

**Sensorless Sinusoidal Brushless Motor Drive  
Circuit Using TC78B011FTG**

# **Reference Guide**

**RD239-RGUIDE-01**

---

**Toshiba Electronic Devices & Storage Corporation**

## Table of Contents

<b>1. Introduction</b>	<b>5</b>
<b>2. Specifications</b>	<b>6</b>
2.1. System Configuration	7
2.2. Circuit Block Diagram of PCB	8
2.3. PCB Appearance and Component Layout	9
2.4. GUI Appearance	12
<b>3. Schematic, Bill of Material, PCB Pattern, Software</b>	<b>13</b>
3.1. Schematic	13
3.2. Bill of Material	13
3.3. PCB Pattern	13
3.4. GUI Software	15
3.5. USB-I2C Converter Software	15
<b>4. Description of This Design</b>	<b>16</b>
4.1. PCB Description	16
4.1.1. Connectors	16
4.1.2. Switches	20
4.1.3. Potentiometer (VR1)	22
4.1.4. LEDs	23
4.1.5. Jumpers	24
4.2. GUI Description	25
4.2.1. Speed Curve Display Section	25
4.2.2. Connection Control Section	26
4.2.3. MCD Status Display Section	26

4.2.4.	Save and Load Setting File Buttons.....	26
4.2.5.	Tooltip .....	27
4.2.6.	Tabs.....	27
4.3.	USB-I2C Converter Description .....	30
<b>5.</b>	<b>Operation .....</b>	<b>31</b>
5.1.	Mode 1: GUI Control Operation .....	31
5.1.1.	Starting Procedure.....	32
5.1.2.	Stopping Procedure .....	32
5.2.	Mode 2: External MCU Control Operation .....	33
5.2.1.	Starting Procedure.....	34
5.2.2.	Stopping Procedure .....	34
5.3.	Mode 3: Manual Operation .....	35
5.3.1.	Starting Procedure.....	36
5.3.2.	Stopping Procedure .....	36
5.4.	Mode 4: Minimum Hardware Operation with Main Board Only .....	37
5.4.1.	Starting Procedure.....	38
5.4.2.	Stopping Procedure .....	38
5.5.	Writing NVM of MCD.....	39
<b>6.</b>	<b>Setting Up GUI and USB-I2C Converter .....</b>	<b>40</b>
6.1.	GUI Setup.....	40
6.2.	USB-I2C Converter Setup .....	40
<b>7.</b>	<b>Troubleshooting .....</b>	<b>41</b>
7.1.	I2C Connection Error in GUI.....	41
7.2.	Speed Control Not Working .....	41
7.3.	Sub Board Switches and Potentiometer Not Working .....	41

**8. Precautions for Evaluation (To Prevent Electric Shock, Burn Injury, etc.) .....41**

## 1. Introduction

This Reference Guide (hereafter referred to as this guide) describes the reference design (hereafter referred to as this design) of Sensorless Sinusoidal Brushless Motor Drive Circuit Using TC78B011FTG.

The use of brushless motors is increasing in various applications and as a result the requirement for a stable brushless motor control with less noise and customizable speed profile is also increasing. This design is based on Toshiba's [TC78B011FTG](#) (hereafter referred to as MCD (Motor Control Driver)) which is a 3-phase PWM chopper predriver, it uses the sensorless sinusoidal control method to run a brushless motor. Sinusoidal drive motor control makes it possible to run the motor with less noise and vibration. It also provides closed loop motor speed control without requiring any external microcontroller for a more stable motor speed control. The speed profile curve can be set in this MCD for precise motor control.

The motor control settings are done in MCD registers via I2C communication. Once these settings are done, the motor can be controlled via external signals which affect motor speed, direction, brake etc. The motor speed can be controlled via three different input methods, which includes PWM, analog voltage and I2C. The motor control setting of MCD registers can be stored in its internal non-volatile memory (NVM), and the registers are loaded when the MCD starts next time. Therefore, stable motor control can be achieved by simply performing various settings without an external microcontroller.

This design also uses Toshiba's [TPH1R204PB](#) power MOSFETs to drive the brushless motor and [TCR1HF50B](#) LDO to produce a 5 V power supply. We also developed a GUI to easily perform the MCD register setting via I2C communication. An USB-I2C converter is used for communication between the GUI (on PC) and the board (PCB) of this design is also prepared.

Applications of this design include server fans, blowers, pumps, vacuum cleaners, etc.

## 2. Specifications

Table 2.1 lists the main specifications of the board (PCB) of this design.

**Table 2.1 Specifications of This Design**

Item	Specifications
Input Power Supply Voltage	DC 11 to 27 V
Input Control Signal Voltage	DC 5 V
Rated Output Current	20 A (Max.)
Driven Motor Type	Brushless Motor
Motor Control Type	Sensorless Sinusoidal Drive
MCD Setting Method	I2C
Main Motor Control Signals	Brake, Direction, Speed signal
Board Size	90 x 50 mm (Consisting Main Board of 39 x 50 mm)
Board Layer Configuration	4-layer through-hole (outer layer 35 μm, inner layer 70 μm)

Table 2.2 lists the main specifications of the GUI of this design.

**Table 2.2 Specifications of the GUI of This Design**

Item	Specifications
GUI Platform	Processing (4.2)
Library Used	ControlP5 (2.2.6)
Operating System of PC	Windows 10

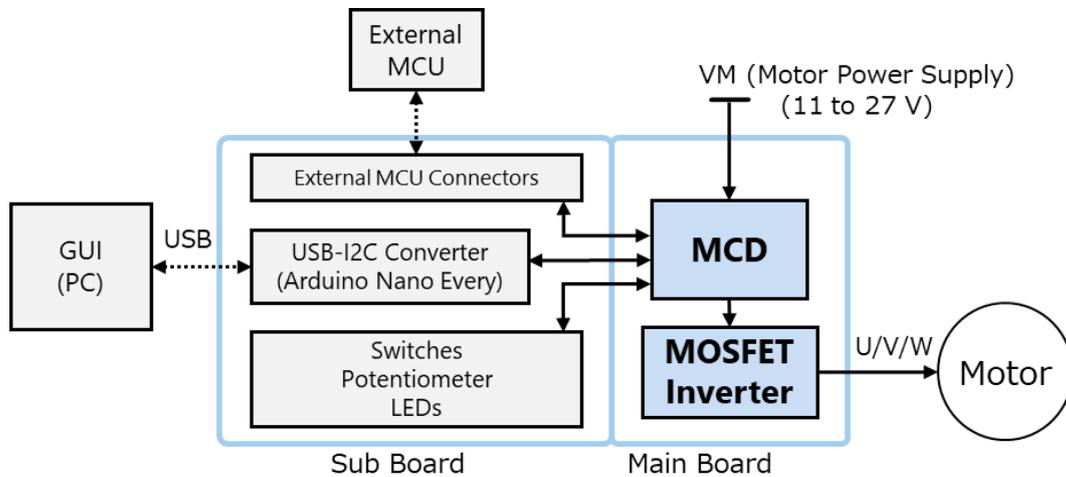
Table 2.3 lists the main specifications of USB-I2C converter used in this design.

**Table 2.3 Specifications of the USB-I2C Converter of This Design**

Item	Specifications
Hardware	Arduino Nano Every
Software Development Environment	Arduino Windows Application (1.8.19)
Library Used	Wire.h
Operating System of PC	Windows 10

### 2.1. System Configuration

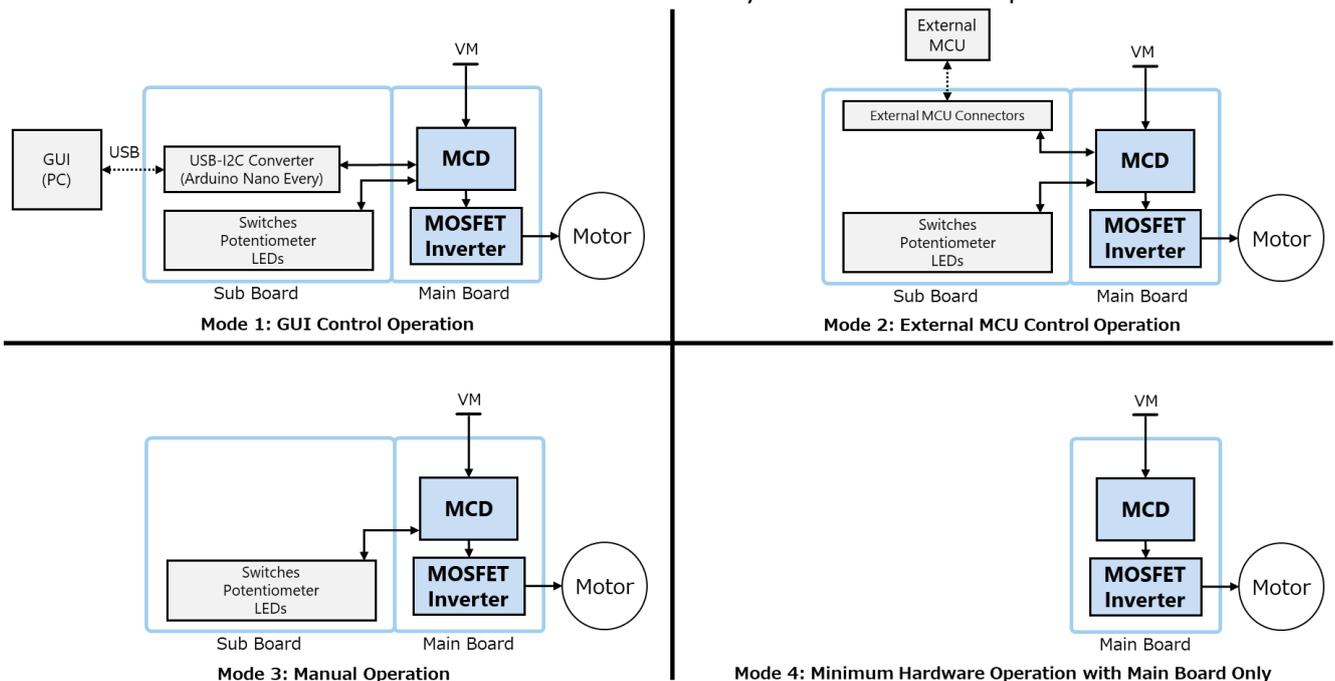
Fig. 2.1 shows the complete system configuration of this design consisting of main board and sub board. This simple design can be used to easily drive a brushless motor via sensorless sinusoidal method.



**Fig. 2.1 System Configuration of This Design**

This design can be operated in various modes. Fig. 2.2 shows all four operation modes of this design. Description of this modes is as follows:

- Mode 1 allow the user to control the Motor using GUI or sub board.
- Mode 2 allows the user to control the Motor using External MCU or sub board.
- Mode 3 allows the user to control the Motor using sub board without any external connections.
- Mode 4 allows the user to run the Motor using only the main board. This shows that TC78B011FTG can run a brushless motor with very less hardware requirement.



**Fig. 2.2 All Modes of Operation of This Design**

### 2.2. Circuit Block Diagram of PCB

Fig. 2.3 shows the block diagram of the board (PCB) of this design.

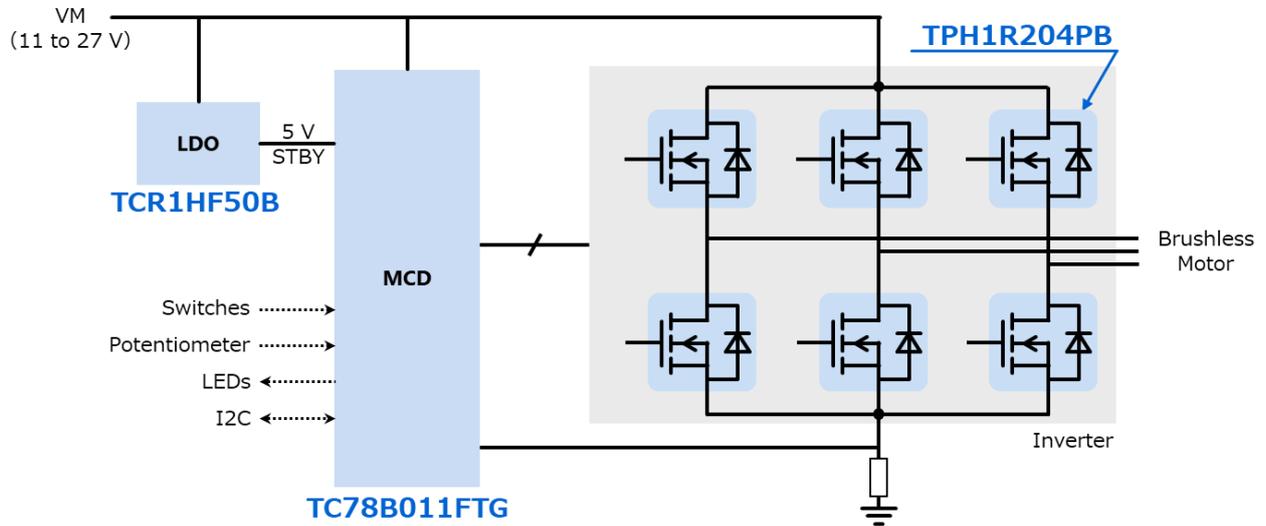
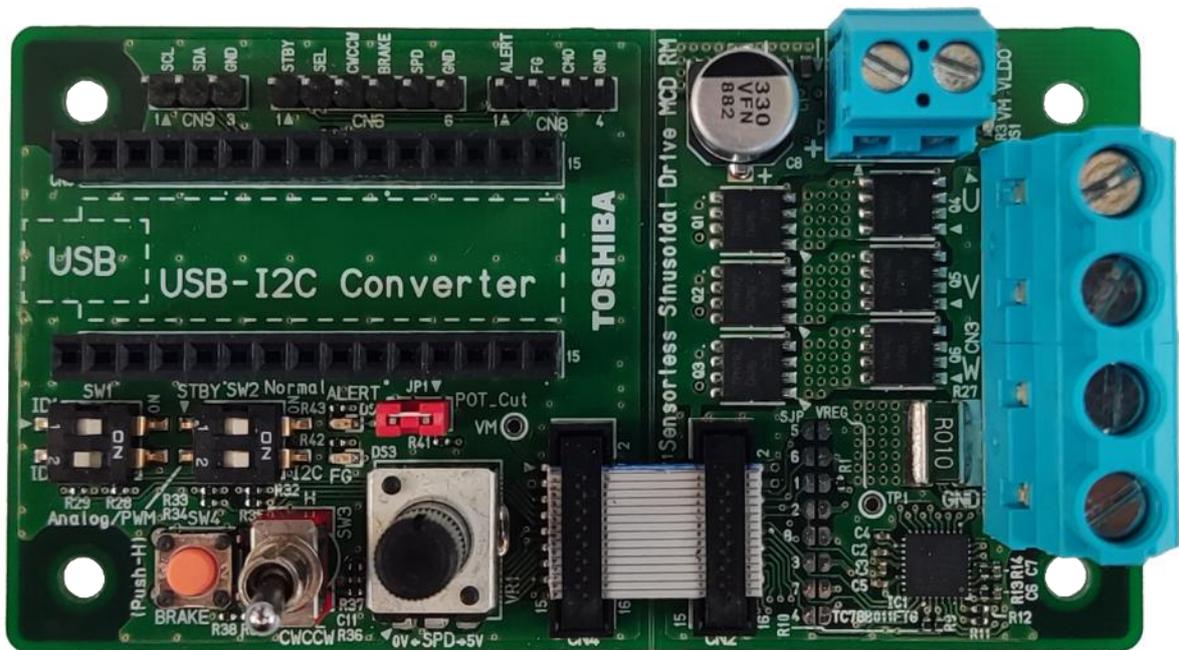


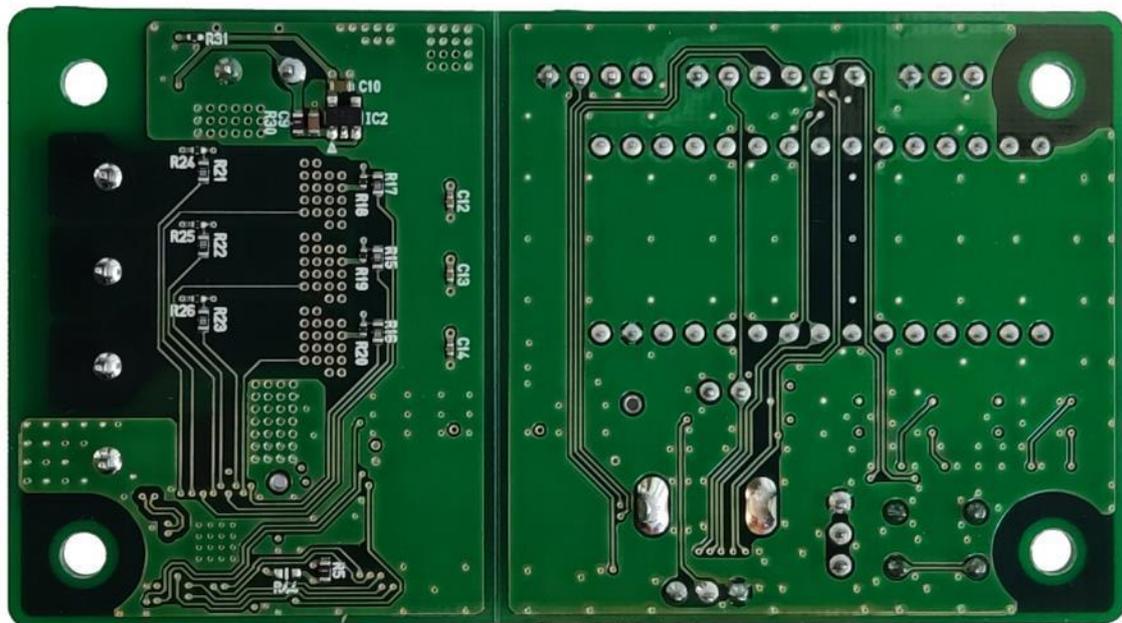
Fig. 2.3 Circuit Block Diagram of This Design

### 2.3. PCB Appearance and Component Layout

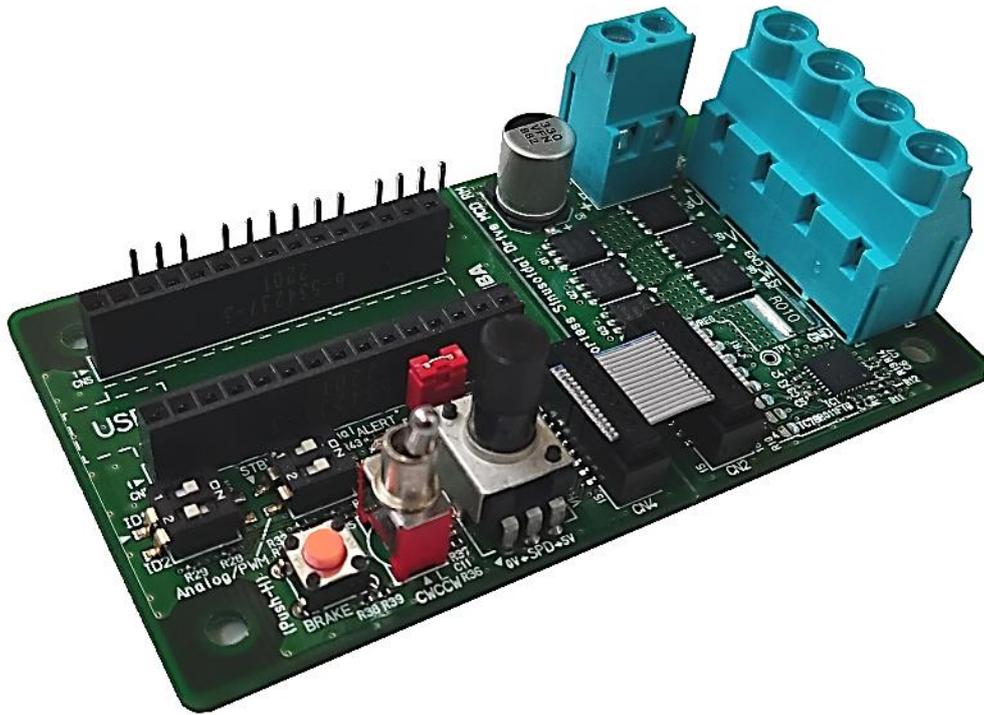
Fig. 2.4, 2.5 and 2.6 show the appearance of the board of this design, and Fig. 2.7 and 2.8 show the layout of main components.



**Fig. 2.4 Sensorless Sinusoidal Brushless Motor Drive Circuit Using TC78B011FTG (Front Side View of the PCB)**



**Fig. 2.5 Sensorless Sinusoidal Brushless Motor Drive Circuit Using TC78B011FTG (Back Side View of the PCB)**



**Fig. 2.6 Sensorless Sinusoidal Brushless Motor Drive Circuit Using TC78B011FTG  
(Side View of the PCB)**

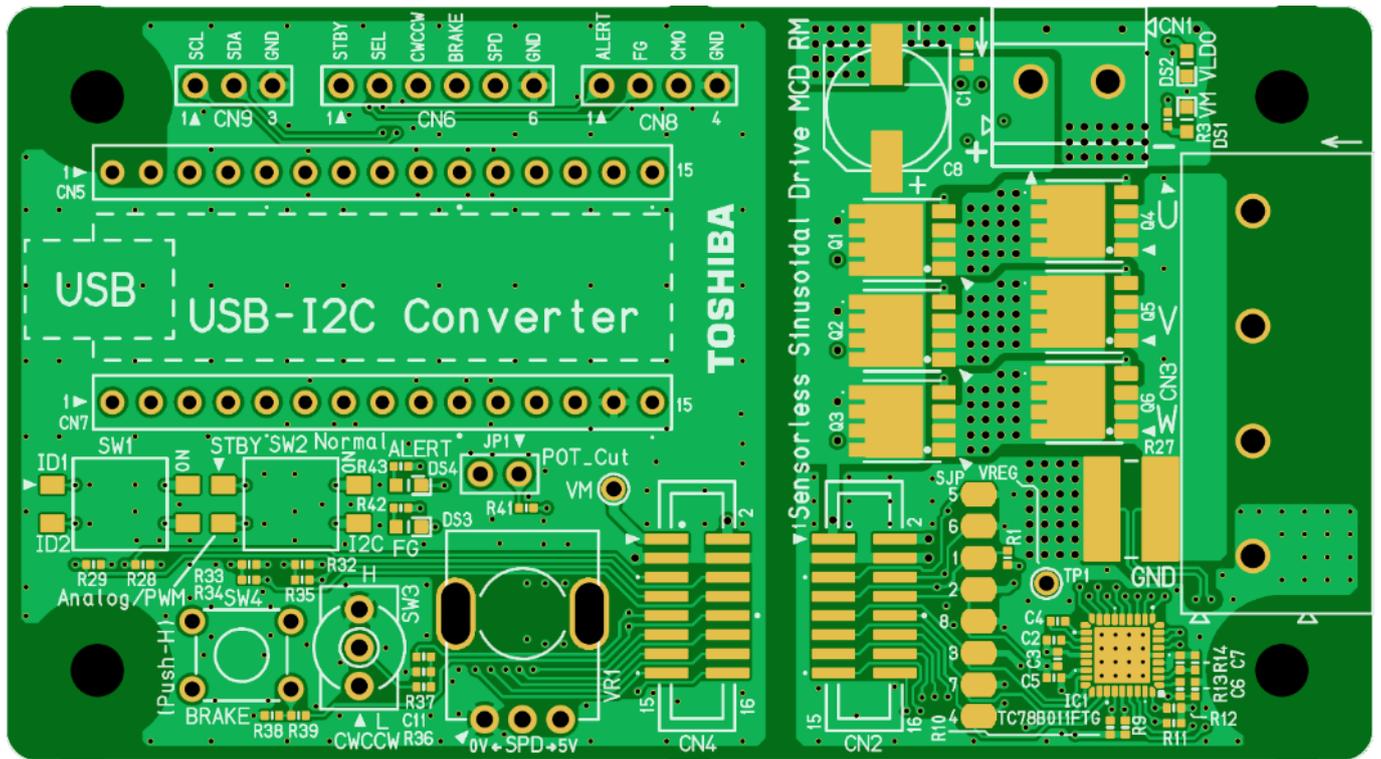


Fig. 2.7 Main Component Layout (Front)

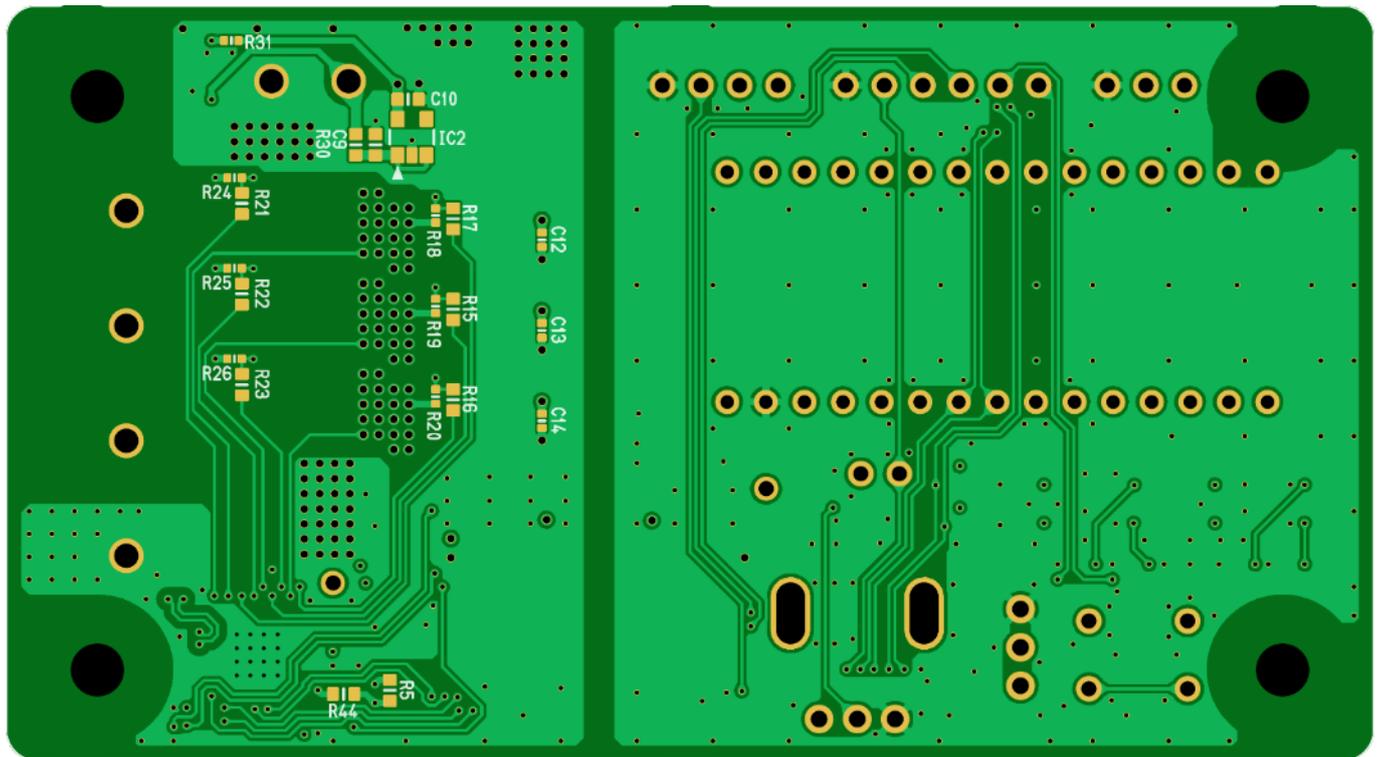


Fig. 2.8 Main Component Layout (Back)

### 2.4. GUI Appearance

Fig. 2.9 shows the GUI of this design.

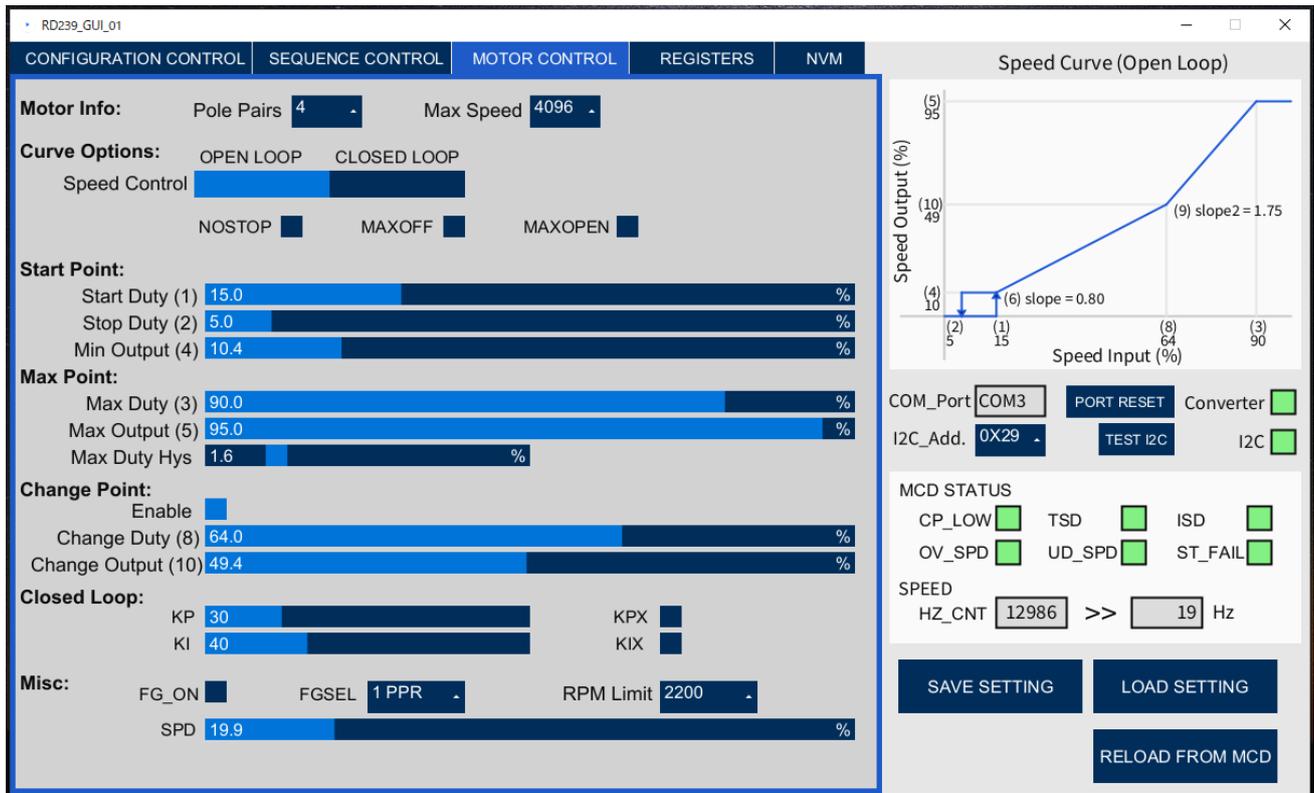


Fig. 2.9 GUI of This Design

## 3. Schematic, Bill of Material, PCB Pattern, Software

### 3.1. Schematic

Refer to the following file:

RD239-SCHEMATIC-xx.pdf

(xx is the revision number.)

### 3.2. Bill of Material

Refer to the following file:

RD239-BOM-xx.pdf

(xx is the revision number.)

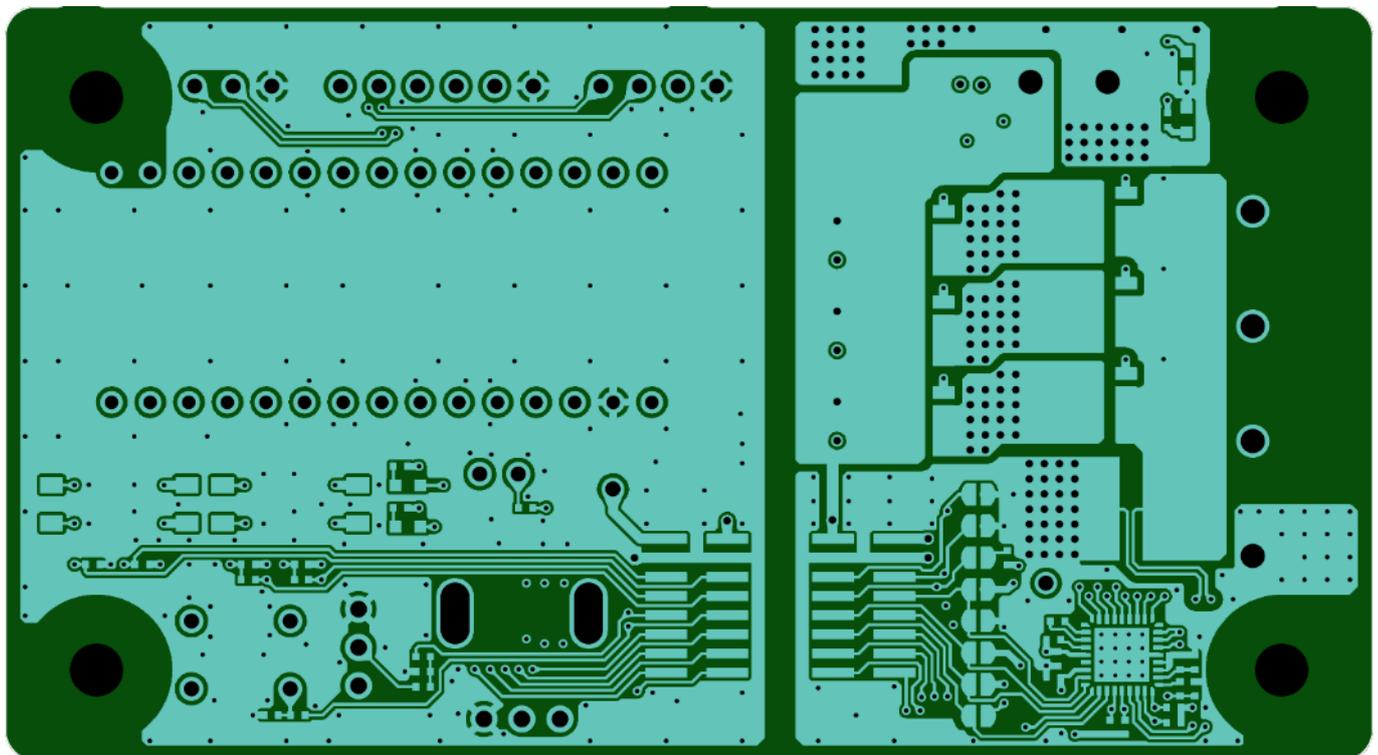
### 3.3. PCB Pattern

Fig. 3.1 shows the PCB pattern of this inverter.

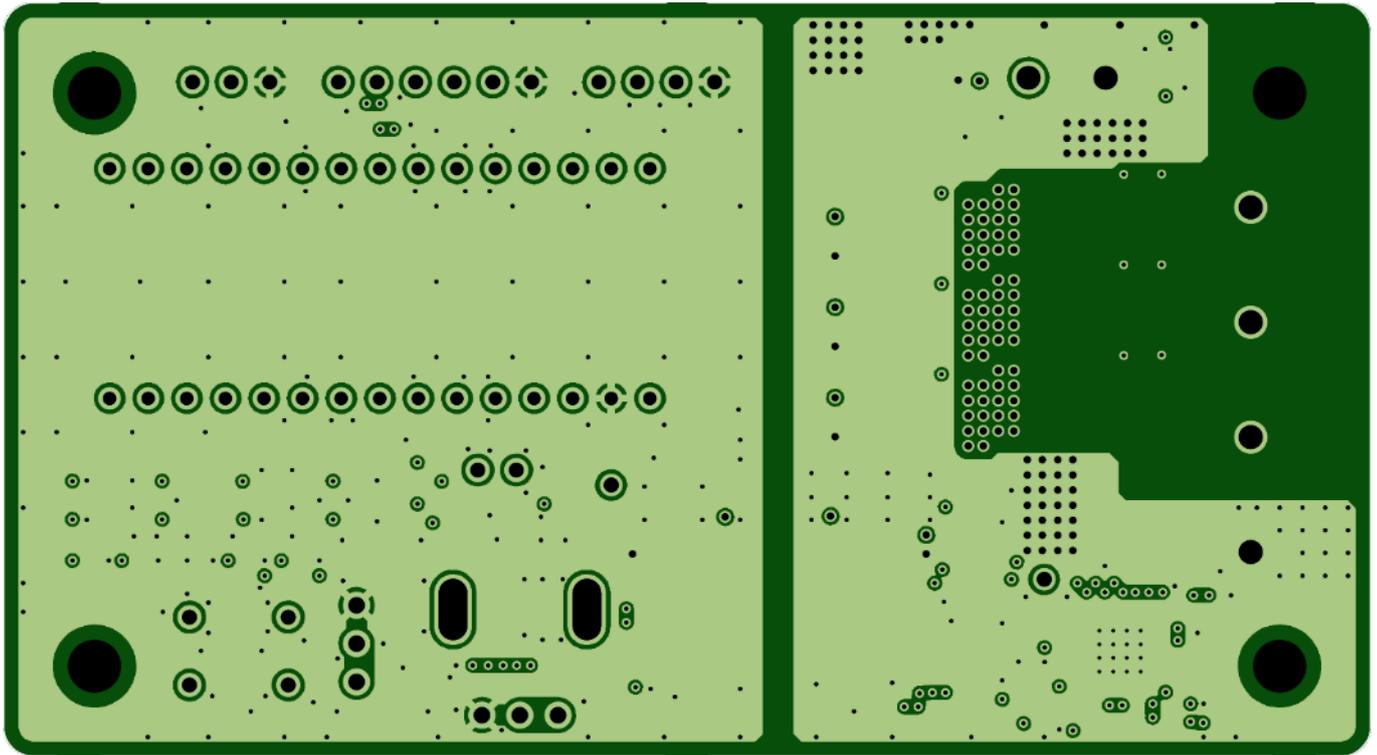
Also refer to the following file:

RD239-LAYER-xx.pdf

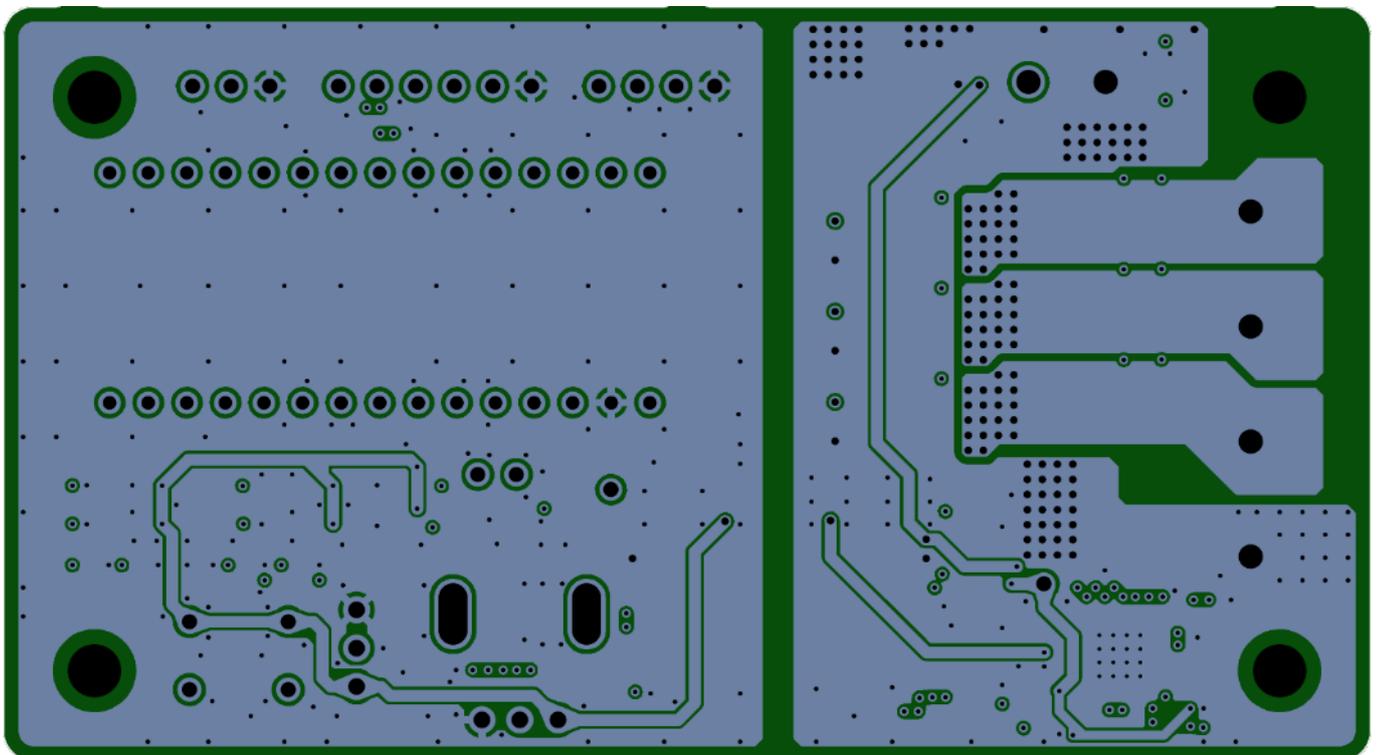
(xx is the revision number.)



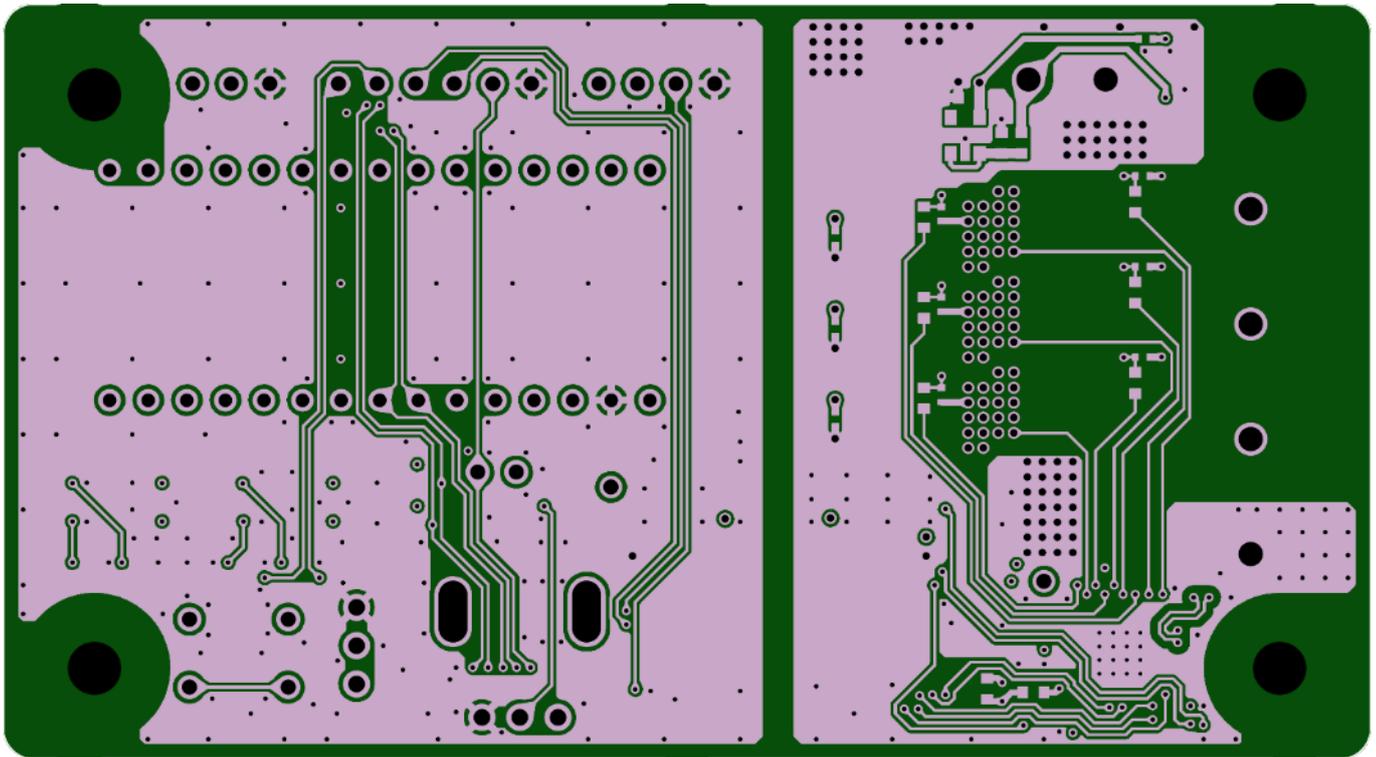
<Layer 1 Front Side>



<Layer 2>



<Layer 3>



<Layer 4>

**Fig. 3.1 PCB Pattern (Front View)**

### 3.4. GUI Software

Refer to the following file:

RD239-GUI-xx.zip

(xx is the revision number.)

### 3.5. USB-I2C Converter Software

Refer to the following file:

RD239-USB\_I2C\_Converter-xx.zip

(xx is the revision number.)

## 4. Description of This Design

### 4.1. PCB Description

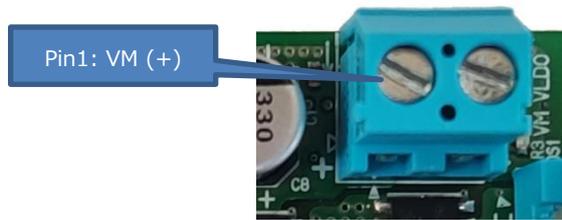
Name and Function of various Components of the PCB are described in this section.

#### 4.1.1. Connectors

Connectors in this design are described in this section.

##### 4.1.1.1. Motor Power Supply Input Connector (CN1)

This connector is used to supply input DC power for the motor power supply (VM). A DC voltage between 11 V and 27 V can be input. TB005-762-02BE (CUI Devices) connector is used.



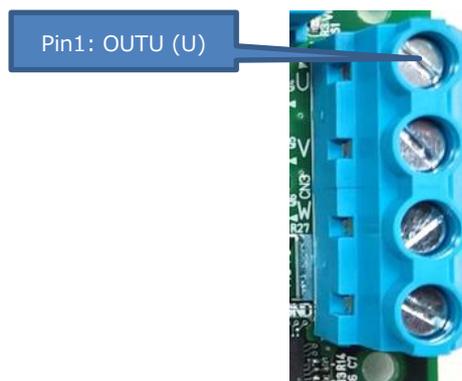
**Fig. 4.1 Motor Power Supply Input Connector (CN1)**

**Table 4.1 Motor Power Supply Input Connector (CN1) Specifications**

Pin	Net Name	Description
1	VM (+)	DC voltage (+) (11 V to 27 V)
2	GND (-)	DC voltage (-) (GND)

##### 4.1.1.2. Motor Connector (CN3)

This connector is used to connect a brushless motor. TB005-762-04BE (CUI Devices) connector is used.



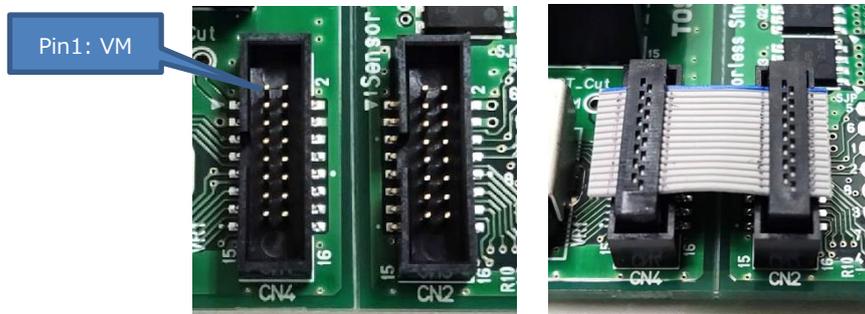
**Fig. 4.2 Motor Connector (CN3)**

**Table 4.2 Motor Connector (CN3) Specifications**

Pin	Net Name	Description
1	OUTU	U of the brushless motor
2	OUTV	V of the brushless motor
3	OUTW	W of the brushless motor
4	GND	GND

### 4.1.1.3. Board to Board Connectors (CN2 and CN4)

This design has a main board which drives motor and a sub board which provides additional features to the user. These connectors CN2 and CN4 are used for connecting these two boards. 3220-16-0300-00 (CNC Tech) connectors are used.



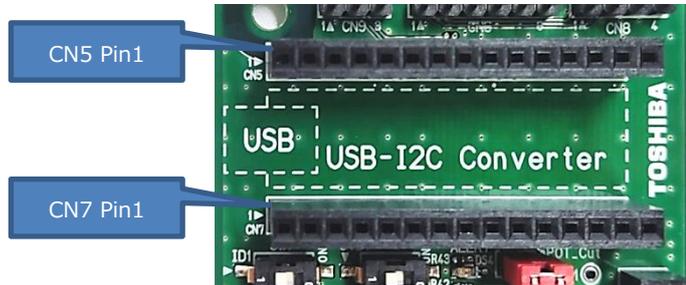
**Fig. 4.3 Board to Board Connectors (CN2 and CN4)**

**Table 4.3 Board to Board Connectors (CN2 and CN4) Specifications  
(same for both connectors)**

Pin	Net Name	Description
1	VM	Motor power supply
2	VLDO	5 V power supply
3	GND	GND
4	GND	GND
5	ID1	MCD I2C address setting signal
6	ID2	MCD I2C address setting signal
7	STBY	MCD standby signal
8	SEL	MCD SEL signal to select speed control
9	CWCCW	MCD CWCCW signal to select motor rotation direction
10	BRAKE	MCD BRAKE signal to apply brake
11	SPD	MCD SPD signal to input motor speed setting
12	SCL	MCD SCL (clock) signal for I2C communication
13	SDA	MCD SDA (data) signal for I2C communication
14	ALERT	MCD ALERT signal to output abnormality detection status of MCD
15	FG	MCD FG signal to output rotation speed of motor
16	CMO	Motor current monitor output signal

### 4.1.1.4. USB-I2C Converter Connectors (CN5 and CN7)

This design uses an USB-I2C Converter (Arduino Nano Every) for connecting the circuit board of this design to the GUI on PC. These connectors CN5 and CN7 are used for connecting the USB-I2C converter to the sub board of this design. 6-534237-3 (TE Connectivity) connectors are used.



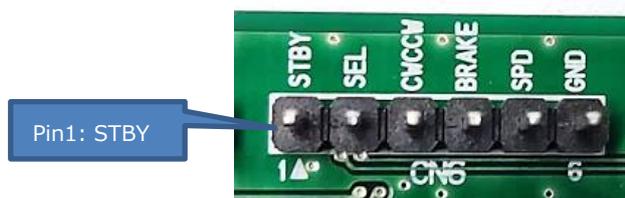
**Fig. 4.4 USB-I2C Converter Connectors (CN5 and CN7)**

**Table 4.4 USB-I2C Converter Connectors (CN5 and CN7) Specifications**

Connector	Pin	Net Name	Description
CN5	1-15	-	Not Connected
CN7	1-7	-	Not Connected
	8	SDA	I2C connection signal
	9	SCL	I2C connection signal
	10-13	-	Not Connected
	14	GND	GND
	15	-	Not Connected

### 4.1.1.5. MCD Control Signal Input Connector (CN6)

This connector is used to input MCD control signals from external MCU. M20-9990646 (Harwin) connector is used.



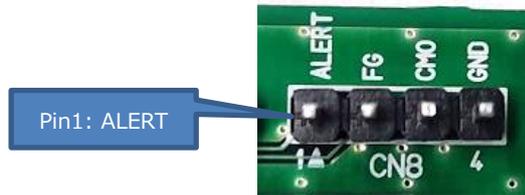
**Fig. 4.5 MCD Control Signal Input Connector (CN6)**

**Table 4.5 MCD Control Signal Input Connector (CN6) Specifications**

Pin	Net Name	Description
1	STBY	MCD standby signal
2	SEL	MCD SEL signal to select speed control method
3	CWCCW	MCD CWCCW signal to select motor rotation direction
4	BRAKE	MCD BRAKE signal to apply brake on motor
5	SPD	MCD SPD signal to input motor speed setting
6	GND	GND

### 4.1.1.6. MCD Status Signal Output Connector (CN8)

This connector is used to output MCD status signals to external MCU. M20-9990446 (Harwin) connector is used.



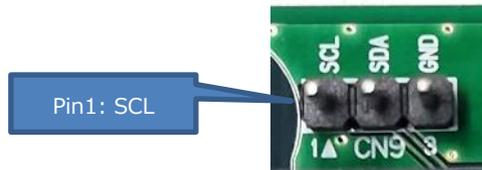
**Fig. 4.6 MCD Status Signal Output Connector (CN8)**

**Table 4.6 MCD Status Signal Output Connector (CN8) Specifications**

Pin	Net Name	Description
1	ALERT	MCD ALERT signal to output abnormality detection status of MCD
2	FG	MCD FG signal to output rotation speed of motor
3	CMO	Motor current monitor output signal
4	GND	GND

### 4.1.1.7. I2C Connector (CN9)

This connector is used for I2C communication from external MCU. M20-9990346 (Harwin) connector is used.



**Fig. 4.7 I2C Connector (CN9)**

**Table 4.7 I2C Connector (CN9) Specifications**

Pin	Net Name	Description
1	SCL	MCD SCL (clock) signal for I2C communication
2	SDA	MCD SDA (data) signal for I2C communication
3	GND	GND

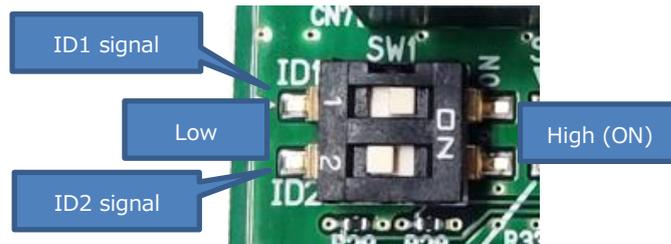
## 4.1.2. Switches

Switches of this design are described in this section.

### 4.1.2.1. I2C Address Switch (SW1)

This switch is used to set I2C address of the MCD. DS04-254-2-02BK-SMT (CUI Devices) switch is used.

I2C address of MCD is set using ID1 and ID2 signals which can be set by this switch. I2C addresses for various configurations of this switch are shown in Table 4.8.



**Fig. 4.8 I2C Address Switch (SW1)**

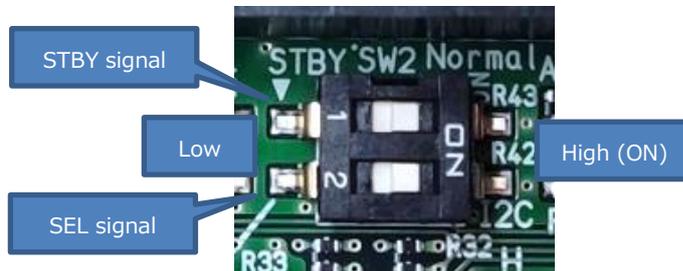
**Table 4.8 I2C Address Switch (SW1) Setting**

ID2	ID1	7-bit I2C Address of MCD	Note
Low	Low	Register 25[7:1] (Initial Value:0x29)	Value in Register 25[7:1]: SLAVE_ADRS becomes the slave address of MCD. (It can be stored in NVM)
Low	High (ON)	0x29	-
High (ON)	Low	0x2D	-
High (ON)	High (ON)	0x32	-

### 4.1.2.2. Standby and Select Switch (SW2)

This switch is used to control two signals of the MCD which are STBY and SEL. DS04-254-2-02BK-SMT (CUI Devices) switch is used.

Table 4.9 shows the function of STBY signal and Table 4.10 shows the function of SEL signal.



**Fig. 4.9 I2C Address Switch (SW1)**

**Table 4.9 STBY Signal (SW2-1) Setting**

STBY	MCD State
Low	Standby Mode
High (ON)	Normal Operation

**Table 4.10 SEL Signal (SW2-2) Setting**

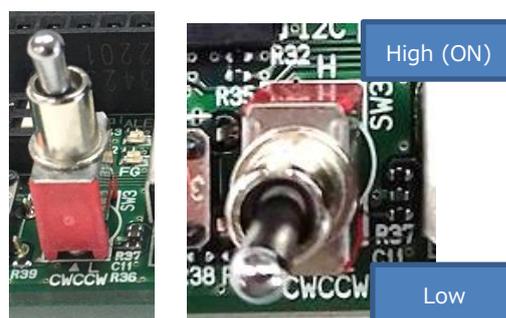
SEL	Motor Speed Control Method
Low	Analog Voltage or PWM speed control (Can be selected using TSPSEL parameter in register)
High (ON)	I2C Speed Control using SPD parameter in register

Note: SPDINV parameter in register can be used to change polarity of speed control.

### 4.1.2.3. Direction Switch (SW3)

This switch is used to change the direction of motor rotation by using CWCCW signal.

2MS1T2B4M2QES (Dailywell) switch is used. Polarity of direction control can be changed by using DIR parameter in register. Description of the direction control is shown in Table 4.11.



**Fig. 4.10 Direction Switch (SW3)**

**Table 4.11 CWCCW Signal (SW3) Setting**

DIR (Parameter in register)	CWCCW (SW3)	Direction
0 (Polarity reversal disabled)	Low	CW
	High	CCW
1 (Polarity reversal enabled)	Low	CCW
	High	CW

### 4.1.2.4. Brake Switch (SW4)

This switch is used to apply brake by using BRAKE signal. B3F-1025 (Omron Electronics) switch is used. Polarity of brake control can be changed by using BRK\_INV parameter in register. Description of the brake control is shown in Table 4.12.



**Fig. 4.11 Brake Switch (SW4)**

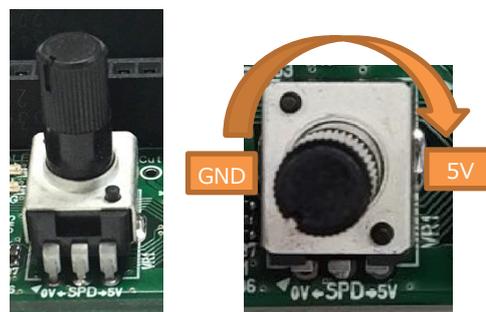
**Table 4.12 BRAKE Signal (SW4) Setting**

BRK_INV (Parameter in register)	BRAKE (SW4)	State
0 (Polarity reversal disabled)	Low	-
	High (Push)	Short brake mode
1 (Polarity reversal enabled)	Low	Short brake mode
	High (Push)	-

### 4.1.3. Potentiometer (VR1)

This potentiometer is used to control the motor speed via analog voltage on SPD signal when analog voltage speed control is selected (SEL signal is Low and TPSEL parameter is 0). PTV09A-4020U-B103 (Bourns) potentiometer is used. Polarity of speed control can be changed by using SPDINV parameter in register.

Rotating it clockwise increases the voltage of SPD signal and vice versa.



**Fig. 4.12 Potentiometer (VR1)**

Note: POT\_Cut jumper can be used to disconnect VR1.

### 4.1.4. LEDs

VM LED (Orange) and VLDO LED (Green) turns on when VM (11 to 27 V) and VLDO (5 V) power supply is available. These LEDs are shown in Fig. 4.13.



**Fig. 4.13 VM LED (DS1, Orange) and VLDO LED (DS2, Green)**

FG LED (Blue) turns on when FG output signal of MCD is Low. FG signal indicates the motor rotation speed. ALERT LED (Red) turns on when ALERT output signal of MCD is Low. Alert signal indicates abnormality detected by the MCD. These LEDs are shown in Fig. 4.14.



**Fig. 4.14 FG LED (DS3, Blue) and ALERT LED (DS4, Red)**

Note: Polarity of ALERT signal can be changed by using ALERTINV parameter in register of MCD.

### 4.1.5. Jumpers

Solder Jumpers on the main board are used to set MCD input signals when only main board is used (without sub board).

These jumpers are shown in Fig. 4.15. Fig. 4.16 shows that in order to short a solder jumper, it's both sides must be joined by using solder. More description of these jumpers is given in Table 4.13.



**Fig. 4.15 Solder Jumper (SJP1 to SJP8)**



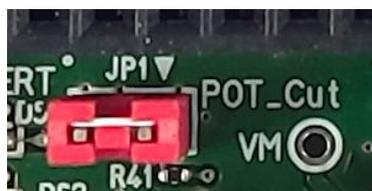
**Fig. 4.16 Solder Jumper Usage Example (Here SJP2 is Open and SJP8 is Short)**

**Table 4.13 Solder Jumper (SJP1 to SJP8) Description**

Jumper	Connects	When Only Main Board is Used
SJP1	STBY and VLDO	Short to pull-up STBY signal to turn on MCD
SJP2	SEL and VLDO	Short to pull-up SEL signal to perform speed control using I2C. (In this case SJP8 must be kept Open)
SJP8	SEL and GND	Short to pull-down SEL signal to perform speed control using analog voltage or PWM via SPD signal. (In this case SJP2 must be kept Open)
SJP3	CWCCW and GND	Short to pull-down CWCCW signal to set motor rotation direction. (DIR parameter in register can be used to change polarity)
SJP4	SPD and Voltage Divider (R44, R5)	Short to apply analog voltage to SPD input signal of MCD. Analog voltage is set using voltage divider circuit (R44, R5). (SPDINV parameter in register can be used to change polarity)
SJP5	ID1 and GND	Short SJP5 and SJP6 to set MCD I2C address to SLAVE_ADRS parameter in register. Default value is 0x29 until it's change via I2C or NVM.
SJP6	ID2 and GND	
SJP7	BRAKE and GND	Short SJP7 to pull-down BRAKE signal. (BRK_INV parameter in register can be used to change polarity)

Note: Do not short SJP2 and SJP8 at the same time as it will short the VLDO power supply to the GND.

POT\_Cut jumper (JP1) shown in Fig. 4.17 is used to disconnect the VR1 potentiometer from the SPD input signal of MCD.



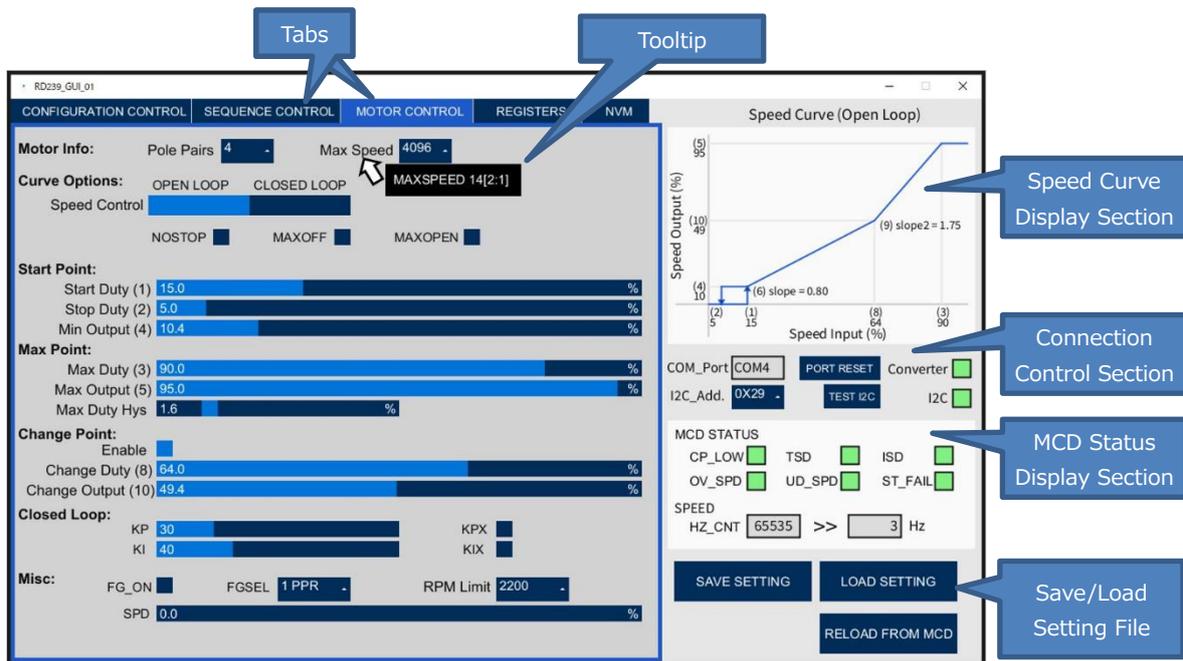
**Fig. 4.17 POT\_Cut Jumper (JP1)**

## 4.2. GUI Description

Overview of the GUI of this design is shown in Fig. 4.18. It is used for configuring and setting MCD register data. It also shows the status of MCD and motor rotation speed.

When GUI starts, it reads all the register setting of the connected MCD, and then sets all the controls on GUI according to the read data. And it also keeps polling the MCD to read MCD status and motor speed status parameter HZ\_CNT every 1 second, and then updates the MCD status section. Whenever any controller on GUI is modified, the corresponding setting according to the new value is also written in the registers of MCD. All current GUI settings can be saved/loaded to/from a file on PC using the SAVE and LOAD SETTING buttons.

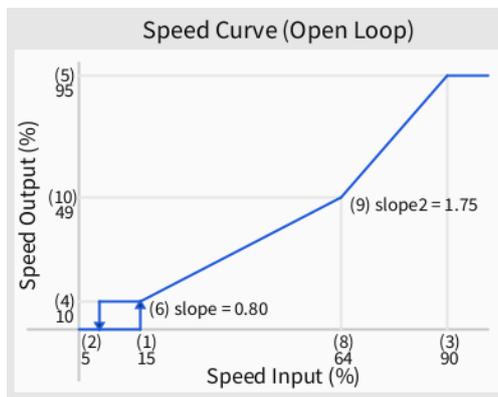
Refer to section 6.1 for procedure of setting up the GUI.



**Fig. 4.18 GUI Overview**

### 4.2.1. Speed Curve Display Section

This section displays the speed curve according to current setting of parameters in the GUI.



**Fig. 4.19 Speed Curve Display Section**

### 4.2.2. Connection Control Section

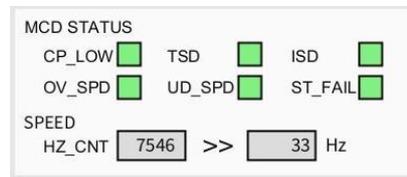
This section shows the status of USB-I2C converter and I2C connection. It is also used to reset COM port and to change MCD I2C address.



**Fig. 4.20 Connection Control Section**

### 4.2.3. MCD Status Display Section

This section shows the status of MCD and the motor speed by reading the MCD register data. This data is updated every 1 second.



**Fig. 4.21 MCD Status Display Section**

### 4.2.4. Save and Load Setting File Buttons

Save setting button allows the user to save the all the register settings to a file. Settings are saved in a c language structure format which can be easily used in any microcontroller application.

Load setting button allows the user to read and load a setting file.

Reload from MCD button reads all the registers from MCD and then updates the whole GUI.

Register values can be seen in Registers tab of GUI. Since the GUI always have same setting as the MCD, this button is just for confirmation.



**Fig. 4.22 Save and Load Setting File Buttons**

### 4.2.5. Tooltip

Tooltip appears when mouse is stopped over any setting name in GUI. Tooltip can be seen as the text in black box in the following image. Tooltip shows the name of the parameter in MCD register affected by this setting, it also shows the register number (in decimal) of parameter and its MSB bit (start bit in register) and LSB bit (end bit in register).

As shown in the example in following image, Max Speed setting will change MAXSPEED parameter which is in the register 14 and its MSB (start bit in register) is 2 and LSB (end bit in register) is 1, therefore it is a 2-bit parameter. For more description on any parameter, refer to TC78B011FTG datasheet.



Fig. 4.23 Tooltip

### 4.2.6. Tabs

Tabs are used to classify various settings of MCD.

#### 4.2.6.1. Configuration Control Tab

Configuration Control Tab is used to configure the motor drive parameters and protection control parameters. Refer to TC78B011FTG datasheet for more description about various parameters.

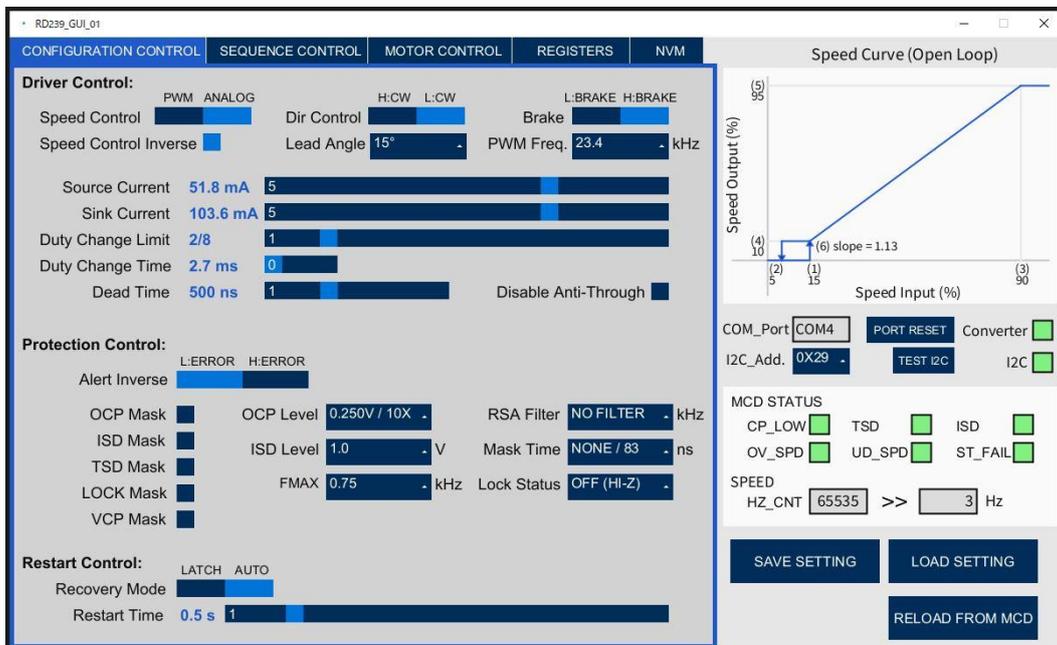


Fig. 4.24 Configuration Control Tab

### 4.2.6.2. Sequence Control Tab

Sequence Control Tab is used to configure the parameters which affect start and brake sequence. Refer to TC78B011FTG datasheet for more description about various parameters.

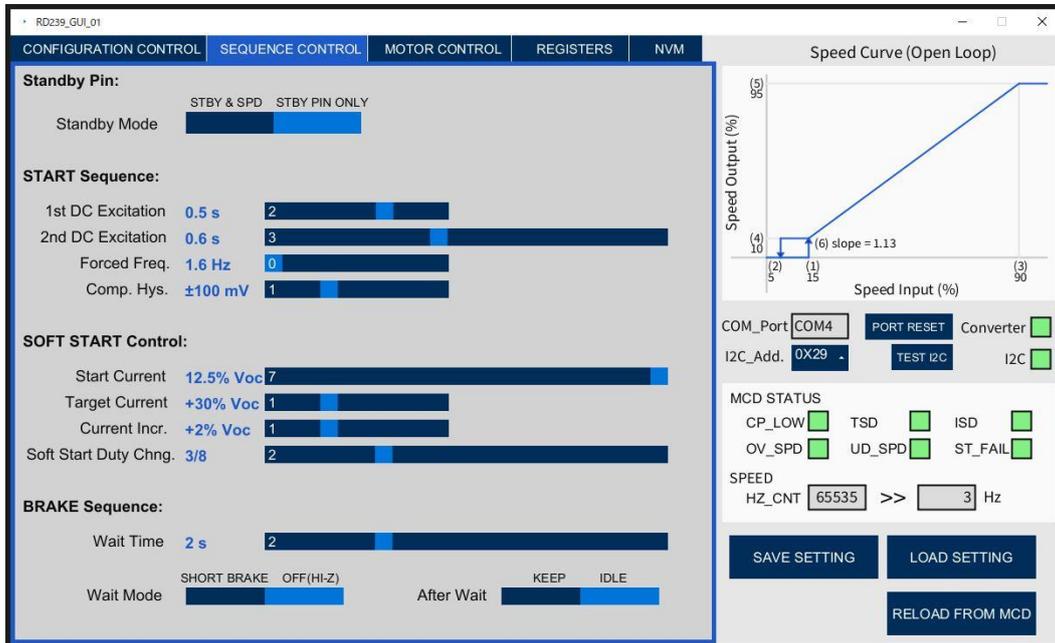


Fig. 4.25 Sequence Control Tab

### 4.2.6.3. Motor Control Tab

Motor Control Tab is used to configure parameters which affect the speed curve used to control motor during operation. Refer to TC78B011FTG datasheet for more description about various parameters.

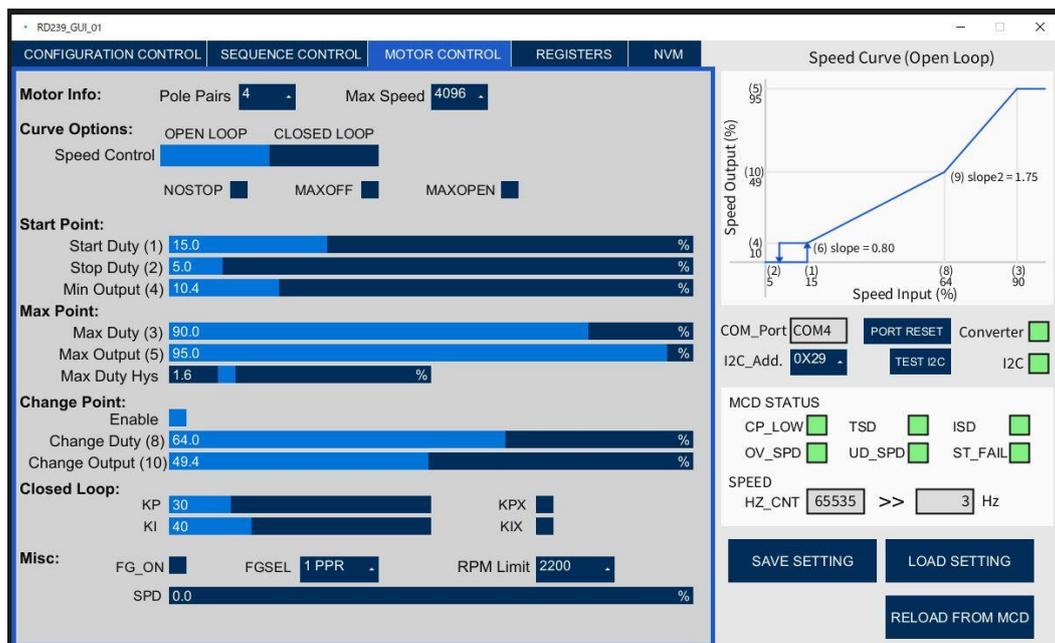


Fig. 4.26 Motor Control Tab

### 4.2.6.4. Registers Tab

Register Tab shows the status of MCD registers and parameters. It can also be used to read or write a single register of the MCD. Refer to TC78B011FTG datasheet for more description about various parameters.

Register addresses in this tab are in decimal number format (same as in datasheet of MCD). More description of data format of Register Data and Parameter data is available in Fig. 4.28 and Fig. 4.29 respectively.

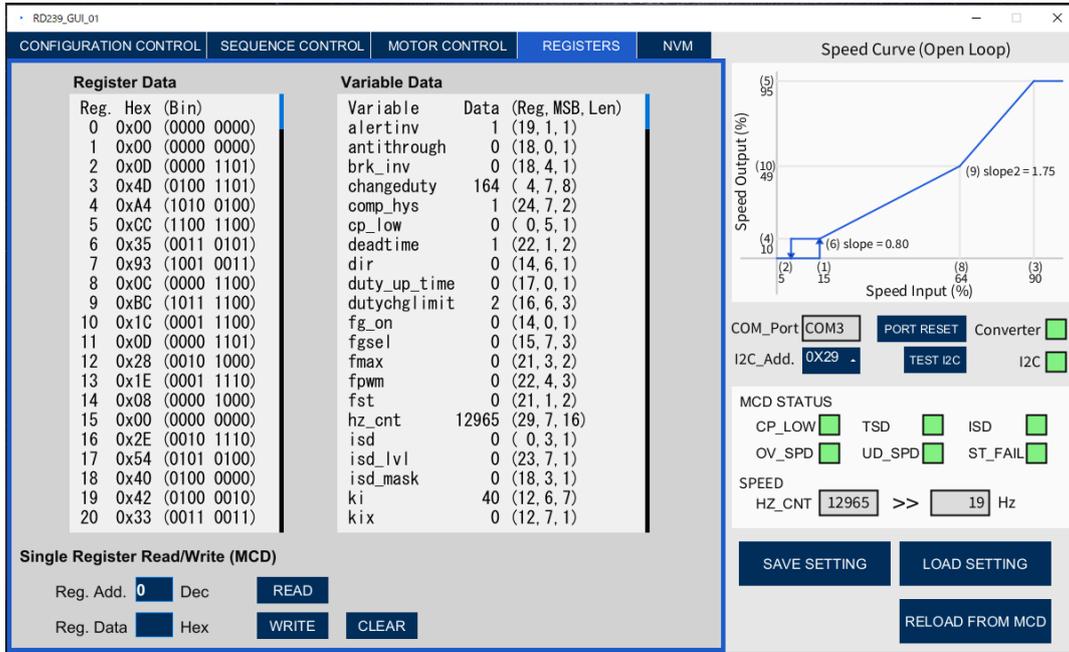


Fig. 4.27 Registers Tab

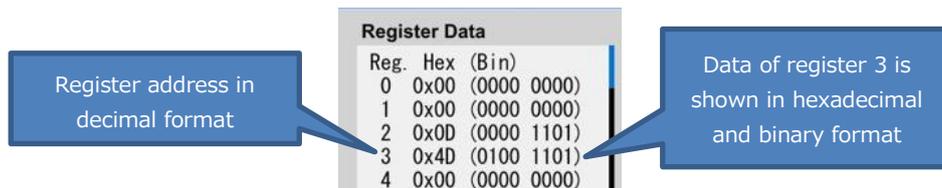


Fig. 4.28 Description of Register Data Format

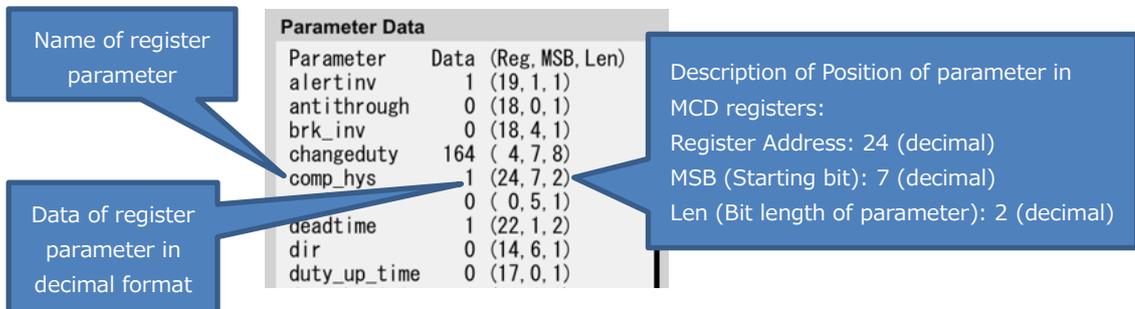
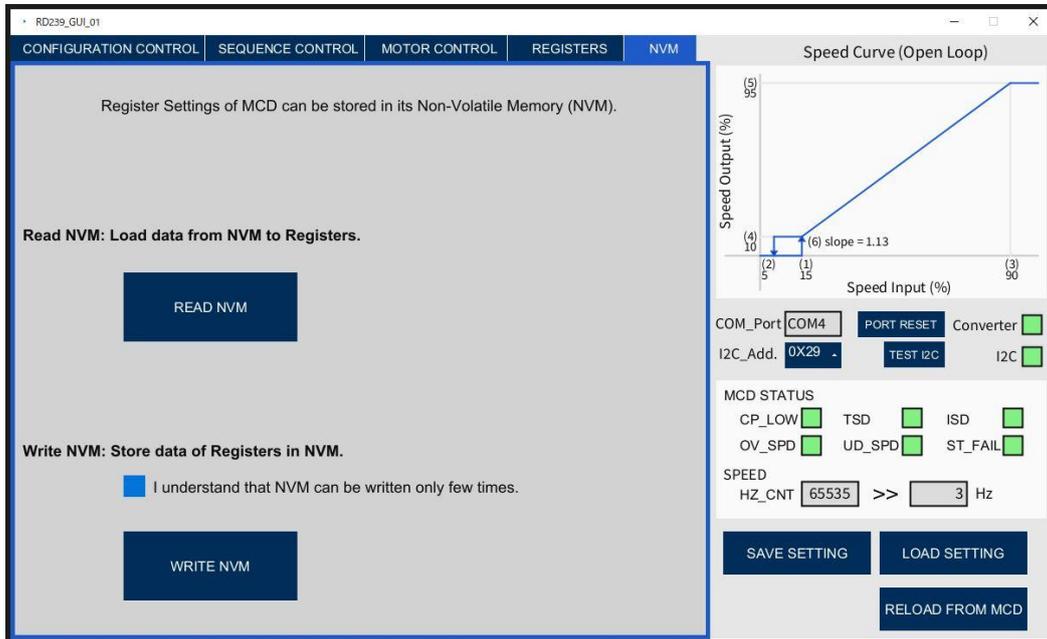


Fig. 4.29 Description of Parameter Data Format

### 4.2.6.5. NVM Tab

NVM Tab is used to read and write the non-volatile memory (NVM) of the MCD. Refer to TC78B011FTG datasheet for more description of NVM.



**Fig. 4.30 NVM Tab**

Note: NVM can be written only few times (minimum value of program/erase cycle limit is 10). And VM of at least 10.8 V is required for writing NVM.

### 4.3. USB-I2C Converter Description

USB-I2C Converter is used to convert the data received from PC via USB to data to be sent to MCD via I2C and vice versa.

In this design, the “Arduino Nano Every” microcontroller (MCU) board is used as a USB-I2C Converter. Refer to section 6.2 for procedure of setting up the USB-I2C converter. The board is required to be programmed only once because the program is stored in the MCU on “Arduino Nano Every” board.

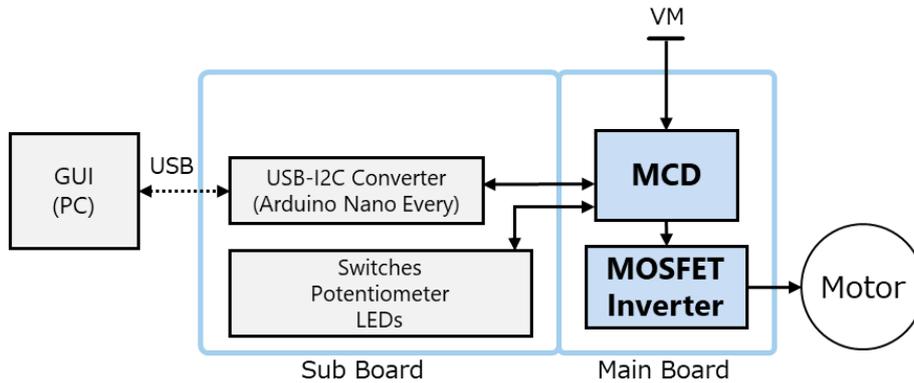


**Fig. 4.31 USB-I2C Converter**

## 5. Operation

### 5.1. Mode 1: GUI Control Operation

In this mode, this design (including main and sub board) is connected to VM (external power supply), brushless motor and PC GUI (via USB-I2C converter). The motor can be controlled by GUI or switches and potentiometer of the sub board. Motor speed control can be performed in GUI by using SPD register via I2C or with analog voltage speed control (VR1) present on the sub board. Registers of the MCD can be programmed by using GUI or can be loaded automatically from NVM of MCD at startup.



**Fig. 5.1 Mode 1: GUI Control Operation**

### 5.1.1. Starting Procedure

This section describes the procedure for starting this design in Mode 1.

Board Configuration:

- Open all the solder jumpers on main board (SJP1 to SJP8 on main board)
- Select I2C address of MCD (SW1 on sub board) (Default value: 0x29 set using ID1:High(On) and ID2:Low)
- Set STBY to High(On) (SW2-1 on sub board)
- Set SEL to select intended mode of motor speed control (SW2-2 on sub board)
- Short POT\_Cut jumper to enable VR1 Potentiometer (JP1 on sub board)

Connection:

- Connect main and sub board (CN2 and CN4)
- Connect brushless motor to main board (CN3)
- Connect USB-I2C converter (Arduino Nano Every) to sub board (CN5 and CN7)
- Connect PC to USB-I2C converter (using USB cable)
- Connect VM power supply to main board (CN1)

Startup:

- "Arduino Nano Every" must be programmed with provided code for USB-I2C Converter (refer to section 6.2)  
(This step is required only one time, as the program is stored in MCU of "Arduino Nano Every".)
- Turn On VM power supply
- Start GUI (refer to section 6.1)
- Set MCD register parameters from GUI (or load a setting file using LOAD SETTING button) (MCD register data can also be loaded automatically from NVM of MCD, refer to section 5.5 for more details)
- With correct setting the motor should start running (refer to TC78B011FTG datasheet for more details)
- Motor (speed, direction, etc.) can be controlled using switches and potentiometer of sub board or from GUI

Note: In case of any problems refer to section 7 on Troubleshooting.

### 5.1.2. Stopping Procedure

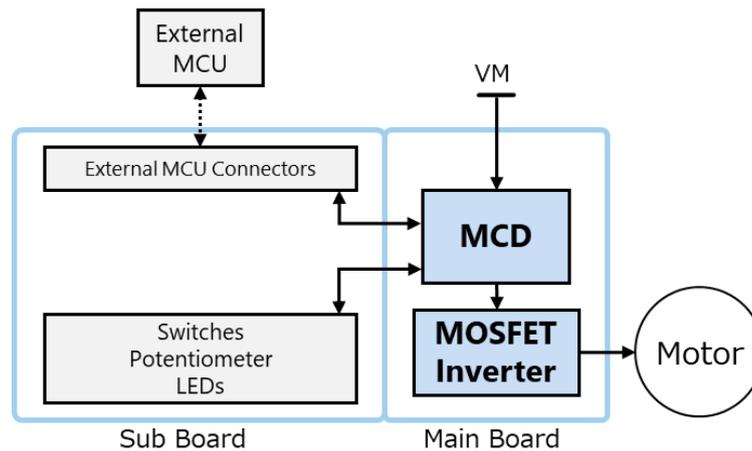
This section describes the procedure for stopping this design.

- Close GUI
- Turn off VM power supply

## 5.2. Mode 2: External MCU Control Operation

In this mode, this design (including main and sub board) is connected to VM (external power supply), brushless motor and an external MCU (via connectors C6 and C9). The motor can be controlled by external MCU or switches and potentiometer of the sub board. Motor speed control can be performed by external MCU or with analog voltage speed control (VR1) present on the sub board.

Registers of the MCD can be programmed by external MCU via I2C using connector C9 or can be loaded automatically from NVM of MCD at startup.



**Fig. 5.2 Mode 2: External MCU Control Operation**

### 5.2.1. Starting Procedure

This section describes the procedure for starting this design in Mode 2.

Board Configuration:

- Open all the solder jumpers on main board (SJP1 to SJP8 on main board)
- Select I2C address of MCD (SW1 on sub board) (Default value: 0x29 set using ID1:High(On) and ID2:Low)
- Set STBY to High(On) (SW2-1 on sub board)
- Set SEL to select intended mode of motor speed control (SW2-2 on sub board)
- If VR1 is not used, open POT\_Cut jumper (JP1 on sub board).

Connection:

- Connect main and sub board (CN2 and CN4)
- Connect brushless motor to main board (CN3)
- Connect External MCU to sub board (CN6, CN8 and CN9)
- Connect VM power supply to main board (CN1)

Startup:

- Turn On VM power supply
- Start external MCU
- Set MCD parameters using I2C (from external MCU)  
(MCD register data can also be loaded automatically from NVM of MCD, refer to section 5.5 for more details)
- With correct setting the motor should start running (refer to TC78B011FTG datasheet for more details)
- Motor (speed, direction, etc.) can be controlled using switches and potentiometer of sub board or from external MCU using I2C

Note: In case of any problems refer to section 7 on Troubleshooting.

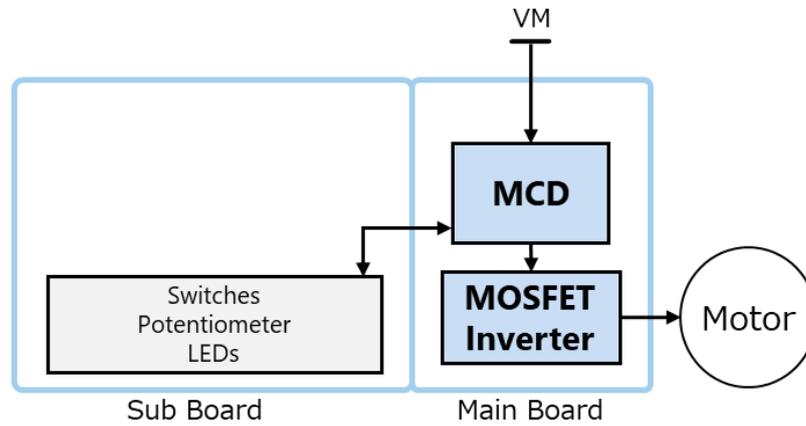
### 5.2.2. Stopping Procedure

This section describes the procedure for stopping this design.

- Stop external MCU (optional)
- Turn off VM power supply

### 5.3. Mode 3: Manual Operation

In this mode, this design (including main and sub board) is connected to VM (external power supply) and brushless motor. The motor can be controlled by switches and potentiometer of the sub board. Motor speed control can be performed with analog voltage speed control (VR1) present on the sub board. This mode requires register setting to be present in NVM. This allows registers of the MCD to be loaded automatically from NVM at startup.



**Fig. 5.3 Mode 3: Manual Operation**

### **5.3.1. Starting Procedure**

This section describes the procedure for starting this design in Mode 3.

Board Configuration:

- Open all the solder jumpers on main board (SJP1 to SJP8 on main board)
- Set STBY to High(On) (SW2-1 on sub board)
- Set SEL to select intended mode of motor speed control (SW2-2 on sub board)
- Short POT\_Cut jumper to enable VR1 Potentiometer (JP1 on sub board)

Connection:

- NVM of MCD on the main board must be preloaded with desired setting (refer to section 5.5)
- Connect main and sub board (CN2 and CN4)
- Connect brushless motor to main board (CN3)
- Connect VM power supply to main board (CN1)

Startup:

- Turn On VM power supply (MCD register settings are automatically loaded from NVM at startup)
- With correct setting the motor should start running (refer to TC78B011FTG datasheet for more details)
- Motor (speed, direction, etc.) can be controlled using switches and potentiometer of sub board

Note: In case of any problems refer to section 7 on Troubleshooting.

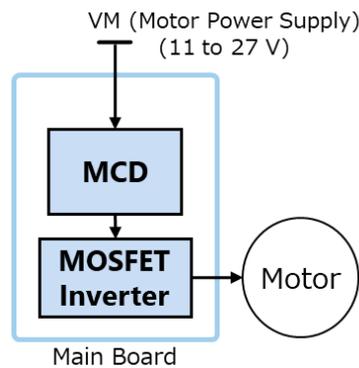
### **5.3.2. Stopping Procedure**

This section describes the procedure for stopping this design.

- Turn off VM power supply

#### 5.4. Mode 4: Minimum Hardware Operation with Main Board Only

This mode shows that by using this design a brushless motor can be easily run using very less and simple hardware. In this mode, only main board is used and is connected to VM (external power supply) and brushless motor. Motor control signals are set using the solder jumpers SJP1 to SJP8 on main board. And motor speed is set using the analog voltage produced by voltage divider consisting of R44 and R5 on the main board. This mode requires register setting to be present in NVM. This allows registers of the MCD to be loaded automatically from NVM at startup.



**Fig. 5.4 Mode 4: Minimum Hardware Operation with Main Board Only**

### 5.4.1. Starting Procedure

This section describes the procedure for starting this design in Mode 4.

Board Configuration: The solder jumpers on main board must be configured as described below:

- Short SJP1 (Sets STBY to High)
- Short SJP8 and Open SJP2 (Sets SEL to Low for analog voltage speed control)  
(Don't short both SJP8 and SJP2 at the same time as it will short the power supply VLDO)
- Short SJP3 (Sets CWCCW to Low)  
(DIR parameter in register can be used to change polarity)
- Short SJP4 (Connects SPD to analog voltage output of voltage divider (R44 and R5) for speed setting)  
(Speed setting can be changed using these resistors (R44 and R5) of voltage divider)  
(SPDINV parameter in register can be used to change polarity)
- Short SJP5 and SJP6 (Sets ID1 and ID2 to Low)
- Short SJP7 (Sets BRAKE to Low)  
(BRK\_INV parameter in register can be used to change polarity)

Connection:

- NVM of MCD on the main board must be preloaded with desired setting (refer to section 5.5)
- Connect brushless motor to main board (via CN3)
- Connect VM power supply to main board (via CN1)

Startup:

- Turn On VM power supply (MCD register settings are automatically loaded from NVM at startup)
- With correct setting the motor should start running (refer to TC78B011FTG datasheet for more details)
- Motor speed can be controlled by changing voltage divider resistor (R44 and R5) values. Motor rotation direction can be changed by changing direction polarity parameter (DIR) in register of TC78B011FTG.

### 5.4.2. Stopping Procedure

This section describes the procedure for stopping this design.

- Turn off VM power supply

## 5.5. Writing NVM of MCD

Non-Volatile Memory (NVM) of MCD can store the register data of MCD even after the power supply is turned off. Register Data from NVM is loaded into MCD when it is turned on. NVM of TC78B011FTG can only be written a few times (minimum value of program/erase cycle limit is 10). At least 10.8 V VM is required for writing NVM.

Following steps must be followed to write NVM.

- VM must be at least 10.8 V to write NVM.
- Start up this design in mode 1 (refer to section 5.1)
- Perform the desired setting of MCD
- Go to NVM tab of the GUI
- Check the Box with text "I understand that NVM can be written only few times."
- Use Write NVM button to write current MCD register data to NVM.

## 6. Setting Up GUI and USB-I2C Converter

### 6.1. GUI Setup

The code for GUI provided with this design works on “Processing” application. It can be downloaded from internet.

Steps for starting GUI of this design are as follows:

- Download processing application from internet (Processing version 4.2 was used to develop GUI)
- Open processing.exe file
- Install “ControlP5” library (ControlP5 Version 2.2.6 was used to develop GUI)
  - Go to “Sketch” in menu bar and select “Import Library” and then select “Manage Libraries”
  - In this new dialog box under the “Libraries” tab type “ControlP5”
  - Select “ControlP5” library and click on “Install” button
  - After intallation is complete, close this dialog box
- In processing application, Go to “File” in menu bar and select “Open”
- Then browse to the GUI code provided with this design (refer to section 3.4)
- Then select and open the GUI code file
- Click button to start

### 6.2. USB-I2C Converter Setup

USB-I2C Converter uses “Arduino Nano Every” board, and its software was written using Arduino Windows Application (1.8.19). The board is required to be programmed only once because the program is stored in the MCU on “Arduino Nano Every” board.

Steps for writing software of USB-I2C Converter in “Arduino Nano Every” are as follow.

- Download arduino application from internet
- Add “Arduino Nano Every” support i.e. “Arduino megaAVR Core”
  - Got to “Tools” in menu bar, select “Board” and then select “Board Manager”
  - Search for “megaAVR” and install “Arduino megaAVR”
- Connect “Arduino Nano Every” board to the PC via USB cable
- Go to “Tools” in menu bar, select “Board” and then select “Arduino Nano Every”
- Go to “Tools” in menu bar, select “Port” and then select correct port of “Arduino Nano Every” board
- Go to “File” in menu bar and select “Open”
- Then browse and open the USB-I2C converter code provided with this design (refer to section 3.5)
- Then compile and upload the code to “Arduino Nano Every” board

## 7. Troubleshooting

### 7.1. I2C Connection Error in GUI

Possible solutions:

- VM and VLDO must have power. Check LEDs DS1 and DS2.
- MCD STBY signal must be High. Can be set using SW2 switch.
- Main board and sub board must be connected by using CN2 and CN4.
- Check I2C address setting by checking ID2 and ID2 signal input of MCD.
- I2C address used by the master should be same as I2C address of MCD set via ID1 and ID2.
- USB-I2C controller must be properly connected to the sub board.
- Arduino Nano Every must be programmed to work as USB-I2C Converter (refer to section 6.2)

### 7.2. Speed Control Not Working

Check following as speed control method selection is affected by following:

- SEL input pin of MCD
- TSPSEL parameter in register 15 of MCD. Can be set using “Speed Control” in configuration control tab of GUI.
- SPDINV parameter in register 15 of MCD. Can be set using “Speed Control Inverse” in configuration control tab of GUI.

### 7.3. Sub Board Switches and Potentiometer Not Working

Possible solutions:

- Main board and sub board must be connected by using CN2 and CN4.
- All solder jumpers (SJP1 to SJP8) must be open when using sub board.
- POT\_Cut jumper must be short.
- In this design, the signals from switches and potentiometer of sub board are overwritten by signals from external MCU connector (CN6), therefore CN6 must be disconnected.

## 8. Precautions for Evaluation (To Prevent Electric Shock, Burn Injury, etc.)

Pay special attention to the following when using this design:

- Make sure that the polarities of all connections are correct before supplying electricity.
- Make sure that the capacitor is sufficiently discharged before touching the BOARD.
- When checking the operation, cover the BOARD with an acrylic case for safety.
- MOSFET and other components may generate heat during operation. Be careful not to get burned when handling them.

## Terms of Use

This terms of use is made between Toshiba Electronic Devices and Storage Corporation ("We") and Customer who downloads or uses this Reference Design. Customer shall comply with this terms of use. This Reference Design means all documents and data in order to design electronics applications on which our semiconductor device is embedded.

### **Section 1. Restrictions on usage**

1. This Reference Design is provided solely as reference data for designing electronics applications. Customer shall not use this Reference Design for any other purpose, including without limitation, verification of reliability.
2. Customer shall not use this Reference Design for sale, lease or other transfer.
3. Customer shall not use this Reference Design for evaluation in high or low temperature, high humidity, or high electromagnetic environments.
4. This Reference Design shall not be used for or incorporated into any product or system whose manufacture, use, or sale is prohibited under any applicable laws or regulations.

### **Section 2. Limitations**

1. We reserve the right to make changes to this Reference Design without notice.
2. This Reference Design should be treated as a reference only. WE ARE NOT RESPONSIBLE FOR ANY INCORRECT OR INCOMPLETE DATA AND INFORMATION.
3. Semiconductor devices can malfunction or fail. When designing electronics applications by referring to this Reference Design, Customer is 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 semiconductor devices could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Customer must also refer to and comply with the latest versions of all relevant our information, including without limitation, specifications, data sheets and application notes for semiconductor devices, as well as the precautions and conditions set forth in the "Semiconductor Reliability Handbook".
4. Designing electronics applications by referring to this Reference Design, Customer must evaluate the whole system sufficiently. Customer is solely responsible for applying this Reference Design to Customer's own product design or applications. WE ASSUME NO LIABILITY FOR CUSTOMER'S PRODUCT DESIGN OR APPLICATIONS.
5. WE SHALL NOT BE RESPONSIBLE FOR ANY INFRINGEMENT OF PATENTS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS OF THIRD PARTIES THAT MAY RESULT FROM THE USE OF THIS REFERENCE DESIGN. NO LICENSE TO ANY INTELLECTUAL PROPERTY RIGHT IS GRANTED BY THIS TERMS OF USE, WHETHER EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE.
6. THIS REFERENCE DESIGN IS PROVIDED "AS IS". WE (a) ASSUME 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 (b) DISCLAIM ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO THIS REFERENCE DESIGN, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF FUNCTION AND WORKING, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.

### **Section 3. Terms and Termination**

It is assumed that Customer agrees to any and all this terms of use if Customer downloads or uses this Reference Design. We may, at its sole and exclusive discretion, change, alter, modify, add, and/or remove any part of this terms of use at any time without any prior notice. We may terminate this terms of use at any time and without any cause. Upon termination of this terms of use, Customer shall eliminate this Reference Design. Furthermore, upon our request, Customer shall submit to us a written confirmation to prove elimination of this Reference Design.

### **Section 4. Export Control**

Customer shall not use or otherwise make available this Reference Design 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). This Reference Design may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Act and the U.S. Export Administration Regulations. Export and re-export of this Reference Design is strictly prohibited except in compliance with all applicable export laws and regulations.

### **Section 5. Governing Laws**

This terms of use shall be governed and construed by laws of Japan, without reference to conflict of law principle.

### **Section 6. Jurisdiction**

Unless otherwise specified, Tokyo District Court in Tokyo, Japan shall be exclusively the court of first jurisdiction for all disputes under this terms of use.