Fast Prototyping Platform for Human Machine Interfaces Design in Smart Vehicles

Jakub Pilch [1, 2], Benaoumeur Senouci [1]
Centre for Industrial Mechanics (CIM)
[1] University of Southern Denmark (SDU)
Sønderborg, Denmark
japil20@student.sdu.dk

Morten Bøjskov Rothmann [2]
[2] Hamsø Engineering ApS
Sønderborg, Denmark
morten@hamso.dk

Abstract — In this paper, we present a solution to combat some of the development issues in the implementation of modern Human-Machine interfaces (HMI) solutions. We introduce the Capacitive Sensing Haptic Evaluation platform (CASHE), which is an evaluation board (EVK) with plug-and-play connections for sensing, haptics, and LED controlling, that supports the integration of modern HMI technologies to improve ease-of-use, user experience, and user safety. This paper shares current market issues and how the CASHE tool can help developers and designers create prototypes rapidly by: 1) Abstracting prototyping from code to PC program with intuitive User Interface. 2) Providing an all-in-one solution for capacitive sensing, haptic feedback, sound feedback, and LED control, and 3) allowing for seamless transitions from design to demo environment.

Keywords — Human-Machine-Interface (HMI), Haptic feedback, Capacitive Sensing, Rapid Prototyping, Evaluation Board (EVK), User Interface (UI), Hamsø Engineering ApS, Autonomous vehicles

I. INTRODUCTION

During the last decade, screens and smart surface solutions has become a synonym with high quality and top-of-the-line products. In automotive, mechanical buttons are slowly being exchanged with big dashboard screens, while household appliances and pharma equipment are getting flush smart surfaces with capacitive sensing and LEDs, making it easy to clean and use. [1]

With the transition from mechanical buttons to smart surfaces and digital displays, the tactile feedback has been missing. To solve this issue, haptic components, such as ERM's, LRA's, Piezo actuators and Exciters are used to create a vibration on the surface, mimicking the physical action taken by the user. The increased development time and cost plays a large role in the adaptation of the technologies, but are outweighed by benefits of functional safety, design possibilities, cleanliness, and production costs. Research has found that interactions that use visuals, sound, and haptics create the deepest understanding and if they are not tuned together, the overall quality perception is diminished and deemed poor. [2] [3]

The CASHE by Hamsø Engineering is designed to alleviate the upfront development costs and to allow the designer to focus on the application, not on the technical development of the hardware and software. The CASHE platform comes with intuitively designed UI program that allows the user to set and configure the sensing input, the haptic feedback, and the visual feedback.

II. RELATED WORKS

The Drv2624 [4] evaluation board designed by Texas Instruments was investigated as a related work. The board contains 7 capacitive sensing buttons and one haptic actuator output, with a PC application to interact with the device. While the program enables changing of many parameters and adjusting the performance of actuators, it requires knowledge about the device's registers and is not intuitive out of the box. Furthermore, it lacks LEDs control as well as sound feedback. The toolkit is designed for evaluation purposes only and cannot be used to directly develop new products.

The Da728xevalkit [5] by Renesas was also researched as a related work. This evaluation board and developer toolkit use similar principles to Texas Instruments' solution. However, the Da728xevalkit has several drawbacks that make it less than ideal. Firstly, it only includes three capacitive sensing buttons, additionally, while the PC program is slightly more user-friendly than its competitors, it still requires a certain level of expertise to use effectively. Furthermore, while the Da728xevalkit's pattern generating tool is extensive, it still has limitations and does not allow for as much customization as other tools on the market. Finally, the kit is only designed as an evaluation tool and cannot be used for direct product development, which may limit its usefulness to some users.

III. METHODOLOGY

A. Hardware

The CASHE builds on an embedded architecture that consists of an industry leading microcontroller that can be used for various sensing methods (capacitive sensing, inductive sensing, etc.) together with a haptic driving IC and LED controller IC. The power to the device comes with USB-C Power Delivery methods allowing for various power settings depending on the need. A barrel jack connector is also present, so that power can be directly taken from the application itself (i.e., power from a car). The CASHE has an inbuild ADC + DAC & Amplifier combo, so that it can record and play sounds, either directly through a speaker system or through its own amplifier

B. Computer Application

To ensure that the developer of HMI solutions can focus on the user experience and the application, the Capacitive Sensing Haptic Evaluation Board (CASHE) comes with a comprehensive PC program. Unlike current solutions on the market, the program is cross-platform, which means that it can be used on Windows, Linux, and Mac-OS. In the future, we plan to explore a web-assembly version and make the CASHE Configurator available from the browser to further enhance accessibility.

1) Communication

To make the usage of the platform as open and modular as possible, the communication between devices has been designed to resemble terminal/console commands. All the inputs to the microcontroller are sent using a human-readable protocol that utilizes parameters and flags similar to well-known operating system consoles, ensuring ease of use. More advanced users who would like to have more control over the setup can manually type in commands into a built-in terminal window or use their serial communication terminal of choice. The microcontroller will parse the commands and deserialize them into a configuration file that will be saved on non-volatile memory. Moreover, it is planned to include direct access to peripherals from the terminal to, for example, allow a manual trigger of a haptic feedback.

The communication layer of the application has been designed to be completely dependency-free, which provides possibilities to be a pluggable component for other UI applications, making it fully independent of the UI layer.

2) Flexibility

Moreover, the whole system has been designed to adapt to new board versions and types, supporting more or fewer capabilities with minimal to no changes. The platform design specifies the capabilities of each board through configuration files, depending on its type and software version. This approach not only allows for scalability and flexibility in the future when it comes to new products but also guarantees backward compatibility when new versions of the PC and microcontroller programs are released. This ensures that developers can use the CASHE platform without worrying about changes in the underlying hardware or software.

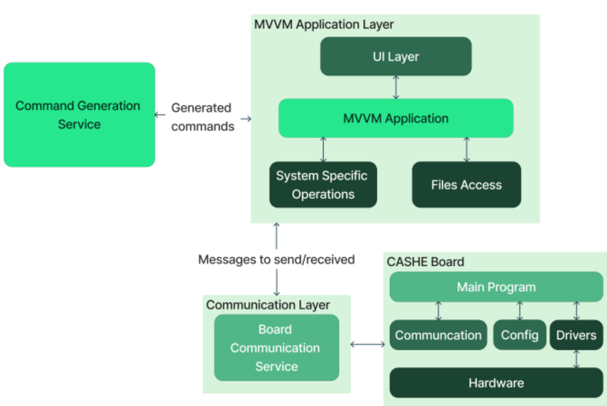

Fig. 1. CASHE Platform Diagram

C. Embeded program

1) Modules

To achieve maximum versatility and flexibility in the embedded program, it has been broken down into distinct modules, which can be easily reused in future application boards or different versions of the CASHE board. Among the modules is the communication module responsible for handling serial communication between the board and PC, while the command parsing module is responsible for parsing commands and their parameters from user input. The configuration module contains and handles all the necessary information about the uploaded configurations, which are sets of actions to be taken whenever specific input is detected.

2) Drivers

Additionally, there is a hardware abstraction layer defined by a set of interfaces to the drivers. This approach decouples the rest of the modules from the hardware they are utilizing and shifts the responsibility to control specific components on the board such as DAC/ADC, haptic driver, or LED controller to the drivers that are specifically written for the board. By implementing this modular design, the tool can be easily extended to accommodate new hardware and functionality while maintaining compatibility with existing modules.

IV. Resources

For the embedded program to respond to the user's inputs, it needs to go through a process that involves multiple modules and configurations. This process enables the program to perform the necessary actions based on the user's input. Here is a simplified explanation of how the sensing input is processed and dispersed through different configurations to perform all the necessary actions.

Algorithm 1: CASHE board input trigger handler

```
Data: Vector of triggered inputs
foreach trigger in the Vector do
    foreach config in Configs do
        if config.inputs contains trigger then
            foreach action in config.actions do
                find appropriate driver;
                driver handle action;
            end
        end
    end
end
```

The command parsing algorithm in the embedded program is responsible for taking in user input from serial communication and parsing it into predefined commands that the system can understand and utilize. Once the input is received, the algorithm identifies the command and its parameters, then verifies the correctness of the input before executing the corresponding action.

```
Algorithm 2: CASHE board command parsing
  Data: Vector of characters
  Result: Command with parameters
  initialize buffer variable;
  foreach c1 in the Vector do
      if c1 not a " " then
          add c1 to buffer;
          increment size_b;
      else if c1 is " " then
          match the command to buffer;
          if not matched then
              return inavalid command;
          end
          break loop;
  end
  initialize mode, buffer1, buffer2 variables;
  foreach c2 in the Vector starting at size_b do
      if c2 is "-" and mode is STANDARD then
          set mode to PARAMETER;
      else if c2 is letter and mode is PARAMETER then
          add c2 to buffer1;
      else if c2 is " " and mode is PARAMETER then
          set mode to VALUE;
      else if c2 is letter/number and mode is VALUE then
          add c2 to buffer2;
      else if c2 is " " and mode is VALUE then
          if buffer2 is not empty then
              proccess parameter of type buffer1 and value buffer2;
          else
              proccess a flag of type buffer1;
          end
          clear buffer1;
          clear buffer2;
          set mode to STANDARD;
  end
  return command with set parameters;
```

V. EXPLANATIONS

Developing an HMI interface implements multiple technologies and requires expert know-how in all of them with a high investment cost.

A. Computer Application

The PC program brings the embedded architecture together with the functionality. The main window of the UI shows the overall configuration on the board and will adapt to which EVK (in future also application board) is connected. With this, the developer can start setting up what should happen and when it should happen. This means that the developer only must consider the functionality of the application and not the extensive work that comes before it. The UI will handle all logic needed for the LEDs, haptic feedback, and sensing behavior. Normally developers are required to write custom code for each component, often spending considerable time studying technical documentation and troubleshooting errors arising from the integration of different functionalities. However, the introduction of the CASHE system has significantly streamlined this process. By leveraging an intuitive UI, even non-experts can achieve the desired outcomes with minimal programming knowledge. Through a simple selection process, the UI automatically configures the relevant components, enabling the rapid deployment of fully functional devices within minutes. Thus, CASHE represents a significant advancement in the field of HMI development, promoting ease-of-use and accessibility without compromising on the richness of available features.

1) Home View

The CASHE Configurator application is composed of two primary screens, each serving a distinct purpose. The first screen displays a list of previously created scenarios/configurations. These configurations are comprised of specific actions that are executed when designated conditions, such as digital or sensing inputs, are met.

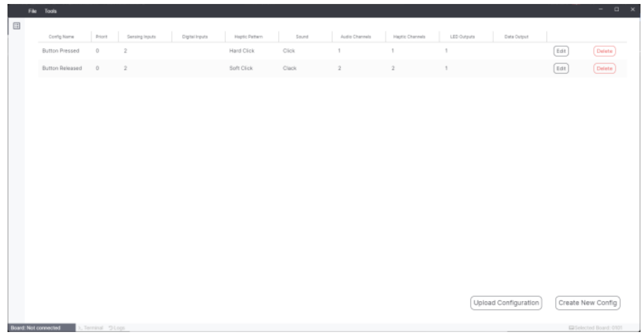

Fig. 2. CASHE Configurator - List of Created Configs Screen

2) Editor View

The second screen is an editor that enables the creation of new scenarios/configurations. This screen is specifically designed to follow the design process from left to right, starting with specifying the inputs on the left and the outputs on the right. In this editor, the developer can define triggers, such as digital or sensing inputs, and specify the corresponding actions to be taken once the trigger is activated. These actions include haptic or sound feedback, output channels, LED output, and digital logic outputs, such as a value sent over I2C to other devices. Each of the configuration entries has customizable parameters that can be modified to further personalize the configuration. An example of such parameters would be trigger for sensing inputs being either press or release, color and state of the LEDs or strength of haptic feedback output.

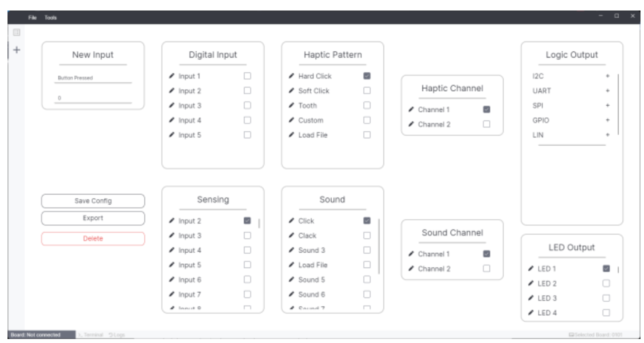

Fig. 3. CASHE Configurator - Config Editor Screen

The user-friendly interface of the CASHE application simplifies the process of creating and modifying scenarios, allowing developers to focus on the user experience and the application functionality.

VI. Discussions

With the world becoming increasingly digitalized, we interact more and more through various products and applications where we do not gain immediate feedback. As literature has outlined, our understanding is at its highest when the visual, tactile, and acoustic senses are used in combination. [6] For applications where fatalities or injuries can be the outcome of a misunderstood action, we must put an increased effort to the development and the interfaces we interact with play a large role in this. [7] Visuals and acoustics have been studied thoroughly over the past decades; however, haptic feedback has not received the same focus within research even though it is one of the most essential senses (and one of the first we as humans use). [8]

Within HMI solutions, several industries and technologies meet to create one final solution for the consumer. Firstly, you must consider the application and how it should look. Once you have settled on a smart surface or a screen solution, you must consider the electronics needed for the application. For a smart surface solution, you need to consider LEDs, sensing inputs, the haptic behavior, the tactile material of the surface, how it gets mounted and the curvature. For screens, you must select whether it should be resistive or capacitive screens. The next topic is the mechanical integration which is heavily affecting the haptic feedback design. Designers must consider where to mount the haptic actuator and if it should be global or localized feedback. These are only a few of the topics that must be addressed when designing a HMI solution, which is setting a high barrier of entry for any developing company looking into creating their own HMI solution for their prototypes or products. [9] [10] [11] [12]

By shortening the development time, companies of any size, can create professional and high-end quality products at a much lower expense. Not only are they able to save costs and have a shorter time-to-market, but with the shift of focus, more efforts can be put into creating user experiences that increase user satisfaction and user safety in industries where a lack of focus can lead to fatalities.

With the use of the CASHE EVK and Configurator application, Hamsø Engineering aims to disrupt the market in the hope to simplify the design process of HMI solutions and to put the focus where it really matters – on the application and the user.

VII. Conclusion

This paper has outlined the goal of the CASHE project from Hamsø Engineering, while defining some of the issues within HMI development. The paper has described the ongoing project development and the market perspective of HMI solutions within the industries of particularly automotive and household appliances. The paper has outlined the outlook of HMI solutions and addressed potential fields of interest for research.

References

[1] E. Altinsoy and S. Merchel , "Audiotactile Feedback Design for Touch Screens," *Haptic and Audio Interaction Design,* vol. 5763, pp. 136-144, 2009.

[2] S. J. Breitschaft, S. Clarke and C. C. Carbon, "A Theoretical Framework of Haptic Processing in Automotive User Interfaces and Its Implications on Design and Engineering," *Front. Psychol,* vol. 10, 2019.

[3] F. Beruscha, C. Brosi, W. Krautter and T. Altmüller, "22-3: Invited Paper: Deriving User Requirements for Haptic Enhanced Automotive Touch Screen Interaction," *SID Symposium Digest of Technical Papers,* vol. 47, no. 1, pp. 271-273, 2016.

[4] "DRV2624 evaluation board with capacitive touch," Texas Instruments, [Online]. Available: https://www.ti.com/tool/DRV2624EVM-CT. [Accessed April 2023].

[5] "Low Power High-Definition Haptic Driver Evaluation Kit," Renesas, [Online]. Available: https://www.renesas.com/br/en/products/interface/haptic-drivers/da728x-eval-kit-low-power-high-definition-haptic-driver-evaluation-kit. [Accessed April 2023].

[6] C. C. Carbon and M. Jakesch, "A Model for Haptic Aesthetic Processing and Its Implications for Design," *Proceedings of the IEEE,* vol. 101, no. 9, pp. 2123-2133, 2013.

[7] Gani, A., Pickering, O., Ellis, C., Sabri, O., Pucher, P. "Impact of Haptic Feedback on Surgical Training Outcomes: A Randomised Controlled Trial of Haptic versus Non-Haptic Immersive Virtual Reality Training". Annals of Medicine and Surgery, vol 83, p. 104734, 2022

[8] A. R. See, J. A. G. Choco and K. Chandramohan, "Touch, Texture and Haptic Feedback: A Review on How We Feel the World around Us," *Applied Sciences,* vol. 12, p. 4686, 6 May 2022.

[9] Y. Ujitoko and Y. Ban, "Survey of Pseudo-Haptics: Haptic Feedback Design and Application Proposals," in IEEE Transactions on Haptics, vol. 14, no. 4, pp. 699-711, 1 Oct.-Dec. 2021,

[10] M. Azadi and L. Jones, "Evaluating Vibrotactile Dimensions for the Design of Tactons," *IEEE Transactions on Haptics,* vol. 7, no. 1, pp. 14-23, 2014.

[11] Hossain, A., & Zaman, T. (2012, June), HMI Design: An Analysis of a Good Display for Seamless Integration between User Understanding and Automatic Controls Paper presented at 2012 ASEE Annual Conference & Exposition, San Antonio, Texas. 10.18260/1-2--21454

[12] Nikolaj Borisov, Benjamin Weyers, Annette Kluge, "Designing a Human Machine Interface for Quality Assurance in Car Manufacturing: An Attempt to Address the "Functionality versus User Experience Contradiction" in Professional Production Environments", Advances in Human-Computer Interaction, vol. 2018, Article ID 9502692, 2018.