IMPROOVED ELECTRONIC DESIGN FOR ROBOT SOCCER PLAYER (MIROSOT)

MAREK SUKOP[1], STEFAN ONDOCKO[1], ARTEM KAGYROV[1]

[1]Technical University of Kosice, Faculty of Mechanical Engineering, Kosice, Slovakia

DOI: 10.17973/MMSJ.2025_12_2025115

marek.sukop@tuke.sk

The article presents the design and implementation of a low-level control system for a multi-agent system (MAS) intended for mobile robots participating in robotic soccer.

The robot was developed in accordance with the specifications of the FIRA MiroSot category and currently serves as an educational platform for demonstrating multi-agent system concepts. The paper includes the complete design of a new printed circuit board, microcontroller programming using free RTOS, and the implementation of motor control algorithms with PID regulation. In addition, the article provides a comparison with the previous electronic design, with which the SjF TUKE Robotics team became the world champion in this category in 2010.

KEYWORDS

Multi-agent system, MiroSot, robot soccer, STM32, FreeRTOS, motor control, PCB design

1 INTRODUCTION

MAS have become an important area of research in mobile robotics, especially in applications requiring cooperation, coordination, and communication among multiple autonomous entities. Robot soccer competitions, such as FIRA MiroSot, provide an excellent platform for the development and testing of multi-agent control strategies in a dynamic and competitive environment [Sukop 2013, Hajduk 2014 & 2019, Stefanisin 2018].

The MiroSot category presents unique challenges due to its strict size (7.5 × 7.5 × 7.5 cm) and weight (650 g) limitations, which require careful integration of electronics, sensors, and particularly mechanical components. It is crucial that the center of gravity of the players be as low as possible and, ideally, very close to the vertical central axis of the robot to achieve optimal dynamic characteristics. This factor determines how the robot will behave dynamically on the field. These constraints demand innovative solutions in both hardware design and software implementation [FIRA 2024, RoboCup 2024, Stefanisin 2018].

This article describes the complete development process of a mobile robot acting as an agent within a multi-agent system. The robot is primarily designed for educational purposes and demonstrates key concepts of distributed control, real-time operating systems, and embedded system design to engineering students. Although the robot complies with the MiroSot specifications, its primary goal is to provide a practical educational platform rather than competitive performance (Fig. 1).

Including real robots in mobile robotics education is essential for a deeper understanding of the subject. Even when operated solely under laboratory conditions, they simulate the complex challenges of the real world, such as unpredictable sensor data, noise, and mechanical imperfections. Hands-on work with hardware enables students to apply theoretical knowledge and develop problem-solving skills that would not emerge in purely virtual simulations. This approach better prepares them for working with real industrial or research robotic systems [Hajduk 2019, Stefanisin 2018, Bellas 2017].

In the 1990s, two leading robotics competitions: RoboCup and FIRA (Federation of International Robot-soccer Association) merged independently, both focused on simulating the game of soccer. The primary objective of these competitions is to test and apply advanced solutions in the fields of robotics and artificial intelligence (AI) in an entertaining and competitive format. The idea of soccer as a challenge for AI was first introduced by Professor Alan Mackworth in 1992 and independently discussed by a group of Japanese researchers [Hajduk 2019, FIRA 2024, RoboCup 2024].

RoboCup, originally known as the Robot J-League, held its first official competition in 1997 in Nagoya, with the participation of over 40 teams. The ultimate and ambitious goal of the RoboCup initiative is to develop, by 2050, a team of fully autonomous humanoid robots capable of defeating the human world champion in traditional soccer [Hajduk 2024, Sukop 2014, FIRA 2024].

Figure 1. MiroSot, category 11 vs. 11 [Hajduk 2019]

FIRA, founded in 1996, continues to organize competitions across multiple categories. Notably, Slovakia achieved first place in the MiroSot category in 2010. The MiroSot category is among the most dynamic in real-world settings, where robot speeds can exceed 3 m/s and accelerations can reach up to 10 m/s². Robot control is achieved through a camera system that transmits visual data to a central control computer, which processes the information, generates strategic commands (often using multi-agent systems, MAS), and sends them to robots via high-frequency wireless communication (Fig. 2). Robotic teams must implement reliable communication protocols with adjustable frequencies to prevent signal interference during matches [Hajduk 2019, FIRA 2024].

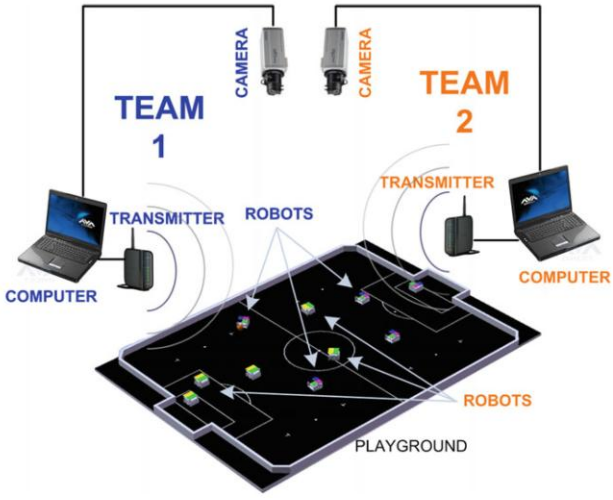

Figure 2. Mirosot architecture [Hajduk 2019]

2 ROBOT DESIGN

The two-wheeled differential drive platform represents one of the most used chassis configurations for mobile robots due to its simplicity and maneuverability. This drive system consists of two independently driven wheels, which allow the robot to change its direction by rotating the wheels at different speeds, or even rotate in place when the wheels turn in opposite directions. Owing to its mechanical simplicity and relatively low cost, this architecture is a popular choice for research, educational, and many commercial applications. The geometry of differential steering simplifies kinematic modelling, which is essential for path planning algorithms and precise motion control (Fig. 3). Although this platform may exhibit limitations in lateral stability or when operating on slippery surfaces, its overall efficiency and robustness make it well suited for a wide range of indoor and moderately demanding outdoor environments [Kagyrov 2024, Moritz 2024, Hoang 2022, Stefanisin 2018].

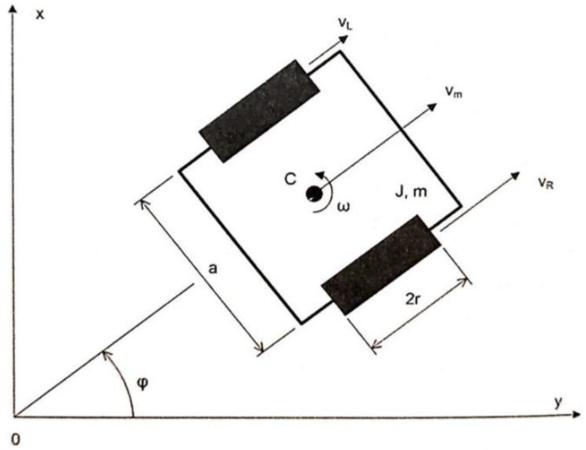

Figure 3. Model of the two wheeled robot [Hajduk 2019, Kagyrov 2024]

The speed of the robot's right and left wheels is a function of its linear and angular velocity [Hajduk 2019, Kagyrov 2024]:

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = f(v, \omega) \tag{1}$$

The equations for directly calculating wheel speed are as follows [Kagyrov 2024]:

$$v_L(t) = v(t) - \omega(t)\frac{a}{2};\ v_R(t) = v(t) + \omega(t)\frac{a}{2} \tag{2}$$

Finally, it is necessary to calculate the wheel rotations so that the embedded PID controller can compute the required PWM duty cycle used to control the wheels [Kagyrov 2024]:

$$n_L = \frac{v_L(t).N_r}{2\pi r f};\ n_R = \frac{v_R(t).N_r}{2\pi r f} \tag{3}$$

The wheel speed (n) is expressed in pulses per sampling period. (Nr) represents the number of pulses per revolution - the encoder resolution. Sampling frequency is f.

3 ROBOT ELECTRONIC DESIGN

The old electronic hardware of the fourth-generation robots (SjF TUKE Robotics) included an 8-bit Atmel AVR microcontroller, an nRF24L01 communication module, an H-bridge for motor control, and 5V and 3.3V voltage regulators (Fig. 4 and Fig. 5) [Hajduk 2019]. The main issue with this configuration was the 8-bit ATmega88-16 microcontroller, which was overclocked to 22.11 MHz [Hajduk 2019]. Despite this, it was necessary to optimize the encoder interrupt code down to the assembly level. This was required because, at the motors' maximum speed (above 6000 rpm) and with an encoder resolution of 512 pulses per revolution, the signals from both encoder channels arrived with a delay of only 4.88 µs or less. Due to the lack of hardware support, these pulses had to be processed entirely in software, which placed an excessive computational load on the microcontroller.

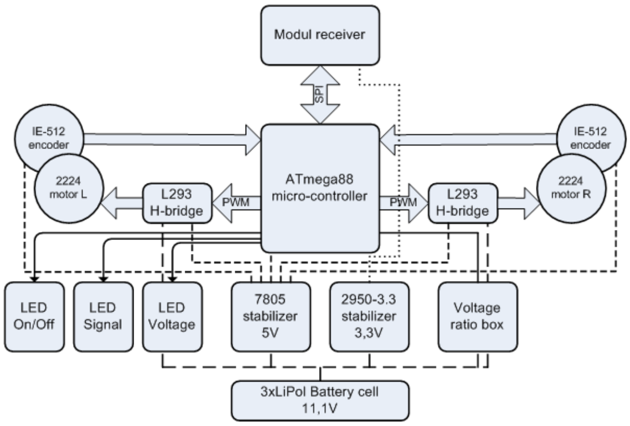

Figure 4. Block diagram of robot hardware [Hajduk 2019]

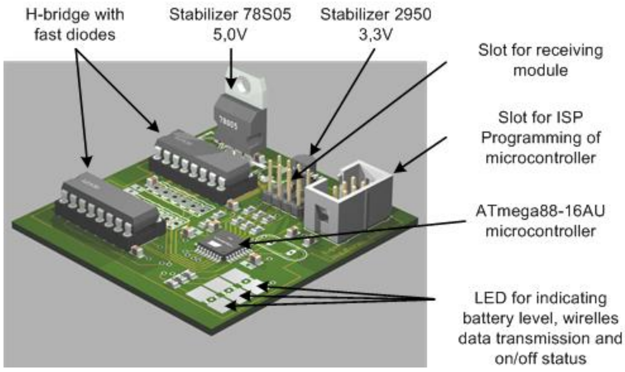

Figure 5. DPS designed for robot electronics [Hajduk 2019]

A further, though considerably smaller, issue was the use of the nRF24L01 module for data transmission between the control computer and the robot. While this module met the requirements for low latency and high-speed communication (up to 2 Mb/s), its compatibility with standard PC interfaces was rather limited. As a result, an additional intermediary module was required to connect it to the computer, which increased the complexity and lack of transparency of the communication system. Although such a setup could be useful for teaching communication principles specifically, in the context of mobile robotics education, it represented an unnecessary complication. Due to the shortcomings, the robot's electronic architecture was redesigned according to the block diagram shown in Figure 6. In addition to the major change involving the replacement of the microcontroller with an STM32F301, the communication module was replaced by an ESP8266 module, which enables communication via both Bluetooth and 2.4 GHz Wi-Fi. The 32-bit microcontroller, equipped with hardware support for encoder signal processing, relieves the processor from computational load and improves overall system performance.

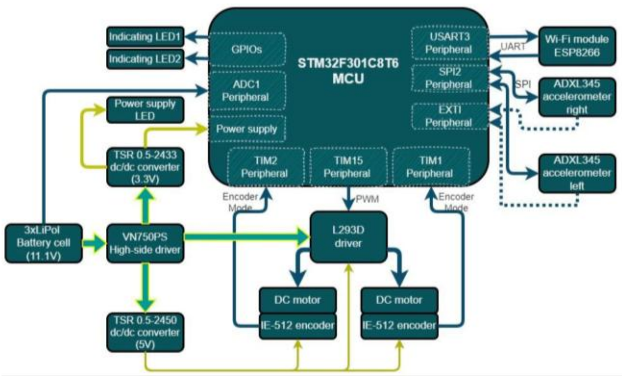

Figure 6. Block diagram with STM32 microcontroller and improved peripherals [Kagyrov 2024]

The ESP-01 Wi-Fi module, developed by Ai-Thinker, is based on the highly integrated ESP8266EX core processor in a compact form factor. This core features the Tensilica L106 32-bit Microcontroller Unit (MCU), engineered for ultra-low-power consumption, and is capable of clock frequencies up to 80 MHz and 160 MHz. The module is compliant with the IEEE 802.11 b/g/n standard and integrates the Wi-Fi MAC/Baseband (BB)/Radio Frequency (RF)/Power Amplifier (PA)/Low Noise Amplifier (LNA), along with an integrated antenna. Furthermore, it supports a Real-Time Operating System (RTOS) and incorporates a comprehensive TCP/IP protocol stack, providing users with flexibility for integration into existing systems or for operation as a standalone network controller [Ai-Thinker 2025]. The ESP8266EX provides a complete and self-contained Wi-Fi networking solution, suitable for either hosting the main application or offloading Wi-Fi networking tasks from another application processor. It enables the integration of Wi-Fi functionality into various systems or independent operation with minimal cost and spatial requirements. When hosting the application, the ESP8266EX boots directly from external flash memory and utilizes an integrated cache to enhance system performance. Alternatively, it can function as a Wi-Fi adapter, enabling wireless internet access for microcontroller-based designs via simple connectivity options such as SPI/SDIO or I2C/UART interfaces. The ESP8266EX is recognized as one of the most highly integrated Wi-Fi chips in the industry, incorporating numerous components—including the power amplifier, antenna switches, filters, RF balun, and low-noise amplifier—which significantly reduces the need for external circuitry and minimizes the required Printed Circuit Board (PCB) area. In addition to its Wi-Fi capabilities, the ESP8266EX includes an enhanced version of the Tensilica L106 Diamond Series 32-bit processor complemented by integrated Static Random-Access Memory (SRAM).

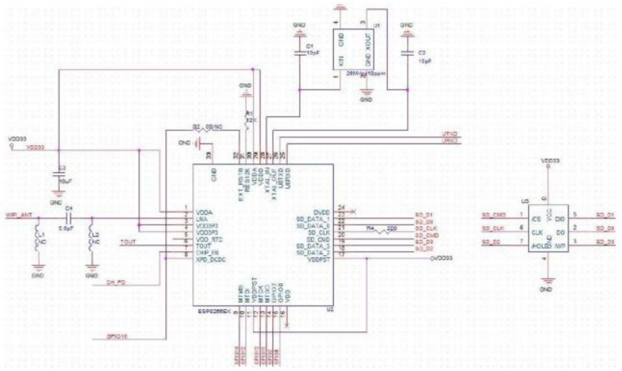

Figure 7. ESP8266 module electronic circuit

The application schematic for the ESP8266EX chip is presented in Figure 7. Essential steps in designing a custom PCB with the ESP8266EX involve integrating an external oscillator, implementing the antenna (either as a trace antenna on the PCB or an external component), and incorporating external Serial Peripheral Interface (SPI) flash memory for application program storage. For the scope of this thesis, the ESP-01 module was utilized with a pre-installed program stored in the flash memory, enabling communication with the Wi-Fi module through the AT command set. The electronic circuit diagram is derived from the block diagram shown in Figure 6. First, we analyze the core of the circuit: the STM32F301C8T6 microcontroller (MCU). Figure 8 illustrates the MCU section of the electronic schematic. As shown, passive components, specifically decoupling capacitors and inductors, were incorporated to ensure a stable and low-noise power supply to the MCU.

The subsequent analysis focuses on the power supply circuit, which illustrates the voltage distribution throughout the entire system. As depicted in Figure 8, the voltage from the LiPo battery is stepped down into three distinct voltage levels: +3.3V, +5V and 12V. The +3.3V rail is utilized to power the microcontroller (MCU), the ESP-01 module, and the accelerometers. The +5V supply energizes the encoders and the logic circuitry of the L293D motor driver. Finally, the +12V is dedicated to supplying the DC motors. Furthermore, the battery is connected to a high-side driver, the VN750PS, which applies voltage to its output when the mechanical switch is in the "on" position.

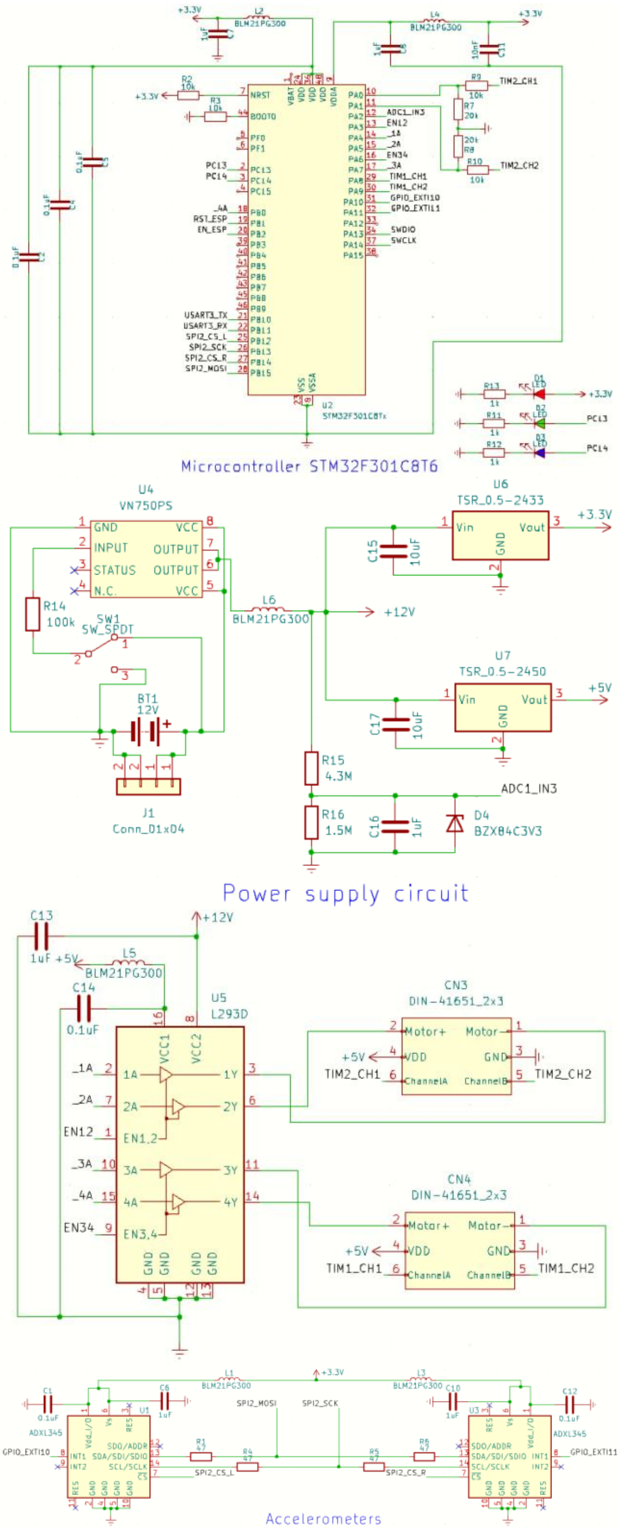

Figure 8. Electronic diagram of the basic circuit parts [Kagyrov 2024]

In addition, the battery voltage is monitored by routing the output of the high-side driver through a voltage divider to one of the MCU's pins, enabling voltage measurement via the Analog-to-Digital Converter (ADC). The L293D driver is connected to two

DIN-41651 connectors, which are configured for attaching two DC motors with integrated encoders.

Figure 8 in its lower part shows the circuit diagram for the two ADXL345 accelerometers. Both accelerometers share a single Serial Peripheral Interface (SPI) bus, necessitating the use of a Chip Select (CS) pin by the MCU to choose the target device for communication. Furthermore, each accelerometer's interrupt pin is connected to an MCU pin configured for External Interrupt (EXTI) functionality, allowing the accelerometer to promptly alert the MCU to significant events.

Upon completion of the robot's electronic schematic, the Printed Circuit Board (PCB) layout design commenced in the PCB editor. PCB layout is a time-consuming process that requires careful pre-planning of component placement on the board. All images and corresponding design files for this PCB, which can be accessed and opened using the KiCad 7.0 software, are provided on the supplemental CD accompanying this work.

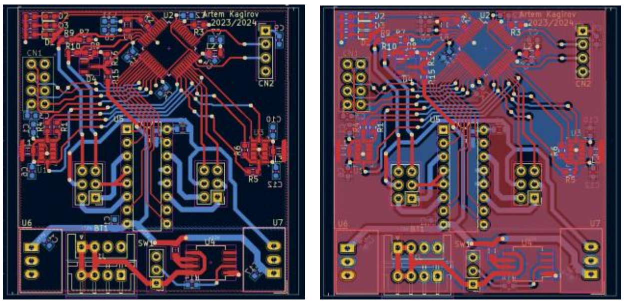

Figure 9. Designed PCBs

Figure 9 presents the Printed Circuit Board (PCB) layout as designed in the KiCad software, clearly showing the copper layers. These layers were implemented due to their inherent advantages, which relate to the practice of copper pouring. Copper pouring refers to the technique of creating large, unbroken copper areas (planes) on the PCB. These planes are typically connected to either the ground (GND) or power supply (VCC) network and serve to fill the unused blank spaces between components and traces.

The benefits of incorporating copper pours (or copper planes) in the PCB design are multifaceted [Spencer 2023]:

- Enhanced Thermal Dissipation: The primary motivation for integrating copper planes into Printed Circuit Board (PCB) design is their superior capability for thermal management. Copper planes effectively conduct heat generated by active components, distributing it across the board and preventing localized hot spots. This practice ensures optimal operating temperatures for components, thereby increasing their lifespan and reliability.
- Reduction of Electromagnetic Interference (EMI): Continuous copper planes function as effective shielding mechanisms, significantly minimizing Electromagnetic Interference (EMI) within the PCB.
- Mitigation of Electromagnetic Field Coupling: The use of copper reduces the coupling of electromagnetic fields between components and conductive elements. This mitigation lowers undesirable noise, cross-talk risk, and signal degradation, thus improving overall system performance.
- Voltage Distribution and Integrity: Copper pours aid in achieving a uniform voltage distribution across the board, minimizing voltage drop and ensuring stable power delivery to components, particularly in high-current draw areas.

The Printed Circuit Board (PCB) design can be systematically divided into three functional sections. The first section, positioned at the bottom of the board, comprises all components necessary for power management and supply regulation to the various electrical parts. Subsequently, the central area of the board accommodates two connectors for the DC motors and the motor driver. These three components operate using two voltage levels: 5V for the control logic and 11V for the motors. The upper section of the PCB was utilized to place all electrical components operating with 3.3V logic, primarily the microcontroller and the Wi-Fi module. Crucially, calculations were performed to determine the required trace width, particularly for the power tracks supplying the motors. This consideration is vital because the motors can draw high currents, especially under load. In accordance with the IPC-2221 standard, the trace width calculations for the motor power supply lines were executed. The formula for calculating the maximum current that a PCB trace can sustain under specific conditions is given by:

$$I = K.\Delta T^{0,44}.(W.H)^{0,725} \quad (4)$$

The variables are defined as follows: *I* is the maximum continuous operating current in amperes [A]; *ΔT* represents the acceptable temperature rise above ambient in degrees Celsius [°C]; *H* and *W* denote the trace thickness and width, respectively, in mils [mil]; and *K* is a constant, set to 0.048 for external traces. The variable W is calculated from the equation below:

$$W = \frac{\sqrt[0,725]{\frac{I}{K.\Delta T^{0,44}}}}{H} \quad (5)$$

Based on the datasheet of the DC motors utilized in the robot, the "rated current (thermal limit)" for continuous operation is specified as 0.7A [Faulhaber 2025]. As the robot incorporates two motors, the total continuous rated current for the drive system is 0.7A+0.7A=1.4A. Furthermore, other low-power electronic components draw current in the milliampere range. To account for this low power consumption overhead, an additional 100mA was conservatively added to the maximum continuous current flowing through the power rail. Consequently, the power rail is designed to continuously handle a maximum current of 1.5A. It is acknowledged that the current flowing to the motors will exceed the rated current during specific short-term intervals, such as during motor start-up, rapid acceleration, or high-torque conditions (under load). In these transient events, the peak current may reach several amperes. This short-term surge is not critical, provided its duration is limited to the order of tens of milliseconds. The PCB traces can carry high currents for brief periods, and the selected LiPo battery (400mAh 11.1V, 60/120C, 4.44Wh) has a maximum discharge current characteristic of 48A, thus eliminating any concern regarding battery performance during current peaks. The next consideration involves the permissible temperature differential (ΔT) between the PCB trace and the ambient environment. The resistance of a copper trace increases with temperature. To ensure that the trace resistance does not change significantly during operation, a maximum temperature difference of 10°C was selected. The resulting change in trace resistance can be calculated using the following formula:

$$R = R_0.[1 + \alpha.(t - t_0)] \quad (6)$$

In this context, R represents the resistance at the final temperature t (in Ω), while R_0 denotes the resistance at the initial temperature t_0 (in Ω). The values t and t_0 are expressed in °C, and α is the temperature coefficient of the material.

We consider $t\text{-}t_0$ is 10°C a α for cooper $3.9*10^{-3}$[1/°C], then: R/R_0=1.039 what indicates that a 10°C temperature increase in the copper trace results in a 3.9% increase in its electrical resistance. This change is considered acceptable for the application requirements of our robotic system.

The final step involves determining the copper trace width on the Printed Circuit Board (PCB). This width is dependent on the copper thickness of the PCB layer, which in our case is 18um. The conversion factor from mm to mils (thousandths of an inch) is 39.3701. The minimum trace width was converted from [mil] to [mm], resulting in 1.0217mm. During the Printed Circuit Board (PCB) design process using the KiCad PCB Editor software, trace widths carrying continuous currents up to 1.5A were ensured to meet or exceed this calculated minimum width.

The designed, dimensioned, and assembled Printed Circuit Board is shown in Figure 10.

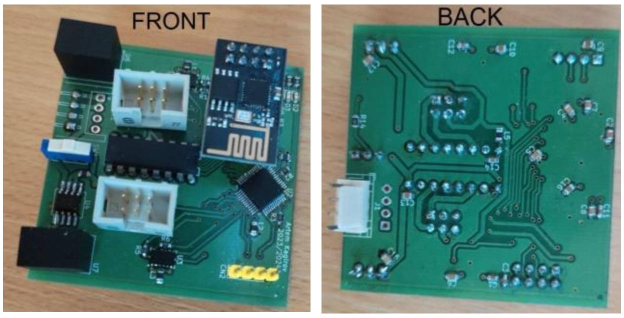

Figure 10. Assembled PCB

4 ROBOT CONTROL SYSTEM

The central component of this robotic footballer is the STM32F301C8T6 Microcontroller Unit (MCU), as it manages all system processes, tasks, communication, and motor control. The MCU program was developed in the C programming language using the STMCubeIDE and STMCubeMX software environments. Furthermore, the STM32 Hardware Abstraction Layer (HAL) driver library was predominantly utilized for the initialization of the microcontroller's peripherals [STMicroelectronics 2025].

To effectively manage tasks on the MCU, the FreeRTOS Real-Time Operating System (RTOS) was employed for task scheduling and management [FreeRTOS 2025]. FreeRTOS is an RTOS kernel specifically designed for embedded systems, providing essential features such as preemptive multitasking, task scheduling, inter-task communication, and synchronization. Being an open-source solution, FreeRTOS is widely adopted across various embedded applications, offering excellent portability and scalability for diverse microcontrollers and processors.

Beyond its core task management capabilities, FreeRTOS provides several features critical for embedded system development. It offers various synchronization primitives, including semaphores, mutexes, event groups, and task notifications, which enable efficient synchronization and communication between concurrently running tasks. These functionalities are fundamental for coordinating real-time operations within the application [Kagyrov 2024].

An **event group** is a 24-bit collection used for signalling or awaiting the occurrence of specific events. Tasks can set or clear individual bits within the event flag group, while other tasks can wait for the establishment or clearance of particular bit combinations before proceeding with execution. This mechanism enables task synchronization based on the occurrence of defined events.

Mutexes (Mutual Exclusion Semaphores) are employed to safeguard shared resources such as peripheral registers, data structures, and variables from simultaneous access by multiple tasks. A task that successfully obtains a mutex acquires exclusive access to the protected resource, thus preventing other tasks from accessing it until the mutex is released. This mechanism ensures that only one task can access a shared resource at any given time, thereby preventing data corruption.

Task Notifications serve as a mechanism for inter-task communication and synchronization. Tasks can send notifications to other tasks to signal events, wake them from a blocked state, or trigger specific actions. Notably, task notifications enable the implementation of very lightweight inter-task signalling, often eliminating the need for event groups or semaphores.

FreeRTOS supports both binary and counting **semaphores**. Binary semaphores function as simple binary flags, which tasks can utilize for signalling or basic synchronization. Counting semaphores, conversely, permit multiple tasks to access a shared resource. Tasks can take (acquire) and give (release) counting semaphores to control access to the shared resource based on its current availability.

Figures 11 and 12 present the block diagram of the robotic footballer's software structure.

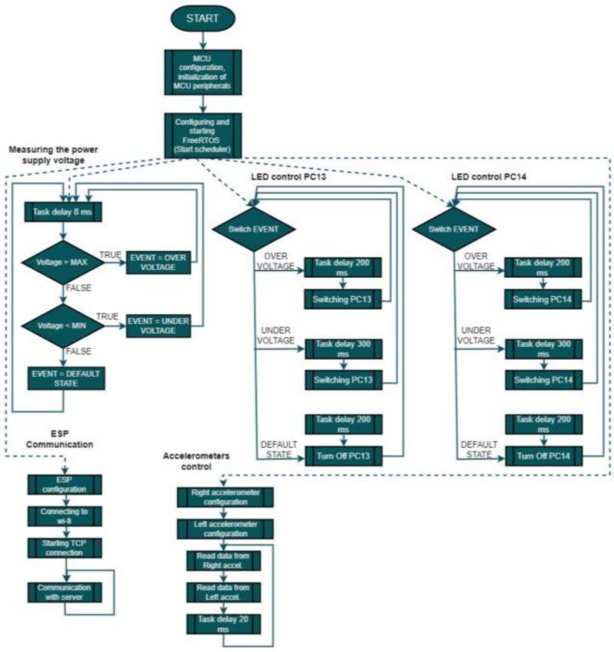

Figure 11. Robot algorithm – first part [Kagyrov 2024]

The first part of the diagram (Figure 11) addresses the initial system setup. Upon microcontroller (MCU) power-up, the MCU configuration and peripheral initialization are executed. Subsequently, the FreeRTOS Real-Time Operating System (RTOS) is configured and launched. Once the FreeRTOS scheduler is running, it manages five concurrent tasks: power supply voltage measurement, control of the PC13 Light-Emitting Diode (LED), control of the PC14 LED, communication with the ESP8266 Wi-Fi module, and control of the accelerometers.

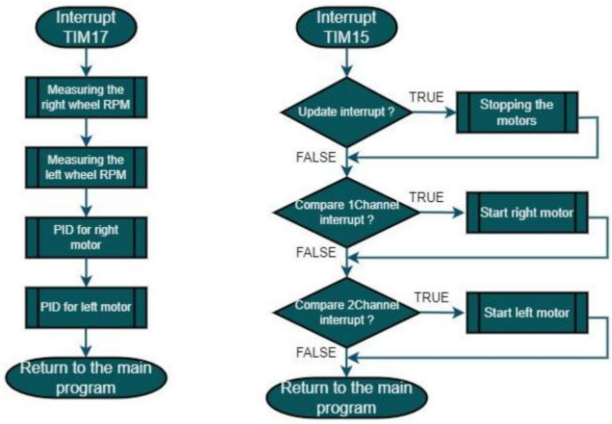

Figure 12. Robot algorithm – second part [Kagyrov 2024]

The second part of the software schematic (Figure 12) illustrates two tasks invoked by timer interrupts. Timer 17 is configured to generate an interrupt every 1ms. Timer 15 is configured in encoder mode, generating an interrupt upon a timer update or a compare event on its first or second channel.

A technical challenge was encountered during the electronic schematic design, which resulted in the inability to utilize the physical output channels for TIM15. Consequently, the motor driver now employs standard General-Purpose Input/Output (GPIO) pins (PA4, PA5, PA7, PB0), necessitating the use of software Pulse-Width Modulation (PWM) methods rather than relying exclusively on hardware-based timer PWM.

The voltage from the LiPo battery is routed through a voltage divider to scale down its value before being measured by the microcontroller's Analog-to-Digital Converter (ADC). The ADC operates in continuous conversion mode and transmits all data to the designated buffer, named battery Voltage, via Direct Memory Access (DMA). This utilization of DMA efficiently offloads the data transfer task from the central processing unit (CPU).

A FreeRTOS task named psMeasure is responsible for calculating the average battery voltage from ten ADC readings. Subsequently, this task compares the average voltage against the defined maximum and minimum voltage thresholds that represent a fully charged and a depleted battery, respectively. Based on this comparison, the task sets or clears specific bits within an Event Group. Consequently, the tasks controlling the PC13 and PC14 Light-Emitting Diodes (LEDs) adjust the state of these LEDs according to the current status of the Event Group.

In the context of FreeRTOS, an Event Group serves as a synchronization primitive employed for inter-task communication and coordination. It allows tasks to suspend execution until a specific combination of event bits is either set or cleared. Event Groups are particularly valuable when multiple tasks are required to coordinate their actions based on the occurrence of specific system events. Tasks can wait for either the simultaneous or individual occurrence of event combinations, depending on the application's synchronization requirements. When an event occurs, the tasks awaiting the corresponding event bits are unblocked, allowing them to resume execution. This mechanism is crucial for designing efficient and synchronized task interactions within the embedded system [FreeRTOS 2025].

Communication with the ESP Wi-Fi module is managed by a dedicated FreeRTOS task. The task begins with the configuration of the ESP module, followed by establishing a Wi-Fi connection and initiating a TCP connection to the server (running on a personal computer). Subsequently, a perpetual loop is initiated to handle all communication between the STM32 MCU and the ESP module. Within this loop, the MCU performs two sub-tasks: sending data to the server and receiving data from the server. Data reception utilizes Direct Memory Access (DMA) for the efficient transfer of data from the Universal Synchronous/Asynchronous Receiver/Transmitter (USART) peripheral to a circular buffer, designated as 'rxBuffer', without incurring CPU overhead. This buffer is dimensioned to hold 512 elements of type 'char'. During each loop iteration, the MCU evaluates the number of bytes received since the last iteration within the circular buffer and processes the new data accordingly. Data transmission is executed within the same loop as reception; however, it is governed by a FreeRTOS software timer to ensure that data packets are sent to the server at precisely regulated intervals.

Motor control represents a complex function within this project. The Microcontroller Unit (MCU) does not directly drive the motors but interfaces with an L293D H-bridge (motor driver). As depicted in Figure 32, motor control involves two sub-tasks triggered by interrupt requests. TIM17 generates interrupts every 1ms to measure the rotational speed of the right and left wheels. Subsequently, the Pulse Width Modulation (PWM) values for the motors are calculated using a Proportional-Integral-Derivative (PID) controller. Timer TIM15 is utilized for PWM generation and generates interrupts corresponding to three conditions: timer update, Channel 1 compare interrupt, and Channel 2 compare interrupt.

A velocity Proportional-Integral-Derivative (PID) controller was employed as the speed regulator for the drives:

$$u(t) = K_P . e(t) + K_i \int_0^t e(\tau) . d\tau + K_d \frac{de(t)}{dt} \qquad (7)$$

The PID control constants were tuned using the experimental Ziegler–Nicholson method, yielding values of K_p = 0.000945, K_i = 0.00000405 and K_d = 0.

5 FINAL REALIZATION

Figure 13 shows the resulting implementation of the new electronics, which is part of the robot chassis.

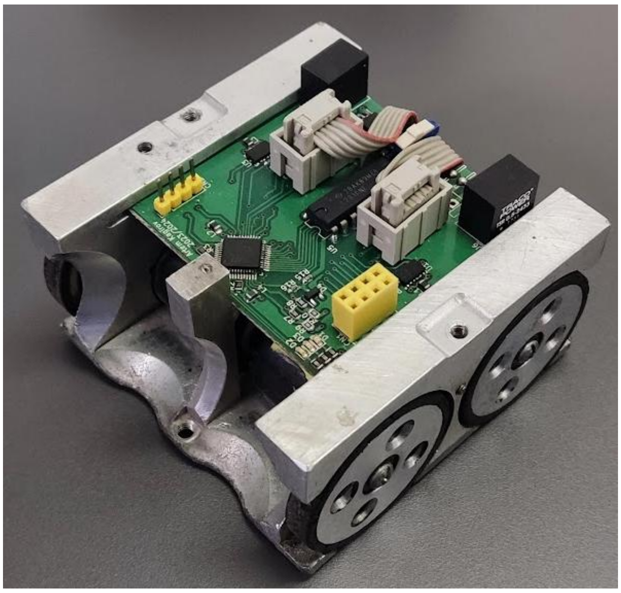

Figure 13. The final realization of the robot

After designing and manufacturing the printed circuit board, functional testing was required. To facilitate this process, a web application was developed to serve as the main agent controlling the subordinate agent — the robotic football player. The decision to implement a web application was made due to its cross-platform compatibility with Windows, Android, and Linux operating systems, requiring only a standard web browser for access.

The web application is conceptually divided into two main components: backend and frontend. The backend was implemented in C++ using the winsock2.h library, which provides network communication capabilities. The frontend was developed using fundamental web technologies such as HTML, CSS, and JavaScript. To enhance the user interface and visual appeal, the Bootstrap framework and Font Awesome library were utilized during frontend development.

Figure 14 illustrates the experimental scenario in which the robot is connected to a server. In this configuration, the application displays various operational parameters that the robot periodically transmits to the server. At the bottom of the user interface, a control joystick with x and y coordinates is located. When the joystick is moved, the coordinate values vary within the range of –100 to 100. Upon receiving these values, the server converts the x and y coordinates into the robot's angular and tangential velocities, which are then regularly transmitted back to the robot.

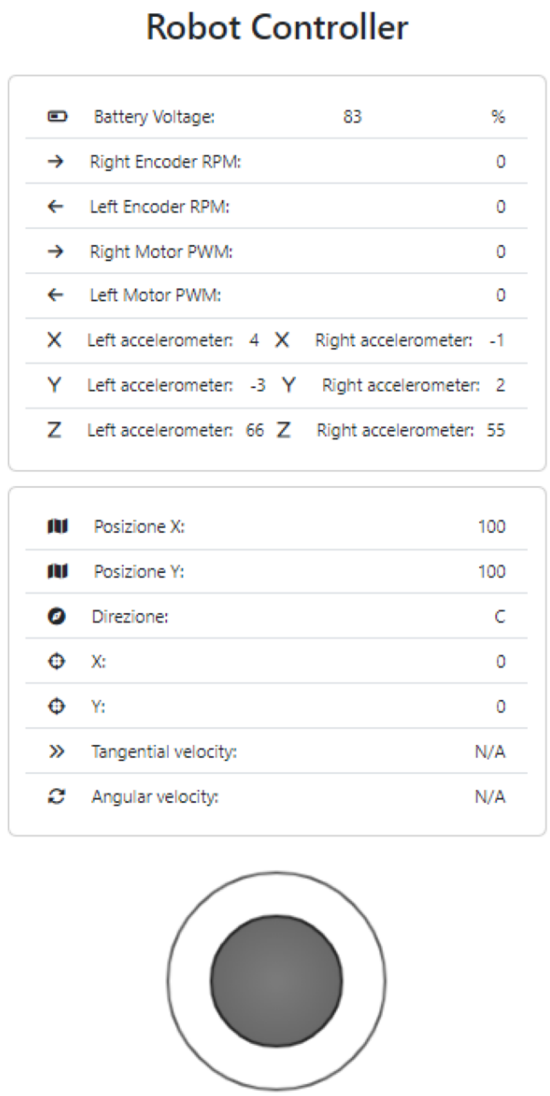

Figure 14. Web application for experimental movement

6 CONCLUSIONS

The design and implementation of an improved electronic architecture for a mobile robot intended for robotic soccer in the MiroSot category began with an analysis of the current state of the robotic player. Based on this analysis, a printed circuit board was developed to meet the requirements of the MiroSot category in accordance with the FIRA regulations. The STM32 microcontroller was selected as the core of the control system, for which low-level control algorithms governing the behaviour of the soccer robot were subsequently designed and implemented. Subsequently, comprehensive testing was conducted to verify the functionality and performance of the robot. To extensive testing, a specialized web application was developed, enabling real-time monitoring and control of the robot's operation.

During the testing phase of the robot, considerable attention was devoted to tuning the proportional–integral (PI) controller for the two DC motors. The Ziegler–Nichols ultimate oscillation method was applied to optimize the control performance. After the proper adjustment of the PI controller parameters, the final stage of robot testing was initiated. Throughout the testing process, several issues related to the robot's control algorithms were identified, requiring careful tuning and refinement. Occasionally, the robot experienced connectivity problems with the web application used for its control, which necessitated a manual restart. Additionally, during the design of the printed circuit board, the microcontroller pins assigned to timer TIM15 were omitted. As a result, part of the PWM signal generation for motor control had to be implemented in software. Despite these difficulties, the robot demonstrated stable and reliable operation under normal conditions.

In addition, the operating time of the robot was evaluated from a fully charged battery to complete discharge, with an average duration of 14 minutes. The discharge rate varied depending on the robot's activity level and the torque on the motor shaft, with higher torque requiring a greater current draw. The robot exhibited the highest energy consumption during frequent changes in motor speed, such as during turning or acceleration and deceleration phases. This effect is caused by the robot's inertia, which forces the motors to draw additional current to overcome dynamic speed variations.

In summary, although certain technical issues and design deficiencies were identified during testing, the overall performance and operational time of the robot remained stable. To ensure smooth operation in real-world environments, it will be necessary to address the identified issues and further optimize the control algorithms.

The proposed design of a mobile robot with control electronics is suitable for use in the educational process focused on the control of mobile robot motion during task execution. It represents a versatile solution that can also be directly applied for testing and verifying various designed strategies of robot group behavior within a multi-agent system (MAS).

ACKNOWLEDGMENTS

The authors would like to thank the Slovak Grant Agency - project VEGA 1/0294/24 - Research and development of a multi-robotic system with distributed intelligence in the cloud, KEGA 043TUKE-4/2024 - Creation of prospective educational tools for the field of additive manufacturing with the implementation of progressive elements of virtual reality. This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-18-0413.

REFERENCES

[Ai-Thinker 2025] ESP-01 WiFi Module Version1.0. Available from: <http://www.ai-thinker.com>

[Bellas 2017] Bellas, F., et al. Robobo: The Next Generation of Educational Robot. Advances in Intelligent Systems and Computing, 2017, Vol. 694, pp. 359-369.

[Faulhaber 2025] Datasheet. DC-Micromotors Precious Metal Commutation. FAULHABER. Available from: <https://www.faulhaber.com/en/products/series/2224sr/#lower>

[FIRA 2024] FIRA. FIRA RoboWorld Cup. Available from: <https://firaworldcup.org/>

[FreeRTOS 2025] The FreeRTOS™ Reference Manual. Amazon Web Services. Available from: <http://www.FreeRTOS.org>

[Hajduk 2014] Hajduk, M., et al. Strategic behavior of the group of mobile robots for robosoccer (category Mirosot). In: IEEE RAAD 2014: Robotics in Alpe-Adria-Danube Region. ISBN 978-1-4799-6797-1.

[Hajduk 2019] Hajduk, M., et al. Cognitive Multi-agent Systems – Strategies and Aplications to Mobile Robotics and Robosoccer. Springer Nature, 2017, 1st edition, 127 p. ISBN 978-3-319-93687-1.

[Hoang 2022] Hoang, T., et al. Navigation for Two-Wheeled Differential Mobile Robot in the Special Environment. In: Int. Conf. on Industrial Networks and Intelligent Systems, 2022, pp. 167-183.

https://link.springer.com/chapter/10.1007/978-3-031-08878-0_12
[Kagyrov 2024] Kagyrov, A. Lower-level control development of multiagent system for mobile robots. Master's Thesis. Kosice: TUKE, 2024.
[MiroSot] Fira MiroSot game rules. Available from: <https://www.inovamicro.com/support/MiroSot_Rules.pdf>.
[Moritz 2024] Moritz, J., et al. Design and Modeling of a Two-wheeled Differential Drive Robot. International Journal of Control, Automation, and Systems, 2024, Vol. 22, No. 7, pp. 2273-2282, 2024.
[RoboCup 2024] RoboCup. Available from: <www.robocup.org/a_brief_history_of_robocup>
[Spencer 2023] Spencer, S. The Importance of Copper Pour in Empty Areas on PCBs. Available from: <https://www.theengineeringknowledge.com/copper-pour-pcb/>.
[Stefanisin 2018] Stefanisin, J. Design and implementation of a mobile robotic football strategy of the MiroSot category within the ROS system (Navrh a realizacia strategie mobilneho robotickeho futbalu kategorie MiroSot v ramci systemu ROS), 2018. Available from: <https://opac.crzp.sk/?fn=detailBiblioFormChildE2C441&sid=A3D1C8DEF803EE7E3A76222CCFB9&seo=CRZP-detail-kniha>. (in Slovak)
[STMicroelectronics 2025] UM1786 User manual Description of STM32F3 HAL and low-layer drivers, 2020. Available from: <https://www.st.com>
[Sukop 2013] Sukop, M., et al. Higher speeds of data between computers and mobile robots based on increase in the number of transmitters (robosoccer). In: Computer Aided Production Engineering. Lublin: Lublin Univ. of Technol., 2013, pp. 75-80. ISBN 978-83-63569-72-3. https://doi.org/10.1007/978-3-319-10783-7_18.

CONTACTS:

Marek Sukop, Prof. Ing. PhD.
Technical University of Kosice, Faculty of Mechanical Engineering
Institute of Automation, Mechatronics, Robotics and Production Techniques
Letna 9, 04200 Kosice, Slovak Republic
marek.sukop@tuke.sk