Project Semester 4 (IoT)

**Socio-Technical Analysis Report**

Project Title: < Using Agile with micro:bit Robot Robot Control >

Student Name: <XU Anbo 徐安博 Amber>

Student Id: <202283890035>

# 1. Project Outline and Objectives

This project involves the design, development, and implementation of a small, programmable robot using the micro:bit microcontroller. The primary objective is to create a functional robot that can be controlled via Python scripts to perform basic movements, including user-defined speed, direction, and path control. A key aspect of the project is the application of Agile (Scrum) methodology to manage the development process iteratively and collaboratively.

The purpose of this artefact is both educational and practical. It serves as a hands-on learning tool for understanding embedded systems programming, robotics fundamentals, and modern software development practices. A robot of this kind could be useful in STEM education to teach students programming logic, hardware integration, and the principles of automation in an engaging manner. 2. Functional Requirements

The high-level functional requirements of the system are:

1. Robot Assembly: The system shall be comprised of a physically assembled micro:bit-based robot with necessary components (motors, wheels, chassis, power supply).
2. Basic Movement: The robot shall be able to move forward and backward upon command.
3. Direction Control: The robot shall be able to turn left and right to change its direction of movement.
4. Speed Control: The user shall be able to define and adjust the robot's movement speed.

5. Path Control: The robot shall be able to follow a pre-defined path (e.g., a square, a figure-eight) based on user input.

6.Control Interface: Control commands shall be executed via a Python script running on a connected computer.

# 3. Technologies Used

Hardware:

1. BBC micro:bit V2 board
2. Micro:bit edge connector breakout board
3. Robot chassis kit (including wheels and motors)
4. L9110 or L298N Motor Driver Module
5. AA Battery Holder (for power)
6. Jumper wires

Software:

1. Python 3.x
2. MicroPython (flashable onto the micro:bit)
3. microbit Python library for serial communication
4. Git & GitHub (for version control and collaboration)

5. A code editor (e.g., Mu Editor, VS Code)

# 4. Social Analysis and Issues

In this section you should consider the broader social, human, legal and ethical issues pertaining to your project/artefact. You simply have to reflect on each of the following issues and comment briefly on their relevance to your project. If you do not think a given issue is relevant then be sure to explain why. If it is relevant, outline how the project will take appropriate account of it.

## 4.1. Privacy Issues

This project has minimal direct privacy implications. The robot in its current educational form does not collect, process, or transmit any personal data. It does not have cameras, microphones, or sensors that would allow it to identify individuals or monitor environments in a way that infringes on privacy.

However, considering a broader perspective: if this technology were scaled up for use in environments like homes or offices (e.g., as a future smart vacuum cleaner or delivery robot), privacy concerns would become highly relevant. The project currently acts as a foundational step; future iterations would need to incorporate Privacy by Design principles, ensuring any data collection is minimal, transparent, and has a legitimate purpose.

## 4.2. Data Protection Issues

Similar to privacy, data protection issues are not a primary concern for this specific prototype. The robot operates based on real-time commands and does not store any data persistently. No user data is logged or transmitted over networks.

If the project were extended to include data logging (e.g., recording paths for analytics), data protection principles would immediately apply. In such a case, the project would adhere to guidelines like GDPR by:

* **Minimization:** Only collecting data essential for the defined purpose.
* **Transparency:** Informing users about what data is collected and why.
* **Security:** Implementing measures to protect any stored data from unauthorized access.

For the current scope, these measures are not required but are important considerations for future development.

## 4.3. Intellectual Property

The intellectual property of components used is a relevant consideration.

**Software:** The project uses open-source software (Python, MicroPython) and libraries (microbit) that are freely available for educational and personal use. Their licenses (e.g., MIT, Python Software Foundation License) permit modification and distribution, provided license terms are included.

**Hardware:** The micro:bit board is designed as an open-source hardware platform for education. Other components (chassis, motor drivers) are generic, off-the-shelf parts.

Regarding the project's own IP: if this robot were to be marketed, it would be crucial to ensure that the unique code written and the specific design integration constitute our own intellectual property. The risk of infringing on existing IP is low because the project is a basic implementation for learning. However, a product like LEGO Mindstorms holds patents on its specific system integration, so a commercial product would need a unique design that does not copy protected elements.

## 4.4. Stakeholder and Risk Analysis

Stakeholders:

1. Primary: Project Team (Developers), Educators/Teachers, Fellow Students.
2. Secondary: University (as the facilitating institution), Future students who may build upon this work.

Positive Impacts/Risks:

1. For Developers: Gain valuable skills in robotics, programming, and project management. Risk: Poor time management could lead to project failure.
2. For Educators: Acquire a demonstrable example for teaching STEM concepts. Risk: If the instructions are unclear, its educational value diminishes.

Negative Impacts/Risks:

1. Physical Safety: The robot, though small, has moving parts. A minor risk exists of pinching fingers or damaging property if it moves uncontrollably.
2. Reliability: If the robot fails to function during a demonstration, it could lead to a negative learning experience.

Mitigation:

1. Minimize physical safety risks by including clear operational guidelines and supervising initial tests.
2. Maximize positive impacts by documenting the project thoroughly on GitHub, making it a reusable resource for others.
3. Ensure reliability through rigorous testing within each development sprint.

# 5. Technical Analysis and Design

In this section you should present a technical analysis and design for your project/artefact. You are free to use text, UML diagrams, and screenshots as appropriate. Keep things concise, though. Remember that the technical aspects of the system should take account of the social aspects previously considered.

## 5.1. Functional Design and Non-Functional Requirements

The system works by having a Python script on a host computer send movement commands via a serial connection to the micro:bit. The micro:bit, running a MicroPython program, interprets these commands and controls the motor driver accordingly.

* **Use Case:** A user runs a Python script (path\_control.py). The script sends a sequence of commands. The micro:bit receives these commands and sets the appropriate PWM signals to the motor driver, causing the robot to move.
* **Non-Functional Requirements:**
  + **Usability:** The control scripts should be well-commented and easy to modify for users with basic Python knowledge.
  + **Reliability:** The robot should respond correctly to commands consistently. This is addressed by iterative testing in each sprint.
  + **Maintainability:** The code is structured modularly (e.g., separate functions for move\_forward, turn\_left) and version-controlled with Git, making it easy to debug and improve.
  + **Safety:** As a social consideration, the robot's speed is capped at a safe maximum in the code to mitigate physical risks.

## 5.2. Data Requirements and Design

The data flow in this system is simple and transient.

**Data Gathered/Generated:** The only data involved are the control commands generated by the Python script.

**Data Gathering:** These commands are generated in real-time based on the script's logic and user input.

**Data Storage/Representation:** **No data is stored persistently.** Commands are sent as simple ASCII strings over the USB serial connection. They exist only in the computer's memory while the script is running and in the micro:bit's memory for the brief moment they are being executed. This ephemeral nature aligns with the data protection principles discussed earlier, as there is no long-term data retention.

# 6. Professional Conduct and Ethics

If this project were developed into a marketable product, ensuring high ethical standards would be paramount. This would involve:

1. **Safety First:** Conducting thorough safety testing to prevent any physical harm to users or damage to property. This would include implementing hardware failsafes and software limits.
2. **Transparency:** Being completely transparent about the capabilities and limitations of the robot. For example, clearly stating that it is an educational tool and not a industrial-grade device.
3. **Privacy by Design:** If future versions included data collection features, these would be designed with privacy as a core principle, not an afterthought. Users would have clear opt-in controls.
4. **Accessibility:** striving to make the product and its documentation accessible to a diverse audience, including those with different learning styles or physical abilities.
5. **Environmental Responsibility:** Considering the environmental impact of production and packaging, aiming to use sustainable materials and promoting recyclability.

Adhering to these principles would not only build trust with customers but also ensure the product contributes positively to society.