HiLExtension 37IO

User Manual

Document revision: 1.0
Document release date: 25.07.2020
Notes:

Mitkov Systems GmbH
+ 49 (0) 6162 9899544

Friedrichstraße 41
D-64354 Reinheim

mail(at)mitkov-systems.de
www.mitkov-systems.de

Table of Contents

List of Figures

List of Tables

1. Introduction

This user manual serves as a reference point for the application and usage of HiLExtension 37IO as part of the HIL testing system. The document focuses on the main characteristics of the product, installation routine, software interaction and hardware implementation methods. For any further assistance, technical support or information concerning other products, please feel free to contact Mitkov Systems GmbH.

2. General information

2.1. Product overview

HiLExtension 37IO (Figure 1.) is a multi-functional measuring tool. It is designed to contribute to proof of concept during the developing and testing of a product. HiLExtension 37IO is the link between the device under test (DUT) and the hardware in the loop (HIL) testing system. It simulates possible scenarios before the product's final release. As part of the HIL testing system, HiLExtension 37IO is placed in a compact D-SUB connector with a 37-pin female pin-out. Supply voltage and PC communication is delivered by a standard Micro USB Type B interface. HiLExtension 37IO can be configured and controlled by a set of commands transmitted via serial communication. The interaction with the device can also be established by the graphic UI software called HiL Simplicity.

Figure 1. HiLExtension 37IO

2.2. Technical data

Table 1: HiLExtension 37IO specifications

Technical data	
Enclosure dimensions	69.3 x 55 x 24 mm
Degree of protection	IP30
Operating temperature range	-20 °C to +70 °C
Connection method of signals	D-SUB 37 pin female connector
Computer connection method	Micro USB Type B receptacle
Rated supply voltage	4.5 ~ 5.5 VDC (powered by USB bus, max. 200mA)
Digital characteristics	
Digital I/O channels	24 channels divided in 3 ports (A, B, C) x 8 channels each. Port C can be divided in 2 sub-ports by 4 channels each Each channel is bidirectional input/ output.
Output current	50mA for every group of 8 channels
Logic level	Low 0 ~ 0.8 V, High 3.3 ~ 5 V
ESD Protection	± 8kV contact discharge, ± 6kV air gap discharge
Analog characteristics	
Analog inputs	8 single-ended channels
Parameters:	-
• Input voltage range	0 ~ 3.3 V
• Resolution	12 bits
• Accuracy	±3.46 mV
• Sampling rate	Max.1 MHz
• ESD Protection	8kV contact discharge, 15kV air gap discharge
Analog outputs	2
• Output voltage range	0 ~ 3.3 V
• Output current	16 mA
• Resolution	12 bits
• Accuracy	±13.16 mV
• ESD Protection	±1500 V (HBM), ±100 V machine model
• Update rate	Max 1 MS/s
Programmable features	
Digital input	Low/ High level sense

Digital output	Steady state, Pulse generation[1], Invert signaling[1]
Analog inputs	Single-ended analog input read
Analog outputs	Independent voltage level set
Device interaction	
Option 1	User defined via command set and terminal interface
Option 2	HiLSimplicity Software (Linux/ Windows compatible)

2.3. IO distribution

Under the robust case of HiLExtension 37IO, there is a 32-bit Arm® Cortex® - M3 microcontroller which allows access to 24 bidirectional digital IOs, 8 analog inputs and 2 analog outputs. The figure and table below show the pin assignment of the 37-pin D-SUB female connector by which the digital/analog lines are accessible.

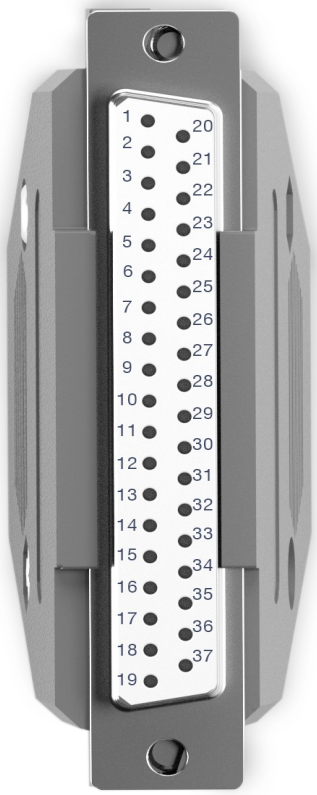

Figure 2. Pin assignment of the 37-pin female connector

[1] The functionality is possible only in HiL Simplicity software. Please contact Mitkov Systems GmbH for additional information.

Table 2: 37-pin female connector pin-out description

Pin №	Function/Port/№	Pin №	Function/Port/№
1	DIO A 00	20	DIO A 01
2	DIO A 02	21	DIO A 03
3	DIO A 04	22	DIO A 05
4	DIO A 06	23	DIO A 07
5	DIO B 00	24	DIO B 01
6	DIO B 02	25	DIO B 03
7	DIO B 04	26	DIO B 05
8	DIO B 06	27	DIO B 07
9	DIO C 00	28	DIO C 01
10	DIO C 02	29	DIO C 03
11	DIO C 04	30	DIO C 05
12	DIO C 06	31	DIO C 07
13	GND	32	AIN 00
14	AIN 01	33	AIN 02
15	AIN 03	34	AIN 04
16	AIN 05	35	AIN 06
17	AIN 07	36	GND
18	AOUT 1	37	AOUT 2
19	+5V	-	-

3. Getting started with HiLExtension 37IO

3.1. Connecting to PC

3.1.1. Step 1 Physical connection - HiLExtension 37IO is connected to the PC's USB 2.0 port via a standard Micro-B USB / USB A plug cable. The Micro-B USB socket is located at the backside of the device. Please use the provided cable to establish a physical connection.

Figure 3. HiLExtension 37IO backside

Caution!

The power supply of the device is provided by the USB port via the connection cable. Thus, HiLExtension 37IO requires a +5V operating voltage! The user may be required to make an additional configuration of their operating system to meet the supply requirements.

3.1.2. Step 2 Driver integration Once the device is connected to the computer system, it will be visible as a Virtual COM port (VCP) device. HiLExtension 37IO is supported as follows:

- Windows® 's versions: 98SE, 2000, XP, Vista®, 7, and 8.x – requires installation of STM32 Virtual COM Port Driver (STSW-STM32102). The driver package and installation instructions are available for download at ST Microelectronics' website: www.st.com
- Windows® 10 - no need of extra driver installation

- Linux – no need of extra driver installation

3.1.3. Step 3 Device recognition - In order to check if the HiLExtension 37IO is properly recognized by the operating system, the following must be done:

- for Windows® OS go to Device Manager→ Select the tab “Ports (COM&LPT)”→ look for “STMicroelectronics Virtual COM Port (COM*x*)”, where *x* is the number of the corresponding COM port associated with that particular device.
- for Linux OS open Terminal window→ type in “lsusb” and press “Enter”→ look for “STMicroelectronics STM32F407” preceded by the device’s unique ID.

Note: If two or more devices are connected to the system simultaneously, they will differ in associated COM port number or device ID.

3.2. Communication via serial terminal tool

The user’s interaction with HiLExtension 37IO is established by a serial terminal tool which sends a special set of commands and receives a response from the device.

The examples below are created and tested by Realterm terminal software. Therefore, the following default serial interface settings must be applied beforehand in order to achieve proper communication:

- Baud rate: 115200
- Data bits: 8
- Stop bits: 1
- Parity: None
- Set device’s port number
- Set data displaying as hexadecimal numbers with space

When Realterm is used, the sent command string must be composed as a series of space separated hexadecimal numbers with prefix “0x” and transmitted to the device as numbers.

Example1: String for setting DIOA 00 in logical HIGH level
Send:”0x81 0x00 0x01”
Answer:”0x81 0x00 0x01

In the above-mentioned example, the denotation of the transmitted string is:

- “0x81” → code of the target operation
- “0x00 0x01” → value to be written (recorded number)

The received response is a repetition of the sent command by the device.

4. Interaction with HiLExtension 37IO via set of commands

4.1. Types of commands

4.1.1. Configuration type – used for configurating the digital I/O ports direction

4.1.2. Writing type – used for setting the digital output's channel state or analog outputs voltage level

4.1.3. Reading type - used for reading the values of the actual digital/analog inputs channels

4.2. Commands: possible use cases

4.2.1. Digital inputs

In order to use the digital inputs of the HiLExtension 37IO, the user must observe the following routine:

1. Send configuration command to set the target port direction as input
2. Send reading command in order to check the selected port channels' state
3. Convert the received hexadecimal number to binary and check with inputs channels' read map (refer to Table 8 or 9 at section 4.5)

Note: Port C offers an extra functionality that allows to control its channels' direction as a whole port (DIOC [0...7]) or as two independent sub-ports - C_{LOW} (DIOC [0...3]) and C_{HIGH}(DIOC [4...7]).

4.2.2. Digital outputs

In order to use the digital outputs of HiLExtension 37IO, the user must observe the following routine:

1. Send configuration command to set the target port direction as output
2. Send writing command for the desired port's channel state (refer to section 4.4)

4.2.3. Analog inputs

Reading of the analog inputs of HiLExtension 37IO requires the following steps:

1. Send a command to read the desired analog input AIN[00...07] (refer to Table 7)
2. Pay attention to the adapter's response and only note the hexadecimal number from the last 2 bytes. Then convert it to decimal value and calculate the input voltage level (refer to section 4.5)

4.2.4. Analog outputs

Setting the analog outputs of HiLExtension 37IO requires the following steps:

1. Calculate the value of the target output voltage level which has to be recorded (refer to section 4.4)
2. Convert the calculated decimal value to hexadecimal number
3. Send writing command to implement the target voltage level

4.3. Configuration commands

Table 3: List of possible configuration commands

Command	Description
21 00 00	Port A → sets DIOA[00...07] as input
22 00 00	Port B → sets DIOB[00...07] as input
23 00 00	Port C → sets DIOC[00...07] as input
27 00 00	Port C_{LOW} → sets DIOC[00...03] as input
2B 00 00	Port C_{HIGH} → sets DIOC[04...07] as input
A1 00 00	Port A → sets DIOA[00...07] as output
A2 00 00	Port B → sets DIOB[00...07] as output
A3 00 00	Port C → sets DIOC[00...07] as output
A7 00 00	Port C_{LOW} → sets DIOC[00...03] as output
AB 00 00	Port C_{HIGH} → sets DIOC[04...07] as output

This set of commands is aimed at configuring HiLExtension 37IO's digital ports as input or output.

Example 2 - Configuring Port A as output.
Send: ”0xA1 0x00 0x00”
Answer: ”0xA1 0x00 0x00”

Example 3 - Configuring Port B as input.
Send: ”0x22 0x00 0x00”
Answer: ”0x22 0x00 0x00”

Example 4 - Configuring Port C_{LOW} as input and Port C_{HIGH} as output.
Send: ”0x27 0x00 0x00”
Answer: ”0x27 0x00 0x00”
Send: ”0xAB 0x00 0x00”
Answer: ”0xAB 0x00 0x00”

4.4. Writing commands

Table 4: List of possible writing commands

Command	Description
81 00 xx	Port A→ write states to DIOA[00...07]
82 00 xx	Port B→ write states to DIOB[00...07]
83 00 xx	Port C→ write states to DIOC[00...07]
87 00 xx	Port C_{LOW}→ write states to DIOC[00...03]
8B 00 xx	Port C_{HIGH}→ write states to DIOC[04...07]
C1 yy zz	Write value to AOUT1
C2 yy zz	Write value to AOUT2

This set of commands is aimed at controlling the output channels' logic state or voltage level. In addition to the command's operation code, the user must add:

- Digital outputs - In order to select a channel, a value has to be calculated and written (position "xx" from Table 4). The calculation can be done by summing the corresponding states for the channels' values as power of 2 (refer to Example 5 and Table 5). When a new port configuration is passed to the device, it will overwrite the new states for all the channels.
- Analog outputs - In order to select the desired output voltage level (position "yy zz" from Table 4) a value has to be calculated by the following formula 1.0:

$$value = \frac{voltage_{AOUT} \, x \, 4095}{3{,}3} \quad (1.0)$$

Note: In order to keep the correct format of the commands, "xx" and "yy zz" must be replaced with real numbers in hexadecimal format!

Example 5 – Write a value to Port A:
Channels DIOA[00...02] and DIOA[06,07] → LOW.
Channels DIOA[03...05] → HIGH.

Send: ”0xA1 0x00 0x00” → configure all channels from port A as outputs
Answer: ”0xA1 0x00 0x00”
Send: ”0x81 0x00 0x38” → write channels’ states
Answer: ”0x81 0x00 0x38”

Table 5: Example for 8-bit port output channels selection map

Channel No:	07	06	05	04	03	02	01	00
State[2,3]	0	0	1	1	1	0	0	0
Value[4]	0x38							

Example 6 - Write a value to Port C_{LOW}:
Channels DIOC[00,03] → LOW
Channels DIOC[01,02] → HIGH

Send: ”0xA7 0x00 0x00” → configure all channels from Port C_{LOW} as outputs
Answer: ”0xA7 0x00 0x00”
Send: ”0x87 0x00 0x06” → write channels’ states
Answer: ”0x87 0x00 0x06”

Table 6: Example for 4-bit port output channels selection map

Channel No:	3	2	1	0
State	0	1	1	0
Value	0x06			

Example 7 – Setting a voltage value to AOUT1 → 1,65V:

The target AOUT1 output voltage is:

$$voltage_{AOUT1} = 1{,}65\,V \quad (1.1)$$

When it is replaced in formula 1.0, the result is:

[2] 0 → Logic Low (0 ~ 0.8V); 1→ Logic High (3.3 ~ 5V)
[3] Binary representation
[4] Hexadecimal representation of the “state” value

$$value_{dec} = \frac{1{,}65 \times 4095}{3{,}3} = 2047 \quad (1.2)$$

$$value_{hex} = 07FF \quad (1.3)$$

Send: ”0xC1 0x07 0xFF” → command to set the output voltage of AOUT1 to 1.65V
Answer: ”0xC1 0x07 0xFF”

Example 8 – Setting a voltage value to AOUT2 → 3,30V:

The target AOUT1 output voltage is:

$$voltage_{AOUT2} = 3{,}30\,V \quad (1.4)$$

When it is replaced in formula 1.0, the result equation is:

$$value_{dec} = \frac{3{,}30 \times 4095}{3{,}3} = 4095 \quad (1.5)$$

$$value_{hex} = 0FFF \quad (1.6)$$

Send: ”0xC2 0x0F 0xFF” → command to set the output voltage of AOUT2 to 3.3V
Answer: ”0xC2 0x0F 0xFF”

4.5. Reading commands

Table 7: List of possible reading commands

Command	Description
01 00 00	Port A→ read states of DIOA[00...07]
02 00 00	Port B→ read states of DIOB[00...07]
03 00 00	Port C→ read states of DIOC[00...07]
07 00 00	Port C_{LOW}→ read states of DIOC[00...03]
0B 00 00	Port C_{HIGH}→ read states of DIOC[04...07
40 00 00	Read value of AIN00
41 00 00	Read value of AIN01
42 00 00	Read value of AIN02
43 00 00	Read value of AIN03
44 00 00	Read value of AIN04
45 00 00	Read value of AIN05
46 00 00	Read value of AIN06
47 00 00	Read value of AIN07
4F 00 00	Read value of AIN[00...07]

This set of commands is to be used for reading the input channels' logic state or voltage level:

- Digital inputs – After a reading command is transmitted, the device will respond with the command's operation code and the result of a reading in the last byte of the message. That result is the actual channels' state of the corresponding port. It might be analyzed by converting its hexadecimal value to a binary number and comparing it with the 4/8-bit port input channels' reading map (refer to Table 8 and Table 9).
- Analog inputs - After a reading command is transmitted, the device will respond with the command's operation code and result of a reading in the last 2 bytes of the message. That respond will have the following format:
  <command code><$byte_2$><$byte_1$>

The reading result contains the hexadecimal values directly read from the device's ADC. Thus, the real voltage level has to be converted according to the formula 2.0:

$$voltage_{AIN} = \frac{(byte_2 \times 256 + byte_1) \times 3{,}3}{4096} \quad (2.0)$$

Example 9 – Reading the input state of port A:

Send: ”0x21 0x00 0x00” → configure all channels from port A as inputs
Answer: ”0x21 0x00 0x00”
Send: ”0x01 0x00 0x00” → command for reading port A channels’ states
Answer: ”0x01 0x00 0x04” → results from reading

Table 8: Example for 8-bit port input channels read map

Channel No:	07	06	05	04	03	02	01	00
State	0	0	0	0	0	1	0	0
Value	0x04							

Example 10 – Reading the input state of port C_{HIGH}:

Send: ”0x2B 0x00 0x00” → configure all channels from port C_{HIGH} as inputs
Answer: ”0x2B 0x00 0x00”
Send: ”0x0B 0x00 0x00” → command for reading port C_{HIGH} channels’ states
Answer: ”0x0B 0x00 0x08” → result from reading

Table 9: Example for 4-bit port input channels read map

Channel No:	3	2	1	0
State	1	0	0	0
Value	0x08			

Example 11 – Reading input voltage of 1.65V on AIN05:

Send: ”0x45 0x00 0x00” → command for reading AIN05
Answer: ”0x45 0x07 0xFF”

The read input value is:

$$byte_1 = FF_h = 255 \quad (2.1)$$

$$byte_2 = 07_h = 7 \quad (2.2)$$

When those values are replaced in formula 2.0, the result is:

$$voltage_{AIN05} = \frac{(7 \times 256 + 255) \times 3{,}3}{4096} \quad (2.3)$$

$$voltage_{AIN05} \approx 1{,}65V \quad (2.4)$$

Example 12 – Reading input voltage at all inputs:

Send: ”0x4F 0x00 0x00” → command for reading AIN05
Answer: ”0x4F 0x00 0x01 0x01 0xEC 0x03 0xF3 0x06 0x08 0x08 0x0B 0x09 0xFE 0x0B 0xE9 0x0D 0xF1”

Only when the aforementioned command is used, the adapter’s response has the structure shown in Table 10. And each value field is consisted by two bytes - <byte$_2$><byte$_1$>.

Table 10: Example for reading all analog inputs

Byte No:	15	[14,13]	[12,11]	[10,9]	[8,7]	[6,5]	[4,3]	[2,1]	0
Field:	CMD code	$value_{AIN0}$ 0	$value_{AIN0}$ 1	$value_{AIN0}$ 2	$value_{AIN0}$ 3	$value_{AIN0}$ 4	$value_{AIN0}$ 5	$value_{AIN0}$ 6	$value_{AIN0}$ 7

When those values are replaced in formula 2.0, the results are:

$$voltage_{AIN00} \approx 0{,}00V \quad (2.5)$$
$$voltage_{AIN01} \approx 0{,}40V \quad (2.6)$$
$$voltage_{AIN02} \approx 0{,}82V \quad (2.7)$$
$$voltage_{AIN03} \approx 1{,}25V \quad (2.8)$$
$$voltage_{AIN04} \approx 1{,}66V \quad (2.9)$$
$$voltage_{AIN05} \approx 2{,}06V \quad (2.10)$$
$$voltage_{AIN06} \approx 2{,}46V \quad (2.11)$$
$$voltage_{AIN07} \approx 2{,}88V \quad (2.12)$$

5. Practical use of HiLExtension 37IO

5.1. Example schematics for peripherals routing

5.1.1. Connecting button

The button can be used for simulation of input signal and testing the digital inputs. The following steps are required for the proper integration of HiLExtension 37IO:

1. Set the wiring connections as shown on Figure 4:

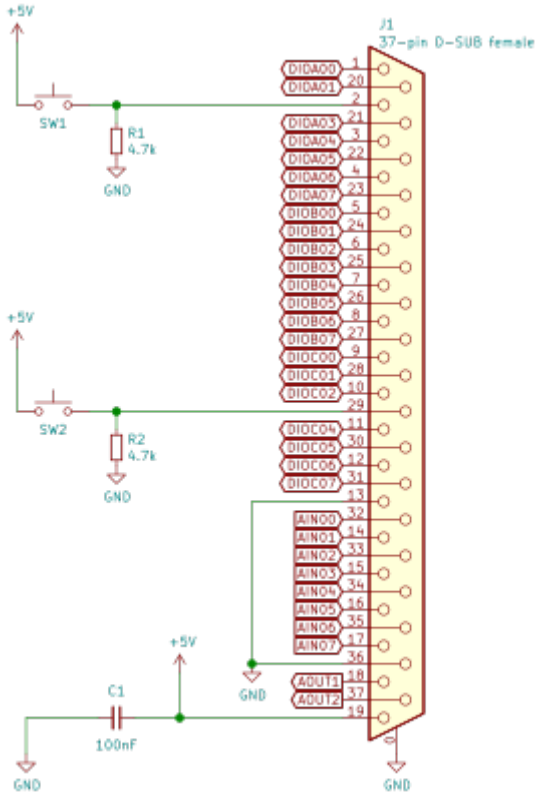

Figure 4. Schematic for connecting buttons to HiLExtension 37IO.

2. Connect with the HiLExtension 37IO and execute examples 9 and 10.
3. Press the button for testing.

5.1.2. Connecting LED

A LED can be used for visual indication of the current state of the digital output. The following steps are required for the proper integration with HiLExtension 37IO:

1. Set the wiring connections shown on Figure 5:

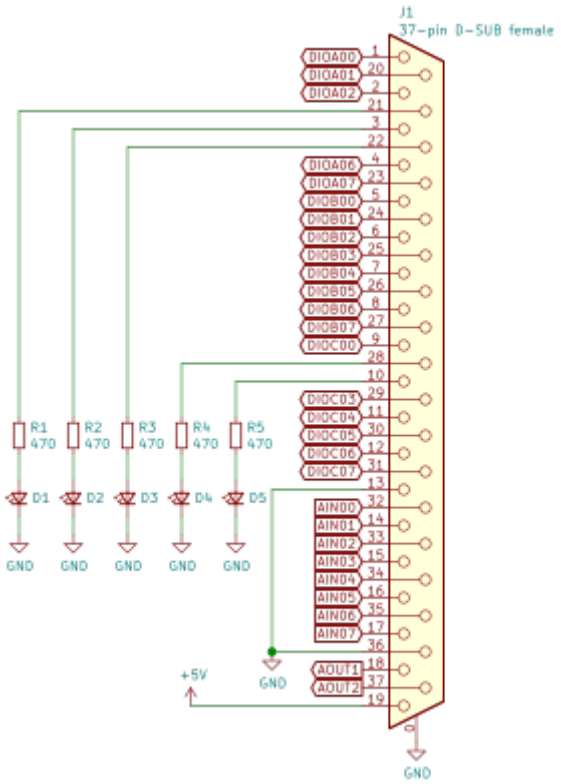

Figure 5. Schematic for connecting a LED to HiLExtension 37IO.

2. Connect with the HiLExtension 37IO and execute examples 5 and 6.
3. Observe the LED's indication.

5.1.3. Connecting potentiometer

A potentiometer can be used for simulation of input voltage and testing the analog inputs. The following steps are required for the proper integration with HiLExtension 37IO:

1. Set the wiring connections shown on Figure 6:

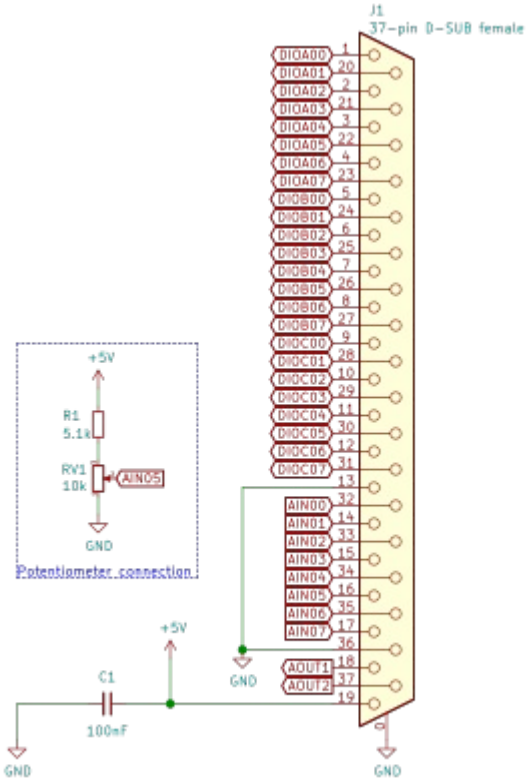

Figure 6. Schematic for connecting a potentiometer to HiLExtension 37IO.

2. Connect with the HiLExtension 37IO and execute example 11. *(If the concept described in example 12 has been tested , "potentiometer connection" must be implemented in all analog inputs).*
3. Spin the potentiometer and observe the analog input readings.

5.1.4. Connecting voltmeter to analog output

A voltmeter can be connected to HiLExtension 37IO in order to inspect the output voltage generated by the DAC. The following steps are required:

1. Set the connections according to the schematic shown in Figure 7:

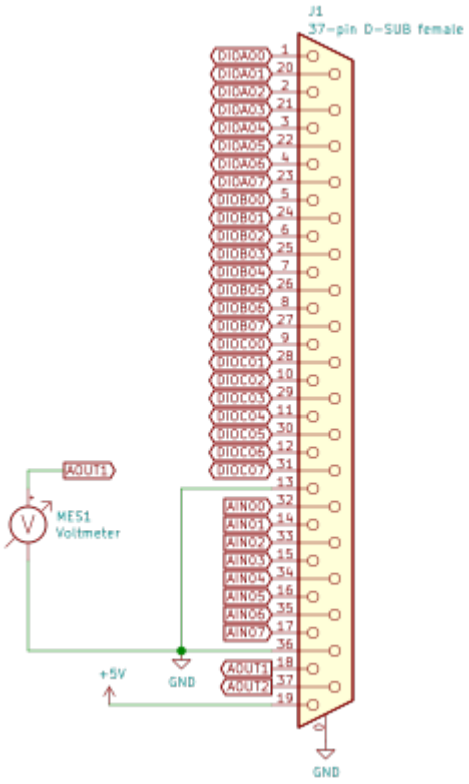

Figure 7. Schematic for connecting a voltmeter to HiLExtension 37IO.

2. Connect with the HiLExtension 37IO and execute example 7 or 8.
3. Observe the voltmeter readings.

5.1.5. Connecting transistor-controlled relay

A relay can be connected to HiLExtension 37IO if the switching of power lines is required. The relay must be controlled by a transistor. Follow the steps below:

1. Set the wiring connections shown in Figure 8:
2. Connect with the HiLExtension 37IO and execute the following command sequence:

Send: ”0xA1 0x00 0x00” → configure all channels from port A as outputs
Answer: ”0xA1 0x00 0x00”
Send: ”0x81 0x00 0x01” → write DIOA 00 in HIGH state
Answer: ”0x81 0x00 0x01”
Send: ”0x81 0x00 0x00” → write DIOA 00 in LOW state
Answer: ”0x81 0x00 0x00”

3. Observe the relay switching.

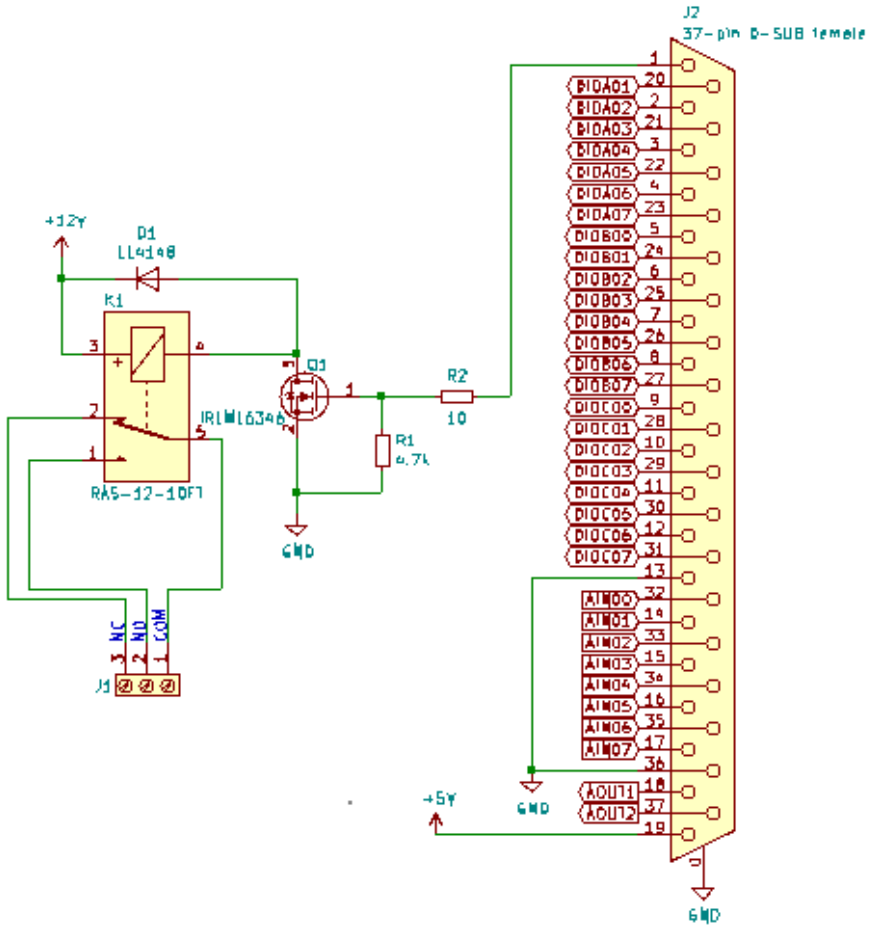

Figure 8. Schematic for connecting a relay to HiLExtension 37IO.

5.2. Complete HiL system

Mitkov Systems GmbH has designed a full HiL testing system. It includes a series of special extension modules and testing software. Their purpose is to reduce the time for hardware assemblies and to facilitate the proof of concept.

5.2.1. HiLPowerBoard

The module has:

- 24 green LEDs → 8 for each port
- 4 MOSFET controlled outputs → $I_{MAX} = 7A$ at $V_{DS} = 13V$
- 8 relay-controlled outputs → $I_{contact} = 7A$ at 250VAC or 10A at 24VDC
- wires connection via 5.0 mm pitch screw terminal block connectors
- 37-pin D-SUB male connector assembled on board
- option for multiple modules stacking by spacers (Figure 9.)

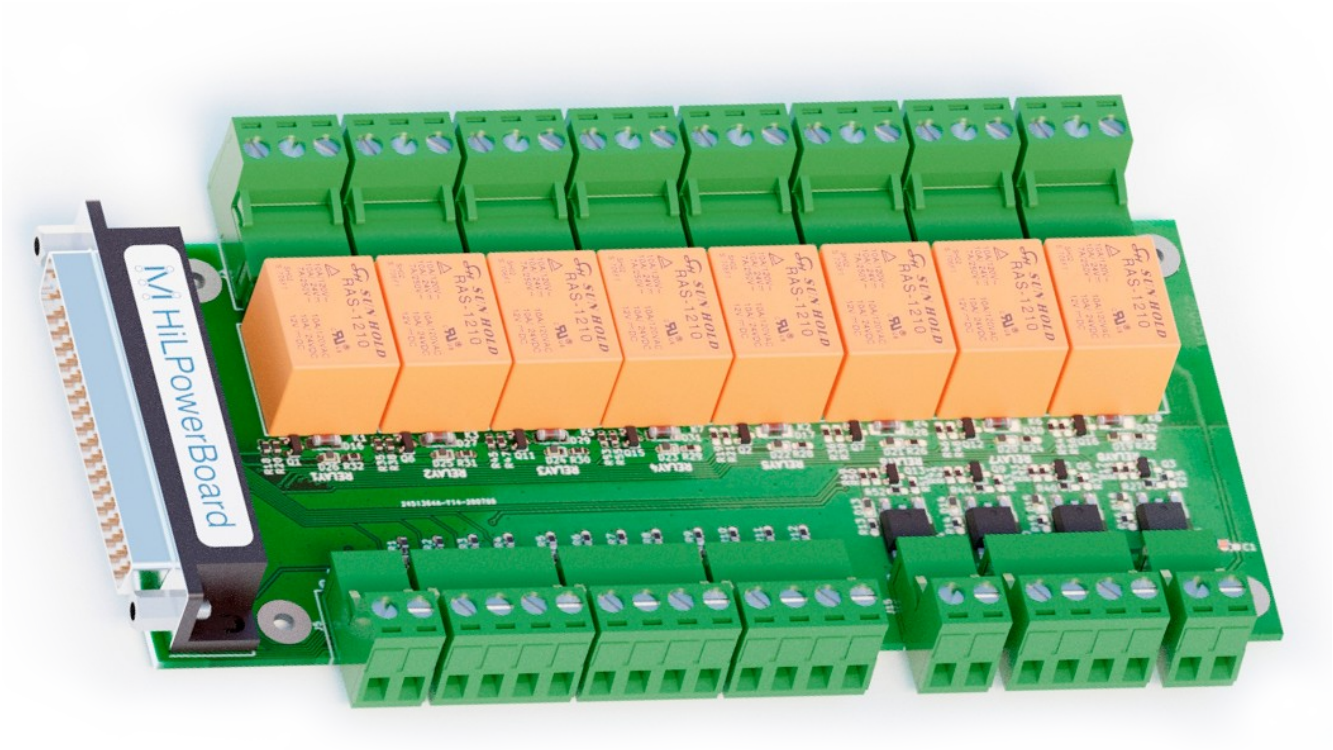

Figure 9. HiLPowerBoard

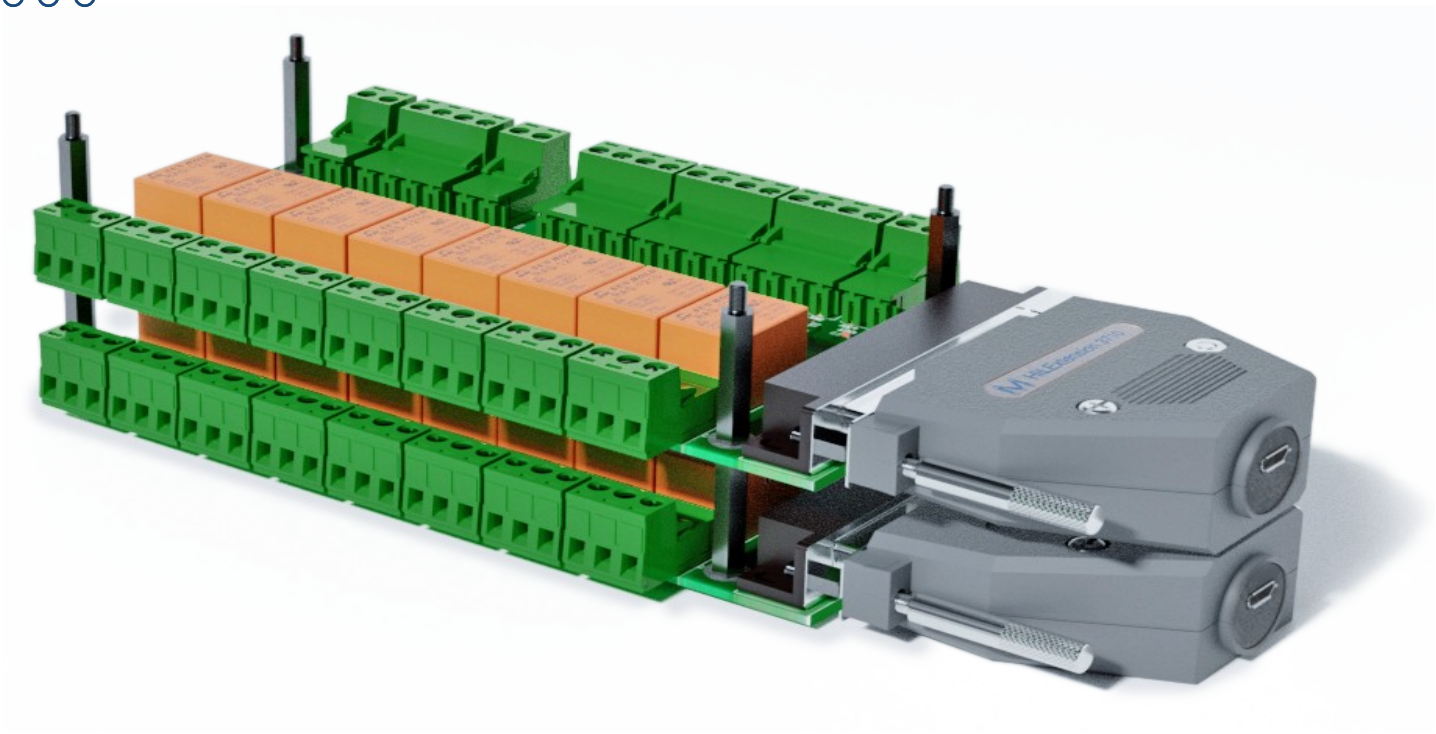

Figure 10. Stacking of multiple HiLPowerBoard modules

5.2.2. HiLTestBoard

The module has:

- 24 green LEDs
- 24 push buttons
- 8 potentiometers
- 2 headers for analog outputs measurement
- 37-pin D-SUB male connector assembled on board

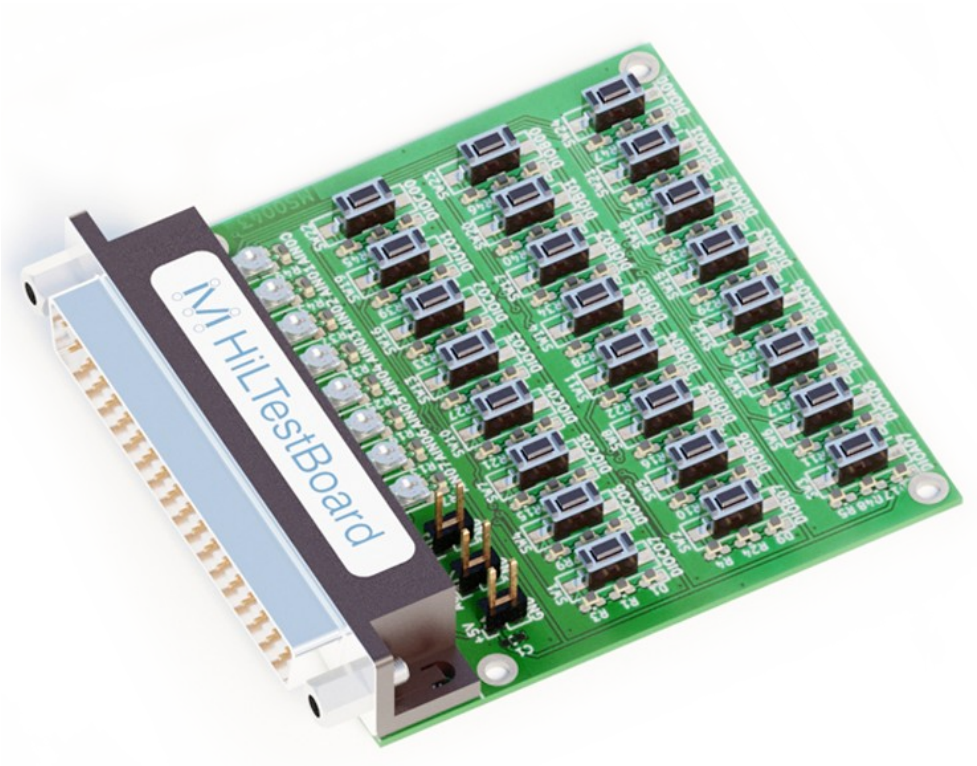

Figure 11. HiLTestBoard

5.2.3. HiL Simplicity

HIL Simplicity is created to facilitate the device testing by the end-user. It allows the connection of multiple HiLExtension 37IO adapters. The testing process can be set in individual steps with accuracy of 20 ms minimum. The user can modify the sequence of the test steps and hardware capabilities which is in use.

HIL Simplicity is provided in an easily executable format (Linux and Windows compatible). After downloading, the user has to click on the icon in order to start the software. No installation is required!

For addition information or download of HIL Simplicity software, please refer to the document "*HIL Simplicity User manual*" or contact Mitkov Systems GmbH.

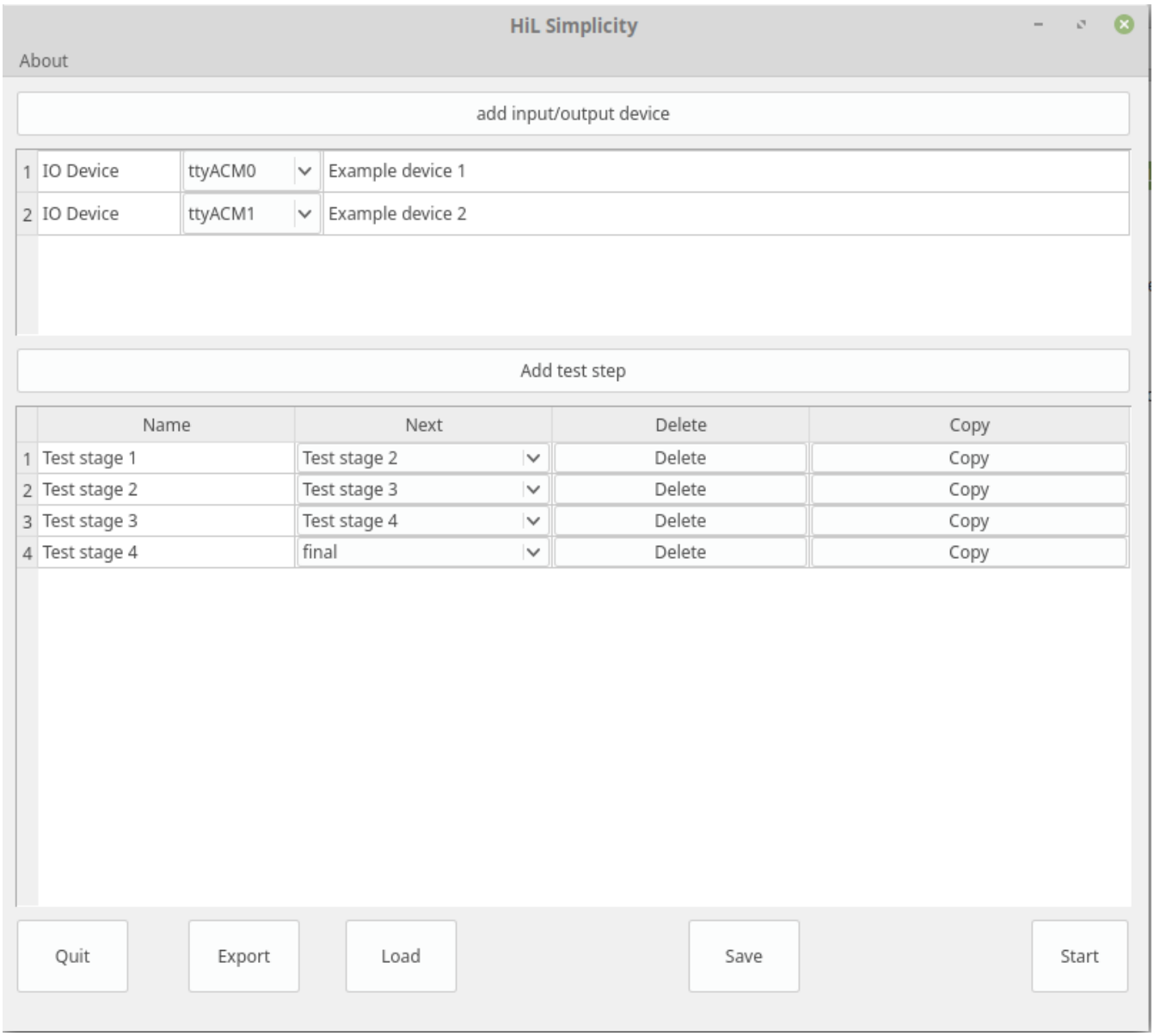

Figure 12. HiL Simplicity

6. Important notes

- For better measurement results, use it with shielded cables.
- Pin 19 of the 37-pin D-SUB female connector is a +5V voltage source. Do not apply any voltage to pin 19 - it will damage the device!
- The device is operating with low voltage signals. Please observe the relevant power regulations!
- Only use the device with closed enclosure. ESD voltages at open traces may cause malfunction.
- Avoid using the device in conditions other than the specified.

7. Revision history

Date	Version	Changes
25.07.2020	1.0	Initial release