Raspberry Pi Robotics Extension Board

1st Christian Kargl
HTL St.Pölten
St. Pölten, Austria
christian.kargl21@gmail.com

2nd Peter Strommer
HTL St.Pölten
St. Pölten, Austria
p.strommer@outlook.at

3rd Christoph Hackl
HTL St.Pölten
St. Pölten, Austria
christoph.hackl@htlstp.at

4th Moritz Klein
HTL St.Pölten
St. Pölten, Austria
moritzklein0810@gmail.com

5th Manuel Bichl
HTL St.Pölten
St. Pölten, Austria
manuel.bichl@htlstp.at

***Abstract*—This paper presents the development of a custom Raspberry Pi extension board for use in the European Conference on Educational Robotics (ECER). Instead of relying on separate add-on modules, the design combines motor control, servo control, and sensor interfacing on a single printed circuit board (PCB). The resulting hardware increases the number of available motor and servo outputs while reducing integration effort. Flexibility and expandability were key requirements throughout the design process, ensuring that the platform remains useful beyond the competition and can serve as a basis for future robotics projects.**

***Index Terms*—Robotics Controller, Raspberry Pi, Motor Driver, Servo Control, Educational Robotics**

I. Introduction

Robotics competitions such as ECER demand hardware that is both reliable and adaptable under uncertain conditions. Single-board computers like the Raspberry Pi are attractive for such systems because they combine processing power, community support, and software flexibility [1]. However, competition robots also require actuator control, analog sensing, and robust power distribution, which are not provided directly by the Raspberry Pi itself.

This paper presents a custom Raspberry Pi extension board developed for these requirements. The design integrates motor interfaces, servo control, sensor connectivity, inertial sensing, and power distribution into a single PCB, with the goal of reducing wiring complexity and improving integration compared with a modular controller setup.

The hardware platform has also been upgraded significantly. Where the previous system was based on a Raspberry Pi 3, the new design uses a Raspberry Pi 5 with 16 GB of RAM. An AI accelerator [3] capable of up to 26 tera operations per second (TOPS) complements the platform, enabling more neural-network-based approaches.

II. Related Work

A. Overview of Existing Controller Approaches

In previous years, the Wombat controller was used as the central robotics platform in the ECER competition. The Wombat is an established educational robot controller with its own software environment and hardware ecosystem [7]. For the requirements of our previous competition setup, however, its built-in interfaces were no longer sufficient on their own.

To overcome these limitations, supplementary boards were added to the Wombat in the previous project, including a servo shield and a GPIO expander [2]. This approach was functional, but it also increased wiring complexity, reduced mechanical robustness, and distributed related subsystems across several modules.

The present work follows a different approach by integrating the required robotics interfaces directly onto a dedicated Raspberry Pi extension board. Instead of extending the existing controller step by step, the new design consolidates everything on one hardware platform.

III. Concept

A. Design Goals

Because the exact competition tasks are not disclosed in advance, the design was intended to support a wide range of robot configurations. The main goals were to reduce wiring complexity, integrate the most important robotics interfaces on a single board, and separate high-level software execution from time-critical control tasks.

More specifically, the platform was intended to provide the following capabilities:

- up to six motor outputs for robot locomotion
- eight servo outputs controlled through a dedicated Inter-Integrated Circuit (I^2C) servo controller [9]
- five digital sensor interfaces and analog acquisition through three single-ended and two differential analog channels
- onboard inertial measurement units
- separate power paths for logic electronics and high-current actuators

B. Embedded Robotics Constraints

While the Raspberry Pi provides a flexible software platform, it does not provide native analog inputs, cannot drive motors directly, and is not intended for deterministic real-time actuator control. These limitations motivate the use of a dedicated extension board and an additional microcontroller for time-critical tasks.

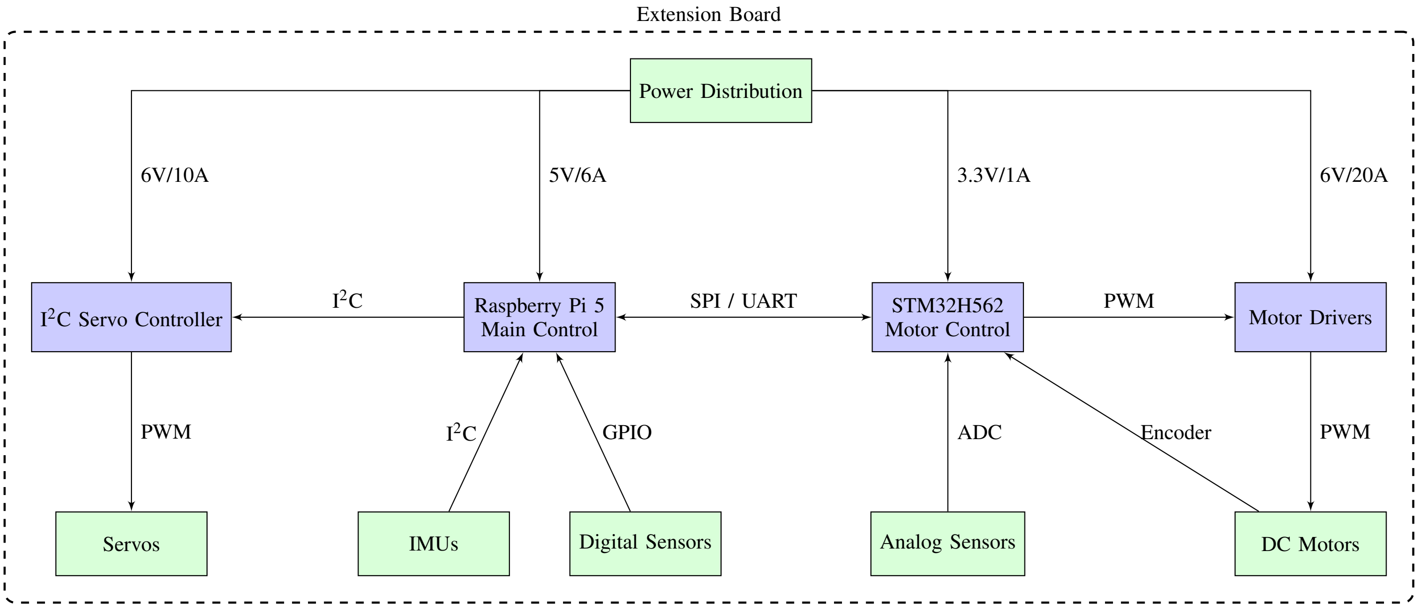

Fig. 1. Functional block diagram of the Raspberry Pi robotics extension board system.

IV. Implementation

The Raspberry Pi remains the central processing unit of the robot, executing the main control software and handling high-level decision making. The extension board bridges this controller to the physical actuators and sensors.

Figure 1 illustrates the functional structure of the system. The Raspberry Pi acts as the central controller and communicates directly with the digital sensors, the inertial measurement units (IMUs), and the servo controller via I^2C. A dedicated microcontroller from STMicroelectronics (STM) is used only for time-critical motor control and analog sensor acquisition.

A. Motor Driver Interfaces

One of the primary functions of the board is the control of DC motors used for robot locomotion. Since direct motor actuation requires dedicated driver circuitry, motor driver stages are implemented on the extension board. In the current design, the motor stage is based on three DRV8411 motor drivers [10] and is designed for up to six motor outputs.

These drivers provide the required actuation for the motors, while their control signals are generated by the STM microcontroller. Encoder feedback is available for all driven motors, enabling closed-loop motor control through the same microcontroller.

B. Servo Interfaces

Servos are one of the most important components in robotics. The extension board therefore includes dedicated servo connectors that allow several servo motors to be connected directly to the board. Servo control is implemented through an I^2C-based servo controller addressed by the Raspberry Pi. In the current design, this function is realized using the PCA9685 PWM controller [9], of which eight channels are currently assigned to servo outputs.

C. Sensor Interfaces

Robotic systems require sensors to detect environmental conditions. The extension board supports five digital sensor interfaces through GPIO connections as well as analog acquisition through three single-ended and two differential analog channels. The digital sensors are connected directly to the Raspberry Pi, while the analog sensors are connected via the STM microcontroller. The single-ended and differential analog channels are implemented as separate acquisition paths.

D. Inertial Measurement Units

In addition to the external sensor interfaces, the board also integrates onboard inertial measurement units. These IMUs provide motion and orientation data directly on the controller board and are connected to the Raspberry Pi via an I^2C interface for tasks such as stabilization and motion tracking.

The current design includes a CEVA BNO085 [4] and an ST LSM6DSO [5]. Both devices are planned to operate via I^2C. By integrating these components directly on the PCB, the system gains immediate access to acceleration, angular rate, and orientation-related data without requiring additional external modules. Two different IMUs are used for redundancy.

E. Power Distribution

Motors and servos place significantly higher power demands on the system than the logic electronics. For this reason, the design includes a generously dimensioned power distribution stage for both logic electronics and high-power actuators.

The board is supplied from a 12 V input, corresponding to a 3S1P battery configuration. From this input, four buck converters generate the required lower supply voltages for the different subsystems.

Three of these converters provide 6 V at up to 10 A. One 6 V rail is used for the DC motors, while two separate 6 V rails are dedicated to the servos. The servo power stages were intentionally dimensioned with a high current reserve in order to handle the high holding currents that can occur during operation.

Another buck converter provides 5 V at up to 6 A for the Raspberry Pi. From this 5 V rail, an additional conversion stage generates 3.3 V for peripheral electronics such as the STM microcontroller, the IMUs, and other low-voltage components.

Separate power paths are used for logic electronics and high-power actuators in order to reduce electrical disturbances and improve overall system stability.

V. Planned Motor Control Microcontroller

To handle time-critical tasks, the controller board includes an additional microcontroller, the STM32H562RIT6 [6]. This microcontroller is intended to handle all real-time motor control operations once firmware integration is completed.

Motor control in robotics often requires deterministic timing and fast feedback processing. Because the Raspberry Pi runs a full operating system, it is not ideal for precise real-time control. The STM microcontroller is therefore intended to execute the complete motor control logic.

The planned firmware for the microcontroller includes Proportional-Integral-Derivative (PID) control algorithms to regulate the motor speed and movement of the robot. In the intended architecture, the controller continuously reads feedback signals from the motor encoders and calculates the control output required to achieve the desired motor speed.

Another important planned function of the microcontroller is handling analog sensors. Since the Raspberry Pi does not provide native analog inputs, analog sensor signals are connected directly to the microcontroller's Analog-to-Digital Converter (ADC) inputs. Some of these channels can also be used for differential measurements. In the intended system architecture, the measured values are then transferred to the Raspberry Pi.

Communication between the Raspberry Pi and the STM microcontroller is planned to use a Serial Peripheral Interface (SPI). Through this interface the Raspberry Pi is intended to send commands such as motor speed targets and receive sensor data from the microcontroller.

For development and debugging purposes, the microcontroller firmware was initially intended to be uploaded directly from the Raspberry Pi using a Universal Asynchronous Receiver-Transmitter (UART) connection. The available bootloader interfaces and their activation conditions are device-dependent and documented by STMicroelectronics in application note AN2606 [8], while the USART bootloader protocol itself is described in AN3155 [11]. In the current prototype, difficulties with this flashing path led to a transition towards Serial Wire Debug (SWD) programming for more reliable firmware upload and debugging.

VI. Results

At the current stage of the project, the controller board hardware has already been designed and assembled. The main components, including the Raspberry Pi interface, motor drivers, microcontroller, and sensor interfaces, are implemented on the board.

Figure 2 shows the current PCB layout of the prototype board in its present development stage.

Initial bring-up confirms correct power-up, the presence of the required 5 V, 6 V, and 3.3 V supply rails, and successful Raspberry Pi operation. These results already validate the basic electrical design and the general power-distribution concept.

On the Raspberry Pi side, several subsystems are already functional. The I^2C-based servo controller for all eight assigned servo channels operates as intended, all five digital sensor interfaces have been tested successfully, and one of the two integrated IMUs, the LSM6DSO, can already be detected and read out. This shows that the chosen architecture is viable in practice.

The following table summarizes the current validation status of the most important subsystems.

Table I

Current subsystem status of the prototype board.

Subsystem	Status	Observation
Power rails	Verified	5 V, 6 V, and 3.3 V rails present
Raspberry Pi	Verified	System boots successfully on the board
Servo control	Verified	All 8 PCA9685-based servo channels tested
Digital sensors	Verified	All 5 GPIO-based sensor interfaces tested
LSM6DSO	Verified	Device detected and sensor values read
BNO085	Open	Device not yet detected via I^2C
STM flashing	Open	UART flashing not yet reliable
Motor control	Open	Depends on STM32 programming and communication

The main limitation of the current prototype is the incomplete STM integration. At the moment, the STM microcontroller cannot yet be programmed reliably through the intended UART flashing path, which also prevents validation of the SPI communication between the Raspberry Pi and the microcontroller. As a consequence, the three single-ended analog channels, the two differential analog channels, PWM-based motor control, and encoder processing for all motor channels could not yet be evaluated on the assembled board.

Since the STM32 bootloader behavior and supported interfaces are device-specific [8], the debugging workflow is currently being migrated to SWD programming.

Further open issues include the missing detection of the BNO085 IMU and the absence of load tests for the power stage and actuator interfaces. These aspects currently limit the evaluation to basic bring-up and subsystem checks, but they also define the next validation steps.

VII. Limitations and Future Work

The current paper focuses on the system architecture and hardware implementation of the first prototype, but it does not yet provide a full experimental evaluation. Future work will therefore include systematic validation of the motor control performance, communication reliability, IMU integration, and power stability under realistic operating conditions.

In addition, the firmware and software stack will be extended to make better use of the integrated interfaces during real competition tasks. Once STM programming and communication are stable, quantitative measurements will be added to compare the new controller against the previously used modular Wombat-based setup.

VIII. Conclusion

This paper presented the development of a Raspberry Pi extension board for robotics applications in the ECER competition. The proposed design integrates motor drivers, servo interfaces, sensor connections, inertial sensing, and power distribution into a single hardware platform.

Compared with the previously used modular Wombat-based setup, the new design reduces wiring complexity and provides a clearer separation between high-level processing on the Raspberry Pi and time-critical tasks on the STM microcontroller. Initial bring-up results already confirm the validity of the electrical design and several Raspberry Pi-side subsystems.

Although the controller is still under development, the current prototype provides a solid basis for further validation, firmware integration, and future robotics projects.

References

[1] Raspberry Pi Ltd., “Raspberry Pi 5 Product Brief,” 2023. [Online]. Available: https://pip-assets.raspberrypi.com/categories/892-raspberry-pi-5/documents/RP-008348-DS-6-raspberry-pi-5-product-brief.pdf

[2] C. Kargl, M. Bichl, P. Strommer, and C. Hackl, “Extension Boards for the Wombat,” HTL St. Pölten Robotics Project Report, 2025.

[3] Raspberry Pi Ltd., “Raspberry Pi AI HAT Product Brief,” 2024. [Online]. Available: https://pip-assets.raspberrypi.com/categories/1081-raspberry-pi-ai-hat/documents/RP-008133-DS-1-raspberry-pi-ai-hat-plus-product-brief.pdf

[4] CEVA, “BNO085 Smart Sensor Hub Datasheet,” 2023. [Online]. Available: https://www.ceva-ip.com/wp-content/uploads/BNO080_085-Datasheet.pdf

[5] STMicroelectronics, “LSM6DSO IMU Datasheet,” 2023. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6dso.pdf

[6] STMicroelectronics, “STM32H562xx Datasheet,” 2024. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32h562ri.pdf

[7] KIPR, “KIPR Hardware/Software: KIPR Wombat,” [Online]. Available: https://www.kipr.org/kipr/hardware-software

[8] STMicroelectronics, “AN2606: STM32 Microcontroller System Memory Boot Mode,” [Online]. Available: https://www.st.com/resource/en/application_note/an2606-stm32-microcontroller-system-memory-boot-mode-stmicroelectronics.pdf

[9] NXP Semiconductors, “PCA9685: 16-Channel, 12-Bit PWM Fm+ I^2C-Bus LED Controller,” [Online]. Available: https://www.nxp.com/docs/en/data-sheet/PCA9685.pdf

[10] Texas Instruments, “DRV8411 11-V to 50-V, 7-A Peak H-Bridge Motor Driver,” [Online]. Available: https://www.ti.com/lit/ds/symlink/drv8411.pdf?ts=1775644197177

[11] STMicroelectronics, “AN3155: USART Protocol Used in the STM32 Bootloader,” [Online]. Available: https://www.st.com/resource/en/application_note/an3155-usart-protocol-used-in-the-stm32-bootloader-stmicroelectronics.pdf

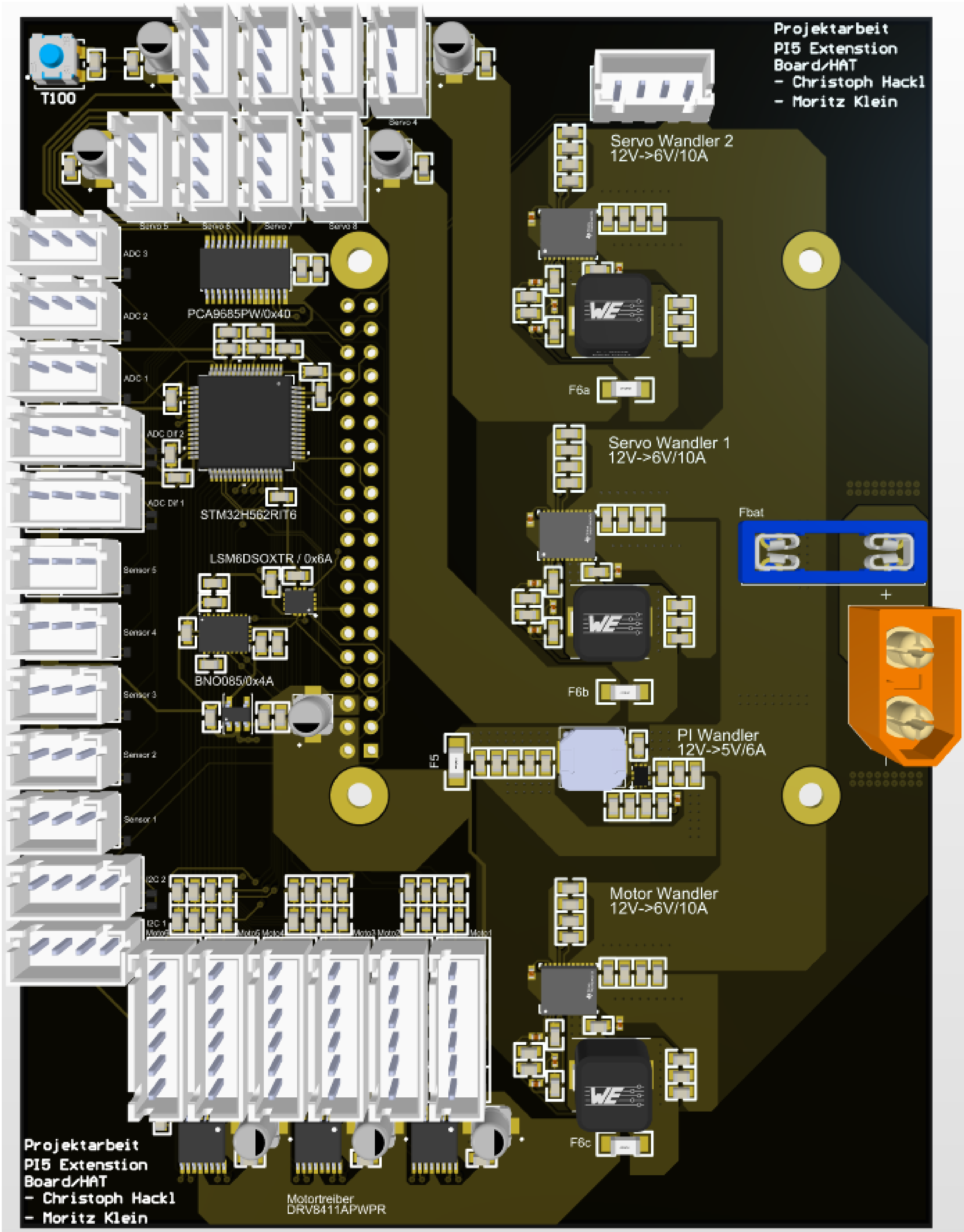

Fig. 2. Current PCB layout of the Raspberry Pi robotics extension board prototype.