Knights Information Processing System (KIPS) MKII

Jessenia M. Argueta, Jonathan S. Meyers, Scott E. Valentine, and Harrison G. Mills

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Knights Information Processing System Mark II (KIPS Mk. II) is a wearable, wrist-mounted computing device, designed as an extensive improvement and overhaul of the earlier KIPS Mk. I prototype. This device is a combination of the most convenient aspects of desktop and mobile computing. To this end, the device has features including a low power operation mode, several sensors that collect outside data in real time, and various general purpose applications for everyday use. Additionally, there has been functionality and space left for users if they wish to add more specific applications. This paper presents a summary of both the hardware and software for the KIPS Mk II, and how these components work together to create a uniform functioning device.**

***Index Terms* — Wearable technology, mobile, desktop, battery, KIPS .**

I. Introduction

Inspired by the pocket PC and other smart devices, our group wanted to create a portable PC device with higher functionality, versatility, and modifiability than today's commercially available devices. KIPS Mk I was a project started for the 2024-2025 UCF IEEE Internal Projects Competition. It exists as a physical proof of concept to the actual vision of KIPS, which inspired us to develop Mk II for our senior design project.

In the same manner as KIPS MKI, Mk II is meant to be a bridge between mobile and desktop computing devices. It is a complete overhaul of the Mk I system, integrating new hardware systems and a new, separate software system. These new features are what allow Mk II to work beyond a proof of concept.

It operates as both a mobile smartwatch-like device and a desktop-class computing platform, switching dynamically between the two depending on power source or user initiated switching. When powered by battery, the system defaults to a mobile mode where a low-power mobile operating system and GUI is launched, whereas when connected to wall-power, the system enters desktop mode where a full desktop Linux environment is launched and peripheral interfaces (such as HDMI and USB) are enabled. The switch between modalities can also be initiated by the user as well.

One of the core focuses and features of KIPS Mk. II is its robust sensor suite integration, which includes an onboard heart rate sensor, two combined temperature and humidity sensors and a daylight sensor for screen brightness with auto-dimming. These sensors interface directly with the ESP32-S3 microcontroller unit (MCU), which enables the coordination of sensor data acquisition and transition to the Raspberry Pi Compute Module 4, which serves as the single board computer (SBC) of the project. Additional hardware features include AM/FM radio capability, an LED flashlight, D-pad controls, and digital-to-analog audio output.

II. System Components

KIPS Mk.II is composed of several major different individual components that work together to allow the device to function. These include the single board computer (SBC), microcontroller (MCU), sensor suite, batteries, and the digital to analog converter (DAC).

A. Single Board Computer

One of the main components of KIPS Mk. II is the SBC This component serves as the central computation module for our device. It is where the main software architecture is housed, and contains the operating system. The SBC also receives data from the MCU, and initiates the switch between mobile and desktop mode. The SBC takes care of the high level tasks that involve interfacing with the user and receiving outside data. Additionally, the SBC also communicates with the MCU to collect and process sensor data and control peripheral components of the device such as USB peripherals and HDMI outputs.

B. Microcontroller

Another crucial component to this build is the MCU. KIPS utilizes a microcontroller unit to communicate from the low level hardware to the SBC. The main purpose of the MCU is to handle and process analog data from the sensors, the audio suite, and the device's onboard AM/FM radio. The MCU will be connected to the SBC via a wired UART connection which will allow the SBC to send commands for the MCU to send information to the SBC. The MCU will then be connected to the sensor and peripheral systems in an I2C layout.

C. Sensor Suite

For use in K.I.P.S Mk II, several external and environmental readings were required. In order to satisfy these requirements, we selected a handful of sensors. Among these sensors we utilized two combination temperature and humidity sensors, a heartbeat sensor, and a Light Dependent Resistor (LDR) which functions as a detector of effective brightness.

The two temperature and humidity sensors are used for separate purposes. One will be accessible from a prebuilt app, where once the user enters the app, the sensor will be activated to give the user live readings of the temperature and humidity within their immediate area. The second sensor will be used to measure the internal temperature of KIPS as it operates with the intention of preventing overheating. Its readings will not be visible to the user except when the temperature exceeds acceptable values to operate, during which it will warn the user to take measures to prevent the device and the user from being affected by overheating and extreme temperatures.

The heartbeat sensor works as a built-in application within the device, and can be activated once the user enters the app. Once in the app, the user can request for a heartbeat reading, during which the heartbeat sensor will be activated for a period of 30 seconds, sending back readings to the MCU. The result displayed is an average heartbeat taken from all measurements gathered within that 30 second time frame.

The LDR is used to implement the autobrightness feature within the system. This feature can be toggled by the user in settings, and once it is set, the LDR will constantly read and send data to the MCU. This data is not seen by the user unless the current autobrightness settings are checked. Otherwise, brightness is manually adjustable by the user within the settings application.

D. Batteries

In order for KIPS to be usable on the go when not connected to wall power, another portable source of power was required. For this purpose, we ultimately decided to use batteries as the most optimal source of power in mobile mode. More specifically, the system will be running from Lithium-Polymer (LiPo) batteries because of their energy density, lightweight physical design, and rechargeability [1]. The configuration of these LiPo batteries will be in series at 8000 mAH with a protection circuit for a higher charge storage capacity and for safety of design. This will produce a voltage of 7.2V for the system, and any system that requires more or less than voltage can be run through various regulators. By using 2 cell LiPo batteries in series running at 8000 mAH, KIPS will be able to run at around an estimated 5 hours between full charges.

E. Radio System

The radio system was originally intended to be a feature of K.I.P.S Mk I, but due to time constraints, this was not possible. The radio system of Mk II features an AM/FM radio that is accessible in both power modes. This feature is meant to be both for entertainment and utility, being able to access common and emergency frequency bands. The mobile radio circuit is formatted as an integrated circuit (IC) due to ICs compact nature and lower power consumption, making it ideal for portable devices. For the purpose of signal clarity, the radio has two antennas, one for the amplitude modulation (AM) band and the other will be for the frequency modulation (FM) band. Ultimately, it was settled to use a whip antenna and a loop antenna for their small size, negligible reflection loss, and only needing to be used for receiving. The antennas themselves will be retractable for easier storage in the shell of KIPS, and must be manually extended if the user wishes to access the radio with clear signal quality,

III. System Concept

The overall project can be split into two separate systems: hardware and software. The software system runs on a combination of Python and C++ on a Linux operating system, which executes programs within the SBC and MCU respectively. It can essentially be grouped as the higher level Python programming running on the SBC, and the lower level embedded C++ scripting running on the MCU.

The Python code on the SBC is used to control the MCU and run scripts controlling other peripherals. The main mobile mode program runs automatically upon start-up, and evaluates what state the batteries are in. If KIPS Mk II is charging, it will set the device into desktop mode and run all peripherals. If it is not charging, the mobile mode application and GUI will launch instead, and automatically shut down all unnecessary processes, ports, etc to conserve battery power. Within the GUI, the user will have access to certain settings to enable features such as autobrightness which will poll the LDR and set the screen brightness accordingly. While in mobile mode, the internet will be inaccessible, and all programs will run locally.

The C++ code on the MCU is used to control all peripheral devices. Depending on what commands are sent to the MCU, it will poll the requested peripheral and transmit this data to the SBC. For example, if the user wishes to record their heartbeat, the MCU will activate and poll the heartbeat sensor for 30 seconds, and send the data in real time to the SBC. When the time window has passed, it will shut off the heartbeat sensor and show the user their average heartbeat.

The hardware consists of several different groups of sensors and circuit boards under the control of either the SBC or MCU. The SBC controls the following hardware groups: the interface and ports, peripherals, power supply, and MCU. When receiving a command from the SBC,the MCU controls the sensors and the audio suite in order to gather data. The user has a small touchscreen to navigate the GUI in mobile mode. However, there is also the option of using a joystick and arrow buttons to select items on the screen, similar to a game controller. To open applications or execute a software feature, there will also be a physical select button on the device that can be pushed to register input as well.

Additionally, as for ports, the USB Type A ports have been included for extending to storage devices, syncing data, or regular charging of certain other devices. So far the current plan is using USB 2.0, but USB 3.0 ports are a stretch goal moving forward. The micro USB-C port is essential for charging the main device itself. The Mini HDMI port will be an output that can be used for displaying the device as well as an alternative to using the embedded screen. This is due to the SBC chosen to have the capability of scaling up to 4K resolution.

B. Equations

In order to scale certain parts of KIPS Mk. II's radio circuit components to meet the requirements needed to allow them to function properly, several equations were needed.

Whip antennas generally do not need additional tuning in order to make them compatible with the FM band; nor is impedance matching extremely important due to their configuration as a receive only component. This means they can get away with using 75Ω coaxial connectors despite them having 36Ω impedances due to having a low reflection loss (appx -0.572 dB). This can be demonstrated with a simple mathematical calculation (1). However, since whip antennas don't have a positive and negative end to create a voltage centered at 0V, like a loop or dipole antenna, they need a sufficiently sized grounding plane to function properly.

$$e_r(dB) = 10log_{10}(1 - |\frac{Z_{load} - Z_0}{Z_{load} + Z_0}|^2) \text{ (1)[90]}$$

$$e_r = 10log_{10}\left(1 - \left|\frac{36-75}{36+75}\right|^2\right) = \sim -0.572dB \tag{1}$$

Loop antennas are commonly used in AM radios and shortwave (SW) applications for amateur radio enthusiasts. They tend to be much more compact than dipole antennas, which is important for our design. One feature of note is that loop antennas operate off of the magnetic field as opposed to the electric field like whip antennas and dipole antennas. This allows them to have excellent reception indoors [2], within reason (e.g. no thick walls or isolation). The ability to operate clearly indoors comes from the fact that e-field based antennas might experience more interference from electricity within a building due to conduits, outlets, etcetera. Another benefit of a loop antenna is that they have a narrow bandwidth influenced by their design. This, however, means they have lower sensitivity to signals [2], and their polarization depends on their orientation; either vertical or horizontal. A formula for a simple ferrite rod loop antenna can be found below (2).

$$\boldsymbol{L_f = \mu_0 \mu_e N^2 \frac{A}{l_f}} \text{(2)[92]} \tag{2}$$

IV. Hardware Detail

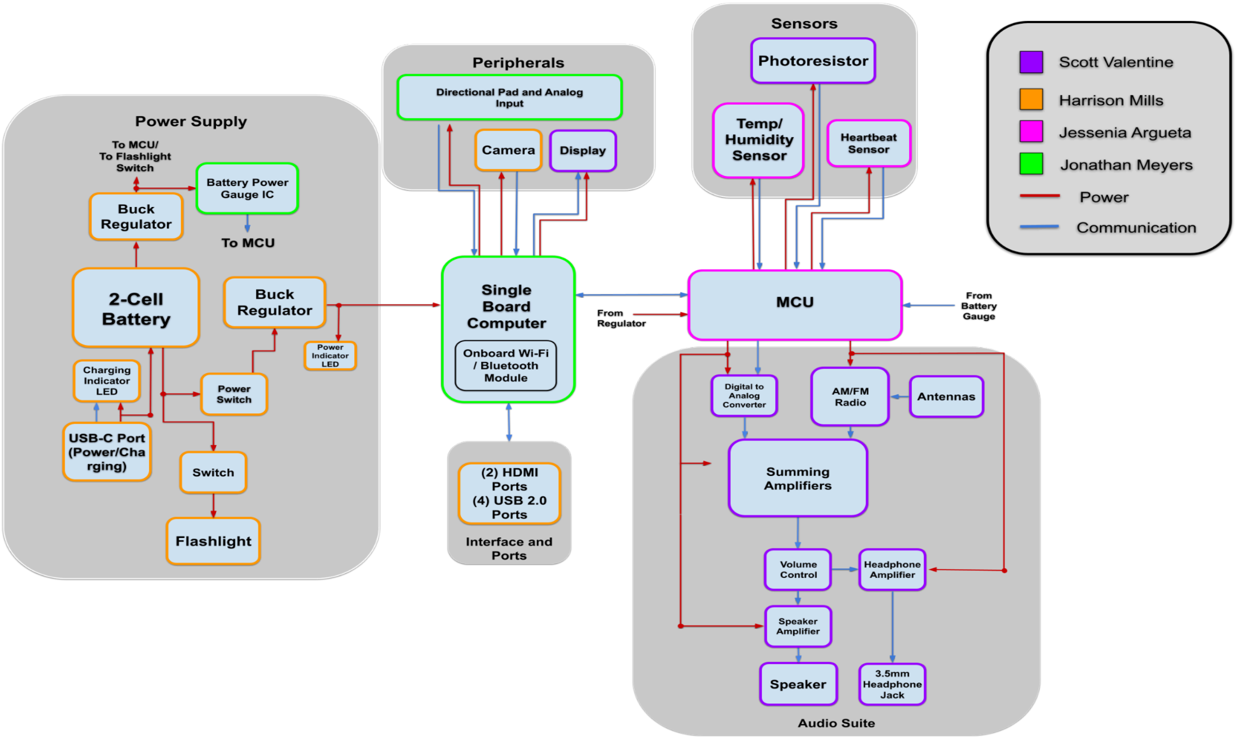
The hardware system consists of multiple components, connections, and subsystems. Namely, it is the power delivery system (PDS), SBC and MCU connections, sensors, ports, peripherals, flashlight, and audio suite.

A. Power Delivery Systems

The power delivery system is the backbone of KIPS Mk. II and supplies power for the entire device. The PDS controller ensures that the wall power line is 9V when it can. This connects to the USB-C port, which connects to a switching MUX. The MUX changes between the 9V line and the battery which has a nominal voltage of 7.2V and a max voltage of 8.4V. There will be some drop due to diode drops but that is due to the range, the charging, and the fuel gauge IC. Another aspect of the PDS called the buck regulator daughter board. There are two ICS, one which outputs a 3.3V line and another which outputs 5.2V. The 3.3V line will go into a sensor which needs the smaller voltage and the MCU while the 5.2V line powers the SBC and the USB controller. The input for the regulator will come in from the switching MUX. There is an LED to indicate activity on the board which is for troubleshooting.

B. SBC and MCU Connections

The main board contains the most important connections, which include the SBC and MCU. It is powered by the 3.3 V input rail and includes two ground pins tied to a common ground reference. The board features jumper headers that interface with the sensors, providing power,



Hardware Block Diagram

ground, and data connections for each. In addition to the sensor interfaces, the board also includes connections for a digital-to-analog converter (DAC). These interfaces ensure that all sensor readings and data collection are routed through this board. Due to its role as the central hub for sensor communication, it has the highest number of jumper connections, accommodating all necessary data lines.

C. Sensors

Next, are the sensors, the temperature and humidity sensor operates using a straightforward pin configuration. There is the supply pin (VDD), which is connected to a decoupling capacitor to ground in order to stabilize the local power supply and reduce noise, while also powering the internal circuitry of the sensor. Two pins serve as I²C communication lines (SDA and SCL). These lines must be pulled up to VDD with resistors to ensure proper logic-level operation and are connected to the I²C bus of the microcontroller for data transfer. Finally, there is the ground (GND), which provides the return path and completes the power circuit. This arrangement ensures stable power delivery, reliable communication over I²C, and proper grounding for the sensor's operation.

The photoresistor is arranged so that it functions as part of a voltage divider circuit. In this configuration, changes in light intensity translate into changes in voltage, which can then be measured as lux readings. The LDR itself varies its resistance depending on the amount of light it receives — higher light levels reduce resistance, while lower light levels increase it. A fixed resistor is placed in parallel with the LDR and connected to the 3.3V power rail, forming the other half of the divider. The junction between the photoresistor and the fixed resistor is then connected to the ADC port of the Raspberry Pi. This allows the system to measure the varying voltage and interpret it as corresponding light intensity values.

D. Ports/Peripherals

Next are the ports and peripherals. KIPS Mk II has 1x USB 2.0, 2x HDMI, and 2x 15 pin FPC connectors for the screen and camera respectively. All driver circuitry for the USB 2.0 port and HDMI ports are handled internally by the SBC. All circuitry for the screen connector is handled by a driver board behind the screen, and all circuitry for the camera is handled on the camera's driver board. These have predetermined trace sizes and lengths. Additionally, there is a daughterboard for the D-Pad and the thumbstick.

These are used for controlling the cursor of the KIPS Mk II and for scrolling. The board is powered by the 3.3 V rail for all of its inputs. Each direction is implemented as a separate component that comes together to provide the full range of directional inputs. Each input has a 10 kΩ resistor for current limiting. There are also X and Y inputs for the joystick, which connect to jumpers leading to the main board. These signals are programmed and processed on that board.

E. Flashlight

Next, the flashlight circuit is one of the simplest circuits. It consists of three main components: a resistor, an LED, and a switch. This configuration allows the flashlight to operate independently of the main computer, using power from the 3.3 V rail. The circuit essentially forms a loop from the voltage input to ground through the LED, with the switch controlling the connection to ground. When the switch is closed, the circuit completes, allowing current to flow and the flashlight to turn on.

F. Audio Suite

Last, the audio suite contains multiple vital components and subsystems. The Digital-to-Analog Converter (DAC) and the AM/FM integrated circuit are contained within one board. Both components operate at 3.3V but have separate power lines. Since these are analog components, they are connected to the analog ground (AGND). The DAC uses both polarized and ceramic capacitors for filtering and stability. It interfaces with the MCU through GPIO pins. The AM/FM radio IC is also connected to an external antenna. Both the AM/FM IC and the DAC output signals that feed into the summation amplifier. For summation amplifiers, there are two of them, one for the speaker and one for the headphone jack. The amplifiers combine the signals from the radio and the DAC. Since this is an entirely analog stage, it uses AGND instead of the digital ground. The headphone jack only requires a single output channel, so the InB and OutB pins are left unconnected. The speaker, on the other hand, requires both left and right outputs to operate properly. The remainder of the audio suite beyond the summing amplifiers includes the volume controls for the speaker, allowing for adjustment of the output signal amplitude. The 3.5mm headphone jack is also located there as an audio output. Both the speaker and the headphone jack have their own dedicated amplifiers. A switching MOSFET is used to select between different output signals, such as the left and right channels for the speaker and the headphone jack. This circuit operates from the 3.3V rail and, since it is entirely analog, all components are connected to the analog ground.

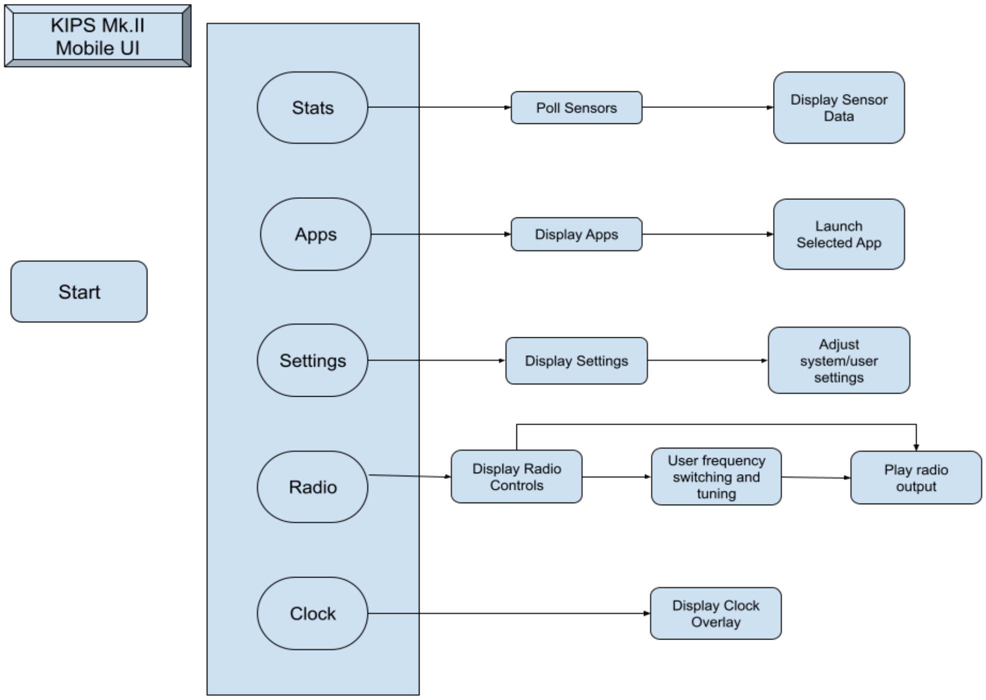
VI. Software Detail

In terms of pure software, it can be split into three major components. The first is the mobile software, which is what runs when the user is wearing the device. This is meant to allow the user to complete important tasks that don't rely on internet access, while consuming the least amount of power possible to preserve battery life. The second component is the desktop software, which can be switched to when the user takes off the device and plugs it into a power source. This will allow for less critical functions to be available to the user, since power is no longer a concern. Additionally, the user will be able to access the internet, and perform operations that would only be feasible on a desktop computer. Desktop mode is ideal for when the user may have a moment to sit down and search the internet for any information that is needed, or run any applications they may have installed that requires more power than what the mobile mode can afford to safely provide. The last component would be the lower level software that runs within the MCU. It controls the sensor and audio suite regardless of what mode the device is in.

A. Mobile Mode

In mobile mode, the user will navigate essential KIPS Mk II functions via a custom GUI created and supported using Glade, a RAD development tool used to make user interfaces for the GTK toolkit and the GNOME desktop [3]. This is not its own operating system, but rather a custom program that launches upon start up. One of the first things the program does is determine if the device is in a charging state or not by checking the status of the fuel gauge IC. If it is not, it will continue to launch into the full mobile program.

The main screen will feature options to navigate to different areas such as settings, applications, stats, health, clock, and radio. There is also additional information displayed on a taskbar at all times, which is the Wi-Fi signal strength, current time, volume, and battery charge. Settings will allow the user to change background image, font size, background colors, overall volume, and screen brightness. Applications come with preinstalled basic essentials such as a calculator, a camera, a flashlight, and an internet browser (The browser is only accessible here if mobile mode is on while connected to Wi-Fi). Other applications can be downloaded as needed, for a level of customizability depending on what the user needs. Stats will allow the user to look at the current brightness, live temperature, and live humidity. Health will allow the user to measure their heartbeat and will plot the change in BPM over time. The clock is slightly different, and will only display the current time. This can be exited with a small exit button, but the purpose of the clock application is to act as a smartwatch, preventing the user from



Mobile UI Flowchart

accidentally tapping anything when not in use while also not shutting down the device. Lastly, the radio app features a digital dial that can be used to swipe to a radio station of the user's choice when the antennas are in use. Additionally, the user can select favorite radio bands to automatically tune into.

B. Desktop Mode

In desktop mode, the user will be met with a more standard desktop interface. This mode is accessed one of two ways, either the user can manually enable this mode even if K.I.P.S is not charging, albeit at the cost of a shorter battery life between charges. Or, it can be entered during its boot up, and the device detected it was already charging. The desktop mode runs Ubuntu with Linux as its operating system. This mode, when active while charging, allows KIPS Mk II to run much more like a portable computer. All non-essential processes and ports are activated under the assumption that power is not an issue since it should be connected to power. This will allow the user to connect peripherals such as an HDMI cable for use in a larger screen, and USB ports for USB drives or a mouse and keyboard. Its computing capabilities as a desktop are limited by what the SBC is capable of handling. In other words, more "heavy" software such as modeling or simulation software, intensive games, or intensive computing/calculations will be more demanding to the SBC. A majority of basic, day-to-day desktop computer functionalities such as internet browsing, light gaming, document reading, file storage, etc however will be well within the bounds of what the SBC is capable of. Additionally, much like in mobile mode, the desktop mode can also download applications as needed for the user's needs, exactly as they would on a standard desktop.

B. Lower Level Software

The lower level software will be a "master program" that has been downloaded to the MCU. This program is structured to initialize many of the libraries, packages, pins, and embedded protocol variables needed to allow the MCU to control the audio suite, sensors, buttons, and joystick. The program itself runs such that the MCU expects commands from the SBC in order to poll for sensor data. It has two UART lanes, one for handling sensor polling, and the second monitoring for input from the buttons and joystick. This is split into two main sections of the MCU code, the first checks for a command from the SBC. This comes in the form of an integer value, and if this value is detected, it will then be sent to a switch case which correlates a command to a function that activates a sensor, the results of which are then sent back to the SBC once it is finished executing. For the buttons and joystick, it will initialize to a point on the screen, and using digital reads to detect voltage changes, it allows the MCU to interpret changes in button status (pressed vs not pressed) and joystick movement to move the target to select different items depending on the current UI layout.

VII. Conclusion

The development of the KIPS Mk. II represents a major overhaul in transforming the initial idea of a wearable wrist-mounted computing device into a robust and versatile real-life device. Expanding upon the baseline laid by the Mk. I prototype, Mk. II integrates a sophisticated combination of mobile and desktop computing modalities including an enhanced sensor suite and a carefully structured hardware architecture built around the ESP32-S3 and Raspberry Pi Compute Module 4.

This project showcases a wide array of technical accomplishments from advanced power-mode switching to comprehensive sensor integration and robust peripheral support, clearly establishing the Mk. II as a significant improvement over the original system. The documentation of standards, design constraints, schematics, software and hardware architecture, manufacturing processes, and testing results highlights the interdisciplinary nature of the work and the careful ethical and collaborative considerations that guided the development process.

The KIPS device is intended to be an alternative and a middle-ground to standard mobile and desktop technologies. In using the device, the user will be able to have access to essential features found in both technologies, all while on the go and being wearable, eliminating the need to carry the device on them in some form. Additionally, the device's extensive customization options to add applications according to user needs increases its versatility and convenience. In designing and constructing the KIPS Mk. II, it is not only meant to be a significant improvement upon its proof of concept (KIPS Mk. I), but also allow its users a level of functionality not commonly found in mobile and desktop devices. The KIPS Mk. II features address the issue of accessibility and common troubles of having to switch between a mobile and desktop device in order to access operations exclusive to those devices. This paper summarizes the technical growth and collaborative effort that enabled the team to achieve this goal.

Acknowledgement

We wish to acknowledge the assistance and guidance of both Dr. Chung Yong Chan and Dr. Arthur Weeks as well as our reviewers: Dr. Sonali Das, Dr. Nazanin Rahnavard, Dr. Justin Phelps, Dr. John Aedo, and Mark Maddox. We, the authors, also wish to thank the University of Central Florida College of Computer and Electrical Engineering for their support.

References

[1] Adminw, "Rechargeable Lithium Batteries vs Alkaline Batteries." Accessed: Sept. 03, 2025. [Online]. Available: https://www.redwaypower.com/rechargeable-lithium-batteries-vsalkaline-comprehensive-guide/

[2] American Radio Relay League, "Test Procedures to Measure BPL Interference."

[3] "Glade - A User Interface Designer" Accessed: Mar. 27, 2026. [Online]. Available: https://glade.gnome.org/

(1) "Antenna Theory Analysis & Design 4th Ed."

(2) P. Koskimaa, "Ferrite Rod Antenna in a Nanosatellite Medium and High Frequency Radio