JIEE Časopis priemyselnej elektrotechniky
Journal of Industrial Electrical Engineering
ISSN 2454-0900
www.jiee.eu

Hardware development for a two-wheeled self-balancing robot

[1]*Adam PAČNÁR*, [2]*Matej BEREŠ*

[1,2] Department of Theoretical and Industrial Electrical Engineering, FEI TU of Košice

[1]adam.pacnar@student.tuke.sk, [2]matej.beres@tuke.sk

***Abstract* — Self-balancing robots represent a convergence of technology and mechanics, showcasing the quest for stability in dynamic environments. This article delves into the critical role of hardware development in shaping the capabilities of these robots, exploring the intricate interplay of sensors, actuators, and microcontrollers. The article underscores the importance of thoughtful hardware development for self-balancing robots, offering insights into the nuanced process of selecting and integrating components to achieve optimal performance in dynamic environments.**

***Keywords* — accelerometer, gyroscope, microcontroller, MPU, self-balancing robot, stepper motor**

I. Introduction

Self-balancing robots characterize a fusion of technology and mechanics, embodying the pursuit of stability in dynamic environments. Engineered to maintain balance while traversing diverse terrains. These robots have garnered increasing attention for their potential applications in fields ranging from personal transportation to industrial automation. At the core of their functionality lies the complex interplay of hardware components, underscoring the critical role of hardware development in shaping the capabilities of these robotic systems.

The article begins with the formulation of appropriate hardware input requirements and specifications, establishing a foundation rooted in accuracy and functionality. The selection of suitable sensors and actuators is explored, point up their fundamental role in enabling the robot to respond to its environment. Navigating through the difficulties of choosing a suitable control element – the microcontroller – becomes important. This element is essential for composing the complex dance of hardware components, ensuring seamless self-balance and achieving the required movement.

II. Hardware Input Requirements and Specifications

When formulating the necessary sensors for a self-balancing robot, key parameters influencing its ability to maintain balance, control movements, and execute effective stabilization are crucial. In this section, specific requirements that sensors must meet are precisely specified, essential for the design of a self-balancing robot. The application of theoretical knowledge provides a suitable inspiration for implementing specific tasks in the design of the self-balancing robot. From the analysis and theoretical insights, it is possible to define the main final requirements:

1) Implement robust stabilization and control algorithms to enable the robot to maintain balance in motion or in a stationary state.
2) Integrate sensitive sensors and a control system that quickly responds to changes in the robot's movement.
3) Analyze the control system and ensure that the motors provide sufficient power for a rapid and efficient response to commands from the control system.
4) Select a microcontroller with adequate computational power and suitable interfaces for effective control and motion regulation of the robot.
5) Choose batteries with sufficient capacity and select an appropriate battery type considering its

advantages.

III. ANALYTICAL EXPLORATION OF HARDWARE OPTIONS

The chapter delves into an in-depth examination of existing hardware options. It involves the evaluation of their advantages, disadvantages, and suitability for application in a self-balancing robot. Furthermore, the hardware can be divided into three key components: the sensor components, actuators, and the microcontroller.

A. Sensor components

The sensor component ensures precision in measuring the tilt angle, a critical factor for maintaining stability and achieving precise robot movement. The accuracy of these measurements directly impacts the robot's ability to respond effectively to changes in its environment.

The accurate measurement of inclination is ensured by accelerometers, and their dynamic range must be sufficient to capture the tilt angle. A fast-sampling frequency is another crucial parameter that allows an immediate response to changes in inclination, critical for handling the dynamic movements of the robot. Another sensor essential for achieving precise evaluation is the gyroscope, responsible for measuring angular velocities or rotations. The combination of an accelerometer and gyroscope into a single compact solution is known as a Motion Processing Unit (MPU) chip.

Among the most used MPU sensors are MPU-9250, MPU-6050, and MPU-9255. Graphic comparison is on fig. 1. All three sensors have similar measurement ranges for both the gyroscope and accelerometer. MPU-9250 and MPU-9255 additionally feature magnetometer functionality, which can be advantageous. The communication of all three sensors occurs through the I2C protocol, streamlining their connection to the microcontroller. [1]

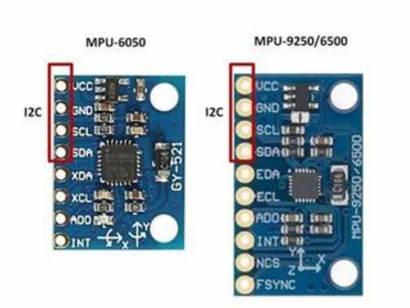

Fig. 1. Graphic comparison mpu-6050 vs mpu-9255

B. Actuators

Actuators play a crucial role in stabilizing the robot's components and serve as the output based on sensor inputs and microcontroller computations. Their function is vital in translating the information gathered by the sensors into physical movements that maintain balance and stability. In the context of self-balancing robots, commonly used actuators include stepper motors and servo motors. Comparison of those two motors is on Fig. 2.

Fig. 2. Comparison of Stepper Motor (on the right) and Servo Motor (on the left)

The choice between stepper and servo motors depends on specific requirements, considering factors such as precision, speed, and overall system complexity in the design of self-balancing robots.[2]

C. Microcontroller

The microcontroller serves as the heart of the system, ensuring continuous coordination between sensors and actuators. It is responsible for processing data from sensors, executing control algorithms, and generating commands for the actuators.

When choosing a microcontroller for controlling a self-balancing robot, several popular platforms are available, including Arduino, Raspberry Pi, esp32, and others. Each of these platforms has its advantages and disadvantages, and the decision depends on the specific needs and requirements of the project. Graphics comparison of those microcontrollers is on Fig. 3.

Arduino is known for its simplicity and beginner-friendly environment. It has a wealth of available libraries, simplifying development. However, its limited performance may be a disadvantage for more complex control and data processing. The frequency usually ranges from 8-16 MHz, and RAM is limited to hundreds of bytes or a few kilobytes. [3]

Raspberry Pi applies single or multi-core processors with a frequency ranging from 1 GHz to several GHz. RAM ranges from 1 to several gigabytes, providing significantly higher computational power and memory compared to Arduino. It is ideal for projects requiring complex control and data processing. [3]

The esp32 combines the advantages of simple programming like Arduino with high performance and equipment like Raspberry Pi. Its frequency reaches several hundred MHz, providing sufficient power for more demanding tasks. RAM ranges from a few hundred kilobytes to several megabytes. [3]

Fig. 3. Comparison of Arduino, Raspberry Pi, and esp32

The decision on selecting a microcontroller depends on the specific requirements of the project, with Arduino suitable for simpler applications, Raspberry Pi for more demanding computations, and esp32 serving as a compromise between simplicity and performance. [3]

IV. SELECTION OF OPTIMIZED HARDWARE

The optimal selection of hardware components, involving MPU sensors, actuators, and microcontrollers, relies on a thorough comparison of the input hardware requirements. This comparative analysis serves as a critical foundation for making informed decisions in the hardware design process.

When examining the specific demands and characteristics of each component, customization of the selection to meet the unique needs of the self-balancing robot is essential.

A. Optimal choice for sensorics part

For example, the MPU-6050 sensor come out as a suitable choice, providing features such as a dynamic range, sampling frequency, and integration capabilities crucial for precise inclination measurement, which is fundamental for maintaining balance.

The MEMS triple-axis gyroscope in the MPU-6050 offers a wide range of features, including digital output sensors for angular velocity (gyroscopes) on the X, Y, and Z axes. Users can programmatically set the full-scale range of the gyroscope to ±250, ±500, ±1000, and ±2000°/s. Integrated 16-bit ADC enables simultaneous sampling of gyroscopes. Additional advantages include improved temperature stability of inclination and sensitivity, low noise levels at low frequencies, and a digitally programmable low-pass filter. The gyroscope's operating current is 3.6 mA, and in standby mode, it is only 5 μA. The gyroscope is fully calibrated and has a user self-test option. [1],[5]

The MEMS triple-axis accelerometer in the MPU-6050 has an equally broad range of functionalities. The digital output triple-axis accelerometer allows a programmable full-scale range of ±2g, ±4g, ±8g, and ±16g. Integrated 16-bit ADC enables simultaneous sampling of accelerometers without the need for an external multiplexer. The accelerometer's operating current is 500 μA, with low-power modes at 10 μA at 1.25 Hz, 20 μA at 5 Hz, 60 μA at 20 Hz, and 110 μA at 40 Hz. Additionally, it includes features such as orientation detection and signaling, tap detection, and user-programmable interrupts, including high interrupts. [1],[5]

B. Optimal choice for actuator

Actuators, as mechanical output elements, embody considerations such as torque, speed, and control mechanisms to identify suitable options. For example, the NEMA 17 stepper motor is regarded as a practical actuator, offering precise control over angular displacement, making it well-suited for self-balancing robots. This motor stands out for its ability to offer precise control over angular displacement. Additionally, it's important to acknowledge that its operational requirements include the need for a dedicated driver to ensure optimal performance in robotic applications. [6]

C. *Choosing a suitable microcontroller*

Finally, the esp32 microcontroller occurs as a suitable choice, given its features, including integrated Wi-Fi and Bluetooth modules. This addition provides connectivity options for software extensions, making it an ideal choice for applications that may benefit from wireless communication capabilities.

In the Wi-Fi domain, it supports protocols 802.11 b/g/n, operating within the frequency channels ranging from 2412 to 2484 MHz. Within the Bluetooth spectrum, it embraces version 4.2 BR/EDR and Bluetooth LE specifications, featuring a sensitive receiver with a sensitivity of –97 dBm. Furthermore, it combines a 40 MHz crystal, a 4 MB SPI flash, and supports various interfaces from SD card to UART and SPI. Operating voltage spans from 3.0 V to 3.6 V, and the microcontroller establishes flexibility to temperatures ranging from –40 °C to +85 °C.[4]

These characteristics mutually position the ESP32 microcontroller as an ideal choice for the control system of a self-balancing robot, seamlessly combining power, efficiency, and cost-effectiveness.

V. CONCLUSION

In conclusion, the development of a self-balancing robot requires precise consideration of hardware components, as they form the stability and functionality. The complex interplay of sensors, actuators, and microcontrollers is crucial for achieving seamless balance and precise movement.

The formulation of hardware input requirements and specifications sets the stage for a focused hardware design process. The selection of sensors, represented by the MPU-6050, ensures accurate measurement of tilt angles, a foundational element for maintaining balance. Actuators, represented by stepper and servo motors, play a fundamental role in transforming sensor inputs and microcontroller computations into physical movements that maintain balance and stability. The microcontroller serves as the heart of the system, coordinating the intricate dance of hardware components. The comparison of popular microcontroller platforms, including Arduino, Raspberry Pi, and esp32, reveals varying advantages. The optimized hardware selection involves a careful balance of components. For example, with the MPU-6050 sensor, NEMA 17 stepper motor, Li-Ion battery, and esp32 microcontroller developing as a well-wide combination. The integration of these components provides not only stability and precision but also connectivity options through the esp32's Wi-Fi and Bluetooth module.

REFERENCES

[1] A. Syed. (2021). MPU-6050 (Accelerometer and Gyroscopic sensor). Available on the Internet: <https://www.researchgate.net/publication/353103399_MPU-6050_Accelerometer_and_Gyroscopic_sensor>

[2] H. Gyll. (2016). Stepper Motor or Servo Motor: Which should it be? Available on the Internet: <https://www.kollmorgen.com/sites/default/files/Servo_or_Stepper_08_05_16_WhitePaper_FINAL.pdf >

[3] S. Dhawan, S. Amanpreet, T. Aditi, P. Nikhil. (2020). An Overview of IoT Hardware Development Platforms. Available on the Internet: <https://www.researchgate.net/publication/344207338_An_Overview_of_IoT_Hardware_Development_Platforms>

[4] Espressif Systems. (2023). ESP32-WROOM-32D (Datasheet). Available on the Internet: <https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf>

[5] TDK Electronics. (2013). MPU-6050 (Datasheet). Available on the Internet: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132807/TDK/MPU-6050.html>

[6] Arefin, S. Erfan. (2023). Simple Two-wheel Self-Balancing Robot Implementation. Available on the Internet: <https://www.semanticscholar.org/reader/317ba5663ab5b2d552d58dd7fe7f2783fd8809c5>