

IoT プロトタイピングを円滑にするハードウェアと連携した Figma 拡張の提案

高橋 和也^{1,a)} 村上 雄哉^{1,b)} 渡邊 恵太^{2,c)}

概要：

Figma や Adobe XD といったツールは画面上のインタラクティブなプロトタイピングに優れているが、IoT のようなハードウェアと画面が連携した体験はプロトタイピングできない。本研究では、画面設計が容易な Figma を M5StickC と連携させる拡張手法を提案する。これにより、例えば加速度センサの数値に応じて Figma の画面を遷移させる、Figma 上のボタンを押して LED を点灯させるといったことができる。提案手法を用いて、IoT 製品のプロトタイプ作成をテーマとしたワークショップを実施した。

1. はじめに

新製品やサービスを創造する過程において、アイデアを具体化し意思疎通を円滑にする手法としてプロトタイピングが用いられる。例えば IoT サービスをテーマとするプロトタイピングやハッカソンでは、ハードウェアとソフトウェアが連携する様子を見せることや、試行錯誤を繰り返すことが大切である。

ビデオプロトタイピングや Wizard-of-Oz 法は、主に動画や演技を用いてアイデアを具体化して伝える手法として用いられる。一方、ビデオプロトタイプはユーザが直接体験することが出来ない、Wizard-of-Oz 法はハードウェアの挙動がユーザの操作に連動していると感じさせるのが難しいといった問題がある。プロトタイプをユーザ操作と連動させるためにハードウェアを実装した場合、ユーザが触れる画面の試行錯誤が難しく、製品に近い見た目を作るのに時間がかかる。ソフトウェアとハードウェアが連携していて、製品に近い見た目の体験可能なプロトタイプを作成するには、UI を簡単にデザインできるツールと、センサーと連携可能な仕組みを用意する必要がある。

ソフトウェア開発の領域では、簡単な操作で見た目のデザインや画面 UI の設計をおこなえるツールである Figma^{*1} や Adobe XD^{*2} が用いられている。これらのツールは、プロ

グラミング知識なしで簡単にモダンな画面を作成でき、レビュー機能を用いて実際のデバイスで動作確認を行うことができる。また画面制作のためのプラグインも数多く提供、開発されており、簡単に UI をデザインするプラットフォームとしてプロトタイピングに最適である。我々は、Figma にハードウェアを連携する仕組みを実装することで、製品のような見た目の体験可能なプロトタイプを素早く作成できるのではないかと考えた。本研究では、Figma を用いた UI、画面設計と実環境における電子回路の連携に着目し、センサーや LED 等の電子部品を活用できるようにした Figma の拡張手法を提案する (図 1)。Figma 上のオブジェクトや画面遷移設定と電子部品を Figma の機能を用いて連携することで、プロトタイピングにおけるソフトウェアとハードウェア間の連携を簡素化し、アイデアの価値検証をより円滑に行うことを目指す。

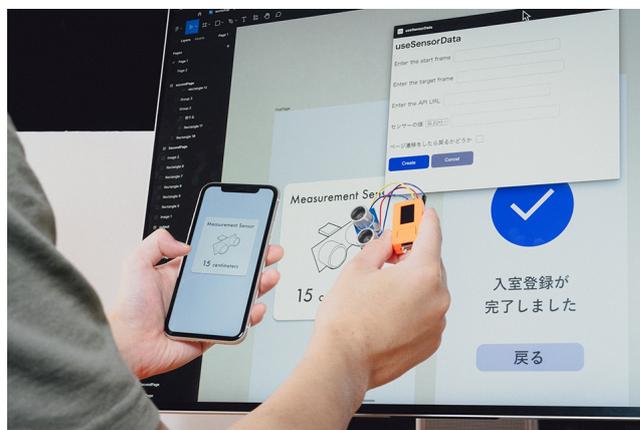


図 1 Figma を拡張し、超音波センサで取得した距離を画面に表示させた様子

¹ 明治大学大学院先端数理科学研究科

² 明治大学総合数理学部

a) cs222024@meiji.ac.jp

b) cs222035@meiji.ac.jp

c) keita_w@meiji.ac.jp

*1 <https://Figma.com>

*2 <https://www.adobe.com>



図 2 作成可能なプロトタイプ例. Figma で作成したクオリティの高い画面を、プラグインを用いてハードウェアと連携させることができる。

2. 関連研究

ハードウェアとソフトウェアを連携した IoT 製品のプロトタイピングは、ネットワーク通信や回路設計など様々な知識を必要とするため、特に初学者にとっては開発が難しいと感じる要因の 1 つになっている [1]。これまでの研究や企業製品は、開発コストを下げる手法と、ソフトウェアとハードウェアの連携を最適化する方針に分けられる。

ソフトウェアの開発コストを下げるアプローチとして、ビデオプロトタイピングのように映像をハードウェアに連携する手法がある。例えば、シースルー HMD やプロジェクタでハードウェアのモックアップに映像を投影する手法 [2] や、センサ情報に応じて動画の再生位置を動的に変更する手法 [3] があり、ハードウェアに対してソフトウェアが連動の様子をプロトタイピングできる。他にも、プログラミング言語の複雑さを解決するために、ビジュアル操作可能なブロックを繋げて Java 言語での IoT アプリケーション作成を支援する研究がある [4]。本研究の提案手法では、プロトタイピングツール Figma を用いてソフトウェアの画面制作を行う。プログラミング知識を必要とせずに簡単な操作で画面を作ることができるため、画面設計の価値検証を素早く行える。

Arduino^{*3}はオープンソースハードウェアとして広く用いられ、IDE や入出力ポートの提供によりハードウェアの制作ハードルを下げる役割を果たしている。同様に SONY の MESH^{*4}や SwitchBot^{*5}といった製品は、ブロックベースのプログラミング環境や汎用的な入出力をサービスとして提供し、ユーザは簡単にセンシングやデータの送受信機能を実装できる。他にも、Arduino[5] や、ハードウェアの入出力を Web ブラウザで可視化する手法 [6] がある。提案手法では、同じくオープンソースハードウェアである

M5StickC を用いる。

ソフトウェアとハードウェアの連携を簡単にする取り組みもある。Corno らは、学生の IoT 開発経験を分析し、開発の障壁として複数のツールを連携することに課題があることを明らかにした。学生がコードを理解を素早く理解できるよう、プログラミング環境やランタイムに依存しないドキュメントを提案した [7]。他にも、ハードウェアの入出力状態をクラウド管理して提供する開発者向けツールキット [8] や、AR と IoT デバイスを紐付けるオープンソースフレームワーク [9]、異なるハードウェアを一元管理するファームウェアと開発環境 [10] などがある。提案手法では、Figma と M5StickC を連携するために Figma のプラグインを実装する。これにより、ソフトウェアとハードウェアの連携のために新しいツールや開発言語を学ぶ必要がなく、プロトタイピングの検証がより簡単になる可能性がある。

3. Figma 拡張の提案

本研究では、Figma を用いた UI・画面設計と実環境における電子回路との連携を実現する Figma 拡張を提案する。提案手法を用いて、例えば図 2 のように Figma を用いて作成した製品に近い画面を M5StickC のようなハードウェアと連携させることで、体験可能なプロトタイプを簡単に作成できる。

3.1 実装

Figma 拡張は M5StickC に 4 種類の入出力を備えたハードウェアとそれを管理する API サーバ (以下ミドルウェア)、TypeScript で記述された Figma プラグインを組み合わせたシステムである。Figma プラグインは外部の IP アドレスに直接アクセスできないため、ミドルウェアを経由して M5StickC の IP アドレスにリダイレクトする仕組みを実装した。M5StickC にはそれぞれ API サーバを設置

*3 <https://www.arduino.cc/>

*4 <https://meshprj.com/jp/>

*5 <https://www.switchbot.jp/>

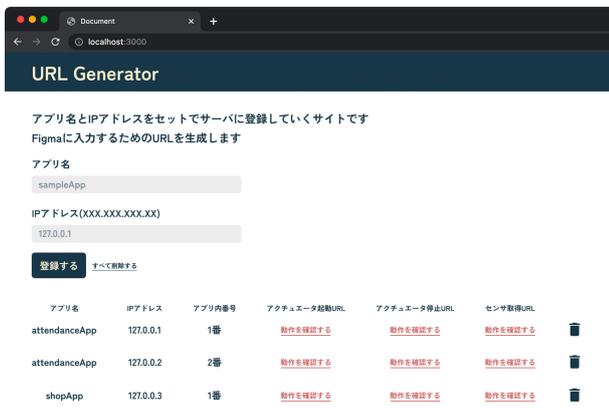


図 3 M5StickC の IP アドレスを登録する Web ページ

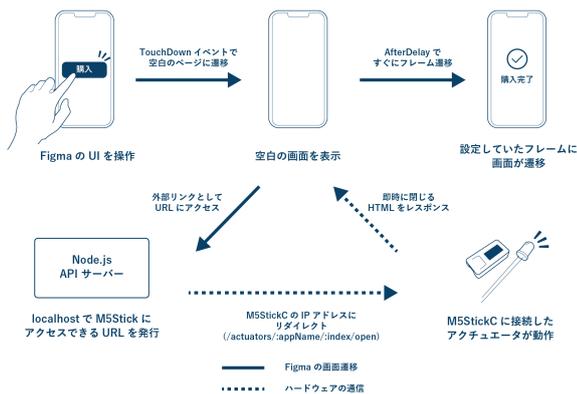


図 4 M5StickC のアクチュエータを Figma で操作する処理のシステム関係図

し、アクチュエータを操作する API とセンサの値を JSON 形式で返す API を構成した。

3.2 使用方法の手順

3.2.1 M5StickC をミドルウェアに登録する

はじめに、ユーザは M5StickC を Figma 上で用いるために IP アドレスを登録する。図 3 は M5StickC をミドルウェアに登録する Web ページ画面である。プログラムが書き込まれた M5StickC を起動すると、ディスプレイに IP アドレスが表示される。ユーザが IP アドレスを任意のアプリ名で登録すると、ミドルウェアがセンサ取得 URL(/sensors/:appName/:index/), アクチュエータ起動 URL(/actuators/:appName/:index/open), アクチュエータ停止 URL(/actuators/:appName/:index/close) の 3 つを発行する。Figma からこれらの URL に API リクエストを送信すると、該当する M5StickC の API サーバへリダイレクトされ、センサ情報の取得やアクチュエータの起動、停止を実行する。

3.2.2 Figma 上の UI から M5StickC を操作する

図 4 は、Figma のボタンを押す動作に合わせて M5StickC のアクチュエータを起動する場合のシステム関係図である。Figma 側では、ボタン押下前の画面と押下後の画面を空白のフレームを挟んで接続する。空白のページには Touchdown イベントを使用し、アクチュエータ起動 URL を開くように設定する。Figma の Presentation ビューを

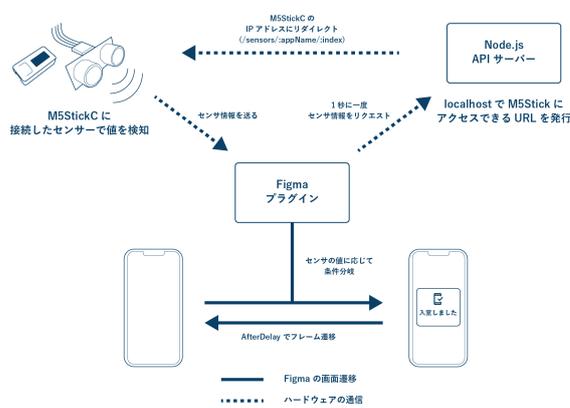


図 5 センサの値に応じて Figma の画面遷移を行う処理のシステム関係図

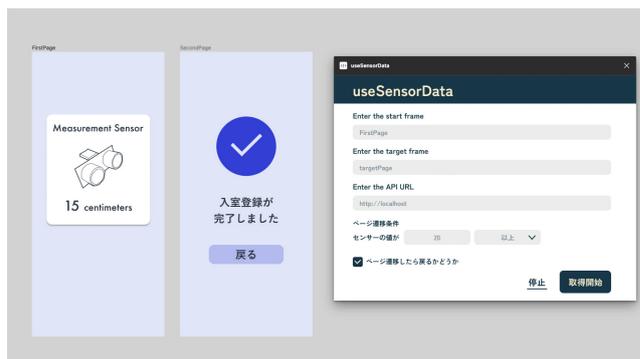


図 6 Figma プラグインの UI。図中の設定ではセンサの値が 20 を超えたら次ページに遷移する

起動して Figma 上のボタンを押すと、アクチュエータ起動 URL から M5StickC の IP アドレスにリダイレクトされ、アクチュエータが動作する。Figma 上では M5StickC からのレスポンスとして Web ページを開くが、この Web ページは JavaScript ですぐに閉じられ、ボタン押下後の Figma 画面へ遷移する。

3.2.3 センサ情報に応じて Figma の画面遷移を行う

ミドルウェアが発行したセンサ取得 URL を用いて、センサ情報に応じて Figma の画面遷移を行うことができる。図 5 は、超音波センサの値に応じて Figma の画面遷移を行う場合のシステム関係図である。

ユーザは Figma プラグインを起動して移動元と移動先のフレーム名、ミドルウェアで生成したセンサ取得 URL を入力する(図 6)。ページ遷移するセンサのしきい値設定や、しきい値を下回った際に元のページに戻るかどうかを設定した後、取得開始ボタンを押すと、Figma プラグインが 1 秒に 1 回ミドルウェアにセンサー情報をリクエストする。リダイレクトされた M5stickC から送られたセンサ情報の値としきい値を比較し、ページ遷移を行う。他にも、センサの値を直接テキストオブジェクトへ代入する機能や、複数台のセンサ数値をもとに画面遷移の条件を詳細に設定するプラグインの開発も可能である。

4. ワークショップの実施

提案手法を実際にどのように使ってもらおうべきか確か

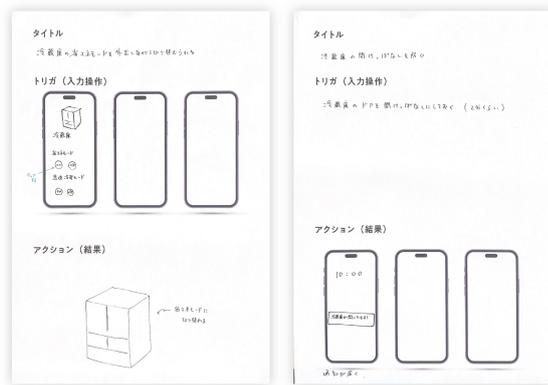


図 7 ワークショップで使用したアイデアシート。ユーザが操作する画面や、動作によって起こる画面の変化もイラストで自由に記述してもらった

めるために、IoT 製品のプロトタイプ制作をテーマにしたワークショップを開催した。ワークショップでは、参加者に対してプラグインの使用感や改善点、プロトタイピングの感想を収集するためのインタビューも行なった。情報系学部に通う男子大学生 3 名が参加し、いずれも Figma を用いた画面設計と ESP32 を用いたハードウェア制作の経験があった。ワークショップは大学の研究室内で実施し、3 名の参加者は 1 組として 1 つのプロトタイプ制作を行なった。なお、参加者には実験終了後それぞれ 1000 円の報酬を与えた。

4.1 手順

参加者には、新しい IoT 製品のアイデアを複数出してもらい、その中から一つを選んでプロトタイピングしてもらった。プロトタイピングには提案手法を用いて拡張した Figma 環境と電子部品の連携を利用した。プロトタイプが完成した後、実験者に対して制作したデモを実演してもらった。

実験中、参加者が Figma プラグインの使用法や電子回路の制作に不明点があればいつでも実験者 2 名に質問できることとし、実験者は参加者の制作状況に応じてアドバイス可能にした。

4.1.1 アイデア出し

はじめに、参加者に対して今回制作するプロトタイプのテーマを共有し、その後アイデア出しを行なった。テーマは「ハードウェアとスマートフォン画面を用いた IoT 製品の開発」とし、アイデアの作成には図 7 のようなワークシートを用いた。アイデア出し用ワークシートには、「スマートフォンの画面操作で実環境に影響を与える」ものと、「実環境の変化に応じてスマートフォンの画面表示が切り替わる」ものを記入できるような 2 種類のフォーマットを用意した。また、それぞれの例として実際の IoT 製品事例を数種類示した。アイデア出しは、設定した 10 分間の制

限時間に達するまで続け、参加者からは 7 個のアイデアが出された。

4.1.2 事前教示

ワークショップでは、出力用のハードウェアとして発光ダイオード、入力用のセンサとして超音波センサ、圧力センサ、加速度センサを用意した。ブレッドボードには予めアルファベットの記載で接続箇所を分類しており、LED やセンサの導線を、それぞれ対応するアルファベットの位置に接続して入出力を行う形式とした。電子部品の接続をする際には、ユーザが M5StickC 本体のボタンを押下してモードを切り替え、どの電子部品が接続されているかを入力することを伝えた。

次に、Figma のプラグインと M5StickC の利用方法について 15 分程度の操作説明を行なった。超音波センサの値に基づいて Figma のページ遷移を行うデモと、Figma 上のボタンを押すと LED が点灯するデモの 2 つについて、実験者が制作を実演した。

制作用の PC は 3 台用意した。それぞれ同じ Figma ファイルを開いた状態で参加者へ渡し、共同編集が始まっている段階でプロトタイピングを行なってもらった。また、Figma ファイルにはあらかじめハードウェア入出力に必要な最低限の画面を用意した。参加者には、ハードウェア出力用リンク設定もしくはセンサ入力に係るプラグインを用いた画面遷移設定、および画面設計を行なうことが求められた。

4.1.3 プロトタイプの制作

参加者が制作方法について理解したことを確認したのち、最初の手順で出たアイデアの中からひとつを選び、そのプロトタイプ制作を 40 分間で行なってもらった。実験参加者には目安の時間が経過した後、「自分たちのアイデアを実験者 2 名に説明できる」と感じたところで制作を終了するよう伝えた。完成したプロトタイプを 1 分程度で実験者 2 名に対して実演してもらい、最後にアンケートへ回答してもらった。

4.2 制作物

図 8 に、参加者が制作したプロトタイプの概要と画面構成を示す。大学構内でエレベータを呼び出す際、4 基あるエレベータそれぞれの乗車人数と現在位置を画面に表示することで、近くの階にいて、かつ空いているカゴを呼び出せるというアイデアのプロトタイプを制作した。参加者は、研究室内の狭い場所をエレベータのカゴに見立て、M5StickC に接続された超音波センサを設置した。エレベータに人が乗る、すなわち設置した超音波センサに人が近づくと、画面に表示された複数のエレベータのうち 4 番目のカゴの人数表示が 1 人増えるというプロトタイプを用意した。空いているエレベータをタップすると、自分がいる 3 階に当該エレベータが降りてくるというインタラク

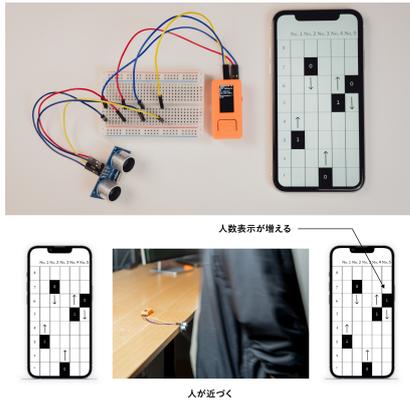


図 8 ワークショップで参加者が作成したモックアップとハードウェア構成。超音波センサに人が近づくとエレベータ内の人数表示が増加するアイデアのプロトタイプ

ションも実装された。

4.3 制作中の様子

プロトタイピングの前半において参加者 3 名は同時に Figma で画面を設計していた。後半では、Figma で画面設計する 2 名と、ハードウェアの設定等を行なう 1 名に役割を自主的に分担した。その際、画面設計を担っていた参加者のうち 1 名がこのアイデアの発案者であり、もう一方の画面設計者と頻りにアイデアのすり合わせを行なっている様子が観察された。Figma の編集画面はリアルタイムで同期するため、一方の参加者が自身の考えを画面上に再現し、もう一方がそれについて意見していた。画面の大まかな設計に関する議論だけでなく、色や文字のフォントなどにも気を配っている様子も観察された。ハードウェアの配線・設定を行なっていた参加者は、ハードウェアの配線および設定が画面設計よりも早く終わったため、再度 3 台目の PC を用いて Figma の共同編集に参加し、主にプラグイン関連の設定をする様子が見受けられた。

プロトタイプ制作中には、最終成果物の画面のクオリティをどこまで上げたほうが良いのかについて質問を受けた。今回のワークショップではハードウェアとソフトウェアの連携部分に関する機能のプロトタイピングに重きを置いていたため、最低限、アイデアのコアとなる機能が再現されていることを目標とするよう参加者に求めた。

4.4 インタビュー

ワークショップ後に実施したインタビューでは、「リアルタイムに表示人数が変わるため、センサによる自動判定であることの効果を改めて実感した」との意見が得られた。提案手法によってハードウェアとソフトウェアが連携したからこその価値検証ができたと考えられる。

ハードウェアについて「音の鳴るデバイスや GPS 入力のできるデバイスが欲しい」との意見が得られた。また、

Figma プラグインについては、「圧力センサの重量のしきい値を複数設定して、それぞれの状況に応じた画面遷移をしたい」、「操作を教えてもらいながらでないと使いにくい」との意見が得られた。ハードウェア構成、Figma プラグイン双方において対応する機能の幅に関しては再考する必要がある。

参加者からは、「もし私が同じようなプロトタイプを制作する際には Processing を使用するだろう」との意見も得られた。これには、設計の自由度が高いことやインターネット上などにサンプルプログラムが多く公開されていて開発のハードルが低いことが理由として挙げられていた。しかし、同時に「プロトタイプを作る早さは Figma のほうが早いため、すぐにアウトプットする機会としては良いかもしれない」との意見も得られた。そのため、複数人が共同編集でき、すばやく見た目の整った画面を設計できる今回の手法を用いることが有用な場面もあると考えられる。

4.5 考察

今回実施したワークショップにおける実制作時間は約 40 分と比較的短く、参加者からも「すぐアウトプットできるので良い」との意見があったことから、提案手法によってプロトタイプ制作が円滑になった可能性がある。一方で、Figma プラグインに関しては「操作を教えてもらいながらでないと使いにくい」といった意見も得られた。これは Figma プラグインの操作手順にやや複雑な部分があったことが原因の可能性はあるが、「一回やっつけてしまえばできそう」といった意見も得られた。そのため、ツールそのものが価値検証の効率を悪くしないために、操作に関するインストラクションの時間をより長く取って十分な説明を行なうことや、Figma プラグインのインターフェースの改善が必要である。

5. 議論

5.1 入出力の限界

本研究のワークショップでは、入力に M5StickC と 3 種類のセンサを使用した。提案手法では、接続したセンサの値を M5StickC 内部で処理してから送信するため、前もってセンサ値の処理をマイコンボードに書き込んでおく必要がある。また、本研究で実装した処理は数値の大小関係を利用した真偽値判定であり、センサの出力値がユーザの設定した値を超えているか否か、もしくはその逆かどうかを判定して Figma の画面を制御している。そのため、音声入力や GPS といったセンシングが難しくかつ複雑な処理が必要で、出力値が単純な 1 次元の数値でない場合、プロトタイプのクオリティを上げるのが難しい。

今回のワークショップでは、「ハードウェアやセンサと Figma を連携させる必要性が感じられなかった」という意見が得られ、プロトタイピングで実現したいアイデア

の規模が大きい場合、連携させるメリットが感じづらくなるのが課題として挙げられた。これは、今回のワークショップが小型電子部品の制御や限られたセンサ入力のみを前提としたものだったためであり、高電圧に耐えられるリレーを使用して大電力のアクチュエータを制御したり、SwitchBotのような形状の汎用的な操作デバイスを使用すれば、より大規模なものを制御できる。

5.2 Figma の限界

ワークショップの参加者からは、「センサ入力のしきい値を増やして画面遷移を細かく行ないたい」など、Figma プラグイン機能を増強して自由度を上げてほしいという主旨の意見が得られた。しかし、Figma プラグインの自由度を上げるとさらに機能が複雑になり編集作業が容易でなくなる可能性がある。提案手法はプロトタイピングツールの一部であり、簡単な操作で画面設計を実現する Figma においてツール自体の操作が複雑になるのは避けるほうが良いと考えられる。センサ入力を前提としたプロトタイプ制作に求められる必要最低限の機能を検討し、実装する必要がある。

5.3 プロトタイプと実装の垣根

IoT 製品の開発においては、操作に用いるスマートフォン等の画面設計を含めた、ユーザとソフトウェア、ハードウェアのインタラクションが体験に大きな影響を及ぼす。そのため、画面操作とハードウェアの連携まで含めたデザインが重要であり、本番環境での実装に先立ったプロトタイピングでの価値検証も同様に重要である。Figma のコミュニティは活発で多くのデザインリソースが公開されており、これらを有効活用するとクオリティの高いプロトタイプの制作が容易になる。

このまま発展すれば、プロトタイピングツールの使いやすさはそのままに、最終的な実装・製品化まで同じ制作ツールで一貫して行なえるようになると分析する。ハードウェアとの連携を前提にしたソフトウェアにおいても同様に、プロトタイプがそのまま製品化される可能性もある。

6. おわりに

本研究では、Figma や Adobe XD といった画面上のインタラクティブなプロトタイピングに優れているツールとハードウェアを連携するための手法として Figma の拡張手法を提案した。本手法をどのように使ってもらえるべきか検証するためにワークショップを実施した結果、ハードウェアとソフトウェアが連携した製品のプロトタイプを短時間で制作できることがわかった。

しかし、ワークショップなどを通してハードウェア構成と Figma プラグイン双方の機能制限が原因でプロトタイプの忠実度を下げることがあった。今後はより多く

のアイデアをプロトタイプとして再現できるようにするため、どのような機能や構成が最適か検討する。

参考文献

- [1] Mäenpää, H., Varjonen, S., Hellas, A., Tarkoma, S. and Männistö, T.: Assessing IOT projects in university education: a framework for problem-based learning, *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*, ICSE-SEET '17, IEEE Press, pp. 37–46 (2017).
- [2] Nam, T.-J. and Lee, W.: Integrating hardware and software: augmented reality based prototyping method for digital products, *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, New York, NY, USA, Association for Computing Machinery, pp. 956–957 (2003).
- [3] Nicholas, M. J., Marquardt, N., Pahud, M., Riche, N., Romat, H., Collins, C., Ledo, D., Kadekodi, R., Chandramouli, B. and Hinckley, K.: Escapement: A Tool for Interactive Prototyping with Video via Sensor-Mediated Abstraction of Time, *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, No. Article 799, New York, NY, USA, Association for Computing Machinery, pp. 1–14 (2023).
- [4] Pramudianto, F., Kamienski, C. A., Souto, E., Borelli, F., Gomes, L. L., Sadok, D. and Jarke, M.: IoT Link: An Internet of Things Prototyping Toolkit, *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pp. 1–9 (2014).
- [5] Millner, A. and Baafi, E.: Modkit: blending and extending approachable platforms for creating computer programs and interactive objects, *Proceedings of the 10th International Conference on Interaction Design and Children*, IDC '11, New York, NY, USA, Association for Computing Machinery, pp. 250–253 (2011).
- [6] Dax, J., Ludwig, T. and Pipek, V.: Remotino: Supporting End-User Developers in Prototyping Embedded Devices, *SERVE@AVI* (2016).
- [7] Corno, F., De Russis, L. and Sáenz, J. P.: Pain Points for Novice Programmers of Ambient Intelligence Systems: An Exploratory Study, *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1, pp. 250–255 (2017).
- [8] Mora, S., Gianni, F. and Divitini, M.: RapIoT Toolkit: Rapid Prototyping of Collaborative Internet of Things Applications, *2016 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 438–445 (2016).
- [9] Blanco-Novoa, Ó., Fraga-Lamas, P., Vilar-Montesinos, M. and Fernández-Caramés, T.: Towards the internet of augmented things: An open-source framework to interconnect IoT devices and augmented reality systems, *The 6th International Electronic Conference on Sensors and Applications*, Basel Switzerland, MDPI (2019).
- [10] Kubitzka, T.: Towards a Toolkit for the Rapid Creation and Programming of Smart Environments.