YRKESHÖGSKOLAN
NOVIA

# Maze Solving Quadruped Robot

A Study Using Microbit XGO Robot V2

Gregory Echekwube

Degree Thesis

Thesis for a Bachelor's (UAS) - degree

Information Technology

Vaasa 2025

**DEGREE THESIS**

Author: Gregory Ojotule Echekwube
Degree Programme and place of study: Information Technology, Vaasa
Supervisor(s): Hans Linden and Kaj Wikman

Title: Maze Solving Quadruped Robot

---

Date: 14.05.2025 Number of pages: 32 Appendices: 2

---

**Abstract**

Maze-solving has long been acknowledged as a fundamental robotics challenge, serving both a practical and symbolic assessment of a robot's intelligence and autonomy. Navigating through unfamiliar surroundings with obstacles has been used as a foundation for assessing fundamental robotic skills in both older mechanical systems and today's intelligent machines. This makes maze-solving more than just an academic exercise; it reflects real-world problems in fields like planetary exploration, autonomous cars, autonomous food delivery, warehouse automation, and search and rescue missions.

This thesis investigates autonomous navigation within a maze environment with a Micro:bit XGO Robot Kit V2 by evaluating various ranging sensors. Different time-of-flight (ToF) laser and ultrasonic modules were evaluated. The Micro:bit XGO Robot Kit V2 is a quadruped robot dog that serves as an educational platform with potential for advanced robotics applications.

The methodology involves testing performance through detailed case studies and an experimental framework. The thesis also evaluates the educational benefits of this project for bachelor's-level students, emphasizing skills in programming, robotics, and sensor integration.

Recommendations for future alternative high-performance sensors, such as motion tracking sensors that utilize 3-axis accelerometers and gyroscopes, as well as replacing single-direction ranging sensors with more advanced 2D and 3D scanning LIDAR sensors.

This work contributes to the field of educational robotics by demonstrating how cost-effective tools can be utilized for complex tasks, providing a scalable model for teaching robotics principles.

---

Language: English
Key Words: obstacle avoidance, maze navigation, sensor integration, robotic applications

# Table of Contents

## List Of Figures

# 1 Introduction

This thesis investigates the autonomous navigation of a maze environment using a Micro:bit XGO Robot Dog V2. This was done by integrating various sensors to the robot dog to enable it to perceive the environment. The micro:bit microcontroller on the robotic dog reads sensor inputs and controls the robot's movements. The robot's ability to make intelligent navigation decisions was driven by the logic programmed into the microcontroller.

## 1.1 Background

Maze solving has long been recognized as a foundational challenge in the field of robotics, representing both a symbolic and practical test of a robot's autonomy and intelligence. From older mechanical systems to today's intelligent machines, navigating through unknown environments with obstacles has served as a base for evaluating core robotic capabilities. These include autonomous navigation, real-time decision-making, sensor integration, and path optimization. Maze-solving is not just an academic exercise, it mirrors real-world challenges in domains such as autonomous vehicles, warehouse automation, service industries, industrial inspection, search-and-rescue operations, and planetary exploration.

Robotics, as a cornerstone of modern technological advancement, has permeated industries from manufacturing to healthcare. Of the numerous challenges in this field, autonomous navigation remains one of the most significant. Maze navigation and obstacle avoidance encapsulate many of the technical difficulties encountered in real-world applications. These include precise sensing, environment mapping, obstacle detection, and real-time processing- all of which are necessary for intelligent and adaptive behavior in complex and dynamic settings. In order to find the existing routes from an initial location to a target point, the maze-solving robot often needs to navigate a maze (Alamri et al., 2023).

Four-legged robots have enormous promise for several uses, such as disaster relief, military reconnaissance, disaster response, industrial inspection, agricultural operations, and service industries (Q. Li et al., 2025). For example, they can navigate through earthquake

rubble to search for survivors or inspect equipment in intricate industrial settings (Q. Li et al., 2025). These scenarios demand high levels of stability, flexibility, and autonomy from robots (Q. Li et al., 2025). Quadruped robots are perfect for rough terrain and heavily forested areas, where wheeled robots find it difficult to maneuver. This is because they can readily climb stairs and low barriers, unlike wheeled robots (Chen et al., 2024). The COVID-19 pandemic presented previously unheard-of difficulties for healthcare systems around the world, making it more and more obvious that robots may help in a variety of ways. By carrying out duties that often call for intimate human interaction, like delivering medication, cleaning surfaces, and even conducting remote consultations, robots can help lower the danger of viral transmission (Nguyen, n.d.).

## 1.2 Why Quadruped Robots?

There have been a lot of changes in quadruped robots since their existence. This is because there have been improvements in control technologies, materials science and computational capabilities(Q. Li et al., 2025).

### 1.2.1 Comparing Quadrupeds with Other Legged Systems

Quadruped robots represent one of several architectural paradigms in the field of legged robotics, each having unique benefits and drawbacks. It is crucial to compare quadruped robots to other legged configurations, including bipeds, hexapods, and multilegged robots, even if the recent explosion in research and commercialization of quadrupedal systems shows a promising balance between complexity, mobility, and adaptability (Q. Li et al., 2025).

Bipedal robots are based on human movements and therefore offer unparalleled promise in human-designed environments, with uses including stair climbing, tool use, and confined space navigation (Q. Li et al., 2025). However, significant challenges arise with respect to maintaining dynamic stability and energy efficiency, requiring sophisticated control algorithms and computationally intensive processes to maintain balance (Q. Li et al., 2025).

Conversely, hexapods and other multi-legged robots benefit from their natural attribute of static stability and redundancy. Their capability to stay supported on several legs, even when encountering failures or uneven ground, makes them exceptionally reliable for uses

like biological studies, space exploration, and search-and-rescue operations (Q. Li et al., 2025). However, their usage in time-constrained or quickly changing situations is hampered because they have increased mechanical and computational sophistication (Q. Li et al., 2025).

Between these two extremes, quadruped robots achieve a workable balance. They move more effectively on uneven terrain than hexapods, perform better dynamically, and have greater balance than bipedal robots. Their quick adoption in both commercial applications and academic research can be attributed to their stability. Although long-term dominance is difficult to forecast, present patterns indicate that quadrupeds are a basic platform for future robotic advancement rather than merely a fad. Although the development of hybrid or reconfigurable morphologies may change the field of legged robotics, additional advancements in actuation, sensing, and adaptive control may solidify their position (Q. Li et al., 2025).

## 1.3 Problem Statement

Achieving autonomous navigation is still a key difficulty in robotics, particularly when operating in dynamic and complex environments. A critical aspect of this challenge is enabling robots to detect and avoid obstacles while effectively navigating through complex environments. Traditional path-planning algorithms often struggle with real-time decision-making in unknown or partially known spaces, leading to inefficiencies, collisions, or failures in navigation tasks.

The goal of this thesis is to integrate sensors with the micro:bit XGO Robot Kit V2, a quadruped robot and powered by the BBC micro:bit microcontroller. This robot provides a simple, engaging, and flexible platform for exploring advanced robotics concepts. Despite its small size and low cost, the XGO V2 allows students to connect a variety of sensors using interfaces such as GPIO (General Purpose Input/Output), I2C (Inter-Integrated Circuit) and UART (Universal Asynchronous Receiver-Transmitter) communication bus.

## 1.4 Objectives

The primary objective of this thesis is to design, implement, and evaluate an autonomous maze-solving and obstacle-avoiding robot using the Micro:bit XGO V2 Dog Robot Kit,

enhanced with sensors. The system aims to enable the robot to navigate both static and dynamic mazes with improved spatial awareness, decision-making, and adaptability. To achieve this, the following specific objectives are defined:

- Integrate four key sensors, two different ultrasonic time-of-flight sensors (HC-SR04 and Parallax Ping), one laser time-of-flight sensor (VL53L0X) and one single-direction LiDAR senor (TF-Luna). I2C, GPIO, and UART are utilized for communication with the sensors.
- Analyze and document the functionality and limitations of each sensor, focusing on their role in real-time environmental perception and navigation.
- Design and implement obstacle detection and avoidance, and maze-solving algorithms optimized to run within the micro:bit's constrained processing environment using real-time sensor utilization.
- Design and construct mazes to test the robot's navigation performance.
- Identify hardware and software bottlenecks.

### 1.5 Significance of the Study

The research holds significant educational value by providing a practical framework for students to explore robotics concepts, sensor integration, and algorithm development on an accessible platform. Also, by tackling the foundational challenge of maze-solving using sensors, this thesis contributes to the broader field of robotics, offering insights into multi-sensor systems and algorithmic flexibility that can be applied to more complex real-world autonomous navigation challenges.

## 2 Technical Overview

This technical overview covers the components and systems that are essential to this project. It includes a detailed examination of the Micro:bit XGO Robot Kit V2 and its integration with the Micro:bit board, the capabilities of the microcontroller, and the

expansion provided by the XGO Adapter. It also outlines the coding platforms and evaluates the selected sensors, and the communication methods used to interface the sensors with the robot.

## 2.1 The Robot

### 2.1.1 The Micro:bit XGO Robot Kit V2

The XGO Robot Kit V2, made by ELECFREAKS, is a small quadruped robot created for learning and exploring robotics (*Index | LEARN*, n.d.). It moves like an animal using 12 small motors (3 per leg), allowing it to walk in all directions. It is distinguished by a sophisticated mechanical structure featuring 15 degrees of freedom, allowing for a wide range of dynamic and lifelike movements, mimicking the agility of a real pet dog (*ELECFREAKS XGO Robot Dog Kit V2 - Advanced STEM Learning*, n.d.). It also has a small robotic arm with three more motors for simple tasks like grabbing or moving things. (*Index | LEARN*, n.d.)

The robot's design incorporates an aluminum alloy chassis, providing a robust yet lightweight platform for experimentation. Notably, it features an integrated 6-axis Inertial Measurement Unit (IMU) and a built-in speaker, enabling functionalities such as posture feedback for self-stabilization and the provision of auditory cues (*Micro*, n.d.-b). The robotic arm is equipped with three axes, capable of performing actions like gripping, pushing, and carrying small objects, thereby expanding the scope of potential tasks and interactions.

The platform is designed for seamless integration with the Micro:bit microcontroller, facilitating graphical programming through the MakeCode environment, which is particularly suitable for individuals with limited or no prior coding experience. Furthermore, the robot includes an exclusive expansion board, the XGO Adapter, that extends the Micro:bit's capabilities by providing additional GPIO and I2C interfaces, enabling the integration of a wide range of external sensors and modules (*Hardware Introduction | LEARN*, n.d.).

The robot is powered by a built-in rechargeable battery, offering a substantial operational time of approximately 120 minutes on a single charge. It takes about 200 minutes to get a full charge (*ELECFREAKS XGO Robot Dog Kit V2 - Advanced STEM Learning*, n.d.).

The Micro:bit XGO Robot Dog V2 serves as an ideal platform for investigating the concepts of autonomous robotics and sensor integration in an educational setting because of its sophisticated capabilities, easy-to-use programming, and sturdy design.

The robot is controlled by a microcontroller, micro:bit, which interfaces with it via the XGO Adapter. This adapter enable communication between the micro:bit, the robot's motors and connected sensors.

The dog comes pre-programmed with 19 basic movements (such as sitting or shaking hands), making initial programming of the robot easier.


Figure 1: Micro:bit XGO Robot V2(*Hardware Introduction | LEARN*, n.d.)

### 2.1.2 The Micro:bit Board

The British Broadcasting Corporation (BBC) created the BBC micro:bit microcontroller to aid students in learning electronics and programming. A 5x5 LED matrix, a 32-bit ARM Cortex-M0 processor, plus additional sensors and interfaces such as accelerometer, magnetometer, Bluetooth, and a USB connector are all included. (*Hardware Introduction | LEARN*, n.d.).

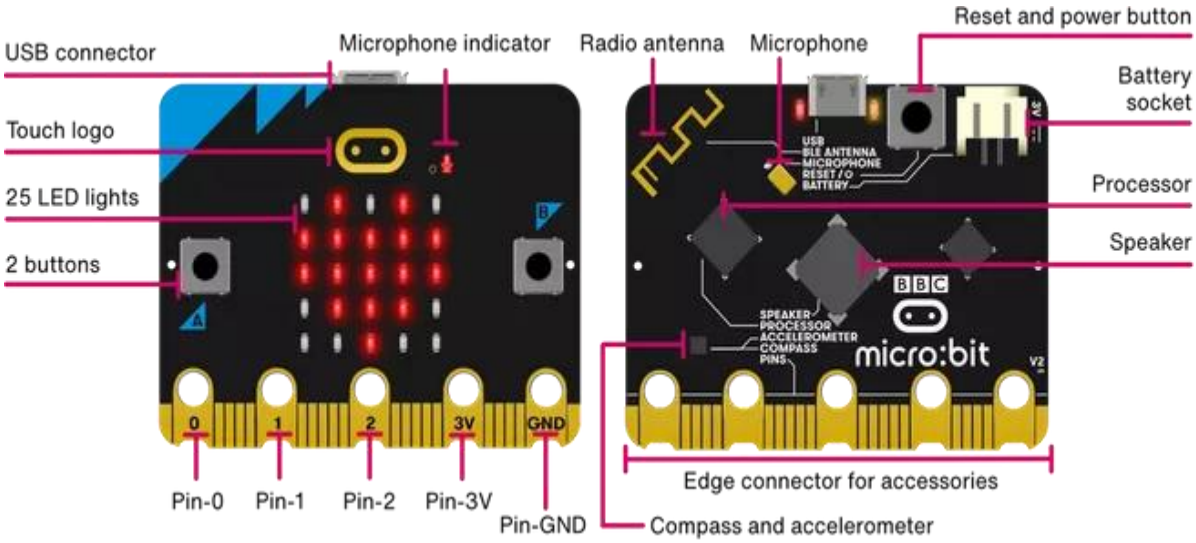



Figure 2: The Micro:bit Board (*Hardware Introduction | LEARN*, n.d.)

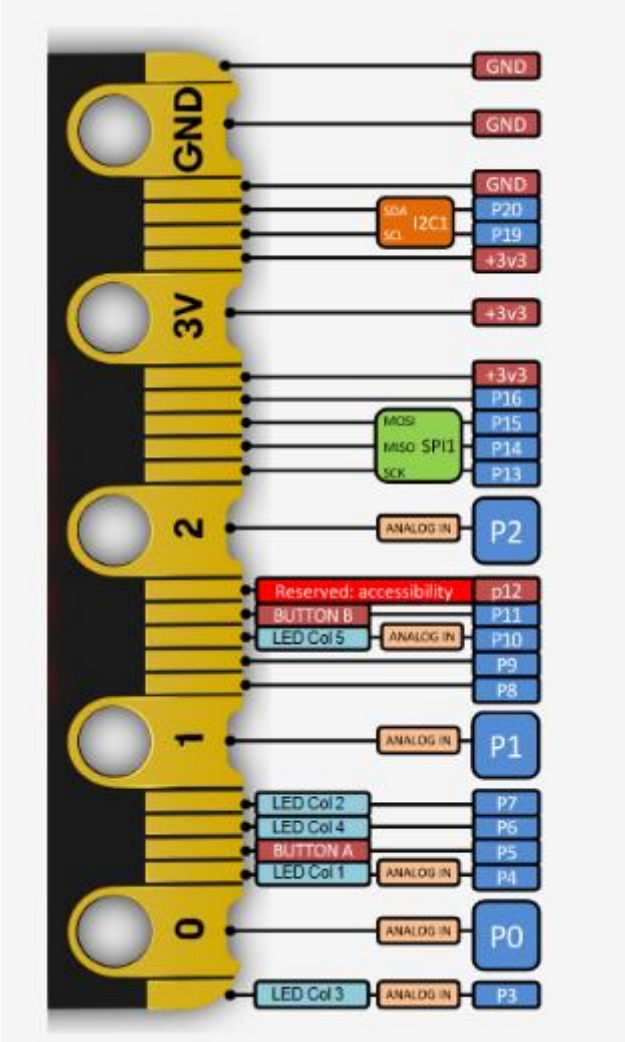



Figure 3: Micro:bit Board Connectors *(Micro, n.d.-a)*

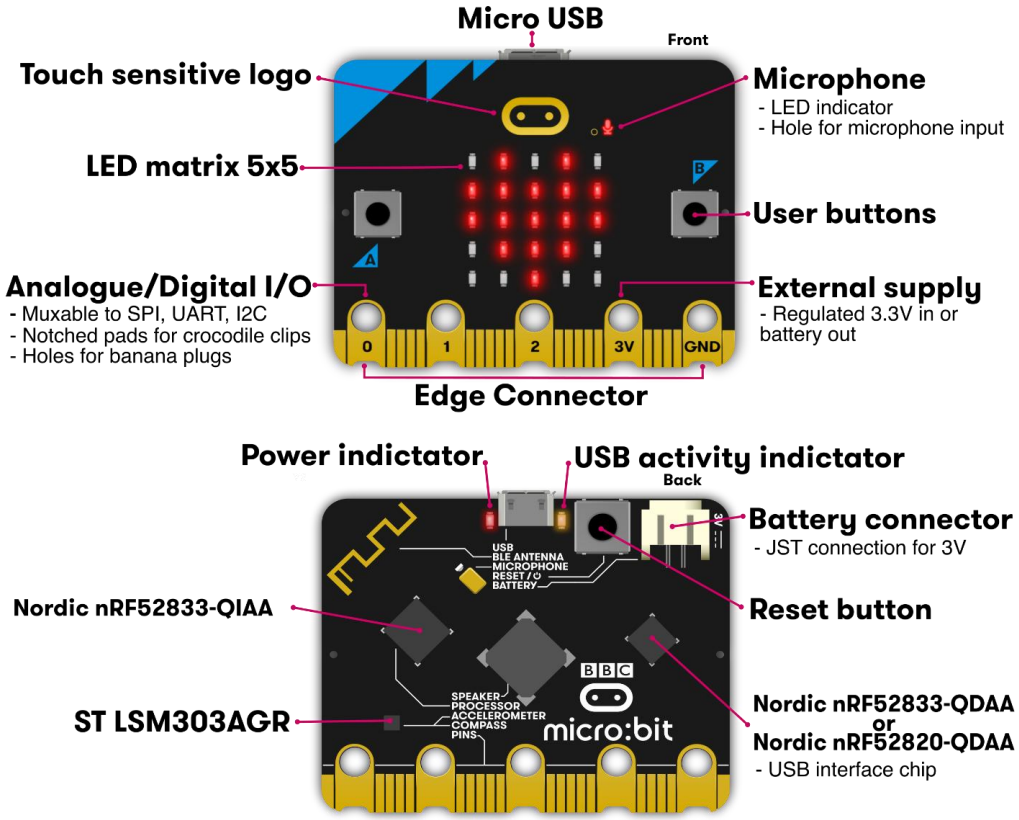



Figure 4: Micro:bit Hardware Block Diagram (*Hardware*, n.d.)

The BBC micro:bit's onboard motion sensors are integrated within the LSM303AGR chip, which combines a 3-axis accelerometer and a 3-axis magnetometer *(STMicroelectronics, 2022)*. This chip is soldered directly onto the micro:bit's printed circuit board (PCB).

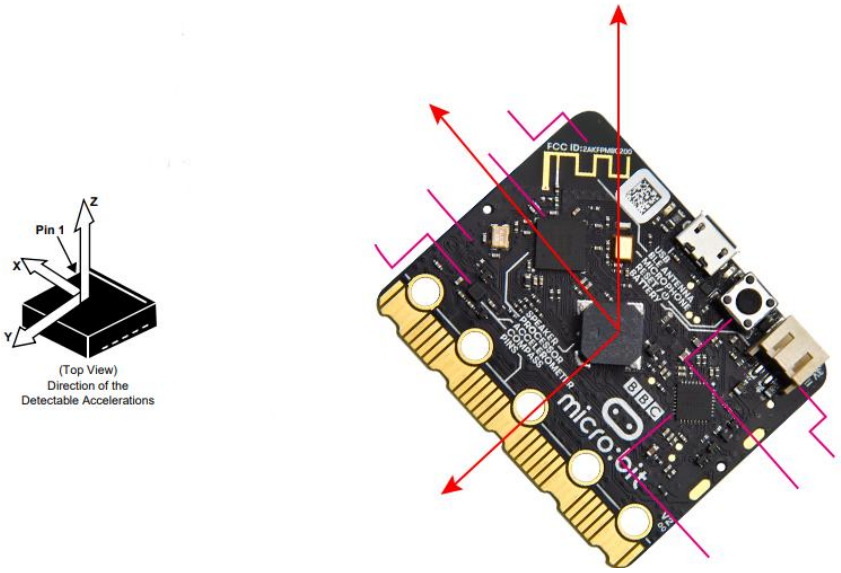



Figure 5: Micro:bit Accelerometer (*How to Use Accelerometer Sensor with Micro:Bit?*, n.d.)

### 2.1.3 The Micro:bit - The Robot's Brain

The **micro:bit** is the microcontroller that controls the robot. It incorporates the nRF51822 SoC from Nordic Semiconductor, which is based on an ARM Cortex®-M0 processor, alongside an NXP Kinetis KL26 SoC powered by a Cortex-M0+ core (Ltd, n.d.). The first processor (nRF51822) is dedicated solely to BLE (Bluetooth Low Energy) wireless communication. The second processor (KL26) is used for the robot's execution. In short, it reads sensor data, executes the programmed code, and controls the robot's movement.

The micro:bit has the following features:

- It uses a compact yet powerful processor and sufficient memory to execute programs written in MicroPython.
- It includes pins that enable connectivity to sensors and motors using different types of connections- general digital I/O pins and protocols such as I2C and UART.
- It operates at 3.3V and is powered by the robot's main battery.
- It has built-in motion sensors.

Its program is executed about 10 times per second. However, due to the micro:bit's limited power, execution is slower when multiple sensors are used, potentially causing latency and minor inaccuracies in the robot's movements.

Despite its limitations, the micro:bit is a great learning tool for students to learn how to write efficient code and make the most of limited hardware.

### 2.1.4 XGO Adapter Expansion Board

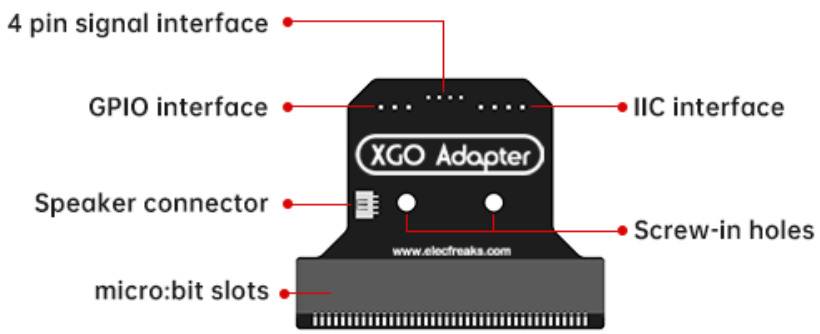



Figure 6: XGO Adapter (*Hardware Introduction | LEARN*, n.d.)

The XGO Adapter, which is an expansion board, can be used with the micro:bit to give users access to additional functionalities and interfaces. The expansion board is attached to the XGO-lite2 front cover using screws, for practical and secure fastening.

It consists of the following components:

1. 4-wire connection port: This interface is used to connect the 4-wire cable of XGO-lite V2, which not only supplies power to the micro:bit motherboard, but also enables command transmission between micro:bit and XGO-lite V2.
2. Speaker connection port: This port allows a speaker to be connected, enabling sounds.
3. GPIO and IIC interface: These interfaces can be used to connect various sensors and modules.

### 2.1.5 Code Platform

To write and upload code to the micro:bit, the following are required:

- **MakeCode (online)** for writing no-code blocks, Javascript or MicroPython(https://makecode.microbit.org/)
- **Micro:bit Python editor**(https://python.microbit.org/)
- **Mu Editor** (download from https://codewith.mu/ and install on computer)

## 2.2 Sensors

To enable the Microbit XGO Robot Kit V2 to autonomously navigate mazes and avoid obstacles, multiple external sensors will be integrated with the robot. These sensors provide distance, orientation, and environmental awareness from multiple directions.

These four sensors were evaluated:

- **HC-SR04**: Uses ultrasound to detect nearby objects.
- **Parallax PING**: Another ultrasound sensor.
- **VL53L0X**: Measures short distances with great accuracy using infrared light.
- **TF-Luna**: Measures longer distances using LIDAR, useful for detecting distant walls or objects.

### 2.2.1 HC-SR04

The HC-SR04 is a widely used ultrasonic ranging sensor that operates by emitting an ultrasonic pulse at a frequency of 40 kHz. It then measures the time for the echo to return after reflecting off an object. By knowing the speed of sound, the distance to the object can be calculated. The HC-SR04 typically offers a detection range from 2 cm to 400 cm with an accuracy of approximately 3 mm and an effective measuring angle of less than 15 degrees (Sparkfun, n.d.). This sensor operates on a 5V power supply, and while the trigger signal can be 3V or 5V, the echo is a 5V logical signal (Adafruit Industries, n.d.). The HC-SR04 uses digital signals for both input and output. The pulse width of the output signal is proportional to the distance to the object.

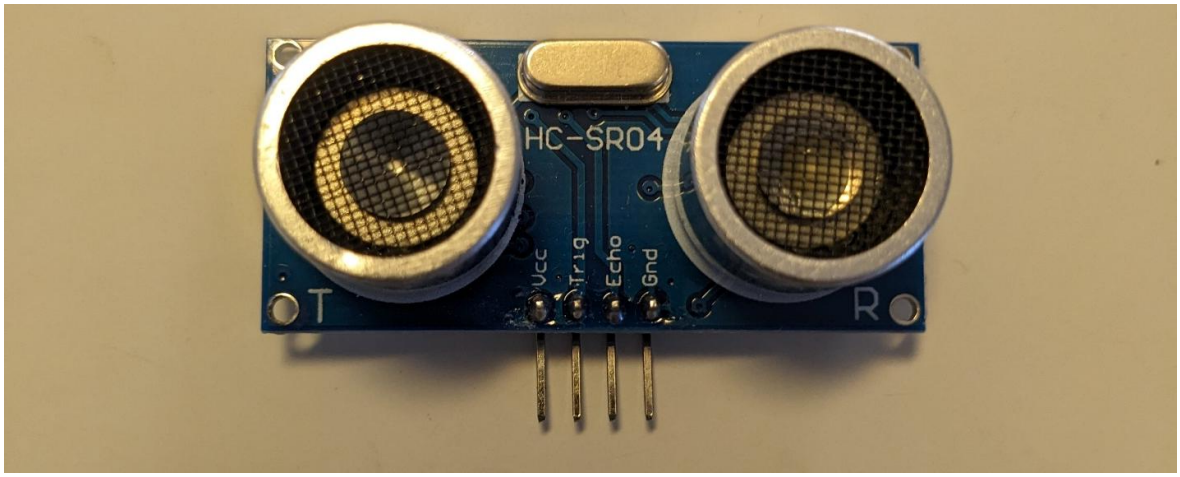



Figure 7: HC-SR04 (Photograph by the author, 2025)

### 2.2.2 Parallax PING

The Parallax PING ultrasonic distance sensor functions similarly to the HC-SR04, using ultrasound to measure distance. It transmits a short 40 kHz ultrasonic burst and then listens for the echo (Pololu, n.d.). The sensor output pulse width corresponds to the time taken for the echo to return, allowing for distance calculation. The PING sensor typically has a range of 2 cm to 3 meters with a trigger input of a positive TTL pulse and an echo pulse that varies in width depending on the distance (Lynxmotion Wiki, n.d.). It operates on a 5VDC supply and has a narrow acceptance angle. The output is a digital pulse.

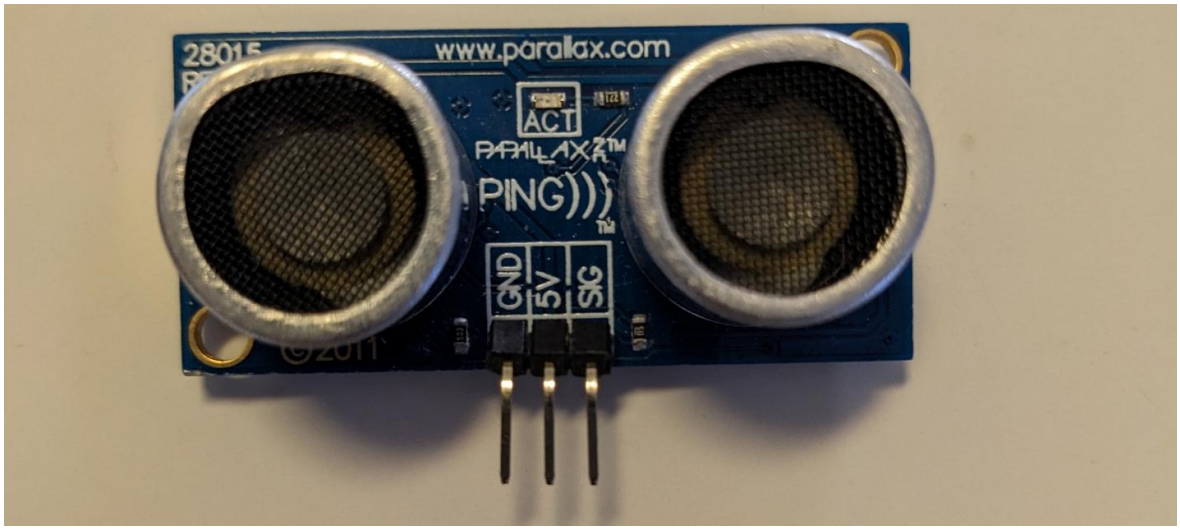



Figure 8: Parallax PING (Photograph by the author, 2025)

### 2.2.3 VL53L0X

STMicroelectronics manufactures the Time-of-Flight (ToF) high-accuracy proximity sensor VL53L0X. In contrast to ultrasonic sensors, it measures the time it takes for an infrared light pulse to reach an item and return to the sensor via an invisible 940 nm Class 1 laser emitter. Up to two meters of distance can be measured with high accuracy thanks to this technology. It is a digital sensor powered by a 2.6 to 5.5 V supply and communicates via the I2C protocol (*Pololu - VL53L0X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 200cm Max*, n.d.).


Figure 9: VL53L0X (*VL53L0X - Time-of-Flight (ToF) Ranging Sensor - STMicroelectronics*, n.d.)

### 2.2.4 TF_LUNA

The TF_Luna is a single-point ranging LiDAR module based on the Time-of-Flight principle. It is similar to the VL53L0X but operates at a different wavelength (850nm). It is designed

for stable, accurate, and high-frame-rate range detection, with a typical operating range of 0.2m to 8m for objects with 90% reflectivity. The accuracy is reported as ±6cm at distances from 0.2m to 3m, and ±2% at distances from 3m to 8m. The TF_Luna has a small field of view of 2 degrees and can achieve frame rates from 1 to 250 Hz. It is a digital sensor that supports both UART and I2C communication interfaces and operates on a supply voltage of 3.7V to 5.2V. The TF_Luna is known for its compact size, lightweight design, and low power consumption (*TF-Luna Datasheet | PDF | Lidar | Equipment*, n.d.).

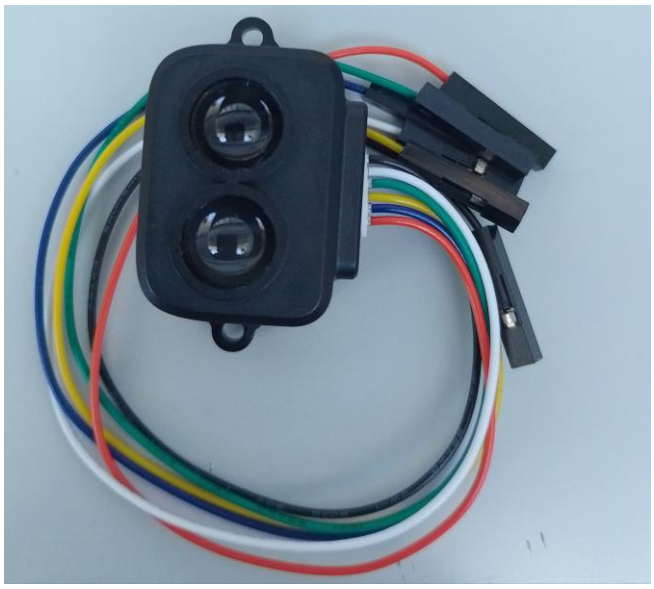


Figure 10: TF_Luna - Front view (Photograph by the author, 2025)

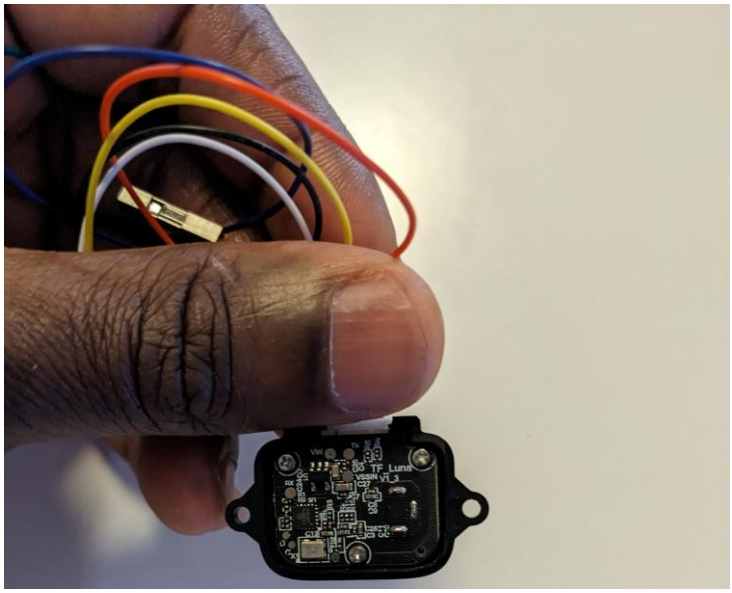


Figure 11: TF_Luna - Rear view (Photograph by the author, 2025)

## 2.3 Sensor Communication Methods

### 2.3.1 I2C (Inter-Integrated Circuit)

I2C is a simple two-wire protocol that enables the micro:bit to communicate with multiple sensors using only two pins. It is slower than some other options. It uses SDA (Serial Data Line) and SCL (Serial Clock Line) lines. This enables communication between numerous sensors and a single master, which in this case is the micro:bit. To communicate with each sensor, every device on the I2C bus has a unique address. Imagine the I2C bus as a classroom in which a single teacher (master) refers to each student (sensor) by name (address).

### 2.3.2 GPIO (General Purpose Input/Output)

GPIO pins are general-purpose digital input/output pins that can be configured to send or receive digital signals. A sensor typically requires multiple GPIO pins, whereas I2C uses only two pins to connect all sensors on the bus.

### 2.3.3 UART (Universal Asynchronous Receiver/Transmitter)

UART is a serial communication protocol that uses two wires: transmit (TX) and receive (RX). It is capable of interacting with only one device at any given time. It is just like walkie-talkies: only one sensor can communicate with the micro:bit at a time.

### 2.3.4 SPI (Serial Peripheral Interface)

SPI is another widely used electronics communication protocol that allows synchronous, high speed data transfer between a master device and one or more slave devices (Ibraheem & Adrees, 2023). SPI usually requires four wires. For certain applications, this allows for more flexibility and faster communication.

# 3 I2C

This section provides more detail about the I2C (Inter-Integrated Circuit) protocol. It covers how it works, its application in this project, its key features and practical considerations such as resolving addressing conflicts when using multiple I2C devices.

## 3.1 What is I2C?

The Inter-Integrated Circuit (I2C), or IIC, is a popular communication protocol used in digital electronics. I2C was created in the early 1980s by Philips Semiconductors (now NXP Semiconductors) to make it easy for chips on the same circuit board to communicate with each other. This innovative two-wire system changed the way devices communicate and is nowadays used in many digital electronics applications, including LCD screens (Celeste, 2024).

It is designed to allow multiple devices to communicate with each other over a two-wire interface and is commonly employed in embedded systems. Most microcontrollers have built-in support for I2C communication, including the BBC micro:bit. It is used to connect low-speed peripheral devices (like sensors) to a microcontroller. It is popular because of its simplicity and efficiency.

I2C enables a master device (e.g., the micro:bit) to exchange data with one or more slave devices using two wires:

1. SDA (Serial Data Line): Carries the data being transmitted between devices.
2. SCL (Serial Clock Line): Provides the clock signal generated by the master to synchronize data transfer.

In addition to these two lines, devices typically require power (VCC, often 3.3V or 5V) and ground (GND) connections, making I2C a four-pin interface in practice.

## 3.2 Key Features

1. Master-Slave Architecture: The master (micro:bit) initiates and controls communication, while slaves (sensors) respond. Multiple slaves can share the same bus, each identified by a unique 7-bit or 10-bit address.
2. Two-Wire Simplicity: Reduces wiring complexity compared to protocols like SPI, which uses four or more lines.
3. Multi-Device Support: Up to 128 devices (with 7-bit addressing) can connect to the same I2C bus, as long as their addresses do not conflict.
4. Speed: Standard speeds include 100 kHz (original), 400 kHz (fast mode), and up to 1 MHz or more in newer variants, though the micro:bit typically operates at 100–400 kHz.
5. Bidirectional Data Transfer: Data can flow from master to slave, like configuration commands, or slave to master, like sensor readings.
6. Pull-Up Resistors: SDA and SCL need pull-up resistors (e.g., 4.7kΩ to VCC) to ensure proper signaling. These resistors are often built into breakout boards.

## 3.3 Application in this project

I2C is the only communication interface available for the VL53L0X and the TF_Luna. Both sensors can connect to the same SDA (Pin 20) and SCL (Pin 19) lines on the micro:bit. The sensors use distinct addresses to avoid conflicts.

Advantages:

- Simplicity: Fewer pins and wires simplify hardware setup on the XGO V2 and is fitting to XGO's compact design
- Expandability: Additional I2C sensors can be added without rewiring, as long as addresses are unique.
- Compatibility: micro:bit's I2C support is built-in making it a natural choice.

Limitations:

- Speed: Slower than alternatives like SPI (e.g., 400 kHz vs. 10 MHz), which may cause delays with multiple sensors.
- Bus Contention: Address conflicts or noise can occur if not managed properly.
- Pull-Up Resistors: External resistors (typically 4.7 kΩ) are needed on SDA and SCL to ensure reliable signaling, though the micro:bit includes internal pull-ups.

## 3.4 Resolving I2C address conflicts

The I2C slave device address is one of the most crucial elements of the I2C protocol because of the "one master, multiple slaves" connection between devices on the I2C bus (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). Every slave device linked to the I2C communication protocol must have a distinct 7-bit or 10-bit address (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). Nevertheless, a lot of manufactured I2C devices have fixed addresses or configurable limitations. Therefore, when multiple devices with the same address are linked to the same I2C, communication problems arise (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). The inability of the master controller to distinguish between them causes data misinterpretation or breakdowns in communication (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.).

Resolving these address clashes is possible by implementing hardware or software methods.

### 3.4.1 Hardware Solutions

**1. I2C Multiplexer (I2C MUX)**

A hardware device, known as an I2C multiplexer, manages address conflicts by linking various slave devices to the master controller through multiplexing (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). By choosing different channels, the multiplexer allows the master controller to communicate with devices that share the same address (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). To prevent conflicts, each channel can be linked to a slave device separately. This approach is particularly effective in scenarios such as multi-sensor setups, where several devices might

use the same address. The Gravity: Digital 1-to-8 I2C Multiplexer exemplifies this by allowing multiple devices with identical addresses to be connected on a single I2C bus through cascading, thus avoiding address clashes (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.).

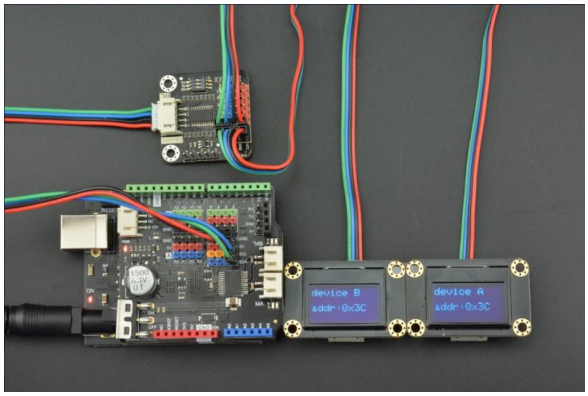



Figure 12: Two OLED displays on the same I2C bus using an I2C multiplexer (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.)

**2. Device Address Modification**

Some I2C devices allow their addresses to be modified either through software commands or by adjusting hardware pins. This enables developers to assign unique addresses by sending particular signals or setting pin configurations. An example of a device that utilizes hardware pins for this purpose is the Gravity: BMM150 Triple Axis Magnetometer Sensor. It is crucial to record the updated address to prevent complications during later debugging (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.).


Figure 13: Gravity: BMM150 Triple Axis Magnetometer Sensor (*Gravity*, n.d.)

**3. Using Multiple I2C Buses**

Some master controllers support multiple I2C interfaces. For example, the Arduino Due features two I2C ports (SDA/SCL and SDA1/SCL1). Likewise, the Raspberry Pi 4 Model B and

the latest Raspberry Pi 5 Model B can be configured to operate multiple I2C buses (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). The STM32 microcontroller series, including models like the STM32F103 and STM32F407, also offer support for several independent I2C interfaces. By using separate I2C buses, devices sharing the same address can communicate without causing address conflicts (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.).

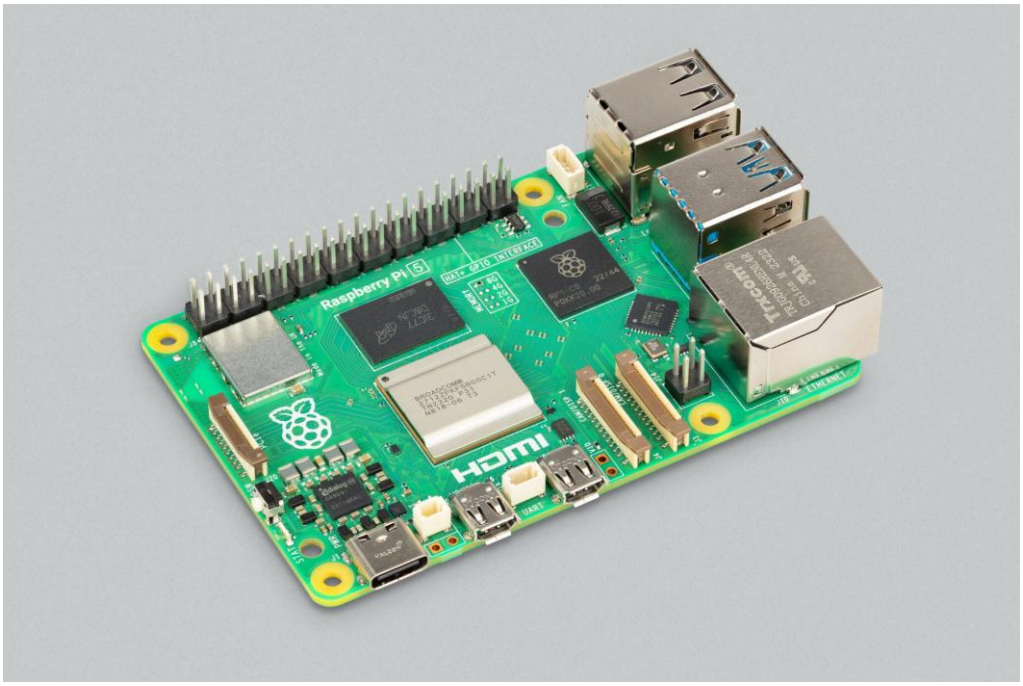



Figure 14: Raspberry Pi 5 Single Board Computer(*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.)

### 3.4.2 Software Solutions

Software solutions are flexible and more complex. They are used when hardware options are unable to resolve conflicting addresses.

**1. Timing Control and Sequential Device Activation**

Different slave devices can be turned on at different times by the master controller using timing control (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). For instance, it can prevent conflicts by regulating the I2C bus's power supply or the devices' enable pins to guarantee that only one device is active at a time (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). This method works well in situations where communication frequency is lower. By employing software to successively activate various channels, the Texas Instruments TCA9548A I2C Multiplexer can accomplish time control.

The TCA9548A is an eight-channel, bidirectional translating I2C switch. Eight target device channels, SC0/SD0-SC7/SD7, receive the controller's SCL/SDA signal pair. Both a single downstream channel and any combination of the eight channels may be chosen. Conflicts between I2C target addresses can be resolved via these downstream channels (*TCA9548A Data Sheet, Product Information and Support | TI.Com*, n.d.).

**2. Dynamic Address Management**

The master controller can dynamically modify the device's address prior to every transmission, guaranteeing uniqueness, for devices that allow dynamic address changes while in use (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). Despite its flexibility, this approach is somewhat difficult to execute, requiring stringent time control and error-handling procedures. One device that has this ability is the Analog Devices LTC4316 I2C Address Translator (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). This approach is ideal for embedded systems with high complexity (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.). It should be noted that dynamic address change capability is still limited to a small selection of devices on the market (*How to Resolve I2C Address Conflicts in Embedded Systems*, n.d.).

# 4 Methodology

This section outlines the methodology used to implement autonomous maze navigation using the Micro:bit XGO Robot Dog V2. It starts by detailing the system architecture, robot movement test, experimental setup, sensor integration and sensor tests. The final part explains the navigation algorithm which autonomously guides the robot’s movement based on real-time obstacle detection.

## 4.1 System Architecture

The architecture comprises three layers: sensing, processing, and actuation, each leveraging the robot’s hardware and connected sensors to achieve obstacle avoidance and spatial awareness.

1. **Sensing Layer**: Spaces and nearby obstacles are detected by the sensors. The data from the sensor is sent to the micro:bit through its communication channel.
2. **Processing Layer**: The micro:bit, as the central "brain", processes the data and controls the robot according to the programmed instructions that was uploaded to it.
3. **Actuation Layer**: Servo commands, relayed through the XGO Adapter, adjusts the robot for motion.

## 4.2 Movement Test

Before starting coding and sensor integration, it is essential to properly set up the XGO Robot Kit V2. The guide for this is available in Appendix 1.

The communication between micro:bit and XGO-lite V2 is transmitted through the serial port, therefore, you need to set the TX (send) and RX (receive) serial port numbers. XGO-lite V2 defaults to TX-14 and RX-13 (*MakeCode Graphical Programming | LEARN*, n.d.). The code shown below is for the robot to perform actions at the pressing of respective buttons.

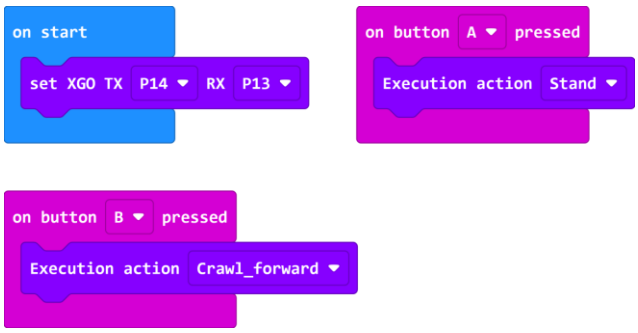



Figure 15: MakeCode Blocks (*Xgo-Lite2*, n.d.)


Figure 16: JavaScript (*Xgo-Lite2*, n.d.)

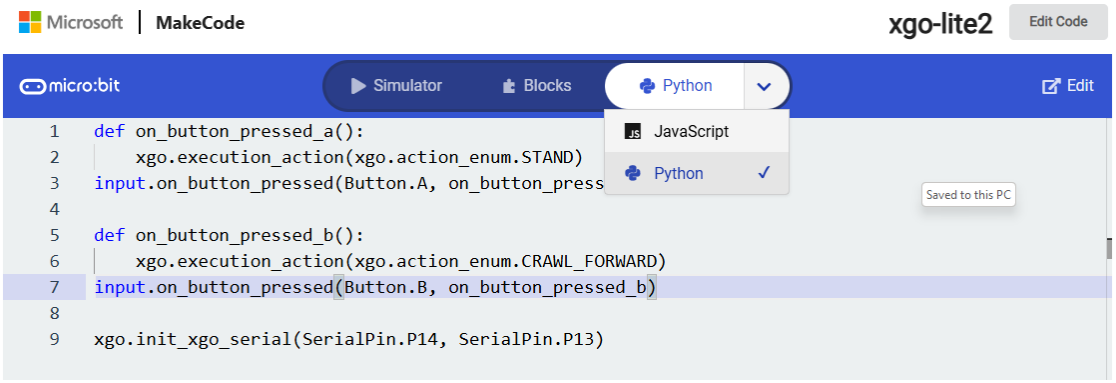



Figure 17: MicroPython (*Xgo-Lite2*, n.d.)

To upload code to the micro:bit, it is removed from its slot in the XGO Adapter on the robot. It is now connected via USB cable to a computer. The code is then downloaded to it, and it is now inserted back in its slot. The robot is then powered on.

If successful, when button A is pressed, the robot will stand. When button B is pressed, the robot will crawl forward. Always place the robot on a stable surface before testing motion.

It should be possible to download code to the micro:bit while still connected to the robot and powered off. Unfortunately, this can sometimes generate problems. Once, it created an error situation on the robot where all the lights stayed on and nothing worked.

A power cycle was required to reset the system by opening the robot and unplugging the power for about a minute to reset the system. The robot returned to normal operation after that. Therefore, it is recommended to upload the code when the microbit is not connected to the robot.

The robot can also be controlled by an Android app which communicates with the robot through Bluetooth.

## 4.3 Experimental Setup

The primary platform for this research is the Micro:bit XGO Robot V2 Dog. Four sensors were initially considered for integration- the VL53L0X, TF_Luna, HC-SR04 and the Parallax PING. The HC-SR04, Parallax PING, and the TF_LUNA sensors could not work with the XGO

Robot because of power constraints. They require 5V for operation while the micro:bit has a 3.3V voltage limit. Because of this, only the VL53L0X was used in the final system.

The initial sensor that was started with was the VL53L4CD sensor. Several attempts were made to make it communicate with the the Micro:bit but it did not work. The extension that was available on the MakeCode website was for VL53L0X. The sensor was then changed to the VL53L0X, and it worked.

The maze environment used for testing was a basic U-maze. It was constructed using wooden boards to create a confined path with two 90-degree turns. The dimensions of the maze were approximately 0.4m and 0.9m for the U branches, and 0.8m for the base path width. The path length was about 1.85m at the base, ensuring sufficient space for the robot to navigate while still presenting a challenging environment. The whole maze had a height of about 33cm.

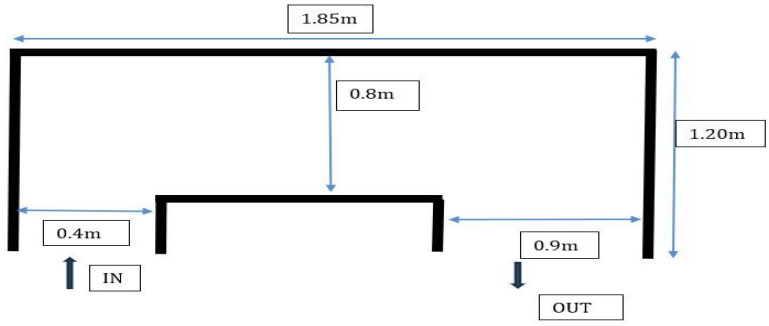



Figure 18: Basic U-Maze (Illustration by the author, 2025)

The VL53L0X was strategically mounted in front of the robot for unobstructed distance measurement and obstacle detection and avoidance. The sensor was inserted into a breadboard and attached to the front base of the XGO Adapter using a Velcro strip. Four wires were plugged into the breadboard and from there extended to the XGO Adapter's I2C pins to be able to communicate with the Micro:bit- the VL53L0X communicates through I2C. The four wires are for the Ground and Voltage, and the SCL to Pin 19, and SDA to Pin 20. The Velcro strips offer flexibility for easy repositioning and experimentation. The breadboards and small wire connectors were secure and semi-permanent attachments. The wiring was carefully done to prevent interference with the robot's movement.

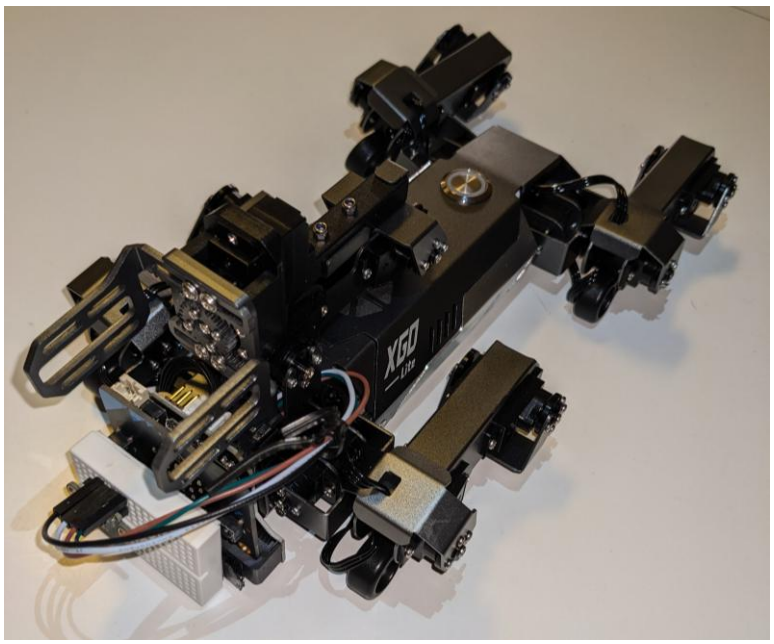



Figure 19: Robot with sensor mounted on breadboard and wiring (Photograph by the author, 2025)

## 4.4 Sensor Test

In the code, a limit of 600mm was set for obstacle detection for the VL53L0X sensor. At that distance, the robot should stop. The robot was placed in front of a wall at a distance much greater than 600mm. Then it was activated to move. A tape was used to measure where it stopped from the wall and it was noted that at the stop distance setting of 600mm, the robot stopped when the wall was 350mm from it. A height and gap test on the sensor was also done. This time, two sheets of paper were held at the 350mm observed stop distance from it, with the robot in the middle. The sheets were gradually separated, and it was noted that the robot started moving when the gap was 200mm wide. A similar test was carried out for the height. It was observed that the sensor detects gaps that are 200mm wide and 200mm high. Below that, it will detect an obstacle.

## 4.5 Navigation Algorithm Design

A reactive obstacle avoidance and left-first, pathfinding algorithm was used. The robot has a set, stop distance of 350mm from the front using VL53L0X. If the sensor detects that the distance of an object in front of it is less than that, robot turns toward the left-side first to detect a greater clearance and a possible path.

# 5 Tests and Results

This section presents the testing procedures used to evaluate the robot's behaviour in obstacle avoidance and maze navigation, along with the results and analysis of its performance based on the programmed algorithms and sensor feedback. It also discusses the limitations encountered during development and compares the robot's capabilities with other robotic platforms in terms of functionality, mobility, and cost.

## 5.1 Reactive Obstacle Avoidance

The robot dog was programmed to turn left when it detects an obstacle. It was placed on the floor at some distance from a wall and then activated. It moved in a straight line towards the wall, stopped, and changed direction when it got to the wall. The code is such that when it gets to the wall or an obstacle, it gradually turns 90 degrees to the left in small steps of 10 degrees per second.  Below is the code that was used to achieve it.

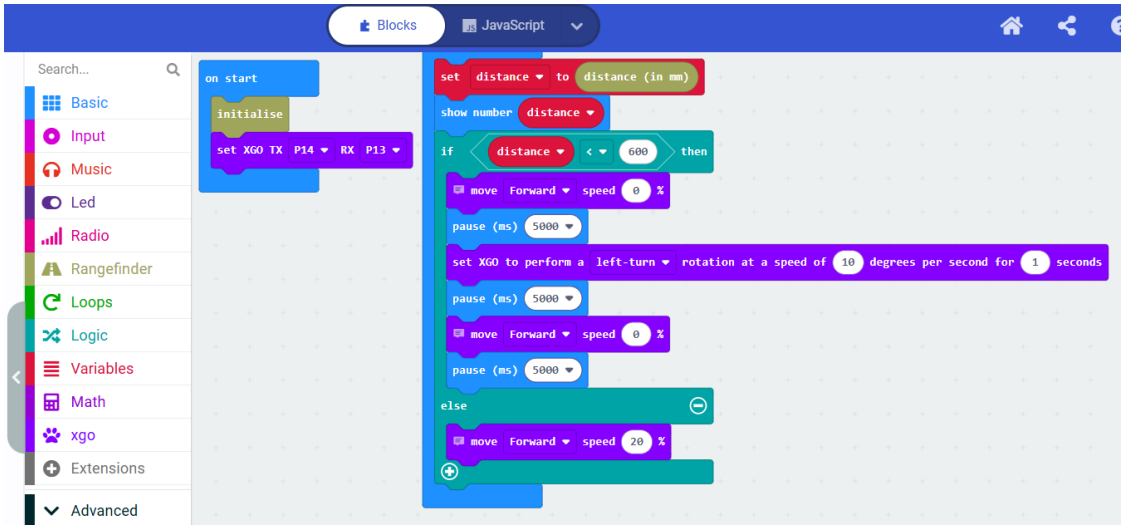



Figure 20: Turn left obstacle avoidance code (Screenshot by the author, 2025)

## 5.2 Path Finding

The program, this time, was that when the robot gets to an obstacle, it first turns 90 degrees left in search of a path. If it does not find one, it turns back 90 degrees to its original heading. It will then turn another 90 degrees to the right, in incremental steps of 10 degrees, in search of a path. The robot was able to move as it was instructed.

Here is the code I used to achieve this.

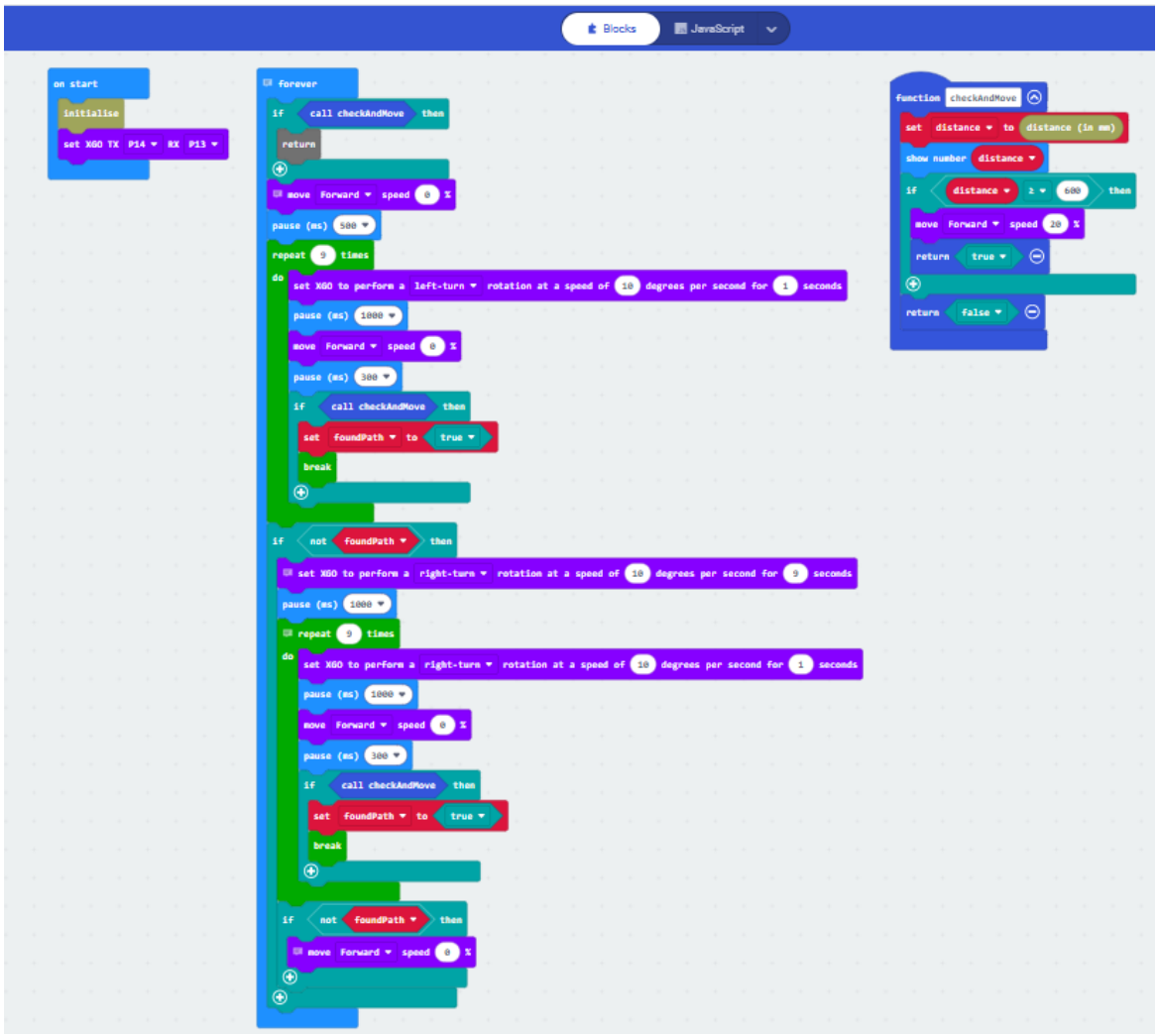



Figure 21: Turn left then right (Screenshot by the author, 2025)

## 5.3 Path Finding Improved

When the robot gets to a wall obstacle that is near a corner, it was observed that while turning, the distance between the robot and the corner can increase significantly above the threshold (600mm). This will in turn trigger the code to move the robot forward under the assumption that a path is clear. The code was modified to increase the threshold for the distance check after initial obstacle detection to 1200mm. But the initial obstacle check distance remains 600mm. This is to eliminate false positives and ensure there is a clear path forward before it can proceed. The robot was able to correctly detect a clear path with this improvement.

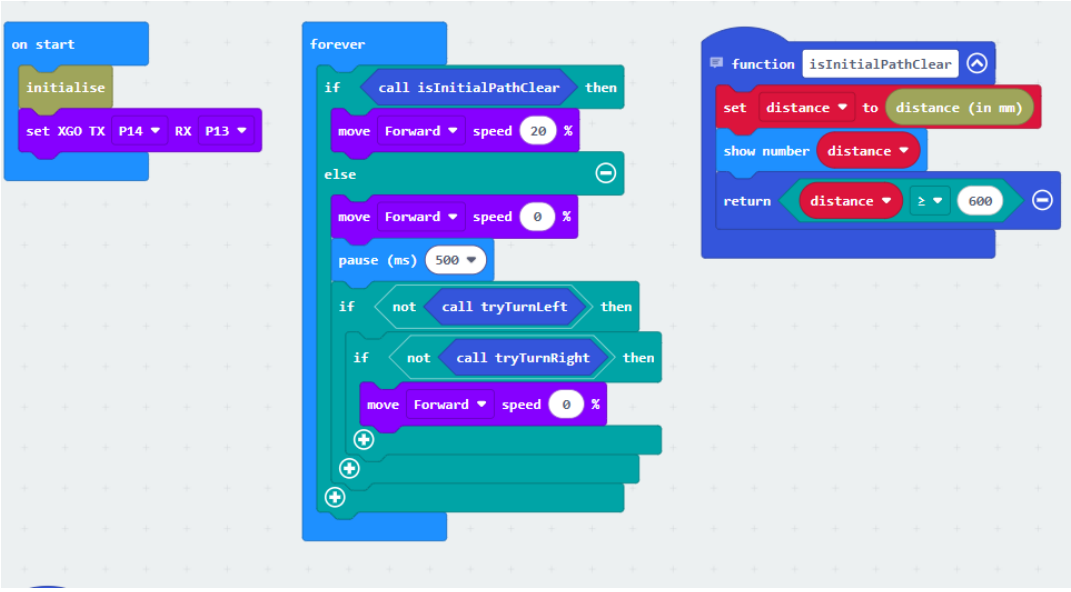



Figure 22: Improved code snippet 1 (Screenshot by the author, 2025)

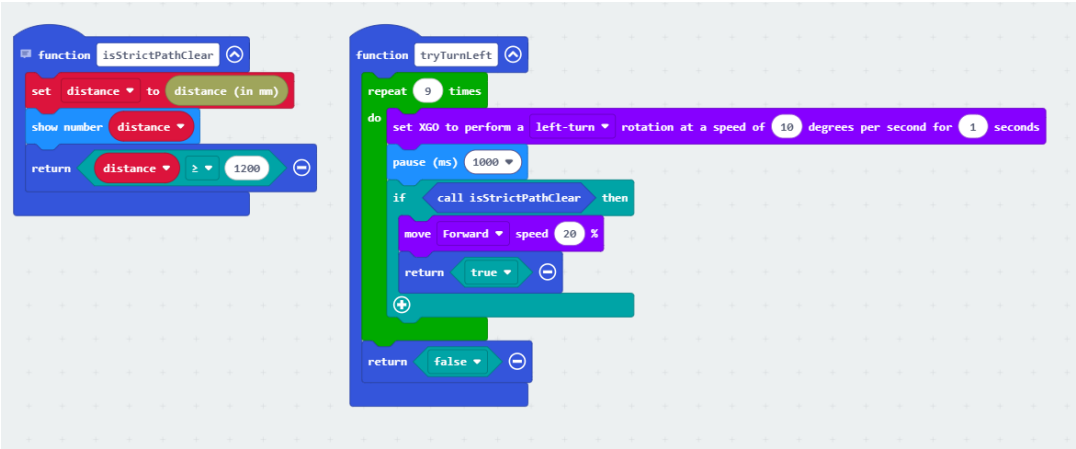



Figure 23: Improved code snippet 2 (Screenshot by the author, 2025)

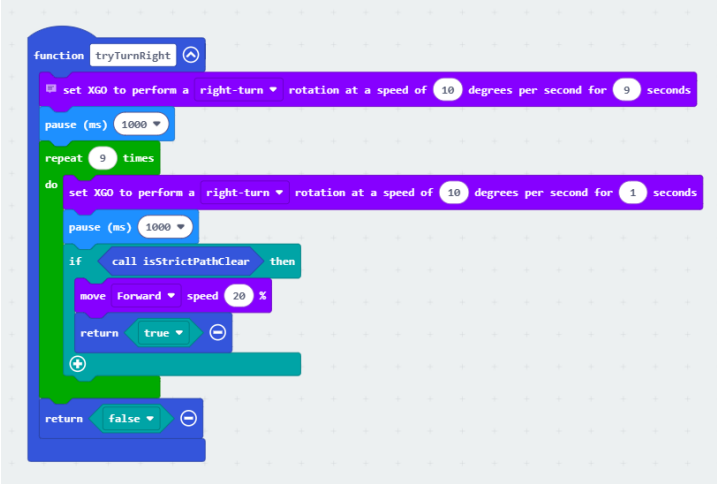



Figure 24: Improved code snippet 3 (Screenshot by the author, 2025)

## 5.4 Discussion of Results

### 5.4.1 Comparison to Objectives

The objective of this project was to develop an autonomous quadruped robot using the Micro:bit XGO Robot V2 dog, capable of solving mazes using integrated sensors and an efficient decision-making algorithm. The XGO robot successfully demonstrated autonomous navigation in the tests. It was able to detect obstacles and find new paths using an algorithm and the VL53L0X ToF sensor. The primary goals—autonomous maze-solving behavior and reliable path detection—were achieved within the constraints of the hardware. However, optimization in terms of speed, accuracy, and turning efficiency remains an area for further improvement.

### 5.4.2 Algorithm Effectiveness

The maze-solving capability was based on a reactive obstacle avoidance and left-first, path finding algorithm, chosen for its simplicity and suitability for unknown, connected mazes. During testing, the algorithm was effective in ensuring the robot found a path.

Also, in these tests, it was observed that the robot only moved when it had a clearance that is 200mm wide and 200mm high. This is perfect for the XGO because the robot's dimensions from the front, about 145mm wide and 170mm fully standing with the clamp in rest position, means that it will only attempt to go through spaces that it can fit into.

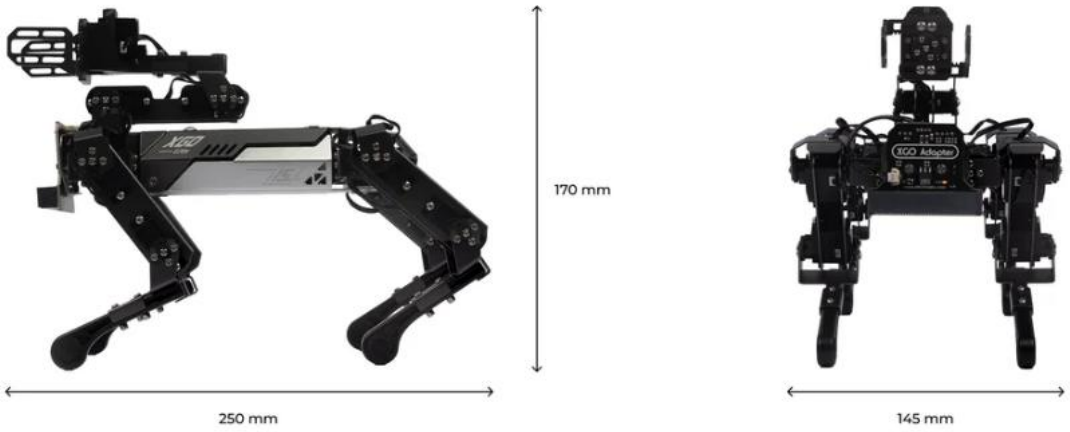



Figure 25: Robot dimensions (*ELECFREAKS XGO Robot Dog Kit V2 - Advanced STEM Learning*, n.d.)

### 5.4.3 Locomotion Performance

The XGO robot's locomotion system, based on servo-driven articulated legs, provided a decent degree of stability during movement. But it vibrates with each movement and sometimes slides a bit and deviates from its path. This also affected the robot when making turns. The algorithm is such that the robot should complete a 90-degree turn in 9 seconds, and if a path is not found, it should turn right back to the central position and start checking for another path from there. But the robot vibrates and sometimes changes direction with each movement. So, it is not able to fulfil the movement requirements accurately. Sometimes, the robot loses its balance and falls when trying to turn.

## 5.5 Limitations

There are several limitations we encountered in this project.

**Voltage Compatibility:**

The micro:bit and XGO Robot operate at 3.3V logic, which prevented the use of commonly available distance sensors such as the TF-Luna, HC-SR04, and Parallax PING, all of which require 5V input. Attempting to use them without proper voltage regulation risked damaging the microcontroller.

**Processing Limitations:**

The micro:bit's limited processing power restricted the complexity of algorithms that could be implemented. More advanced techniques, like SLAM (Simultaneous Localization and Mapping), are not feasible on this platform without external computational modules.

**Algorithm Simplicity:**

The path finding algorithm lacks memory or mapping, so it is unable to optimize routes.

## 5.6 Comparison with other robots

When compared to other autonomous maze-solving quadruped robots, like the Boston Dynamics Spot, using advanced algorithms like Simultaneous Localization and Mapping (SLAM) (Hua, 2024), this robot's pathfinding approach was more basic but easier to

implement with limited hardware. Overall, the project achieved a unique balance between mechanical complexity and algorithmic simplicity, showing that quadruped robots can work in structured but constrained environments.

Other robots can move by using wheels. Wheeled robots are more sturdy, easier to handle, and have a simpler structure, among other benefits. On level ground, they can travel quickly and effectively. Due to their great stability and maneuverability, wheeled robots have also emerged as the most popular type of mobile robot. Wheeled robots' limited mobility and ability to maneuver only in reasonably well-structured settings are major drawbacks. Robots must, however, occasionally navigate gaps and obstructions. Legged robots are more nimble than wheeled robots and can traverse a wider range of terrain, including hills, slopes and valleys, but they also have a lot of drawbacks..(Z. Li et al., 2019)

In terms of cost, this is a low-cost platform for experimenting with robots. The Micro:bit XGO Robot kit V2 cost $449 (*ELECFREAKS XGO Robot Dog Kit V2 - Advanced STEM Learning*, n.d.), the Unitree Go1 costs $2,700 (*Unitree Go1*, n.d.) while the Boston Dynamics Spot costs $75,000 (*The Spot Robot by Boston Dynamics*, n.d.). The low cost makes it more affordable and accessible to students.

# 6 Conclusion and Recommendations

This conclusion summarizes the key findings of this project and reflects on its achievements and limitations. Suggestions are made for future work to enhance the robot’s capabilities.

## 6.1 Summary

In this thesis, the aim was to design and implement a quadruped robot capable of autonomously navigating a maze using the Micro:bit XGO Robot V2 and integrated sensing. It was successfully demonstrated that a low-cost, programmable quadruped robot can achieve autonomous maze navigation using simple sensor input and control logic.

The robot made use of a reactive obstacle avoidance and left-first, path finding algorithm combined with a VL53L0X time-of-flight sensor to detect obstacles and guide movement.

Through a series of structured test environments, it was able to navigate around obstacles and find new paths. It consistently reached the goal in structured mazes.

However, the work also revealed key hardware and algorithmic limitations, such as restricted sensor range, lack of mapping capability, turning inefficiencies, and voltage compatibility issues that limited sensor selection. These challenges offer clear opportunities for future improvements.

Finally, this study shows the potential of compact, quadruped robots in maze-solving tasks and lays the groundwork for more advanced implementations involving smarter navigation and sensor integration.

## 6.2 Future Work and Recommendations

Although this project has successfully demonstrated basic autonomous maze-solving using a quadruped platform, several enhancements are recommended to improve performance, scalability, and real-world applicability.

Additional sensors can be mounted on the sides of the robot to enable it to maintain a consistent distance from the maze wall. A program can be written to use the distance data from the side-mounted sensors to correct the movement of the robot. This will prevent it from bumping into the sides of the wall if it drifts while moving.

Future designs should consider integrating multi-directional sensors, like the RPLIDAR A1, to provide wider field-of-view and more precise environmental awareness. Additionally, the use of voltage level shifters or a dedicated power supply would allow integration of more sensors like the TF-Luna, or HC-SR04/ Parallax PING. Multiple sensor utilization (e.g., combining infrared, ultrasonic, GPS and time-of-flight data) could help mitigate the limitations of any single sensor type. Adding gyroscopic or IMU sensors like the MPU-6050 or the MPU-9250 could improve movement stability and orientation awareness.

The performance of the robot can be improved by using Dijkstra's or Flood-Fill algorithms, which would allow for shortest-path navigation. Implementing simple memory or mapping logic (e.g., grid-based occupancy maps) would help the robot avoid repeated paths and

improve decision-making in complex mazes. Also, integration with machine learning models or reinforcement learning for adaptive maze-solving could be explored using offboard processing or upgraded controllers, as the micro:bit is a very limited microcontroller.

# 7 References

Adafruit Industries. (n.d.). *HC-SR04 Ultrasonic Sonar Distance Sensor + 2 x 10K resistors*. Retrieved May 30, 2025, from https://www.adafruit.com/product/3942

Alamri, S., Alamri, H., Alshehri, W., Alshehri, S., Alaklabi, A., & Alhmiedat, T. (2023). An Autonomous Maze-Solving Robotic System Based on an Enhanced Wall-Follower Approach. *Machines*, *11*(2), 249. https://doi.org/10.3390/machines11020249

Celeste. (2024, February 27). I2C: A Brief History. *Focus LCDs*. https://focuslcds.com/journals/i2c-a-brief-history-and-what-it-means-for-lcd-display-technologies/

Chen, T., Huangfu, Y., Srigrarom, S., & Khoo, B. C. (2024). Path Planning and Motion Control of Robot Dog Through Rough Terrain Based on Vision Navigation. *Sensors*, *24*(22), 7306. https://doi.org/10.3390/s24227306

*ELECFREAKS XGO Robot Dog Kit V2—Advanced STEM Learning*. (n.d.). ELECFREAKS. Retrieved May 31, 2025, from https://shop.elecfreaks.com/products/elecfreaks-xgo-robot-dog-kit-v2-for-micro-bit

*Gravity: BMM150 Triple Axis Magnetometer Sensor*. (n.d.). Retrieved May 30, 2025, from https://www.dfrobot.com/product-2626.html

*Hardware*. (n.d.). Retrieved June 1, 2025, from https://tech.microbit.org/hardware/

*Hardware Introduction | LEARN*. (n.d.). Retrieved May 30, 2025, from https://wiki.elecfreaks.com//en/microbit/robot/xgo-robot-kit-v2/hardware-introduction/

*How to Resolve I2C Address Conflicts in Embedded Systems*. (n.d.). Retrieved May 30, 2025, from https://wiki.dfrobot.com/How_to_Resolve_I2C_Address_Conflicts_in_Embedded_Systems

*How to Use Accelerometer Sensor with Micro:bit?* (n.d.). Retrieved June 1, 2025, from https://www.keyestudio.com/blog/how-to-use-accelerometer-sensor-with-microbit-132

Hua, S. (2024). *Evaluation of selected localiza- tion and mapping techniques for the Boston Dynamics Spot robot*.

Ibraheem, S. M., & Adrees, S. (2023). *Embedded Systems for Engineers and Students* (2nd ed.). National Library of Pakistan.

*Index | LEARN*. (n.d.). Retrieved May 25, 2025, from https://wiki.elecfreaks.com/en/microbit/robot/xgo-robot-kit-v2/

Li, Q., Cicirelli, F., Vinci, A., Guerrieri, A., Qi, W., & Fortino, G. (2025). Quadruped Robots: Bridging Mechanical Design, Control, and Applications. *Robotics, 14*(5), Article 5. https://doi.org/10.3390/robotics14050057

Li, Z., Su, H., & Hereid, A. (2019). *Mechanical Design, Analysis and Simulation of a Quadruped Robot*.

Ltd, A. (n.d.). *BBC micro bit*. Arm | The Architecture for the Digital World. Retrieved May 30, 2025, from https://www.arm.com/company/success-library/arm-designs/microbit

Lynxmotion Wiki. (n.d.). *PING)))™ Ultrasonic Distance Sensor (#28015)*. Retrieved May 30, 2025, from https://wiki.lynxmotion.com/info/wiki/lynxmotion/download/servo-erector-set-system/ses-electronics/ses-sensors/WebHome/pingv1.2.pdf

*MakeCode Graphical Programming | LEARN*. (n.d.). Retrieved June 1, 2025, from https://wiki.elecfreaks.com//en/microbit/robot/xgo-robot-kit-v2/quick-start/makecode-graphical-programming/

*Micro:bit pins*. (n.d.-a). Microsoft MakeCode. Retrieved June 2, 2025, from https://makecode.microbit.org/device/pins

*Micro:bit XGO Robot Kit V2 | LEARN*. (n.d.-b). Retrieved June 6, 2025, from https://wiki.elecfreaks.com//en/microbit/robot/xgo-robot-kit-v2/xgo-lite-v2-products-instruction/

Nguyen, P. (n.d.). *ROBOTICS IN HEALTHCARE*.

Pololu. (n.d.). *PING)))™ Ultrasonic Distance Sensor (#28015)*. Retrieved May 30, 2025, from https://www.pololu.com/file/0J214/PING_documentation.pdf

*Pololu—VL53L0X Time-of-Flight Distance Sensor Carrier with Voltage Regulator, 200cm Max*. (n.d.). Retrieved May 30, 2025, from https://www.pololu.com/product/2490

Sparkfun. (n.d.). *Ultrasonic Ranging Module HC-SR04*. Retrieved May 30, 2025, from https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

STMicroelectronics. (2022). *LSM303AGR: Ultra-compact high-performance eCompass module* [Datasheet]. STMicroelectronics. https://www.st.com/resource/en/datasheet/lsm303agr.pdf

*TCA9548A data sheet, product information and support | TI.com*. (n.d.). Retrieved May 30, 2025, from https://www.ti.com/product/TCA9548A

*TF-Luna Datasheet | PDF | Lidar | Equipment*. (n.d.). Scribd. Retrieved May 30, 2025, from https://www.scribd.com/document/815121049/TF-Luna-Datasheet

*The Spot Robot by Boston Dynamics: Features & Use Cases - Standard Bots*. (n.d.). Retrieved June 1, 2025, from https://standardbots.com/blog/spot-robot

*Unitree Go1*. (n.d.). UnitreeRobotics. Retrieved June 1, 2025, from https://shop.unitree.com/products/unitreeyushutechnologydog-artificial-intelligence-companion-bionic-companion-intelligent-robot-go1-quadruped-robot-dog

*VL53L0X - Time-of-Flight (ToF) ranging sensor—STMicroelectronics*. (n.d.). Retrieved June 1, 2025, from https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html

*Xgo-lite2*. (n.d.). Microsoft MakeCode. Retrieved May 31, 2025, from https://makecode.microbit.org/44323-08936-26857-99886

# 8 Appendices

Appendix 1

**Getting Started with the XGO Robot Kit V2**

Before beginning coding and sensor integration, it is essential to properly set up the XGO Robot Kit V2.

**Installing the micro:bit**

Insert the BBC micro:bit into the XGO Adapter board, ensuring the edge connector is fully seated and aligned correctly. The LED matrix on the micro:bit should face forward when installed.

**Software Setup**

To write and upload code to the micro:bit, the following are required:

- **MakeCode (online)** for writing no-code blocks, Javascript or MicroPython(https://makecode.microbit.org/)
- **Micro:bit Python editor**(https://python.microbit.org/)
- **Mu Editor** (download from https://codewith.mu/ and install on computer)

A **micro-USB cable** is needed to connect the micro:bit to your computer.

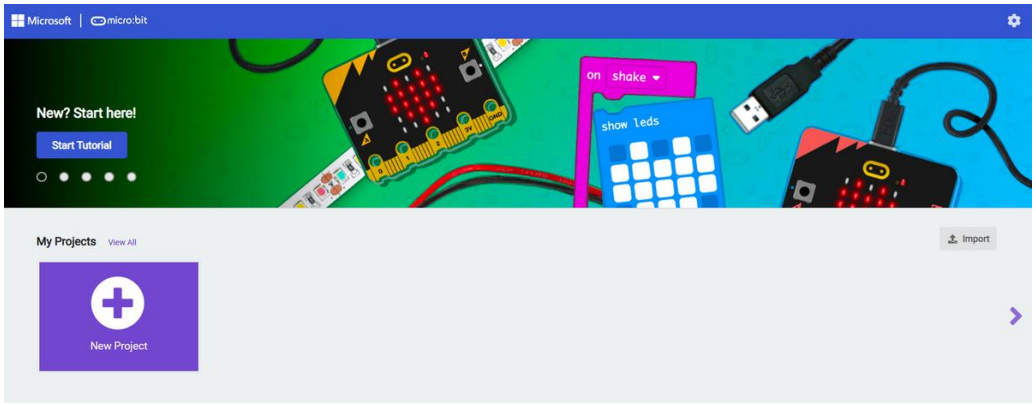
Once installed or when the website is accessed:

1. Plug in the micro:bit using the USB cable.
2. Pair the devices.
3. Write a simple script (e.g., flash one or several LEDs or move a leg).
4. Click download to upload the code to the micro:bit or save the script as .py or .hex and upload it to the micro:bit.

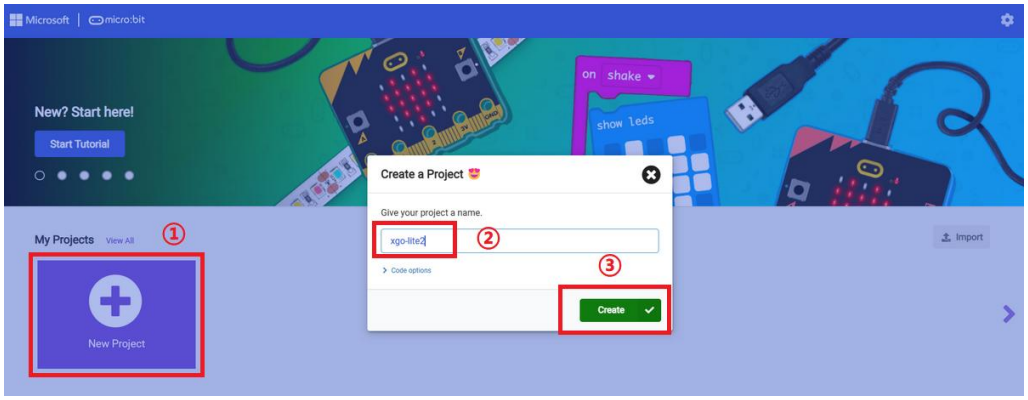
## MakeCode Graphical Programming

### Starting a Project

Open the official website of the MakeCode platform https://makecode.microbit.org/, Microsoft MakeCode for micro:bit, click "**New Project**", name the project and click "**Create**".



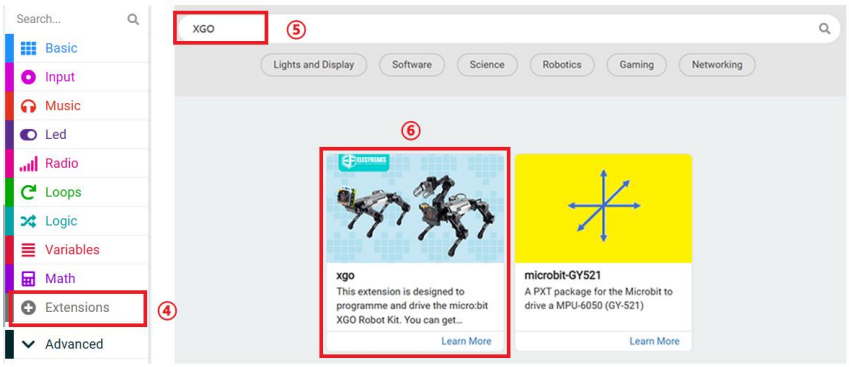

New MakeCode Project (*MakeCode Graphical Programming | LEARN*, n.d.)




Create a Project (*MakeCode Graphical Programming | LEARN*, n.d.)

**XGO Library File Loading**

Select **Extension** and click, enter **XGO** in the search bar and press Enter to search, select **XGO** library, so that you can load the XGO library to the Makecode platform.




XGO Library File Loading (*MakeCode Graphical Programming | LEARN*, n.d.)

## Appendix 2 – Maze Navigation code in JavaScript

Blocks | JavaScript

```
// Initial movement check (>= 600 mm)
function isInitialPathClear () {
    distance = Rangefinder.distance()
    basic.showNumber(distance)
    return distance >= 600
}
function tryTurnLeft () {
    for (let index = 0; index < 9; index++) {
        xgo.rotate_angle_continue(xgo.rotate_direction_enum.turn_left, 10, 1)
        basic.pause(1000)
        if (isStrictPathClear()) {
            xgo.move_xgo(xgo.direction_enum.Forward, 20)
            return true
        }
    }
    return false
}
function tryTurnRight () {
    // Return to original heading
    xgo.rotate_angle_continue(xgo.rotate_direction_enum.turn_right, 10, 9)
    basic.pause(1000)
    for (let index = 0; index < 9; index++) {
        xgo.rotate_angle_continue(xgo.rotate_direction_enum.turn_right, 10, 1)
        basic.pause(1000)
        if (isStrictPathClear()) {
            xgo.move_xgo(xgo.direction_enum.Forward, 20)
            return true
        }
    }
    return false
}
```

```
// Stricter check during scan (>= 1200 mm)
function isStrictPathClear () {
    distance = Rangefinder.distance()
    basic.showNumber(distance)
    return distance >= 1200
}
let distance = 0
Rangefinder.init()
xgo.init_xgo_serial(SerialPin.P14, SerialPin.P13)
basic.forever(function () {
    if (isInitialPathClear()) {
        xgo.move_xgo(xgo.direction_enum.Forward, 20)
    } else {
        xgo.move_xgo(xgo.direction_enum.Forward, 0)
        basic.pause(500)
        if (!(tryTurnLeft())) {
            if (!(tryTurnRight())) {
                xgo.move_xgo(xgo.direction_enum.Forward, 0)
            }
        }
    }
})

```