Challenges and Issues of Modern Science, 3 (2024). ISSN: 3083-5704
URL: https://cims.fti.dp.ua/j/article/view/228
PURL: https://purl.org/cims/2403.013

Using micro:bit for building distributed sensor networks in robotic systems

Roman Duban, Yurii Yurko

Purpose. The purpose of the paper is to explore the possibilities of integrating the micro:bit microcontroller with external devices through UART, I2C, and SPI protocols to create modular and distributed sensor networks in robotic systems. The article focuses on the compatibility issues of these protocols with other devices and possible solutions. **Design / Method / Approach**. The paper uses an experimental approach, examining the use of standard communication protocols to connect micro:bit with external sensors and controllers, such as Raspberry Pi, to build a modular sensor network. **Findings**. It was found that micro:bit supports all major communication protocols, but there are compatibility issues with the radio protocol and external devices. UART, I2C, and SPI allow micro:bit to integrate with a wide range of sensors and devices, but additional configuration is needed for stable operation. **Theoretical Implications**. The article demonstrates that the use of standard protocols for integrating devices into robotic systems is a key aspect of creating modular sensor networks. It shows that micro:bit can be an important component to developing such systems. **Practical Implications**. The results can be applied in educational projects and for the creation of modular robotic systems where micro:bit acts as a manager module of different sensors. This expands the potential use of inexpensive microcontrollers in real-world scenarios. **Originality / Value**. The article contributes to research on integrating micro:bit into robotic systems, which is important for educational robotics, as it shows new possibilities for using microcontrollers in distributed systems. **Research Limitations / Future Research**. The limitation of the study is that it is based on theoretical assumptions about protocol stability when connecting multiple modules. Future research should focus on real-world testing of compatibility with other devices and optimizing wireless communication. **Paper Type**. This is a technical note with both experimental and theoretical elements, exploring the issues of integrating micro:bit into distributed systems.

Keywords:

distributed sensor network, microcontroller, robotic system, communication protocols, modular system

Contributor Details:

Roman Duban, Cand.Sc., Sr.Lect., Kharkiv University of Technology "STEP": Kharkiv, UA, duban_r@itstep.academy

Yurii Yurko, Cand.Sc., Assoc.Prof., Kryvyi Rih National University: Kryvyi Rih, UA, yyurko@yahoo.com

In modern robotic systems, distributed sensor networks play a crucial role in providing efficient data collection, processing, and transmission for real-time decision-making. These networks can be built using microcontrollers and connected sensors with configured communication protocols. Modern microcontroller boards are becoming increasingly powerful and often include basic sensors, making it easier to integrate them into robotic projects. One of the most attractive options today is the micro:bit board from BBC, which includes built-in sensors and supports wireless communication through its built-in radio channel (Ruch, 2022).

Micro:bit also provides an additional layer of abstraction that significantly simplifies the programming and configuration of communication protocols. This makes micro:bit a convenient tool for building distributed sensor networks using standard communication protocols such as UART (Universal Asynchronous Receiver/Transmitter), I2C (Inter-Integrated Circuit), and SPI (Serial Peripheral Interface). Therefore, it is important to explore how micro:bit can be used to build distributed sensor networks with flexible architecture and reliable communication, as well as its integration into more complex systems like ROS2 (Robot Operating System 2), where communication capabilities play a key role in ensuring the reliability and efficiency of such networks.

Purpose

The purpose of this study is to explore the possibilities of integrating micro:bit into robotic systems using standard communication protocols. The central micro:bit module will be connected to the Raspberry Pi 4 via the UART protocol, which ensures full-duplex data exchange. I2C will be used for inter-module communication, allowing for the creation of a distributed sensor network, while SPI will serve for connecting powerful sensors to individual microcontrollers. Special attention is given to the use of the built-in micro:bit radio protocol as a backup channel for wireless communication between modules, enabling reliable communication in the event of failures in the primary communication channels or when modules are remotely located.

This study also addresses compatibility and reliability issues of the mentioned protocols when building scalable sensor networks, as well as their integration with more powerful systems such as ROS2.

Materials and Methods

Hardware component

In recent years, the micro:bit board from BBC has gained significant popularity due to its ease of use and accessibility for various age groups, from schoolchildren to developers and students. It is widely recognized as a convenient and powerful tool in STEM (Science, Technology, Engineering and Mathematics) education, making it easier to teach fundamental concepts in programming, electronics, and robotics. Micro:bit is equipped with several built-in sensors, such as an accelerometer, magnetometer, thermometer, and light sensor, which makes it suitable for creating basic sensor networks without the need for additional devices. Despite its small size (approximately half the size of a credit card), the micro:bit can be easily integrated into various projects (Voštinár, P., & Knežník, J. 2020). For more convenient interaction with digital ports, an additional connector can be used, allowing the board to be connected similarly to a cartridge.

This research utilized several micro:bit v2.2 boards, functioning as the main communication module, a connected module, and a module interacting via the radio protocol. Each of these modules, in addition to built-in sensors, supported the connection of external sensors via standard communication protocols. This enabled the creation of distributed sensor networks, where certain modules could interact without direct wired connections. Micro:bit supports wireless communication through a radio channel, which operates over distances of up to 20 meters. This radio protocol is particularly useful for robotic systems, where individual modules can identify themselves before physically connecting to the main module. Such an approach is effective for robots with interchangeable parts, allowing them to be detected before being physically connected or for providing a backup communication channel in case of wired connection failures.

A Raspberry Pi 4 single-board computer was used for data collection and processing, serving as the central processing unit for signals from all the micro:bit modules. The computing power of the Raspberry Pi allows to use it in robotics and IoT (Internet of Things) projects (Ahmad, B. et al., 2023). Using Raspberry Pi board in the research allowed for complex calculations that would otherwise be impossible on the micro:bit due to its limited resources.

Thus, the hardware architecture of this research combined the accessibility and modularity of micro:bit with the processing power of Raspberry Pi, enabling the creation of a distributed sensor network for real-time data collection and analysis. This is particularly relevant in the context of STEM education, where micro:bit helps to practically teach the basics of sensor systems and robotics.

Software component

The micro:bit platform is distinguished by its high level of abstraction, which greatly simplifies programming for users of different skill levels. Thanks to this additional abstraction layer, micro:bit can be programmed using a variety of languages and environments, most notably JavaScript (via the MakeCode block editor) and Python. For educational purposes and simpler tasks, the MakeCode environment allows users to program micro:bit using a block-based approach, making it especially suitable for young students or beginners. The blocks can also be converted into JavaScript code, providing an easy transition to text-based programming (Cederqvist, AM. 2022).

When more advanced control over the hardware is required, micro:bit also supports MicroPython. This lightweight version of Python is specifically optimized for microcontroller programming and offers a syntax very close to that of standard Python. However, certain functions in MicroPython are tailored to the specific needs of microcontroller environments, such as handling GPIO pins and interacting with onboard sensors. This research relied on the functionalities provided by MicroPython, which was sufficient to implement the desired features without needing to delve into the lower levels of abstraction available through C. For more advanced tasks that require low-level control of the hardware, C programming can be employed. This approach gives developers direct access to micro:bit's internal registers and peripherals, which can be crucial when optimizing performance or handling hardware interruptions. While MicroPython is easier and faster to use for most tasks, C provides the flexibility to maximize the hardware's capabilities when necessary.

On the Raspberry Pi 4, which served as the central processing unit for the distributed sensor network, the Raspbian (a Debian-based Linux operating system) was installed. This system provided the necessary environment to run ROS2, which is widely used in robotics for handling sensor data, executing control algorithms, and managing communication between various hardware modules (Carreira, R. et al., 2024).

In this research, ROS2 played a key role in integrating the data from the various micro:bit modules into a unified system. Through ROS2, it was possible to manage the sensor data streams coming from multiple sources, visualize the collected data in real-time, and analyze sensor readings for further processing. By leveraging ROS2's modular architecture, the system could be easily scaled to include additional micro:bit modules or other sensors connected via Raspberry Pi.

Moreover, ROS2 enabled seamless communication between the different components of the network, allowing the Raspberry Pi to interact with the distributed micro:bit modules, process the data, and send commands back to individual sensors. This setup facilitated real-time control and decision-making, crucial for efficient robot operation in dynamic environments.

Through the use of MicroPython on micro:bit and ROS2 on Raspberry Pi, this system demonstrated how low-cost microcontrollers can be effectively integrated into more complex and powerful robotics frameworks.

Communication protocols and architecture

The BBC micro:bit microcontroller offers a flexible approach to building sensor networks with its support for several communication protocols, such as UART, I2C, and SPI, while also providing a built-in radio protocol for wireless communication. Despite the additional abstraction layer that simplifies programming, this study aims to explore how these communication protocols can be effectively utilized to create a robust, distributed sensor network. The network was designed to ensure reliable data transmission in real-time, even under potential communication failures. According to research (Zhuang, C. 2024), the UART, I2C, and SPI protocols each have their unique advantages and limitations for integration into measurement and control devices, allowing for an informed choice of the appropriate protocol for a specific system. This study detailed how these protocols impact the reliability and performance of sensor networks in distributed robotics systems.

The system was built with several micro:bit v2.2 boards, where each board had specific responsibilities: one functioned as the central module connected to a Raspberry Pi 4, while the others acted as peripheral modules that communicated through various protocols depending on their roles. Figure 1 has visually represent the central micro connected to the Raspberry Pi via UART, with additional micro:bit modules connected through I2C, and high-speed sensors linked via SPI.

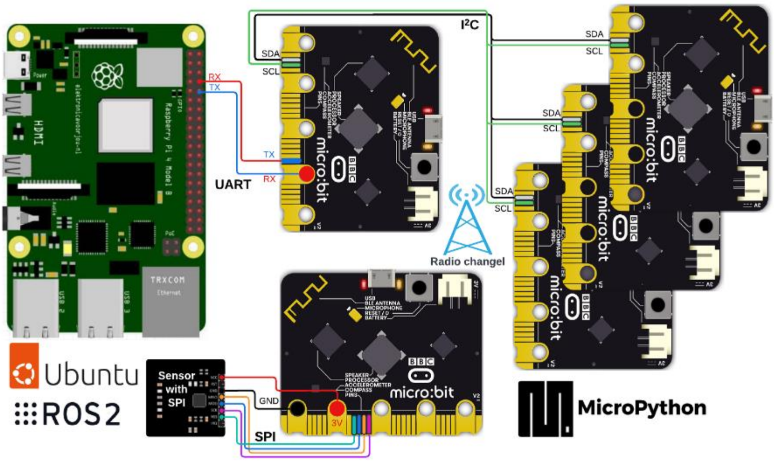

Figure 1 – System Architecture for distributed sensor networks (Source: Authors)

It also shows remote modules communicating over the radio protocol, highlighting the redundancy in the communication channels.

Communication protocols:

– UART: The central micro:bit was connected to the Raspberry Pi 4 using the UART protocol, enabling full-duplex data transfer. This connection served as the backbone for the network, transmitting data between the distributed micro:bit modules and the Raspberry Pi. The real-time processing of sensor data and control signals was handled by ROS2, which was installed on the Raspberry Pi.

– I2C: The I2C protocol was used to interconnect the micro:bit modules, allowing multiple peripheral devices to communicate with the central module. Each micro:bit module, assigned a unique address, contributed to the distributed sensor network. This configuration enabled scalable communication across several modules without adding extra wiring for each device.

– SPI: High-speed sensors, such as cameras or motion detectors, were connected to the micro:bit boards via the SPI protocol. Although SPI requires additional GPIO pins for each connected sensor, it provided the necessary speed and efficiency for transmitting large amounts of data. The microcontroller handled these high-speed sensors to offload processing from the Raspberry Pi.

– Radio Protocol: One of the key features of micro:bit is its built-in radio protocol. In this study, the radio protocol was used to establish a wireless mesh network between the micro:bit modules, allowing communication across distances up to 20 meters. This was particularly useful for scenarios where the modules were not physically connected via I2C, as the radio protocol acted as a backup channel when primary wired communication failed. The central micro:bit served as a bridge between the wireless nodes and the Raspberry Pi, translating data between different communication layers.

The final architecture represented a hybrid approach, leveraging both wired and wireless communication. The architecture ensured redundancy and robustness by allowing fallback to the radio protocol in the event of I2C failure or module disconnection. Additionally, SPI provided high-speed communication for complex sensors, while UART maintained full-duplex communication with the Raspberry Pi for real-time data processing.

Experimentation Process

To validate the functionality and reliability of this hybrid communication system, a series of experiments were conducted, including the following steps:

1. Initial integration: the experiment began with connecting the central micro:bit to the Raspberry Pi 4 through UART for real-time, full-duplex data transmission. The setup ensured that ROS2 on the Raspberry Pi could handle the incoming data from the distributed network efficiently.

2. Peripheral sensor integration: additional sensors were connected to the micro:bit modules using SPI. Stability tests were conducted to evaluate the reliability of SPI communication under different data loads and conditions.

3. Radio communication between modules: the radio protocol was tested to assess the performance of wireless communication between the micro:bit modules. The experiment examined the feasibility of using the radio protocol as a backup for the I2C connection, ensuring that the system could maintain communication even in case of failures.

4. Data analysis: the data collected from the micro:bit modules was transmitted to the Raspberry Pi and processed in ROS2. The analysis focused on visualization of the sensor data, performance evaluation of the network, and command execution on the peripheral modules. Additionally, the experiment assessed the impact of radio protocol range on system performance.

Particular attention was given to the reliability of the system when using the radio protocol as a backup communication channel. In cases where the I2C connection was interrupted, the system automatically switched to radio communication, minimizing data loss. This feature ensured that the network remained operational even in scenarios where wired connections were compromised.

Discussion and Results

The results of the study showed that micro:bit can be effectively integrated into distributed sensor networks through standard protocols such as UART, I2C, and SPI. UART demonstrated stability for full-duplex data exchange between the micro:bit microcontroller and the powerful ROS2 platform on the Raspberry Pi 4, allowing for real-time control of peripheral modules and data processing.

The I2C protocol exhibited sufficient bandwidth for connected modules but required additional configuration for stable operation with a large number of modules. The micro:bit radio protocol proved to be an effective means of providing backup wireless communication between modules. However, it was found that simultaneous use of radio and BLE is not possible, limiting its application in Bluetooth scenarios. The SPI protocol demonstrated high data transfer speeds but needed extra configuration for compatibility with various devices. This protocol is well-suited for sensors that require high-speed data transmission.

Conclusions

The study confirmed that micro bit can be effectively used to build distributed sensor networks by integrating through UART, I2C, and SPI with external sensors and more powerful platforms such as ROS2. The radio protocol provides a reliable backup communication channel, enhancing the system's reliability during failures of primary channels and facilitating interaction with modules disconnected from the central unit. However, issues of stability and compatibility when connecting a large number of devices require further investigation. Future work should focus on optimizing the performance of protocols and improving wireless communication, including support for BLE.

References

Ruch, A. (2022). Micro:bit; Cheap and Simple Hardware for Coding. In *3rd International STEM Education Conference Proceedings* (pp. 26-37). Pusula 20 Teknoloji ve Yayıncılık A.Ş. https://www.stempd.net/wp-content/uploads/2021/01/Proceedings_3rd-International-STEM-Education-Conference.pdf

Voštinár, P., & Knežník, J. (2020). Education with BBC micro:bit. International *Journal of Online and Biomedical Engineering (IJOE), 16*(14), 81–94. https://doi.org/10.3991/ijoe.v16i14.17071

Ahmad, B., Ahmed, R., Masroor, S., Mahmood, B., Hasan, S. Z. U., Jamil, M., Khan, M. T., Younas, M. T., Wahab, A., Haydar, B., Subhani, M., Khan, M. A., & Tariq, S. (2023). Evaluation of Smart Greenhouse Monitoring System using Raspberry-Pi Microcontroller for the Production of Tomato Crop. *Journal of Applied Research in Plant Sciences, 4*(01), 452–458. https://doi.org/10.38211/joarps.2023.04.01.54

Cederqvist, A.-M. (2021). Designing and coding with BBC micro:bit to solve a real-world task – a challenging movement between contexts. *Education and Information Technologies, 27*(5), 5917–5951. https://doi.org/10.1007/s10639-021-10865-w

Carreira, R., Costa, N., Ramos, J., Frazão, L., & Pereira, A. (2024). A ROS2-Based Gateway for Modular Hardware Usage in Heterogeneous Environments. *Sensors, 24*(19), 6341. https://doi.org/10.3390/s24196341

Zhuang, C. (2024). Comparison And Selection of Commonly Used Communication Protocols in Measurement and Control Instruments. *Highlights in Science, Engineering and Technology, 81*, 540–546. https://doi.org/10.54097/em4syb59