Motor Driver

Stepper Motor Driver IC EVK STEPMO_EVK_20x

ROHM Stepper Motor Driver IC Evaluation Kit based on Arduino/Genuino Platform

● Abstract

This evaluation board manual describes the usage of ROHM's stepper motor driver IC evaluation kit (EVK) called STEPMO_EVK_20x. It is designed as a plug-in board (Shield) for popular Arduino microcontroller platform. This document provides guidelines to quickly setup the hardware and software for fast and easy stepper motor driver IC performance evaluation.

● Description of Supported ICs

This EVK supports a variety of pin-compatible ROHM stepper motor driver ICs from standard, microstep, low voltage and high voltage families with CLK-IN (clock input) or PARA-IN (parallel input) control. They integrate PWM constant-current drive with adjustable decay ratio and the ability for full, half and microstepping. The ICs feature single supply operation by integrating the voltage regulator for the low power logic together with highly efficient DMOS output power stages. These do not require an internal charge pump so the motor drivers achieve low EMI performance. The motor drivers integrate various protection functions such as Ghost Supply Prevention (GSP), Thermal Shutdown (TSD), Over-Current Protection (OCP), Under / Over Voltage Lockout (UVLO / OVLO) high ESD resistance and Pin Short Protection for robust and reliable operation. The ICs are housed in compact HTSSOP-B28 power packages. For further details, please also consult the relevant product datasheets at http://www.rohm.com.

● EVK Key Specifications (Across Model Versions)
  ▪ Input voltage range………………..……8~42V
  ▪ Maximum Output Current Range..........1A~2.5A
  ▪ Stepping Modes……………………¼,½,¼,⅛,¹⁄₁₆
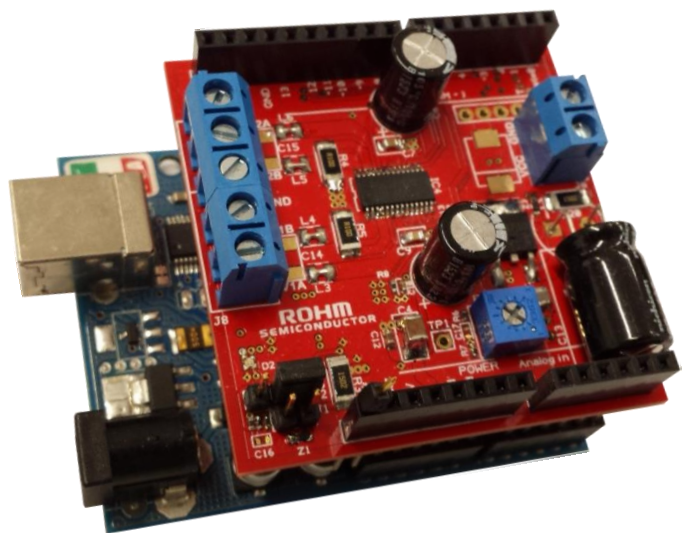
● **EVK Features**
  ▪ Designed as plug-in Shield for Arduino platform
  ▪ Recommended: Arduino Uno
  ▪ Several model variants covering wide range of pin- compatible stepper driver ICs
  ▪ Support of bipolar or unipolar stepper motors
  ▪ Adjustable current limit
  ▪ Adjustable current decay mode
  ▪ Single supply operation
  ▪ Stackable design to allow controlling two motors at the same time
  ▪ Reverse power supply protection
  ▪ Software library for Arduino IDE
  ▪ Arduino example programs (Sketches)
  ▪ Visit
    http://www.rohm.com/web/eu/arduino-stepper-motor-shield
    for details

Table of Contents

1. Safety Instructions

1.1 Warnings

- This evaluation kit (EVK) should be operated in a well ventilated environment and, if used inside a case, the case should not be covered.
- This EVK should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- All peripherals used with the EVK should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met.
- Where peripherals are connected that do not include the cable or connector, the cable or connector used must offer adequate insulation and operation in order that the requirements of the relevant performance and safety are met.
- The connection of incompatible devices to the EVK may affect compliance or result in damage to the unit and invalidate the warranty.

1.2 Instructions for Safe Use

- Do not expose the EVK to water, moisture or place on a conductive surface whilst in operation.
- Do not expose the EVK to temperatures, which are out of the operational range specified for hardware and components populating the EVK.
- Take care whilst handling to avoid mechanical or electrical damage to the printed circuit board and components.
- Avoid handling the printed circuit board while it is powered. Only handle by the edges to minimize the risk of electrostatic discharge damage.
- Do not short any outputs to each other, to the supply or to GND.
- Do not reverse polarize the power supply connection.
- Do not operate the EVK outside its specified ratings.
- Take care to monitor the PCB and motor driver IC temperature when operating with high load currents and do not exceed the absolute maximum ratings of all components.

2. Introduction

This evaluation board manual describes the usage of ROHM's stepper motor driver IC evaluation kit (EVK) called STEPMO_EVK_20x. The purpose of the EVK is to allow the test and evaluation of the stepper driver IC in professional research and development environments. It is designed as a plug-in board (Shield) for popular Arduino microcontroller platform. This document provides guidelines to quickly setup the hardware and software for fast and easy stepper motor driver IC performance evaluation. For further details about the motor driver ICs and other ROHM products, please also consult the relevant product datasheets and general information at http://www.rohm.com.

The EVK provided by ROHM consists of hardware and software. This document refers to hardware revision 1605.
Please check http://www.rohm.com/web/eu/arduino-stepper-motor-shield for downloading additional useful information such as application notes, Gerber files, schematics, bill of materials, software libraries, example programs etc.

For operation of the EVK an Arduino motherboard is required. For all tests and performance evaluation by ROHM the EVK has been used together with an Arduino Uno R3 motherboard.
Arduino (also Genuino) is an open-source electronics microcontroller platform based on easy-to-use hardware and software.
The concept of Arduino includes a defined IO and Power pinning which allows the easy connection of daughter application boards into the microcontroller mother board. These plug-in boards are also called Shields. The microcontroller programs created using the Arduino IDE are called Sketches. For more information about Arduino please visit https://www.arduino.cc/.

Of course, the EVK hardware can be operated with any other microcontroller platform if desired if this is preferred for evaluation of the IC performance. For this case the required information about the necessary IO and power signals is also provided within this document.

2.1 Block Diagram

A block diagram of the EVK hardware is depicted in Figure 1. The motor voltage applied to screw terminal J6 is also the supply for the motor driver IC. For single supply operation a Buck regulator creates a 7V input voltage for the Arduino motherboard (Vin) which again uses this voltage to create the logic supplies of 5V/3.3V. Therefore it is not necessary to power the microcontroller board separately. The 5V created by the Arduino motherboard is used by the EVK to supply logic blocks and voltage levels to configure the stepper motor driver IC.

The EVK is designed to be stackable to allow evaluating two stepper driver ICs at the same time. For this purpose the Jumper J1 is used to configure the board from Master (default, used for the first Shield) and Slave (only used for a second stacked Shield). As Slave the Buck Regulator is turned off with high impedance output because the Vin Voltage is already generated by the Master Shield.

Based on the setting of J1 the control signals for the motor driver IC are routed via multiplexers to different Arduino IOs to allow independent motor drive. The Master or Slave configuration is indicated by an LED (Master: green, Slave red).
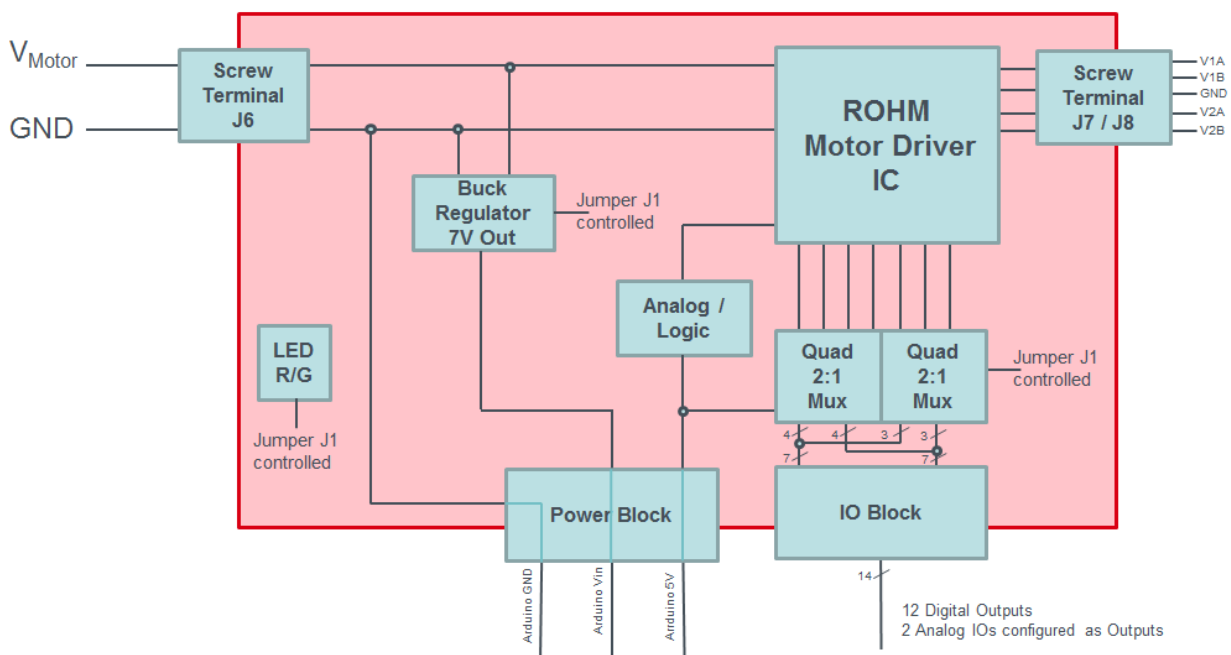The screw terminal J7/J8 allows the connection of a 4-wire bipolar or a 5-to-6-wire unipolar stepper motor.

Figure 1: EVK Shield Block Diagram

2.2 Model Overview

This stepper motor EVK has several different model variants. The main difference between the variants is the equipped ROHM stepper motor driver IC. Table 1 gives an overview of the different IC features and the according naming of the EVK. The ID suffix "x" of the EVK name STEPMO_EVK_20x denotes the model version. For further information about the ICs please follow the hyperlinks to the datasheets on ROHM's global web site.

ID	ROHM Motor Driver IC	Supply Voltage / V		Max. Current per Phase / A	Supported Step Modes	Control Type
x		Min	Max	25°C		
1	BD63510AEFV [1]	8	28	1.0	$\frac{1}{1}$,$\frac{1}{2}$,$\frac{1}{4}$,$\frac{1}{16}$	CLK-IN
2	BD63520AEFV [1]	8	28	2.0 [2]	$\frac{1}{1}$,$\frac{1}{2}$,$\frac{1}{4}$,$\frac{1}{16}$	CLK-IN
3	BD63524AEFV [1]	8	28	2.5 [2]	$\frac{1}{1}$,$\frac{1}{2}$,$\frac{1}{4}$,$\frac{1}{8}$	CLK-IN
4	BD63710AEFV	19	28	1.0	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	CLK-IN
5	BD63715AEFV	19	28	1.5	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	CLK-IN
6	BD63720AEFV	19	28	2.0 [2]	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	CLK-IN
7	BD63843EFV	19	28	1.0	$\frac{1}{1}$,$\frac{1}{2}$,$\frac{1}{8}$,$\frac{1}{16}$	CLK-IN
8	BD63847EFV	19	28	2.0 [2]	$\frac{1}{1}$,$\frac{1}{2}$,$\frac{1}{8}$,$\frac{1}{16}$	CLK-IN
9	BD63873EFV	19	28	1.0	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	CLK-IN
A	BD63875EFV	19	28	1.5	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	CLK-IN
B	Reserved					
C	BD63872EFV	19	28	1.0	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	PARALLEL-IN
D	Reserved					
E	BD63876EFV	19	28	2.0 [2]	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	PARALLEL-IN
F	BD6425EFV	19	42	1.5	$\frac{1}{1}$,$\frac{1}{2}$(A),$\frac{1}{2}$(B),$\frac{1}{4}$	CLK-IN

Note 1: In development.
Note 2: Will require additional heat sinking or cooling techniques to achieve the maximum current continuously.

Table 1: EVK Model Variants

3. Hardware Description

The schematic of the EVK is shown in Figure 2 and Figure 3 and the Bill-of-Materials (BOM) is summarized in Table 2. Please note that the depicted schematics and BOM refer to model version 2 of the EVK. The other model versions differ in some details while the major differences are the in assembled motor driver IC part number (see Table 1) and the different current sense resistor values. For schematics and BOMs of all available model versions please visit http://www.rohm.com/web/eu/arduino-stepper-motor-shield.

Main part of the EVK is the ROHM stepper motor driver labelled IC4. Q2, R20 and Z3 form a reverse power supply protection. Z2 is a TVS diode to protect against transient surge voltages. Supply decoupling and filtering is accomplished by C13, L2, and C6-C9. Voltage dividers R9-R11 and R6, R19, R7 set the necessary voltage levels for VREF and MTH. Test pins are connected to GND via 0Ω-resistors.

The PWM frequency is set by C10 and R8 to the standard value as suggested in the IC datasheet. The resulting chopping frequency is ~ 25 kHz. For fine tuning the stepper motor performance it is usually recommended to find the optimum frequency as a trade-off between motor noise and current waveform distortion. Thus, if desired, the values of C10 and R8 could be changed by the user by soldering different components. Please refer to the IC datasheet for recommended values.

D2 is an instance of ROHM's ultra-compact PICOLED[TM]-Duo two-color type LED indicating Master/Slave mode. It is switched by Q1 and controlled by Jumper J1.

J5 is the defined Arduino compatible set of header rows for IOs and power.

IC1 and IC2 are ROHM BU4551 parts which contain 4 times a 2:1 multiplexer each. They are required to route the correct control signals to the motor driver IC depending on the setting of Jumper J1.

IC3 is ROHM's BD9G101G wide input range DC/DC step-down regulator which integrates a high-side power MOSFET into its small SOT-23 package while able to provide up to 0.5A DC output. It is used together with adjacent circuitry to create the 7V input voltage to the Arduino with a high efficiency.
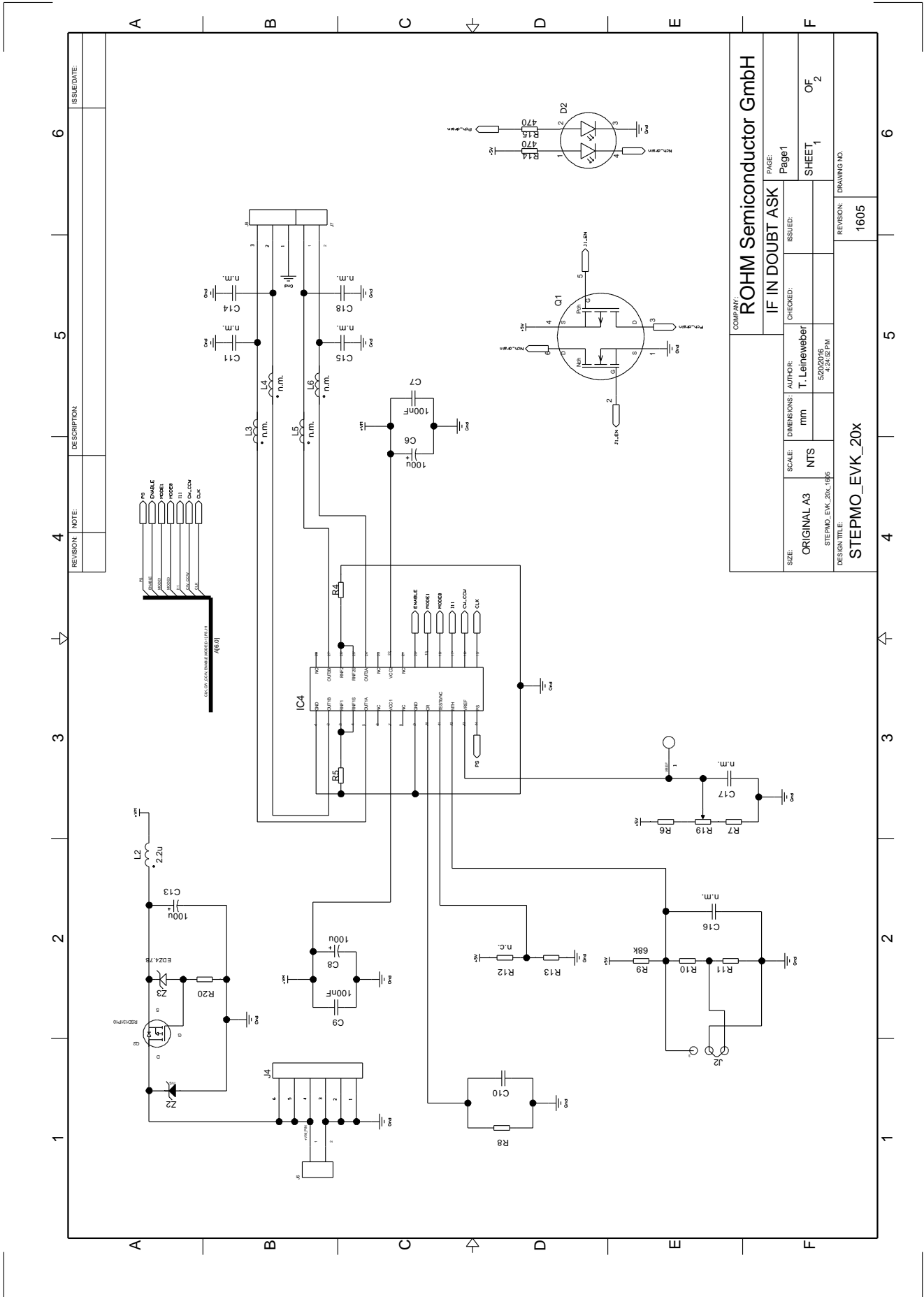
3.1 Schematic

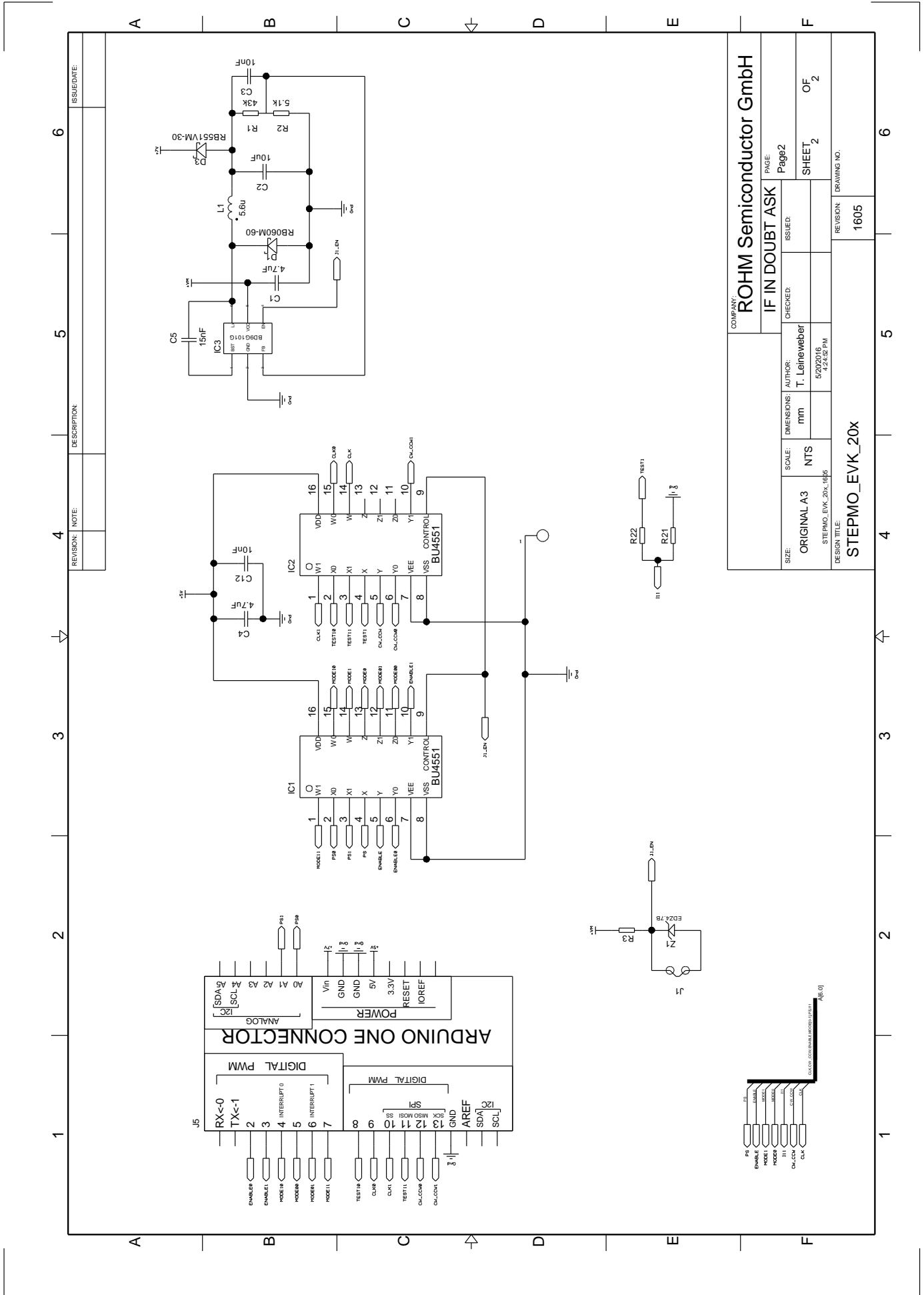Figure 2: EVK Schematic – STEPMO_EVK_20x – Page 1

Figure 3: EVK Schematic – STEPMO_EVK_20x – Page 2

The schematic of the EVK is plotted in Figure 2 and Figure 3. Component values valid for all model versions are annotated, for all other values please refer to the BOM of each model version available at http://www.rohm.com/web/eu/arduino-stepper-motor-shield.

3.2 Bill of Materials

The bill of materials is given Table 2 as example for model version 2.
For all other model version BOMs please refer to http://www.rohm.com/web/eu/arduino-stepper-motor-shield.

Component Name	Value	Description	Supplier	Part number	Qty
C1, C4	4.7u	Cer Cap 22uF 20% 16V X7R 1210	TDK	C3225X7R1H475K250AB	2
C2	10u	Cer Cap 22uF 20% 16V X7R 1210	TDK	C3225X7R1H106M250AC	1
C3, C12	10n	Cer Cap 10nF 10% 25V X7R 0603	TDK	C1608X7R1E103K080AA	2
C5	15n	Cer Cap 15nF 20% 50V X7R 0603	TDK	C1608X7R1H153M080AA	1
C6, C8	100u	100uF 20% 63V Electrolyte Capacitor 8mm x 11.5mm	Nichicon	UVY1J101MPD	2
C7, C9	100n	Ceramic capacitor 100nF 10% 50V X7R 0805 Wurth	Wurth	885012207098	2
C10	1n	Ceramic capacitor 1nF 20% 50V X7R 0402	TDK	C1005X7R1H102M050BA	1
C13	100u	100uF 20% 100V Electrolyte Cap. 10mm x 16mm	Nichicon	UVK2A101MPD	1
C11, C14, C15, C16, C17, C18	n.m.	Ceramic capacitor, not mounted			6
D1		Schottky Diode 2A 60V	ROHM	RB060M-60	1
D2		PICOLED^{TM}-Duo two-color type LED (green + red) in 1 package	ROHM	SML-P24MUWT86	1
D3		Schottky Diode 0.5A 20V	ROHM	RB551VM-30	1
IC1, IC2		Quad 2ch Analog Multiplexer/Demultiplexer	ROHM	BU4551BFV-E2	2
IC3		Wide Input Range DCDC Regulator	ROHM	BD9G101G-TR	1
IC4		Stepper Motor Driver	ROHM	BD63520AEFV-E2	1
J1		Jumper 2 Way	FCI	77311-118-02LF	1
J2		Jumper 3 Way	FCI	77311-118-03LF	1
J4		ARDUINO Stackable Header 6pin 14mm5 height	Arduino	A000084	0.5
J5 C		ARDUINO Stackable Header 6pin 14mm5 height	Arduino	A000084	0.5
J5 A, J5 D		ARDUINO Stackable Header 8pin 14mm5 height	Arduino	A000085	1
J5 B		ARDUINO Stackable Header 10pin 14mm5 height	Arduino	A000086	0.5
J6, J7		Connector Screw 5mm pitch	Wurth	691102710002	2
J8		Connector Screw 5mm pitch	Wurth	691102710003	1
L1	5.6u	5.6µH 2.8A 81mOhm Shielded 20% Tolerance	Wurth	74438356056	1
L2	2.2u	2.2µH 3A 84mOhm Shielded 20% Tolerance	Toko	DFE252012P-2R2M=P2	1
L3, L4, L5, L6	n.m.	Shorted on board, no need to assemble			4
Q1		Nch+Pch MOSFET	ROHM	US6M2TR	1
Q2		Pch MOSFET, 100V, 13A	ROHM	RSD131P10TL	1
R1	43k	Resistor 43kOhm 1% 1/16W	ROHM	MCR01MZPF4302	1
R2	5.1k	Resistor 5.1kOhm 1% 1/16W	ROHM	MCR01MZPF5101	1
R3, R20	3.9k	Resistor 3.9kOhm 1% 0.5W	ROHM	MCR50JZHF3901	2
R4, R5	0.1	Resistor 0.1Ohm 1% 0.5W	ROHM	MCR50JZHFLR100	2
R6	62k	Resistor 62kOhm 1% 1/16W	ROHM	MCR01MZPF6202	1
R7	1.8k	Resistor 1.8kOhm 1% 1/16W	ROHM	MCR01MZPF1801	1
R8	39k	Resistor 39kOhm 1% 1/16W	ROHM	MCR01MZPF3902	1
R9	68k	Resistor 68kOhm 1% 1/16W	ROHM	MCR01MZPF6802	1
R10	12k	Resistor 12kOhm 1% 1/16W	ROHM	MCR01MZPF1202	1
R11	56k	Resistor 56kOhm 1% 1/16W	ROHM	MCR01MZPF5602	1
R12, R22	n.m.	open			2
R13, R21	0	Resistor short, 50m Ohm max, 0.5A max.	ROHM	MCR01MZPJ000	2
R14, R15	470	Resistor 470 Ohm 1% 1/16W	ROHM	MCR01MZPF4700	2
R19	20k		Bourns	3362P-1-203LF	1
TP0, TP1	n.m.				2
Z1, Z3		ZENER DIODE 4.7V 150mW	ROHM	EDZTE614.7B	2
Z2		TVS DIODE	Fairchild Semiconductor	SMCJ26A	1

Table 2: EVK Bill of Materials – STEPMO_EVK_202

4. Setup Instruction

4.1 Hardware

● **Master/Slave Mode**

This section describes how to set up the Shield in Master or Slave mode. This selection is made by Jumper J1. In most of the configurations (default) the Jumper J1 is simply left open and the Shield is configured as Master. Jumper J1 needs only to be in closed position when the Shield is stacked as a Slave Board on a Master Board for operation of two Shields with a single Arduino microcontroller. The Arduino microcontroller (and optional Slave Shield) will be powered by the Master Shield.

Note:
- **For the electrical operation the physical order of Master and Slave Shield actually does not matter.**
- **Do not change the setting of J1 while the power supply is applied.**

Master Board (Default)
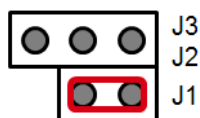
Jumper J1 is **open**:
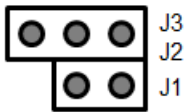
- Buck regulator supply (7V) is enabled to generate VIN for Arduino
- Green LED of D2 is lit
- Arduino IO mapping according to the following table

ALL Model EVK Versions except C, E						
ARDUINO IO	**Schematic**	**MUX Routing Between**		**Signal**	**Motor Driver IC Pin**	
No.	**Net Name**	**Pin**	**Pin**	**Name**	**No.**	**Name**
A1	PS1	IC1_X1	IC1_X	PS	14	PS
3	ENABLE1	IC1_Y1	IC1_Y	ENABLE	20	ENABLE
7	MODE11	IC1_W1	IC1_W	MODE1	19	MODE1
6	MODE01	IC1_Z1	IC1_Z	MODE0	18	MODE0
13	CW_CCW1	IC2_Y1	IC2_Y	CW_CCW	16	CW_CCW
10	CLK1	IC2_W1	IC2_W	CLK	15	CLK
Model EVK Versions C, E						
ARDUINO IO	**Schematic**	**MUX Routing Between**		**Signal**	**Motor Driver IC Pin**	
No.	**Net Name**	**Pin**	**Pin**	**Name**	**No.**	**Name**
A1	PS1	IC1_X1	IC1_X	PS	14	PS
3	ENABLE1	IC1_Y1	IC1_Y	ENABLE	20	I12
7	MODE11	IC1_W1	IC1_W	MODE1	19	I02
6	MODE01	IC1_Z1	IC1_Z	MODE0	18	PHASE2
11	TEST11	IC2_X1	IC2_X	TEST1	17	I11
13	CW_CCW1	IC2_Y1	IC2_Y	CW_CCW	16	I01
10	CLK1	IC2_W1	IC2_W	CLK	15	PHASE1

Table 3: Mapping of Arduino IOs to the motor driver IC pins **(Master Shield)**

Slave Board (for a stacked Shield)

Jumper J1 is **closed**:

- Buck regulator supply (7V) is disabled with high impedance output. VIN is delivered by Master Shield.
- Red LED of D2 is lit
- Arduino IO mapping according to the following table

ALL Model EVK Versions except C, E						
ARDUINO IO	**Schematic**	**MUX Routing Between**		**Signal**	**Motor Driver IC Pin**	
No.	**Net Name**	**Pin**	**Pin**	**Name**	**No.**	**Name**
A0	PS0	IC1_X0	IC1_X	PS	14	PS
2	ENABLE0	IC1_Y0	IC1_Y	ENABLE	20	ENABLE
4	MODE10	IC1_W0	IC1_W	MODE1	19	MODE1
5	MODE00	IC1_Z0	IC1_Z	MODE0	18	MODE0
12	CW_CCW0	IC2_Y0	IC2_Y	CW_CCW	16	CW_CCW
9	CLK0	IC2_W0	IC2_W	CLK	15	CLK
Model EVK Versions C, E						
ARDUINO IO	**Schematic**	**MUX Routing Between**		**Signal**	**Motor Driver IC Pin**	
No.	**Net Name**	**Pin**	**Pin**	**Name**	**No.**	**Name**
A0	PS0	IC1_X0	IC1_X	PS	14	PS
2	ENABLE0	IC1_Y0	IC1_Y	ENABLE	20	I12
4	MODE10	IC1_W0	IC1_W	MODE1	19	I02
5	MODE00	IC1_Z0	IC1_Z	MODE0	18	PHASE2
8	TEST10	IC2_X0	IC2_X	TEST1	17	I11
12	CW_CCW0	IC2_Y0	IC2_Y	CW_CCW	16	I01
9	CLK0	IC2_W0	IC2_W	CLK	15	PHASE1

Table 4: Mapping of Arduino IOs to the motor driver IC pins **(Slave Shield)**

Note:
- **In case a stacked Slave Shield is not used the IOs listed in Table 4 are free to use and can be accessed on the corresponding EVK pin header. However, the additional capacitive load of ~10pF by the turned-off multiplexer path should be considered in this case.**

● **Current Limitation Value**

The ROHM stepper motor driver ICs supported by this EVK have a current limitation function. This must not to be confused with over current protection (OCP) which is another feature of the IC (check datasheet for details). Instead, the purposes of the current limitation are:

- Constant motor current operation independent from the supply voltage
- Operation of motors with low impedance phase without exceeding the rated motor current
- Operation with high supply voltages for faster current rise in the phase windings to achieve higher motor torque
- Operation in microstepping mode

The current limitation is achieved by chopping the output current with pulse width modulation (PWM) as soon as the set limit is reached. As described in the datasheet the set current limit depends on the current sense resistor RNF (resistors R4 and R5 in the schematic) and the voltage applied to the VREF pin. In this EVK the current sense resistor is fixed but the voltage on the VREF pin can be adjusted by the potentiometer R19. The current limit value will depend linearly on the potentiometer setting. For reference, the approximate values for minimum and maximum current limit values are given in Table 5. With the corresponding values of VREF and RNF the set current limit value can be calculated with the formula:

$$IOUT\ (per\ phase) = \frac{VREF}{5 \cdot RNF}$$

VREF can be accessed at test point TP1 for indirect measurement of the current limit.

Note:
- **The highest value which can be set by the potentiometer is limited to the maximum continuous current per phase as allowed by the motor driver IC specifications.**
  Nevertheless, for high output currents (~1.5A and above) additional cooling or heat sinking will need to be applied to the IC and PCB. Please always check the IC temperature!
  Also please take care to set the current limit to a value not exceeding the rated maximum current per phase of your connected motor.

Potentiometer Setting	Minimum Setting		Maximum Setting		
EVK Model Versions	**VREF / V**	**Current Limit / A**	**VREF / V**	**Current Limit / A**	**RNF Value / Ω**
1, 4, 7, 9, C	0.11	~ 0.067	1.64	~ 1.0	0.33 [1]
F	0.16	~ 0.13	1.88	~ 1.5	0.25 [1]
5, A	0.1	~ 0.15	0.98	~ 1.5	0.13 [1]
2, 6, 8, E	0.11	~ 0.17	1.3	~ 2.0	0.13 [1]
3	0.25	~ 0.35	1.75	~ 2.5	0.14 [1]

Note 1: Including parasitic board resistance of ~0.03Ω

Table 5: Current limit settings by potentiometer R19 with references to VREF and RNF values

● **Current Decay Mode**

As explained in the datasheet the ROHM stepper motor driver IC used by this EVK allows external configuration of the current decay mode. It is a way to fine tune the motor performance between vibration and current waveform distortion/harmonics. The optimum decay mode setting depends very much on the application so the influence of each setting should be investigated by lab experiments.

For reference some general recommendations are given:

- Slow decay: Full step mode, low pulse rate half- and micro-stepping modes
- Fast decay: High pulse rate half- and micro-stepping modes
- Mixed decay: Trade-off

The EVK allows selecting the different decay modes with the Jumper J2/J3 as described in Table 6. In mixed decay mode it is also possible to apply an external voltage (not supported by model version F) to find the optimum setting.

Note:
- **The current decay setting is only effective when the IC is operating in current limitation mode. For details please refer to the according section in this document.**
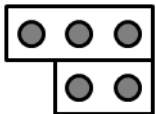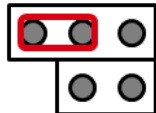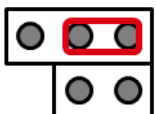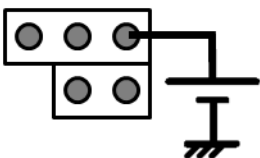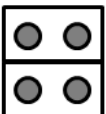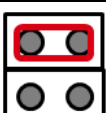
ALL Model EVK Versions except F	
	Jumper J2/J3 open: ▪ Fast Decay Mode (MTH Voltage ~ 2.5V)
	Jumper J2/J3 Pos. 1: ▪ Mixed Decay Mode (MTH Voltage ~ 0.75V)
	Jumper J2/J3 Pos. 2: ▪ Slow Decay Mode (MTH Voltage ~ 0V)
	Analog Input Voltage: ▪ Allows optimum tuning of current decay setting with analog voltage source ▪ Effective voltage range is ~0.4V to ~1.0V ▪ Voltage source should be able to drive load of ~30kΩ
Model Version F	
	Jumper J2/J3 open: ▪ Mixed Decay Mode (Fast Decay at 40%)
	Jumper J2/J3 closed: ▪ Slow Decay Mode

Table 6: Setting of Current Decay Mode

Some further explanations about the supported decay modes are summarized in Table 7.

MTH Voltage / V	Current Decay Mode	Explanation
0.0 – 0.3	Slow	Any voltage in the given range will result in the same "slow" decay mode. Typically just pull to GND to activate.
0.3 – 0.4	Transition region	Do not set the MTH voltage in this region because the resulting decay mode is not guaranteed.
0.4 – 1.0	Mixed	The mixture of "slow" and "fast" decay is not fixed in this region. The optimum setting can be tuned by applying an analog voltage to the MTH terminal. By jumper configuration (see EVK manual) just one predefined setting (~ 0.75V) is provided. If a different Mixed Mode MTH voltage is desired the voltage divider built by R9, R10, R11 must be adjusted.
1.0 – 1.5	Transition region	Do not set the MTH voltage in this region because the resulting decay mode is not guaranteed.
1.5 – 3.5	Fast	Any voltage in the given range will result in the same "fast" decay mode. Simply pull to 3.3V logic pin or use a 1:1 voltage divider for 5V logic.

Table 7: Current decay mode explanations

● Supply and Motor Connection

The STEPMO_EVK_20x allows the connection of a 4-wire bipolar or 5-wire unipolar stepper motor to the screw terminal J7/J8. The outputs V1A and V1B belong to one phase of the motor while V2A and V2B to the other. If the motor spins clockwise when it should go counter clockwise the polarity of one phase should be reversed, i.e. the wiring position of one phase should be swapped (either V1A with V1B OR V2A with V2B not both).

The connection of a bipolar motor is depicted in Figure 4. In case of a unipolar motor the additional common mode wire(s) are connected to the GND middle pin.

The power supply is connected to screw terminal J6. Although the EVK is equipped with a reverse power supply protection care should be taken about the polarity of the supply.

To meet the requirements from electromagnetic compatibility (EMC) all connected cables must not exceed a length of 3m.

Note:
- **In case of stacking two EVK Shields in Master/Slave configuration only the Master Shield must be supplied by the external power supply. The Slave Shield is powered via terminal J4 with the same applied voltage.**
- **When applying power to the Shield please double check the EVK model version you are using and take care to stay within the rated power supply limits as listed in Table 1. Operating the EVK over the given supply voltage ratings may permanently damage the EVK.**
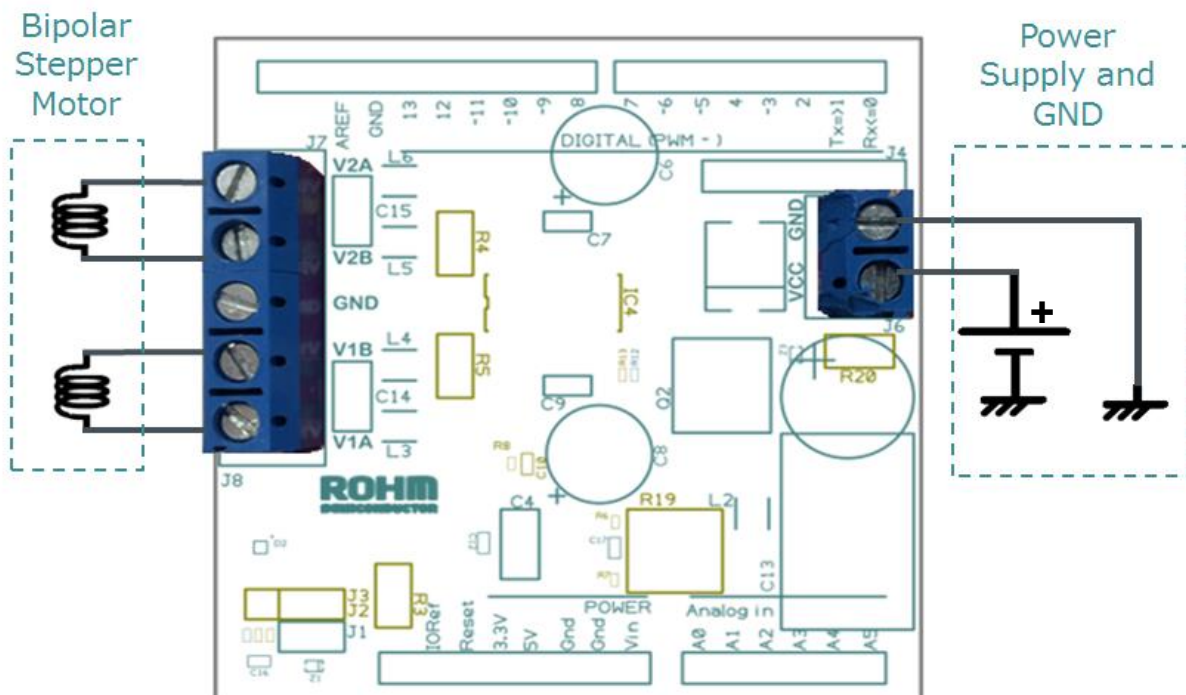

Figure 4: Connection of Stepper Motor and Power Supply

● **Connection to Microcontroller Motherboard**

Since the EVK is designed as Arduino Shield it can be directly plugged into an Arduino microcontroller motherboard such as the Arduino Uno. If the EVK is used with another microcontroller, all control signals and power supplies can be accessed / supplied via the pin header row interfaces as documented in the tables below.

-Pin-	Alias	Function in EVK			
		All Model Versions except C, E		**Models C, E**	
		1 Shield	**2 Shields**	**1 Shield**	**2 Shields**
0	RX	Reserved for Serial Communication via FTDI/USB			
1	TX	Reserved for Serial Communication via FTDI/USB			
2	IRQ0	Free to use	SH2_ENABLE	Free to use	SH2_I12
3	IRQ1	SH1_ENABLE		SH1_I12	
4	-	Free to use	SH2_MODE1	Free to use	SH2_I02
5	-	Free to use	SH2_MODE0	Free to use	SH2_PHASE2
6	-	SH1_MODE0		SH1_PHASE2	
7	-	SH1_MODE1		SH1_I02	
8	-	Free to use	Reserved	Free to use	SH2_I11
9	-	Free to use	SH2_CLK	Free to use	SH2_PHASE1
10	SS	SH1_CLK		SH1_PHASE1	
11	MOSI	Reserved		SH1_I11	
12	MISO	Free to use	SH2_CW_CCW	Free to use	SH2_I01
13	SCK	SH1_CW_CCW		SH1_I01	
14	A0	SH1_PS			
15	A1	Free to use	SH2_PS	Free to use	SH2_PS
16	A2	Free to use			
17	A3	Free to use			
18	A4	Free to use			
19	A5	Free to use			

Table 8: Arduino IO Mapping to EVK Function

-Pin-	Value	Function in EVK
Vin	7V	This pin provides a supply voltage generated by EVK from the applied motor voltage at screw terminal J6. This voltage is used as input voltage for the Arduino motherboard or any other microcontroller board which may be used. The maximum current drawn from this pin must not exceed 500mA (tbc).
5V	5V	This voltage is supplied by the Arduino motherboard to the EVK. In case of using another microcontroller board, please supply a regulated 5V voltage to this pin with a minimum current delivery capability of 10mA.
Gnd	0V	All pins labelled "Gnd" are connected to the ground (0V) of the EVK and are connected to the ground of the Arduino motherboard. In case of using another microcontroller board, please connect these pins to its ground terminal.

Table 9: EVK power supply interface to microcontroller

In operation, the Arduino motherboard is supplied by the motor voltage connected to the EVK so no additional power supply to the microcontroller board is necessary. However, in a typical lab set up the Arduino motherboard may be connected to the USB port of a host while evaluating, testing and programming. The following table gives an overview about the allowed power supply connections in this case if one or two EVKs are plugged into the microcontroller board.

EVK motor voltage	Additional Arduino supply via		
	USB	**Vin**	**DC Jack**
Turned off / not connected	Not recommended	Not allowed	Not allowed
Turned on	Allowed		

Table 10: Matrix of allowed additional Arduino supplies while EVK(s) plugged in

4.2 Software

● **Installation Procedure**

1. The latest Arduino Software (IDE) can be downloaded here:
    https://www.arduino.cc/en/Main/Software
    Please download and install it.

    Note: It is very important to use the latest IDE version. The provided software library and demo sketches have been tested with Arduino IDE Version 1.6.10. They will not run with older versions such as 1.6.5.

2. The latest ROHM STEPMO_EVK_20x software delivery package can be downloaded here:
    http://www.rohm.com/web/eu/arduino-stepper-motor-shield
    Please download it and unzip the package in the subfolder `Arduino\libraries\`.
    This folder is typically located at `C:\Program Files (x86)\`.
    After unzipping a folder called "ROHM_Steppers" is created. For contents of this folder please see the bullet point "Content of the Software Delivery Package".
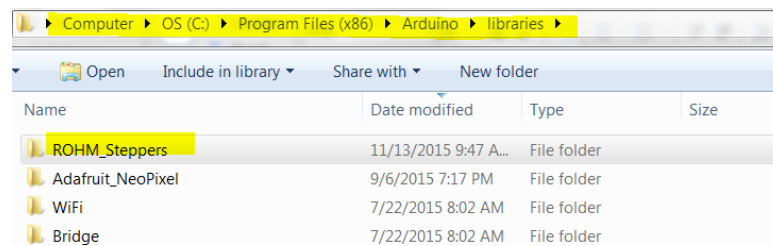

Figure 5: System path showing the location of the unzipped folder "ROHM_Steppers"

3. Run the Arduino IDE and open the provided ROHM example sketches.
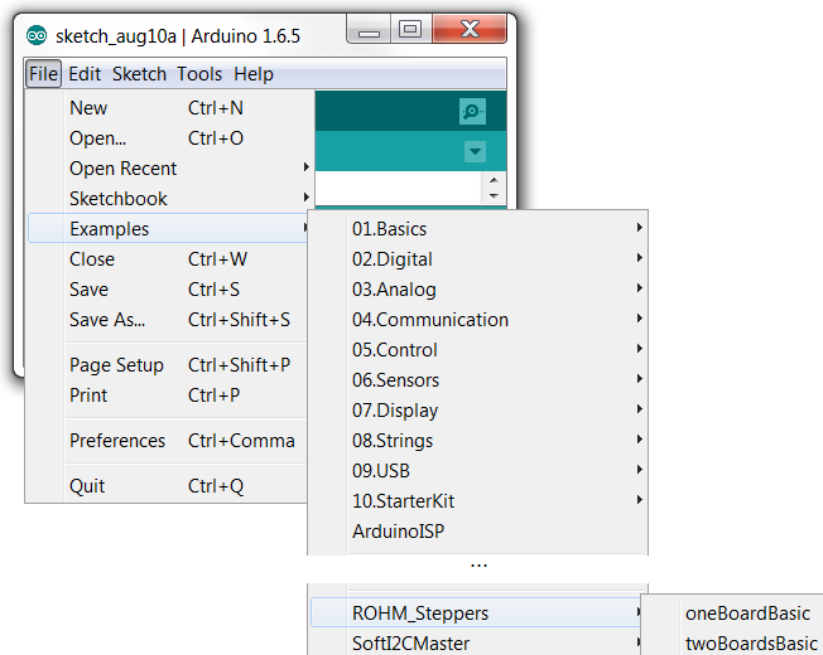   For more information about how to use the Arduino IDE please refer to
    https://www.arduino.cc/en/Guide/Environment

Figure 6: Arduino IDE with the "ROHM_Steppers" Menu Item and the available example programs

● **Content of the Software Delivery Package**

The ROHM STEPMO_EVK_20x software delivery package is part of this EVK. It contains the library with all required functions and also some example programs (Arduino Sketches) to demonstrate the usage of this library.
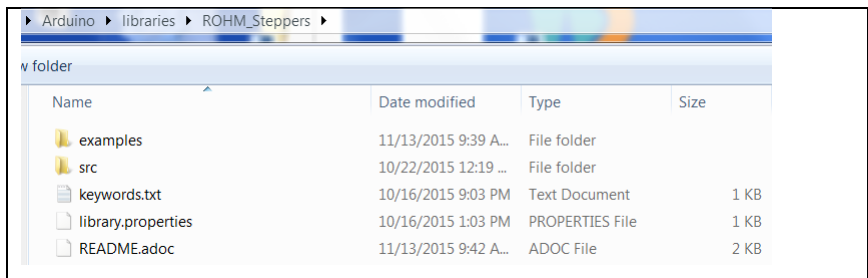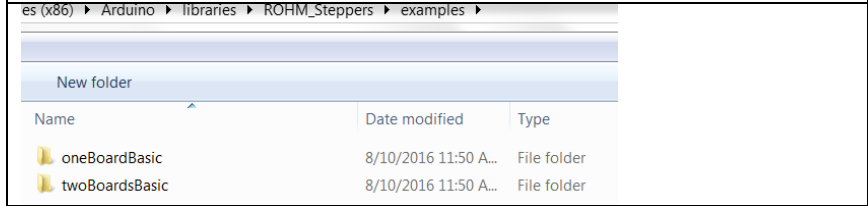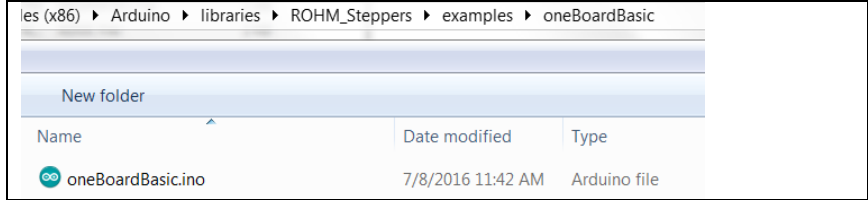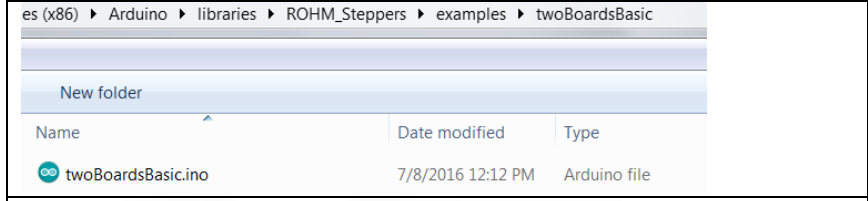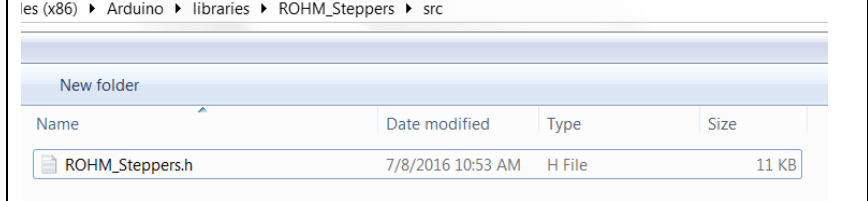
Arduino ▸ libraries ▸ ROHM_Steppers ▸ w folder Name — Date modified — Type — Size 📁 examples — 11/13/2015 9:39 A... — File folder 📁 src — 10/22/2015 12:19 ... — File folder 📄 keywords.txt — 10/16/2015 9:03 PM — Text Document — 1 KB 📄 library.properties — 10/16/2015 1:03 PM — PROPERTIES File — 1 KB 📄 README.adoc — 11/13/2015 9:42 A... — ADOC File — 2 KB	**Standard content for Arduino library**
es (x86) ▸ Arduino ▸ libraries ▸ ROHM_Steppers ▸ examples ▸ New folder Name — Date modified — Type 📁 oneBoardBasic — 8/10/2016 11:50 A... — File folder 📁 twoBoardsBasic — 8/10/2016 11:50 A... — File folder	**Arduino GUI accessible examples**
es (x86) ▸ Arduino ▸ libraries ▸ ROHM_Steppers ▸ examples ▸ oneBoardBasic New folder Name — Date modified — Type ∞ oneBoardBasic.ino — 7/8/2016 11:42 AM — Arduino file	**Example Sketch:** ▪ **One EV board connected. Demonstrating CW/CCW, clock speeds, full/half/micro stepping etc.**
es (x86) ▸ Arduino ▸ libraries ▸ ROHM_Steppers ▸ examples ▸ twoBoardsBasic New folder Name — Date modified — Type ∞ twoBoardsBasic.ino — 7/8/2016 12:12 PM — Arduino file	**Example Sketch:** ▪ **Two EV boards connected. Demonstrating CW/CCW, clock speeds, full/half/micro stepping etc.**
es (x86) ▸ Arduino ▸ libraries ▸ ROHM_Steppers ▸ src New folder Name — Date modified — Type — Size 📄 ROHM_Steppers.h — 7/8/2016 10:53 AM — H File — 11 KB	**Implementation of the library. It only needs a header file. A ".cpp" file is not used.**

Table 11: Contents of the ROHM STEPMO_EVK_20x software delivery package

● **DEMO Sketch**

```
/*
Basic Demo. Running BD63720AEFV in continous full step mode
*/

#define BD63720AEFV
#include <ROHM_Steppers.h>

ROHM_Stepper RS(ONE);

void setup()
{
    RS.ENABLE(ACTIVE);  // OPEN->ACTIVE
    RS.PS    (ACTIVE);  // RESET->ACTIVE
    delayMicroseconds(40); // wait 40us after PS->High as recommended in data sheet
    RS.MODE  (FULL_STEP);
    RS.CW_CCW(CCW);
}

void loop()
{
  RS.CLK(1);
}
```

● **Library function description / FAQ**

How to define the EVK version?
Before including the ROHM steppers library define which stepper motor IC your EVK model is using. E.g. in case of STEPMO_EVK_206 use
```
#define BD63720AEFV
```
See Table 1 for the list of different EVK models.

How to initialize the motor shield?
In the sketch before the setup() function the motor shield is initialized by instantiating ROHM_Stepper class
Example with using the instance name "RS":
```
ROHM_Stepper RS(BoardNumber);
```
BoardNumber depends on how many stacked boards you are using. Predefined values for BoardNumber: `ONE, TWO`
See sketch examples for references.

How to enable the motor?
The motor driver ICs have two pins (PS, ENABLE) which must be set on HIGH level to enable the motor. This can be controlled with the equivalent functions ENABLE(int en) and PS(int ps). For example use
```
[instancename].ENABLE(ACTIVE);
[instancename].PS(ACTIVE);
```
Predefined values as referenced in the IC datasheet are for en: `ACTIVE, OPEN`
and respectively for ps: `ACTIVE, RESET.`

Both functions are similar as when the IC is set to OPEN with the ENABLE function or to RESET with the PS function the motor driver outputs are set to high impedance and will not react on a clocking signal. The motor will no longer draw any current and thus lose its holding torque.
The difference is that the static current consumption of the IC in PS=RESET mode is reduced but also the position of the stepper motor is being re-initialized when setting PS=ACTIVE again.
If the driver was set to ENABLE=OPEN and then back to ACTIVE the motor driver outputs will go back to their latest values.

If the CLK() function is used directly after the PS(ACTIVE) command it is recommended to add a delay of 40us before clocking to fulfill the motor driver's timing requirements.

How to set up the stepping mode (full step, half step, etc)?
Use the method MODE(int mode). For example to set the stepper motor driver in quarter step mode use
```
[instancename].MODE(QUARTER_STEP);
```
Predefined values for mode: `FULL_STEP, HALF_STEP, HALF_STEP_A, HALF_STEP_B, QUARTER_STEP, EIGHTH_STEP, SIXTEENTH_STEP`
Default: `FULL_STEP`
Note that not every stepper motor driver IC supports every stepping mode. See Table 1 and the IC datasheet for the available modes.

How to run the motor for N steps
Use the method CLK(int clk) where clk is the number of steps.
Example1 (5 steps):
```
[instancename].CLK(5);
```
Example2 (5 steps):
```
for (i=0;i<5;i++)
[instancename].CLK(1);
```
Do not use negative values for clk. To change the direction use the method CW_CCW(int dir).

How to set clock speed and direction
Use the method CW_CCW(int dir) to change the direction of the stepper motor rotation from clockwise (CW) to counterclockwise (CCW). Example for CCW direction:
```
[instancename].CW_CCW(CCW);
```
Predefined values for dir: `CW, CCW`

There are two methods to change the stepping speed: setCLK_HP(unsigned int hp) and setCLK_Hz(unsigned int Hz), where the first sets the half period of the clock in microseconds and the second sets directly the clock frequency in Hz. Both methods are effectively equivalent. The default value for the half period is 1000us. This is equivalent to a clock speed of 500Hz. The stepper motor will perform on step on each clock cycle. To give an example a motor with 200 steps per revolution running in continuous full step mode will need 0.4s for one revolution by default.

Note: Do not set half periods longer than 16383us (or clock frequencies lower than ~30.5Hz) otherwise the speed will not be accurate. If you need lower speed use the delay() function in your sketch to add pauses after each step.
Also note that if the clock speed is set too high the motor may not step accurately or appear to be stuck. The maximum clock speed depends on parameters such as the motor specifications (current, phase inductance) and supply voltage etc. Please verify the maximum clock speed your motor can work with.

How to control two stepper motor driver EVKs at the same time

When the instance of ROHM_Stepper class is instantiated with the parameter "TWO" the methods as described in this FAQ are available with the suffixes "_M" and "_S". These will allow controlling the two shields individually.
See sketch examples for references.

N o t e s

This EVK is
- only to be used as device under test for research and development purposes.
- only to be operated by trained professionals.
- not to be assembled into a final product.

Additional Terms, warnings, restrictions and disclaimers of the ROHM Semiconductor STEPMO_EVK_20x (later defined as DESIGN KIT)

ROHM Semiconductor (later defined as ROHM) provides the enclosed DESIGN KIT under the following conditions: The user has to bear all responsibility and liability for the proper and safe handling with regard to this DESIGN KIT. The user shall indemnify ROHM from all claims arising from the handling or utilization of the DESIGN KIT. In the case this DESIGN KIT does not comply with the specifications indicated in the Manual, the DESIGN KIT may be returned within 30 days from the date of delivery for a full reimbursement of the purchase price.

THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY ROHM TO THE USER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES.

Please read the manual carefully prior to handling the DESIGN KIT. This Guide contains essential safety information regarding temperatures and voltages.

No license is granted under any patent right or other intellectual property rights of ROHM covering or relating to any machine, process and procedure, or combination in which such the DESIGN KIT or services might be or are used. Our arrangement with the user is not exclusive as ROHM is currently working with a large number of customers for DESIGN KITs. ROHM bears no liability for applications assistance, customer product design, software performance, or infringement of patents or services described in the Manual.

Code of federal regulations
As noted in the DESIGN KIT Manual, this DESIGN KIT and/or accompanying hardware may or may not be subject to and compliant with the Code of Federal Regulations, Title 47, Part 15.
For DESIGN KITs annotated to comply with the Code of Federal Regulations, Title 47, Part 15. Operation is subject to the following two conditions:
(1) This DESIGN KIT may not cause harmful interference, and (2) this DESIGN KIT must accept any interference received, including interference that may cause undesired operation. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. This DESIGN KIT as a Class A digital apparatus complies with Canadian ICES-003. Changes or modifications not expressly approved by the party responsible for compliance could void the users' authority to operate the equipment.
For DESIGN KITs annotated as not subject to or compliant with the Code of Federal Regulations, Title 47, Part 15. This DESIGN KIT is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY and is not considered by ROHM to be a finished end product fit for general consumer use. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to the Code of Federal Regulations, Title 47, Part 15, which are designed to provide reasonable protection against radio frequency interference. Operation of the equipment may cause interference with radio communications, in which case the user at its own expense will be required to take whatever measures may be required to correct this interference.

For Feasibility Evaluation Only, in Laboratory/Development Environments. The DESIGN KIT is not a complete product. It is intended exclusively for preliminary feasibility evaluation in laboratory/development environments by technically qualified electronics experts. Those experts mandatory have to be familiar with the dangers and application risks in connection with handling electrical mechanical components, systems and subsystems. It should not be used as an end product or as a part of an end product.

Your Sole Responsibility and Risk. You acknowledge, represent and agree that:
a) You have unique awareness of the Federal, State and local regulatory requirements (including but not limited to Food and Drug Administration regulations, if applicable) which affects your products and which refers to your use (and/or the use of your employees, affiliates, contractors or designees) of the DESIGN KIT for evaluation, testing and other purposes.
b) You are unlimited and exclusive responsible for the safety of your DESIGN KIT and for the compliance with all relevant laws and other applicable regulatory requirements. Further you have to assure the safety of any activities to be conducted by you and/or your employees, affiliates, contractors or designees, using the DESIGN KIT. You are also responsible to ensure that any interfaces (electronic and/or mechanical) between the DESIGN KIT and any human body are designed with suitable isolation and means to safely limit the accessible leakage currents to minimize the risk of electrical shock hazard.
c) Since the DESIGN KIT is not a completed product, it may not meet all applicable regulatory and safety compliance standards which may normally be associated with similar completed products. You assume full responsibility to determine and/or assure compliance with any such standards and related certifications as may be applicable. You have to use reasonable safeguards to ensure that your use of the DESIGN KIT will not result in any property damage, injury or death, even if the DESIGN KIT should fail to perform as specified or expected.

Certain Instructions. It is important to handle this DESIGN KIT within ROHM's recommended specifications and environmental considerations as described in the Manual. Surpassing the specified DESIGN KIT classifications (including but not limited to input and output voltage, current, power, and environmental ranges) may cause property damage, personal injury or death. If there are questions concerning these classifications please contact a ROHM external sales representative before connecting interface electronics including input power and intended loads. Any loads applied beyond the specified output range may result in unintended and/or inexact operation and/or possible lasting damage to the DESIGN KIT and/or interface electronics. Please consult the DESIGN KIT Manual prior to connecting any load to the DESIGN KIT output. If there is uncertainty regarding the load specification, please contact a ROHM external sales representative. During normal operation, some circuit components may have case temperatures greater than 60°C as long as the input and output are maintained at a normal ambient operating temperature. These components include but are not limited to linear regulators, switching transistors, pass transistors, and current sense resistors which can be identified by using the DESIGN KIT schematic published in the DESIGN KIT Manual.
Please be aware that the devices of the DESIGN KIT may be very warm in case of placing the measurement test setup close to the DESIGN KIT during normal procedure. Please ensure that only qualified personnel educated in electronic measurement and diagnostics usually found in development environments should use these DESIGN KITs.

Agreement to Defend, Indemnify and Hold Harmless. You agree to defend, indemnify and hold ROHM, its licensors and their representatives harmless from and against any and all claims, damages, losses, expenses, costs and liabilities (collectively, „Claims") arising out of or in connection with any use of the DESIGN KIT that is not in accordance with the terms of the agreement. This obligation shall apply whether Claims arise under law of tort or contract or any other legal theory, and even if the DESIGN KIT fails to perform as specified or expected.

Safety-Critical or Life-Critical Applications. If you intend to evaluate the components for possible use in safety critical applications (such as life support) where a failure of the ROHM product would reasonably be expected to cause severe personal injury or death,
such as devices which are classified as FDA Class III or similar classification, you have to specifically notify ROHM of such intent and enter into a separate Assurance and Indemnity Agreement.

Thank you for your accessing to ROHM product information.
More detailed product information and catalogs are available, please contact us.

ROHM Customer Support System

http://www.rohm.com/contact/

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for Power Management IC Development Tools *category:*

Click to view products by ROHM *manufacturer:*

Other Similar products are found below :

EVAL-ADM1168LQEBZ EVB-EP5348UI MIC23451-AAAYFL EV MIC5281YMME EV DA9063-EVAL ADP122-3.3-EVALZ ADP130-0.8-EVALZ ADP130-1.2-EVALZ ADP130-1.5-EVALZ ADP130-1.8-EVALZ ADP1712-3.3-EVALZ ADP1714-3.3-EVALZ ADP1715-3.3-EVALZ ADP1716-2.5-EVALZ ADP1740-1.5-EVALZ ADP1752-1.5-EVALZ ADP1828LC-EVALZ ADP1870-0.3-EVALZ ADP1871-0.6-EVALZ ADP1873-0.6-EVALZ ADP1874-0.3-EVALZ ADP1882-1.0-EVALZ ADP199CB-EVALZ ADP2102-1.25-EVALZ ADP2102-1.875EVALZ ADP2102-1.8-EVALZ ADP2102-2-EVALZ ADP2102-3-EVALZ ADP2102-4-EVALZ ADP2106-1.8-EVALZ ADP2147CB-110EVALZ AS3606-DB BQ24010EVM BQ24075TEVM BQ24155EVM BQ24157EVM-697 BQ24160EVM-742 BQ24296MEVM-655 BQ25010EVM BQ3055EVM NCV891330PD50GEVB ISLUSBI2CKIT1Z LM2744EVAL LM2854EVAL LM3658SD-AEV/NOPB LM3658SDEV/NOPB LM3691TL-1.8EV/NOPB LM4510SDEV/NOPB LM5033SD-EVAL LP38512TS-1.8EV