MCU Motor Studio
Install manual

Description

This document describes the “MCU Motor Studio” installation procedure on Microsoft windows® PC and connecting Clicker 4 for TMPM4K and Inverter shield.

Table of Contents

List of Figures

List of Tables

Conventions used in this document

Numerical Values

Hexadecimal number: 0xABC or 0h12F
Decimal number: 123 or 0d123 (explicitly indicating the decimal numbers)
Binary number: 0b111

Signals

Active low signals are indicated by a _N at the end of the signal name. Example: RESET_N.
Assertion of a signal shall mean its activation (transition to the active state). Deassertion of a signal shall mean its deactivation (transition to the inactive state).

Bus signals are indicated by [x:y] at the end of signal name. Example: DATA[3:0] indicates a four bit bus with the individual bus signals DATA[3], DATA[2], DATA[1] and DATA[0].

Registers

Register names are indicated by square brackets […]. Example: [ABCD].

Two or more of the same kind of registers, fields, and bit names are collectively referred to by using a numerical suffix n. Example: [XYZ1], [XYZ2] and [XYZ3] are collectively referred to as [XYZn].

The bit width of a register is expressed as [x:y] where x is the number of the most significant bit and y is the number of the least significant bit. Example: [XYZ][3:0] indicates a four bit-wide register named XYZ.

The configuration value of a register is expressed by either a hexadecimal number or a binary number. Example: [ABCD].EFG = 0x01 (hexadecimal), [XYZn].XY = 0b1 (binary).

The following definitions apply for Bytes and Words:

Byte	8 bits
Half Word	16 bits
Word	32 bits
Double Word	64 bits

Unless specified otherwise, registers support only word access.

Register which are indicated to be reserved must not be rewritten. The read value from reserved registers must not be used.

Properties of each bit in a register are expressed as follows:

R	Read only
W	Write only
W1C	Write 1 Clear; the corresponding bit is cleared (=0) when "1" is written to this bit.
W1S	Write 1 Set; the corresponding bit is set (=1) when "1" is written to this bit.
R/W	Read and Write are possible.
R/W0C	Read/Write 0 Clear
R/W1C	Read/Write 1 Clear
R/W1S	Read/Write 1 Set
RS/WC	Read Set/Write Clear; set after read operation, cleared after write operation.

Reading from register bits having a default value of "—" will result in an unknown value.
In case of write accesses to registers containing both read/write (R/W) and read-only (R) bits, the read-only bits shall be written with their default value. If this default is "—", follow the instructions of each register.

Reserved bits of Write-only (W) register should be written with their default value. If this default is "—", follow the instructions of each register.

1. Introduction

This manual describes the setup of hardware platform (“Clicker 4 for TMPM4K” and “Clicker 4 Inverter shield”) and the installation of “MCU Motor Studio” PC tool applications and individual plug-ins that are associated with each device.

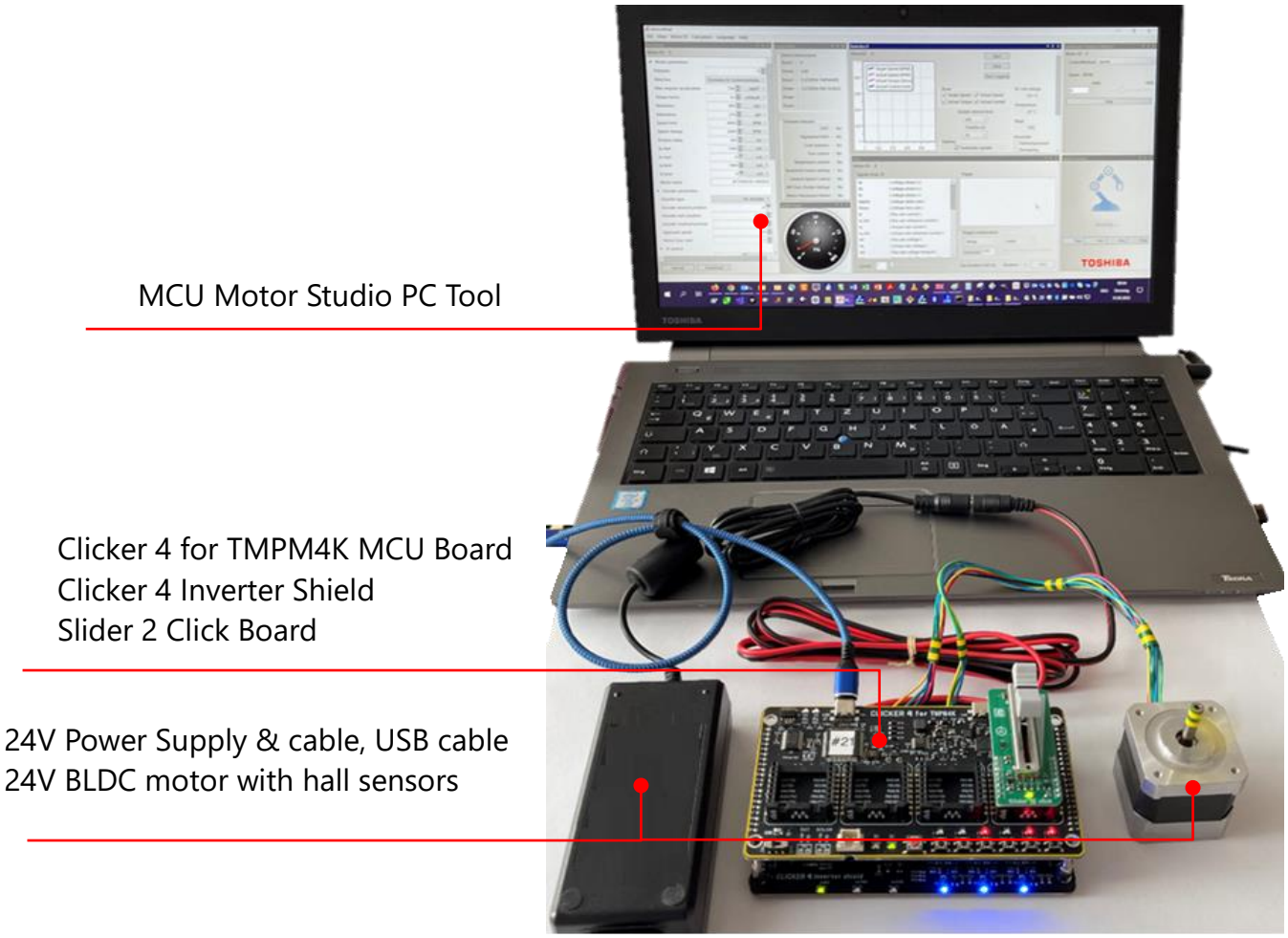

Figure 1.1 Hardware setup

2. Set-up

2.1. Requirements

2.1.1. Supported platforms

The application is built to run exclusively on Microsoft Windows based platforms (Microsoft Windows 10 64-bit OS).

2.1.2. Disk space

5MB of free disk space is needed for the typical installation package.
12MB of free disk space is needed for the application. Please allow some more disk space for logging configuration files that can be generated during the normal operation.

2.1.3. Memory usage

The minimal amount of RAM required to ensure normal operation in all possible cases is **128MB**.

2.1.4. Connectivity

At least one free Serial or USB port for the protocol communication (via the assigned COM port) is required. A second one might be needed given the optional HS-DSO communication will be used.

2.1.5. Additional Software Components

“MCU Motor Studio” is a .NET framework-dependent published application. It does not include the .NET runtime and libraries, just the executable and all third-party components it is utilizing.
If needed .NET resources are not pre-installed, a dedicated set-up program will be automatically started. It will attempt to download and install the missing prerequisite components. In such cases, the user shall ensure there is active Internet connection and the access to the component vendor’s websites is permitted.
Alternatively, an offline installation of the required .NET 3.5 SP1, .NET 4.0 Client Profile and Microsoft Windows® Installer 3.1 packages can be performed prior to the “MCU Motor Studio” installation.

2.2. Package Content

The installer package contains “MCU Motor Studio.msi” - the Microsoft Windows® installer

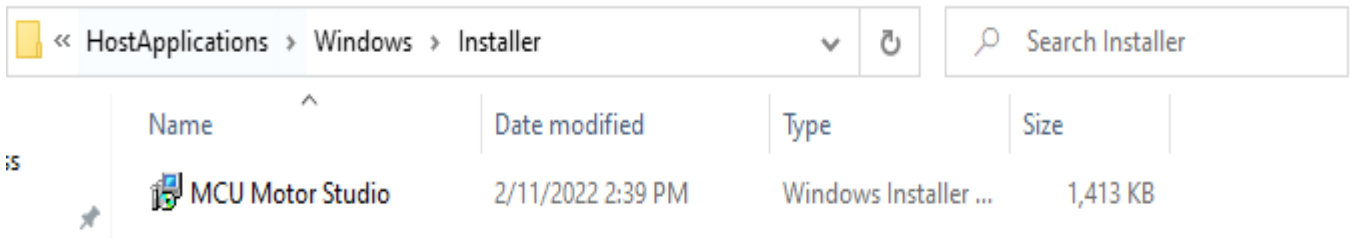

Figure 2.1 Installation package content

3. Driver Installation

3.1. Driver Installation steps

Connect the “Clicker 4 for TMPM4K” board “CN1” connector to Microsoft Windows® PC, and install necessary virtual com port driver based on processor architecture from https://ftdichip.com/drivers/vcp-drivers/.

Confirm that the driver is installed successfully using Microsoft Windows® “Device manager”. Below figure shows snapshot of successful installation of virtual com port driver.

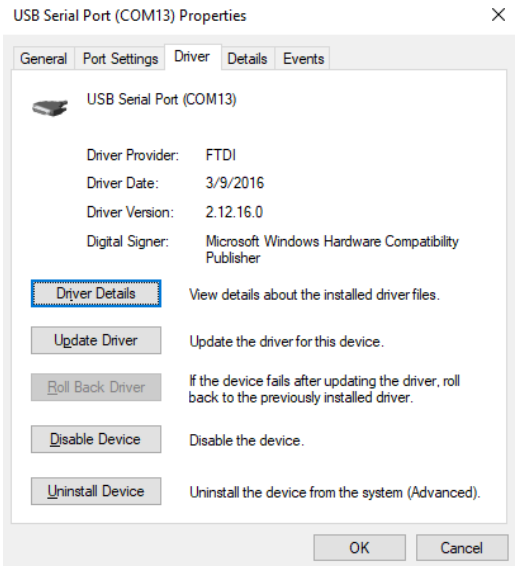

Figure 3.1 Driver installation

4. MCU Motor Studio Installation:

4.1. Installation steps

Execute “MCU Motor Studio.msi” - the Microsoft Windows® installer – please ensure user has admin privileges prior to the installation.

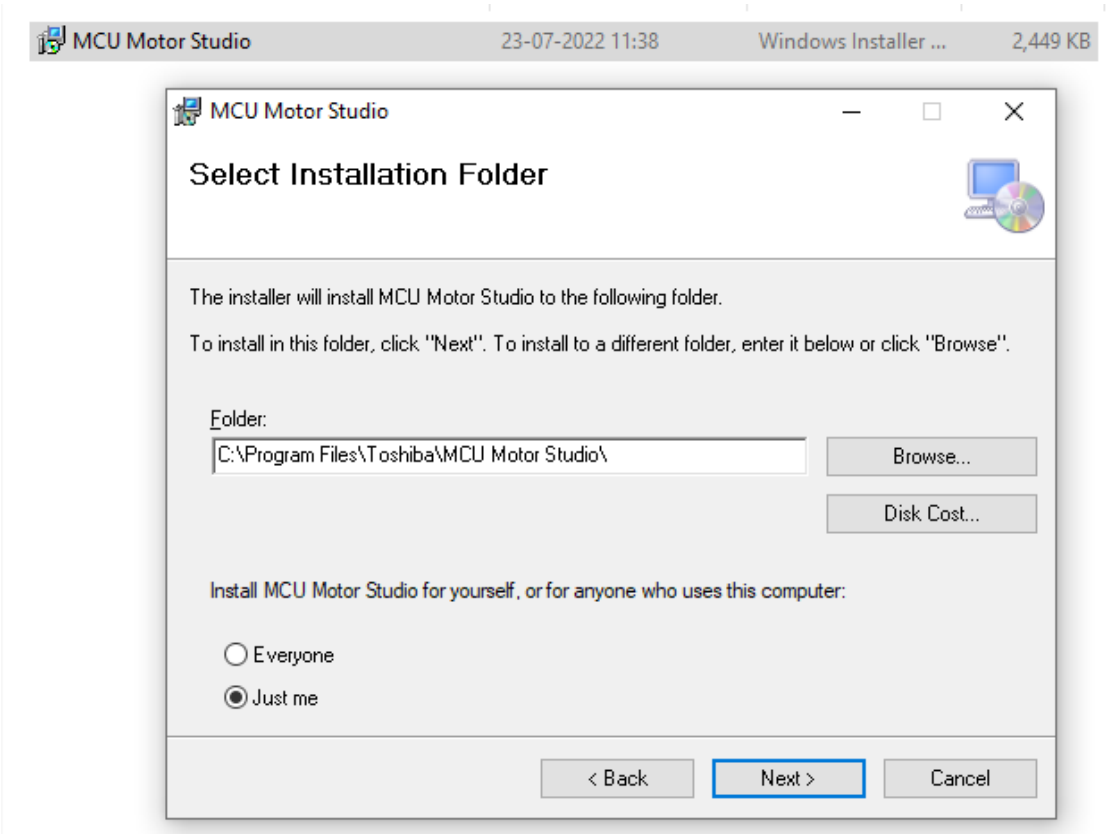

Figure 4.1 Installation window

Please ensure that the installer file is digitally signed using a valid Toshiba certificate by clicking on ‘Show more details’.

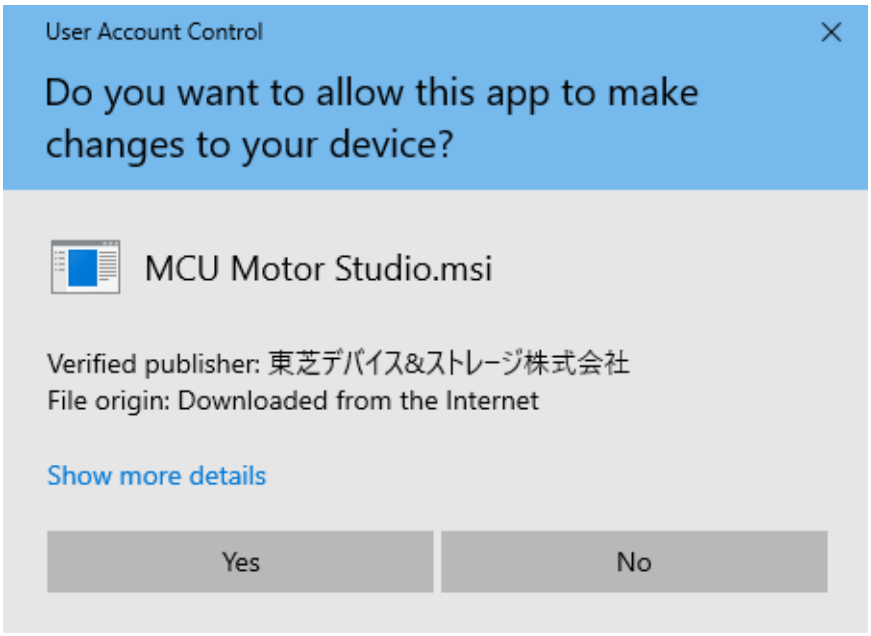

Figure 4.2 Digitally signed installer message window

4.2. Install MSI from command line with Administrator

The MCU Motor Studio installation using command prompt with Administrator

- Right click on Windows Start, choose Command Prompt (Admin)
- In the command prompt, input “msiexec /i “path\MCU Motor Studio_64bit.msi“”
- Press Enter to start the install process.

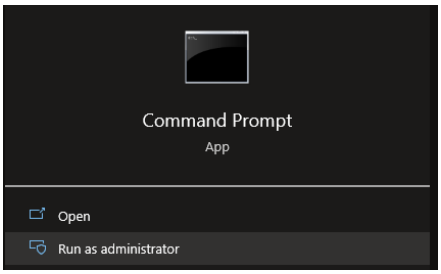

Figure 4.3 Install from Command Prompt

4.3. Initial configurations

After successful installation “MCU Motor Studio” will be automatically started and an initial configuration will be prompted. User need to have a “Clicker 4 for TMPM4K” powered and connected to one of the COM ports. Please select the correct COM port and click ‘Ok’ button:

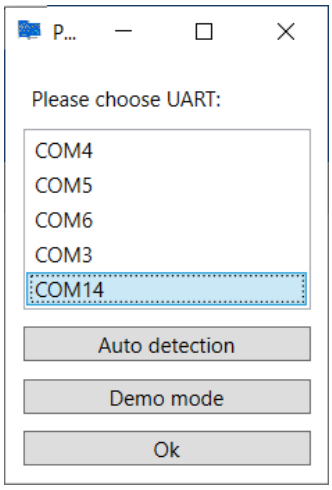

Figure 4.4 COM port detect

The Tool features Auto Detection mode. If selected all available COM ports will be scanned in ascending order, attempting to establish communication with the attached board. The process may be rather slow, depending on the number of ports in the system. It is strongly recommended to have the “Clicker 4 for TMPM4K” board powered and connected to the PC with any required USB to Serial bridge driver properly installed prior to any connection attempts.

The last used COM port will be memorized and used the next time the Tool is started. The COM port selection window will only be prompted again, if the attempted connection cannot be established.

Next you shall select the motor channel you would like to start with. The prompt will be presented even if only one channel is configured and usable on the platform that is currently connected, given the MCU is capable of driving more channels.

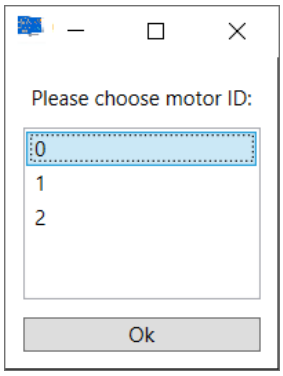

Figure 4.5 Initial motor channel selection

The last used channel will be memorized and used at the next start. The channel selection will be prompted again, only if the channel is not available, typically as different platform with different capabilities is connected.

You are ready to start using the powerful PC tool and dive into the intriguing world of Toshiba's Motor Control Solution. Welcome to **"MCU Motor Studio"!**

4.4. De-installation

There are several cases where the application has to be completely removed:

- New "MCU Motor Studio" version is available and shall be installed – in that case the user needs to remove the previously installed one.
- The current version is to be re-installed – very unlikely, but in some cases this might be one of the first trials to resolve issues with the application.
- The application is not needed anymore – although it is very tiny, sometimes a clean-up is needed.
- The removal is very easy and follows the Microsoft Windows® standard approach. From Control Panel -> Programs select "MCU Motor Studio" in the list (as illustrated below), hit Uninstall and confirm the removal with OK:

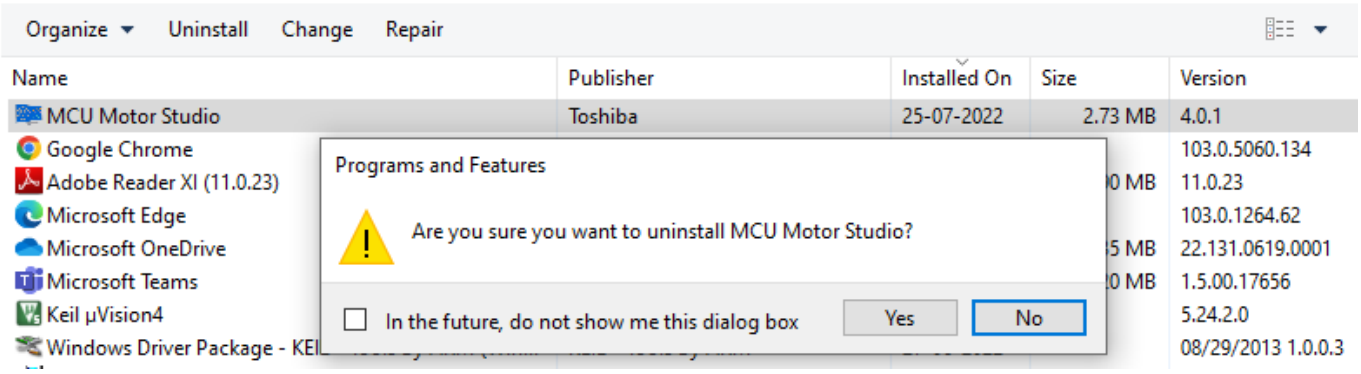

Figure 4.6 "MCU Motor Studio" de-installations

After pressing "OK" to uninstalling "MCU Motor Studio", related files and folder shall be completely removed. Typically, there is no need to perform any manual deletion of any empty folders and/or files, unless you have specified different location for storing auto-generated files, such as logs, parameter configuration files, sequences, etc. It is assumed in this case that the user may want to keep these and therefore these are not automatically removed.

5. Clicker 4 for TMPM4K board Set-up:

5.1. Board setting of “Clicker 4 for TMPM4K” board

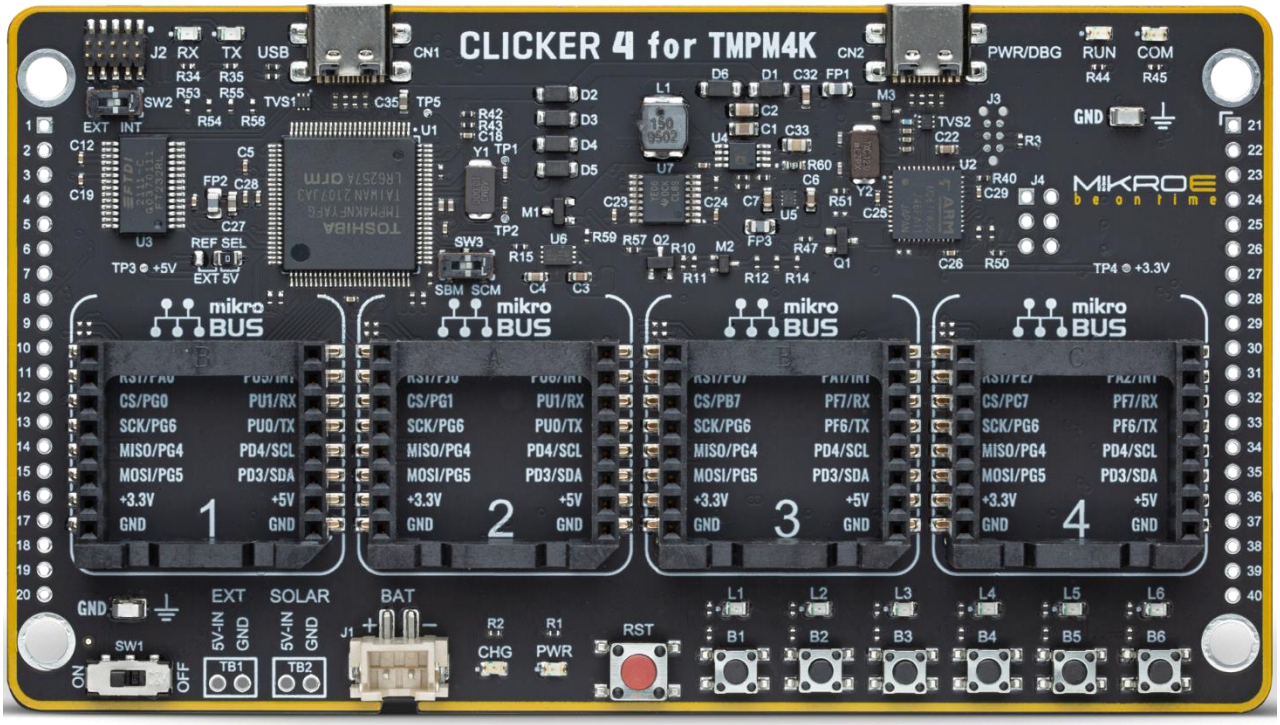

Figure 5.1 Clicker 4 for TMPM4K

Sl no	Functionality	Hardware setting
1	MCU Power ON/OFF	**SW1**: ON / OFF
2	Debugger settings	For on-board CMSIS DAP (Use CN2) For External J-link JTAG SWD: **J2**
3	UART USB support (Connect to PC tool)	Use CN1

Table 5.1 Clicker 4 for TMPM4K Hardware setting

5.2. Board Configuration (Clicker 4 Inverter Shield)

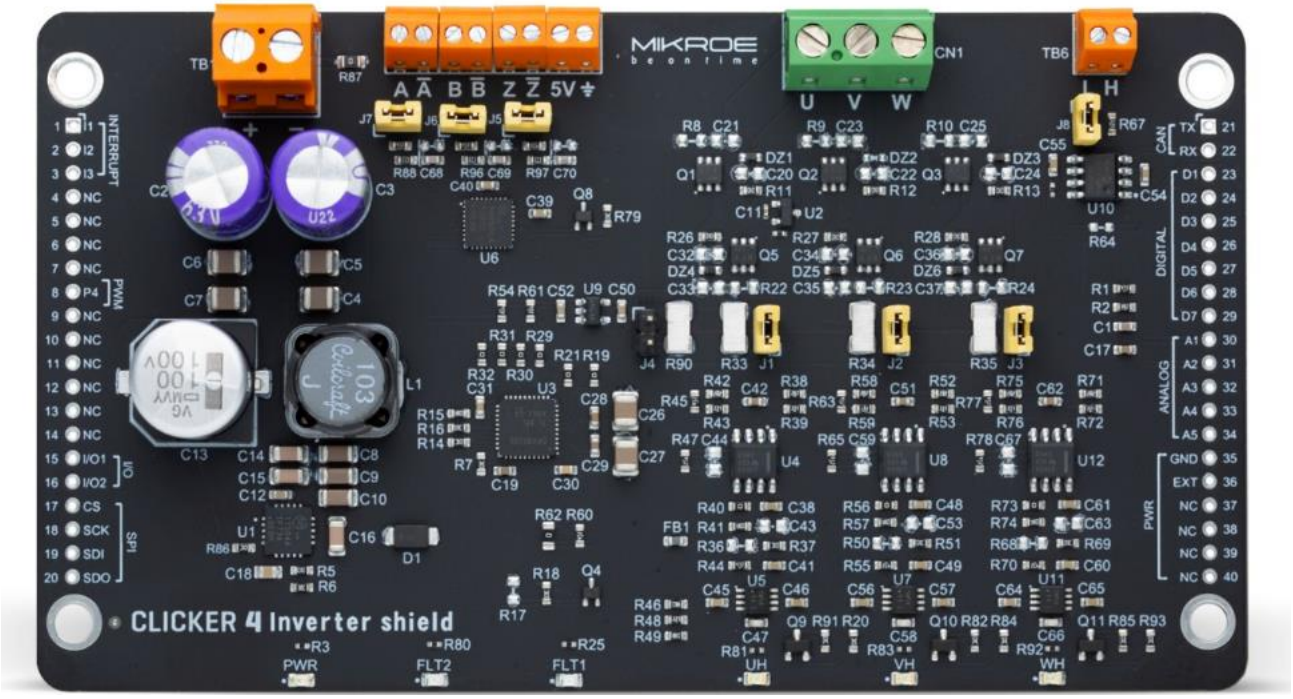

Figure 5.2 Clicker 4 Inverter Shield

Sl no	Functionality	Hardware setting
1	Power supply (DC 12 to 48V)	TB1 (+ and -)
2	Motor connection	CN1 : U, V, W
3	Hall Sensor connection	TB2 ~ 4 (A:Hu, Ā:NC, B:Hv, B̄:NC, Z:Hw, Z̄ :NC) TB5 (5V, GND)
4	Incremental Encoder connection	TB2 ~ 4 (A:A, Ā:Ā, B:B, B̄:B̄, Z:Z, Z̄:Z̄) TB5 (5V, GND)
5	Feedback circuit connection	**Feedback** / **J1** / **J2** / **J3** / **J4** **3-shunt** / Open / Open / Open / Short **1-shunt** / Short / Short / Short / Open

Table 5.2 Clicker 4 Inverter Shield Hardware setting

5.3. Connecting Clicker 4 TMPM4K and Inverter Shield boards

Ensure that “Clicker 4 for TMPM4K” board header pins (1 to 40) are properly connected with “Clicker 4 Inverter shield” header pins (1 to 40) as shown below:

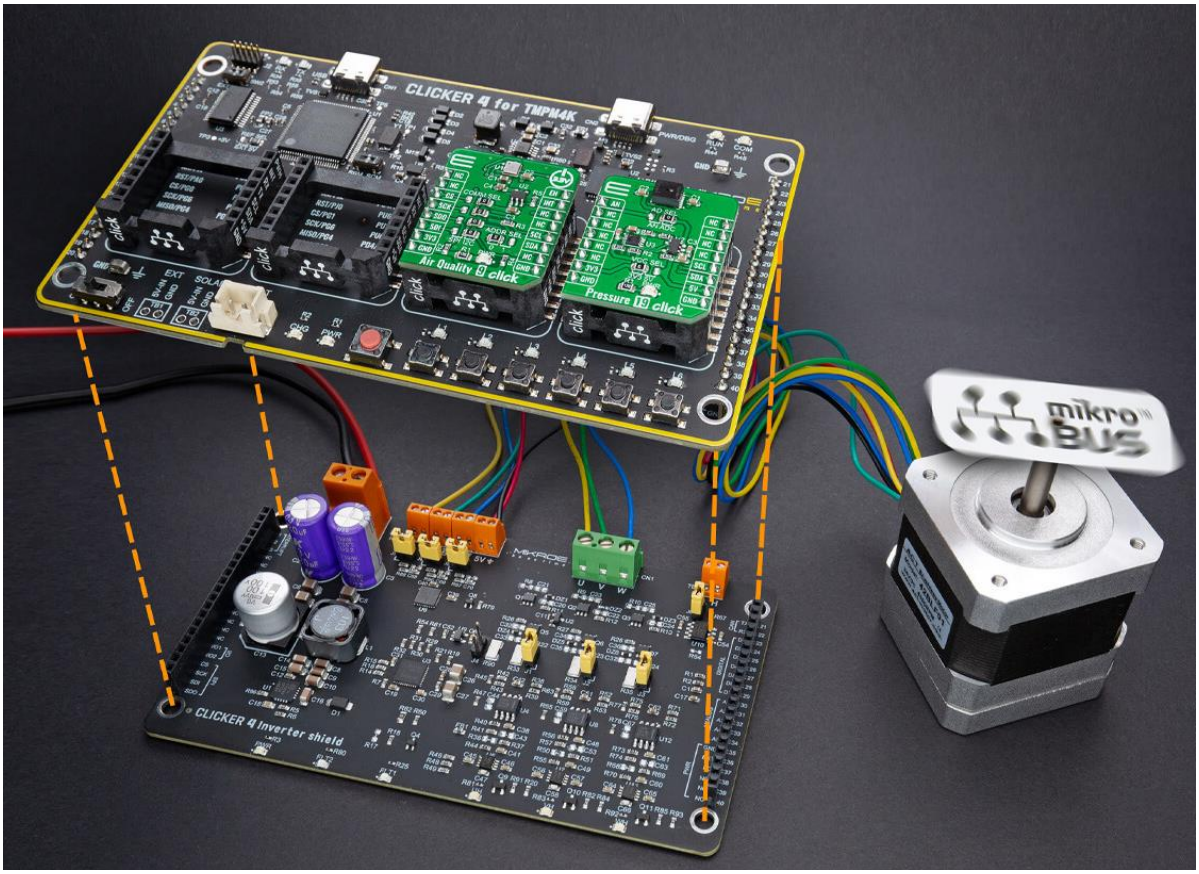

Figure 5.3 Board Assembly Diagram

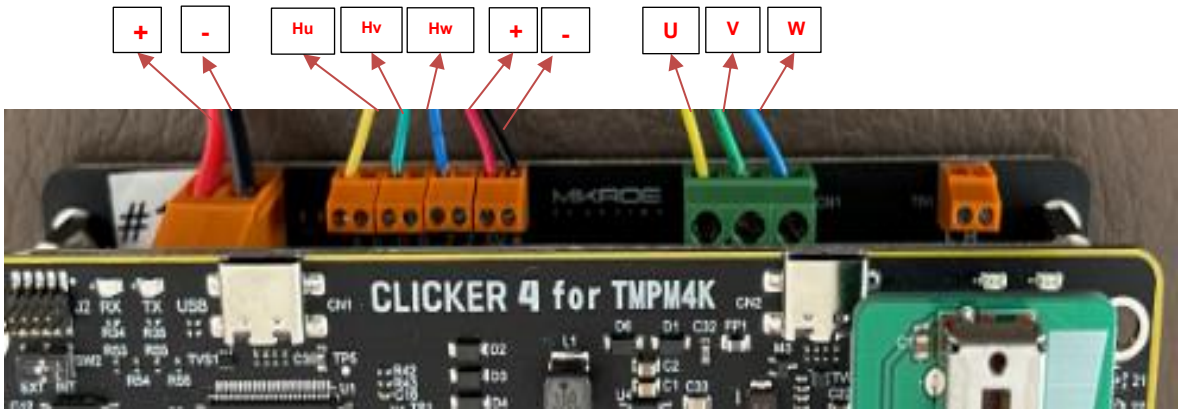

Figure 5.4 Power cable and Motor connection

Ensure the hardware connection for Power supply; Motor connection, Hall Sensor connection and Feedback circuit connection are done as shown in Figure “Power cable and Motor connection” and Table 5.2 "Clicker 4 Inverter Shield Hardware setting".
Once the power supply is switched ON the “Power LED” of both “Clicker 4 for TMPM4K” and “Clicker 4 Inverter shield” will be ON. Now CN1 can be connected to PC where “MCU Motor Studio” PC tool is installed. Once motor start running LED UH, VH, and WH will turned on.

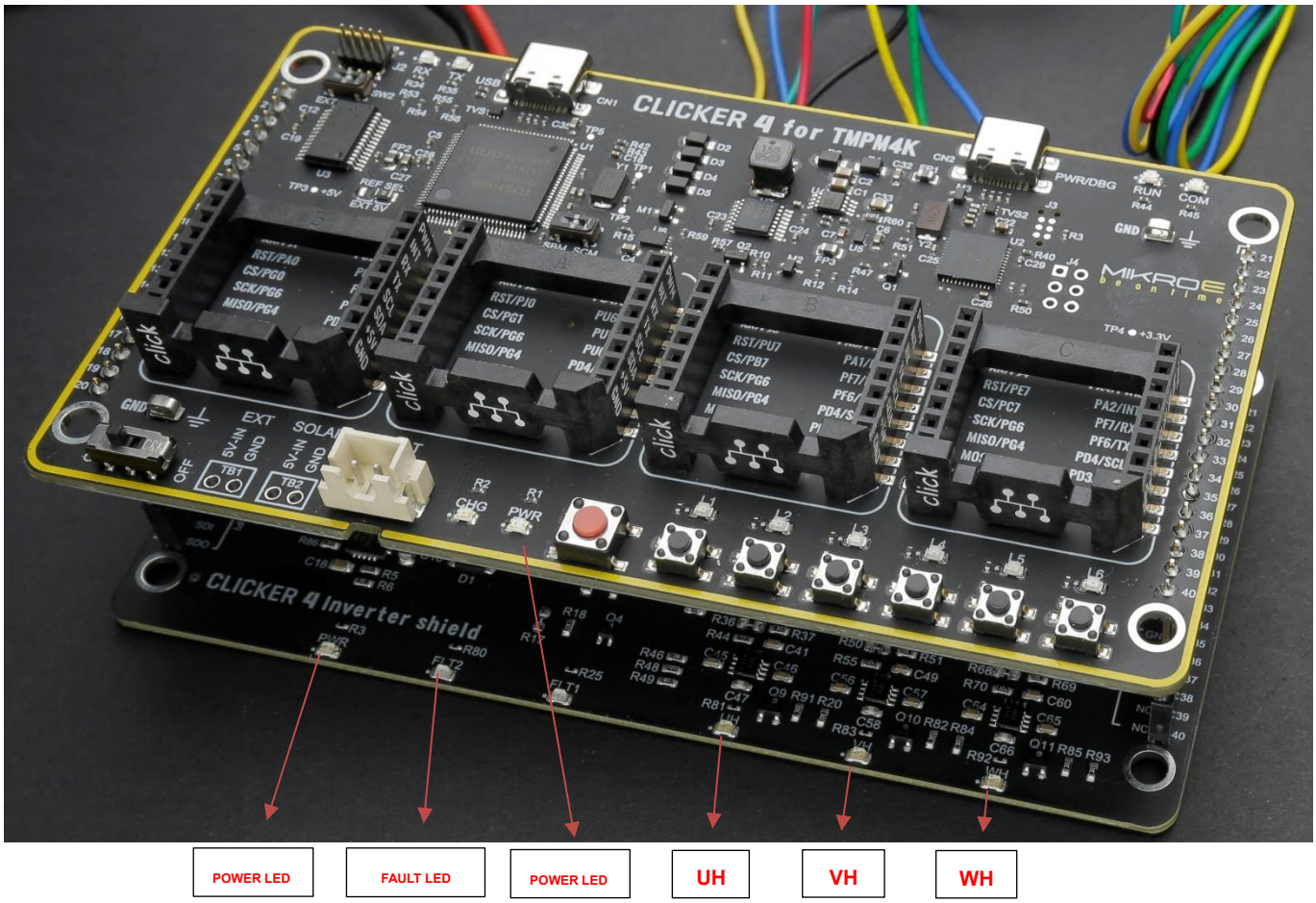

Figure 5.5 PWM indication LED

CN1 USB connector is used for connecting virtual COM port with “Clicker 4 for TMPM4K” board and CN2 USB connector is used for flashing firmware to “Clicker 4 for TMPM4K” board.

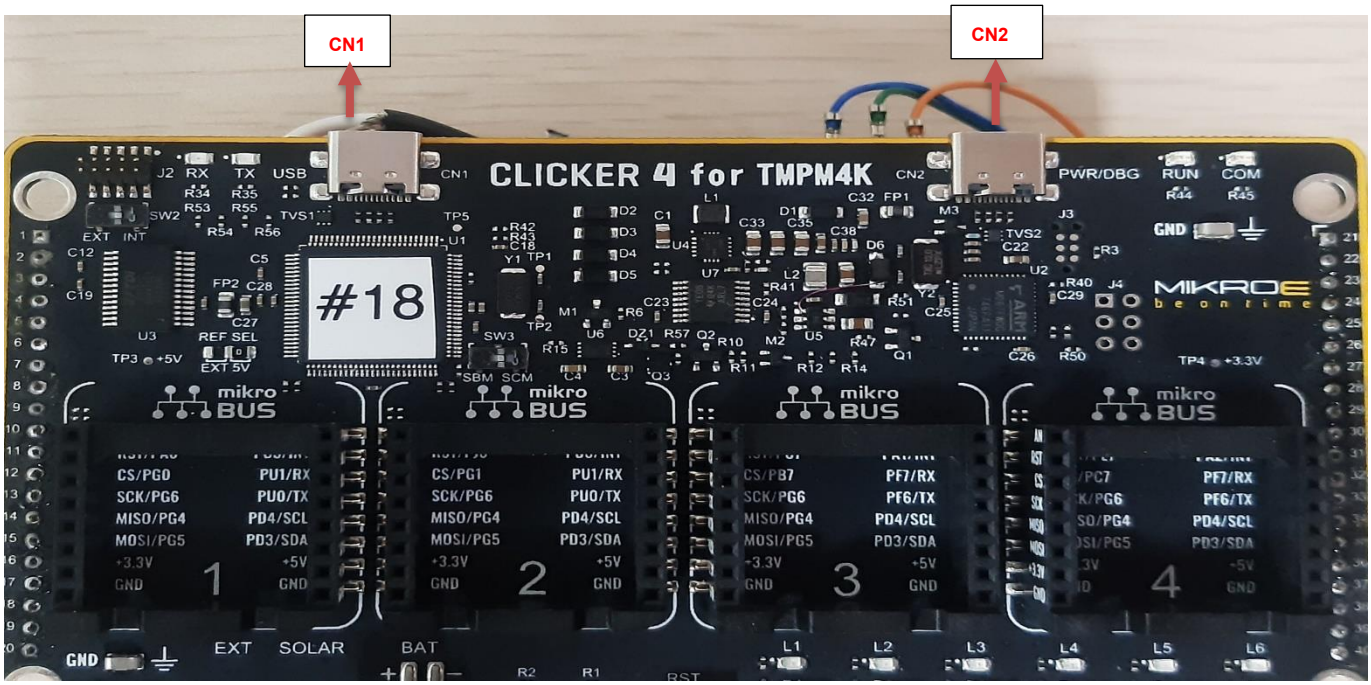

Figure 5.6 Virtual COM port and Debug port connection

6. Standalone Demo

The Standalone demo can be driven in two different ways:

1. Using Slider and Buttons (does not require connection to PC Tool)
2. Demo Control Window of the PC Tool.

6.1. Demo Using Slider and Buttons

The demo can be operated without the need of Motor Studio PC tool. The demo uses “Slider 2 click”, connected to the MicroBus on Clicker 4 MCU board. The “Slider 2 click” along with buttons, can be used to demonstrate Start, stop and speed control.
Slider 2 click - https://www.mikroe.com/slider-2-click

6.1.1. Hardware connection:

Connect the “Slider 2 click” to MicroBus-4.

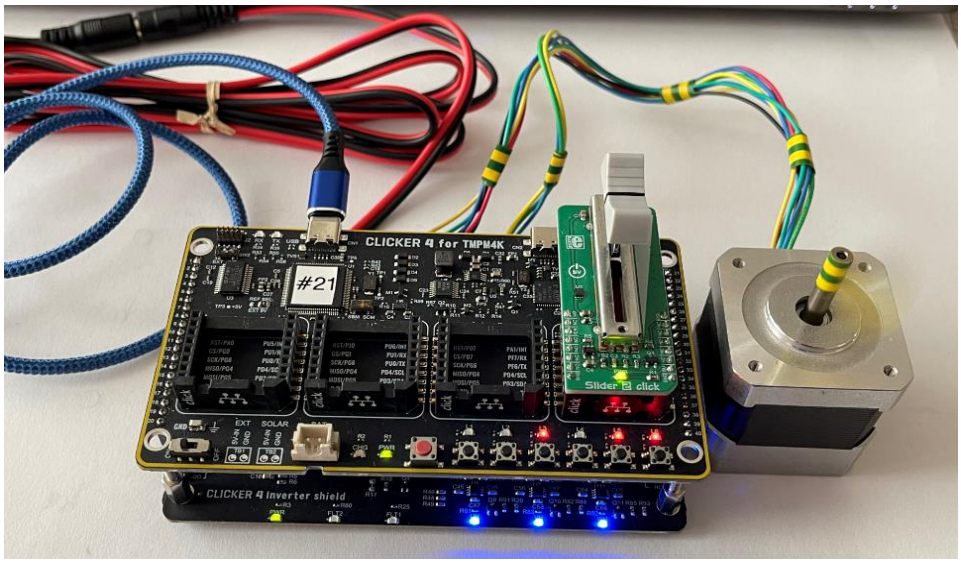

Figure 6.1 Slider 2 Demo Hardware connection

6.1.2. Buttons

Button B6 – Start/Stop the motor. LED L6 indicates the current status, lit when the motor is started. Even started, the motor will not rotate, if the speed selector is set at 0 RPM (see the slider description).

Button B5 – Direction reversal. By default, the motor rotates in clock-wise direction, seen from the top. When B5 is pressed the motor will slow down until full stop and then accelerates in the reverse direction. LED L5 will be lit, when motor rotates in counter-clockwise direction.

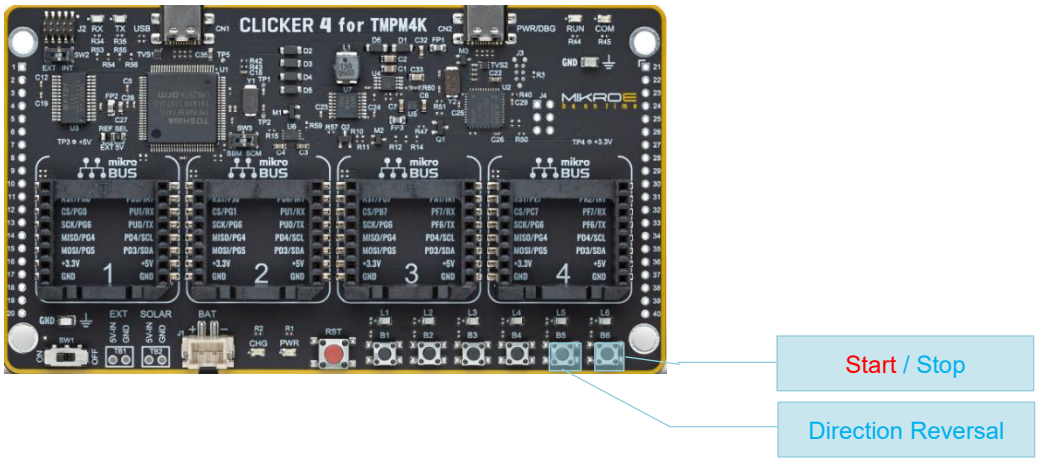

Figure 6.2 Slider 2 Demo Buttons

6.1.3. Slider 2

Slider 2 – it is used to set the target speed of the motor in RPM, ranging from 0 to Max speed as indicated below:

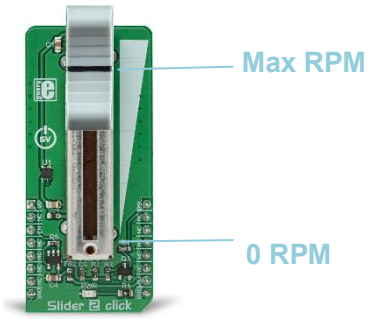

Figure 6.3 Slider 2 Demo Speed control

6.1.4. Status LEDs

The onboard LEDs are used to signal various operational states of the demo:

- **LED L2** – Communication indication. When lit, data communication between Motor Studio FW and PC Tool is ongoing. At power-on the LED will be blinking periodically, indicating an ongoing attempt to establish communication.
- **LED L3** – FOC stage indication. When lit, the demo has switched from the initial sine-wave commutation to Field-Oriented-Control. Default status after power-on is OFF
- **LED L5** – Direction indication. When lit, the motor rotates in counter-clockwise (CCW) direction. Default status after power-on is OFF, e.g. clockwise (CW) direction
- **LED L6** – Motor started indication. When lit the motor is started. Default status after power-on is OFF (motor stopped)

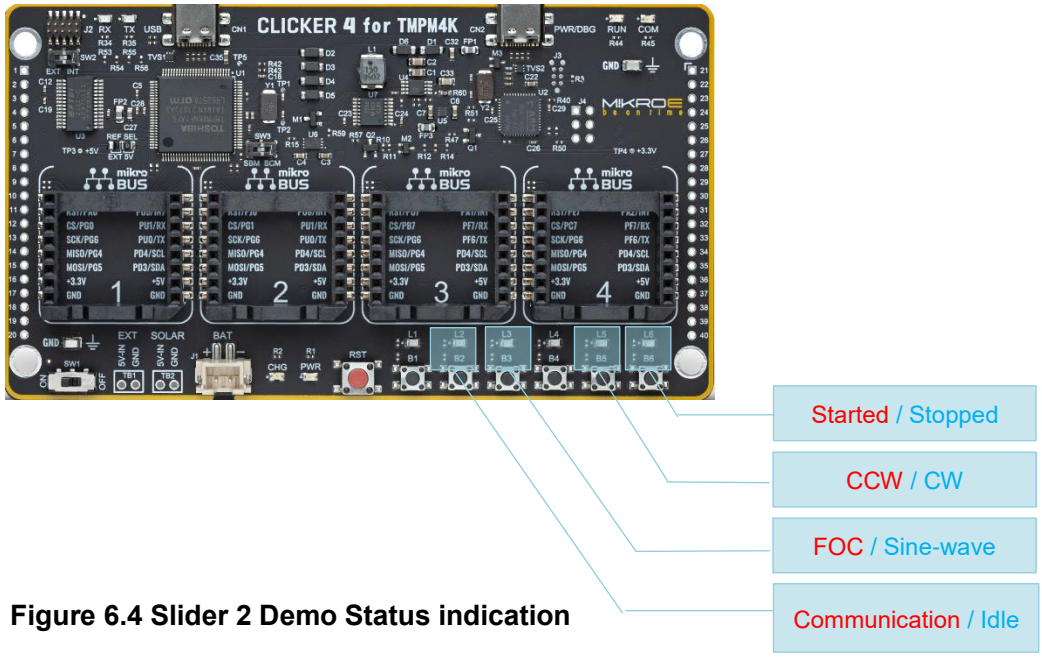

Figure 6.4 Slider 2 Demo Status indication

6.2. Demo Control Window of PC Tool

The Demo Control window provides a basic control for the demo application, offering several buttons for state change and a simple state/transition status indication. However, the set speed is still controlled via the **Slider and the direction via button B5**.
Pressing the Run button will start the motor and the active state will become “Running …”:

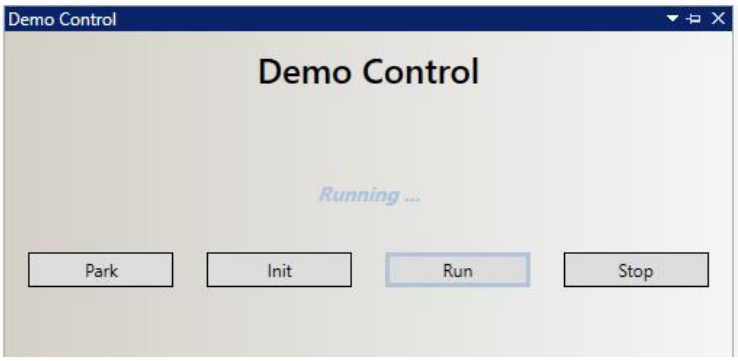

Figure 6.5 Demo control Window during “Run”

Pressing the Stop button will stop the motor and the active state will become “*Stopping …*”. As soon as the motor stands still, it will be changed to “*Idle*”:

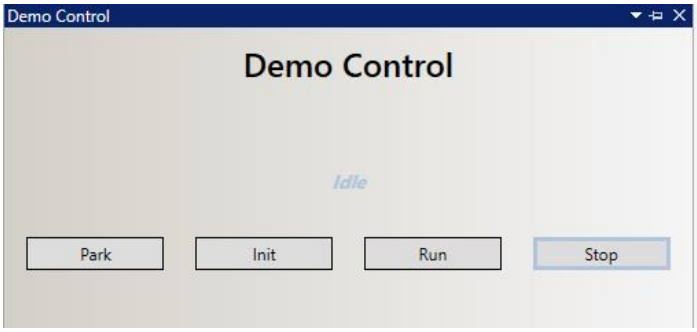

Figure 6.6 Demo Control Window in “Idle”

As both **Park** and **Init** states are not supported, pressing the corresponding buttons will result no action.
LEDs L2, L3, L5 and L6 will be updated accordingly. Mixed usage of Run/Stop and B6 buttons is supported.

7. Firmware building and flashing Procedure:

7.1. Installation Pre-requisites

IAR embedded workbench v8.50.9 has to be installed on Windows PC. (Newer or different version of IAR can be used at own risk.)

7.2. Applying FreeRTOS Patch

"MCU Motor Studio" firmware uses FreeRTOS Kernel V10.2.1. The project folder structure do not contain the FreeRTOS source code, user has to download the FreeRTOS open source code and apply patch using the Patch file available in release package.

Please follow the below mentioned procedure to prepare the "FreeRTOS \ source" folder:

1. Download the FreeRTOS from the path " https://sourceforge.net/projects/freertos/files/FreeRTOS/V10.2.1/FreeRTOSv10.2.1.zip/download" and unzip the downloaded file.

   Note:
   a. If required, please accept website cookies to download FreeRTOS kernel.
   b. Newer or different version of FreeRTOS can be used at own risk.

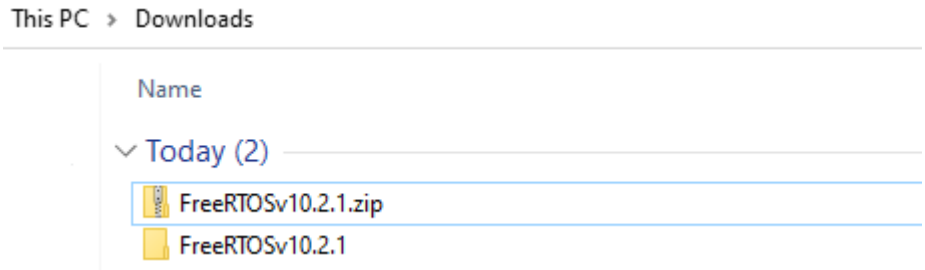

Figure 7.1 Download and unzip FreeRTOS kernel

2. Download Git for windows from the path, "https://git-scm.com/". Install Git by executing downloaded .exe.

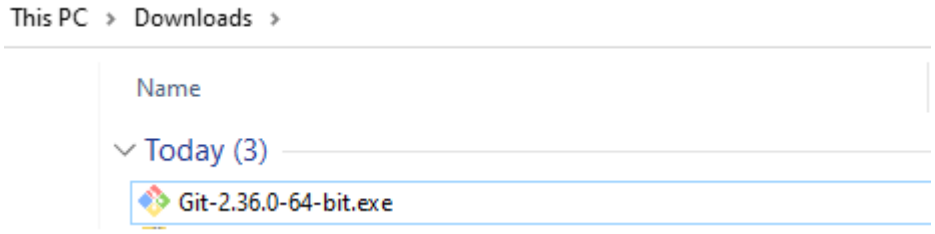

Figure 7.2 Download and install Git

3. Unzip the release package "MCUMotorStudio_1.1.zip".

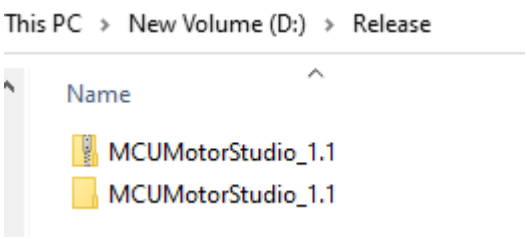

Figure 7.3 Unzip Release package

4. Unzip Firmware source code available at "MCUMotorStudio_1.1\Code".

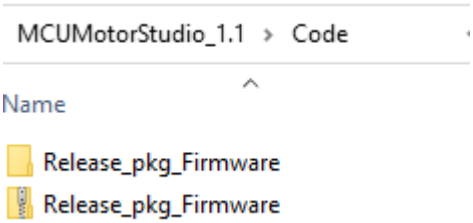

Figure 7.4 Unzip Firmware package

5. Create a temporary folder by name “patch” at any suitable path outside folder “MCUMotorStudio_1.1”.

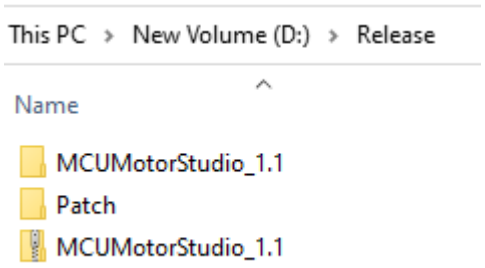

Figure 7.5 Create “patch” folder

6. Copy FreeRTOS source folder “FreeRTOSv10.2.1\FreeRTOS\Source” and patch file “MCUMotorStudio_1.1\Code\Release_pkg_Firmware\FreeRTOS.patch” to “patch” folder as shown below.

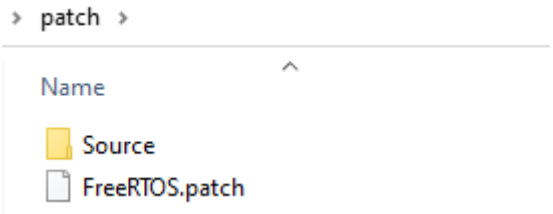

Figure 7.6 Copy FreeRTOS source and patch file

7. Open Git Bash application by right clicking on “patch” folder and clicking “Git Bash Here”.

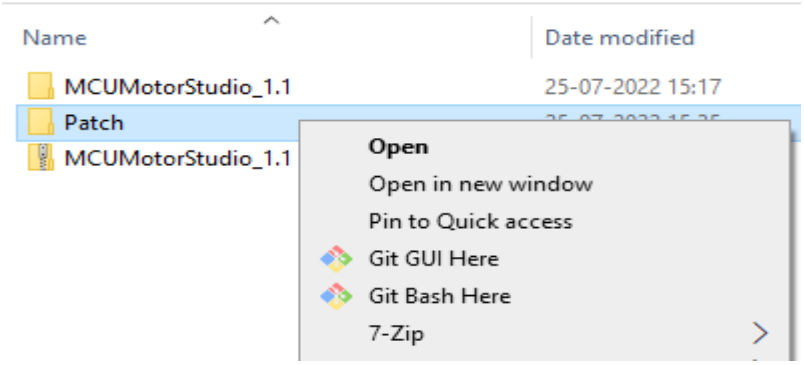

Figure 7.7 Open Git Bash

8. Execute following command in Git Bash. Git bash works on Linux style commands.

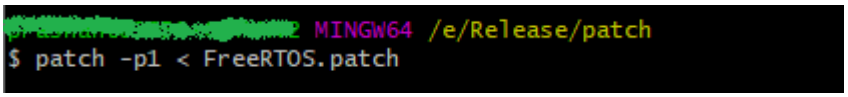

Figure 7.8 FreeRTOS patch command

9. If the patching is successful, message as shown below will appear.

Figure 7.9 FreeRTOS patch message

10. Finally copy "\patch\source" folder to "MCUMotorStudio_1.1\Code\Release_pkg_Firmware\Source\FreeRTOS" as shown below.

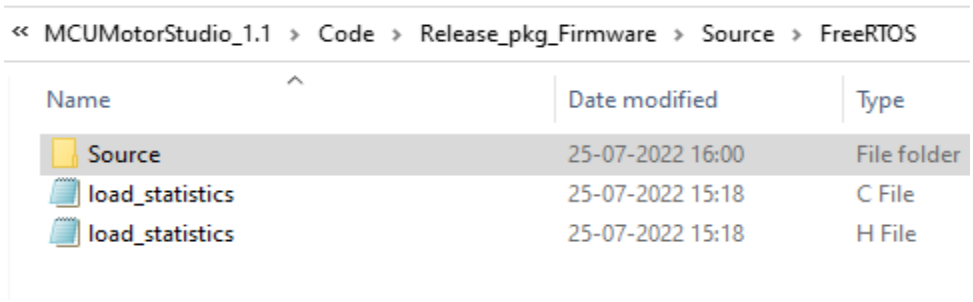

Figure 7.10 FreeRTOS patch message

11. Downloaded FreeRTOS kernel and temporary folder "patch" can be deleted after patching procedure.

7.3. Build procedure using existing IAR Embedded workbench Project

1. Open IAR embedded workbench project "MotorControl" in \Projects\ MotorControlM37x\IAR folder using IAR embedded workbench software.

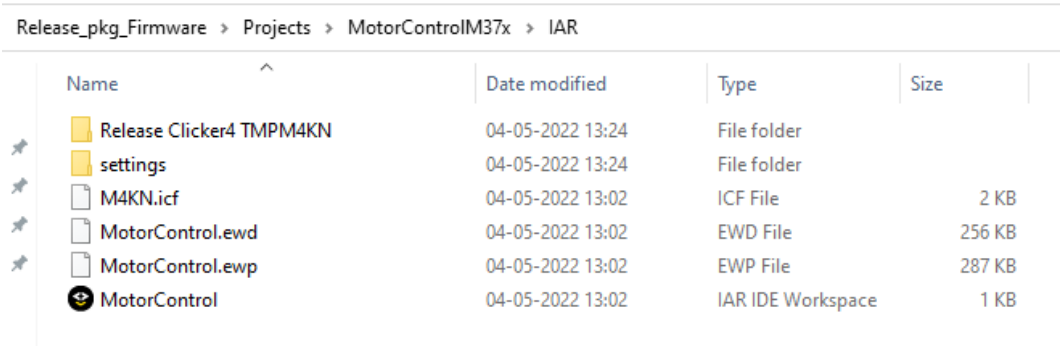

Figure 7.11 IAR project

2. IAR project configuration “Release Clicker4 TMPM4KN” will be opened by default.

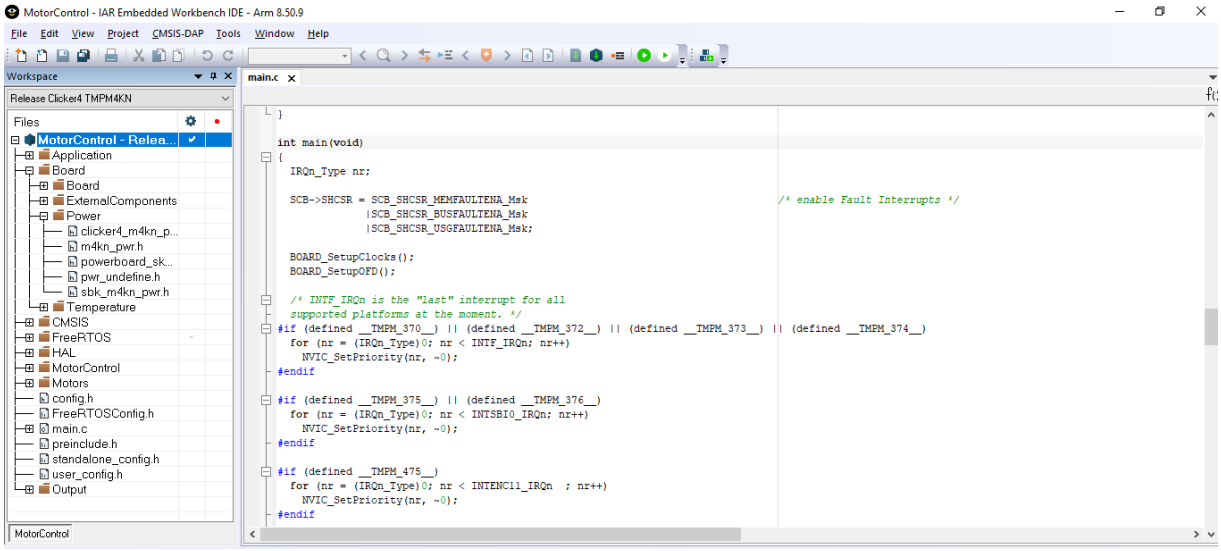

Figure 7.12 IAR project window

3. Build target files using [Project] -> [Rebuild All].

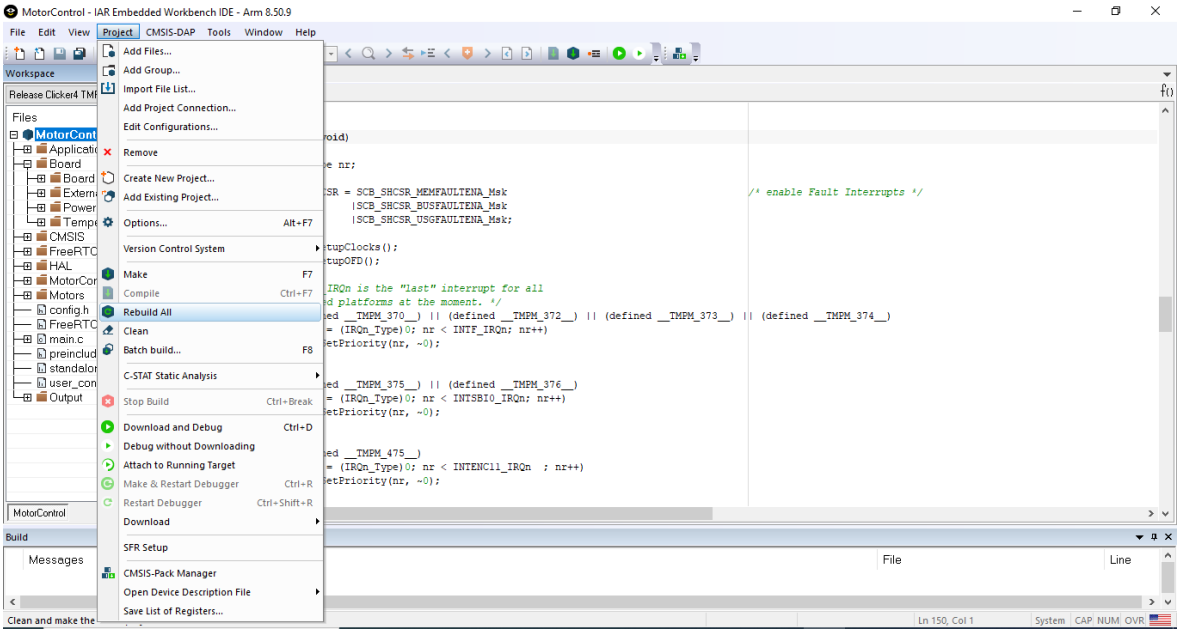

Figure 7.13 Build IAR project

4. Ensure CN2 USB connector of “Clicker 4 for TMPM4K” board is connected to PC using C-type USB cable.

5. Flash the firmware to MCU using [Project] -> [Download] -> [Download active application].

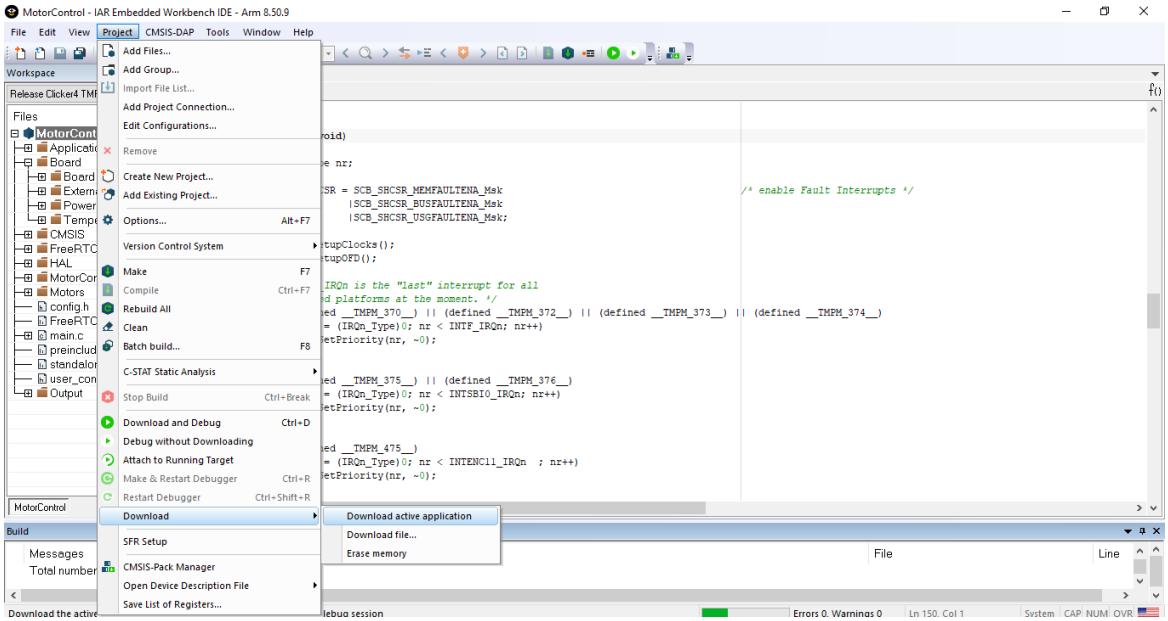

Figure 7.14 Download IAR project

8. Revision History

Revision	Date	Changes
1.0.0	2022-05-05	Baselined Version.
1.1.0	2022-12-01	Internal review comments are incorporated.

Table 8-1 Revision History

Trademarks

- FreeRTOS™ is a trademark of Amazon Web Services, inc in the US and/or elsewhere. All rights reserved.
- Microsoft® and Windows® are either registered trademarks Microsoft Corporation in the United States and/or elsewhere. All rights reserved.
- Arm® , Cortex® ,Cortex®-M3, Cortex®-M4,Keil® and µVision® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
- Click boards™ is a trademark of MIKROELEKTRONIKA. All rights reserved.
- FTDI may be registered trademarks of “Future Technology Devices International Limited”. All rights reserved.
- IAR Systems® and IAR Embedded Workbench® are registered trademarks are owned by IAR Systems. All rights reserved
- SEGGER and J-Link are trademarks or registered trademarks of SEGGER Microcontroller GmbH & Co. KG. All rights reserved.

Other Company names, product names and service names mentioned herein may be trademarks of their respective companies.

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**