# Project Configurability and Field Deployment Plan:

## Project Configurability:

### Reusability:

The reusability of this product stems from the use of generally accessible, worldwide protocols and methods, with communication methods and operating systems such as MQTT, UART, ROS and Zephyr being central to the product. Through utilising this functionality, this product’s firmware can be integrated onto a wide variety of hardware, promoting reusability. Specifically, almost all development boards are capable of UART and Zephyr integration, so the use of these schemes is not limiting in widespread reusability and the Nucleo board can easily be replaced. Further, ROS is the industry standard for autonomous robotic devices, so the TurtleBot can be swapped for almost any autonomous robot, provided they meet the low barrier of entry – having an integrated LiDAR sensor or having the potential to communicate via UART to an externally added LiDAR. From these hardware selections it allows our project to be utilized on many different platforms maximizing its reusability. Finally, to further the reusability of our project the TurtleBot’s code is run through a python script that can be converted into an executable file allowing any OS to drive the autonomous robotic code through ROS.

The steps taken to allow our project to be reusable is in fact that our group has used general worldwide functionality and protocols such as zephyr, MQTT, UART and ROS when developing this project. By utilizing this functionality, it allows our firmware to be integrated onto many different types of hardware allowing it to be reusable. This hardware selection includes any programmable board/platform that supports zephyr and UART communication and even MQTT transmission. As well as any autonomous robotic that can run ROS and can have a LiDAR sensor attached already or installed and utilized through UART communication protocols. From these hardware selections it allows our project to be utilized on many different platforms maximizing its reusability. To further maximise the reusability of our project the TurtleBot’s code is run through a python script that can be converted into an executable file allowing any OS to run our code through ROS.

### Reconfigurations or possible addition features:

As a direct consequence of the emphasis placed on reusability in this product, reconfiguration options are both simplistic and plentiful. To begin, the system’s inputs can be simply modified, as the ultrasonic sensors can be replaced, or added to, by other sensors with UART or MQTT communication features. Adding to the input sensors would allow more complicated ranges and complexity of gestures to be realised, with mappings of 360 degree gestures possible through using LiDAR sensors, more intricate gestures can be mapped utilising an input camera and machine learning techniques and so on. Another reconfiguration is by finding a programmable board that supports MQTT communication allowing for the external m5core2 communication node to be disconnected. This cross-communication removal allows for the tracked position of the TurtleBot’s location to be displayed on many different digital display devices that utilise the UART communication protocol instead of the m5core2. This would significantly simplify the overall system, as it removes one of the programmable modules involved and replaces it with a simple UART LCD display. For further features the programmable board used could also have other features such as being able to utilise low energy Bluetooth capabilities, which could promote communication to multiple different autonomous robot systems, or to communicate with ranging sensors to further guide the robot.

## Field Deployment Plan:

### Ease of use:

To ensure ease of field deployment of this product, some steps have, and can be, taken. To begin, the TurtleBot’s python script can be converted into a python executable file, which will automatically connect to ROS and run the operational script. This process will enhance efficiency and simplicity, as the deployment team will not be required to access the ROS terminal in order to set up the product. Furthermore, the development board which processes the ultrasonic sensor information and transmits this to the core can be pre-flashed, so that powering this device is the only requirement for its operation. This same process can be repeated for the M5Core, meaning that the only code that needs to be executed is the prepared executable file.

Additionally, a manual has been created to describe the gesture actions performed to the Ultrasonic sensors, and what the TurtleBot’s responses to these gestures will be. This allows a deployment team to prepare their actions ahead of deployment, in order to seamlessly deploy the product. Further, despite the intentional simplicity of wiring this device, a wiring manual has been created, which will further aid the deployment team in efficiently setting up the product.