New to ModusToolbox™ software?

If you're new to ModusToolbox™ software, welcome! This document provides a quick explanation about what it is and how to use it.
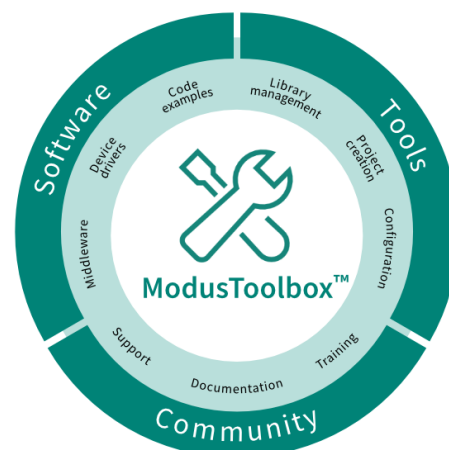
What is ModusToolbox™ software?

A collection of GUI and non-GUI tools, libraries, and programs used to develop embedded applications.

Desktop tools are installed as part of the ModusToolbox™ Tools Package available from the Infineon Developer Center.

Additional patches and packs are also available from the Infineon Developer Center to supplement the base ModusToolbox™ tools installation.

Run-time software is delivered via GitHub repositories during application creation and development.

Key tools used in the ModusToolbox™ development:

Dashboard – provides a central getting starting point for ModusToolbox™ software as well as quick access to documentation, videos, training material, and launch actions to create a new application.

Project Creator – provides quick access to reference code examples on GitHub and facilitates creating IDE-specific applications through an easy-to-use interface.

BSP Assistant – provides a GUI-based tool for developing a board support package (BSP) for Infineon kits and custom hardware. BSPs can be derived from an existing BSP or developed from scratch based on a silicon device part number.

BSP Level Configuration – BSP and device configurators are used to generate initialization structures for clocks, pins, and peripherals. Examples of BSP Level Configurators include Device Configurator, Smart I/O Configurator, and QSPI Configurator.

Library Manager – provides a user interface for managing middleware libraries and BSPs within an application. Middleware with available updates are indicated within the tool and users have full control over the version to be used in the project. The Library Manger is also able to resolve any middleware dependencies to ensure that the included middleware is ready to be used.

IDE – ModusToolbox™ software supports several industry-standard IDEs:

- Eclipse-based IDE (bundled with the ModusToolbox™ Tools Package)
- Microsoft Visual Studio Code
- IAR Embedded Workbench
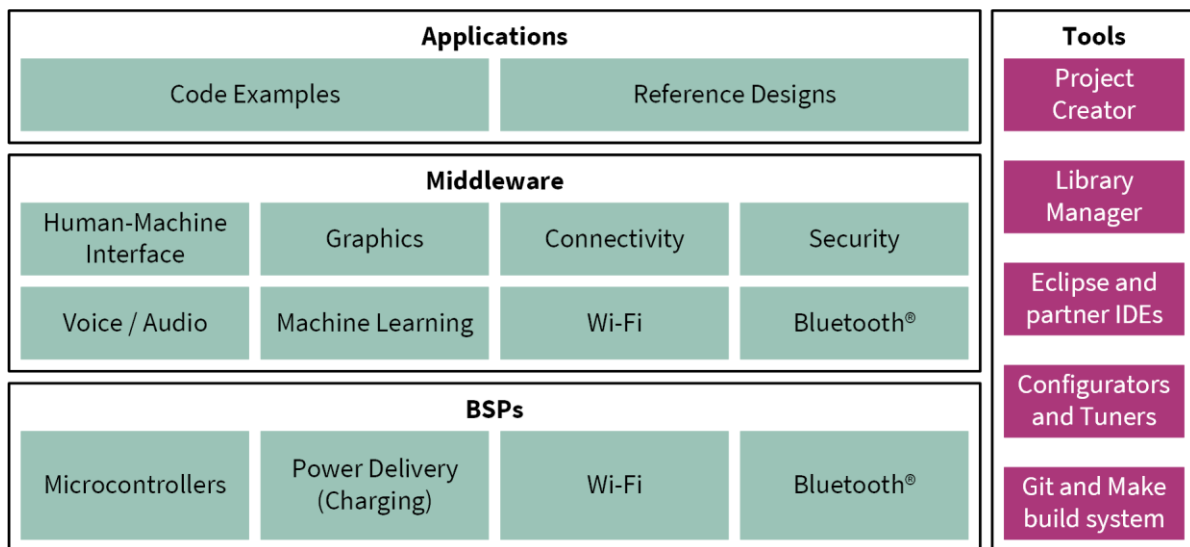- Arm Microcontroller Developers Kit (µVision)

What is ModusToolbox™ software?

Library Level Configuration – library configurators are used to change the behavior of the supported middleware and generate configuration data for the library APIs. Examples of Library Level Configurators include Machine Learning, CAPSENSE™, and Bluetooth®.

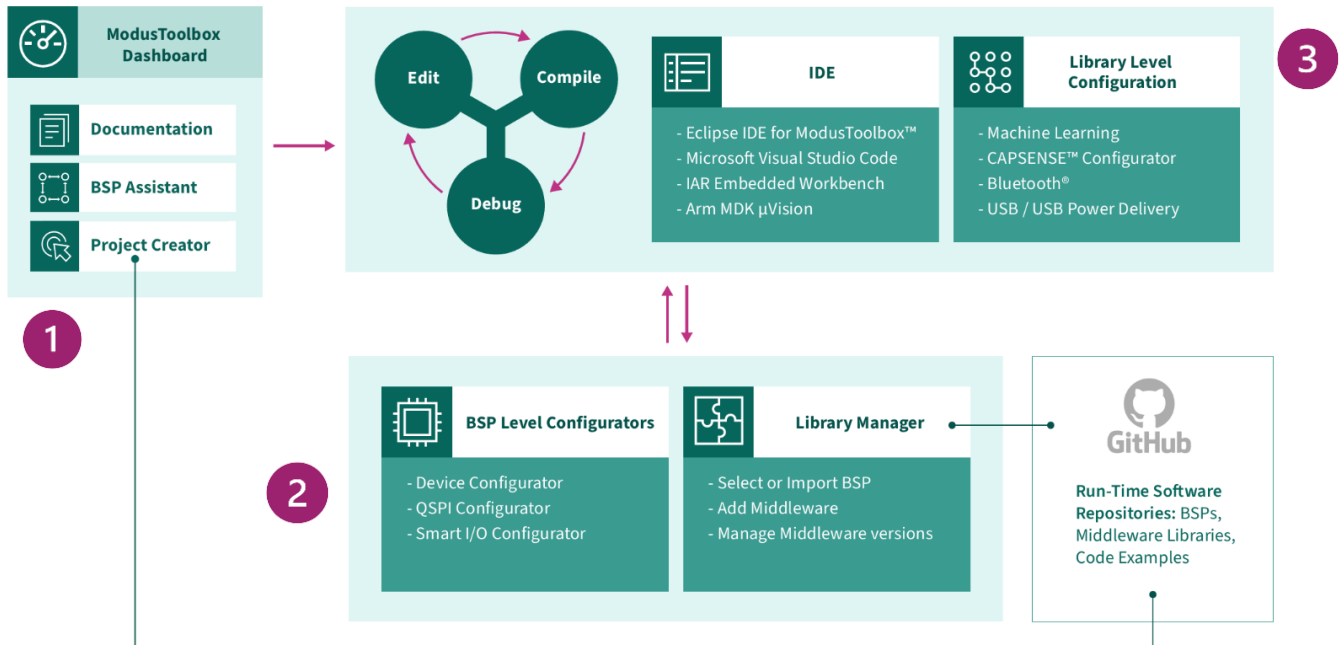## Software and tools in a ModusToolbox™ application:

ModusToolbox™ Software provides a comprehensive development environment for developing embedded applications. This suite of tools and run-time software enables efficient development from the foundational board support package (BSP) and device support through the user application with available middleware libraries and reference code example projects.

Applications				**Tools**
Code Examples		Reference Designs		Project Creator
Middleware				Library Manager
Human-Machine Interface	Graphics	Connectivity	Security	Eclipse and partner IDEs
Voice / Audio	Machine Learning	Wi-Fi	Bluetooth®	Configurators and Tuners
BSPs				Git and Make build system
Microcontrollers	Power Delivery (Charging)	Wi-Fi	Bluetooth®	

How do I develop embedded applications with ModusToolbox™ software?

ModusToolbox™ software is a flexible and comprehensive development environment aimed at enabling iterative development and facilitating an efficient Edit-Compile-Debug cycle.

As depicted in the illustration, ModusToolbox™ software provides a number of desktop applications and tools centered around the typical "Edit-Compile-Debug" cycle.

1   Tools like **BSP Assistant** and **Project Creator** are used during the initial creation of an application, generating the necessary IDE-specific project files and workspace structure. These tools can be used standalone by launching them directly from the computer's menu system. Additionally, they can be access from the **Dashboard** or within the Quick Panel of the **Eclipse IDE for ModusToolbox™**.

With an application created, it can now be used within the respective IDE.

There are IDE specific instructions provided within the documentation and output dialog within the **Project Creator**. Following these steps will guide you through getting the application into your **IDE**. When creating an application through the Quick Panel of the **Eclipse IDE**, these steps will be handled automatically. For **IAR Embedded Workbench** and **Arm MDK μVision**, these are specific steps to that align with the how these IDEs handle externally generated project files.

2   As part of the initial development, the **BSP Level Configurators** and **Library Manager** can be used to modify the device settings initially provided by the BSP or to add additional software libraries as needed for you application. These tools can be used through the development process anytime changes or additional middleware is required.

How do I get started?

The majority of the application development effort occurs within the "Edit-Compile-Debug" cycle.

3 This is typically done within the selected **IDE**, but ModusToolbox™ also enables compiling and programming directly from command-line. An **IDE** provides an integrated development environment with built-in code editor, programming capabilities, and debug control. ModusToolbox™ includes support for SEGGER J-Link and Infineon KitProg3/MiniProg4 debuggers. As middleware is integrated into your application the **Library Level Configuration** tools can be used to configure the supported middleware configurations and help to visualize how the middleware is functioning at run-time with available tuners.

How do I get started?

- Create an application using the Project Creator tool.
  - Select a board support package (BSP).
  - Select a template application from a code example.
  - Optionally specify a target IDE, such as Eclipse, VS Code, Embedded Workbench, or MDK (µVision).

What do I do next?

- Update the BSP to use a different MCU and/or companion device.
- Modify the BSP initialization code to align the application to your design.
- Add middleware libraries to the application workspace and reference the included READMEs for any additional integration steps.
- Start Developing:
  - Edit the application code using the IDE editor and as needed modify middleware configurations using Library configurators.
  - Compile the application using the IDE build/compile functionality or leverage the command-line 'make' functions. Makefiles within the application can be used to control the build process.
  - Debug the application using one of the supported debug probes, stepping through the application with the IDE's debug controls and using the library level configuration tuners to visualize the running middleware. Simultaneous debugging is supported for multicore device projects.

Where can I learn more?

Infineon provides training, videos, documentation, and community pages:

- [Training material](#)
- [Videos](#)
- [Documentation](#)
- [Community pages](#)

Revision history

Revision	Date	Description
**	2023-05-12	New document

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.