

LV8714TASLDGEVK

Dual Stepper Motor Driver Module Solution Kit Quick Start Guide

Overview

The LV8714TASLDGEVB is an **ON semiconductor** motor driver module featuring the **LV8714TA**. This module is capable of easily driving **dual stepper motors**.

Motor driving is made easy with Arduino Micro¹⁾ compatibility. The LV8714TASLDGEVB comes with a Baseboard for facilitated plug-and-play connectivity with an Arduino Micro.

GUI and Open source API functions give strong contribution to your easy and immediate motor driving experiences.

Features

- $V_{CCmax} = 18\text{ V}$, $I_{Omax} = 1.5\text{ A}^2$
(OUT_A-B, OUT_C-D, OUT_E-F, OUT_G-H, Respectively)
- **6 – 12 V Applications** Recommended ($V_{CC} = 4 – 16.5\text{ V}$ Recommended)
- Stand-by Current Consumption = 0
- In Addition to the Full Step (2-phase Excitation) and the Half Step (1-2-phase) Excitation, the External VREF Input Can be Used to Achieve 1/256 Excitation Method, (Higher than the Maximum Resolution of our Conventional Product 1/128 Step)
- User-friendly **GUI Allows Operation Check and Sample Program Generation**

For further product information, please visit:

<http://www.onsemi.com/PowerSolutions/product.do?id=LV8714TA>



ON Semiconductor®

www.onsemi.com

EVAL BOARD USER'S MANUAL



Figure 1. LV8714TASLDGEVB Board



Figure 2. ONBB4AMGEVB Board with
Arduino Micro and LV8714TASLDGEVB
Board

¹⁾ Arduino/Genuino are trademarks registered by Arduino AG.

²⁾ Stress exceeding this voltage and current may damage the device. If these values are exceeded, the functionality of the device may be damaged or the reliability may be affected.

Also, the ICs may generate heat even if the current is less than or equal to the I_{Omax} depending on the operating condition, and the IC may stop operating.

LV8714TASLDGEVK

WHAT YOU NEED

- PC
Recommended OS: Windows 7/ Windows10, 64 bit
Connected to the Internet
(Only for "Including TimerThree Libraries" on page 3)
ArduinoIDE 1. 8.4³⁾ installed
Users with administrator rights
- Power supply
AC adapter Output voltage: 4 – 16.5 V
 Output current: ~6 A
 Connector specification:
 Polarization: Positive Center
 Inner diameter = 2.1 mm, Outer diameter = 5.5 mm
 Recommended: WSU120–1000 (TRIAD Magnetics)
 or DC current stabilizing power supply, batteries, etc.

CONTENTS OF THE KIT

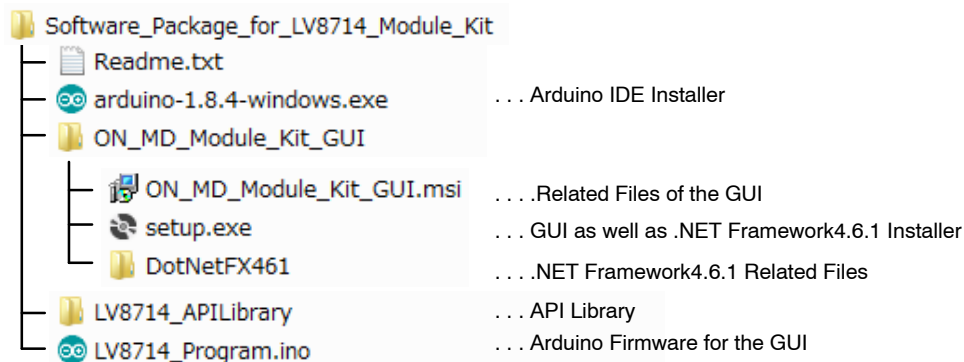
Hardware

- LV8714TASLDGEVB: motor driver module
- ONBB4AMGEVB: baseboard
- Arduino Micro
- USB–Cable (M Micro B–A)
- Flat–tip screw driver
- Stepper motor MDP–35A
(NIDEC SEIMITSU CORPORATION,
step angle = 7.5°, 12 V / 300 mA) x 1

Software

All of the following files are included in LV8714TASLDGEVK Software (English) that is available on the website of the following links.

<http://www.onsemi.com/PowerSolutions/evalBoard.do?id=LV8714TASLDGEVK>

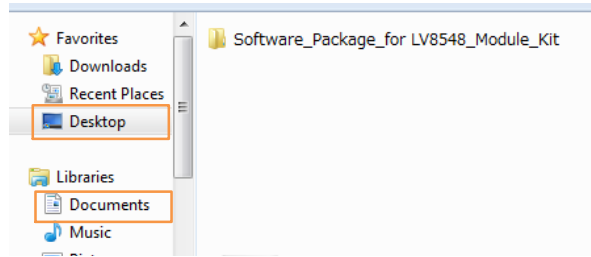


³⁾ Different versions may not work properly. If it is not installed, refer to the Appendix (Supplement).

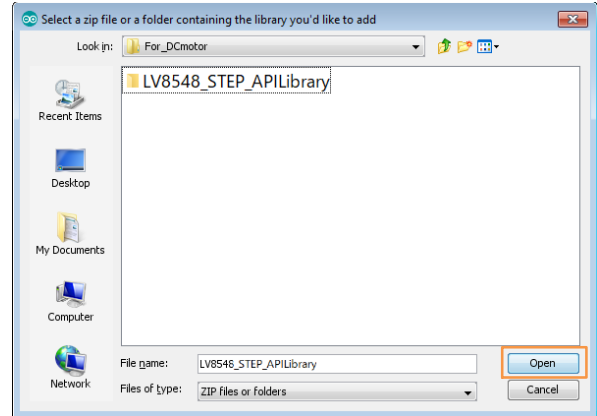
LV8714TASLDGEVK

Software Setup

Please download LV8714TASLDGEVK_SOFTWARE.ZIP and expand **Software_Package_for_LV8714_Module_Kit** to any directories (E.g., desktop or Library\Document).




The following window will appear. Select the "LV8714_APILibrary" folder and open this folder as shown below. Click the folder only once and push "Open" button. Do not double click the folder)



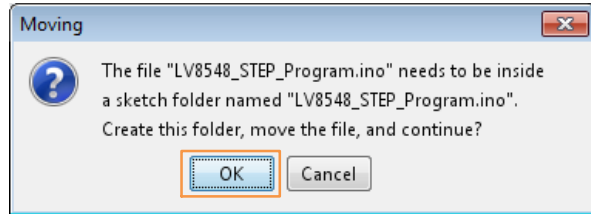
API Library

The following process should be operated with the Arduino Micro disconnected.

1. Double-click on the Arduino sketch for GUI

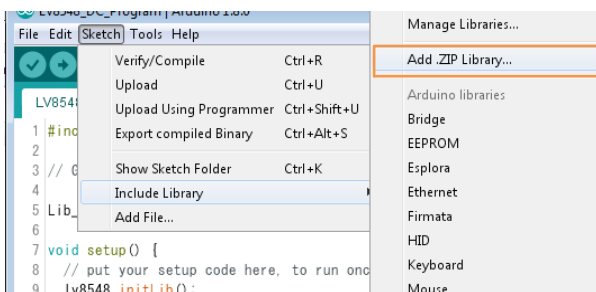
 LV8714_Program.ino

This will launch the ArduinoIDE. (Check the [Appendix. "Arduino IDE Installation"](#), if the Arduino IDE is not installed) The IDE will display the following message when the LV8714_Program.ino file is opened for the first time. Click "OK" to continue.



This message will be displayed if the .ino file is not inside of a folder with the same name. (E.g Sketch12345.ino must be in the Sketch12345 folder)

2. Include the API Library As shown below, navigate to "Sketch → Include Library → Add .ZIP library..."

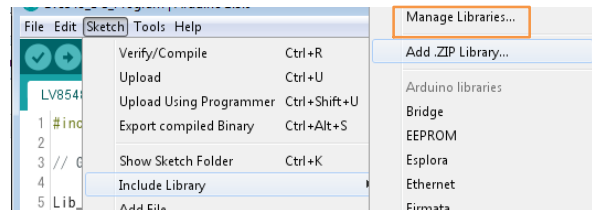


Unless updating the function library, including the API function library only needs to be done once.

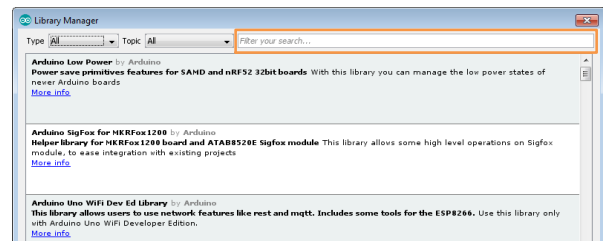
Including the TimerThree Library

(Re-inclusion is not necessary if it has already been included.)

As shown below, navigate to "Sketch → Include Library → Manage Libraries..."



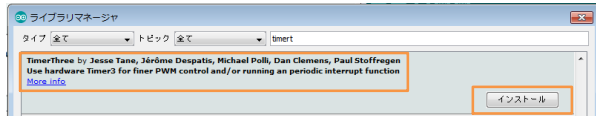
Once the Library Manager starts up, please type "TimerThree" in the search bar at the top.



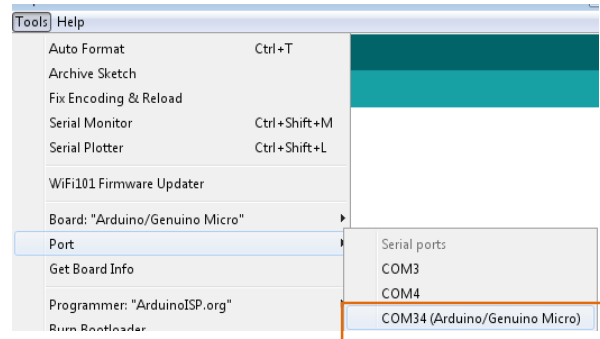
*Some screen captures are taken from different motor driver modules

LV8714TASLDGEVK

Select and install "TimerThree" in the search results.



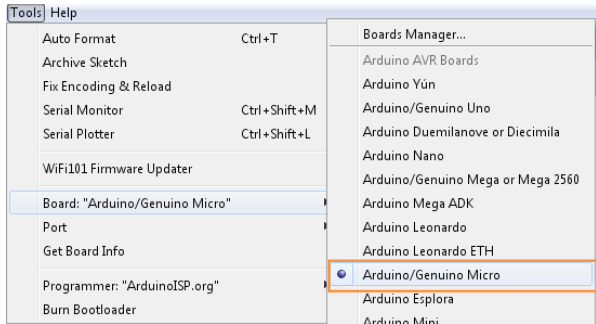
3. Connect the PC to the Arduino Micro via USB and select the corresponding COM port as shown below.




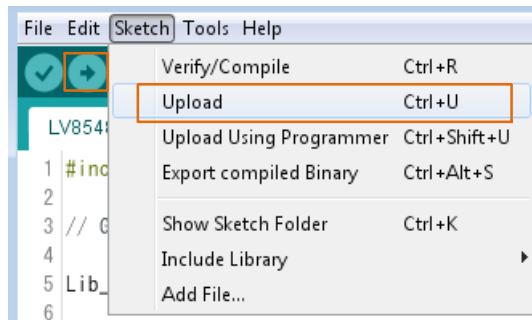
Compiling the Arduino Program

• Write to Arduino

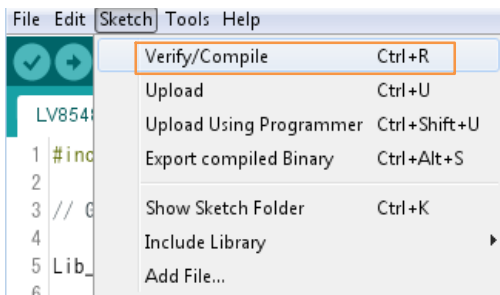
1. Select the Arduino board to upload to by navigating to "Tools → Board → Arduino/Genuino Micro".



4. Upload the sketch by clicking "Sketch → Upload" or by pressing the  button.



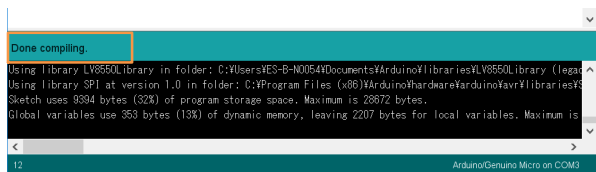
2. Navigate to "Sketch → Verify/Compile" when finished writing



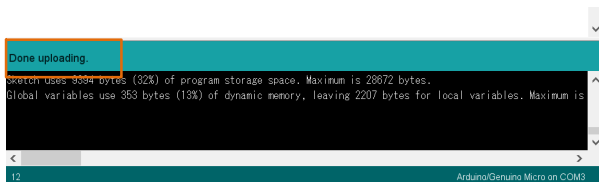
In the process of uploading, the Arduino Micro bootloader will be installed.



The IDE will display "Done Compiling" after a successful compile has been verified.



The IDE will display "Done uploading" after a successful upload to the Arduino.



Sketches written to the Arduino will not be erased unless rewritten.

If the uploading of the program fails, confirm the details in step 1 check board name and step 3 check serial port connection.

*Some screen captures are taken from different motor driver modules



Rewrite the Arduino Program

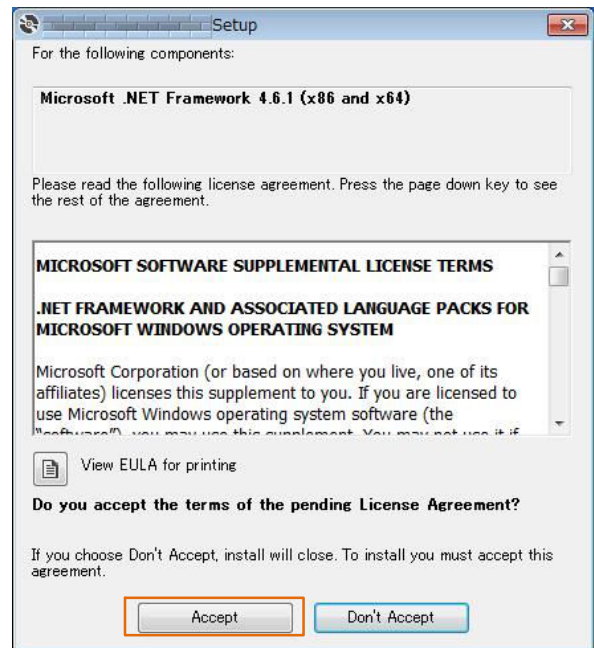
In the case of 1 or 2 below, please rewrite the program to the Arduino.

1. When updating the API function library.
 - a) Delete the Existing API Function Library by navigating to MyDocument\Arduino\libraries and deleting the “LV8714_APILibrary” folder.
 - b) Include the latest API function library. Please save the latest API function library in your preferred directory on your PC.
(See Page 3, “[Software Setup](#)”)
Include the latest API function library.
(See Page 3, “[API Library](#)”)
 - c) Compile and write the program to the Arduino
(See Page 4, “[Compiling the Arduino program](#)”
• [Write to Arduino](#)”)
2. When evaluating other motor driver modules (Assuming other motor driver module libraries have already been included)
 - a) Compile and write the program to the Arduino
(See Page 4, “[Compiling the Arduino program](#)”
• [Write to Arduino](#)”)

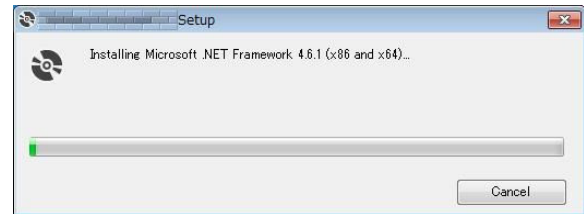
If you are testing the any other motor module library for the first time, please operate according to their corresponding manual.


Installing the GUI

1. Double-click on  **setup.exe** in the  **ON MD Module Kit GUI** folder and proceed with the installation as shown below.
* If a previous version of this GUI (ON_MD_Module_Kit_GUI) has already been installed and the most recent GUI will be installed, please overwrite and update the existing GUI.
(No need for uninstallation)
2. The .NET Framework4.6.1 installation window will appear if the PC does not have the necessary version. If .NET Framework4.6.1 is not installed, click Accept to install.



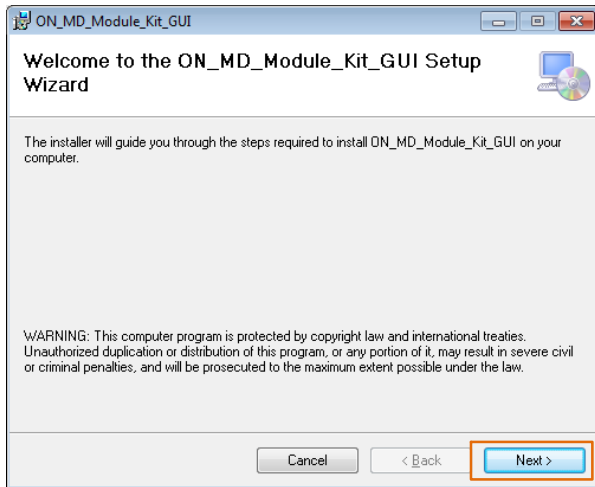
The following window shows the installation in progress. (This may take several minutes.)



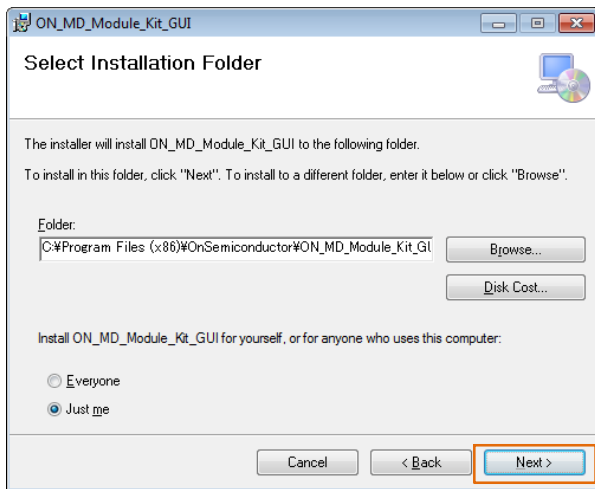
If the installation requires a reboot to complete, a message will appear on the screen. Please press “Yes” to reboot. Once rebooted, run  **setup.exe** installation file again.

*Some screen captures are taken from different motor driver modules

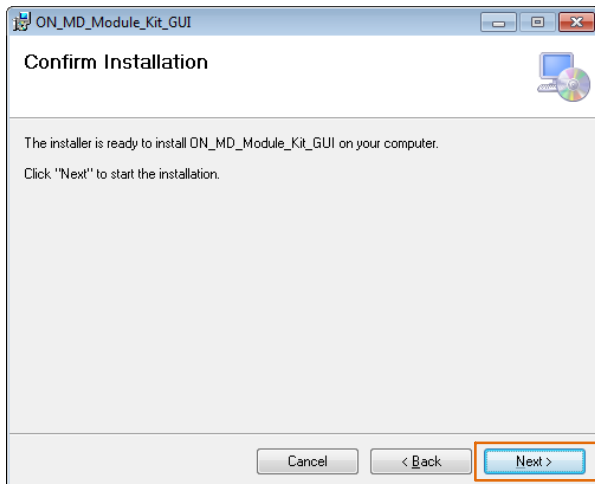
3. Install the GUI
Press [Next].



4. Specify installation folder, or just click [Next] to continue with the installation.

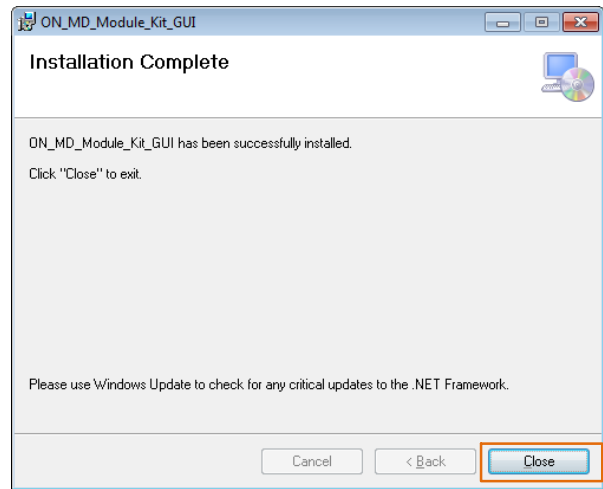


Please Press [Next].

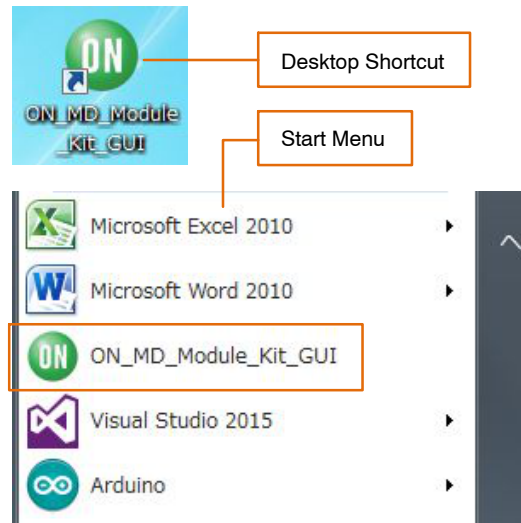


User administrative rights may be required for installation. If an alert for user account control is displayed, please select [Yes].

The following screen will be displayed after a successful installation.



Make sure the GUI shortcut icon is created on the desktop as shown below and that the program has been added to the Windows Start menu.

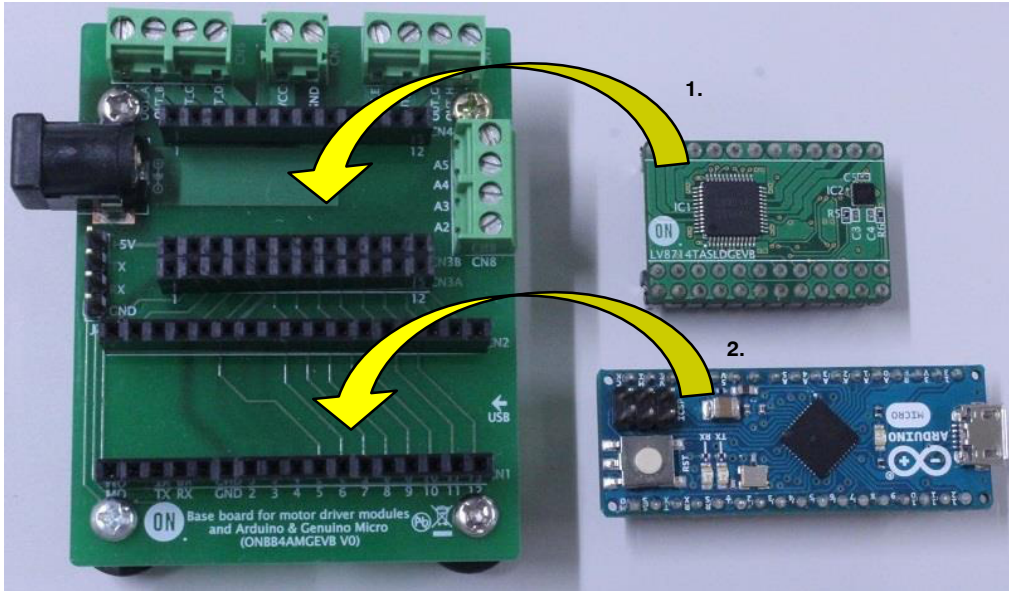


LV8714TASLDGEVK

Hardware Setup

1. Connect the LV8714TASLDGEVB (Motor Driver Module) to the ONBB4AMGEVB (Baseboard). Insert the module into the baseboard, while being careful not to bend the header pins.

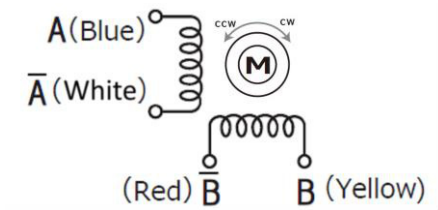
2. Connect the Arduino/Genuino Micro to the base board, as seen in the figure below. The USB connection port on the Arduino should be oriented to the “←USB” mark on the baseboard.



3. Remove the coating on the end of the motor wires by about 5 mm to 10 mm. Plug in them into the output terminals on the baseboard – OUT_A/B/C/D (CN5), and OUT_E/F/G/H (CN7). Firmly fix the wires by screwing down the terminals with a flat-tip screwdriver. Refer to the table below for motor connectivity:

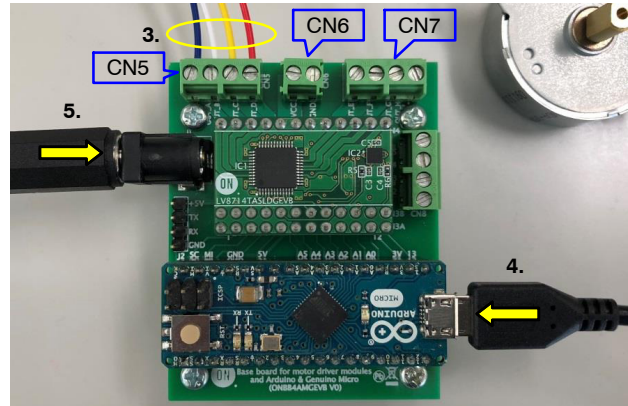
4. Connect the Arduino to a PC using a USB-cable.
5. Insert the AC Power adapter into the DC jack on the baseboard. Be sure that a **Center positive type** adaptor is being used with an **output voltage 4 ~ 16.5 V**.

| CN5 | | CN7 | |
|-------|-------------------|-------|-------------------|
| OUT_A | A (Blue) | OUT_E | A (Blue) |
| OUT_B | \bar{A} (White) | OUT_B | \bar{A} (White) |
| OUT_C | B (Yellow) | OUT_G | B (Yellow) |
| OUT_D | \bar{B} (Red) | OUT_H | \bar{B} (Red) |



Reference Motor Wiring Diagram

(Do not worry about incorrect wiring. Incorrect wiring will not cause any damage to hardware)

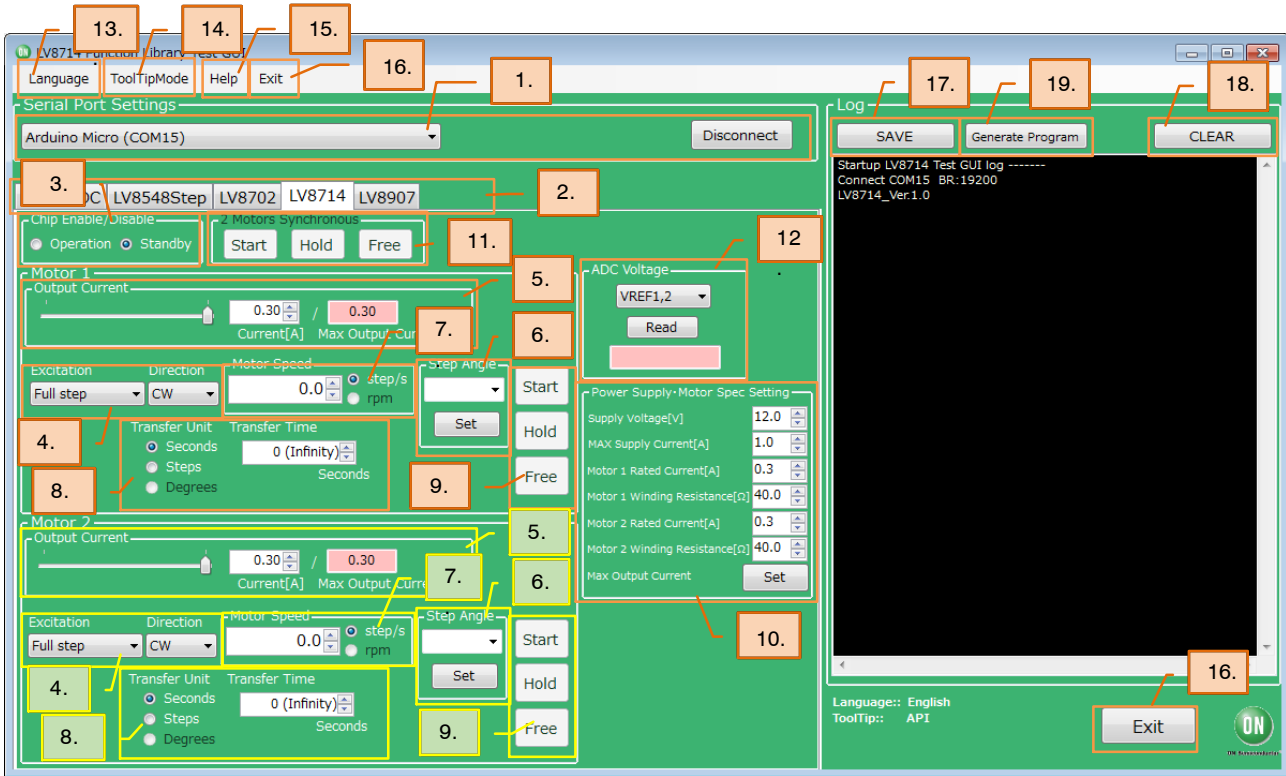


If using power supply cables, please connect the positive terminal to VCC and the negative terminal to GND in the CN6 connector.

The power supply terminals +/- at CN6 are polarity sensitive. (VCC=+, GND=-)

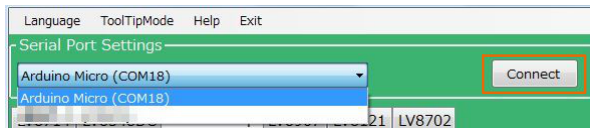
LV8714TASLDGEVK

How to Use the GUI



1. Double-click the GUI shortcut located on the desktop and connect the COM serial port.

4. Specify the desired excitation and direction of rotation.



Available COM ports will appear in the drop-down box. If the Arduino is connected correctly to the PC, an option for "Arduino Micro (COMx)" will be available (where x is the number associated with the port); select that COM port and click [Connect].

2. After connecting the Arduino, the GUI will automatically navigate to the "LV8714" tab.
3. Set the chip enable function.

As LV8714TA to the chip enable function. This function switches between Standby and Operation modes.

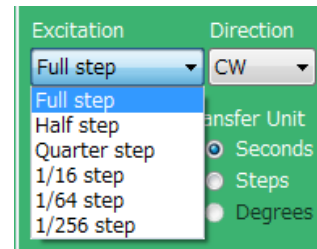
[Operation]... Operation mode
[Standby]... Standby mode

| Excitation (Excitation Method) | One-step Rotation Angle |
|--------------------------------|-------------------------|
| Full step | 1/1 Step angle |
| Half step | 1/2 Step angle |
| Quarter step | 1/4 Step angle |
| 1/16 step | 1/16 Step angle |
| 1/64 step | 1/64 Step angle |
| 1/256 step | 1/256 Step angle |

Direction:

CW...clockwise

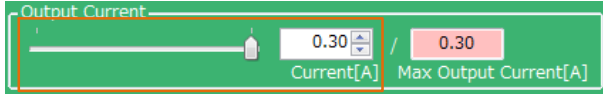
CCW...counter clockwise



NOTE: The direction of rotation may vary depending on how the motor wires are connected to the terminals.

LV8714TASLDGEVK

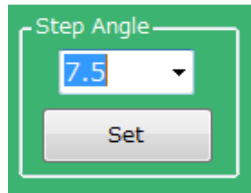
5. Set the output motor current.



The output current can be set within the limit the value (Box on the right) according to 10 by operating the slider or the value of the Box on the left. The setting values are incremented by 0.01 A, but the accuracy is limited due to the specifications of the Arduino.

NOTE: The output current is set to 0.3 [A] for the reference motor and the recommended adapter at startup, and it is not necessary to set it again for use in the same environment.

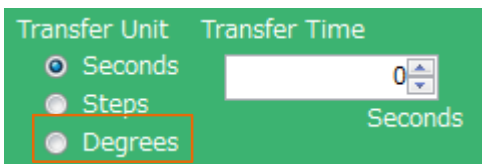
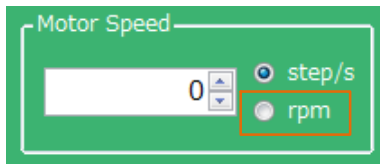
6. Set the step angle for the motor



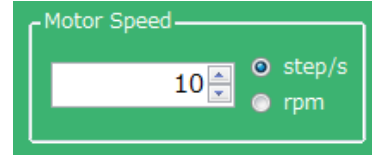
After setting the angle, click the [Set] button. Setting is not performed unless the [Set] button is pressed.

NOTE: The step angle will vary depending on the motor. If using the included reference motor, please set it to 7.5°.

The step angle setting enables the “rpm” setting under Motor Speed and “Degree” setting under Transfer Unit.



7. Set the motor speed in step/s or rpms.



step/s (=pps) : Frequency of the number of steps per second
rpm: Rotational speed of the motor per minute

The relation between step/s and rpm is expressed by the following formula.

$$\text{rpm} = \frac{(\text{Step angle } [^\circ]) \cdot (\text{steps/s}) \cdot 60 [\text{s}]}{360 [^\circ]} \cdot (\text{Excitation mode}) \quad (\text{eq. 1})$$

Substitute the following values into “Excitation mode” in the calculation formula.

| | |
|--------------|-------|
| Full step | 1 |
| Half step | 1/2 |
| Quarter step | 1/4 |
| 1/16 step | 1/16 |
| 1/64 step | 1/64 |
| 1/256 step | 1/256 |

Example)

When driving the reference motor (step angle = 7.5°) at 1/16 step and 160 step/s, rpm beomes

$$\frac{7.5 \cdot 160 \cdot 60}{360} \cdot \frac{1}{16} = 12.5 [\text{rpm}] \quad (\text{eq. 2})$$

Step/s can be entered in increments of 0.1 on the GUI, but for software specifications, the executable step/s is expressed by the following formula.

$$\text{step/s} = \frac{1}{100 [\mu\text{s}] \cdot n \cdot 4} \cdot (\text{Excitation mode}) \quad (\text{eq. 3})$$

(n: an integer)

For Half step only, n is an even number.

The value when n = 1 (In case of Half step, n = 2) is the maximum value for each excitation method.

However, stepper motors have difficulty in sudden acceleration, and cannot rotate correctly in response to a control signal with a speed exceeding the maximum rotation speed, and they stop or vibrate.

LV8714TASLDGEVK

When using the supplied reference motor (no load) with the recommended AC adapter (12 V / 1 A), refer to the table below to set the rotation speed.

| | | |
|--------------|----------------|------------------|
| Full Step | 1~360 step/s | 2~Approx 450 rpm |
| Half Step | 1~720 step/s | 1~Approx 450 rpm |
| Quarter Step | 1~1440 step/s | 1~Approx 450 rpm |
| 1/16 step | 1~4992 step/s | 1~Approx 390 rpm |
| 1/64 step | 1~9984 step/s | 1~Approx 200 rpm |
| 1/256 step | 1~10000 step/s | 1~Approx 49 rpm |

Stepper motors are not intended for sudden acceleration, and will stop or vibrate if the control exceeds the maximum rotation speed. When using the included reference motor with the recommended AC adapter (12 V / 1 A), please refer to the rotation speed according to the table below.

| | | |
|--------------|---------------|-----------|
| Full Step | 1~360 step/s | 2~450 rpm |
| Half Step | 1~720 step/s | 1~450 rpm |
| Quarter Step | 1~1440 step/s | 1~450 rpm |

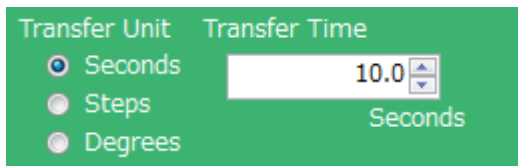
In addition, due to the characteristics of the reference motor, the motor may rotate at angles larger than expected even if the motor is controlled by excitation methods with higher resolutions.

- Set the units and value of the transfer condition for the control signal. After the specified Transfer condition elapses, the motor will stop and hold its torque. Since **the current continues to flow to the motor, it is important to pay attention to heat generation.** Please select the Free button to stop powering the motor.
For an unspecified rotation time, select 0 (Infinity).

Example 1 : Rotating the motor for **10 seconds**

Transfer Unit = *Seconds*

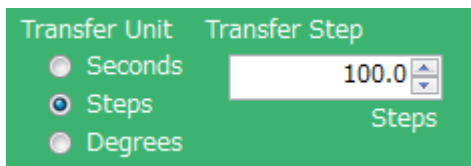
Transfer Time = *10 [seconds]*



Example 2: Changing the motor position by **100 steps**

Transfer Unit = *Steps*.

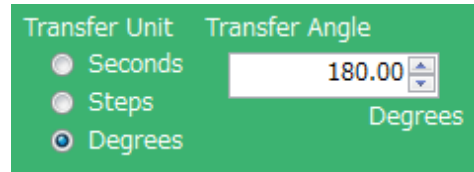
Transfer Step = *100 [steps]*



Example 3: Rotating the motor 180 degrees

Transfer Unit = *Degrees*

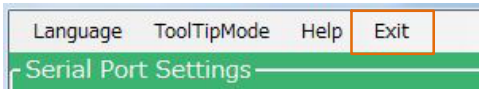
Transfer Angle = *180 [degrees]*



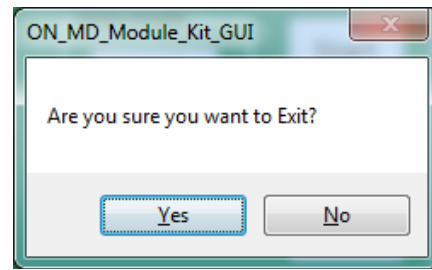
- When the Start button is pressed, the motor will rotate. If changes to the Output current, Excitation or Motor Speed are made, the changes will take effect upon pressing the Start button.
To change the direction of rotation, it is recommended to stop the motor first with the Hold button, change the value and then press the Start button to begin the motor rotation.
Pressing the Hold button causes the motor to stop and hold the torque.
To maintain the position while the motor rotates, or to pause the motor and restart from the same position, select the Hold button.
At this time, because of the current still flowing to the motor, attention must again be paid to heat generation.
- Set the specifications of the power supply and motor to be used.
Supply Voltage... Power supply voltage
MAX Supply Current... Maximum power supply current
Motor Rated Current... Motor Rated Current
Motor Winding Resistance... Motor coil resistance
Max Output Current Set buttons:
Calculate the maximum output current from the above four setting values, and 5 limit the set range of the output motor current to enable safer motor driving.
- Used to start and stop two motors at the same time.
Start: Used to start driving two motors at the same time or to update the settings at the same time.
Hold: Used to stop two motors at the same time. This is effective even when driving is started separately. At this time, the motor **holds the torque and fixes the motor shaft.**
Free: Used to stop two motors at the same time. At this time, **the motor loses torque and moves freely by external force.**
- Voltage reading function (Refer to [Appendix](#))
- GUI Language Settings (Refer to [Appendix](#))
- GUI Tool Tip Display (Refer to [Appendix](#))
- Help function
(See [Appendix](#) for more information)

16. Closing the GUI

When finished, exit the GUI by pressing the [Exit] button at the bottom right of the screen or by clicking “Exit” item at the top menu bar.



The following popup message will be displayed when exiting the GUI.



Select [Yes] to quit the GUI.

Select [No] to cancel the exit and return to the main screen.

If the GUI is closed while the motor is still running, the motor will be stopped and the window will close.

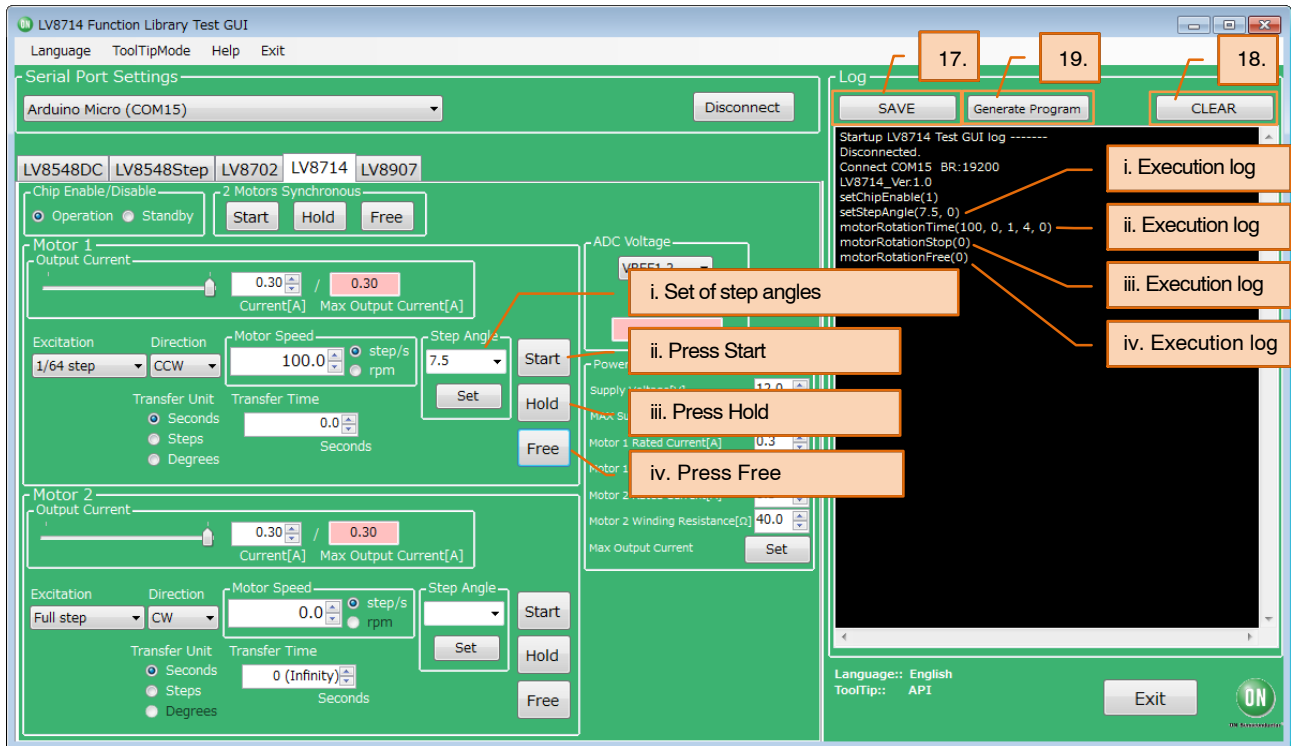
For more details on the following features, refer to the next section on [How to use the GUI Log](#):

- 17. Saving the GUI Log
- 18. Clear the GUI Log
- 19. Program generation

LV8714TASLDGEVK

How to Use the GUI Log

The log screen in the GUI shows the serial data sent to Arduino to control its API functions.



17. Saving the GUI log

By pressing the [SAVE] button, the content displayed on the work log can be saved as a .txt file or a .csv file.

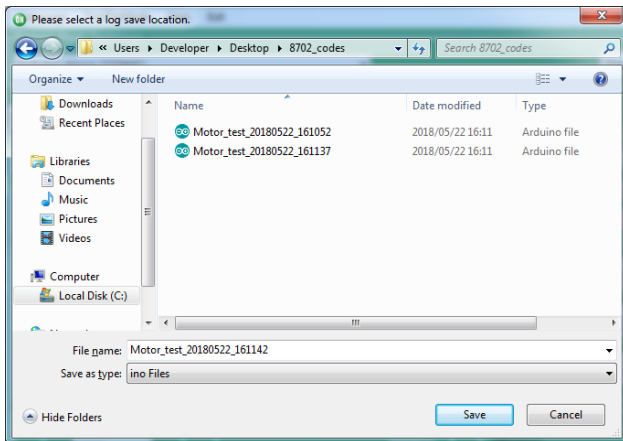
18. Clear the GUI Log

By pressing the [CLEAR] button, the content displayed on the work log will be erased.

19. Program generation

Pressing the [Generate Program] will output the executed API functions on the work log into a .ino program file that can be uploaded to the Arduino. By writing the output of the .ino file to the Arduino, motor control can be executed automatically through standalone operation according to the procedure generated through the GUI Log.


For more details regarding the Arduino program generation function, refer to the attached Appendix under 19 [Arduino Program Automatic Generation](#).

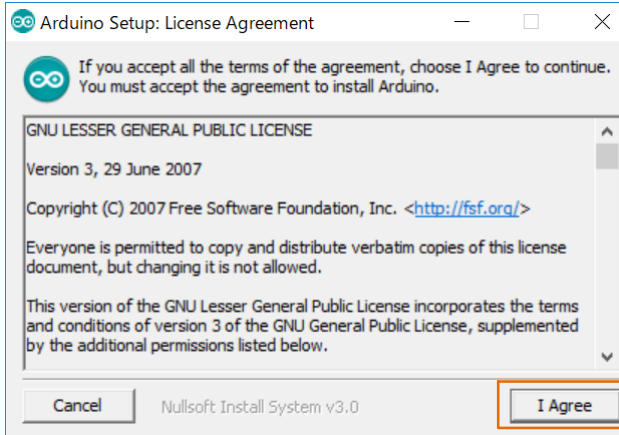


*Some screen captures are taken from different motor driver modules

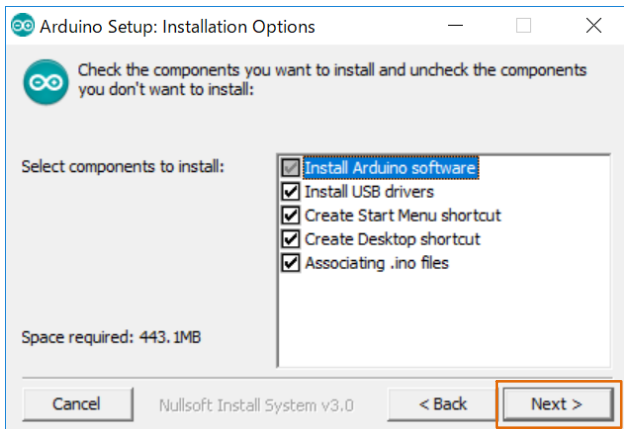
APPENDIX

Arduino IDE Installation

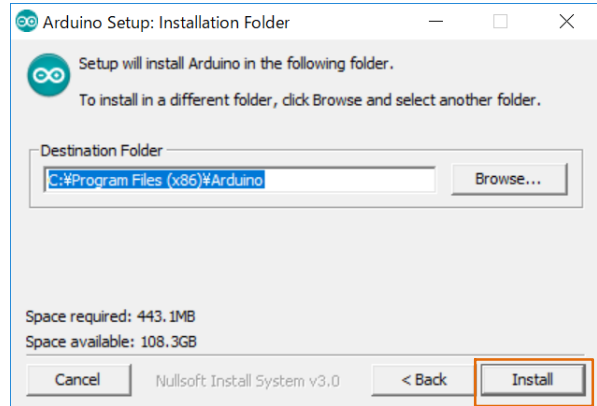
1. Run  `arduino-1.8.4-windows.exe` installer that are included in.ZIP files downloaded from our web site.
(To avoid software from not operating properly, please use this version and refrain from updating)
2. Read through and agree to the terms and conditions during the setup by clicking the “I Agree” button



3. Leave the following default components to install as is and press “Next”



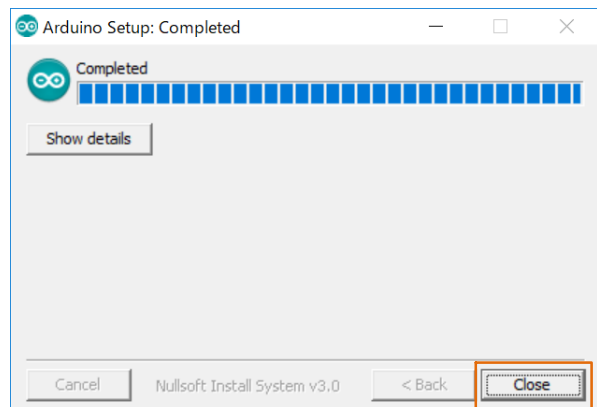
4. Set the install directory and press “Install”



5. If prompted, please install the Arduino USB Drivers
 - Arduino USB Driver
 - Genuino USB Driver
 - libusb-win32
 - Adafruit Industries LLC Port (COM andLPT)
 - Linino Port (COM and LPT)



6. Below shows a completed installation for the Arduino IDE. Feel free to click “Close.”



LV8714TASLDGEVK

Supplemental GUI Content

The following step numbers correspond to the step numbers in the “How to use the GUI” section in the Quick Start Guide

12. Measures the LV8714 VREF terminal voltage and A3, A4, and A5 external inputs.

Select VREF or Ax from the drop-down list and press READ to display the voltage in the pink box.

VREF1, 2... Motor1 Output Motor Current Reference Voltage

VREF3, 4... Motor2 Output Motor Current Reference Voltage

Displays the reference voltage for setting the output motor current of 5.

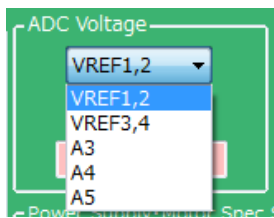
$$\frac{VREF [V]}{3} = \left(\frac{\text{Output Motor Current}}{4000} \right) \cdot 1.5 [k\Omega] \quad (\text{eq. 4})$$

When the output motor current is set to 0.30 A, VREF becomes

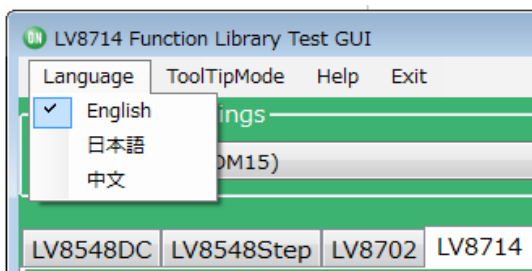
$$VREF [V] = \frac{0.30 [A]}{4000} \cdot 1.5 [k\Omega] \cdot 3 = 0.3375 [V] \quad (\text{eq. 5})$$

The GUI displays the approximate value of this. The settable minimum value of the VREF terminal voltage is around 0.2 V (approximately converted to a current value 0.18 A), and the maximum value is 1.5 V (approximately converted to a current value of 1.3 A).

Since A3, A4, and A5 are open, any signal line can be connected to their each terminals of CN8 on the Base board to indicate the signal voltage levels. (Since A2 is connected to VREF3,4, it can't be used.)

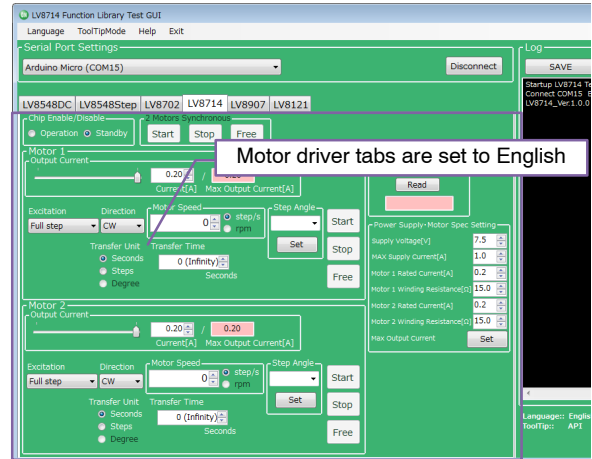


13. GUI Language Settings



Languages can be changed from the menu on the top left of the window.

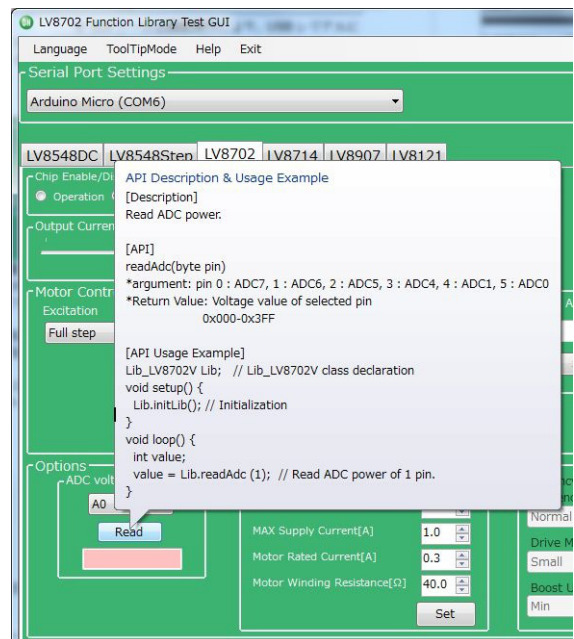
Motor driver tab language are not changed. (Please see tool tip for translations)



14. GUI Tool Tip Display

The GUI sends serial data to execute API functions in the Arduino through USB.

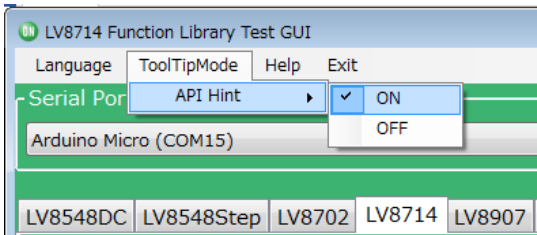
With the API Hints turned on, hover over different buttons and settings to display descriptions and functions associated with those items.



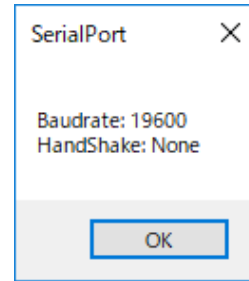
*Some screen captures are taken from different motor driver modules

LV8714TASLDGEVK

The API hints can be turned off by navigating through the ToolTipMode menu at the top of the window

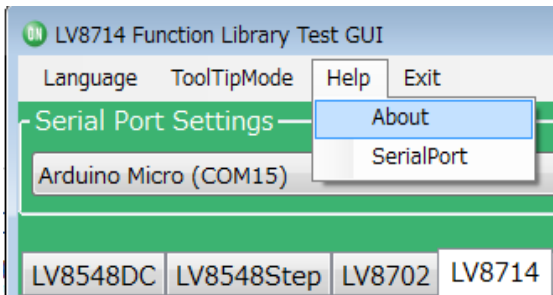


Selecting SerialPort will display the overview of serial communication.

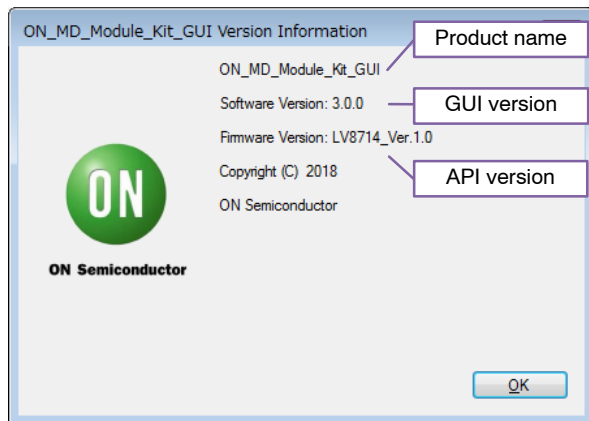


15. Help function

From the help menu, it is possible to view GUI and API version information, as well as serial communication details.



Selecting About will display the following window.



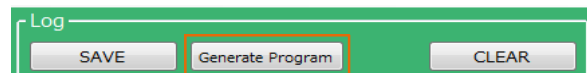
19. Arduino Program Automatic Generation

When using the GUI, automatically generated Arduino code will be output to the log window. It is possible to export the code to an Arduino sketch (.ino) file that can be used with the Arduino IDE.

Once the automatically generated code is installed into the Arduino, the Arduino works in the same sequence which was operated by the user using the GUI.

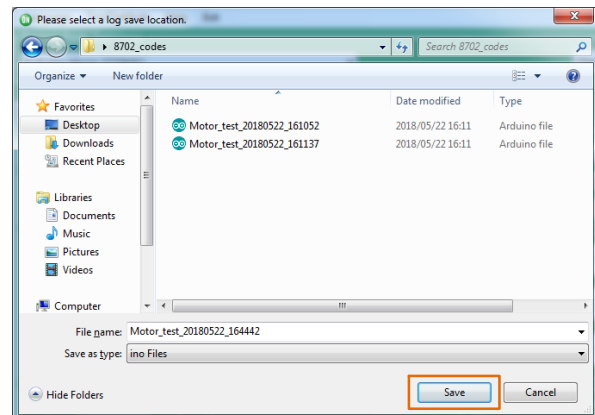
1) Generating and writing Arduino code

Pressing Generate Program will open a save file dialog.



Please select the desired directory for the file to be saved (e.g. Desktop or Documents) and click Save.

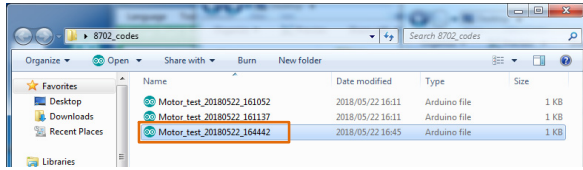
When using the GUI, **LV8714_Program.ino** is the name of the program, so a different name will need to be used to save the exported file.



*Some screen captures are taken from different motor driver modules

LV8714TASLDGEVK

Double click the newly saved Arduino program to open the Arduino IDE.



The GUI and Arduino IDE cannot be connected to a PC at the same time. To upload the Arduino program, please exit or select Disconnect in the GUI, and proceed working in the Arduino IDE. Follow the instructions in the Quick Start Guide in P4 “[Compiling the Arduino Program – Write to Arduino](#)”

2) Using the generated Arduino program

After each API operation in the generated program, a delay (0) is inserted. By changing the value of the argument (0)* to the delay function, the user is able to freely adjust the rotation time, as well as the interval time between each API call execution. This can be used to achieve the desired stand-alone operation.

*: The Arduino standard clock TIMER0 is affected by the API functionality of the LV8714. Therefore, the duration of the delay functions is 1/64000 of a second.

Enter (64000) to set 1 second

Example of Interval Time Change

```
#include <LV8714_Lib.h>
#include <TimerThree.h>
#define TIMER 100
Lib_LV8714TA Lib;
void setup()
{
  Serial.begin(19200);
  Lib.initLib();
  Timer3.initialize(TIMER);
  Timer3.attachInterrupt(interrupt);
  delay(64000); → NOTE
  Lib.setChipEnable(1);
  delay(0);//0msec
  Lib.setStepAngle(7.5, 0);
  delay(0);//0msec
  Lib.motorRotationDeg(10, 720.00, 0, 0, 0);
  delay(640000);//10000msec
  Lib.motorRotationFree(0);
  delay(0);//0msec
  Lib.setChipEnable(0);
  delay(0);//0msec
}
void interrupt()
{
  Lib.timerFire(TIMER);
}
void loop()
{
}
```

Control contents in the frame

“*motorRotationDeg(10, 720.0, 0, 0)*”
will start the Motor 1 with Full step, CW, 10 Step/s, and an angle of 720 deg

“*delay(640000)*”
will drive the Motor for 10000 [msec]
(10 seconds)

“*motorRotationFree(0)*”
will stop the Motor 1

After the time specified in the delay command has elapsed, the next command will be executed.

If the delay setting is 0 or too short, some motor operations will complete so quickly, that it will be imperceptible.

NOTE: Due to this setting, the 5 sec delay is executed at the beginning of the setup method in the case of a new USB connection, Arduino reset, or upon uploading a sketch to the Arduino.
(Time is adjustable)

LV8714TASLDGEVK

BOARD SCHEMATIC (1/2)

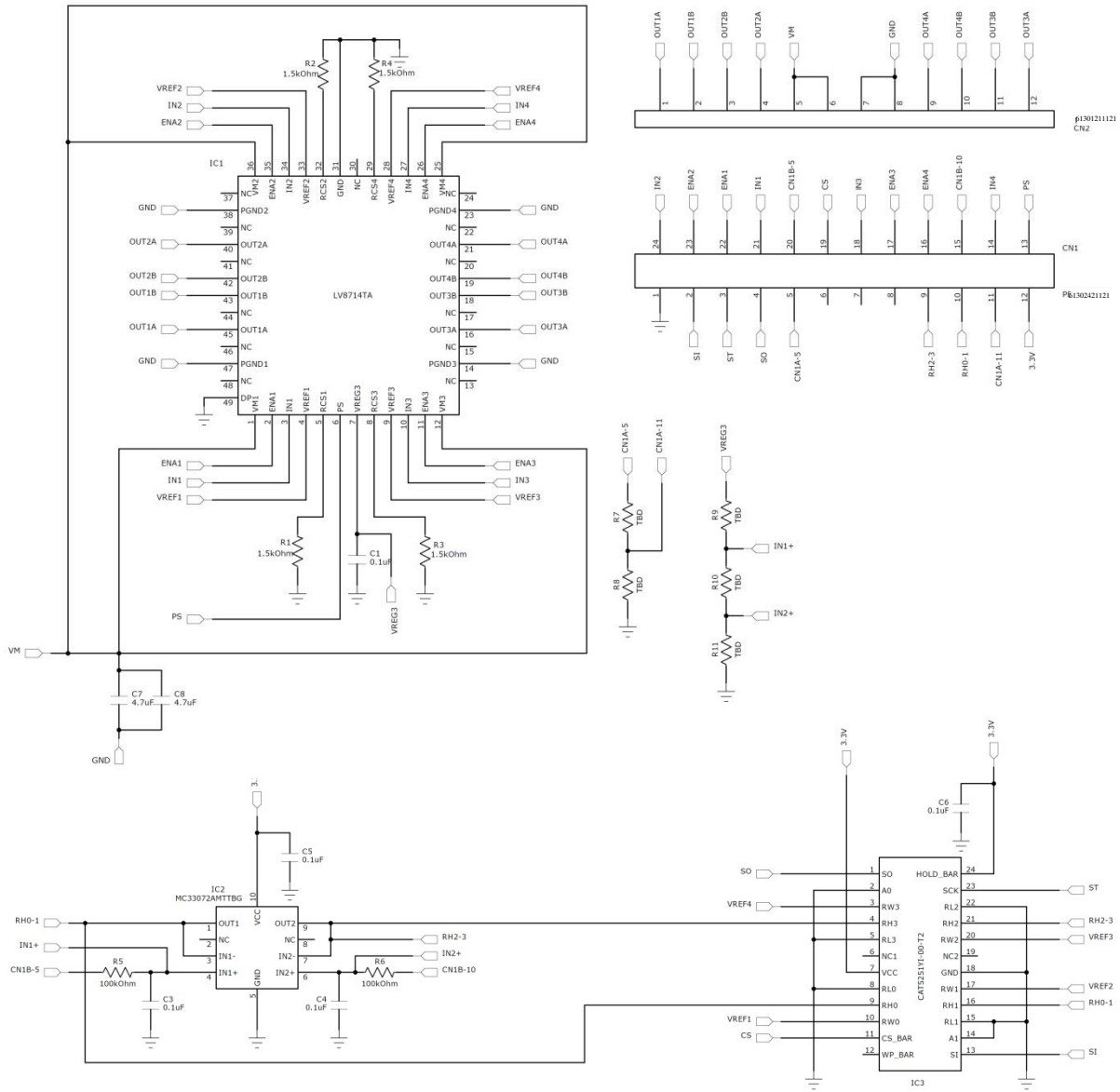


Figure 3. LV8714TASLDGEVB Board Schematic

LV8714TASLDGEVK

Table 1. LV8714TASLDGEVB BILL OF MATERIALS

| Designator | Qty. | Description | Value | Tolerance | Footprint | Company | Part Number |
|--------------|------|-------------------------|---------------|-----------|---------------|----------------------|--------------------|
| IC1 | 1 | Motor driver IC | - | - | TQFP48 | ON Semiconductors | LV8714TA |
| IC2 | 1 | Operational amplifier | - | - | WQFN10 | ON Semiconductors | MC34072AMTTBG |
| IC3 | 1 | Digital Potentiometer | - | - | TSSOP24 | ON Semiconductors | CAT5251YI-00-T2 |
| R1 - 4 | 4 | Chip resistor | 1.5 kΩ, 0.1 W | ±5% | 1005 (0402) | KOA | RK73B1ETT*152J |
| R5, R6 | 2 | Chip resistor | 100 kΩ, 0.1 W | ±5% | 1005 (0402) | KOA | RK73B1ETT*104J |
| R7 | 1 | Chip resistor | 100 kΩ, 0.1 W | ±5% | 1005 (0402) | KOA | RK73B1ETT*104J |
| R8 | 1 | Chip resistor | 33 kΩ, 0.1 W | ±5% | 1005 (0402) | KOA | RK73B1ETT*333J |
| R9, R10, R11 | 3 | Chip resistor | TBD | ±5% | 1005 (0402) | - | - |
| C1 | 1 | Chip capacitor | 0.1 μF, 35 V | ±10% | 0603 (0201) | Murata Manufacturing | GRM033R6YA104KE14# |
| C3,C4 | 2 | Chip capacitor | 0.1 μF, 35 V | ±10% | 0603 (0201) | Murata Manufacturing | GRM033R6YA104KE14# |
| C5,C6 | 2 | Chip capacitor | 0.1 μF, 35 V | ±10% | 0603 (0201) | Murata Manufacturing | GRM033R6YA104KE14# |
| C7,C8 | 2 | Chip capacitor | 4.7 μF, 35 V | ±10% | 1608 (0603) | Murata Manufacturing | GRM188R6YA475KE15# |
| CN1A, 1B | 1 | Pin header to baseboard | 12 pins x 2 | - | 30.48 x 5.08 | Würth Electronik | 61302421121 |
| CN2 | 1 | Pin header to baseboard | 12 pins | - | 30.48 x 2.54 | Würth Electronik | 61301211121 |
| PCB | 1 | PCB | - | - | 30.48 x 20.32 | | |

NOTE: Parts highlighted in yellow are not mounted at the time of product shipment.

LV8714TASLDGEVK

BOARD SCHEMATIC (2/2)

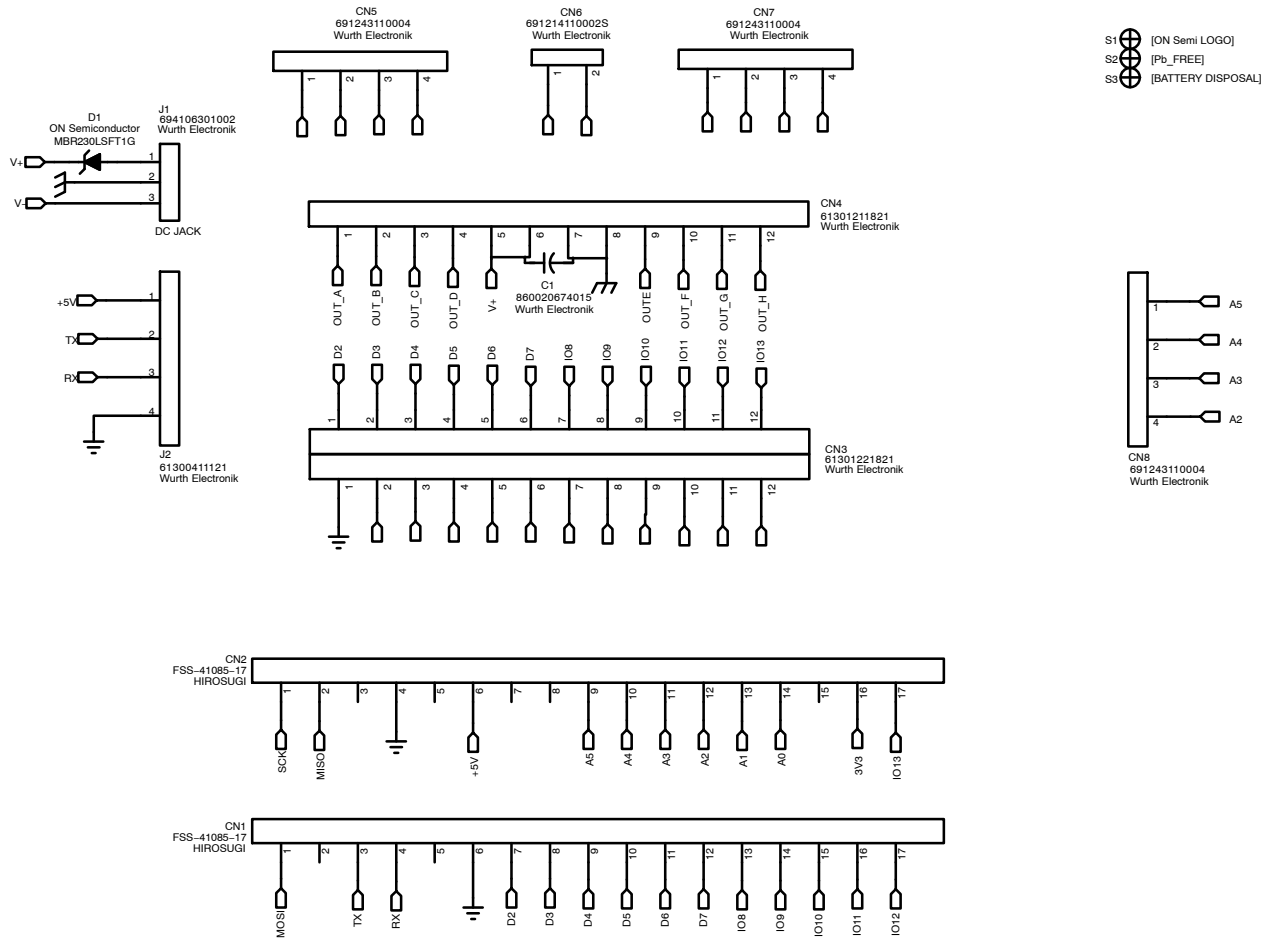


Figure 4. ONBB4AMGEVB Board Schematic

Table 2. BASEBOARD BILL OF MATERIALS

| Designator | Qty. | Description | Value | Tolerance | Footprint | Company | Part Number |
|------------|------|-------------------------|--------------|-----------|-----------------------------------|------------------|---------------|
| D1 | 1 | Diode | - | - | SOD123 | ON Semiconductor | MBR230LSFT1G |
| CN1, 2 | 2 | Arduino Micro connector | - | - | ∅1.02 x 17 - 2.54 pitch | Hirosugi-Keiki | FSS-41085-17 |
| CN3 | 1 | Module connector | - | - | ∅1.02 x 12 x 2 lines - 2.54 pitch | Wurth Electronik | 61302421821 |
| CN4 | 1 | Module connector | - | - | ∅1.02 x 12 - 2.54 pitch | Wurth Electronik | 61301211821 |
| CN5, 7, 8 | 3 | Motor connectors | - | - | ∅1.1 x 4 - 3.5 pitch | Wurth Electronik | 691243110004 |
| CN6 | 1 | Power connectors | - | - | ∅1.1 x 2 - 3.5 pitch | Wurth Electronik | 691214110002S |
| J1 | 1 | DC barrel jack | - | - | 9.0 x 14.5 | Wurth Electronik | 694106301002 |
| J2 | 1 | UART pin headers | - | - | ∅1.1 x 4 - 2.54 pitch | Wurth Electronik | 61300411121 |
| C1 | 1 | Electrolytic capacitor | 100 µF, 50 V | ±10% | - | Wurth Electronik | 860020674015 |
| PCB | 1 | PCB | - | - | 80 x 60 | | |

NOTE: When using a custom-made baseboard, be sure to **mount an electrolytic capacitor equivalent to C1 between VCC and GND.** Neglecting to install this capacitor may lead to damage and malfunction of any connected driver modules.

LV8714TASLDGEVK

LICENSE

The GUI included in this kit developed by ON Semiconductor, GUI firmware and API library are subject to the following licenses.


Copyright 2018 ON Semiconductor. All rights reserved.

This software and/or documentation is licensed by ON Semiconductor under limited terms and conditions.

The terms and conditions pertaining to the software and/or documentation are available at http://www.onsemi.com/site/pdf/ONSEMI_T&C.pdf (“ON Semiconductor Standard Terms and Conditions of Sale, Section 8 Software”).

Do not use this software and/or documentation unless you have carefully read and you agree to the limited terms and conditions.

By using this software and/or documentation, you agree to the limited terms and conditions.

ON Semiconductor and  are trademarks of Semiconductor Components Industries, LLC dba ON Semiconductor or its subsidiaries in the United States and/or other countries. ON Semiconductor owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of ON Semiconductor's product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent-Marking.pdf. ON Semiconductor reserves the right to make changes without further notice to any products herein. ON Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does ON Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using ON Semiconductor products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by ON Semiconductor. "Typical" parameters which may be provided in ON Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. ON Semiconductor does not convey any license under its patent rights nor the rights of others. ON Semiconductor products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use ON Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold ON Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that ON Semiconductor was negligent regarding the design or manufacture of the part. ON Semiconductor is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:

Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800-282-9855 Toll Free
USA/Canada
Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910

ON Semiconductor Website: www.onsemi.com
Order Literature: <http://www.onsemi.com/orderlit>

For additional information, please contact your local Sales Representative