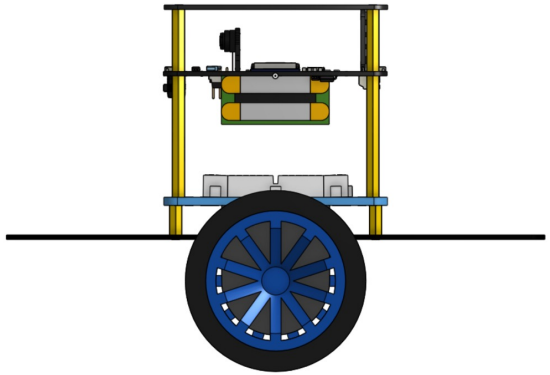
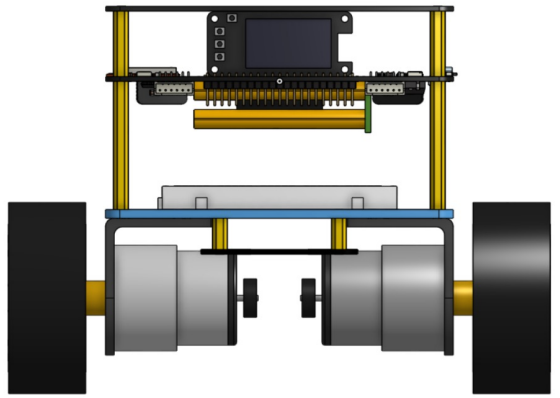
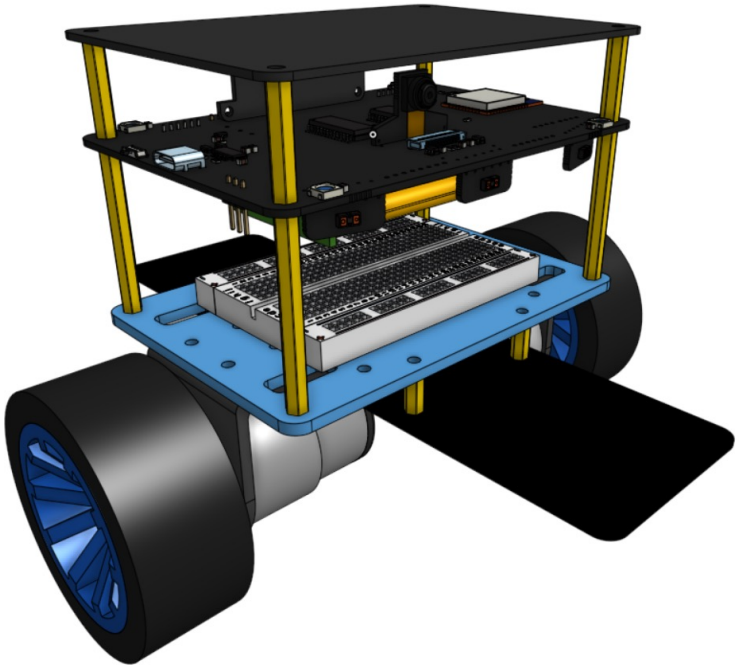
Biped Overview

- Biped (pronounced bai-ped) is a custom two-wheeled self-balancing system designed and built by the CS 431 course staff.
  - Definition of biped: an animal that uses two legs for walking.

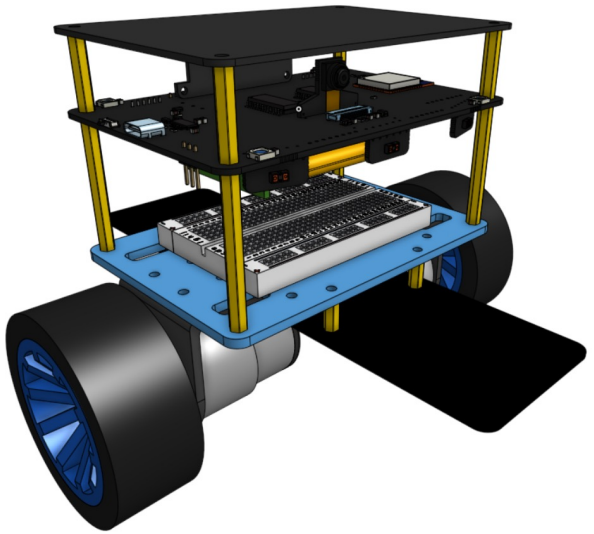


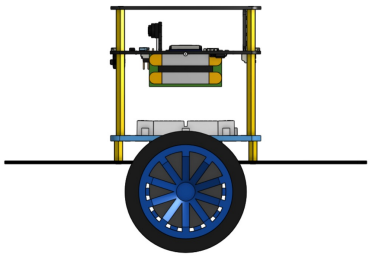


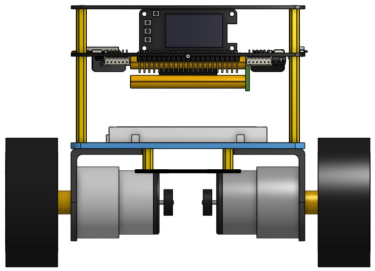


Lecture Learning Objectives

- By the end of this lecture, you will be able to:
  - Have an overall understanding of all the hardware elements onboard Biped.
  - Have an overall understanding of the software framework, toolkits, interfaces, and operating systems you will be using to develop the Biped's firmware during the labs throughout this semester.
  - Have a clear understanding of the learning objectives for all the labs.



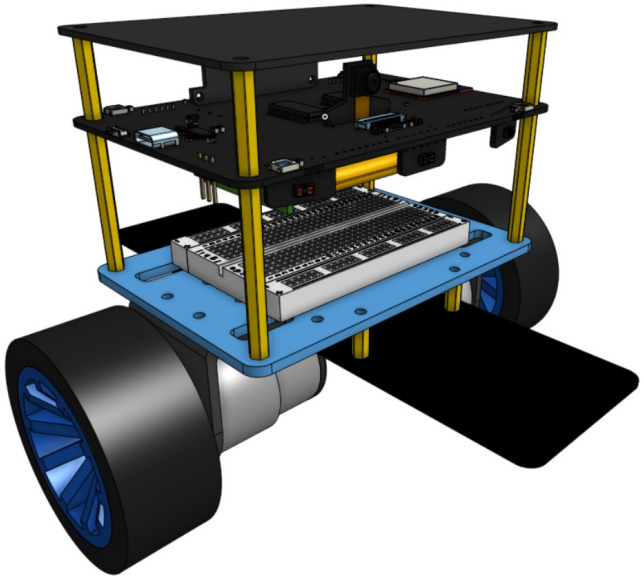




Biped Hardware

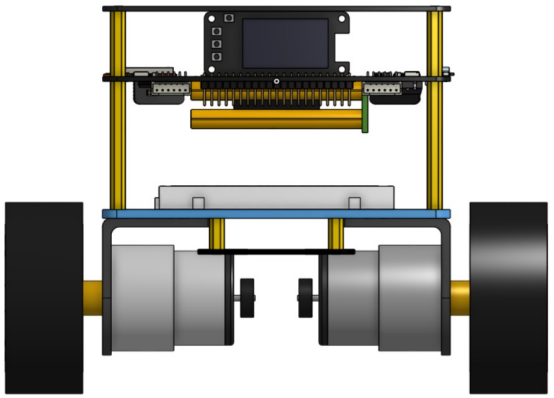
Biped Hardware Overview

- Biped has advanced and comprehensive hardware instrumentations onboard.
- Hardware Sensors
  - Two inertial measurement units (IMU.)
  - Three time-of-flight (ToF) sensors.
  - Two dual-channel incremental rotary encoders.
  - One SVGA 30-bit camera.
- I/Os and User Interfaces
  - One 3-Mbps USB-C serial converter.
  - One 128 x 64 monochrome OLED display.
  - Four NeoPixel RGB LEDs.
  - Two dual-port I/O expanders.
  - One dual-channel H-bridge motor controller.



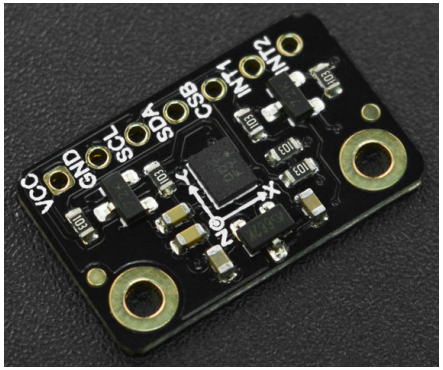
Biped Hardware Overview: Continued

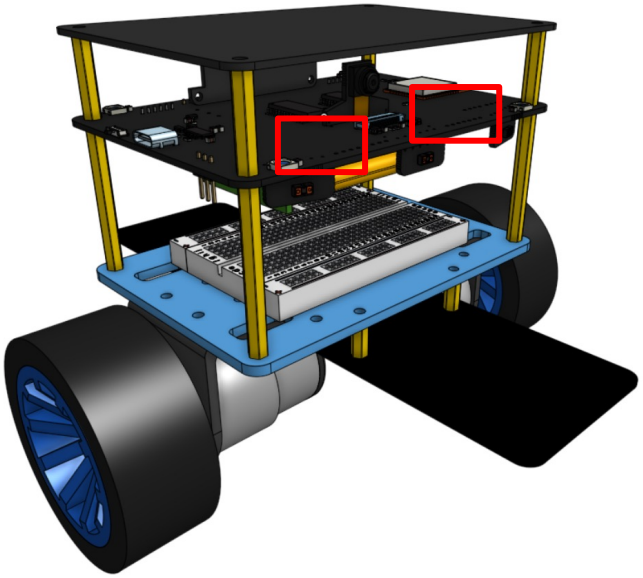
- Microcontroller Unit (MCU)
  - One dual-core ESP32.
- Power Management
  - One 3.3 V instrumentation power regulator.
  - One 6.0 V motor power regulator.
  - One 5.0 V expansion power regulator.
  - One battery management system with charging control and over-voltage and under-voltage protection.
  - One dual-cell 7.4 V lithium-ion battery.
- Actuation
  - Two 12 V brushed DC motors.



Sensor - Inertial Measurement Unit (IMU)

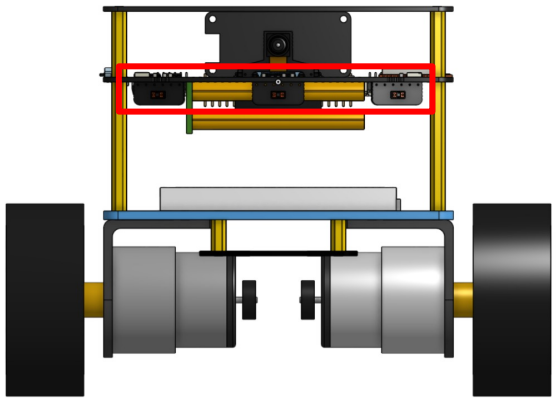
- The inertial measurement unit provides attitude and movement measurements.
- The two IMUs onboard Biped, MPU6050 and BMX160, measure up to 9 degrees of freedom (DoF).
  - 3-DoF accelerometer.
    - Measures linear acceleration, including gravity, and samples up to 1.6 kHz.
  - 3-DoF gyroscope.
    - Measures angular velocity and samples up to 6.4 kHz.
  - 3-DoF compass.
    - Measures geomagnetic strengths and samples up to 12.5 Hz.





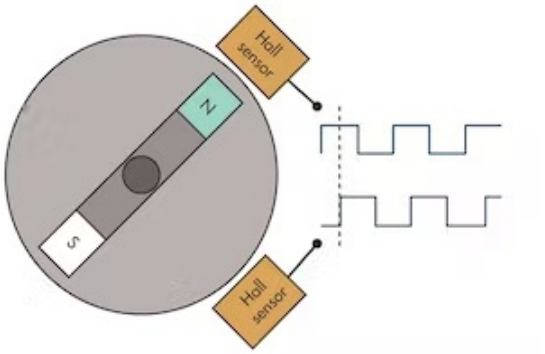
Sensor - Time-Of-Flight (ToF)

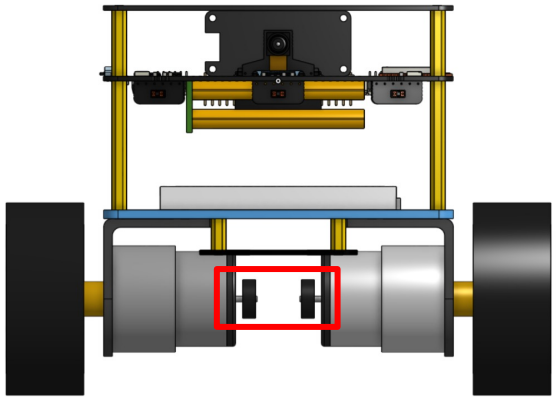
- Time-of-flight (ToF) sensors measure distances using an infrared beam.
  - Equivalent to a one-beam lidar.
- Time-of-flight sensors are useful for distance detection, regulation, and control.
- There are 3 time-of-flight sensors onboard Biped.
  - Measures distances accurately from 10 mm up to 6 meters.
  - 18-degree field of view (FOV).



Sensor - Incremental Rotary Encoders

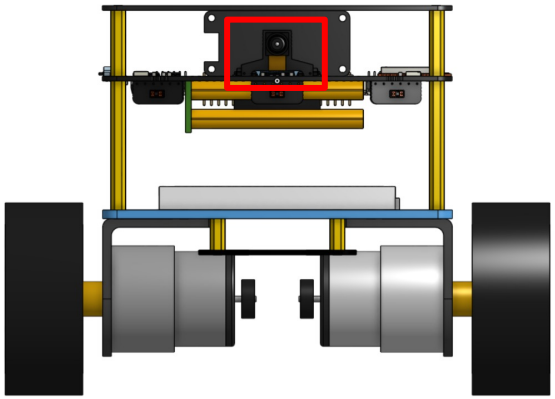
- The incremental rotary encoder provides pulses when its axle is rotated, thus providing a mean for measuring positional displacements.
- The pulses are generated through a magnetic disk rotating near two hall-effect sensors.
- The hall-effect sensors are placed 90 degrees apart, generating phase-shifted pulses, with which the direction of rotation of the magnetic disk can be determined.





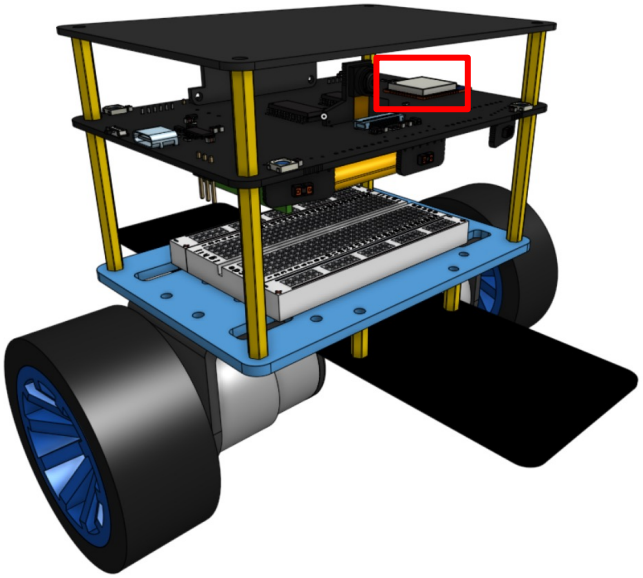
Sensor - Camera

- There is a forward-facing camera onboard Biped, which is useful for analyzing the surrounding environment either onboard or remotely in the cloud.
- The sensor supports up to 800x600 SVGA resolution with up to 30-bit color space.
- For efficiency, the camera images are streamed to the MCU data ram asynchronously using a direct memory access (DMA) controller.



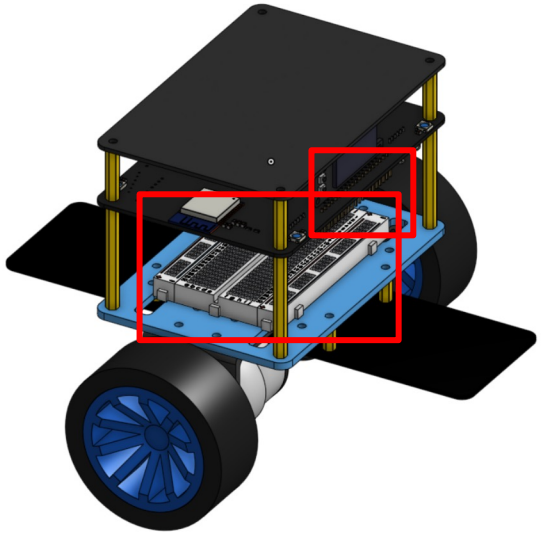
Microcontroller Unit (MCU)

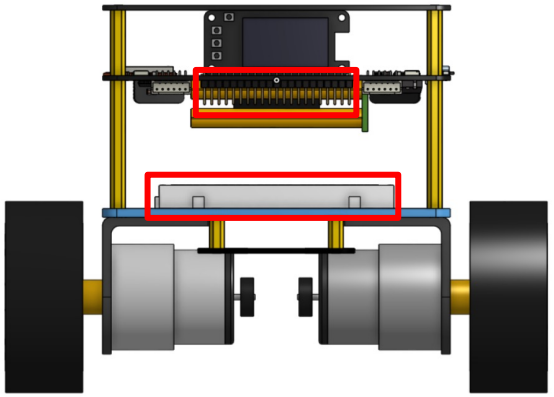
- The microcontroller unit (MCU) serves as a system-on-chip (SoC) for compact embedded systems, in our case, Biped.
- There is one MCU onboard Biped, ESP32.
- Dual 32-bit LX6 Tensilica CPU cores with up to 240 MHz clock speed.
- 520 KB RAM and 448 KB flash memory, with support for external flash and SPI RAM.
- 34 versatile I/O pins all support interrupts and various protocols, including I2C, I2S, SPI, JTAG, UART, CAN bus, etc.
- Built-in Wi-Fi (802.11b/g/n) and Bluetooth LE 4.2 wireless communication.
- Less than $4 for each of the SoCs!



User Interface - I/O-Expander

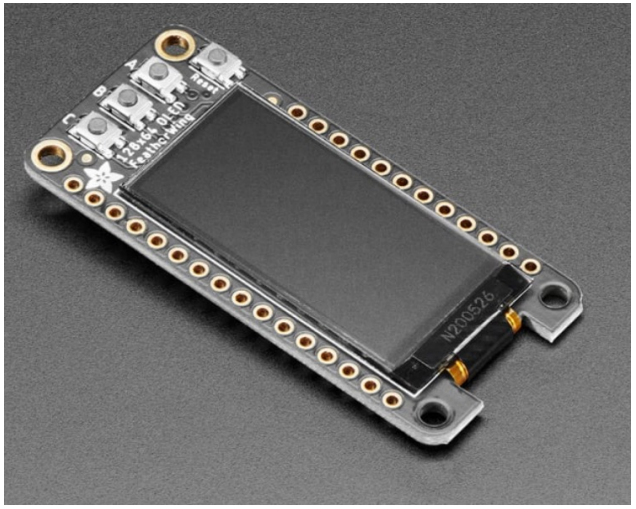
- An I/O expander provides pin expansion for the MCU, allowing more devices to be connected.
- There are two I/O expanders onboard Biped.
  - Dual-port 16-pin communicated through the I2C bus.
  - The first I/O expander is used to host all instrumentations onboard Biped.
  - The second I/O expander is connected to an expansion header, allowing for any external devices and modules to be connected.
  - The expansion header can be connected to the breadboard on the Biped's lower chassis for the 4-credit projects.

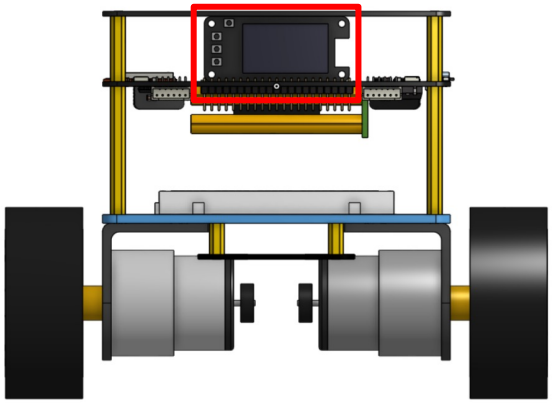




User Interface - OLED Display

- The display is useful for show the operating status of the system, as well as for rapid debugging.
- There is one OLED display onboard Biped.
  - Monochrome.
  - 128 x 64 resolution.
  - Refresh rate up to 30 FPS.
  - 3 push buttons and 1 reset button for user input and control.

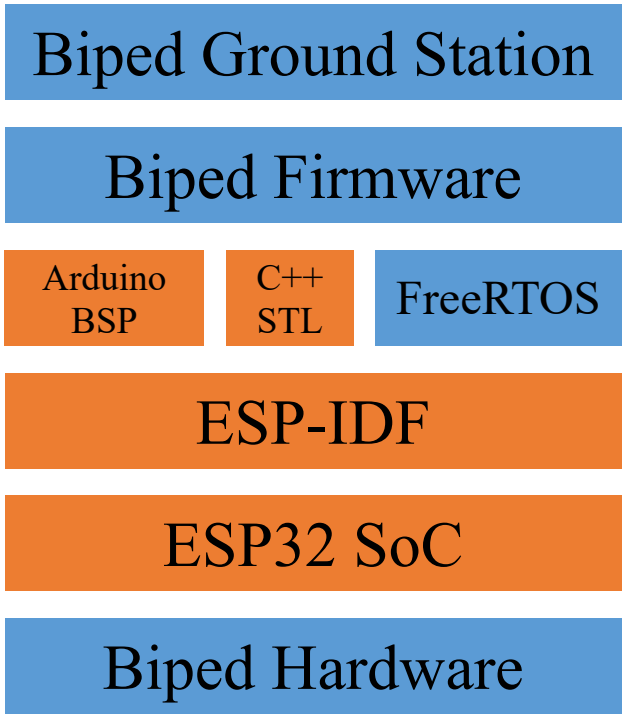




Biped Software

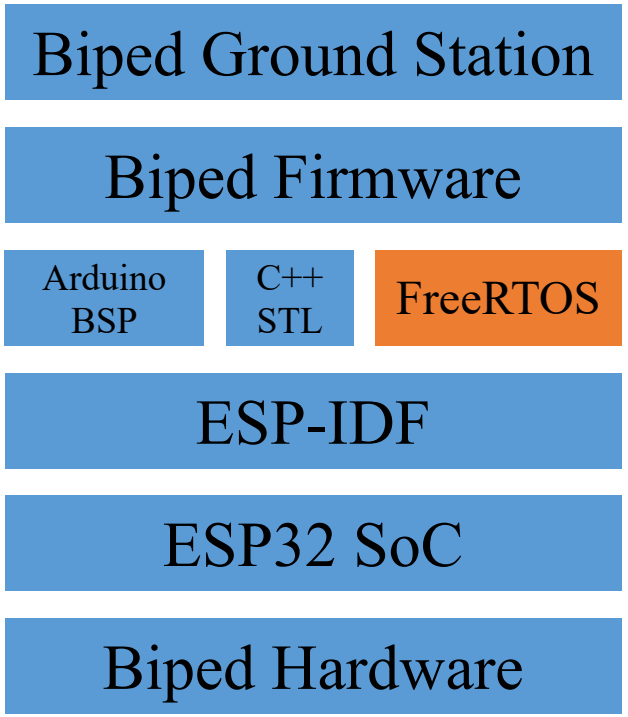
Biped Software Overview

- The Biped software stack employs the ESP-IDF (ESP IoT development framework), which runs bare-metal directly on the ESP32 SoC.
- In this course, we will be using the Arduino board support package (BSP) for ESP32, which wraps around the ESP-IDF.
  - Common Arduino APIs are available and are reimplemented using the ESP-IDF APIs.
  - The software stack also employs the Xtensa C/C++ development toolchain, which means the entire C++ STL library is supported and reimplemented using ESP-IDF APIs.



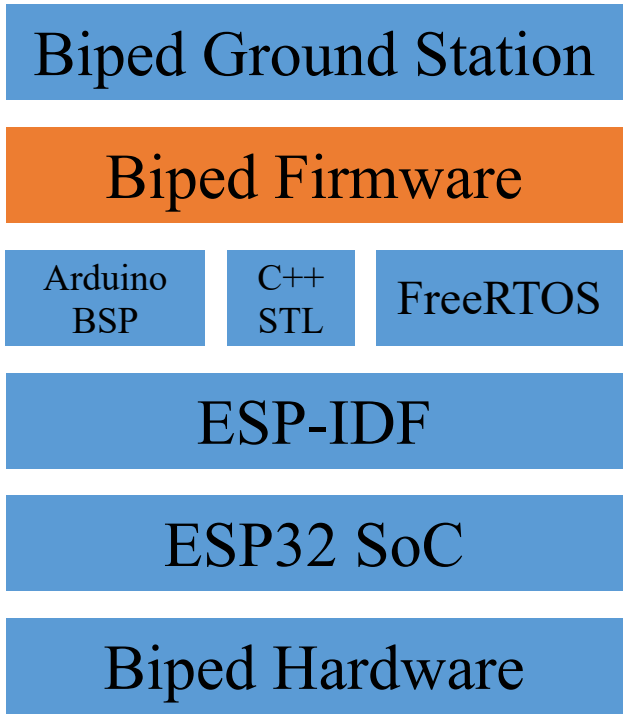
FreeRTOS

- The FreeRTOS is employed as the RTOS running on top of the ESP-IDF.
- FreeRTOS is a popular open-source real-time operating system (RTOS) available on almost every major embedded system and FPGA platform.
  - Employs a fixed-priority preemptive real-time scheduler, with round-robin time slices for equal-priority tasks.
  - Provides common synchronization primitives such as binary semaphores, counting semaphores, mutexes with priority inheritance, etc.
  - Highly configurable with a very low memory footprint.



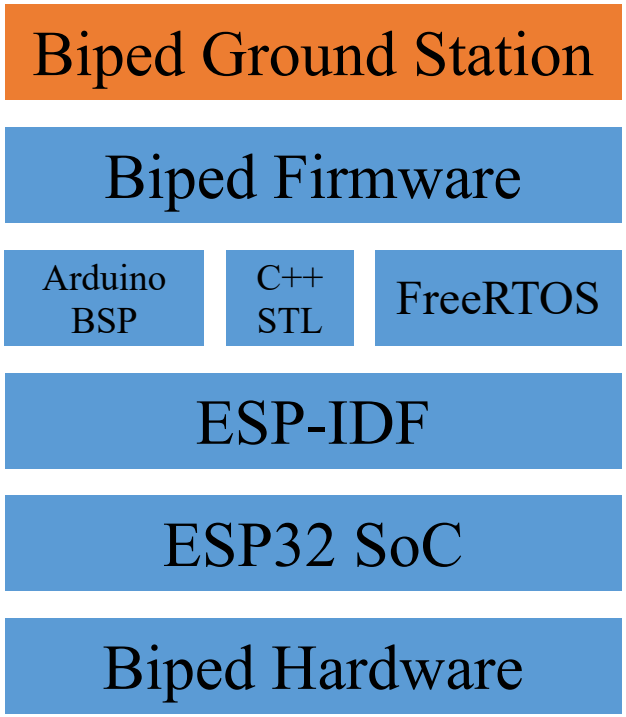
Biped Firmware

- During the labs, you will be using the Eclipse integrated development environment (IDE), common for embedded software enginnering, to develop your version of the Biped firmware and complete the labs.
- The developed Biped firmware is then built and flashed to the Biped hardware using the CMake build system.

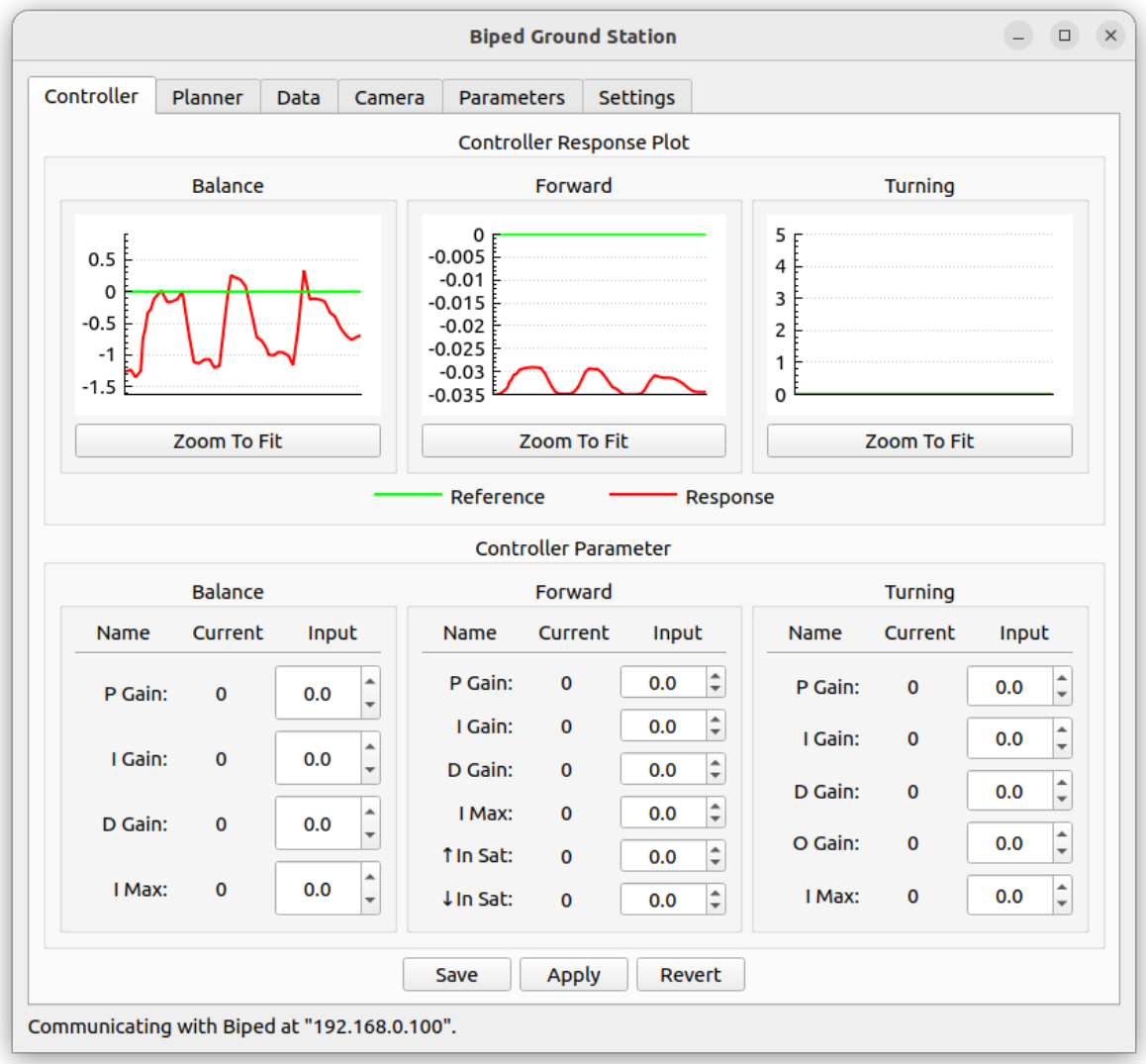


Biped Ground Station

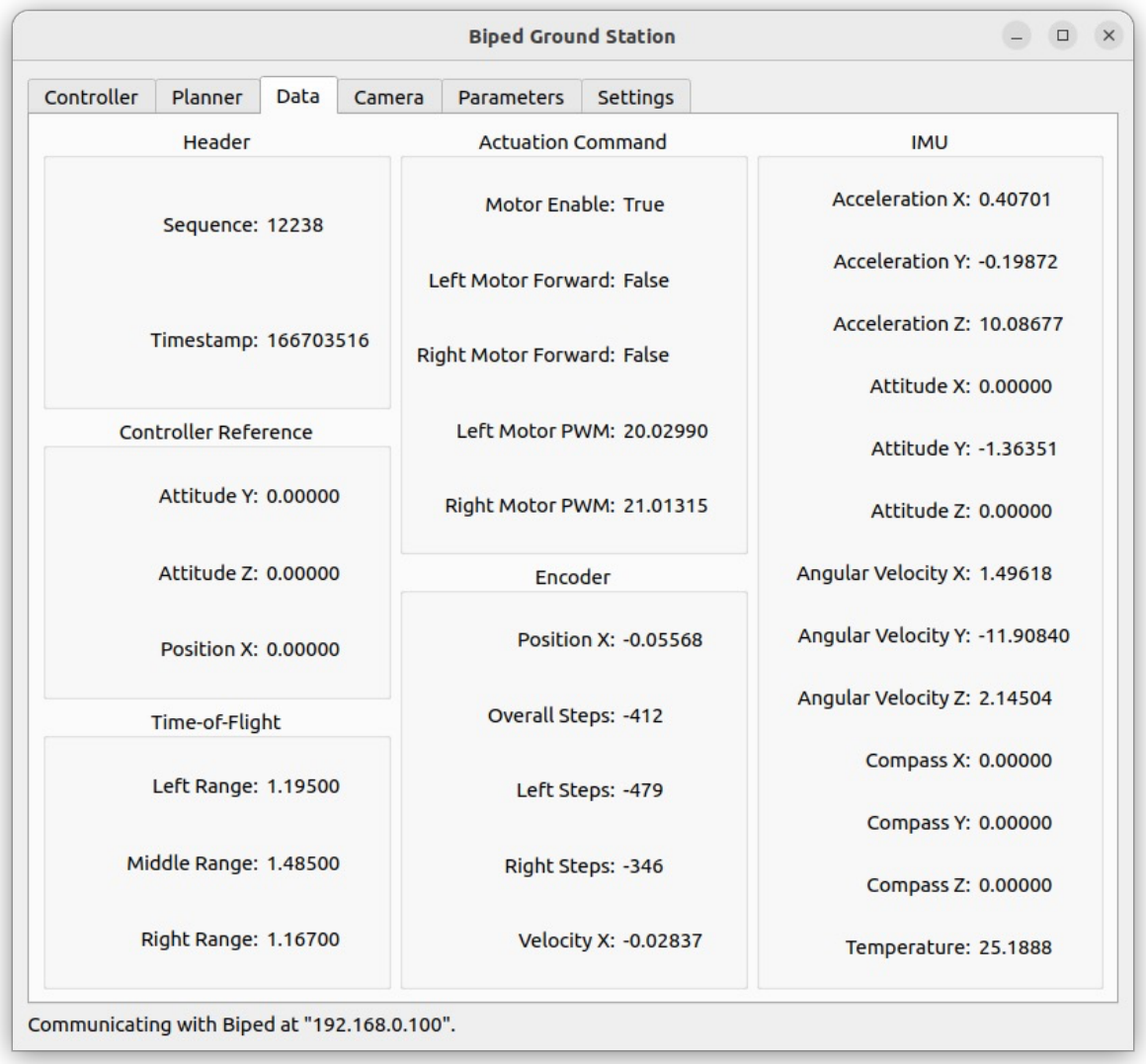
- The Biped Ground Station (BGS) is a graphical user interface developed by the CS 431 course staff to facilitate the development of the Biped firmware.
- BGS communicates with the Biped hardware wirelessly via Wi-Fi.
- BGS receives and displays live telemetry data from the Biped hardware.
- BGS also allows users to send data and commands to the Biped hardware.
- BGS enables users to tune the controller parameters efficiently in real-time.
- BGS can be used by the 4-credit students as a wireless API to Biped to transceive data, such as receiving camera images and sending control commands.



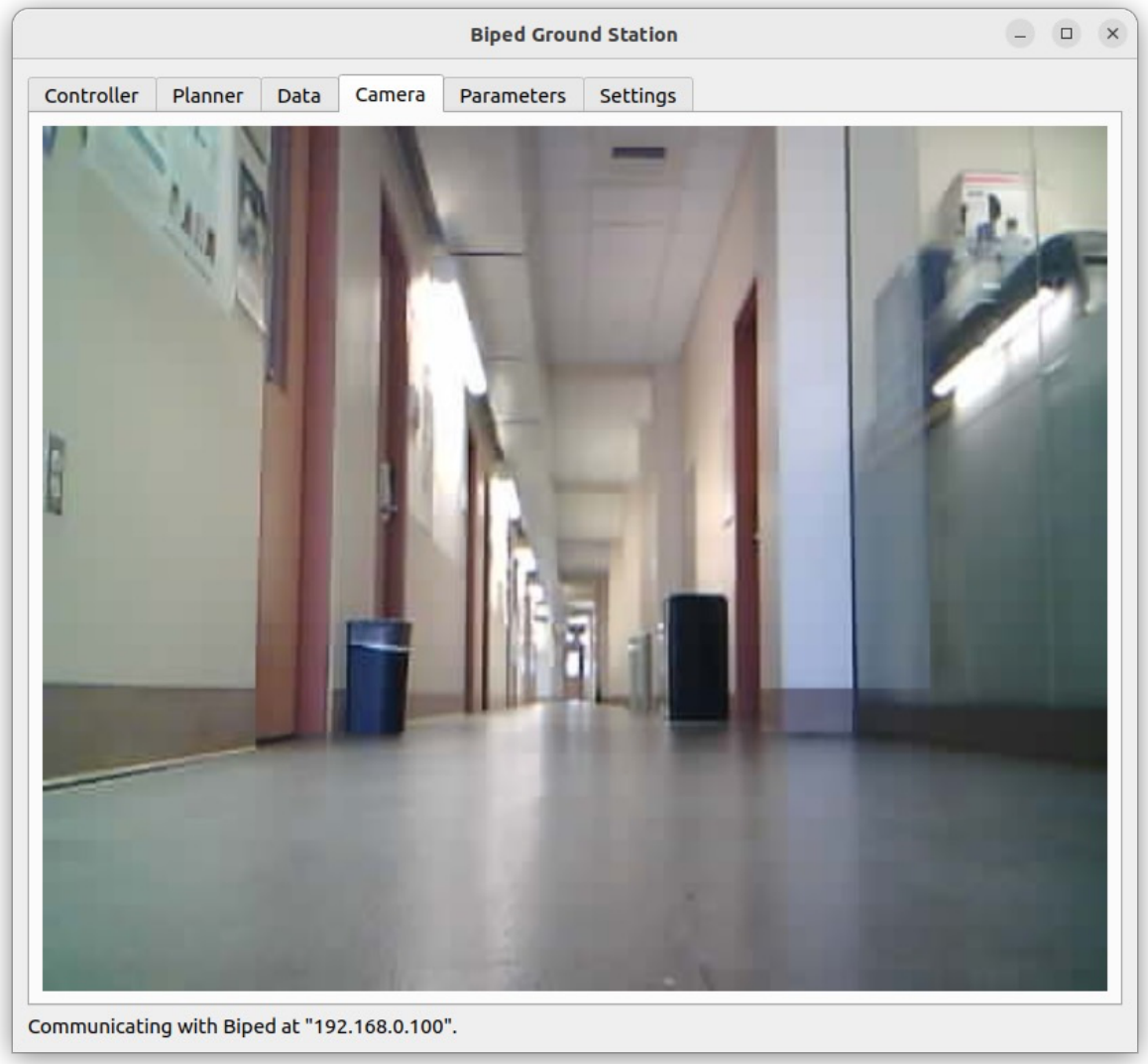
Biped Ground Station - Controller Tuning



Biped Ground Station - Data Visualization



Biped Ground Station - Camera Viewer

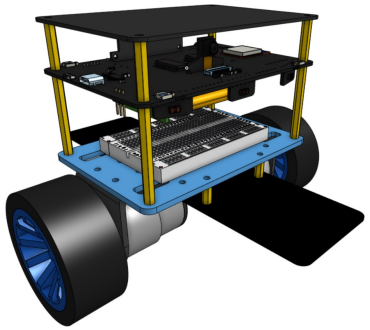


Demo

Labs

Lab Learning Objectives

- Lab 1: Setup and I/O.
  - Set up Biped firmware development environment.
  - Perform basic I/O operations via the OLED display and serial connection.
- Lab 2: Hardware Timer.
  - Initiate hardware interrupts using hardware timers onboard the ESP32 SoC.
- Lab 3: FreeRTOS.
  - Create real-time tasks using FreeRTOS for various software components in the Biped firmware.
- Lab 4: I/O Expander.
  - Set up the I/O expanders and develop an interrupt service framework allowing users to attach interrupt handers to the expanders.



Lab Learning Objectives: Continued

- Lab 5: Biped Ground Station.
  - Set up communications and data transmissions between the Biped and the Biped Ground Station.
- Lab 6: Sensing and Actuation.
  - Set up the sensors and motors, collect data from the sensor hardware, and actuate the motors.
- Lab 7: Control.
  - Implement a PID controller framework and tune the controllers to balance the Biped.
- Lab 8: Planning.
  - Develop two types of planners that produce control commands and move the Biped around.
- Lab 9: Application.
  - Apply everything you have learned and built so far and develop some creative plans for the Biped.