Power PROFET™ + 12V Shield for Arduino

User guide

About this document

Scope and purpose

This document describes how to use the high-side switch Shield for Arduino with the Power PROFET™ + 12V.

Intended audience

Engineers, hobbyists and students who want to use powerful protected high-side switches for Arduino projects.

Table of contents

1 Getting started

The BTS500xx-1LUA Shield for Arduino from Infineon Technologies is a flexible evaluation board dedicated to drive loads. This board is compatible with Infineon's XMC1100 Boot Kit, the Arduino UNO, the Arduino DUE and all the other boards which comply with the Arduino form factor.

The shield can be controlled either with the general logic IO-Ports of a microcontroller or with a PWM signal generated by a waveform generator.

The board is designed with the typical circuit to control one BTS50005-1LUA and one BTS50010-1LUA. This shield offers a quick evaluation of features like the analog current feedback and all the different protection functions.

1.1 Shield overview

The **Figure 1** shows and describes the board. With this board, users are able to drive loads by adding power connectors on BAT, OUT1 and OUT2. All the pins of the two devices are accessible by test points. If a user does not want to use a microcontroller, it is possible to connect the inputs of the two devices to a low frequency generator and do the measurements with an oscilloscope. The demoboard can also be easily connected to any Arduino UNO board or Infineon board which has the Arduino UNO form factor through headers.

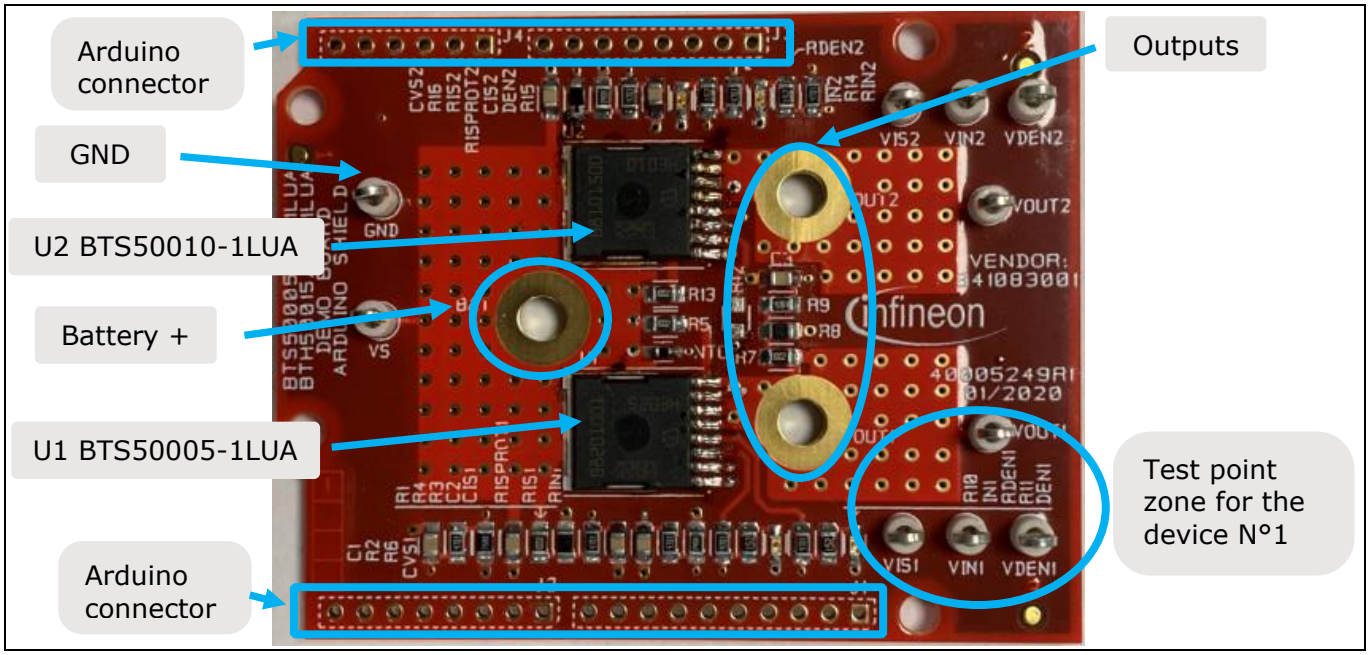

Figure 1 Arduino compatible evaluation board for the Power PROFET™ + 12V

This shield offers a quick evaluation of the product:

- the latch functionality
- the current sensing functionality
- and all protections, e.g. "over temperature shut down" and "over current shut down"

WARNING: Refer to the datasheet for details on functionalities and parameters values. This user guide does not replace the datasheet and the user must be aware of limitations before turning on any supply.

1.2 Typical connection

In **Figure 2**, the evaluation board is connected to an Arduino UNO board, this pinout connection works with all microcontrollers using an Arduino pinout.

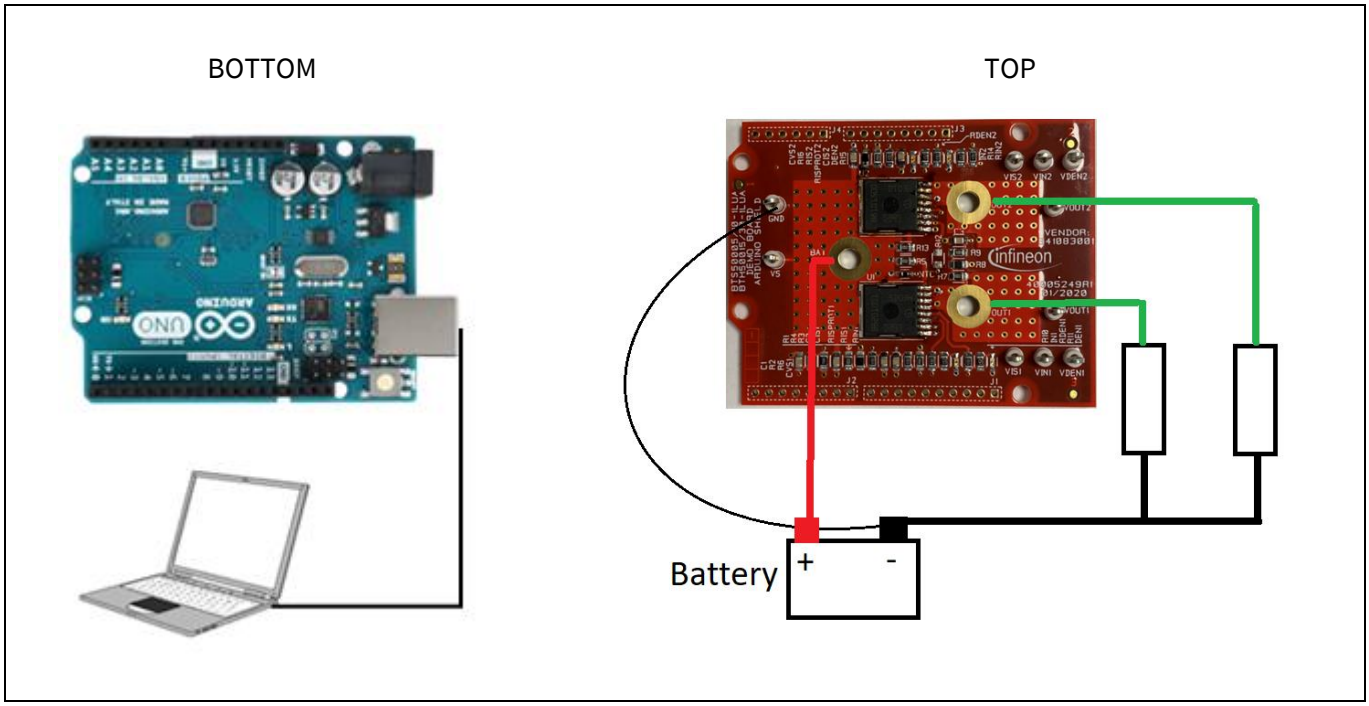

Figure 2 Evaluation board connected to the Arduino UNO board

In **Figure 3**, the evaluation board is connected to a waveform generator, driving the IN and DEN pin.

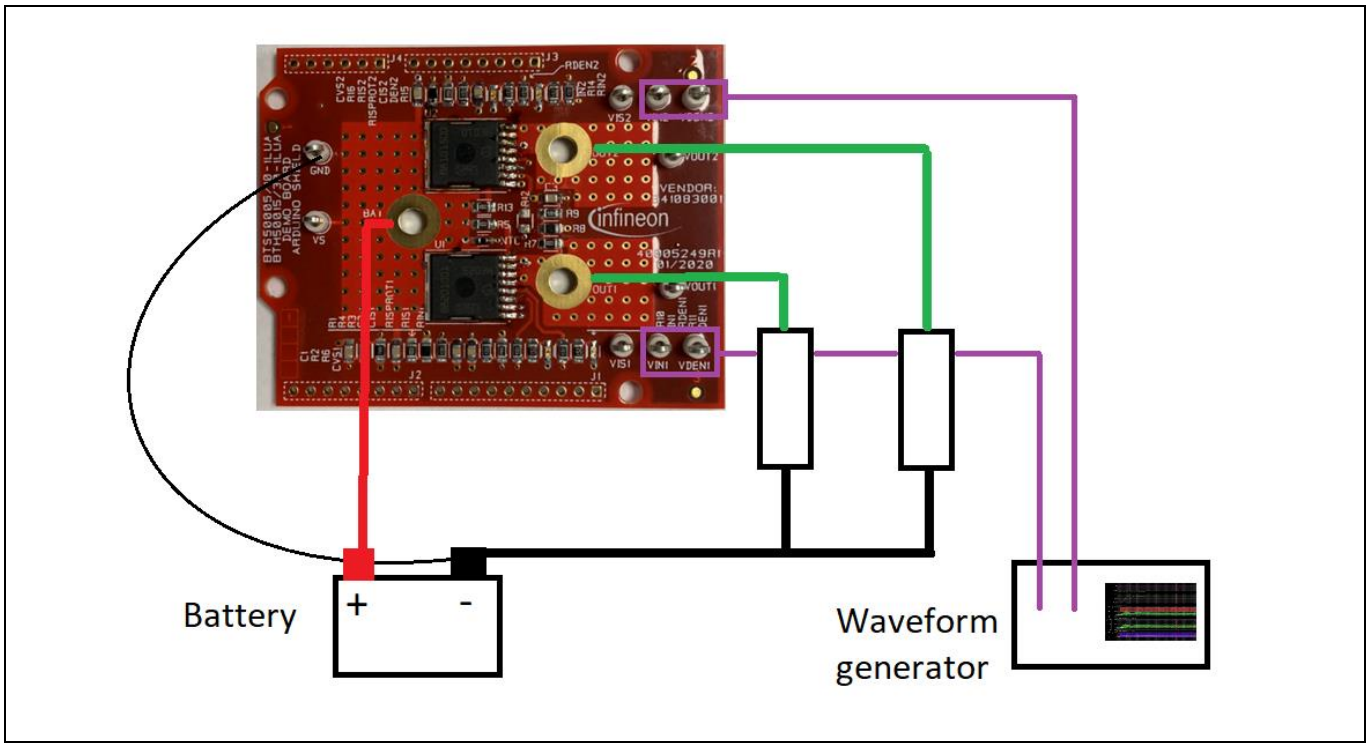

Figure 3 Evaluation board connected to a waveform generator

1.3 Key features

- The demoboard is able to provide continuous load current (12V-50A) between V_OUT and BAT.
- A green LED will turn on when the logic supply voltage is connected and turned on.
- A red LED will turn on when the logic supply voltage is connected and only if the refered device is latched.
- Output voltage, battery voltage, input and DEN logic, output current sense IS voltage can be measured externaly with test points.
- Output voltage, battery voltage, input and DEN logic, output current sense IS can be monitored with the Shield for Arduino and also throught the GitHub library.
- The board temperature can be measured with an NTC, which is populated on the board.

2 How to drive the evaluation board

2.1 Without any Arduino board connected

Users can connect a waveform generator in order to drive the evaluation board manually. The IN pin has to be driven to turn the device on. The DEN pin has to be driven to enable the diagnosis ability.

2.2 Using a microcontroller

Users are able to connect any microcontroller which is compliant with the Arduino UNO form factor. Source code for microcontroller can be written and implemented according the schematic and the pinout of the evaluation board provided in chapter **3**.

2.3 With the GitHub library

Users have the possibility to use the Power PROFET™ + 12V Shield for Arduino GitHub library. The procedure is described step by step below.

1- Open the Arduino IDE.

2- Select “Manage library”, search for “High-Side-Switch” and install it.

3- Choose an example inside the “High-Side-Switch” library. For the Power PROFET™ + 12V, there are three examples. The first one is about switching one single channel, the second is about switching the two channels and the third is about switching the two channels separately and measuring the available parameters like the battery voltage, the temperature and the current sense voltage of the two channels.

Many functions are available with the GitHub library. All functions are described below.

The source file can be found following this link: https://github.com/Infineon/high-side-switch

Public Functions

Bts500xxShield(Hss *sw1, Hss *sw2, ADCPAL *vs, ADCPAL *vOut, ADCPAL *temp, BtxVariants_t *btxVariant1, BtxVariants_t *btxVariant2)

Power PROFET ™ +12V BTS500xx Shield constructor.

Parameters

- **sw1** – **[in]** High-side switch instance 1
- **sw2** – **[in]** High-side switch instance 2
- **vs** – **[in]** ADC instance to measure the supply voltage
- **vOut** – **[in]** ADC instance to measure the output voltage of the high-side switch 1
- **temp** – **[in]** ADC instance to measure the temperature of the PCB

~Bts500xxShield()

Destructor of the PROFET +12V BTS500xx Shield.

***virtual* Error_t init()**

Initialize all necessary objects of the high-side switch board. This function initializes all necessary objects of the high-side switch board. It returns an error code to see if everything was initialized correctly.

Returns: hss::Error_t

***virtual* Error_t deinit()**

Deinitialize all necessary objects of the high-side switch board. This function de-initializes all necessary objects of the high-side switch board. It returns an error code to see if everything was de-initialized correctly.

Returns: hss::Error_t

***virtual* Error_t switchHxOn(uint8_t x)**

Switch on the selected high-side switch. This function turns on the desired high-side switch.

Parameters

x – **[in]** Number of the switch that should be turned on (1-2). Can be either SWITCH1 or SWITCH2.

Returns: hss::Error_t

***virtual* Error_t switchHxOff(uint8_t x)**

Switch off the selected high-side switch. This function turns off the desired high-side switch.

Parameters

x – **[in]** Number of the switch the should be turned off (1-2) Can be either SWITCH1 or SWITCH2.

Returns: hss::Error_t

***virtual* DiagStatus_t readDiagx(uint8_t x)**

Read the diagnosis ouput of the chosen switch. This function uses the current signal of the chosen switch to diagnose the switch. It returns the different states depending on the switch's condition.

Parameters

x – **[in]** Desired switch for the diagnosis (1-2) Can be either SWITCH1 or SWITCH2.

Returns: DiagStatus_t

Returns

- **-1** – Diagnosis read error (chosen channel not valid)
- **0** – Switch is working fine
- **2** – Short circuit to ground
- **4** – Short circuit to Vs

- **5** – Open load detected

virtual float readIsx(uint8_t x)

Read the desired current value of the chosen switch. This function reads the IS pin of the chosen high-side switch and calculates the current which is flowing through the switch with the acquired ADC value.
Parameters: x – **[in]** Number of the desired switch (1-2)
Returns: The value of the current in [A]
Returns: -1.2345 – Invalid channel was chosen

Error_t switchesHxOn(bool h1 = *false*, bool h2 = *false*)

Switches on high-side switches. This function can switch on several high-side switches at once, but is also possible two only switch on the first or second one.
Parameters

- **h1** – **[in]** Bool value for the first switch.*true* means turn on, *false* means keep it as is.
- **h2** – **[in]** Bool value for the second switch.*true* means turn on, *false* means keep it as is.

Returns: hss::Error_t

Error_t switchesHxOff(bool h1 = *false*, bool h2 = *false*)

Switches off high-side switches. This function can switch off several high-side switches at once. But is also possible two only switch off the first or second one.

Parameters

- **h1** – **[in]** Bool value for the first switch.*true* means turn off, *false* means keep it as is.
- **h2** – **[in]** Bool value for the second switch.*true* means turn off, *false* means keep it as is.

Returns: hss::Error_t

float readVs()

Read the supply voltage. This function is reading the supply voltage of the high-side switch board.
Returns: Value of the supply voltage in [V]

float readVOut()

Read the output voltage. This function is reading the output voltage of high-side switch one.
Returns: Value of the output voltage in [V]

float readTemperature()

Read the PCB temperature. This function is reading the temperature of the high-side switch board PCB.
Returns: Value of the temperature in [°C]

3 Board connectors description

3.1 Power connectors

Name	Type	Description
BAT	Power supply	Positive power supply 12V
GND	Ground power	Ground
OUT1	Power output	Output 1
OUT2	Power output	Output 2

3.2 Arduino connector

Connector J1

Name	Pin	Type	Description / Arduino pin name
-	1	No connected	-
-	2	No connected	-
-	3	No connected	-
Gnd	4	Digital Ground	Ground
-	5	No connected	-
IN1	6	Digital Input	IN activation for U1 / Digital 12
DEN1	7	Digital Input	DEN activation for U1 / Digital 11
IN2	8	Digital Input	IN activation for U2 / Digital 10
DEN23	9	Digital Input	DEN activation for U2 / Digital 9
-	10	No connected	-

Connector J2

Name	Pin	Type	Description
-	1	No connected	-
-	2	No connected	-
-	3	No connected	-
-	4	No connected	-
-	5	No connected	-
-	6	No connected	-
-	7	No connected	-
-	8	No connected	-

Connector J3

Name	Pin	Type	Description
-	1	No connected	-
-	2	No connected	-
-	3	No connected	-
-	4	No connected	-
-	5	No connected	-
Gnd	6	Digital Ground	Ground
Gnd	7	Digital Ground	Ground
-	8	No connected	-

Connector J4

Name	Pin	Type	Description/ Arduino pin name
IS_adc1	1	Analog input	Analog measurement for the current sense U1 / A0
VS_adc	2	Analog input	Analog measurement for the Vs voltage 2 / A1
Temp_adc	3	Analog input	Analog measurement for the temperature / A2
VOUT_adc	4	Analog input	Analog measurement for the output voltage U2 / A3
IS_adc2	5	Analog input	Analog measurement for the current sense U2 / A4
-	6	No connected	-

Test points

Name	Type	Description
VIN1	Digital Input	IN activation for U1
VIN2	Digital Input	IN activation for U2
VDEN1	Digital Input	DEN activation for U1
VDEN2	Digital Input	DEN activation for U2
VIS1	Analog output	Analog measurement for the current sense U1
VIS2	Analog output	Analog measurement for the current sense U2
VS	Analog output	Analog measurement for the VS voltage
Out1	Analog power output	Analog power output from U1
Out2	Analog power output	Analog power output from U2

3.3 Schematic

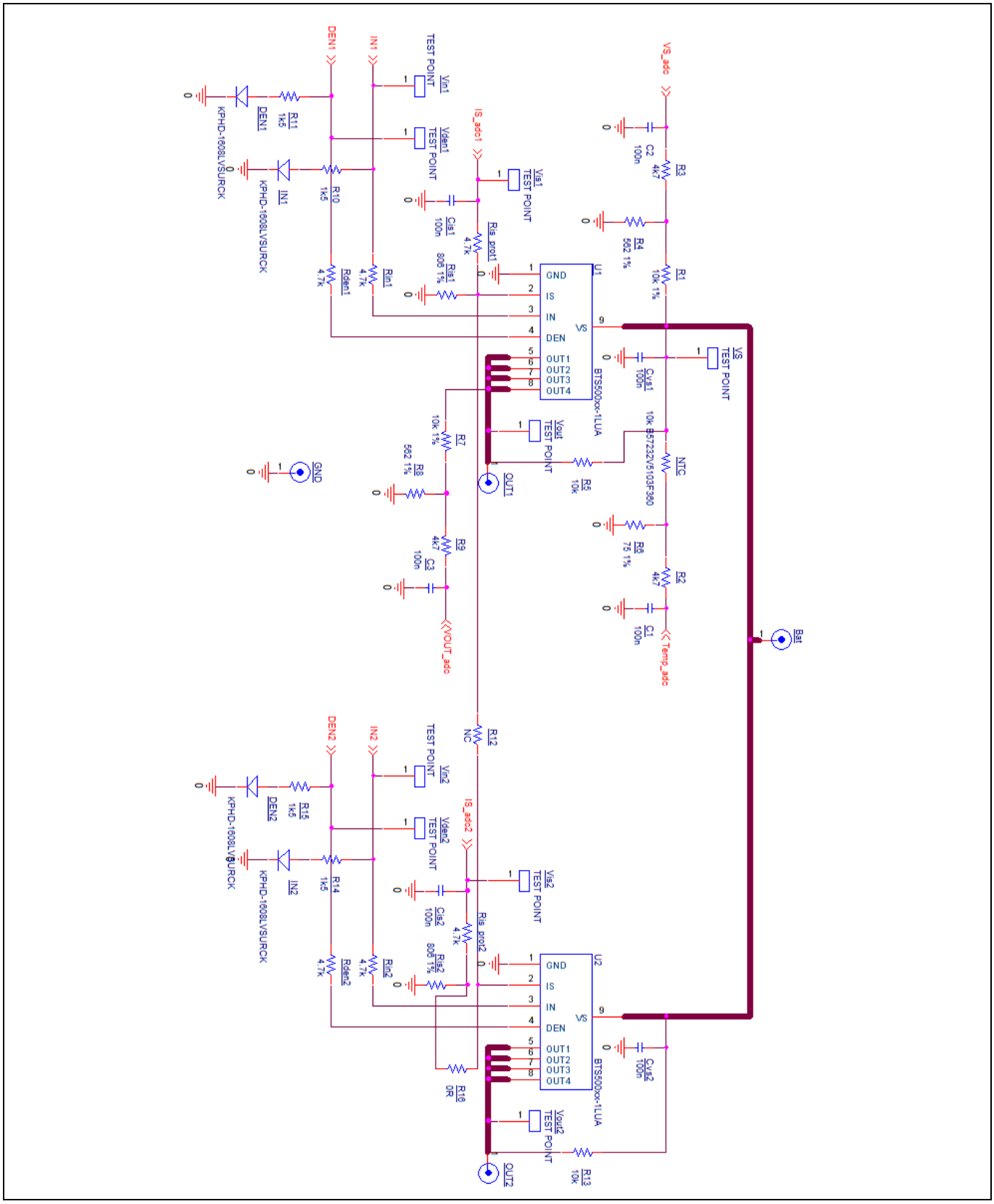

Figure 4 Schematic (U1 is the BTS50005-1LUA, U2 is the BTS50010-1LUA)

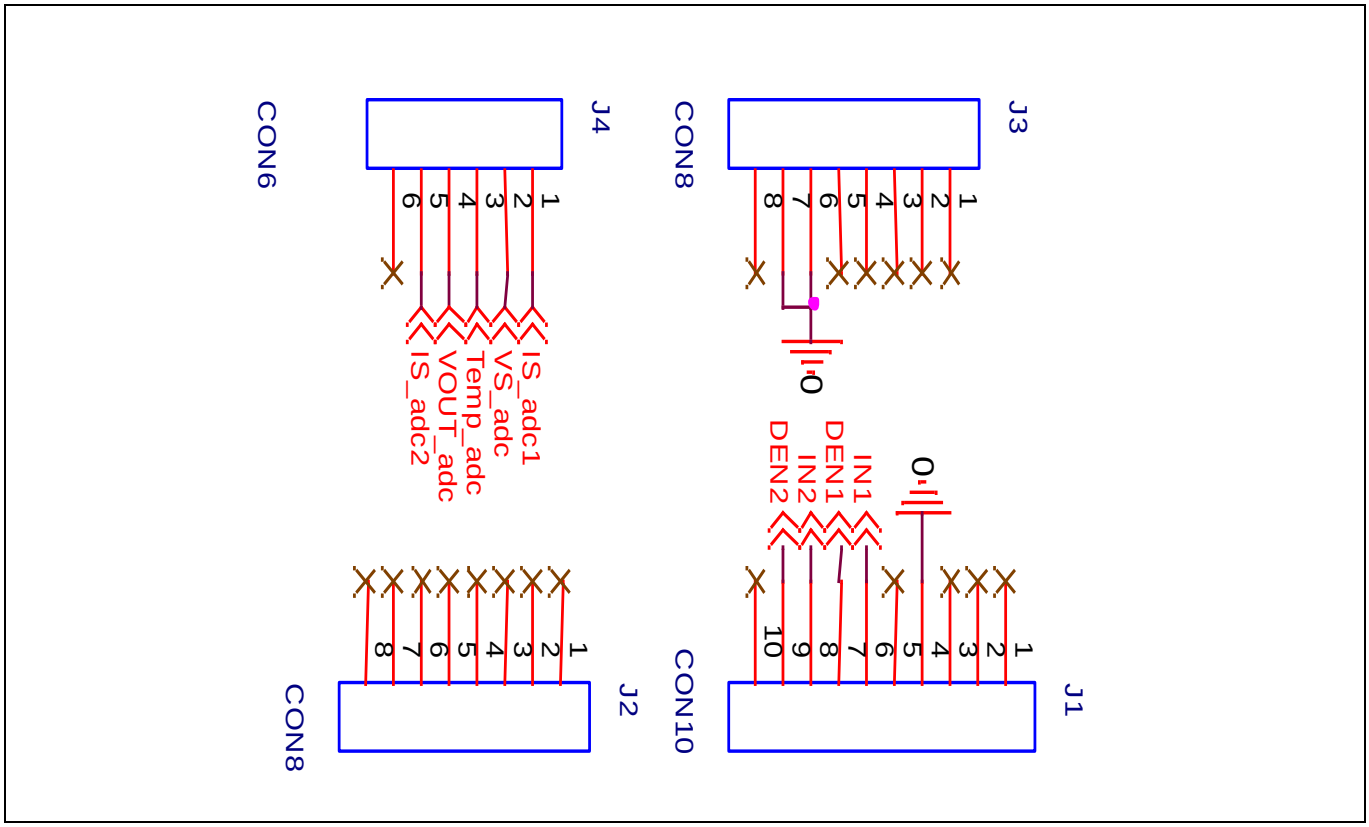

Figure 5 ARDUINO pin compatibility

Revision history

Document revision	Date	Description of changes
1.00	2023-01-30	Initial version released
1.10	2025-05-13	- ARDUINO pin compatibility added - Title updated by “Shield for Arduino” - Designation between U1 and U2 done

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

Edition 2025-05-13

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

Z8F80420456

Important notice

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.