Preventing Prototyping Pitfalls and Going Beyond: A Strategy for Affordable and Modular Wearable Embedded Systems

GLENN FERNANDES, Northwestern University, USA
MAHDI PEDRAM, Northwestern University, USA
NABIL ALSHURAFA, Northwestern University, USA

In recent years, the Human-Computer Interaction (HCI) community has shown a growing preference for employing custom printed circuit boards (PCBs) in research projects involving wearable technology. Despite the advantages of this approach, such as compact form factor and increased efficiency, designing and manufacturing custom PCBs can be time-consuming, costly, and burdensome for researchers, thereby delaying the research process. Furthermore, with recent advancements in chip manufacturing technology, there has been a surge in the availability of system-on-chip (SoC) boards with sensing and communication capabilities. However, the process of choosing the appropriate development board, sensors, and prototype design strategy that align with project requirements and can function efficiently in free-living settings may present a challenge, particularly for researchers who are advanced beginners in this area. Given our experience designing embedded mobile health systems, we present a prototyping strategy for developing wearable embedded systems by identifying potential pitfalls advanced beginners face when selecting and assembling components for prototyping wearable technology research projects. We also provide practical recommendations for selecting appropriate development boards, sensors, and prototype designs, serving as a valuable guide for researchers in the field.

CCS Concepts: • **Computer systems organization → Embedded systems**; *Embedded hardware*; *Embedded software*; *Firmware*; *Real-time operating systems.*

Additional Key Words and Phrases: prototyping, embedded systems, firmware, real-time operating systems

ACM Reference Format:
Glenn Fernandes, Mahdi Pedram, and Nabil Alshurafa. 2018. Preventing Prototyping Pitfalls and Going Beyond: A Strategy for Affordable and Modular Wearable Embedded Systems. In *ACM SIGCHI '23: ACM Conference on Human Factors in Computing Systems, April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 PROTOTYPING STRATEGY

Our proposed prototyping strategy not only addresses potential challenges that but also aims to achieve a reproducible, modular, small form-factor, and cost-effective prototype, as an alternative to designing and manufacturing custom PCBs. We present guidelines for selecting sensing modalities, designing a power supply system, addressing challenges associated with real-time clocks, wireless communication, dev board selection, and firmware design. Our strategy is accompanied by a real-world example and demonstrated through developing a sensing system for human behavior detection. To maintain continuity, we utilize the example of smoking detection throughout the paper.

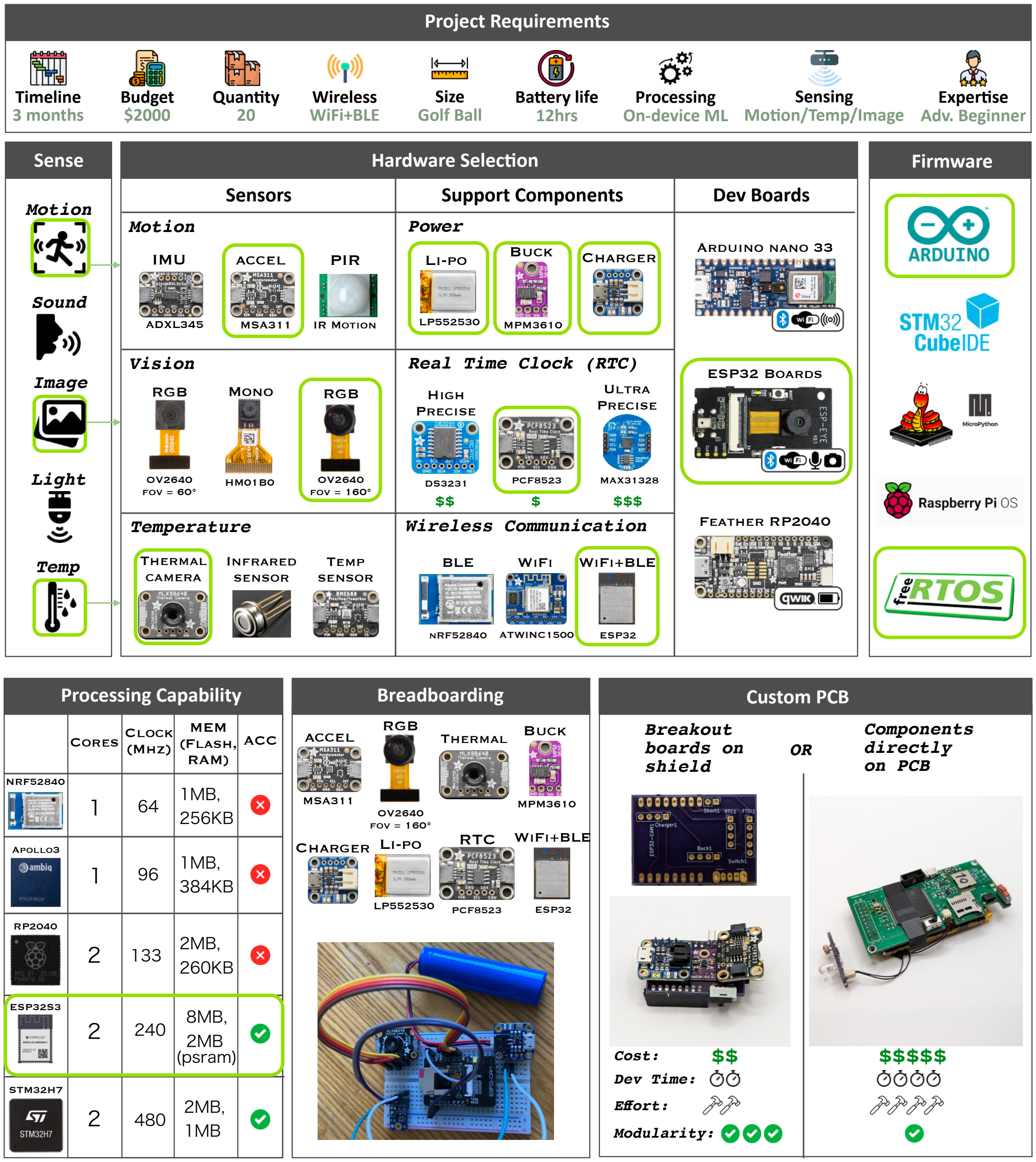

Fig. 1. Prototyping Framework: The figure describes our proposed strategy for developing prototyping systems that are scalable, reproducible, economical, and easy to repair.

1.1 Project Requirements

The first phase of the proposed framework involves identifying project requirements, including development timeline, budget constraints, wireless communication capabilities, device physical dimensions, energy consumption, on-device processing requirements, sensing modalities, platform selection and team expertise [1–3]. Analyzing these requirements provides a clear understanding of the associated challenges and constraints, and our guidelines offer solutions to potential pitfalls. As an illustration, for a wearable system designed to detect smoking behavior, we aim to create a device that primarily senses temperature and motion, uses image capture for visual confirmation, has a golf-ball size, a 12-hour battery life, and can transmit data wirelessly. The requirements for this example system are shown in the section at the top of Figure 1.

1.2 Hardware Selection

In line with our proposed methodology, we suggest the utilization of accessible and economical breakout boards that comprise development boards, sensing modules, and auxiliary power components. For example, developing a smoking detection wearable system with requirements for wireless communication, all-day functionality and accurate timekeeping necessitates a meticulous choice of hardware components. To avoid prototyping pitfalls, we provide an overview of the critical hardware components and their optimal application.

1.2.1 Development Board Considerations. Development boards have traditionally included a microcontroller with processor, RAM, Flash memory, programming interface, and I/O pins. However, recent advances in hardware have expanded the capabilities of development boards to include wireless connectivity, sensors, real-time clock, external memory and RAM, switches, and connectors, as well as LiPo battery support. It is crucial to avoid selecting a development board solely based on its microcontroller without considering whether development boards with integrated sensor or communication features exist. Developers must consider development boards that offer integrated sensor or communication features rather than solely selecting based on the microcontroller. These boards can potentially fulfill multiple prototyping requirements and are widely available and affordable. For instance, the Adafruit FeatherS2 development board can detect VOC gas, offer WiFi and LiPo battery support, and feature a Qwiic connector for easy sensor-to-module connection. On the other hand, the ESP32-CAM development board is suitable for continuous video capture and storage. Therefore, selecting a development board that closely matches project requirements is crucial for efficient and effective prototyping.

1.2.2 Real-time Clock (RTC) Considerations. Accurate timekeeping is a vital aspect of wearable embedded system design. While choosing development boards with integrated sensors and features has been emphasized, using the on-board microcontroller clock for timekeeping purposes can be appealing. However, this approach has drawbacks, including loss of timekeeping functionality when the system battery is depleted. Developers can address this issue through firmware development, but this poses a risk for deploying wearable technology in free-living settings, where repeated synchronization with another device is necessary. To overcome these challenges, we propose using a dedicated RTC breakout board, such as DS3231, PCF8523, or RV-1805 [4]. However, developers must carefully select an RTC unit based on their desired precision and resolution. For example, the inexpensive PCF8523 may gain or lose 2 seconds a day, while a more expensive and high-precision DS3231 can be a suitable alternative for high-precision timekeeping. Once the RTC is set, the dedicated battery enables accurate timekeeping for years, even without system battery charge.

1.2.3 Power Supply Considerations. Ensuring a stable voltage supply to the system is an important consideration in prototype design, as overlooking this aspect may present challenges for advanced beginner engineers. A voltage regulator and a battery charging mechanism are required to achieve this. For example, to maintain a stable 3.3V power supply, voltage regulators breakout boards such as AP63203, LD1117V33, LM3671, TPS62827, MPM3610, and TLV62569 can be used [5]. Developers need to consider the differences in available voltage regulators, such as voltage dropout, efficiency, maximum supported current draw, and quiescent current, when choosing one for their project. For instance, TLV62569 supports up to 1.2 A current continuously with an efficiency of 90-95% and can be used to output 3.3 V within a LiPo battery voltage range of 3.4 to 5.5V. In addition, for regular device charging, a LiPo charger breakout board with a micro-USB, USB-C, or other popular cables can provide a more straightforward operation.

1.3 Breadboard Testing and Custom PCB Shield

In previous discussions, it was mentioned that creating a custom PCB with on-board soldered circuits for communication with sensors, circuits for power supply, charging, and time-keeping requires more time, effort and is not very cost-effective. However, utilizing a custom PCB as a shield to house all breakout boards may present advantages. Nevertheless, before designing and manufacturing a PCB for this purpose, conducting comprehensive testing of all selected components on a breadboard is critical. This step is essential as it allows for a thorough evaluation of the system's performance from various perspectives, such as the system's electrical characteristics, correctness of implemented functions, and wireless communications. For example, a thermal camera was required in our smoking detection system, and sampling from the 32x24 thermal camera and writing frames into a microSD card was necessary. However, sampling from a thermal camera is relatively slow, and storing 768 float values in an external memory makes the entire process time-consuming. By identifying and addressing these issues early on, the performance and limitations of the system can be determined. Therefore, system performance evaluation is necessary to ensure the project's validity. Ultimately, the next step involves designing a shield (custom PCB) to assemble all components.

1.4 Firmware Development (Real-Time Operating Systems)

Selecting a suitable firmware development methodology is a crucial consideration that significantly impacts project outcomes. When the project requires on-device processing alongside concurrent processes such as data collection and wireless communication, the conventional super-loop programming approach, which involves a setup and loop function, may be inadequate. In such circumstances, a real-time operating system (such as Free-RTOS [6]) becomes essential. Besides scalability and modularity, Free-RTOS offers efficient resource utilization, multitasking capabilities, and reduced firmware development times for complex implementations. For instance, in processing thermal frames and detecting smoking on the device while ensuring uninterrupted data collection, RTOS facilitates concurrent process execution as opposed to the sequential execution typical in super-loop programming.

2 CONCLUSION

This position paper outlines potential challenges that advanced beginner researchers could encounter when developing wearable technology prototypes, as well as recommendations derived from our experience designing embedded mobile health systems. Our goal is to provide guidance to researchers in order to facilitate the efficient and effective development of wearable embedded system prototypes. To illustrate the efficacy of our proposed methodology, we intend to showcase demonstrations of mobile-health embedded wearable prototypes we have designed.

REFERENCES

[1] Alexandros Pantelopoulos and Nikolaos G Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2009.

[2] Mannu Lambrichts, Raf Ramakers, Steve Hodges, Sven Coppers, and James Devine. A survey and taxonomy of electronics toolkits for interactive and ubiquitous device prototyping. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–24, 2021.

[3] Hari Kishan Kondaveeti, Nandeesh Kumar Kumaravelu, Sunny Dayal Vanambathina, Sudha Ellison Mathe, and Suseela Vappangi. A systematic literature review on prototyping with arduino: Applications, challenges, advantages, and limitations. *Computer Science Review*, 40:100364, 2021.

[4] Adafruit Industries. Real time clock. [Online]. Available from: https://www.adafruit.com/?q=Real+Time+Clock&sort=BestMatch, 2023. Accessed: 2023-02-25.

[5] Adafruit Industries. Power converters. [Online]. Available from: https://www.adafruit.com/category/139, 2023. Accessed: 2023-02-25.

[6] Fei Guan, Long Peng, Luc Perneel, and Martin Timmerman. Open source freertos as a case study in real-time operating system evolution. *Journal of Systems and Soft ware*, 118:19–35, 2016.