UM3456

User manual

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, the Motor control board, and the Imaging board. 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 Main board and Motor board
- Customized protocol for seamless board communication
- Odometry for precise navigation
- DCMI interface for versatile camera integration
- 8x8 multizone Time of Flight sensor data
- External Flash and PSRAM for efficient data management
- Monitor and control via STRobotics mobile app for android and iOS platforms
- BLE Remote Control and real-time sensor data plotting
- LEDs and Buzzer for error alerts

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

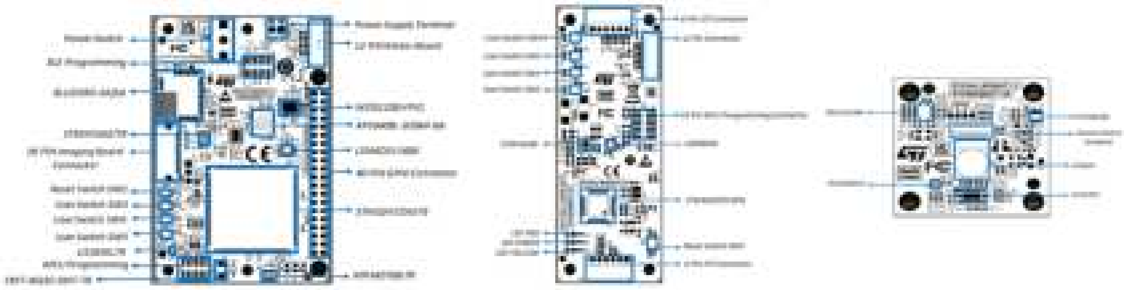
Users can program the STEVAL-ROBKIT1-1 (Main Board), STEVAL-ROBKIT1-2 (Motor Board) and BlueNRG-M2SA module on the main board to run following using STRobotics mobile application:

- Remote Controller
- Free Navigation
- Follow-Me Navigation
- Object Detection
- Edge/Cliff Detection
- QR Code Scanner
- Data plot for the on-board magnetometer, gyroscope, accelerometer and ToF
  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.

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

- BlueNRG-M2SA
- LSM6DSV16X (iNEMO 3D accelerometer & 3D gyroscope)
- LPS22DF (Pressure Sensor)
- STTS22H (Temperature Sensor)
- LIS2MDL (Magnetic sensor)
- LIS2DU12 (Ultra-low-power 3-axis linear accelerometer)

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:

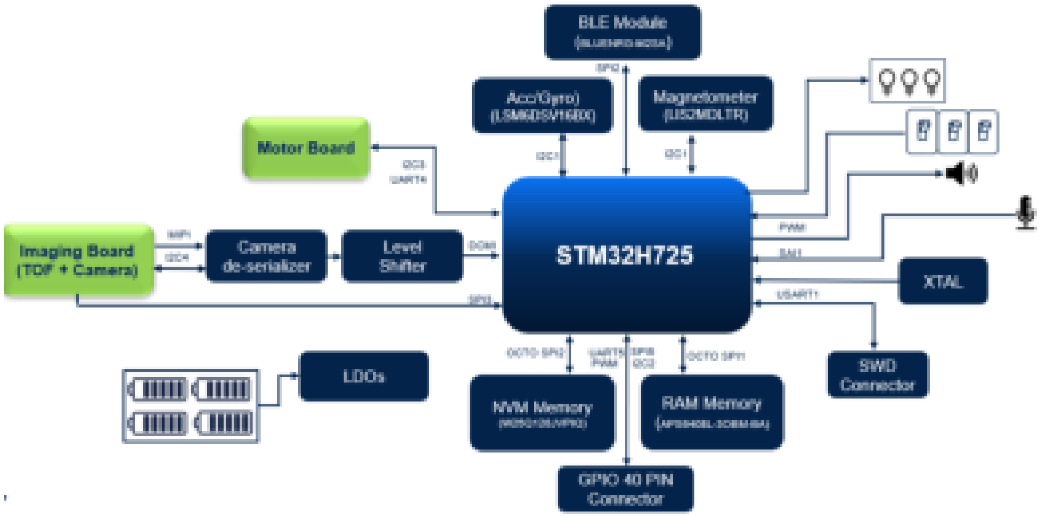


Figure 4. **Block diagram of STEVAL ROBKIT 1-1**

The block diagram of the motor board is as follows:

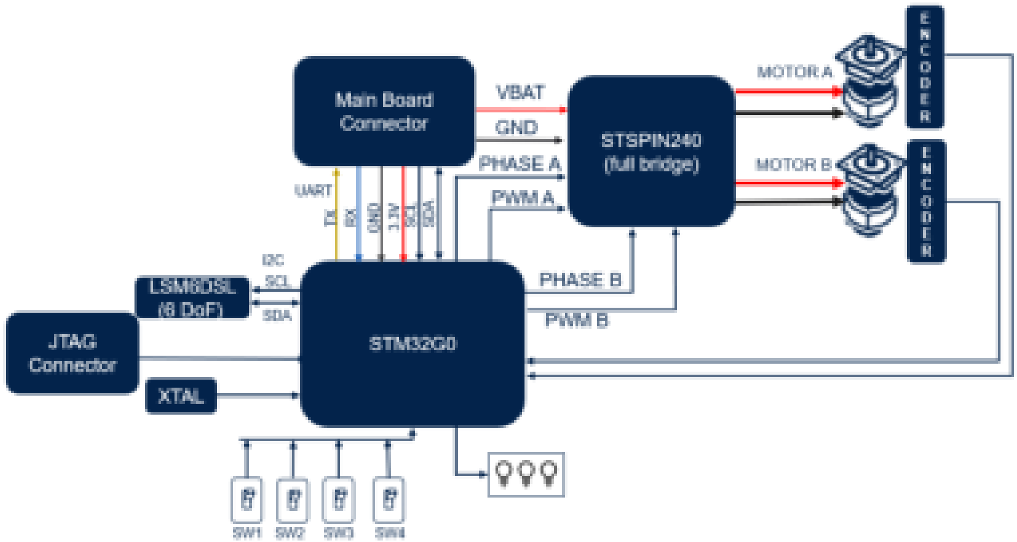
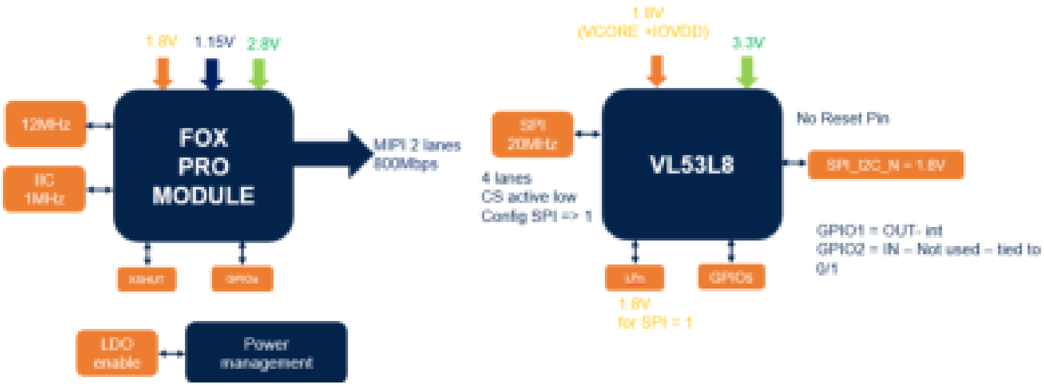


Figure 5. **Block diagram of STEVAL ROBKIT 1-2**

The block diagram of the imaging board is as follows:

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

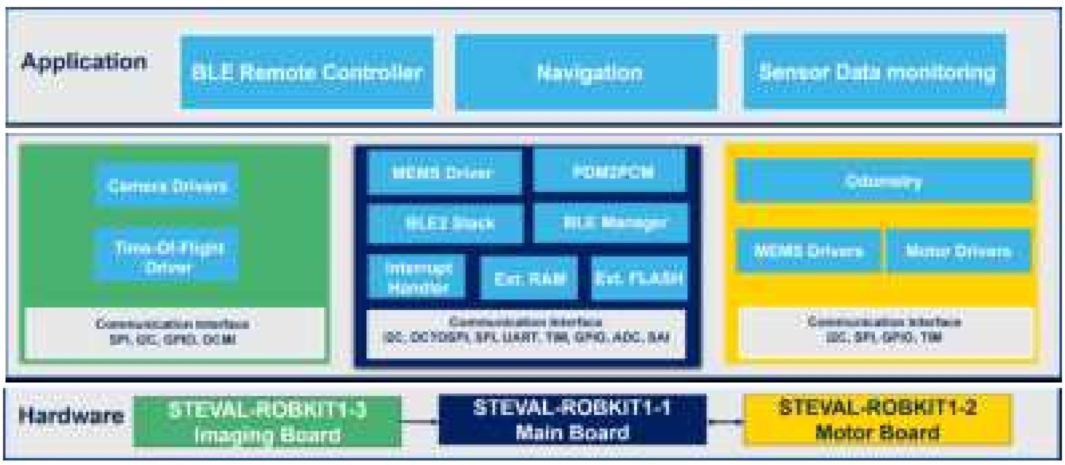


2.2 Architecture

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

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**. These are described in next two sections.

2.3.1 STSW-ROBKIT1-1: firmware for Main Board

The firmware folder consists of the projects for IAR, Keil and STM32CubeIDE with the application source and header files.

- **User:** is the firmware project core. This folder contains the following files
  - **User Application Files**:
    - **appli_blemgr.c**: Application-level Bluetooth Low Energy manager functions
    - **appli_ext_mem.c**: Functions related to external memory management
    - **appli_motion_sensors.c**: Application-level motion sensor functions
    - **appli_perception.c**: Application-level perception functions
    - **appli_robo.c**: Application-level robotic functions
    - **appli_vd56g3.c**: Application-level functions for the VD56G3 global shutter sensor
    - **appli_vl53l8cx.c**: Application-level functions for the ToF VL53L8CX sensor
    - **mtr_ctrl.c**: Motor control functions
    - **trace.c**: Trace and debug functions
    - **vl53l8cx_platform.c**: Platform-specific functions for the VL53L8CX sensor
  - **Standard peripheral files**
    - **adc.c**: Analog-to-Digital Converter (ADC) related functions
    - **BLE_Implementation.c**: Implementation of Bluetooth Low Energy functions
    - **crc.c**: Cyclic Redundancy Check (CRC) related functions
    - **custom_bus.c**: Custom bus interface functions
    - **custom_motion_sensors.c**: Custom motion sensor functions
    - **custom_motion_sensors_ex.c**: Extended custom motion sensor functions
    - **dcmi.c**: Digital Camera Interface (DCMI) related functions
    - **dma.c**: Direct Memory Access (DMA) related functions
    - **gpio.c**: General-Purpose Input/Output (GPIO) related functions
    - **hci_tl_interface.c**: Host Controller Interface (HCI) Transport Layer interface functions
    - **i2c.c**: Inter-Integrated Circuit (I2C) related functions
    - **memorymap.c**: Memory mapping functions
    - **octospi.c**: Octal SPI (OSPI) related functions
    - **pdm2pcm.c**: Pulse-Density Modulation (PDM) to Pulse-Code Modulation (PCM) conversion functions
    - **sai.c**: Serial Audio Interface (SAI) related functions
    - **spi.c**: Serial Peripheral Interface (SPI) related functions
    - **tim.c**: Timer related functions
    - **usart.c**: Universal Synchronous/Asynchronous Receiver/Transmitter (USART) related functions
  - **Standard STM32Cube application files**
    - **main.c**: Main application entry point
    - **stm32h7xx_hal_msp.c**: STM32H7 HAL MSP initialization functions
    - **stm32h7xx_it.c**: STM32H7 interrupt handler functions
- **Drivers**:
  - **BSP**: SDK drivers providing an API interface to the STM32H725IGT6, hardware resources (LEDs, buttons, sensors, I/O channel)
    - **Components**: MEMS sensors (6-axis accelerometer/gyroscope, 3-axis) device drivers, ToF device driver, external memory: PSRAM, device drivers
  - **CMSIS**: ARM® Cortex®-M7 device peripheral access layer system source file.
  - **Peripherals_Drivers**: Drivers for device peripherals (ADC, CRC, DCMI, DMA, clock, GPIO, I²C, PWR, RCC, SAI, SPI, SysTick, TIM, UART and USART)

- **Middleware**:
  - **Bluetooth® Low Energy**: Bluetooth® Low Energy stack binary library and all the definitions of stack APIs, stack, and events callbacks. Bluetooth® Low Energy stack v3.x configuration header and source files
  - **STM32_BLE_Manager**: Middleware for BLE management and hardware sensor features:
    - Battery
    - Extended Configuration
    - Debug Console
    - Inertial
    - Robot Movement
    - Scene Description
  - **HAL**: Hardware abstraction level APIs to abstract certain STM32H7xx features (Crash handler, memory utilities, FIFO management, compiler macros, over-the-air utilities for 2.4 GHz radio proprietary solution, general utilities).
  - **parson**: Library for JSON parsing and serialization
    Data Exchange/parson
  - **PDMFilter**: Library for PDM audio signal processing
    libPDMFilter_CM7_IAR_wc32.a
  - **STM32_Perception**: Middleware for perception algorithms

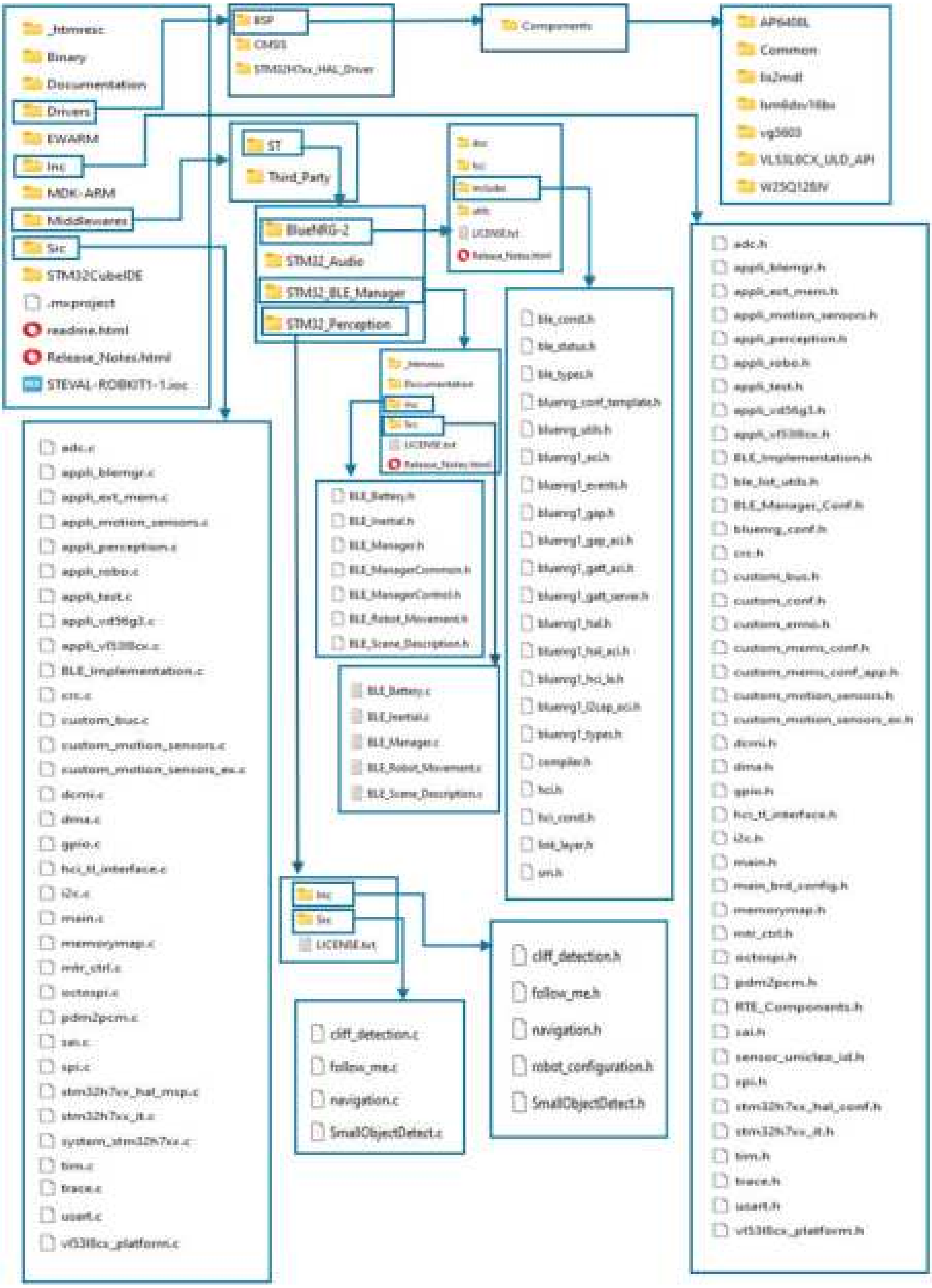
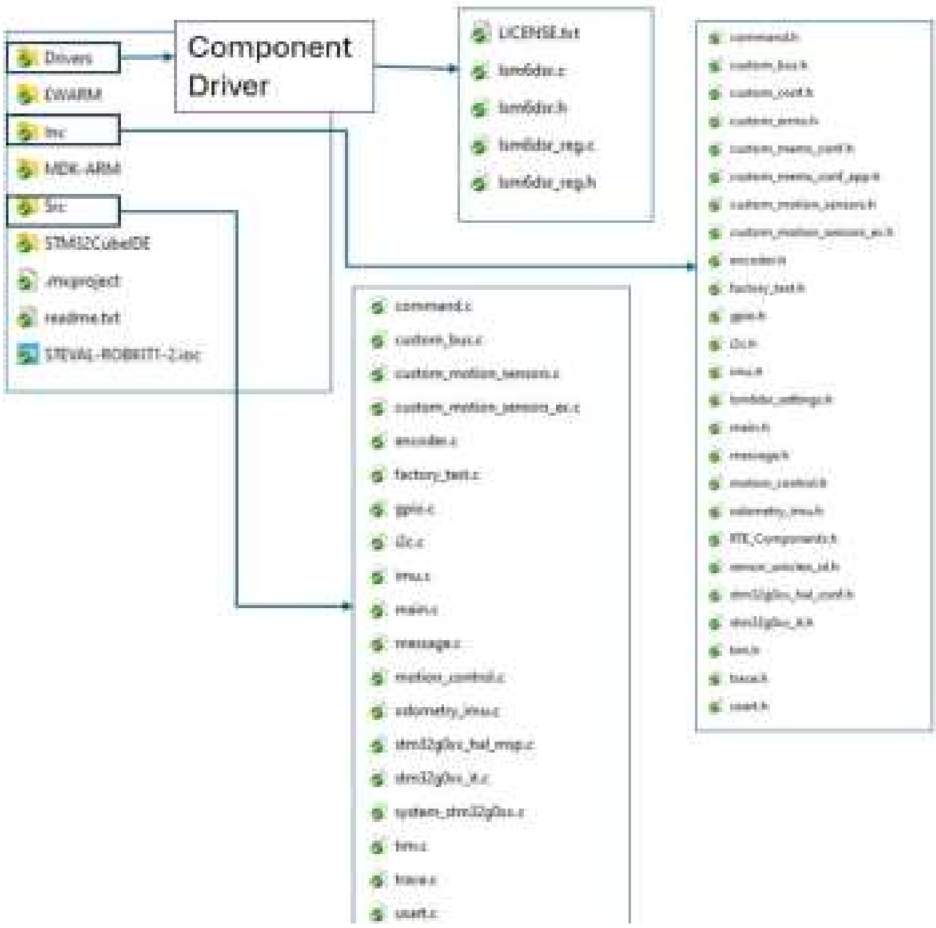


Figure 8. The folder structure of the STEVAL-ROBKIT1-1

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 9. 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 v5.41 / STM32Cube IDE v1.16.1
- RF-Flasher Utility GUI
- STM32CubePROGRAMMER GUI
- STRobotics Mobile App for Android/iOS
- Tera Term

2.6 Board setup

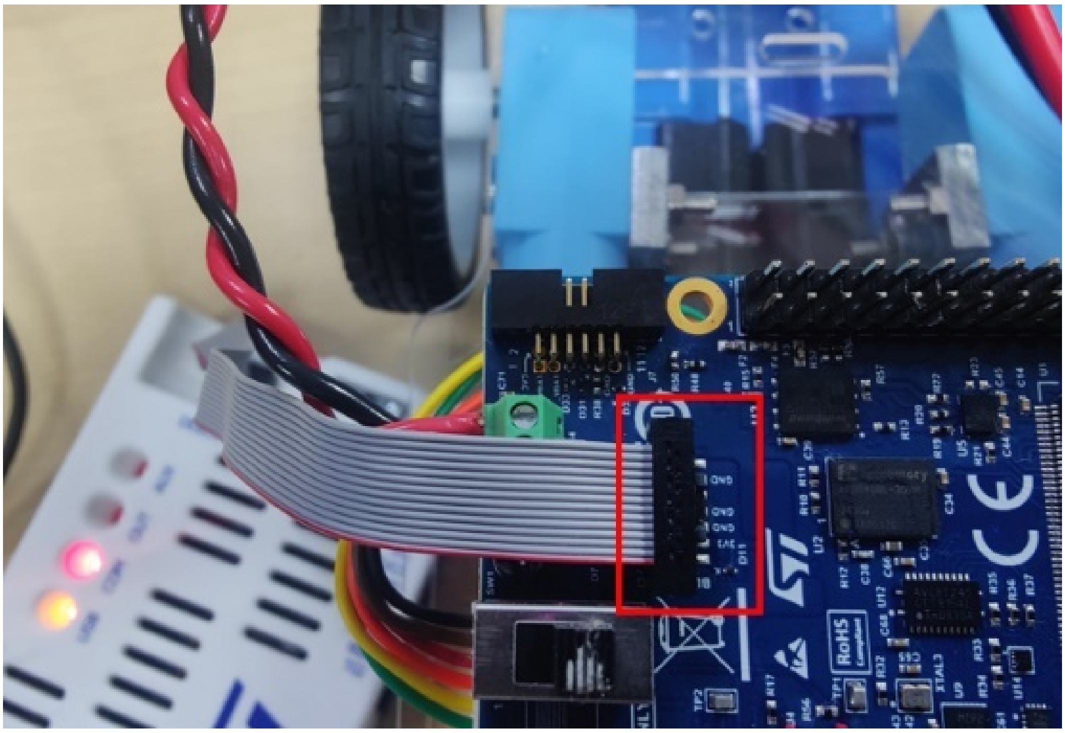
2.6.1 Setup for BLE programming

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 10 below and plug the STLINK debugger to PC via USB Type-A to Type-C USB cable.

Figure 10. STLINK Debugger connected to BLE_PROG connector (J5)

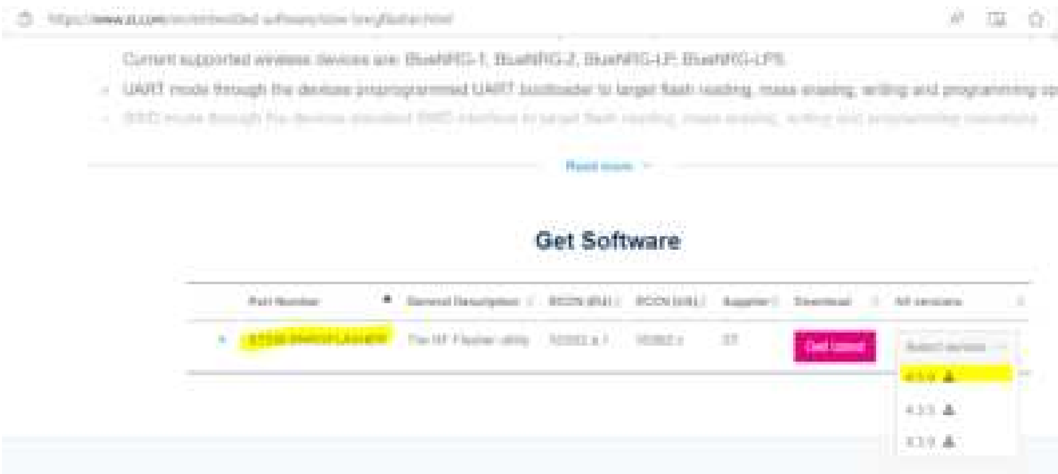


Step 2. Install the Software of RF-Flasher Utility by the given link STSW-BNRGFLASHER - The RF- Flasher utility - STMicroelectronics by following the below Figure 11.

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 11. Steps to install RF- Flasher Utility software



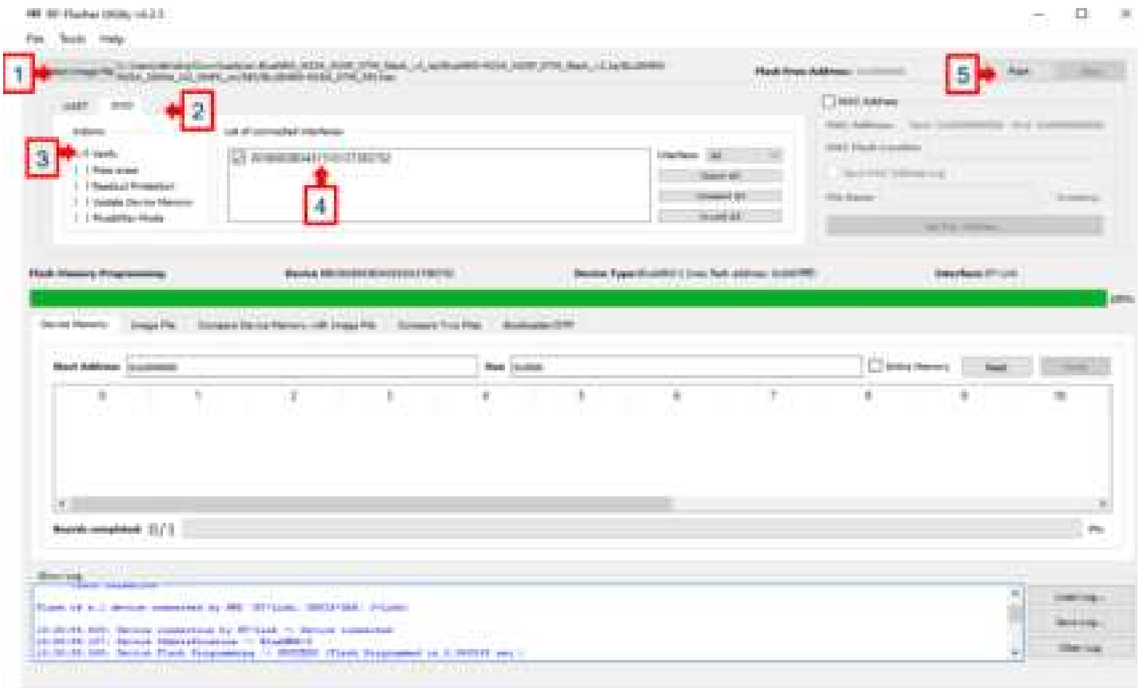
– The binary file is part of the BlueNRG-M2 - BLE v5.2 stack package downloadable here: BlueNRG-M2 - Very low power application processor module for Bluetooth® low energy v5.2 - STMicroelectronics
https://www.st.com/resource/en/utilities/bluenrg_m2sa_m2sp_dtm_stack_v2_1e.zip

– 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 12 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 12. Steps to Flash the DTM using RF-Flasher Utility tool



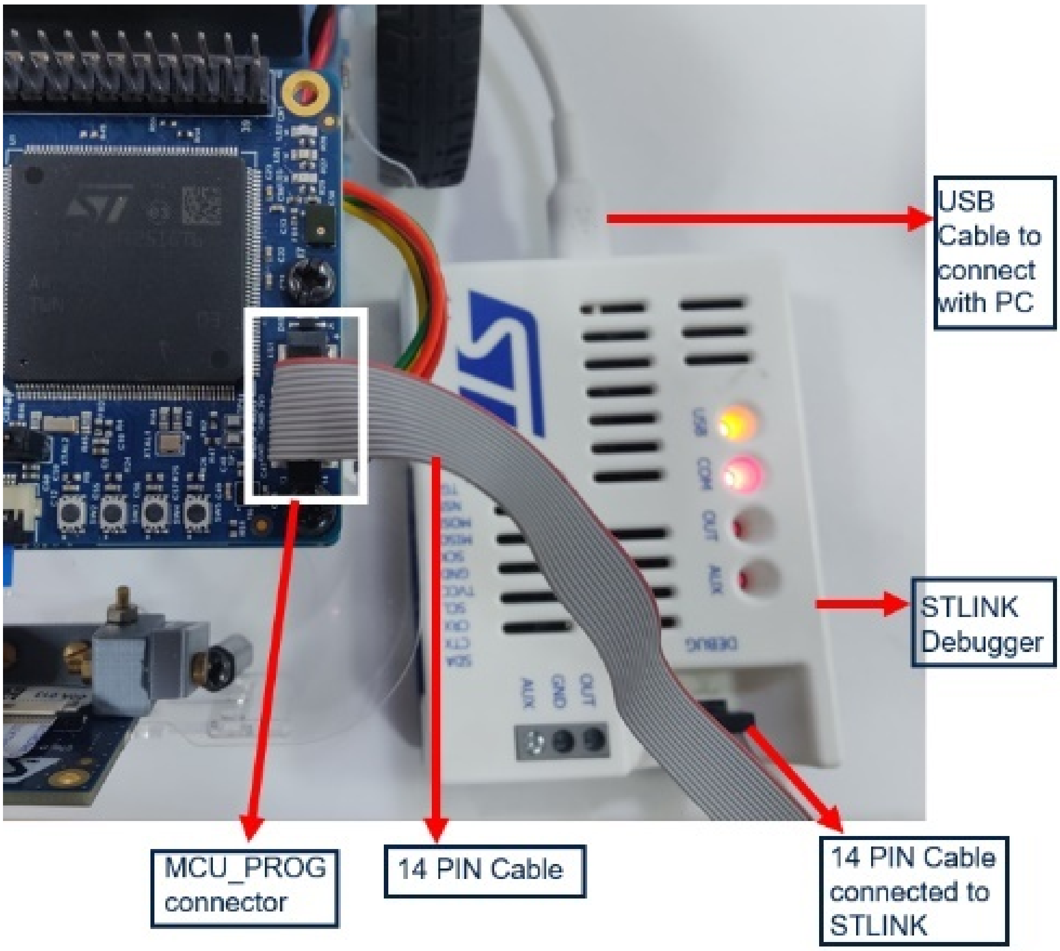
Step 5. If the DTM flash is successful then “Device Flash Programming -> SUCCESS” will be printed in the “Show log” Section

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 13

Figure 13. 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 13

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 14 for reference

Figure 14. Steps to download STM32Cube Programmer software

Step 3b. Follow Steps highlighted in Figure 15

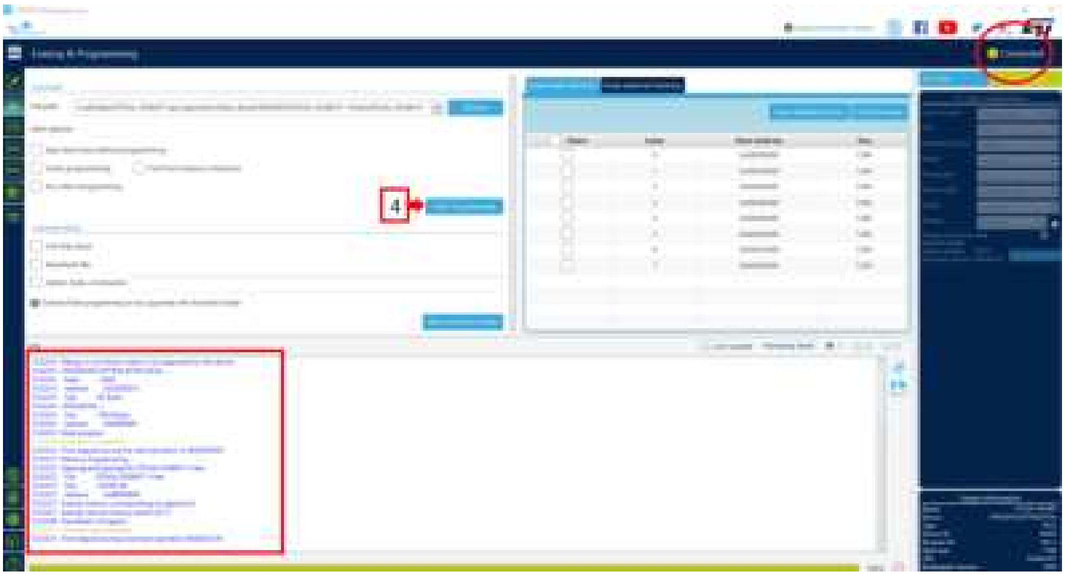
- Select “Erase & Programming tab”
- Select the “STEVAL-ROBKIT1-1.hex” file using “Browse” option

Step 4. Click on the “Connect” tab to establish connection between the software and the hardware as shown in Figure 15

Figure 15. STM32 Cube Programmer steps

Step 5. Once the connection is established the “Connected” icon can be seen as highlighted in Figure 17. 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 16 below.

Figure 16. 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 17 below

Figure 17. 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 18 for reference.

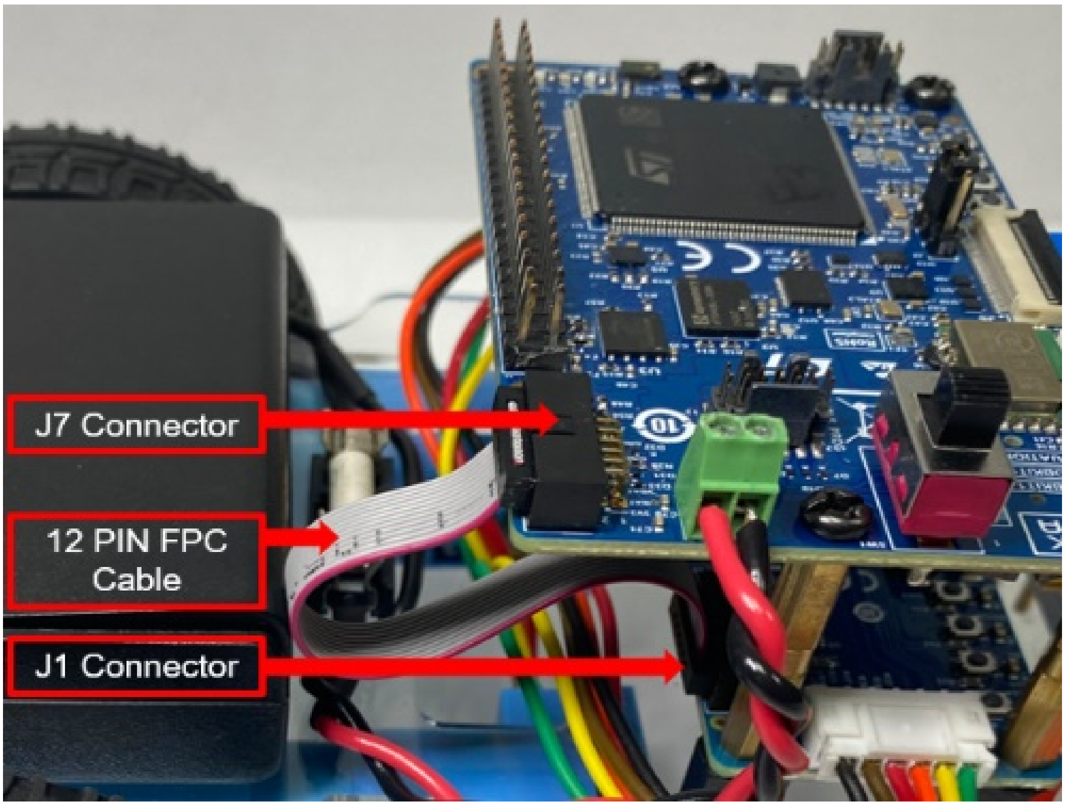
Figure 18. **Tera Term configuration**

2.6.3 Setup for STEVAL ROBKIT1-2

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

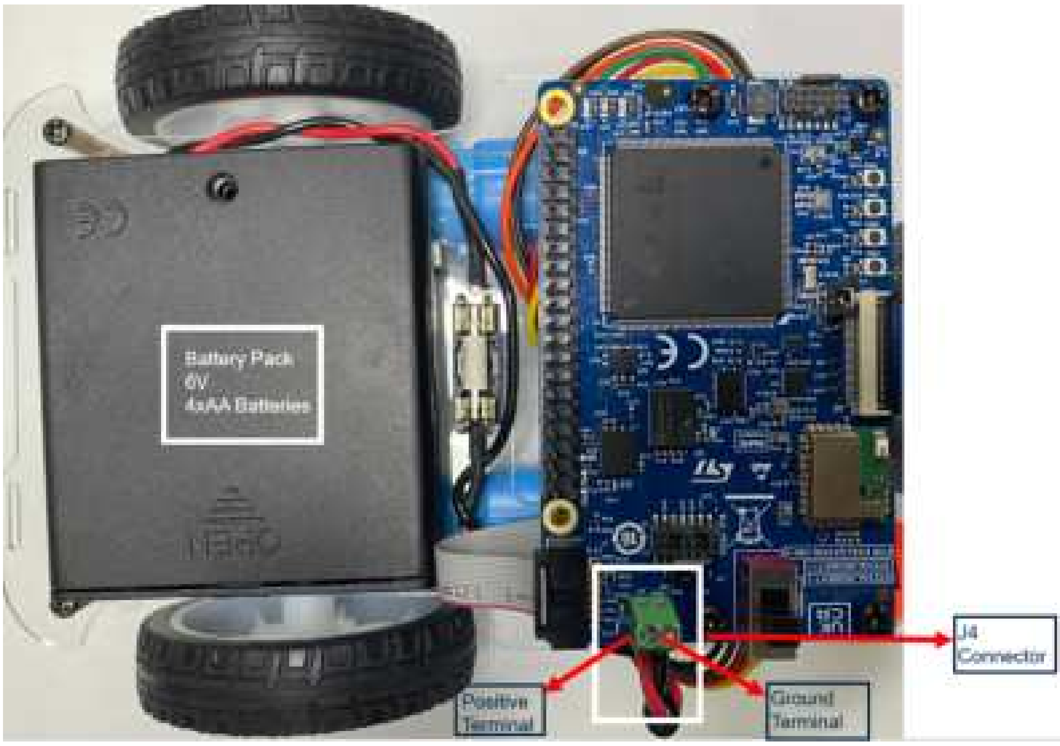
Step 1. Connect the STEVAL-ROBKIT1-2 board using the 12-pin ribbon cable. One end of the cable to be connected to the J7 of STEVAL-ROBKIT1-1 and the other end to the J1 connector of STEVAL-ROBKIT1-2 as highlighted in the Figure 19:

Figure 19. **STEVAL-ROBKIT1-1 connected with STEVAL-ROBKIT1-2**



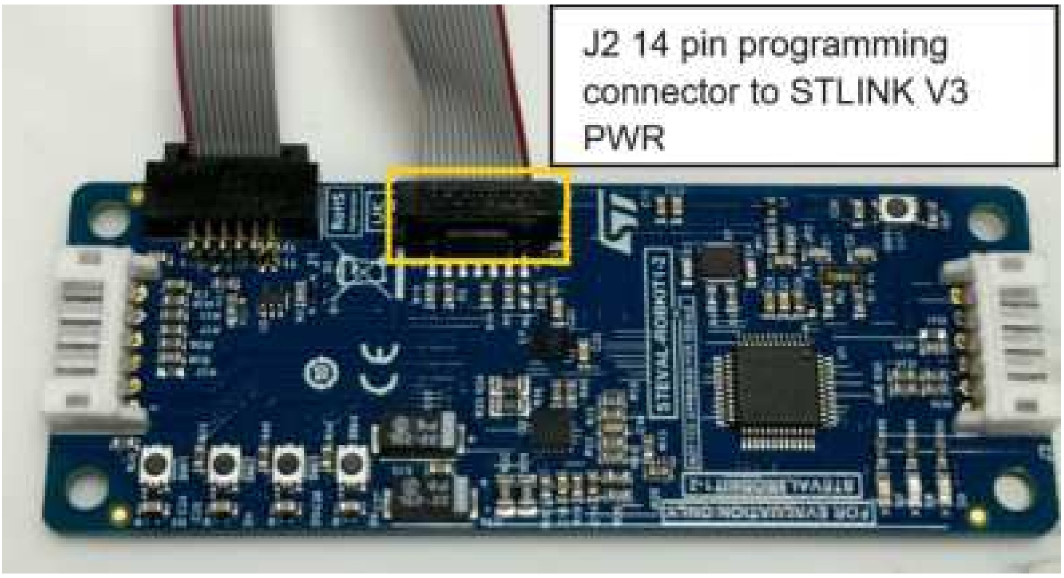
Step 2. Connect the STEVAL-ROBKIT1-1 with the power source using J4 connector on the board as shown in Figure 20. STEVAL-ROBKIT1-1 connected with the power supply

Figure 20. 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 21

Figure 21. 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 15

- 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 16

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 17 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

3.1 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 22. Expected output on STRobotics App for motion sensor data

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

3.2 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.

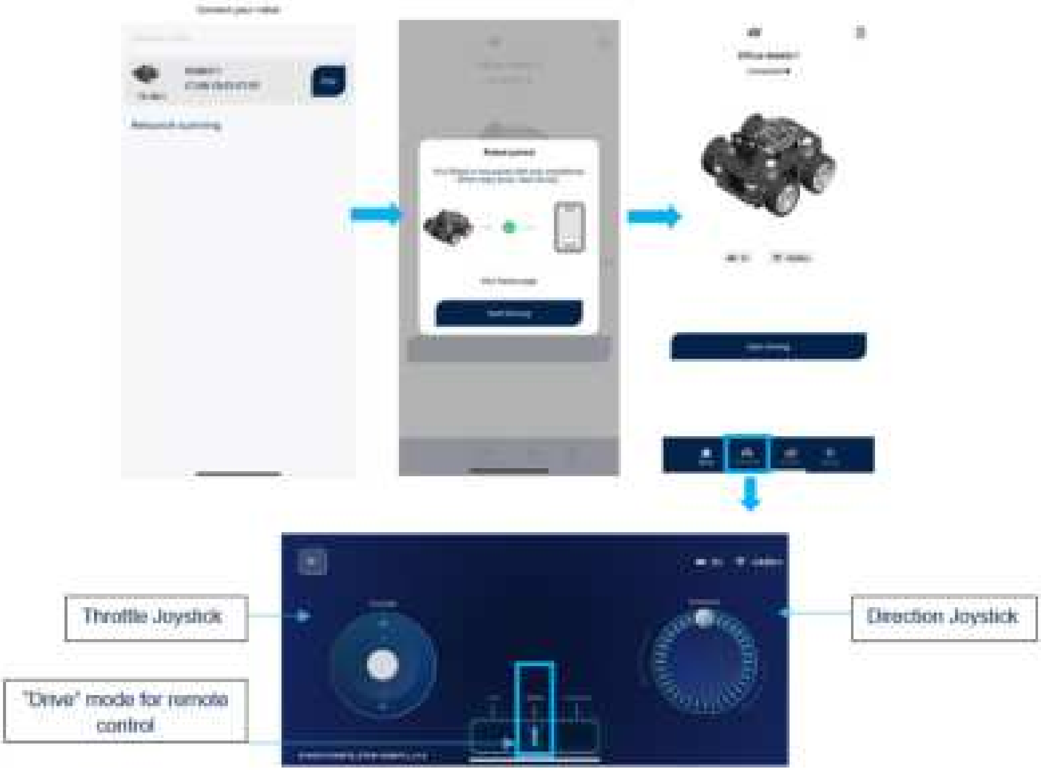
Running the Application:

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 23.

Figure 23. Expected output on STRobotics App for Robotic Movement: Remote Control



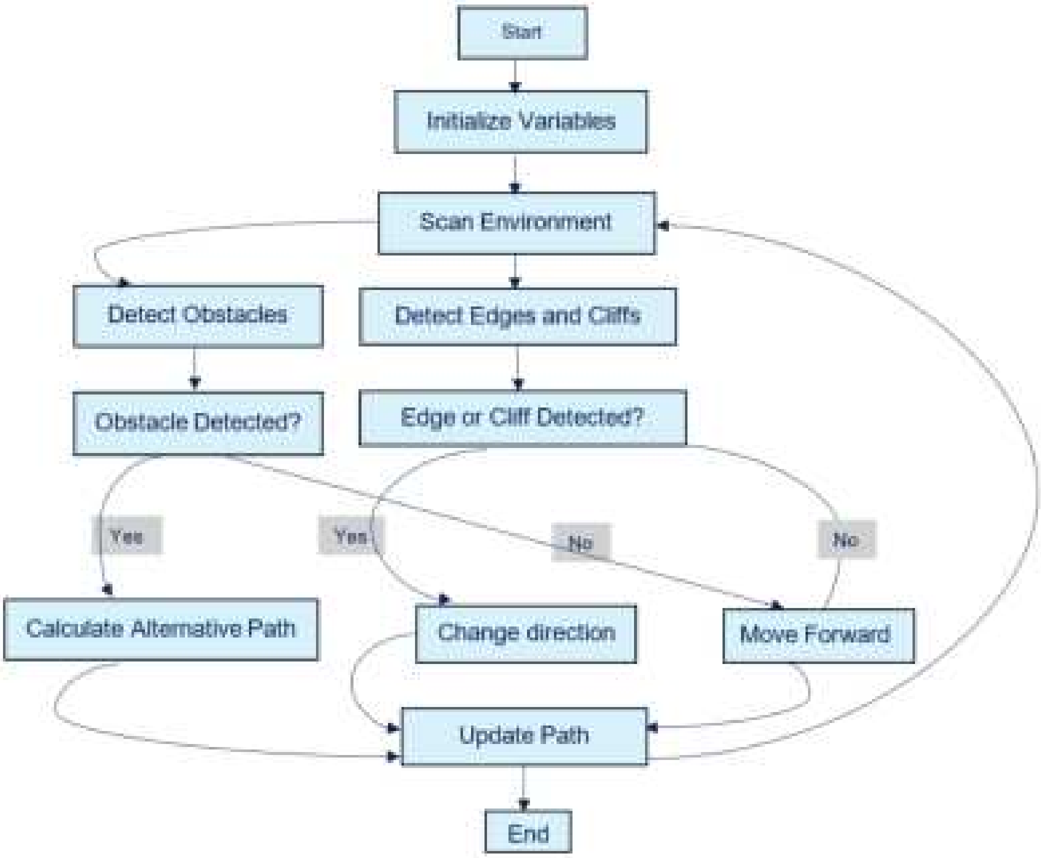
3.3 Free navigation feature

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 24.

Figure 24. **Autopilot mode for free navigation feature in STRobotics app**

Figure 25. **The high-level overview of the free navigation feature**



3.4 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.

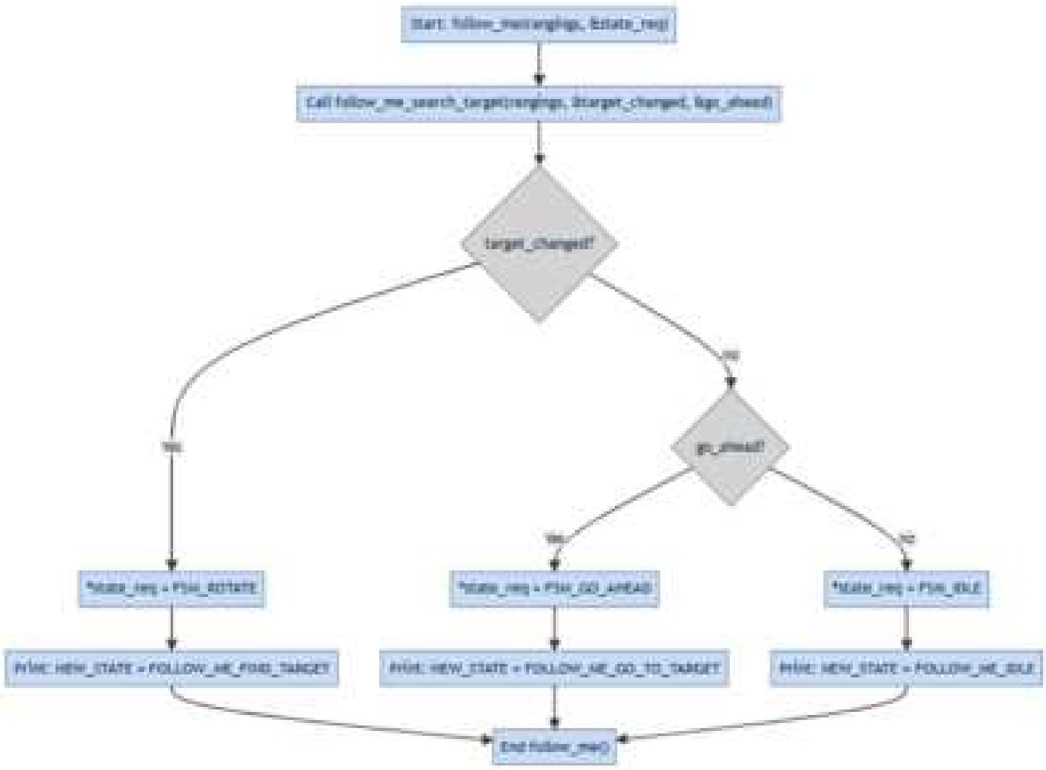
The detailed working of the feature is as follows:

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



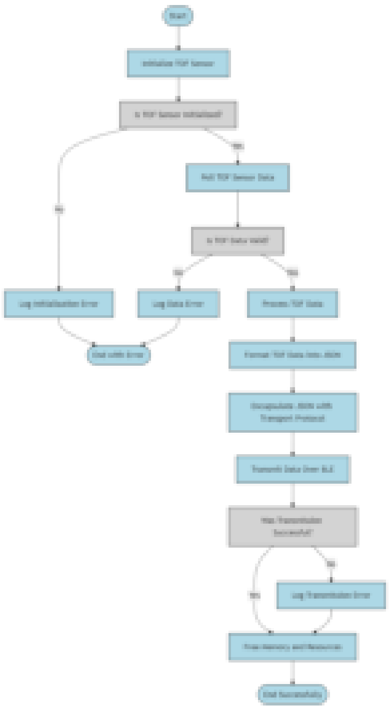
3.5 Scene description for ToF data visualization feature

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 27. Utilize the color indicators and distance indicated in the matrix (in mm) for monitoring the ToF data in real time.

Figure 27. Scene description in STRobotics app

Figure 28. **The high-level overview of the scene description feature**



3.6 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 29. **Debug console feature in STRobotics app**

3.7 Extended configuration feature

The “READ Version” Command under the Extended Configuration Command feature of the BLE Manager is used to display the controller and firmware name along with their versions on the STRobotics mobile app as shown in Figure 30. When executed, the command retrieves version details, formats them into a JSON object, and serializes it into a string. This string is then sent to the mobile app, allowing users to view the version information directly on the app. This feature is essential for verifying firmware versions and ensuring compatibility during development and deployment.

Figure 30. Firmware info and version display on STRobotics app through extended configuration feature

3.8 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 31. 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

Revision history

Table 2. **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 24. Autopilot mode for free navigation feature in STRobotics app and Section 3.8: State machine for STEVAL-ROBKIT1-2: motor board. Added Section 3.5: Scene description for ToF data visualization feature, Section 3.6: Debug console feature, Section 3.4: Follow Me navigation feature and Section 3.7: Extended configuration feature.

Contents

List of tables

List of figures