Evaluation Form – Technical Background Review

Student Name: Robert Nelson

Project Advisor: Dr. David Anderson

Team Name: Team 1

Team Members: Rajan Vivek, Lycia Tran, Anubhav Agrawal, Shez Malik

________ / 30 Technical Content

- Current state-of-the-art and commercial products
- Underlying technology
- Implementation of the technology
- Overall quality of the technical summary

________ / 30 Use of Technical Reference Sources

- Appropriate number of sources (at least six)
- Sufficient number of source types (at least four)
- Quality of the sources
- Appropriate citations in body of text
- Reference list in proper format

________ / 40 Effectiveness of Writing, Organization, and Development of Content

- Introductory paragraph
- Clear flow of information
- Organization
- Grammar, spelling, punctuation
- Style, readability, audience appropriateness, conformance to standards

________ / 100 Total - Technical Review Paper

Robert (Robby) Nelson
Advisor: Dr. David Anderson
Group: Team 1

Choosing and Configuring a Microcontroller for a Smart Tennis Racquet

Section I, Background

In the modern world of sports analytics, athletes constantly desire more data about their performance, game, and areas of weakness. The sport of tennis is no exception to this, as top players and beginners alike look to technology to track their shot types, shot statistics, and even receive feedback about how to improve for their next match. In the current market, several different companies have emerged to offer various types of tennis metric products with the most common being a racquet mounted device that records the player's motion. In this area, embedded machine learning has the potential to disrupt the market by offering unparalleled accuracy and improvement over the existing competitors. In order to efficiently run the machine learning algorithms and deliver rapid user feedback, a state-of-the-art microcontroller setup is needed with capable computing hardware, electrical peripheral configuration, and correct driver code. This study will review the options for choosing a microcontroller for the smart tennis racquet and explore the electrical and driver configurations needed to operate the product effectively.

Section II, Current Microcontroller Technology

When choosing a microcontroller, there are many important factors to consider, and ease of access to development kits is on the top of the list for prototyping code and verifying that the microcontroller will work for the desired application before making a PCB supporting its functionality. Two companies in particular stand out for their development kits and seamless integration IDEs: STMicrolectronics, and Texas Instruments. Both TI Code Composer Studio and STM32 Cube IDE work under the surface to abstract the make file level and flashing procedure one level up from the user, and although moving to using make files without the IDE may be prudent for accuracy due to performance bugs in the IDE, having a well-supported free IDE is critical in the early stages of development. On the hardware side, peripheral interfaces such as SPI, I2C, JTAG, and UART are key since they are used to interface with sensors and modules on the PCB as well as debug the code. In order to see the state of the art in this area, an examination of one of STMicroelectronics top offerings [1], the STM32H743 line shows that the STM32H743ZI satisfies all of these peripheral requirements which will allow for it to communicate with all standard sensors via I2C and SPI while also being debugged with GDB via a cost effective JTAG such as the Segger J-Link [2]. I2C works via a two-wire communication protocol that contains a clock line and a data line for memory mapped IO access on sensors, making its simplicity ideal for small form factor sensors, and according to NXP, I2C is implemented on over 1000 IC's worldwide [7] which makes it a convincing option for the tennis PCB. Additionally, when collecting a rapid amount of data with sensors such as Inertial Motion Units (IMUs), Direct Memory Access (DMA) is an important parameter that is necessary to not occupy CPU time with writing data from peripheral interfaces to the memory, and it can also be seen via inspecting leading products from STM [1] that DMA is a standard feature in today's market. Finally, an important additional constraint is power requirements, especially in a sports context where battery size is limited to avoid disrupting an athlete's shot execution. In a study by Li et al. [3], an examination of power requirement evaluation using the TI CC2640R2 is done to investigate ways to reduce power consumption in edge devices. Li et al. concluded that sampling rate and payload size both had limited effect on the power consumption, whereas transmission via Bluetooth modules had the biggest drain on power. This means that a carefully executed strategy for data transmission to the user will be the most important factor rather than specific microcontroller parameters.

Section III, Microcontroller Specification Requirements

In this section an overview of specific additional requirements for the microcontroller beyond hardware basics is discussed. Specifically, when using embedded machine learning algorithms, it is important to consider which types of microcontrollers may be supported by common frameworks. TensorFlow Lite is one of the leading tools used in edge applications, and it is based on the powerful TensorFlow product by Google. According to their website [4], TensorFlow Lite is extensively supported on Arm Cortex-M Series microcontrollers, a common architecture found in the microcontrollers discussed in Section II. Additionally, TF Lite requires a 32-bit platform, a requirement that is again satisfied by the microcontrollers discussed in Section II. Another embedded machine learning framework, Edge Impulse, also states in their documentation that most microcontrollers are supported since their models can be built from data and then exported as a stand-alone library in C++ 11 [5]. This means that the microcontroller chosen must satisfy the 32-bit Arm architecture, but no other specific hardware is needed due to the easy integration with existing C++ code.

Section IV, Driver and Electrical Implementation

The final area to explore when choosing a microcontroller is the ease to which it can be configured on a custom PCB and the ease to which basic drivers for clock circuits, DMA, peripheral communications, and other interfaces can be built up. In addition to requirements when choosing a microcontroller, this section also explores the technical aspects of the PCB and base driver configuration. One critical tool in modern embedded development is the HAL, or Hardware Abstraction Layer. The

HAL works by hiding direct register reads and writes from the user, making the code easier to write since looing up register values in a datasheet not required. Tools such as HALCOGEN, a Texas Instruments tool allowing users to custom generate their HAL also offer a unique ability to test basic driver level code [6]. While complete features given by HALCOGEN are not likely needed on the smaller code base required for the senior design project, it is important to acknowledge their existence in the event that more customization or unit testing is required. Finally, with regards to electrical configurations, it is important to be aware of the tools available for the testing the microcontroller peripherals. When inputting data into the microcontroller, it might make sense to filter the data through a low pass filter to eliminate low frequency signals from slow accelerations to the racquet. While it is not practical to simulate a whole microcontroller in SPICE, it will be critical to simulate, fabricate, and test the external analog circuits before fabricating a PCB for the microcontroller.

Section V, References

[1] “STM32H743/753,” STMicroelectronics. [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32h743-753.html. [Accessed: 08-Oct-2021].

[2] “J-link edu – the educational J-link,” *J-Link EDU Debug Probe*. [Online]. Available: https://www.segger.com/products/debug-probes/j-link/models/j-link-edu/. [Accessed: 08-Oct-2021].

[3] J. Li, M. Bhuiyan, X. Huang, B. McDonald, T. Farrell, and E. A. Clancy, “Reducing electric power consumption when transmitting ECG/EMG/EEG using a bluetooth low energy microcontroller†,” *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2018.

[4] “Tensorflow Lite for microcontrollers,” *TensorFlow*. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers. [Accessed: 08-Oct-2021].

[5] “Getting started,” *Edge Impulse Docs*. [Online]. Available: https://docs.edgeimpulse.com/docs. [Accessed: 08-Oct-2021].

[6] “Halcogen,” *HALCOGEN IDE, configuration, compiler or debugger | TI.com*. [Online]. Available: https://www.ti.com/tool/HALCOGEN#tech-docs. [Accessed: 08-Oct-2021].