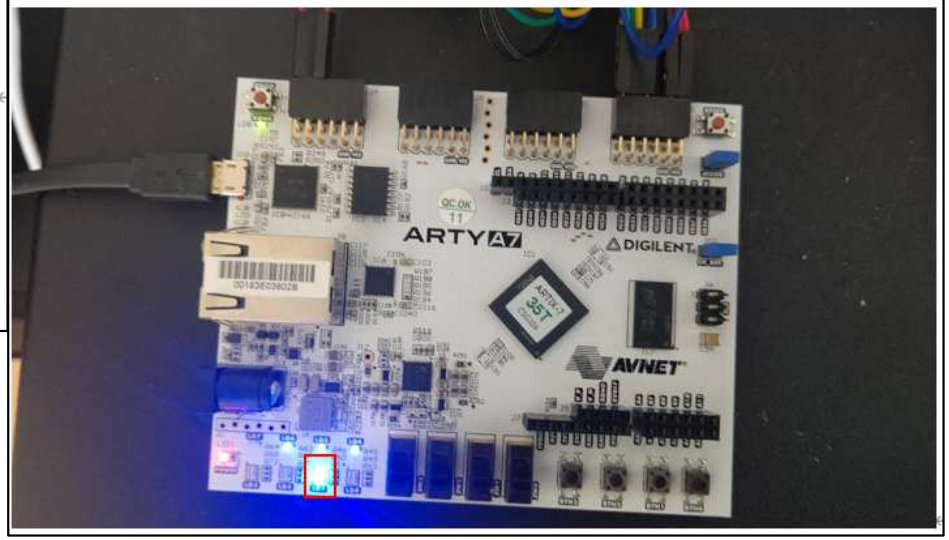
2. デバイス設定

接続するデバイスに関する設定について、下記の URL に掲載されている。
「https://chipyard.readthedocs.io/en/latest/Generators/SiFive-Generators.html」
例として、GPIO を設定する場合は以下のように Scala で記述する。

```
class WithGPIO extends Config((site, here, up) => {
  case PeripheryGPIOKey => Seq(
    GPIOParams(address = 0x10012000, width = 4, includeIOF = false))
})
```

FPGA 書き込みデータ作成手順書で作成したイメージでは既にデバイス設定がされているため、本書ではこの設定を利用する。

プログラム書き込み後、JA_2 と JA_0 を接続すると LD1 が点灯、JA_2 と JA_1 を接続すると LD1 が消灯することが確認できる。



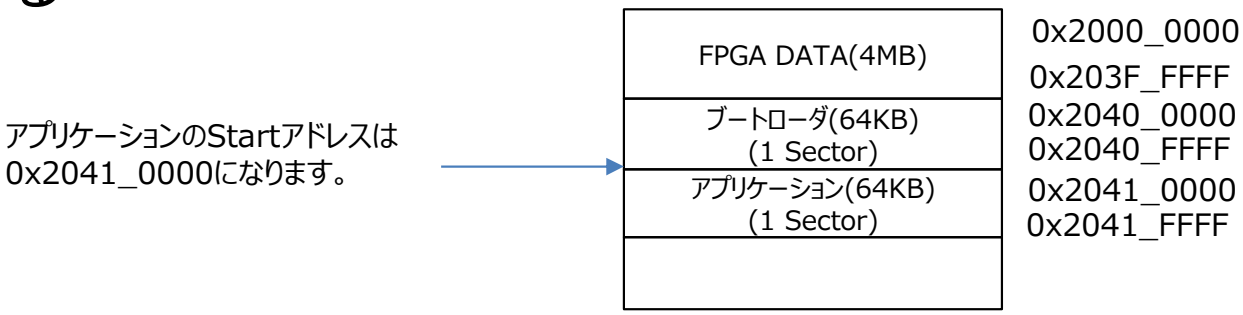
4-4. プラットフォームの概要②

Bootloader development

《②ブートローダの開発》

Arty A7-35T上のRISC-VコアのArduino開発環境から「アプリケーションの書き込み」と「実行開始」が出来るブートローダの開発

(1) ToolchainはArduino開発環境(GNU Toolchain)を使用
(2) アプリケーションの書き込みはAVRDUDE
　　(AVR Downloader/UploaDEr)で実行
(3) Arduino開発環境(GUI)で、アプリケーションおよびブートローダの書き込みを可能にする
(4) SW0(GPIO)による、アプリケーションの起動とブートローダの選択機能を設ける



一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

4-4-1. ブートローダの開発

ブートローダSourceCode一覧

ファイル名	
bootloader/Makefile	Makefile
bootloader/link.lds	Linkファイル
bootloader/bootloader.c	メイン・プログラム
bootloader/start.S	スタートアップ
bootloader/gpio.[ch]	GPIOライブラリ
bootloader/spi.[ch]	SPI/FlashRom読み書きライブラリ
bootloader/uart.[ch]	UARTライブラリ
bootloader/bootloader.hex	プログラム　HEXファイル

ブートローダはArduino開発環境に組み込まれており、ブートローダの書き込みおよびアプリケーションの書き込みはArduinoのGUI操作にて行う事ができる。

ブートローダをコンパイル

```
$ export RISCV_ARDUINO_TOOL_PATH=
        ~/.arduino15/packages/jasa/tools/riscv32-unknown-elf-gcc/3f7b3696217548bc31aeccf9a0c89bdfa4e16a8f/
$ cd ~/.arduino15/packages/jasa/hardware/riscv/1.0.0/bootloader
$ make bootloader.hex
```

ブートローダの書き込み（ Openocd ）

```
$ openocd -f openocd.cfg -c "flash protect 0 64 last off; program {bootloader.hex} verify; resume 0x20400000; exit"
```

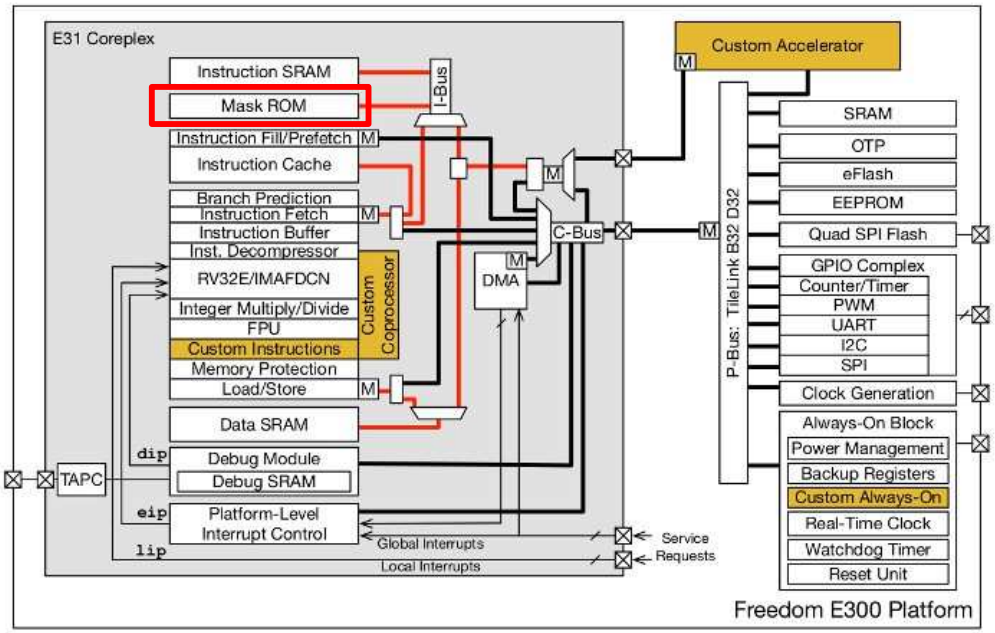
CLIでのアプリケーションの書き込み

SW0がOFFであることを確認した後、Arty A7-35Tの電源を入れる

```
$ ./avrdude -c avrisp -C avrdude.conf -P /dev/ttyUSB2 –b115200  -p rv32imac –U flash:w:hello.hex:i
```

4-4-2. ブートローダの開発

MaskROMを用いたブートローダの起動



E300プラットフォームのトップレベルブロック図 © 2017 SiFive Inc.

Freedom E310(Arty-A7) にはMaskROM（0x0001_0000）が実装されており、ここにブートローダを配置することで、JTAG（Openocd）を使用することなくArduino開発環境の構築が可能になる。

4-5. プラットフォームの概要③

Porting the Arduino IDE environment

《③Arduino IDE環境の移植》

実装したArty A7-35Tボード上のRISC-Vコアのソフトウェア開発が行えるよう、Arduino開発環境を移植

(1) パッケージ、ボードマネージャへの対応
(2) COM(UART)経由のスケッチダウンロード機能の実装
(3) JTAG ICE経由のブートローダ書き込み

「ツール」メニューから「ボード:"xxxxx"」を選択すると、「ARTY-A7_RISC-V Boards」が選択できるようになっています。

ここで、以下の「ARTY-A7_RISC-V」を選択してください。

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

4-5-1. Arduino IDE環境の移植

1. パッケージ化、ボードマネージャ対応

各設定ファイルをカスタマイズした、 GNU、SDK等をパッケージ化したものを開発してサーバに設置、 Arduinoのボードマネージャーを使用して、それらをインストールできるようにした。

「ツール」メニューから「ボード:"xxxxx"」を選択すると、「ARTY-A7_RISC-V Boards」が選択できるようになっています。

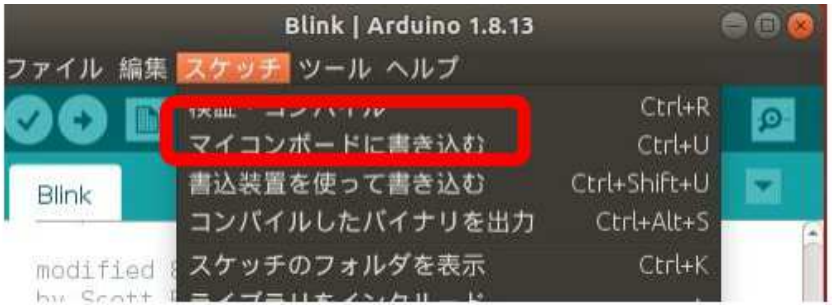
ここで、以下の「ARTY-A7_RISC-V」を選択してください。

4-5-2. Arduino IDE環境の移植

2. COM経由のスケッチダウンロードの実現

後述のブートローダが持つUARTダウンロード機能と通信し、Arduino IDEの「マイコンボードに書き込む」に対応できるようにした。

・各設定ファイルの編集
・必要バイナリファイルのパッケージ化



4-5-3. Arduino IDE環境の移植

3. JTAG ICE経由のブートローダ書き込み

後述のブートローダを、Arduino IDEの「ブートローダを書き込む」に対応できるようにした。

- JTAG ICE (ARM-USB-TINY-H)用設定をファイルに書き込み
- 必要バイナリファイルのパッケージ化

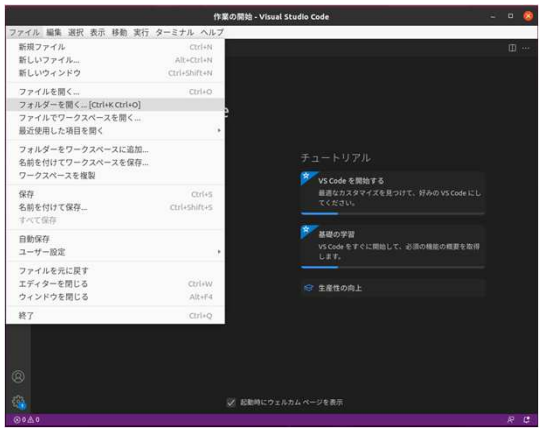
4-6. プラットフォームの概要④

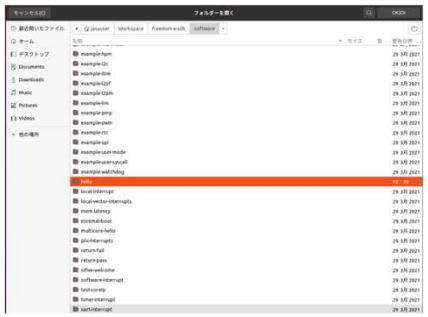
Setting up a VSCODE debug environment

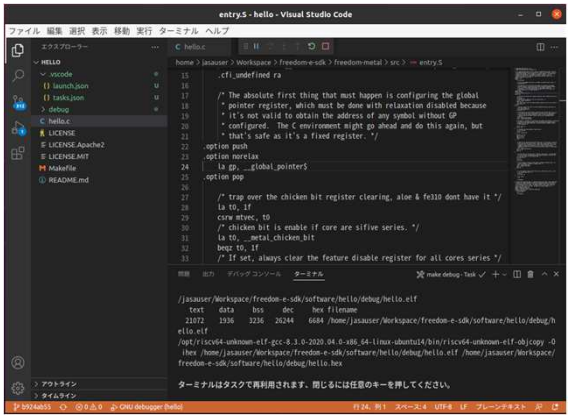
《④VSCデバック環境構築》

Visual Studio Code(VSC)を用いて作成したプログラムのデバッグ実行を行う手順についてのドキュメントを整備

(1) PCの環境やFPGAボードの事前準備
(2) Visual Studio Codeのセットアップ
(3) プロジェクトごとの設定方法
(4) デバッグ実行の実際







一般社団法人
組込みシステム技術協会
Japan Embedded Systems Technology Association

4-7. 開発内容その2:今年度の開発計画

Development schedule for this year

《開発ロードマップ》

2020年度	2021年度	2022年度
・Rocket Chipの FPGAへの実装 ・ブートローダ開発 ・Arduino環境移植	・VSCデバッグ環境構築	・64ビット版RISC-V コアのFPGA実装 ・OS移植 ・ブート環境

・市販FPGAボード上でRISC-Vを開発できるプラットフォームを継続開発します
・**全体を日本語でドキュメント化**し、初心者でも手軽に扱えます
・開発用、教育用プラットフォームとしてご活用いただくことを期待しています

・今年度は**産学連携**で昨年度までと同様なコンセプトの、OSが走るRISC-V プラットフォームを開発・整備する計画です

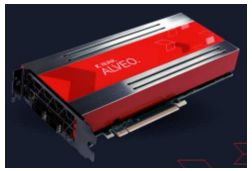
・開発終了後、JASA会員は自由に使えますのでご活用ください

一般社団法人 組込みシステム技術協会 Japan Embedded Systems Technology Association

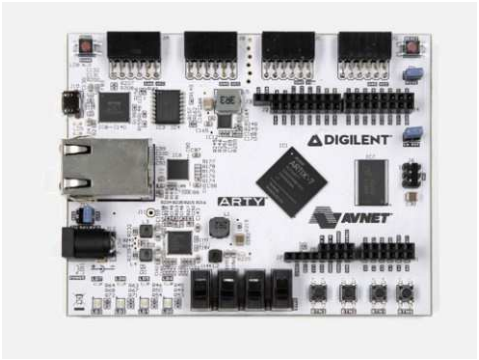
4-8. 今年度の開発内容

◆ 開発体制

東京農工大学 中條研究室
とのコラボによる、入手性が高く
安価なArtix7搭載FPGA
ボードへのRV64の実装



◆ ターゲットボード



◆ 今年度の目標

・「初心者のサンプル」として学部1年生が担当
・RV64を実装し、OS（Linux）の動作例を収集
・各実装例の詳細なドキュメントの整備
・独自プロセッサ開発に向けた設計・実装プロセスの確立

6. RISC-V協会との協創

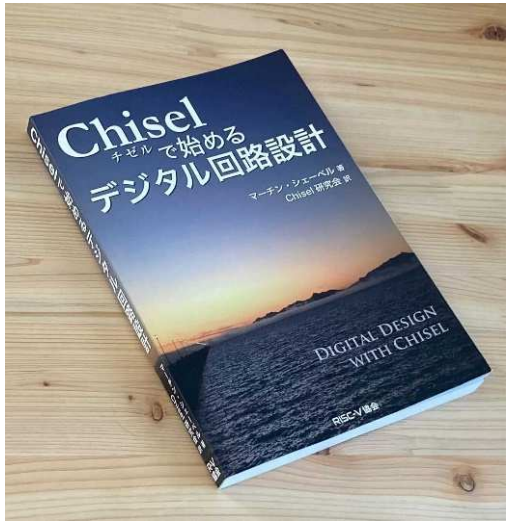
Collaborative creation with RISC-V Alliance Japan

《 ET&IoT展2021にて》

・JASAとRISC-V協会の共同プレスリリースを発行
・RISC-V協会のChisel本をET ＆ IoT展JASAパビリオンで配布
・RISC-V Days初日講演会に協力

《 RISC-V Days Tokyo 2022 Springにて》

・ 2日目の6月1日にJASA,展示会,WG紹介の講演を実施



RISC-V協会様のサイトより転載

今年度もコラボを進めてまいります

ご期待ください!!

一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association

RISC-V WGの活動とWGが進めるRISC-V研究・開発の取り組み

2022/11/16 発行

発行者　一般社団法人 組込みシステム技術協会
東京都 中央区 入船 1-5-11 弘報ビル5階
TEL: 03(6372)0211 FAX: 03(6372)0212
URL: https://www.jasa.or.jp/

一般社団法人 組込みシステム技術協会
Japan Embedded Systems Technology Association