IPA

2023 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

田中 邦裕（さくらインターネット株式会社 代表取締役社長）

2. クリエータ氏名

太田 涼介（東京大学 大学院新領域創成科学研究科）

3. 委託金支払額

2,736,000 円

4. テーマ名

自作マイコンの開発を容易にする開発環境

5. 関連 Web サイト

NextMicon の GitHub リポジトリ：https://github.com/NextMicon

6. テーマ概要

本プロジェクトでは、FPGA を用いた自作マイコンの開発を容易にする開発環境「NextMicon」の構築を目指している。

図 1 に示すように、マイコンとは CPU の他にペリフェラルなどの周辺回路をパッケージ化したものである。代表的なマイコンと開発ボードとして Arduino などが挙げられる。ただ、入出力のポート数やメモリの大きさなどは、既製品のマイコンの場合には自由に増減させることができず、そのため自らのマイコンを作りたいという動機が生じる。

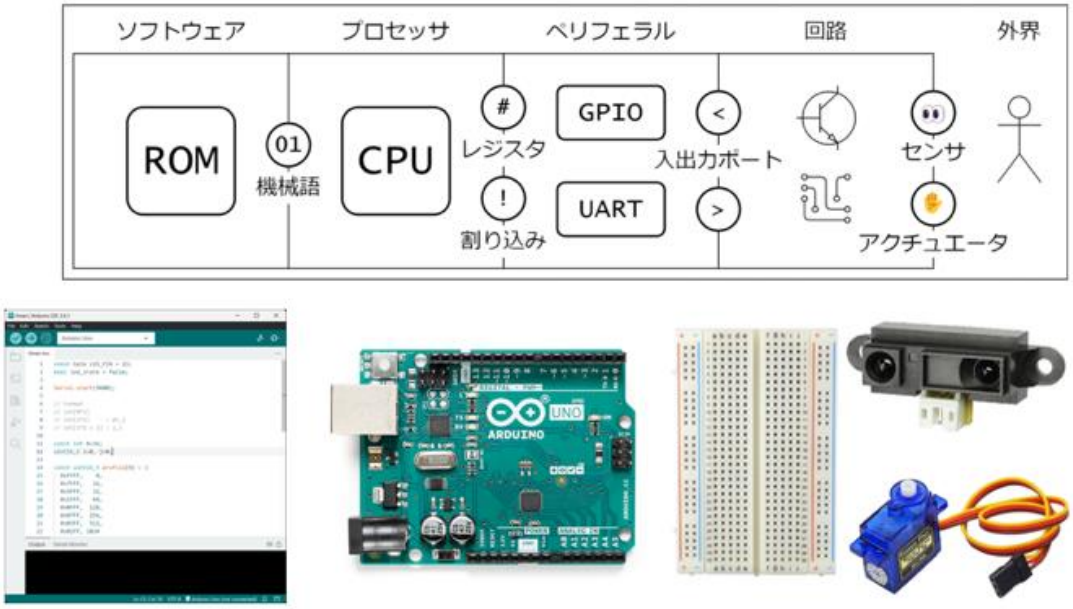

図 1：マイコンの構成と各種部品

FPGA は、ユーザが任意の論理回路をプログラミングによって構成できるデバイスであるが、従来の FPGA 開発環境は初学者にとって高い学習障壁が存在していた。本プロジェクトでは、FPGA 開発未経験者でも直感的に自作のマイコンの設計が行えるよう、ユーザフレンドリーな開発環境を提供することを目指した。

自作マイコンを設計することは、コンピュータアーキテクチャをハードウェア面で理解することができ、学習という観点でも重要性は高い。

7. 採択理由

本プロジェクトでは、FPGA を使って自作マイコンを作成したい人に向けて、GUI で簡単にそれを開発できる環境を作ることを目的としたものである。

提案者は、リレーを使った CPU を開発したり、ロジック IC を組み合わせた CPU を開発したり、多くの人が使うだけである CPU を、自ら作るものとしてモチベーションを持っており、今回のプロジェクトについては、本当に自分がやりたいことをプロダクトとして実現させたいというものである。自分がやりたいだけではなく、教育の観点でも FPGA 上で HDL を使わずに CPU を設計できるというプロダクトの価値は大きい。

その上で、さまざまなユーザが GUI 上で簡単に CPU を作れるようにするための UI/UX の設計やユーザテストなどは、単なる好きなことを超えてプロダクト開発のプロセスをなぞらないといけないものでもある。乗り越えないといけない課題は多くあり、未踏性があるものと考えて採択した。

8. 開発目標

本プロジェクトの開発目標は、FPGA の利用を容易にして、ユーザが作成したハードウェアとソフトウェアをパッケージ化して共有できる環境を作ることである。また、ソフトウェアはオープンソースとして公開して、開発ボードの販売による収益化も目指す。容易に FPGA 開発できるよう、マイコンを GUI 上で作れる IDE をマイコンエディタとして開発し、それらをもとに FPGA への書き込みを行うコマンド群を開発する。

- マイコンエディタ

  マイコンエディタは、ユーザがグラフィカルユーザインタフェース（GUI）を通じて自作マイコンの設計を行えるソフトウェアである。ユーザは、パレットから機能のパッケージを選択し、ドラッグ＆ドロップで CPU に追加することで、マイコンの構成を視覚的に設計できる。

  このエディタは Web フロントエンド技術を用いて開発され、React と SVG を使用してインタラクティブな UI を目指す。最終的には Electron を使用して、Windows, macOS, Linux 向けのソフトウェアとしてリリースすることを目標とする。
- マイコンジェネレータ

  マイコンジェネレータは、マイコンエディタで設計されたマイコン構成を基に、必要なファイルを自動生成するソフトウェアである。具体的には、ハードウェア記述言語（HDL）によるハードウェアの記述、C++のヘッダファイルとしてのファームウェア、メモリマップをコンパイラに伝えるリンカスクリプト、および FPGA への書き込みとプログラムのコンパイルを行う Makefile を生成する。これにより、ユーザはマイコンの設計から実装までのプロセスを効率的に進めることができる。
- パッケージ共有サーバおよびパッケージマネージャ

  パッケージ共有サーバは、ユーザが作成したハードウェアとファームウェアのパッケージを共有するためのプラットフォームである。パッケージマネージャは、これらのパッケージを管理し、他のユーザが自分のマイコンに統合できるようにするためのシステムであり、ユーザは他者の作成した機能を利用して、独自のマイコンを容易に構築できるようになる。

9. 進捗概要

本プロジェクトにおいては開発目標を達成するために、検証用自作マイコン「my-micon」、マイコン構成言語（MCL）、および統合開発環境「NextMiconIDE」を開発した。

本プロジェクトは主に 3 つの段階で開発を進めた。第 1 段階（6 月～7 月）では、マイコンを手動で生成し、自動生成する方法を検討した。第 2 段階（8 月

～9 月）では、マイコンを自動生成するシステムを作成した。第 3 段階（10 月～12 月）は、マイコンを開発するための IDE を開発した。第 4 段階（1 月～3 月）は、ユーザテストを行い、その結果を反映し改良するための設計変更を行った。また、適宜デモンストレーションを行うために、開発システムを用いたサンプルプロジェクトを作成した。

マイコンの自動生成システムを設計するにあたって、検証のために my-micon を実装した。my-micon のハードウェアの構成については、図 2 の通りである。

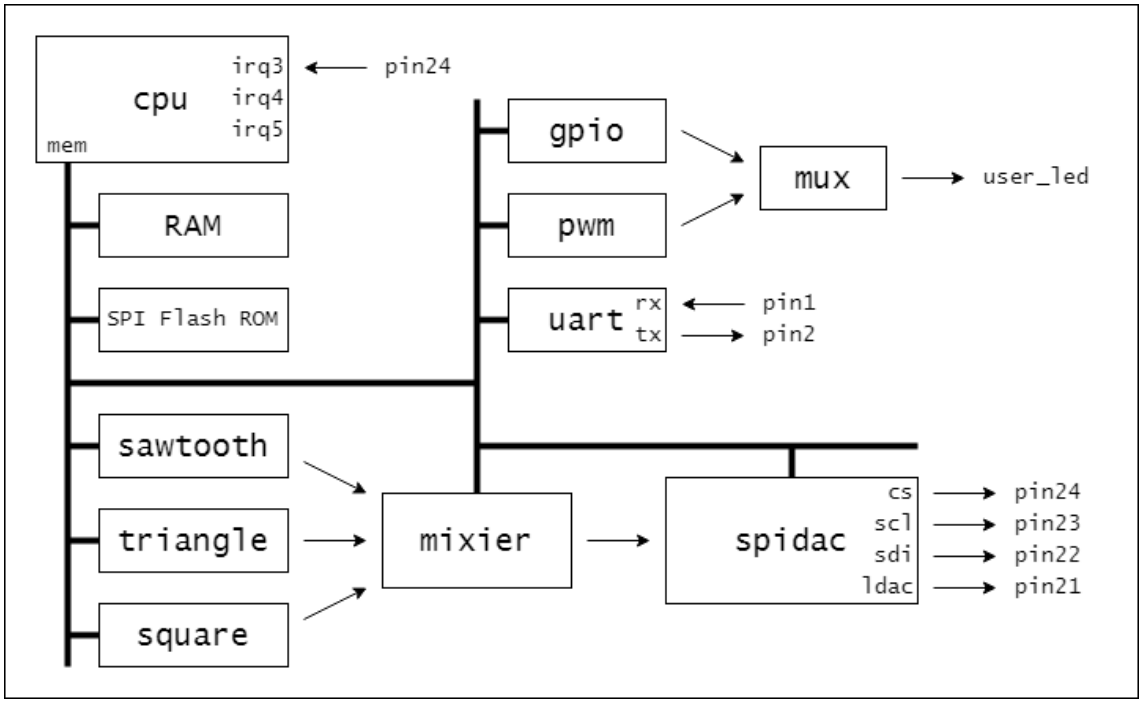

図 2：my-micon のハードウェア構成

ハードウェアの各構成は以下の通りである。

- CPU

  CPU は 32bit の RISC-V の基本命令セット（I）に乗除算命令セット（M）と圧縮命令セット（C）を加えた RV32IMC のオープンソース実装である PicoRV32 を用いた。RV32IMC は組み込み用途で標準的に使われている命令セットである。
- RAM

  FPGA 内部の Block RAM を用いて 8KiB の RAM を構成した。
- SPI Flash ROM

  ROM には FPGA ボード上にある SPI Flash を用いた。SPI Flash の空間のうち、前半部 0x0000_0000～0x0004_FFFF には FPGA のコンフィギュレーションビットストリームが格納されており、後半部 0x0005_0000～0x000F_FFFF にはユーザプログラムのバイナリが格納されている。CPU か

らは、ユーザプログラムの書き込まれている領域のみアクセス可能である。
　SPI Flash には、全二重通信で上下 1 本の通信線を用いる Single SPI と、半二重通信で 4 本の通信線を用いる Quad SPI がある。Quad SPI のほうが高速であるが、FPGA のコンフィギュレーション時には使用できない。そのため、FPGA のコンフィギュレーションは Single SPI で通信し、マイコンに制御が移ったら Quad SPI に切り替えるようにしている。通信方式の切り替えはメモリアドレス 0x0200_0000 にある SPI Flash ROM の Config レジスタで行う。

- Serial
　UART 方式のシリアル通信を行うペリフェラルである。メモリの 0x0300_0000 番地にある serial レジスタに 1byte の文字を書き込むことで、送信が開始される。また、同じレジスタを読み取ることで、受信した値を読み出すことができる。
- GPIO
　汎用入出力（General Purpose In & Out）のペリフェラルである。GPIO を制御するには 3 つのレジスタを使用する。メモリの 0x0400_0000 番地にある gpio.iosel レジスタは、入出力の方向を制御する。0 の場合入力で 1 の場合出力となる。0x0400_0001 番地にある gpio.out レジスタは、iosel を出力に設定した場合のみ有効で、レジスタにセットされた値が gpio から出力される。0x0400_0002 番地にある gpio.in レジスタは、iosel を入力に設定した場合のみ有効で、gpio に入力された電圧が読み取られセットされている。
- PWM
　PWM（Pulse Width Modulation）によって疑似的にアナログ電圧を出力するペリフェラルである。0x0500_0000 番地にある pwm.duty レジスタに 0〜255 のデューティ比を入力すると、pwm からレジスタに設定されたデューティ比の PWM 波形が出力される。
- MUX
　2 本の信号の入力をレジスタの値に従って選択して出力するマルチプレクサである。0x0600_0000 番地にある mux レジスタに 0 をセットすると一方の信号が、1 をセットするともう一方の信号が出力される。my-micon では、FPGA 基板上に 1 つだけ搭載された LED を用いて GPIO と PWM の両方のデモを行うための切り替えスイッチとして使用している。
- Square・Sawtooth・Triangle
　それぞれ方形波、ノコギリ波、三角波を生成する回路である。レジスタの値に反比例した周波数の波形を生成する。
- Mixier
　入力された各チャンネルの波形に音量を乗算し、加算した波形を出力する

回路である。

- SPI DAC
  SPI インタフェースで接続された外部の DAC を用いてアナログ電圧を出力するためのペリフェラルである。入力されたデジタル値を定期的にサンプリングして DAC に送信する。

本プロジェクト独自のマイコン記述言語である MCL を用いた開発フローについては、図 3 のような構成となっている。

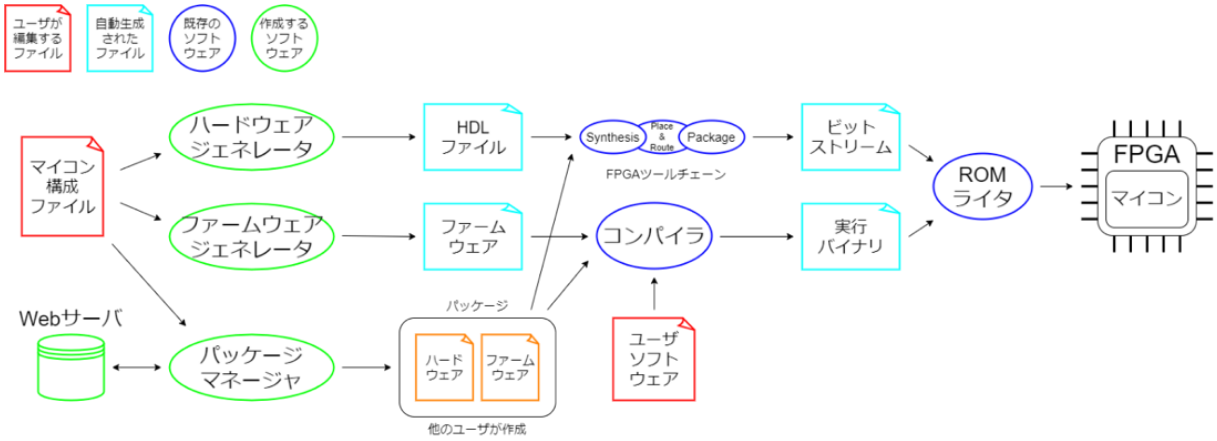

図 3：マイコン記述言語を用いたマイコン開発フロー

ユーザはまずマイコン構成ファイルを記述する必要がある。マイコン構成ファイルは、ペリフェラルや割り込みとそれらの接続関係を MCL で記述したファイルである。

次に、マイコン構成ファイルを 3 つのソフトウェアに渡し実行する。1 つ目のハードウェアジェネレータは、マイコン構成ファイルからマイコンのハードウェアが記述された Verilog ファイルを生成する。2 つ目のファームウェアジェネレータは、マイコン構成ファイルからマイコンのファームウェアが記述された C++ファイルを生成する。3 つ目のパッケージマネージャは、マイコン構成ファイルに依存しているパッケージを Web サーバ上からダウンロードし、Verilog と C++の依存関係を解消する。

次に、ユーザは生成されたファームウェアを使ってソフトウェアを記述する。

最後に、通常の FPGA 開発フローに従って、FPGA の論理合成、ソフトウェアのコンパイル、FPGA への書き込みを行えば、FPGA 上でマイコンを実行できる。

これらに加え、MCL を GUI で簡単に構成するための NextMiconIDE を開発した。アプリケーションフレームワークとしては Electron を使用した。Electron はバックエンドに Node.js、フロントエンドに Chromium を用いてマルチプラットフォームなネイティブアプリケーションを作成できるフレームワークである。フロントエンドのフレームワークには React を、状態管理には Recoil を使

用し、アプリケーションの主要な機能を実装した。バックエンドは Node.js の機能を使用して、ファイルシステム、プロセス起動、ネットワークなど OS に関連する部分を実装した。

ハードウェア編集画面を**エラー! 参照元が見つかりません。**に示す。画面右側には設計中のマイコンの回路図が表示されている。画面左側には、パッケージタブ、基本要素タブ、情報タブの 3 種類のタブが表示され、画面左端のボタンで表示を切り替えることができる。パッケージタブ、基本要素タブの使い方は後述する。情報タブは、設計中のマイコンについての情報が書かれている。情報タブの一番上（Target と書かれている部分）には、ターゲットの情報が書かれている。**エラー! 参照元が見つかりません。**の例では、NextMicon が作成した TinyFPGA BX ターゲットのバージョン 0.0.0 であることを表している。右側の「＜」ボタンを押すとターゲットで利用可能なリソース（入出力ピン、割り込み、メモリ空間）の情報が確認できる。情報タブの 2 段目（Objects と書かれている部分）には、マイコンに含まれている要素の一覧表がある。図 4 の例では、入出力ポートが 2 つとインスタンスが 1 つ存在することを表している。表の要素の「＜」ボタンを押すと、その要素についての詳細情報が確認できる。情報タブの 3 段目（Wire と書かれている部分）には、マイコンに含まれているワイヤの一覧表がある。

図 4：ハードウェア編集画面

続いて、ソフトウェア編集画面を図 5 に示す。ソフトウェア編集画面の左側には、マイコンのペリフェラルとそのメンバ関数の一覧表があり、右側にはマイコンで動かすソフトウェアのエディタがある。ユーザは左側の一覧表を参照しながら右側でソフトウェアを編集する。右側の画面については、Arduino などのマイコンに付属するソフトウェア開発環境と同様のもので、自ら設計したマイコン上で、簡単にソフトウェアを記述することができるようにした。

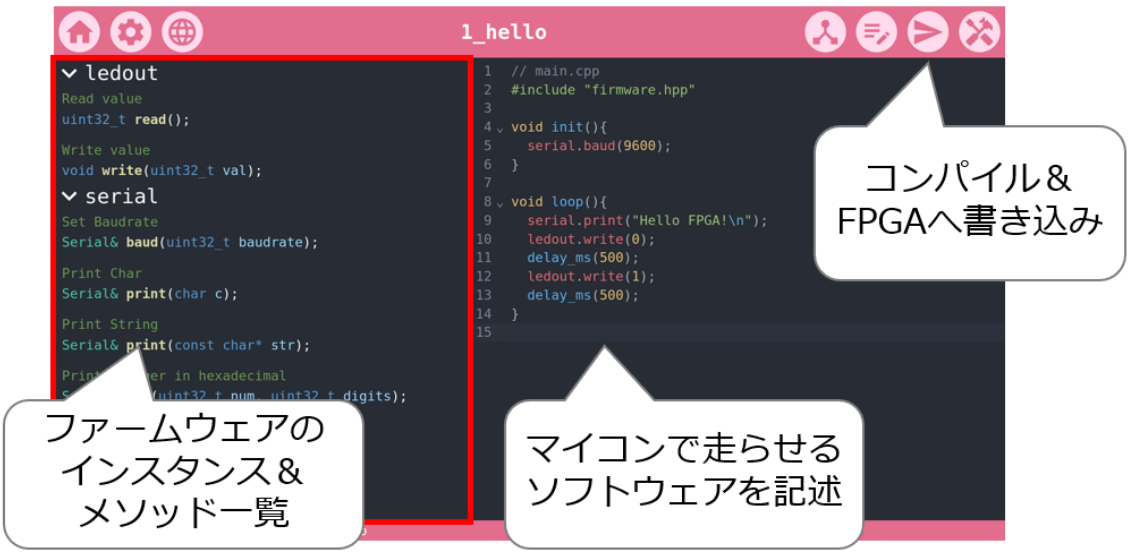

図 5：ソフトウェア編集画面

10. プロジェクト評価

本プロジェクトは、当初予定されていた内容を十分に果たし、大きな成果を達成したと考えられる。特に、ソフトウェアの開発において、非常に使いやすい統合開発環境が提供されており、ユーザが直感的にマイコンの設計を行えるようになったことは、プロジェクトの大きな成功点と言える。

パッケージ共有サーバおよびパッケージマネージャの開発は進行中であるが、プロジェクトの大きな目標である簡単にマイコンを作成できる環境の構築という面では、十分な達成となった。

総じて、本プロジェクトは、FPGA を用いた自作マイコンの開発を大きく前進させ、技術教育や組込みシステム開発の分野において重要な貢献をしていると評価でき、今後のボード開発の進展にも期待できる。

11. 今後の課題

本プロジェクトの今後の課題としては、利用者同士のコミュニティ育成が挙げられ、プロジェクトの継続的な成功において重要な要素である。特に、パッケージマネージャの構築は、この目的を達成するための鍵となる。

パッケージマネージャはユーザが作成したハードウェアとファームウェアのパッケージを共有し、他のユーザが自分のマイコンに統合できるシステムを構築するための重要なコンポーネントである。このシステムにより、ユーザは他者の作成した機能を利用して、独自のマイコンを容易に構築できるようになる。このプロセスは、利用者同士の知識や技術の共有を促進し、コミュニティの育成に寄与する。

併せて、パッケージの品質管理やセキュリティ対策も重要な課題である。さらに、コミュニティの活性化のためには、ユーザが容易にコミュニケーションを取

れるプラットフォームの整備や、定期的なイベントやワークショップの開催も効果的である。

また、当初より独自のボードを作りたいというビジョンが示されていたが、独自ボードがあることによって、コミュニティ間で知見の共有がしやすくなり、マネタイズにもつながることからコミュニティの継続性にも寄与するものと見られる。

総じて利用者同士のコミュニティ育成は、本プロジェクトの将来の継続性において重要な課題であり、これらを達成することで、FPGA を用いた自作マイコンの開発がより一層進展し、組込みシステム開発の新たな可能性が広がることが期待される。