Getting started with STSW-ROBKIT1 for STEVAL-ROBKIT1 Robotics evaluation kit

Introduction

The STEVAL-ROBKIT1 is a comprehensive Robotics Evaluation Kit designed as a platform for the development of robotic technology and its applications. The kit features a modular design consisting of three boards: the Main board STEVAL-ROBKIT1-1, the Motor control board STEVAL-ROBKIT1-2, and the Imaging board STEVAL-ROBKIT1-3. The Main board is powered by an STM32H725 MCU, which integrates various functionalities and controls both the motor board and the imaging board. The Motor board is based on an STM32G071 microcontroller, dedicated to motor control and actuation using motor drivers to regulate the speed and direction of the robot's movements and the Imaging board is equipped with a Time-of-Flight (ToF) sensor and a camera module, enabling the robot to perceive and interact with its surroundings intelligently. Additionally, an Inertial Measurement Unit (IMU) and a magnetometer enhance the board's capabilities by providing precise orientation and motion sensing, crucial for navigation and stability in dynamic environments. The Bluetooth Low Energy (BLE) module on the main board facilitates seamless wireless communication, enabling control via a mobile application interface.

The STSW-ROBKIT1 is a comprehensive software package for the STEVAL-ROBKIT1 Robotics Evaluation Kit, enabling high-performance robotic application development.

Key features:

- Ready examples for robotics application development
- Independent firmware for the STEVAL-ROBKIT1-1 (Main board) and STEVAL-ROBKIT1-2 (motor board)
- FreeRTOS™ task scheduling in main board firmware to ensure a better message handling and improved system performance
- Supports integration of AI models and applications, enabling advanced robotics functionalities
- Odometry for precise navigation
- DCMI interface for efficient and versatile camera integration supporting various image resolutions
- 8x8 multizone Time-of-Flight sensor data for various applications like navigation and Edge/Cliff detection
- Data transmission via Bluetooth® Low Energy connectivity to support remote control and on board sensors data logging of the robot using dedicate mobile application, STRobotics app
- Workspace support for IAR, STM32CubeIDE, and Keil®
- Capability for seamless reconfiguration of the peripherals using the .ioc file for STM32CubeMX, simplifying the development process for the users

Figure 1. **STEVAL-ROBKIT1 Robotic application evaluation kit**

Notice: *For dedicated assistance, submit a request through our online support portal at* ***www.st.com/support****.*

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
STEVAL-ROBKIT1-1	Main board
STEVAL-ROBKIT1-2	Motor board
STEVAL-ROBKIT1-3	Imaging Board
BLE	Bluetooth low energy
MHz	Mega Hertz
MCU	Microcontroller unit
I2C	Inter integrated circuit
ToF	Time-of-Flight
API	Application programming interface
HAL	Hardware abstraction layer
IDE	Integrated development environment
PID	Proportional-integral-derivative controller

2 Getting started

2.1 Overview

Users can program the STEVAL-ROBKIT1-1 (Main Board), STEVAL-ROBKIT1-2 (Motor Board) and BlueNRG-M2SA module on the main board to run following using STRobotics mobile application:

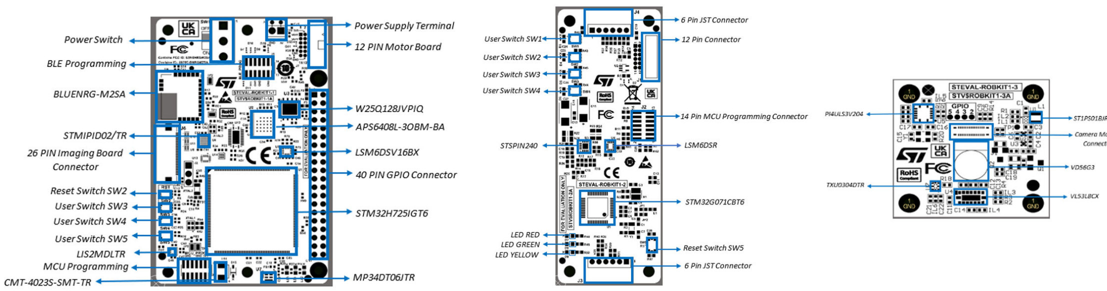
- Remote Controller
- Free Navigation
- Follow-Me Navigation
- Object Detection
- Edge/Cliff Detection
- QR Code Scanner used to link the dedicated mobile app to the correct kit, by reading its unique MAC address
- Data plot for the on board magnetometer, gyroscope, accelerometer and ToF sensor
- Odometry data plot on the mobile app
- In addition to the above applications, the user can make use of the on-board devices and sensors to add and evaluate their own firmware applications

The STSW-ROBKIT1-1 also consists of a dedicated AI example on surface detection using the data of the on-board accelerometer sensor.

There are several ST products used including BLE module and Sensors on STEVAL-ROBKIT1:

- STM32H725IGT6 (Main board controller)
- STM32G071CBT6 (Motor board controller)
- BlueNRG-M2SA
- LSM6DSV16BX (iNEMO 3D accelerometer and 3D gyroscope)
- LIS2MDLTR (Magnetometer sensor)
- MP34DT06JTR (MEMS microphone)
- VD56G3 (Monochrome global shutter image sensor)
- VL53L8CX (Multi-zone time-of-flight sensor)
- STSPIN240 (DC motor driver)
- LSM6DSR (3D accelerometer and 3D gyroscope on motor board)

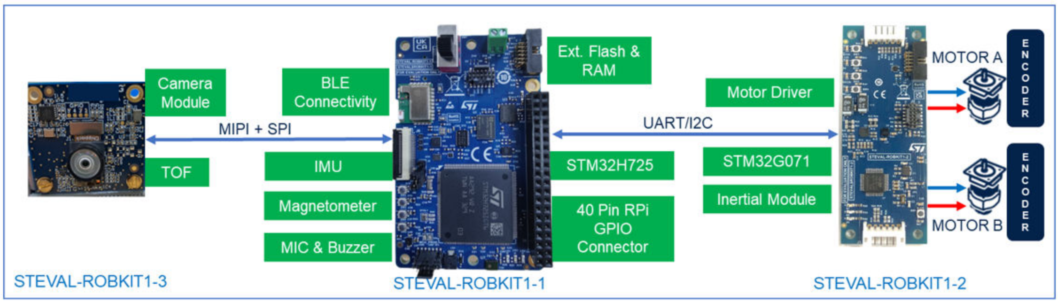
Figure 2. STEVAL-ROBKIT1: Main Board, Motor Board and Imaging Board components



The STEVAL-ROBKIT1 system consists of three interconnected modules. The **STEVAL-ROBKIT1-1** serves as the main board, linking both the **STEVAL-ROBKIT1-3** (Imaging board) and the **STEVAL-ROBKIT1-2** (Motor board). The Imaging board transmits TOF and camera data to the main board for processing. Equipped with onboard sensors, the main board processes this data and, based on the sensor inputs, executes algorithms that send commands to the Motor board to control the motors.

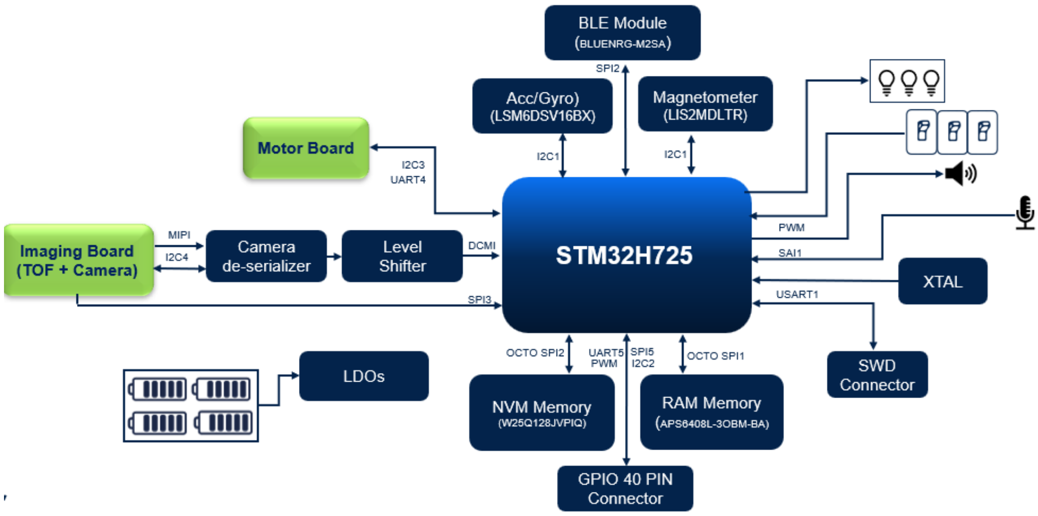
Below is the high-level block diagram of the Robotics evaluation Kit:

Figure 3. Block diagram of STEVAL ROBKIT 1



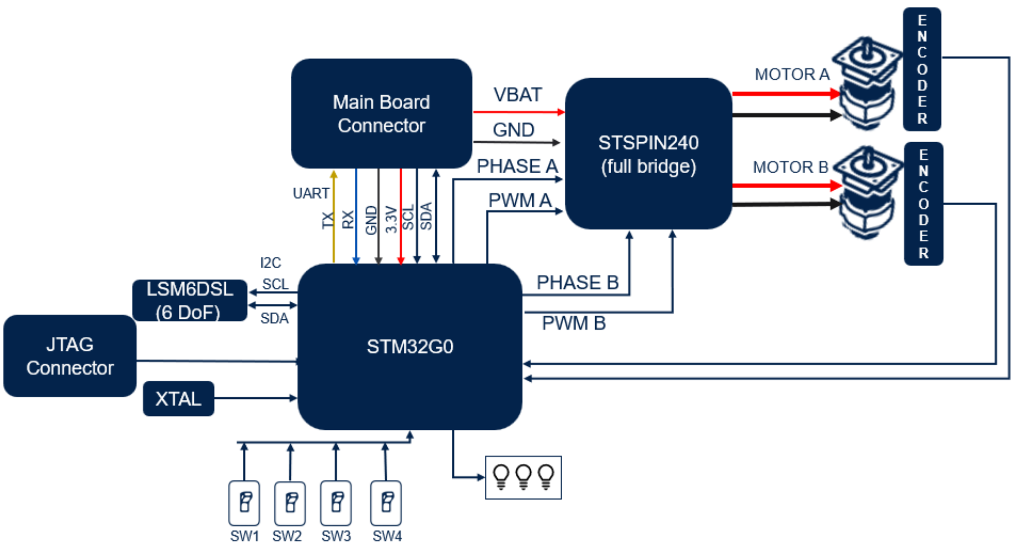
The block diagram of the main board is as follows:

Figure 4. Block diagram of STEVAL ROBKIT 1-1



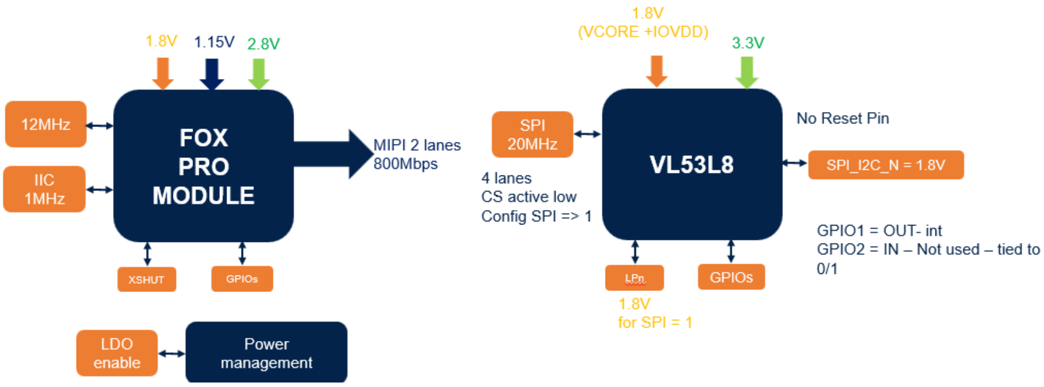
The block diagram of the motor board is as follows:

Figure 5. Block diagram of STEVAL ROBKIT 1-2



The block diagram of the imaging board is as follows:

Figure 6. **Block diagram of STEVAL ROBKIT 1-3**

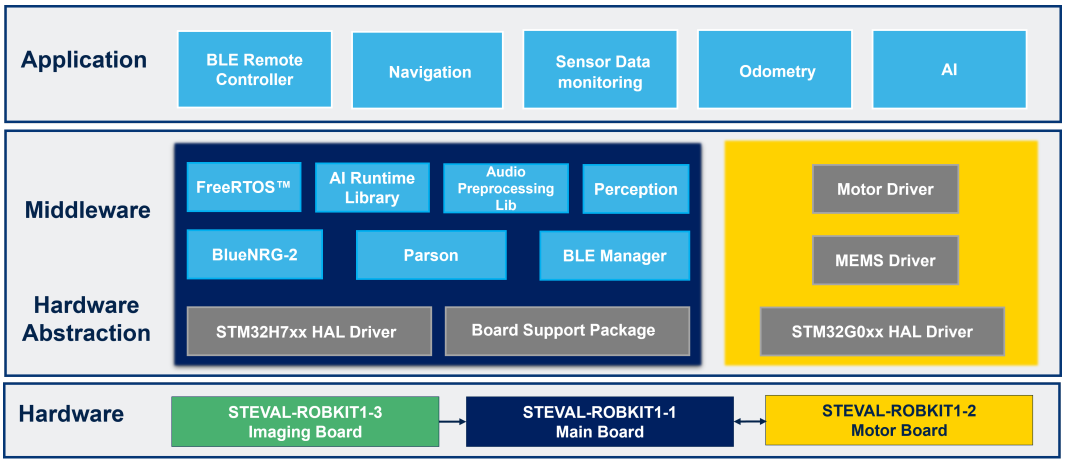


2.2 Architecture

The firmware is based on STM32Cube technology and expands STM32Cube based packages. The package provides a board support package (BSP) for the sensors and the middleware components for Bluetooth® Low Energy communication with any external mobile device. The firmware driver layers to access and use the hardware components are:

- STM32Cube HAL layer: simple, generic, multi-instance application programming interfaces (APIs) that interact with the upper layer applications, libraries, and stacks. The APIs are based on the common STM32Cube framework so the other layers (e.g., middleware) can function without requiring any specific hardware information for a given microcontroller unit (MCU), thus improving library code reusability and guaranteeing easy portability across devices.
- Board support package (BSP) layer: provides firmware support for the evaluation board (excluding MCU) peripherals (LEDs, user buttons, etc.) but can also be used for the board serial and version information, and to support initializing, configuring and reading data from sensors. You can build the firmware using specific APIs for the following hardware subsystems: – Platform: to control and configure all the devices in the battery and power subsystem, button, LEDs and GPIOs – Sensors: to link, configure and control all the sensors involved.

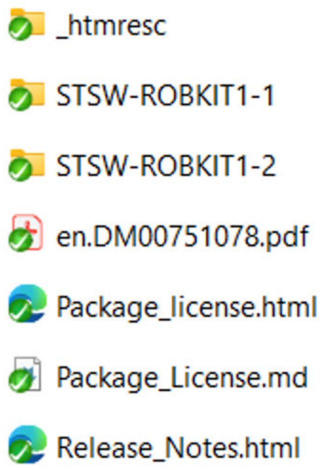
Figure 7. **High Level Firmware architecture of Robotics evaluation kit**



2.3 Folder structure of STSW-ROBKIT1

The STSW-ROBKIT1 is organized in two main folders, one containing the firmware for STEVAL-ROBKIT1-1 (Main Board): STSW-ROBKIT1-1 and the other one containing the firmware for STEVAL-ROBKIT1-2 (Motor Board): STSW-ROBKIT1-2, as shown in figure below:

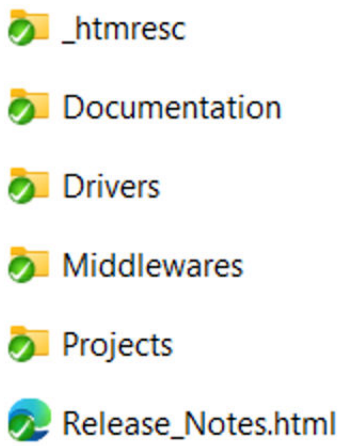
Figure 8. **STSW-ROBKIT1 package folder structure**



These two main folders are described in the next two sections, Section 2.3.1 and Section 2.3.2.

2.3.1 STSW-ROBKIT1-1: firmware for Main Board

Figure 9. **STSW-ROBKIT1 package folder structure**



The following folders are included in the firmware package of the main board, STSW-ROBKIT1-1:

- **Documentation** contains a compiled HTML file generated from the source code which details the software components and APIs.
- **Drivers** contains HAL drivers and the board-specific drivers for the hardware platform, including the on-board components and the CMSIS vendor-independent hardware abstraction layer for ARM Cortex-M processor series.
- **Middlewares** contains libraries and protocols such as the BlueNRG-2 Bluetooth® Low Energy stack, CMSIS-RTOS v2 wrapper implementation, Parson JSON library and serialization, audio library for PDM audio signal processing, perception algorithm implementation, and BLE manager library providing APIs to manage BLE services and custom BLE services exclusively for the Robotics Development Kit.
- **Projects** contains applications for creating customized advanced robotics solutions using the evaluation kit for robotic applications (**STEVAL-ROBKIT1**).

The application projects available are:

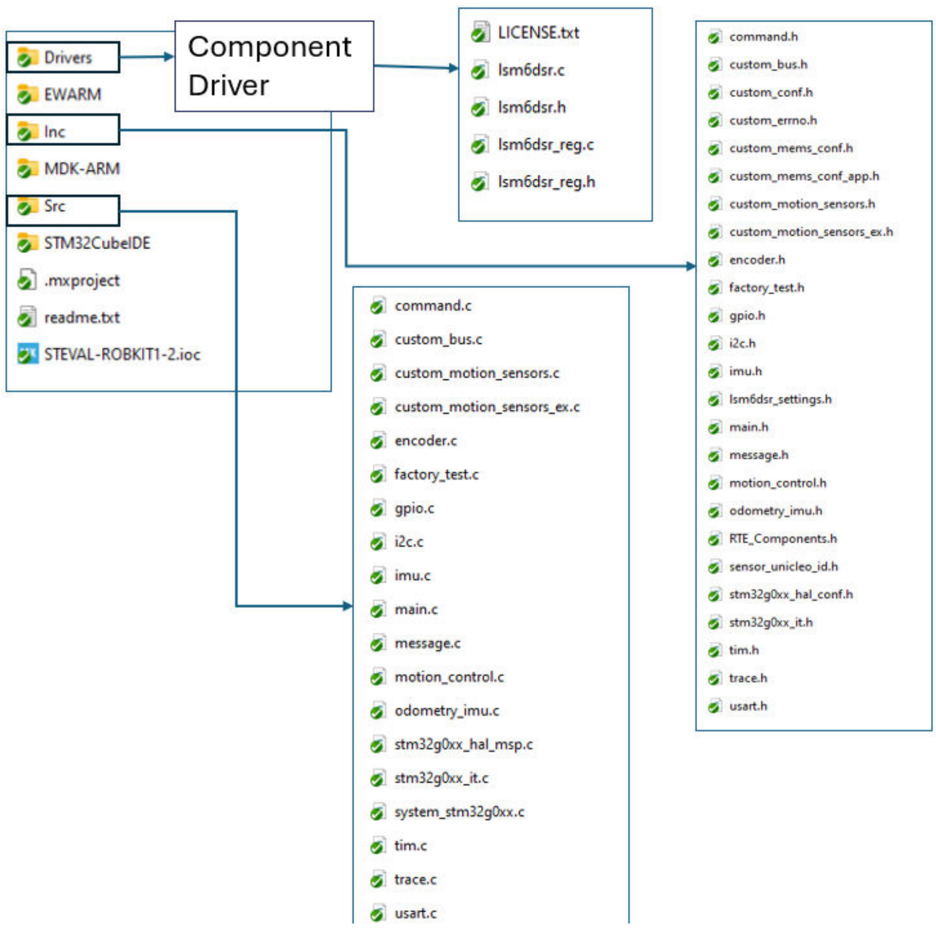
- **AI_Surface_Detection**: An AI example demonstrating how to use the LSM6DSV16BX's accelerometer sensor data to train an AI model and predict the surface on which the robotic kit is moving.
- **Robkit_Control**: An advanced robotics application firmware that connects the kit with the mobile application (STRobotics app), enabling commands to control the kit in different modes (drive, auto-pilot, follow-me navigation) and monitor real-time on-board sensor data.

All projects are available for IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STR), and STM32CubeIDE multi-OS development tools.

2.3.2 STSW-ROBKIT1-2: firmware for Motor Board

- **Firmware**:
  The firmware folder consists of the projects for IAR, Keil and STM32CubeIDE with the application source and header files.
- **Drivers**:
  - **BSP**: SDK drivers providing an API interface to the STM32G071, hardware resources (LEDs, buttons, sensors, I/O channel)
  - **CMSIS**: STM32G0xx CMSIS files
  - **STM32G0xx_HAL_Driver**: drivers for device peripherals (ADC, clock, GPIO, I²C, IWDG, LPUART, PWR, RCC, RNG, RTC, SPI, SysTick, TIM, and USART)
  - **Components**: Sensors Driver for the LSM6DSL Imu sensor

Figure 10. The folder structure of the STEVAL-ROBKIT1-2



2.4 Hardware requirements

To use the STSW-ROBKIT1 package you need following:

- STEVAL-ROBKIT1 (Robotics Evaluation Kit) - (1 Qty)
- STLINK-V3PWR Programmer and 14 pin Cable - (1 Qty)
- PC Windows 10 or higher
- Mobile (Android/iOS)
- AA Non-Rechargeable Batteries - 1.5 V (4 Qty)

2.5 Software requirements

To use the STSW-ROBKIT1 package you need following:

- STSW-ROBKIT1 software package.
- IAR ARM v9.60.3 / MDK-ARM v5.38 / STM32Cube IDE v1.19.0
- RF-Flasher Utility GUI
- STM32CubePROGRAMMER GUI
- STRobotics Mobile App for Android/iOS
- Tera Term

Note: Before using the platform, be sure that the firmware in use in aligned with the final version on www.st.com (STSW-ROBKIT1).

2.6 Board setup

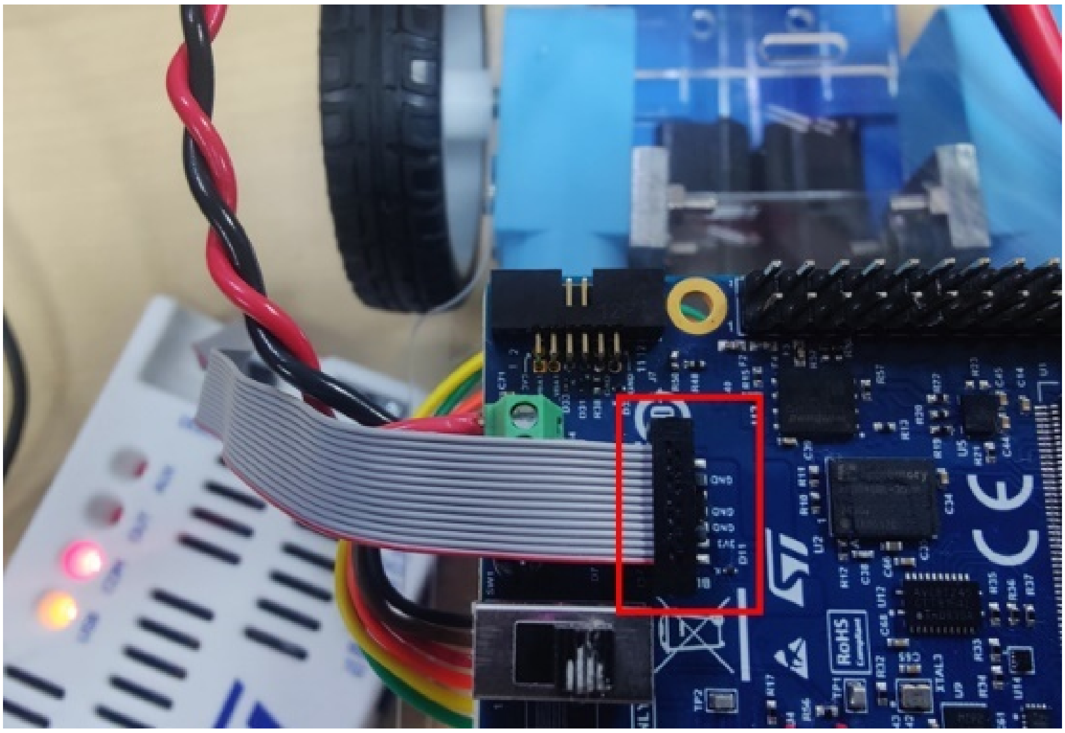
2.6.1 Setup for BLE programming

Setting up a BLE (Bluetooth Low Energy) programming board involves several steps, including hardware setup and software setup. Here's a concise guide to help you get started:

Step 1. Follow the steps below to flash the DTM in BLE module:

– Connect the STLINK Debugger to BLE_PROG connector (J5) as highlighted in Figure 11 below and plug the STLINK debugger to PC via USB Type-A to Type-C USB cable.

Figure 11. **STLINK Debugger connected to BLE_PROG connector (J5)**



Step 2. Install the Software of RF-Flasher Utility by the given link STSW-BNRGFLASHER - The RF- Flasher utility - STMicroelectronics by following the below Figure 12.

Step 3. The BlueNRG-M2SA module on the STEVAL-ROBKIT1 comes preprogrammed with a Bluetooth Low Energy (BLE) stack binary, as described in the steps below:

Note: No reprogramming is needed for typical use cases, as the BlueNRG-M2SA module is ready to operate with this firmware.

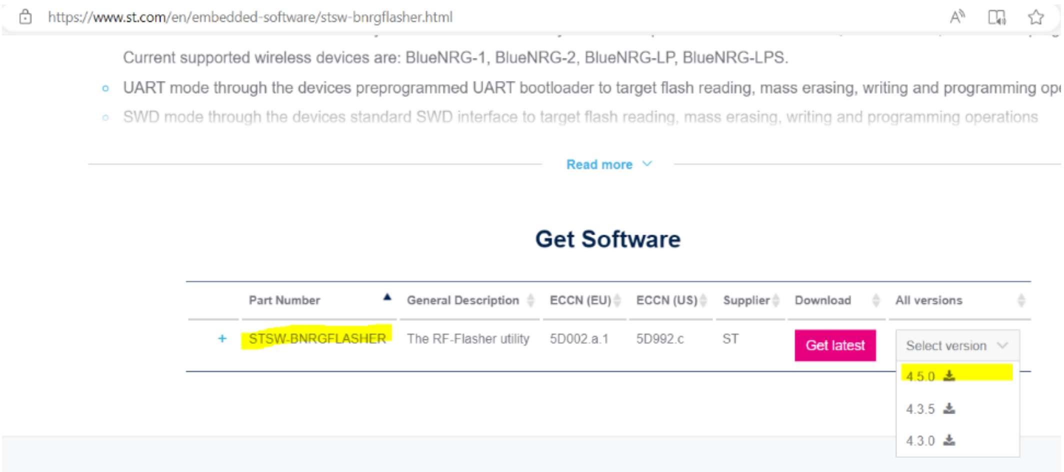


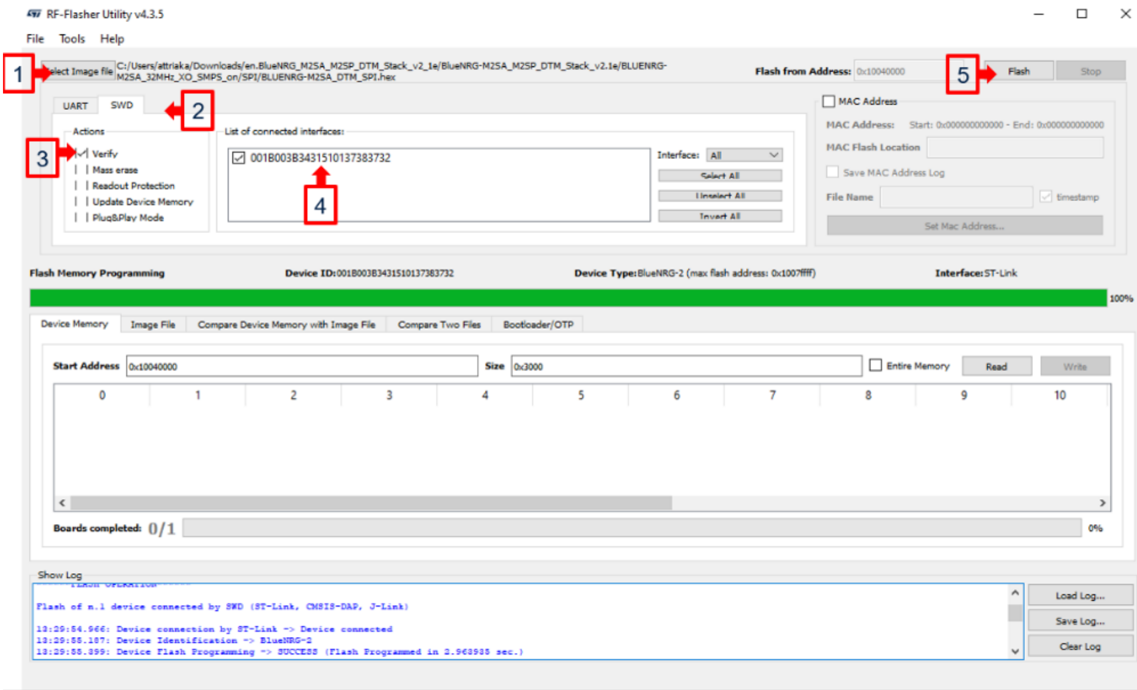
Figure 12. **Steps to install RF- Flasher Utility software**

– The binary file is part of the BlueNRG-M2 - BLE v5.2 stack package downloadable here: BlueNRG-M2 - Very low power application processor module for Bluetooth® low energy v5.2 - STMicroelectronics https://www.st.com/resource/en/utilities/bluenrg_m2sa_m2sp_dtm_stack_v2_1e.zip
– Path to the binary inside the package: BlueNRG-M2SA_M2SP_DTM_Stack_v2.1e\BLUENRG-M2SA_32MHz_XO_SMPS_on\SPI\ BLUENRG-M2SA_DTM_SPI.hex

Step 4. If user need to flash a custom firmware or reprogram the BlueNRG-M2SA module, open the RF Flasher Utility Software and follow the steps highlighted in the Figure 13 to flash the custom DTM binary into the BLE module

- **Step 4a.** Select the Image File to add BLUENRG-M2SA_DTM_SPI.hex
- **Step 4b.** Click on SWD tab
- **Step 4c.** Check on the “verify options” under SWD
- **Step 4d.** Check the interface in the “list of connected interfaces” available
- **Step 4e.** Click on “Flash”

Figure 13. Steps to Flash the DTM using RF-Flasher Utility tool



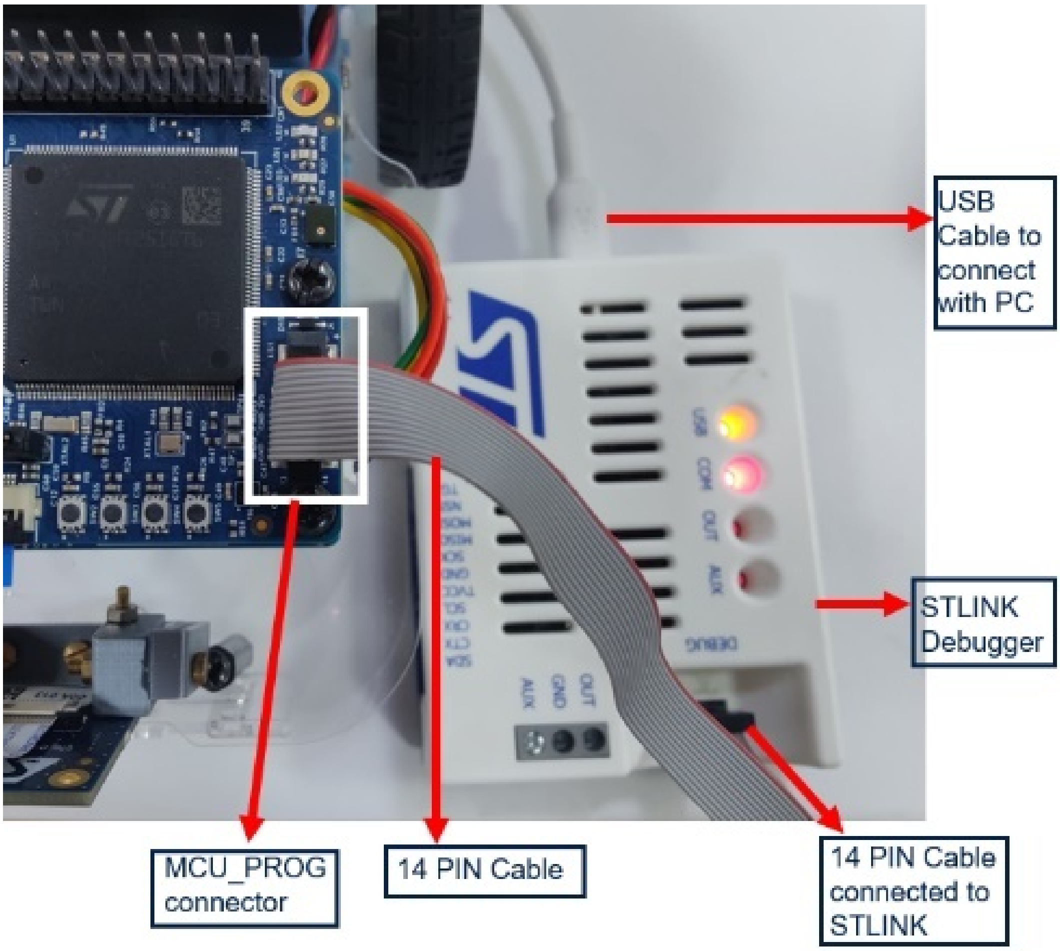
Step 5. If the DTM flash is successful then “Device Flash Programming -> SUCCESS” will be printed in the “Show log” Section

2.6.2 Setup for STEVAL-ROBKIT1-1

Setting up the STEVAL-ROBKIT1-1 involves the following steps:

Step 1. Plug the 14-pin cable to MCU_PROG connector (J1) on STEVAL-ROBKIT1-1 and connect it to the STLINK debugger as shown in Figure 14

Figure 14. Connection with STLINK Debugger

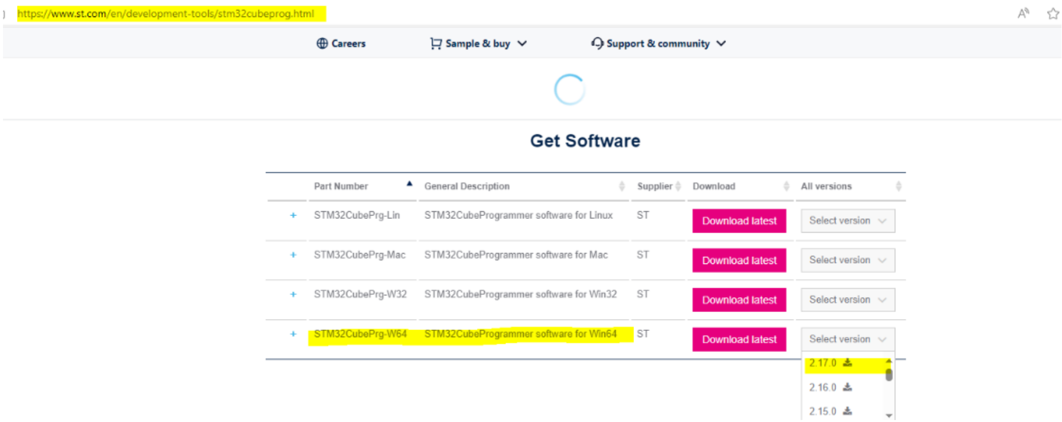


Step 2. Plug the STLINK debugger to PC via USB Type-A to Type-micro–B USB cable as shown in Figure 14

Step 3. Flash the provided testing firmware (STEVAL-ROBKIT1-1.hex) provided in the industrialization package. Follow the steps for flashing below:

Step 3a. Install the software with the given link STM32CubeProg - STM32CubeProgrammer software for all STM32 - STMicroelectronics by following the below Figure 15 for reference

Figure 15. Steps to download STM32Cube Programmer software

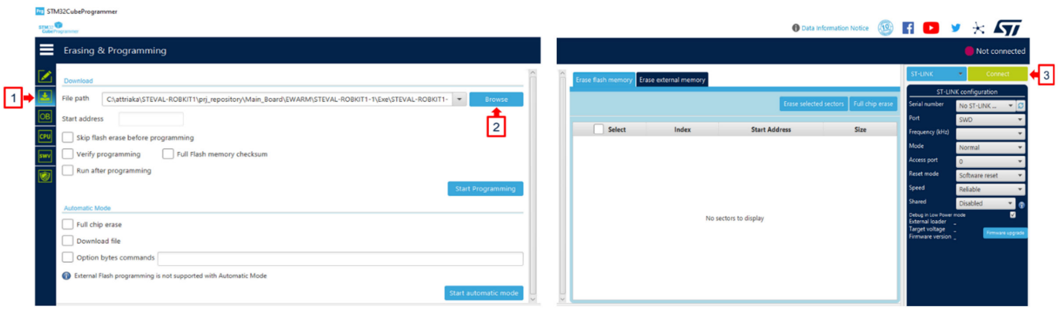


Step 3b. Follow Steps highlighted in Figure 16

- Select "Erase & Programming tab"
- Select the "STEVAL-ROBKIT1-1.hex" file using "Browse" option

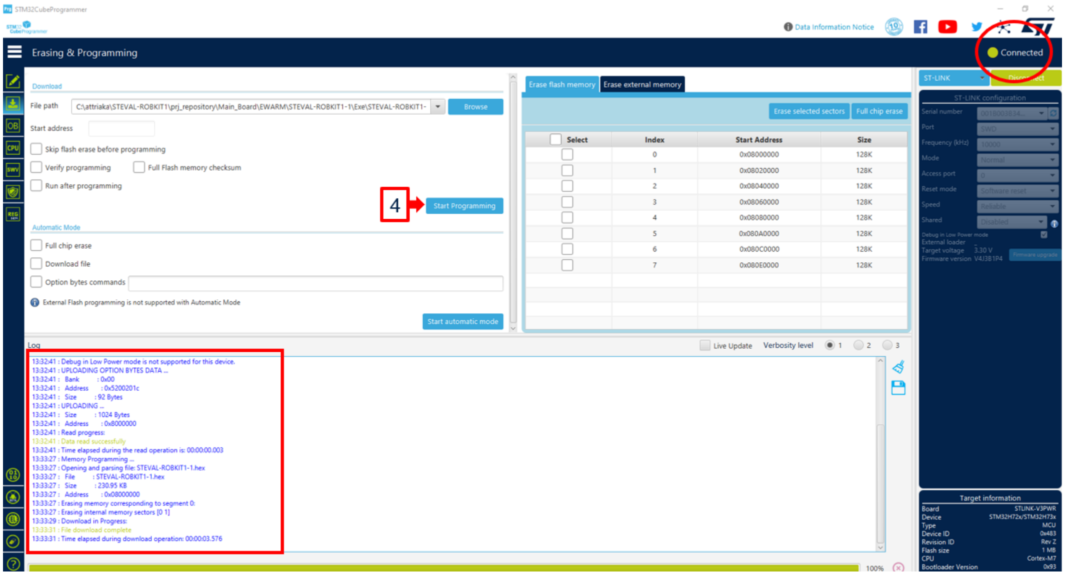
Step 4. Click on the "Connect" tab to establish connection between the software and the hardware as shown in Figure 16

Figure 16. STM32 Cube Programmer steps



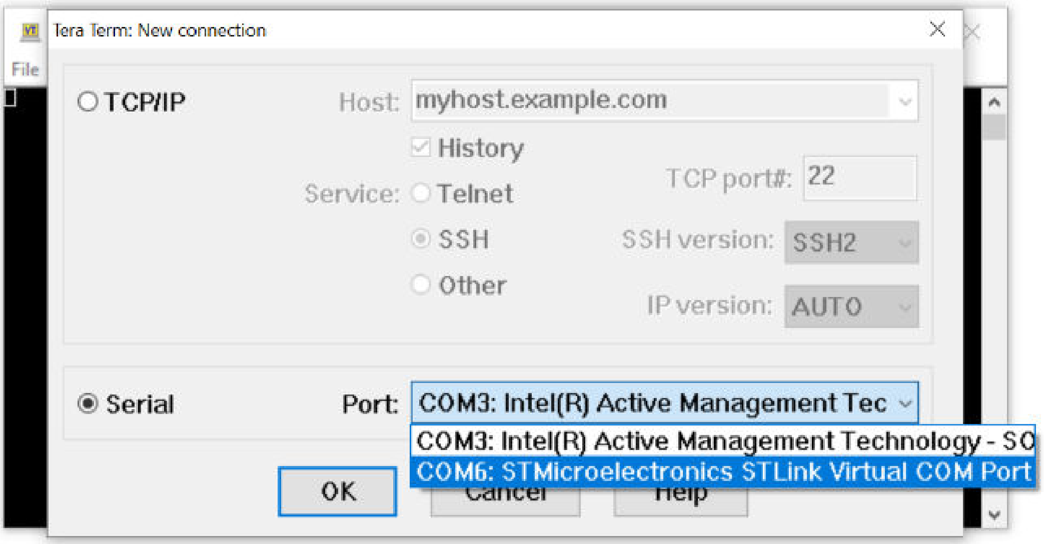
Step 5. Once the connection is established the "Connected" icon can be seen as highlighted in Figure 18. Click on Start Programming to flash STEVAL-ROBKIT1-1 board. If the flash is successful, then "File Download Complete" will be seen in the log section as highlighted in the Figure 17 below.

Figure 17. Steps to flash the binary in STEVAL-ROBKIT1-1 using STM32CubeProgrammer



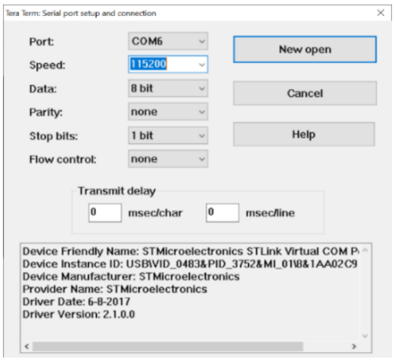
Step 6. Open 'Tera-Term' on Desktop --> select the "Serial" option --> from the drop down, note the COM port with "STMicroelectronics" name. For example, "COM6" in the Figure 18 below

Figure 18. Tera Term opening screen



Step 7. Press the OK button. From the Tera Term menu -> Setup -> Serial Port Select the COM port (noted from the previous step) Set speed to 115200 and click "New Open" This step will connect the user to the STEVAL_ROBKIT1-1. Refer Figure 19 for reference.

Figure 19. Tera Term configuration

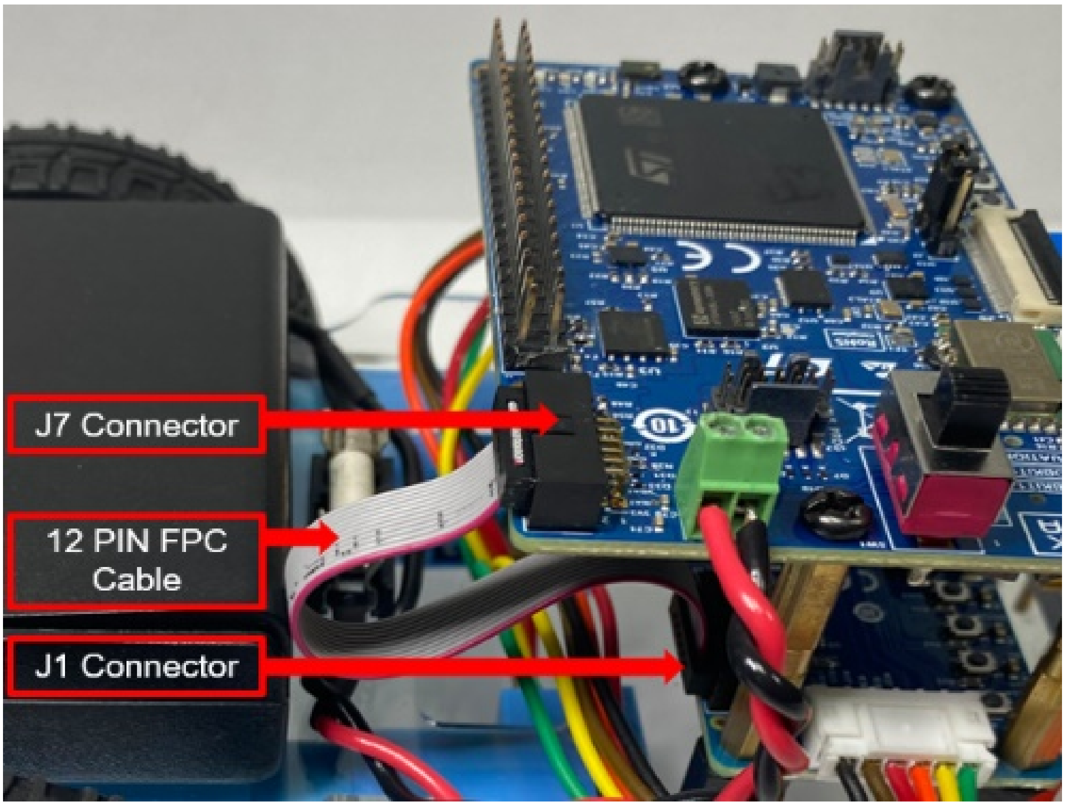


2.6.3 Setup for STEVAL ROBKIT1-2

Setting up the STEVAL-ROBKIT1-2 involves the following steps:

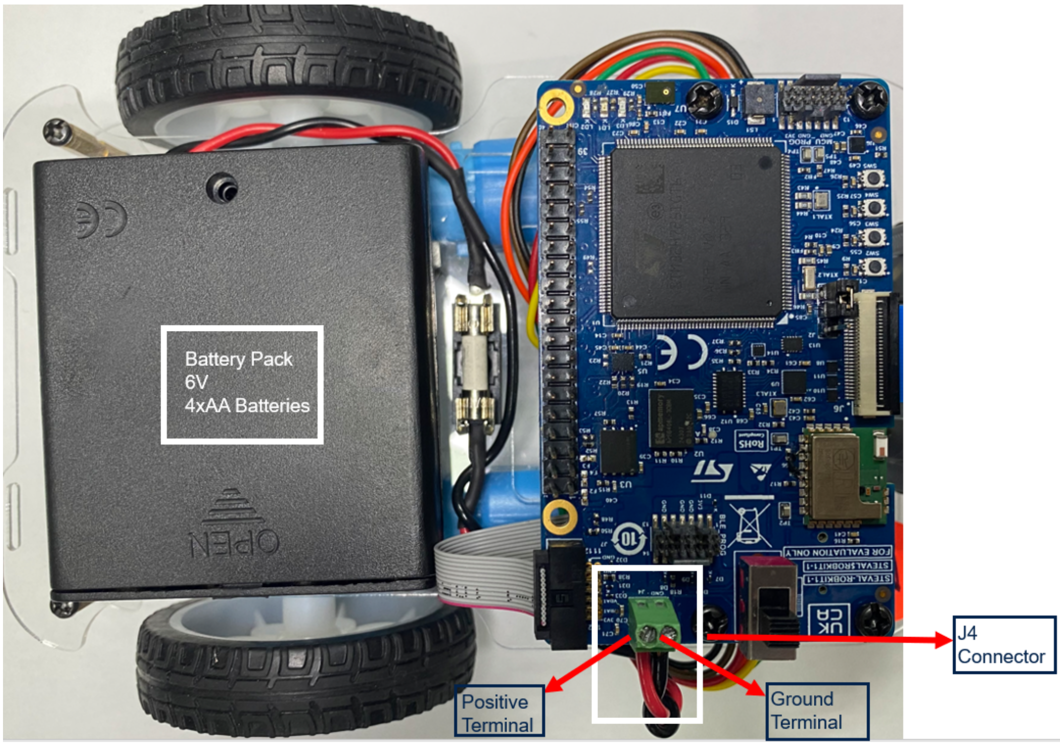
Step 1. Connect the STEVAL-ROBKIT1-2 board using the 12-pin ribbon cable. One end of the cable to be connected to the J7 of STEVAL-ROBKIT1-1 and the other end to the J1 connector of STEVAL-ROBKIT1-2 as highlighted in the Figure 20:

Figure 20. STEVAL-ROBKIT1-1 connected with STEVAL-ROBKIT1-2



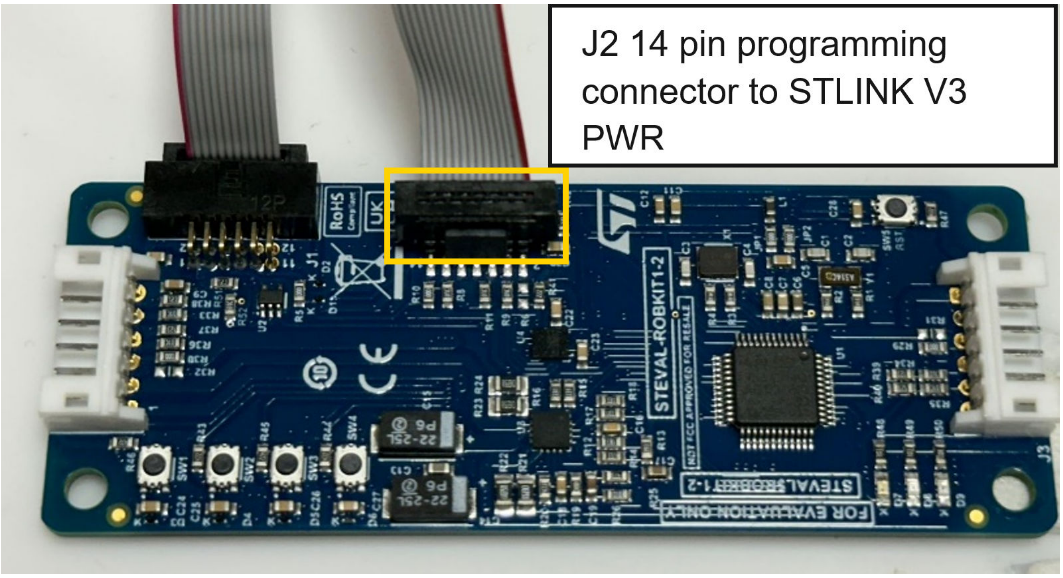
Step 2. Connect the STEVAL-ROBKIT1-1 with the power source using J4 connector on the board as shown in Figure 21. STEVAL-ROBKIT1-1 connected with the power supply

Figure 21. **STEVAL-ROBKIT1-1 connected with the power supply**



Step 3. Plug the 14-pin cable to MCU_PROG connector (J2) on STEVAL-ROBKIT1-2 and connect it to the STLINK debugger as shown in Figure 22

Figure 22. **Connection with STLINK Debugger**



Step 4. Plug the STLINK debugger to PC via USB Type-A to Type-C USB cable.

Step 5. Flash the provided testing firmware (STEVAL-ROBKIT1-2.hex) provided in the industrialization package. Follow the steps for flashing below:

Step 5a. Install the software with the given link STM32CubeProg - STM32CubeProgrammer software for all STM32 - STMicroelectronics

Step 5b. Follow Steps highlighted in Figure 16

- Select “Erase & Programming tab”
- Select the “STEVAL-ROBKIT1-2.hex” file using “Browse” option

Step 6. Click on the “Connect” tab to establish connection between the software and the hardware as shown in Figure 17

Step 7. Once the connection is established the “Connected” icon can be seen. Click on Start Programming to flash STEVAL-ROBKIT1-2 board. If the flash is successful, then “File Download Complete” will be seen in the log section.

Step 8. Open ‘Tera-Term’ on Desktop → select the “Serial” option → from the drop down, note the COM port with “STMicroelectronics” name.

Step 9. Press the OK button. From the Tera Term menu → Setup → Serial Port Select the COM port (noted from the previous step) Set speed to 115200 and click “New Open” This step will connect the user to the STEVAL-ROBKIT1-2. Refer Figure 18 for reference

Note: The imaging board must be kept perpendicular to the surface since the applications mentioned in Section 3 are developed based on VL53L8CX sensor kept with TILT_ANGLE == 0

3 Applications

The STSW-ROBKIT1 contains:

- Two application projects for main board, STSW-ROBKIT1-1 in the STSW-ROBKIT1\STSW-ROBKIT1-1\Projects directory.
  The details of Robkit_Control and AI_Surface_Detection projects are present in Section 3.1 and Section 3.2.
- One application project for motor board, STSW-ROBKIT1-2 in the STSW-ROBKIT1\STSW-ROBKIT1-2\Projects directory.
  The details of Robkit_MotorCtrl project are present in Section 3.3.

Note: The application project for motor board to be used with both the application projects of main board

The STSW-ROBKIT1-1 firmware uses FreeRTOS™ to enable real-time multitasking, ensuring efficient management of sensor data acquisition, BLE communication, and robot control.

The system runs primary tasks with assigned priorities specific to the projects as shown in Table 2:

Table 2. Task overview in STSW-ROBKIT1-1

Task Name	Role	Stack Size	Priority	Projects	
				AI_Surface_Detection	Robot_Control
OS_Task_Default	Background/idle task	512 bytes	Below Normal	✓	✓
OS_Task_UI	User interface and BLE event handling	4096 bytes	Above Normal	✓	✓
OS_Task_Snsr	Sensor data acquisition and queuing	2048 bytes	Normal	✓	✓
OS_Task_Odomtry	Odometry data processing	2048 bytes	Normal	✓	✓
OS_Task_App	Periodic application tasks (e.g., battery check)	2048 bytes	Below Normal	✓	✓
OS_Task_AI	AI inference processing	512 bytes	Below Normal	✓	✗

These tasks communicate and synchronize through message queues, semaphores, and timers to coordinate sensor readings, data transmission, and control commands. The firmware also includes standard FreeRTOS hooks for system stability and error handling.

This multitasking architecture provides a robust and responsive platform for the STSW-ROBKIT1-1, enabling seamless integration of robotics functionalities.

3.1 STSW-ROBKIT1-1, main board application project: Robkit_Control

3.1.1 BLE Motion Sensor Application

The BLE motion sensor demo application in STRobotics app is supported for the STEVAL-ROBKIT1 Robotics kit. This application shows how to implement motion sensor demo custom profile application tailored for interacting with the STRobotics smartphone app. Once configured and connected, the BlueNRG-M2SA device on the main board: STEVAL-ROBKIT1-1 sends the data collected from the motion sensor (Accelerometer, Gyroscope and Magnetometer) to the STRobotics smartphone app, which displays this information.

Running the Application:

Two versions (Android and iOS) of the smartphone STRobotics app are available for download.

Step 1. Install the app and launch it.

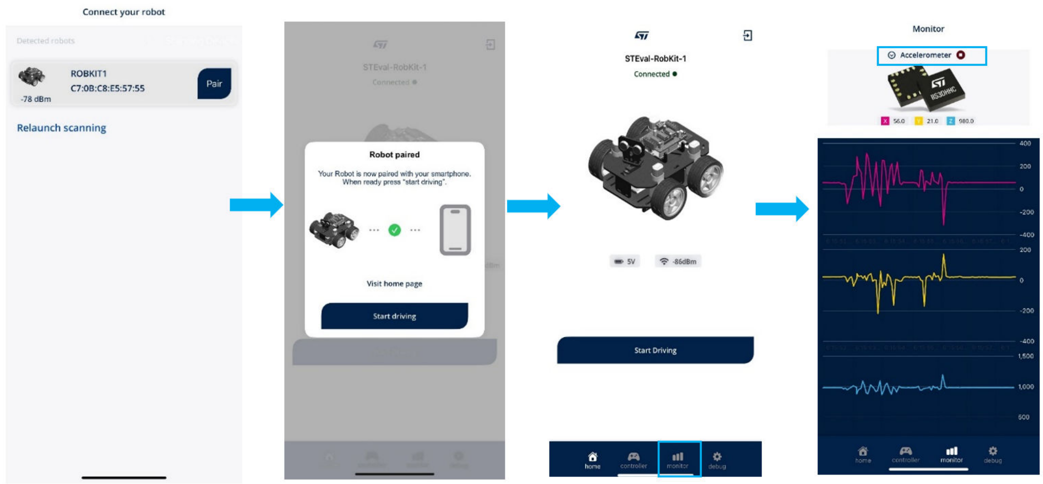
Note: Refer to the user manual STRobotics app, section 3 for detailed download and app installation procedure.

Step 2. Turn on the Robotics Kit: STEVAL-ROBKIT1. The app starts scanning for the peripheral device. A device called “ROBKIT1” appears on the screen.

Step 3. Select the “STEVAL-ROBKIT1” name and connect to the selected platform. The ST BLE Sensor app enables notifications on the motion characteristic (Accelerometer, Gyroscope, Magnetometer).

Step 4. On the home page select the “Monitor” option and select sensor options to plot the received values. The sensor values are displayed on a graphical chart.

Figure 23. Expected output on STRobotics App for motion sensor data



Note: For the details of using the mobile application, Refer to STRobotics app User manual.

3.1.2 BLE Remote control application

The BLE remote control demo application in STRobotics app is supported for the STEVAL-ROBKIT1 Robotics kit. This application is designed to operate the connected Robotic Kit. In order to operate the kit there are two joysticks designed in the smartphone app, one is the “Throttle Joystick”, on the left side of the screen which is used to move the robot in forward and backward direction based on user inputs and the other one is, “Direction Joystick”, on the right side of the screen, which is used to control the direction of the robot based on the angle provided by rotating the joystick wheel.

Running the Application:

Step 1. Turn on the Robotics Kit: STEVAL-ROBKIT1. The app starts scanning for the peripheral device. A device called “ROBKIT1” appears on the screen.

Step 2. Select the “ROBKIT1” name and connect to the selected platform. The STRobotics app enables notifications on the Robot Movement characteristic.

Step 3. On the home page select the “Controller” option and select “drive” on the screen to operate the Robot using the joysticks as indicated in the Figure 24. Expected output on STRobotics App for Robotic Movement: Remote Control.

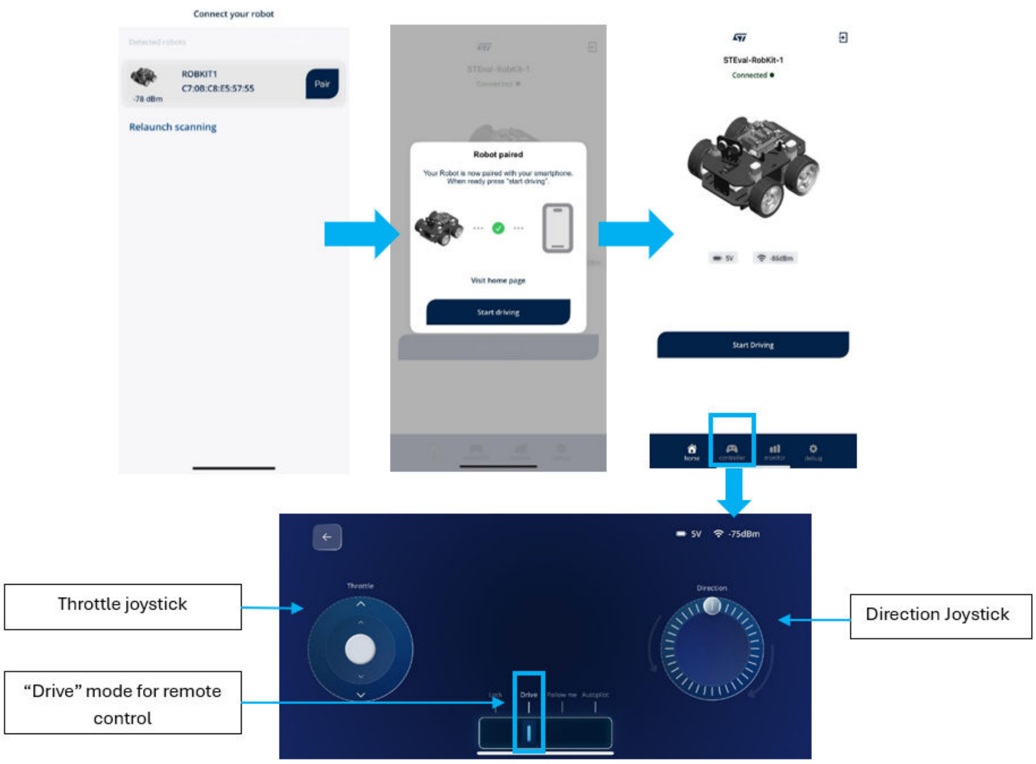


Figure 24. **Expected output on STRobotics App for Robotic Movement: Remote Control**

3.1.3 Free navigation feature

The Free Navigation feature is an advanced functionality supported by the STEVAL-ROBKIT1 Robotics kit. This feature enables the robotic kit to autonomously navigate its environment without any manual user inputs. The robot is capable of avoiding obstacles, detecting edges and cliffs, and changing its direction independently, ensuring safe and efficient movement in various environments.

The free-navigation feature can be selected through the smartphone app: STRobotics app by selecting the “Autopilot” feature as shown in the Figure 25.

Figure 25. **Autopilot mode for free navigation feature in STRobotics app**

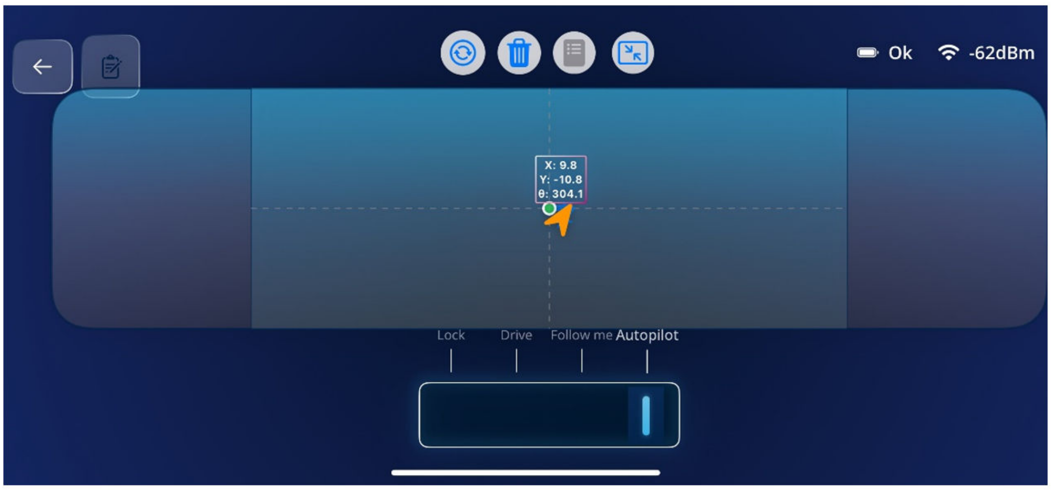
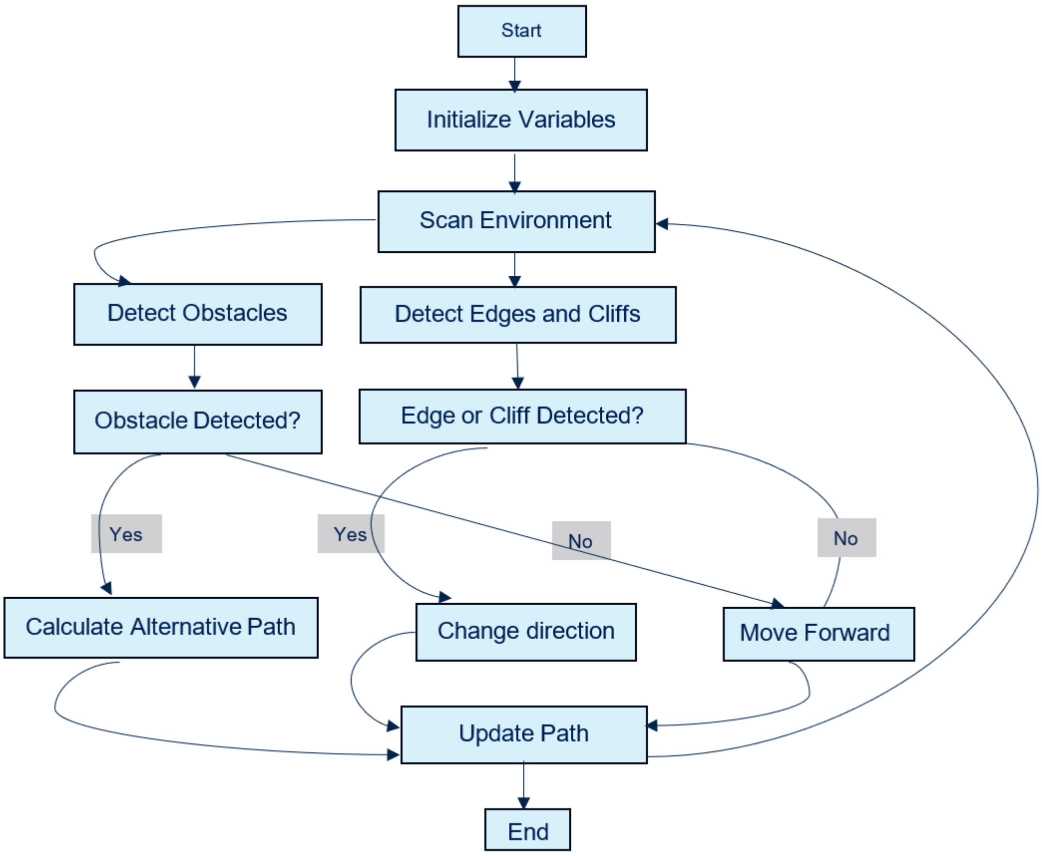


Figure 26. **The high-level overview of the free navigation feature**



3.1.4 Follow Me navigation feature

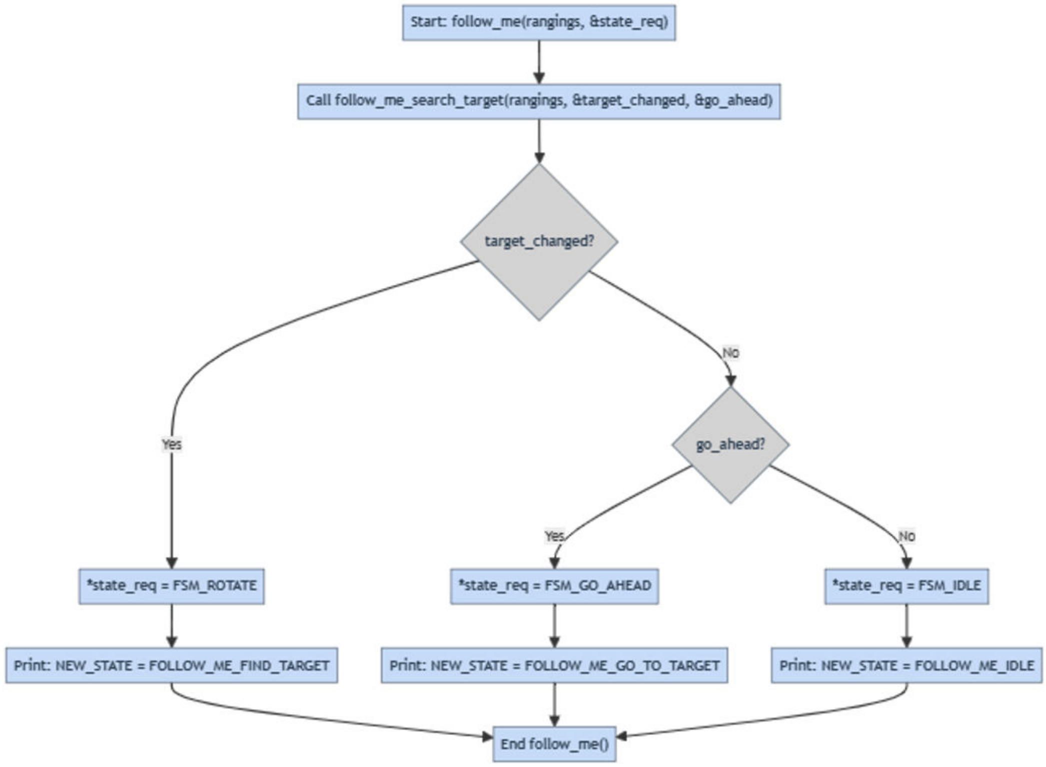
The Follow Me feature enables the STEVAL-ROBKIT1 robot to autonomously track and follow a target using its onboard Time-of-Flight (ToF) sensor. The sensor provides distance measurements by calculating the time that it takes for emitted light pulses to reflect off objects and return, enabling detection of the target's position in real time.

The detailed working of the feature is as follows:

- The robot continuously collects distance data from a ToF sensor covering a wide field of view.
- The Follow Me algorithm processes this multizone distance data to identify the closest target within the sensor range.
- It calculates the angle to the target and the distance to determine the appropriate navigation commands.
- Based on the target's position and proximity, the robot decides whether to:
  - Rotate to face the target if the angle deviates from the forward direction.
  - Move forward to approach the target if it is beyond a minimum safe distance.
  - Remain idle if the target is centered and within the desired range.

Note: When enabling the FollowMe feature, keep the target approximately 10 cm in front of the kit to ensure proper target locking. Since this kit is equipped with only one ToF sensor, it is designed to follow slow-moving objects and may occasionally lose track of the target.

Figure 27. The high-level overview of the Follow Me navigation feature



3.1.5 Scene description for ToF data visualization feature

The scene description feature is a Bluetooth® LE service designed to transmit structured data, such as Time-of-Flight (ToF) sensor readings, in real-time using JSON format. It initializes a Bluetooth® LE characteristic with a notify property, enabling clients to receive updated data seamlessly. ToF data is retrieved, formatted into an 8x8 grid, serialized into a JSON object, and transmitted in chunks using a transport protocol to ensure compatibility with Bluetooth® LE's MTU. The middleware manages notifications, write requests, and characteristic updates, while the application code handles data formatting, serialization, and memory allocation. This feature allows users to monitor the kit's distance from objects in real time via a mobile application, enhancing usability and monitoring efficiency.

Navigate to the monitor screen of the STRobotics mobile application and select the 'ToF' option from the drop-down menu as highlighted in Figure 28. Utilize the color indicators and distance indicated in the matrix (in mm) for monitoring the ToF data in real time.

Figure 28. Scene description in STRobotics app

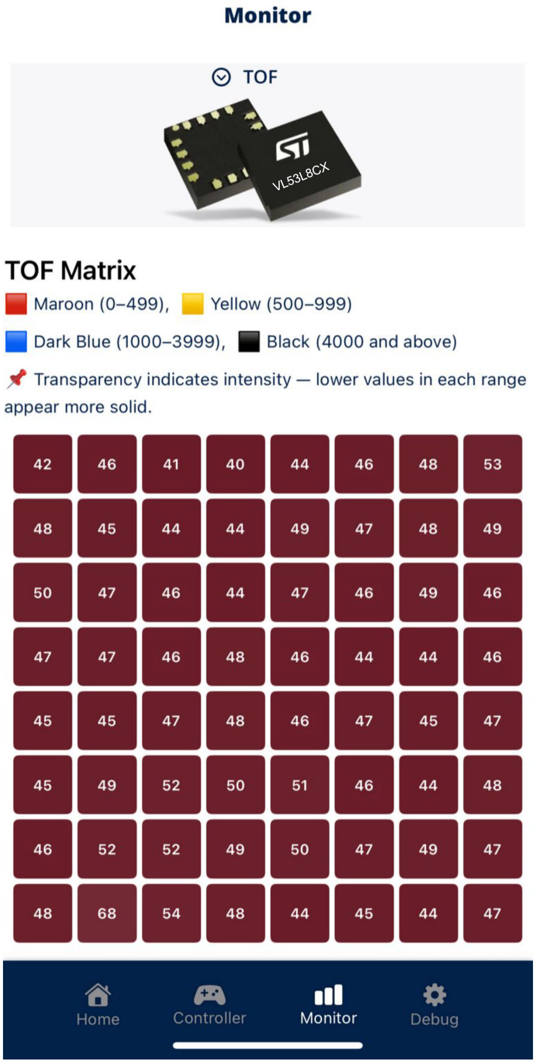
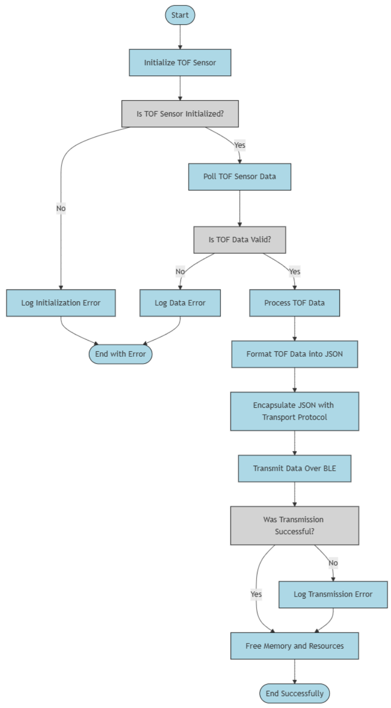


Figure 29. The high-level overview of the scene description feature



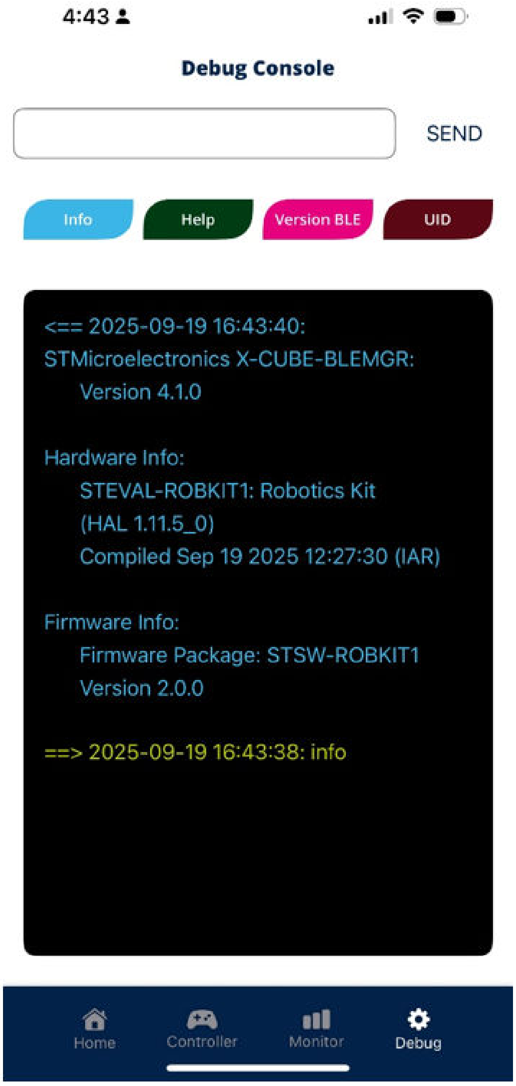
3.1.6 Debug console feature

The debug console feature in the Bluetooth® LE manager allows users to interact with the board via Bluetooth® LE by sending specific commands for debugging and control. It supports commands like:

- **‘help’**: lists available commands
- **‘info’**: provides system and firmware details
- **‘versionBle’**: returns Bluetooth® LE firmware version
- **‘uid’**: displays the STM32 microcontroller's unique identifier

Additionally, single string motor control commands such as ‘F’, ‘B’, ‘S’, L, R, P, and Q enable users to control the kit's movement (for example, forward, backward, stop, turn). Users can also extend this feature by creating their own custom commands to control the kit’s movement as per their requirements, making it a versatile feature for debugging and customization.

Figure 30. Debug console feature in STRobotics app



3.2 STSW-ROBKIT1-1, main board application project: AI_Surface_Detection

The Surface Detection feature enables the STEVAL-ROBKIT1 robot to autonomously identify the type of surface it is moving over-smooth or rough-using onboard accelerometer data processed by a trained AI model. This functionality enhances the robot's environmental awareness and can be used to adapt its behavior based on terrain conditions.

Key steps of the feature:

- Accelerometer data is collected from the STEVAL-ROBKIT1-1.
- A neural network model processes this data to classify the surface type.
- Based on the prediction:
  - **Green LED** lights up for smooth surfaces.
  - **Red LED** lights up for rough surfaces.

This project showcases end-to-end AI integration-from data collection and model training to deployment and real-time inference-using STM32CubeIDE and STM32Cube.AI tools

a. Dataset Information:

- **Sensor Used**: LSM6DSV16BX (Accelerometer)
- **Output Data Rate (ODR)**: 104 Hz
- **Full Scale Range**: ±2 g

- **Samples Collected**: 29 samples per class, approximately 1000 data lines per sample

b. Model Information:

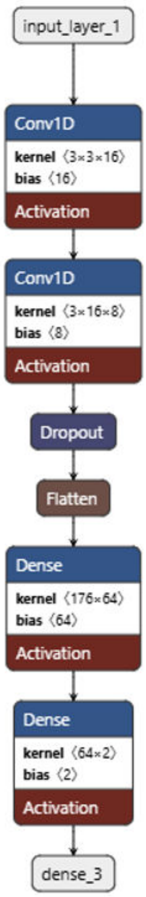
The AI model is a lightweight 1D Convolutional Neural Network (Conv1D) designed for time-series classification. It processes accelerometer data to distinguish between smooth and rough surfaces.

- **Model Size**:
  - RAM: 14 KB
  - Flash: 248.54 KB
- **Activation Memory**: 11.19 KB
- **Inference Time**: ~1.5 ms

c. Model Creation and Training

- Model developed using surface_detection.ipynb in Google Colab.
- Dataset imported and split into training/testing sets.
- The Conv1D architecture is detailed in Figure 31.
- Training includes accuracy and loss evaluation, with results visualized in Figure 32 and Figure 33 (training graphs and confusion matrix).
- The trained model is exported as a TensorFlow Lite (.tflite) file for STM32 deployment.

Figure 31. CNN 1D layers of the model



The model was trained using TensorFlow and converted for STM32 deployment using X-CUBE-AI. It is optimized for real-time inference on the STEVAL-ROBKIT1-1. The training accuracy and training loss can be depicted from the graph as in Figure 32 and the results of the trained model through the confusion matrix are shown in Figure 33:

Figure 32. Graph for training/ validation accuracy and training/validation loss

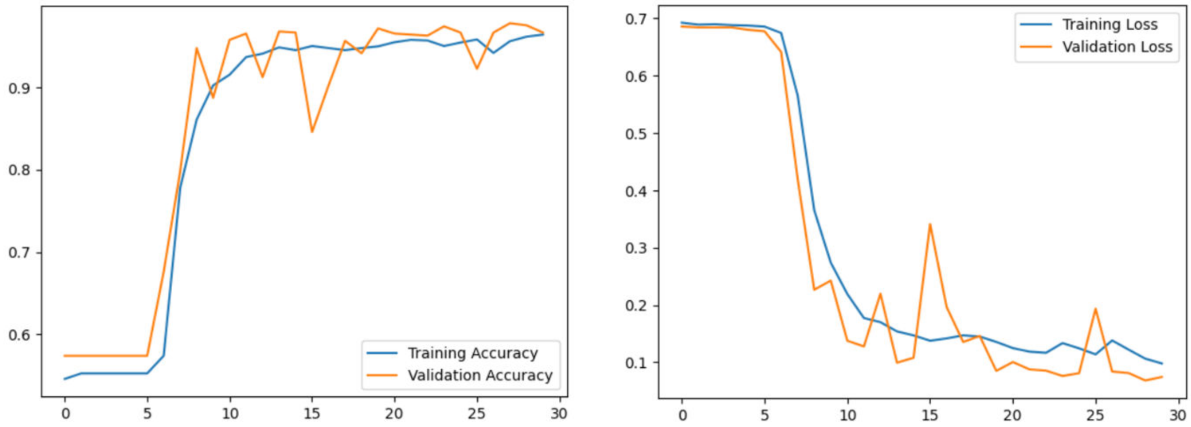
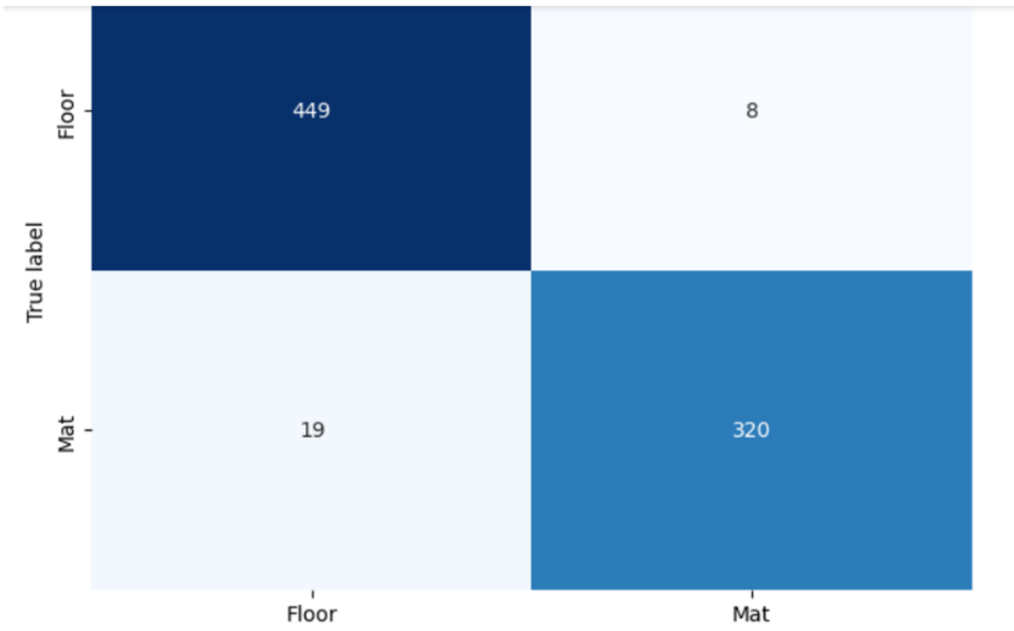


Figure 33. Results of trained model through Confusion Matrix



d. AI Model Preparation

- Use STM32Cube.AI to integrate the pre-trained AI model for surface detection.
- Ensure the model is in TensorFlow Lite (.tflite) format.

e. STM32CubeMX Configuration

- Open the .ioc file of the STEVAL-ROBKIT1 project.
- Navigate to Software Packs → Select Components.
- Download and enable X-CUBE-AI under the Artificial Intelligence category.
- In the Middleware and Software Packs, select X-CUBE-AI.
- Add a network:
- – Choose TFLite as the model format.
  – Browse and select the AI model file.
  – Click Analyze and then Generate Code.

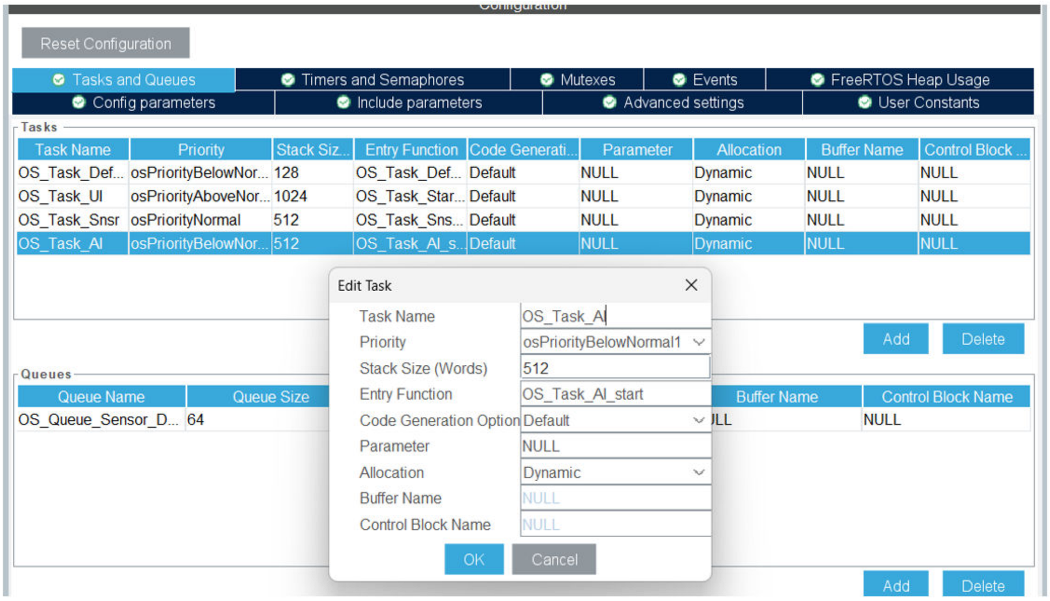
f. FreeRTOS Task Creation for AI Inference

To execute the AI inference within a FreeRTOS environment, follow these steps

- Ensure FreeRTOS Middleware is enabled in STM32CubeMX

- In the FreeRTOS configuration panel, add a new task for AI Inference with an appropriate stack size and priority as shown in Figure 34.

Figure 34. **STM32CubeMX task settings for AI inference**



- Click ok to add the task.
- Click "Generate Code" in STM32CubeMX to create the FreeRTOS task skeleton in your project files.
- Ensure the following files are generated and added in the project:
- **Src Folder**: network_data.c, network.c, freertos.c
- **Inc Folder**: network_data.h, network.h, network_config.h, freertos.h, FreeRTOSConfig.h
- **Middleware/ST/AI**: AI headers and NetworkRuntime1010_CM7_GCC.a library

g. Code Implementation:

- Key functions (AI_Init(), AI_Run(), argmax(), set_led_color()) are defined in freertos.c.
- Use argmax() to determine the predicted surface class.
- Use set_led_color() to toggle LEDs:
- – Green LED for **smooth (mat)** surface.
  – Red LED for **rough (floor)** surface.
- Declare AI_Init() as extern in main.c and call it during system initialization before starting the FreeRTOS scheduler.
- Implement the AI inference logic inside the dedicated AI FreeRTOS task:
- – Continuously acquiring accelerometer data.
  – Call AI_Run() within the task loop to perform inference.
  – Process inference results and update LED indicators accordingly.

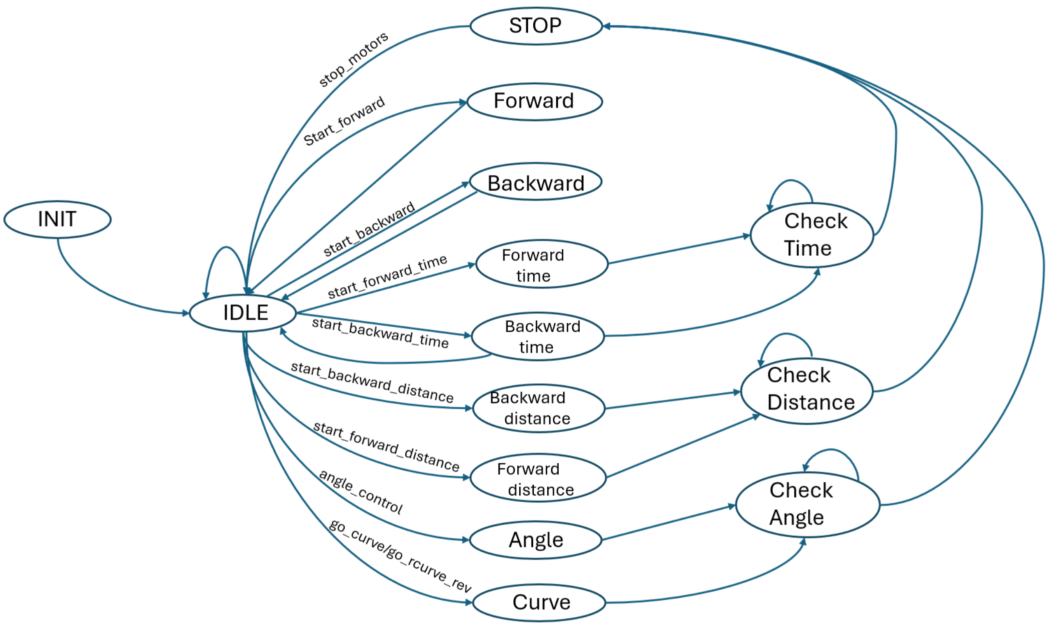
This AI_Surface_Detection project exemplifies seamless integration of AI capabilities on embedded STM32 platforms, enabling the STEVAL-ROBKIT1 robot to intelligently adapt to its environment in real time.

3.3 STSW-ROBKIT1-2, motor board application project: Robot_MotorCtrl

3.3.1 State machine for STEVAL-ROBKIT1-2: motor board

The flowchart below represents a state machine for STEVAL-ROBKIT1-2. It begins in the INIT state where all the peripheral initialization happens and then transitions to an IDLE state from where, it can execute various commands or checks. Each command or check returns the system to the IDLE state, ensuring it is ready for the next command. The design emphasizes simplicity in adding new commands and quick responsiveness to external events.

Figure 35. State Machine for STSW-ROBKIT1-2



States and Transitions:

1. **INIT (Initial State)**
   - The system starts in the INIT state all the peripheral initialization happens in this state.
   - It transitions to the IDLE state.
2. **IDLE (Idle State)**
   - This is the central state from which various commands can be initiated.
   - From the IDLE state, the system can transition to the following states using corresponding commands:
     - **Forward:** “start_forward”
     - **Backward:** "start_backward"
     - **Forward time:** “start_forward_time”
     - **Backward time:** “start_backward_time”
     - **Forward distance:** “start_forward_distance”
     - **Backward distance:** “start_backward_distance”
     - **Angle:**“angle_control”
     - **STOP:** “stop_motors”
     - **Clear odometry:** “clear_odometry”
     - **CHECK TIME**
     - **CHECK DIST**
     - **CHECK ANGLE**
     - **Curve:** "go_curve"
     - **Reverse Curve:** "go_rcurve_rev"
3. **Command States**
   - **“clear_odometry”:** Command to clear the odometry data.
   - **“start_forward”:** Command to move forward infinitely.
   - **“start_backward”:** Command to move backward infinitely.
   - **“start_forward_time”:** Move forward for a specific amount of time.
   - **“start_backward_time”:** Move backward for a specific amount of time.
   - **“start_forward_distance”:** Move forward for a specific distance.
   - **“start_backward_distance”:** Move backward for a specific distance.
   - **“angle_control”:** This command rotates the robot by specified angle.
   - **“stop_motors”:** This command stops the rotation of both motors.
     Each of these command states transitions back to the IDLE state after execution.
   - **"go_curve":** This command moves the robot making a curve towards the left or right direction based on the angle provided. Positive angle will make right curve and negative angle will make left curve.
   - **"go_rcurve_rev":** This command moves the robot making a reverse curve towards the left or right in reverse direction based on the angle provided.
     Positive angle will make reverse right curve and negative angle will make reverse left curve.
4. **STOP**
   - This state represents the stop after the required action is achieved (**distance time and angle)**.
   - The motors are stopped and the robot transitions back to the IDLE state.
5. **CHECK States**
   - **CHECK TIME**: Check the time-related parameter
   - **CHECK DIST**: Check the distance-related parameter
   - **CHECK ANGLE**: Check the angle-related parameter

Each of these states transitions back to the IDLE state after execution

3.3.2 Odometry feature

Overview

The STEVAL-ROBKIT1 Robotics kit is equipped with an odometry system that continuously tracks the robot’s position and orientation as it moves. This system enables precise monitoring and visualization of the robot’s path, which can be viewed in real-time on the companion STRobotics mobile app.

Details on Odometry:

- Wheel Encoders:
  The kit uses high-resolution encoders on both the left and right wheels to measure the distance each wheel travels.
- Position Calculation:
  The motor board processes encoder data to calculate the robot's current position (x, y) and orientation (theta) using a mathematical model known as the unicycle model. This model estimates the robot's movement based on the difference and average of the wheel rotations.
- Data Transmission:
  The calculated odometry data is sent from the motor board to the main board. The main board then attaches a timestamp to each data packet and transmits this information to the mobile app via a dedicated Bluetooth Low Energy (BLE) Odometry service periodically inside task "OS_Task_Odomtry" of FreeRTOS.

Odometry Bluetooth Low Energy data packet Structure

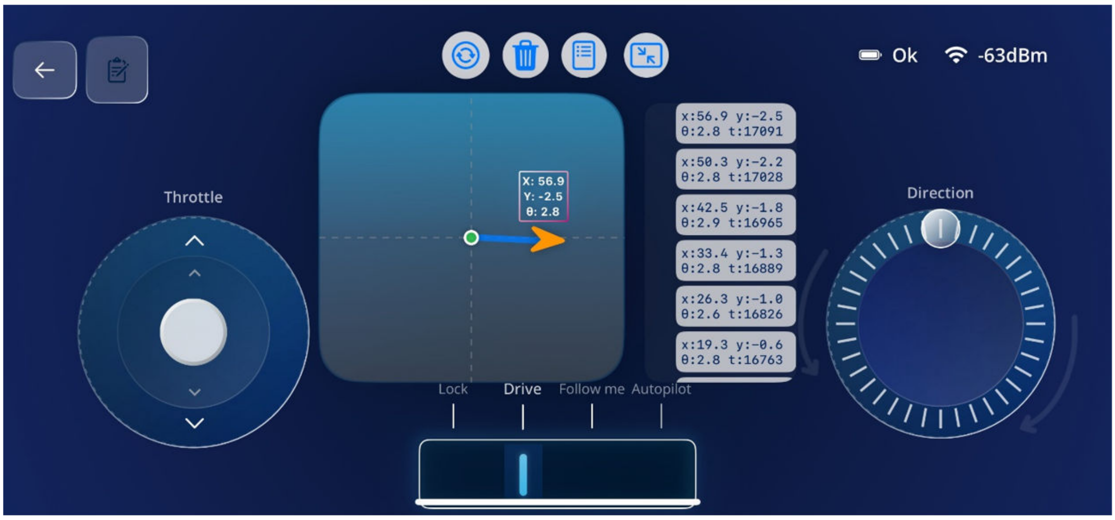
Each odometry data packet (total 14 bytes) sent to the mobile app contains:

- Timestamp (Size 2 Bytes): The exact time the data was recorded, ensuring accurate path reconstruction.
- X Position (x) (Size 4 Bytes): x-coordinate of the center of mass of the robot (in cm).
- Y Position (y) (Size 4 Bytes): y-coordinate of the center of mass of the robot (in cm).
- Orientation (theta) (Size 4 Bytes): The robot's heading angle (in degrees).

Visualization in the STRobotics Mobile App

- The mobile app receives odometry data in real-time through the "odometry" Bluetooth® Low Energy service.
- The app plots the robot's movement path on a graphical interface, allowing users to visualize the trajectory and orientation of the robot as it navigates its environment.
- This feature is especially useful for debugging, performance analysis, and educational demonstrations.
- The Odometry feature can be selected through the smartphone app: STRobotics app by selecting the "Drive Mode" feature as shown in the below figure:

Figure 36. Odometry feature demonstration in STRobotics Mobile application



Benefits

- Accurate Path Tracking:
  The odometry system provides reliable and precise tracking of the robot's movement, even in environments without external positioning systems.
- Real-Time Feedback:
  Users can monitor the robot's path live on the mobile app, making it easy to observe and analyze the robot's behavior.

Notes

- The odometry calculations are based on the incremental changes in wheel encoder ticks, which are periodically reset after each update to ensure accuracy.
- The orientation angle (theta) is normalized to remain within a standard range, preventing errors due to angle overflow.
- The system assumes a standard coordinate system, with the X and Y axes defined relative to the robot’s starting position.

Tip: *For best results, ensure the robot operates on a flat, obstacle-free surface and that the wheel encoders are free from dust and debris.*

4 Firmware customizations

While regenerating the existing project using the .ioc of the respective project, the users need to make sure to do the following firmware changes:

- **BLE_Implementation.c and Header File:**
  These files are generated by STM32CubeMX and handle Bluetooth initialization, enabling standard and custom BLE services. Currently, the robotics application-specific custom feature files are not included in the X-CUBE-BLEMGR custom services. Additionally, the robotics kit platform is not yet registered as a platform in the Cube package. Therefore, the code changes highlighted in Section 4 and Figure 38 must be applied manually to initialize these custom robotic application services and enable the robotics kit to operate with a unique platform.

Figure 37. Implementation customization in BLE_Implementation.c

```
void ble_init_custom_service(void)
{
  /**
    * User can added here the custom service initialization for the selected BLE features.
    * For example for the environmental features:
  */
  /* Characteristc allocation for the battery features */
  ble_manager_add_char(ble_init_battery_service());
  /* Characteristc allocation for inertial features */
  ble_manager_add_char(ble_init_inertial_service(ENABLE_ACC_DATA,
                                                 ENABLE_GYRO_DATA,
                                                 ENABLE_MAG_DATA));
  /* Characteristc allocation for machine learning core features */
  ble_manager_add_char(ble_init_machine_learning_core_service(BLE_MLC_8_REG));
  /* Characteristc allocation for Binary Content features */
  ble_manager_add_char(ble_init_binary_content_service());
  /* USER CODE BEGIN 1 */
  /* Characteristc allocation for Odometry features */
  ble_manager_add_char(ble_init_odometry_service());
    /* Characteristc allocation for Robot Movement features */
  ble_manager_add_char(ble_init_robot_movement_service());
  /* Characteristc allocation for Scene Description features */
  ble_manager_add_char(ble_init_scene_description_service());
  /* USER CODE END 1 */
}
```

Figure 38. Implementation customization in BLE_Implementation.h

```
/*USER CODE 0*/
#define STEVALROBKIT_ID                   0xC3
/* Used platform: to be updated in CubeMX */
#define BLE_MANAGER_USED_PLATFORM         STEVALROBKIT_ID
/*USER CODE END 0 */
```

- **FreeRTOS Files for Keil and STM32CubeIDE:**
  When regenerating the project for Keil, FreeRTOS-specific files are generated in the following directory: Middlewares/Third-Party/FreeRTOS/Source/portable/RVDS/ARM_CM4F

However, when regenerating the project using STM32CubeIDE (which uses the GCC compiler), the FreeRTOS files are generated in:

Middlewares/Third-Party/FreeRTOS/Source/portable/GCC/ARM_CM4F

To ensure proper FreeRTOS functionality, replace the RVDS files with the GCC files after regenerating the project in STM32CubeIDE.

This is a known limitation of the FreeRTOS package, and applying this workaround ensures stable FreeRTOS operation.

Revision history

Table 3. Document revision history

Date	Revision	Changes
18-Dec-2024	1	Initial release.
01-Aug-2025	2	Updated Figure 1. STEVAL-ROBKIT1 Robotic application evaluation kit , Figure 7. High Level Firmware architecture of Robotics evaluation kit, Section 2.3.1: STSW-ROBKIT1-1: firmware for Main Board, Section 2.6: Board setup, Figure 25. Autopilot mode for free navigation feature in STRobotics app and Section 3.3.1: State machine for STEVAL-ROBKIT1-2: motor board. Added Section 3.1.5: Scene description for ToF data visualization feature, Section 3.1.6: Debug console feature, Section 3.1.4: Follow Me navigation feature and Extended configuration feature.
07-Oct-2025	3	Updated Section Introduction, Section 2: Getting started and Section 3: Applications. Added Section 4: Firmware customizations. Minor text changes.

Contents

List of tables

List of figures