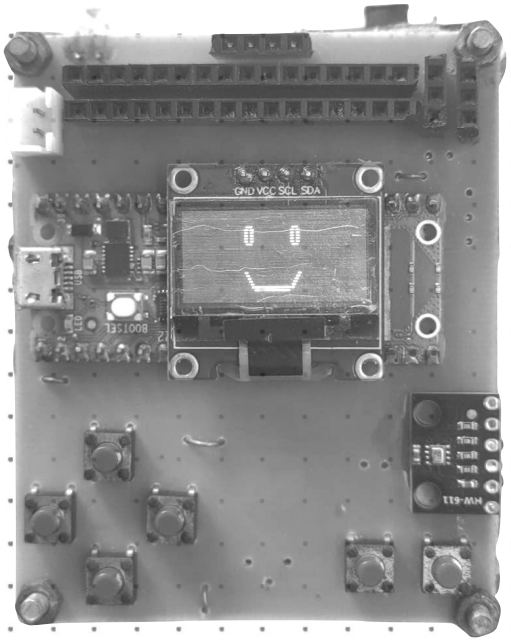
Pixie Developer Board



Developed and Designed y Team X

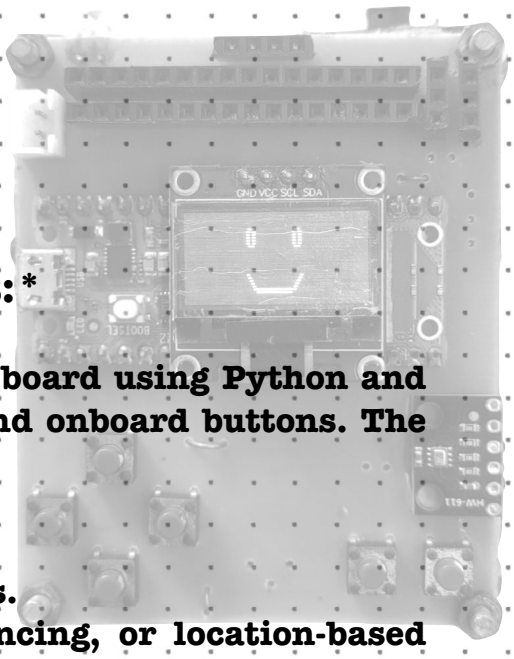
Sakib R. Dalal

Harshit Rawal

Rohan Suryavanshi

Tanmay Janak

Pixie Developer Board



Problem Statement:

Design and implement a versatile development board using Python and Raspberry Pi Pico that integrates GPS, WiFi, and onboard buttons. The board should allow users to:

1. *GPS Integration:*
- Acquire and display real-time GPS coordinates.
- Implement features such as tracking, geofencing, or location-based triggers.

2. *WiFi Connectivity:*
- Establish a wireless connection to networks.
- Enable data transfer and communication over WiFi.

3. *Onboard Buttons:*
- Assign functions to onboard buttons for user interaction.
- Implement customisable actions triggered by button presses.

4. *User Interface:*
- Create a user-friendly interface to display GPS data and board status.
- Provide feedback through LEDs or other indicators.

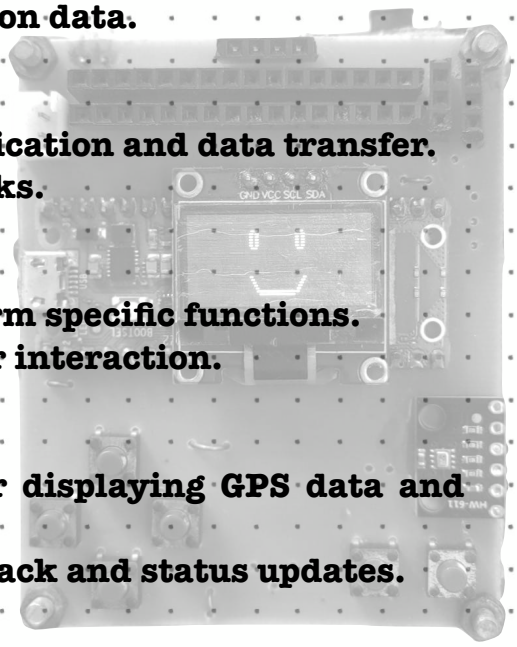
5. *Expandability:*
- Design the board with expansion capabilities for additional sensors or modules.
- Ensure modularity to support future enhancements.

6. *Documentation:*
- Develop comprehensive documentation covering hardware setup, software functionalities, and codebase explanation.

The goal is to create a robust, user-friendly development board that serves as a foundation for various projects involving location-based services, wireless communication, and user interaction.

Objectives

1. *GPS Integration:*
   - Implement a GPS module to accurately acquire and display real-time coordinates.
   - Develop functions for tracking and logging location data.



2. *WiFi Connectivity:*
   - Integrate WiFi capabilities for wireless communication and data transfer.
   - Ensure secure and reliable connection to networks.

3. *Onboard Buttons:*
   - Configure and program onboard buttons to perform specific functions.
   - Implement a responsive button interface for user interaction.

4. *User Interface:*
   - Create a clear and user-friendly interface for displaying GPS data and board status.
   - Utilize LEDs or other indicators to provide feedback and status updates.

5. *Expandability:*
   - Design the board with modular components to support additional sensors or modules.
   - Develop a system for easy integration of future expansions.

6. *Power Management:*
   - Implement efficient power management to optimise battery life or power usage.
   - Include features such as sleep modes when not in active use.

7. *Error Handling and Logging:*
   - Implement robust error handling mechanisms to address unexpected situations.
   - Develop a logging system for recording events and errors for debugging purposes.

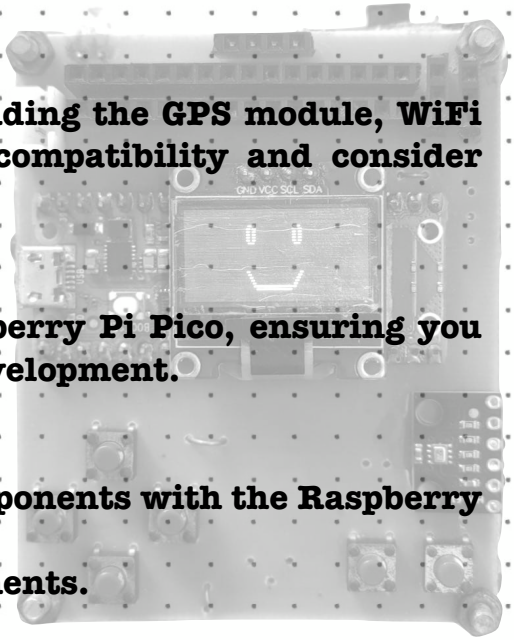
8. *Documentation:*
   - Create comprehensive documentation covering hardware setup, software functionalities, and usage instructions.
   - Include code comments for better understanding and future development.

9. *Testing and Validation:*
   - Conduct thorough testing to ensure the reliability and accuracy of GPS, WiFi, and button functionalities.
   - Validate the board's performance in various scenarios and environments.

10. *Security Considerations:*
    - Implement security measures for WiFi communication.
    - Ensure data integrity and user authentication if applicable

Methodology

1. *Define Requirements:*
   - Clearly define the requirements and functionalities of the development board, considering GPS integration, WiFi connectivity, onboard buttons, and any other specific features you envision.



2. *Select Hardware Components:*
   - Choose appropriate hardware components, including the GPS module, WiFi module, buttons, and Raspberry Pi Pico. Ensure compatibility and consider expandability for future additions.

3. *Setup Development Environment:*
   - Set up the development environment for Raspberry Pi Pico, ensuring you have the necessary tools and libraries for Python development.

4. *Hardware Integration:*
   - Connect and integrate the selected hardware components with the Raspberry Pi Pico.
   - Verify proper communication between the components.

5. *Software Development:*
   - Develop Python scripts to interface with the hardware components.
   - Write functions for GPS data retrieval, WiFi communication, button handling, and any other planned functionalities.

6. *User Interface Design:*
   - Design a user-friendly interface to display GPS data and board status.
   - Implement LED indicators or other visual feedback elements.

7. *Button Functionality:*
   - Program the onboard buttons to trigger specific actions or functions.
   - Ensure responsiveness and reliability of button inputs.

8. *Expandability and Modularity:*
   - Design the board with a modular architecture to support future expansions.
   - Consider compatibility with additional sensors or modules.

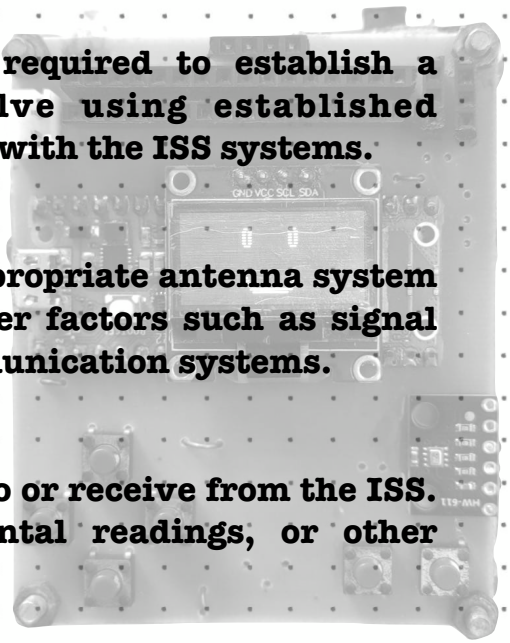
9. *Power Management:*
   - Implement power management features to optimise energy consumption.
   - Test the board's performance in different power modes.

10. *Error Handling and Logging:*
    - Develop robust error-handling mechanisms to address potential issues.
    - Implement a logging system to record events and errors for debugging.

11. *Testing:*
    - Conduct comprehensive testing of the development board in various scenarios.
    - Test GPS accuracy, WiFi connectivity, button responsiveness, and overall

International Space Station

Connecting your development board with the International Space Station (ISS) adds an exciting dimension to your project. Here are some considerations for achieving this:



1. *Communication Protocol:*
- Determine the communication protocol required to establish a connection with the ISS. This could involve using established communication standards or protocols compatible with the ISS systems.

2. *Antenna and Signal Strength:*
- Ensure that your development board has an appropriate antenna system to establish communication with the ISS. Consider factors such as signal strength and compatibility with space-based communication systems.

3. *Data Transmission:*
- Define the type of data you intend to transmit to or receive from the ISS. This could include telemetry data, environmental readings, or other relevant information.

4. *International Space Station API:*
- Investigate if the ISS provides any APIs (Application Programming Interfaces) or data streams that your board can interact with. This could provide real-time data from the ISS or allow you to send specific commands.

5. *Security and Authorisation:*
- Ensure that your communication with the ISS follows security protocols and that you have the necessary authorisation to interact with its systems.

6. *Orbital Dynamics:*
- Understand the orbital dynamics of the ISS to optimise communication windows. The ISS orbits the Earth approximately every 90 minutes, so timing and coordination are crucial.

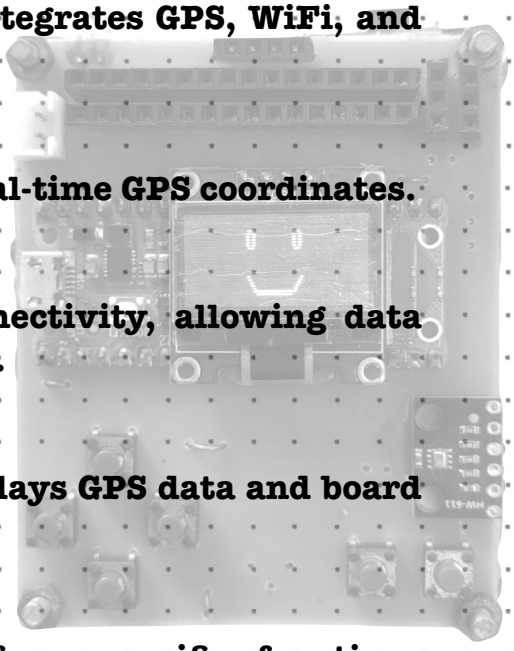
7. *Testing in Simulated Environments:*
- Before attempting communication with the actual ISS, conduct tests in simulated environments to ensure the reliability of your system.

8. *Documentation and Compliance:*
- Document the entire process, including the communication setup, protocols used, and any compliance requirements for interacting with space-based systems.

Connecting to the ISS can provide a unique opportunity to engage with space-based technologies. However, it's essential to adhere to regulations, ensure the compatibility of your system, and consider the challenges associated with space communication. Keep in mind that direct communication with the ISS may require collaboration or authorisation from relevant space agencies or organisations.

Expected Outcomes

The expected outcomes for our versatile development board project include:



1. *Functional Development Board:*
- A fully functional development board that integrates GPS, WiFi, and onboard buttons seamlessly.

2. *Accurate GPS Data:*
- Reliable acquisition and display of accurate real-time GPS coordinates.

3. *Wireless Connectivity:*
- Successful establishment of wireless connectivity, allowing data transfer and communication over WiFi networks.

4. *User-Friendly Interface:*
- A user-friendly interface that effectively displays GPS data and board status, enhancing user interaction.

5. *Responsive Onboard Buttons:*
- Onboard buttons programmed to perform specific functions responsively and reliably.

6. *Expandability and Modularity:*
- A design that allows for easy expansion and integration of additional sensors or modules in the future.

7. *Power-Optimized Operation:*
- Efficient power management features implemented for optimized energy consumption and potential battery-powered usage.

8. *Error Handling and Logging:*
- Robust error-handling mechanisms and a logging system to facilitate debugging and troubleshooting.

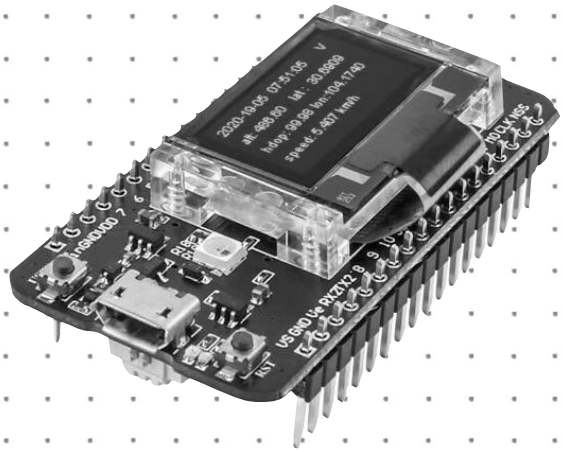
9. *Comprehensive Documentation:*
- Comprehensive documentation covering hardware setup, software functionalities, and user instructions.

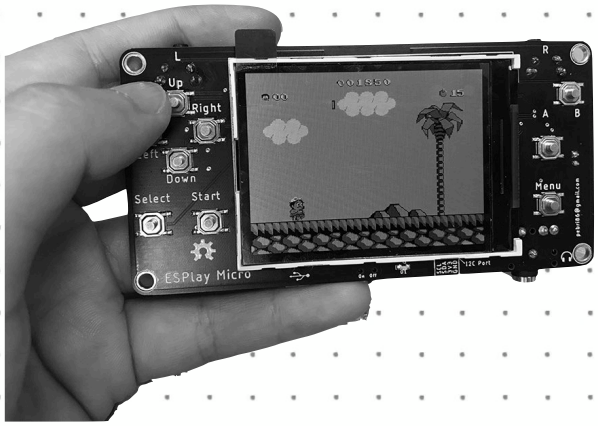
10. *Security Implementation:*
- Implemented security measures for WiFi communication, ensuring data integrity and user privacy if applicable.

11. *Successful Testing Results:*
- Successful testing results showcasing the reliability and accuracy of GPS, WiFi, and button functionalities in various scenarios.

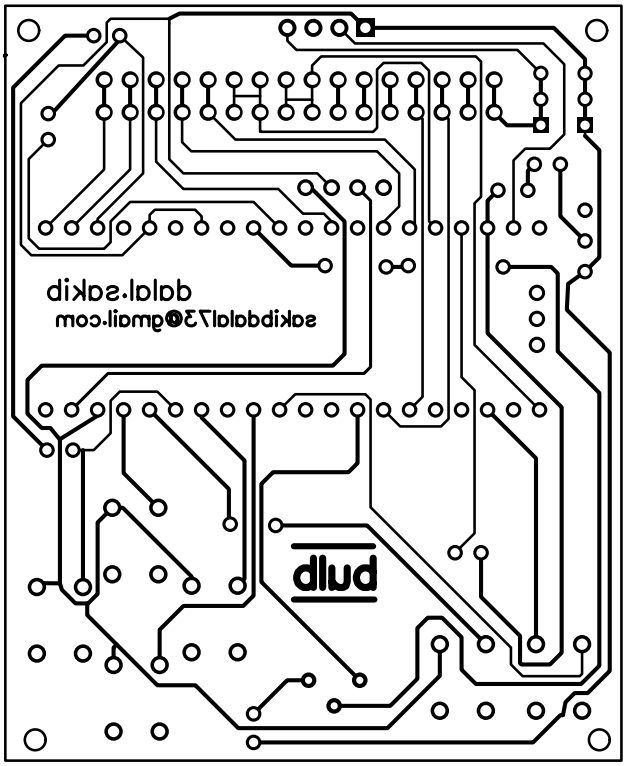
12. *Ready for Deployment:*
- A finalised, polished development board ready for deployment, if applicable.

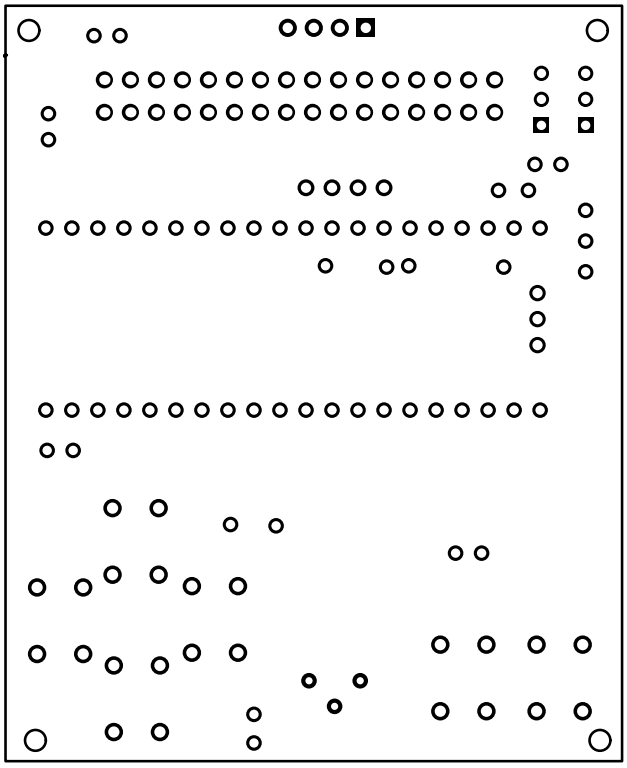
Rival Boards

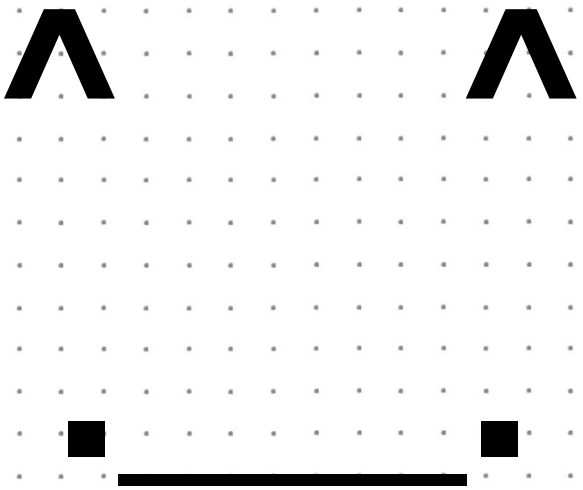




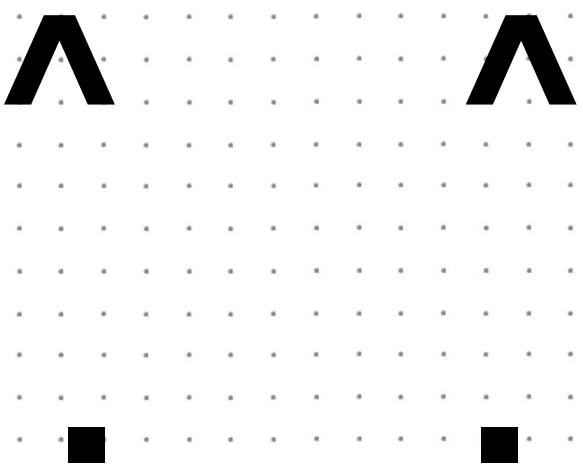
PCB Diagram







Pixie Developer Board



Pixie Developer Board