

STEVAL-MKI109D Professional MEMS Tool motherboard for MEMS adapter boards

Introduction

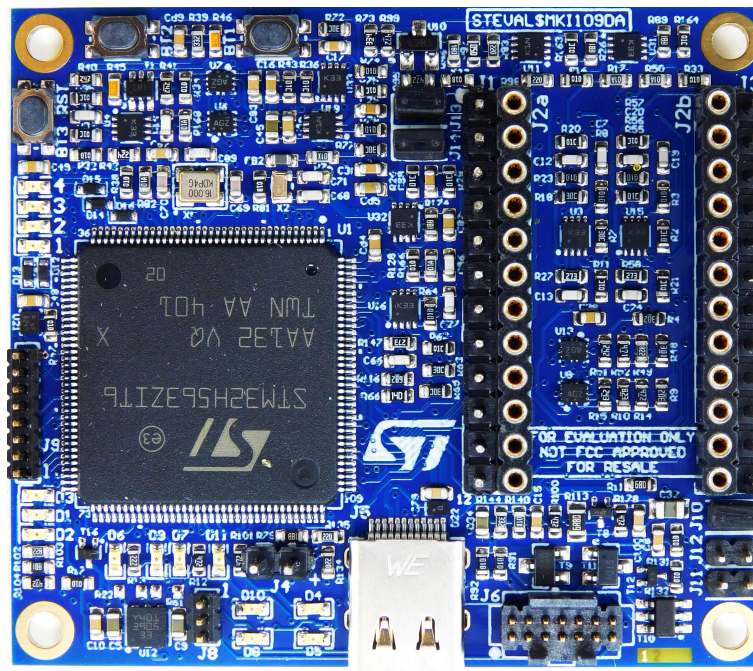
The STEVAL-MKI109D motherboard provides users with a complete, ready-to-use platform for the evaluation of STMicroelectronics MEMS products.

It includes a high-performance 32-bit microcontroller which functions as a bridge between the sensors and a PC, on which you can download and run the graphical user interface (GUI) or dedicated software routines for customized applications.

The board features a DIL24 socket to mount all available adapters for both digital and analog output MEMS devices.

This motherboard is mechanically backward compatible with its predecessor STEVAL-MKI109D including holes and connectors positions (with exception of USB-C and programming connector J6).

Figure 1. Professional MEMS tool kit STEVAL-MKI109D

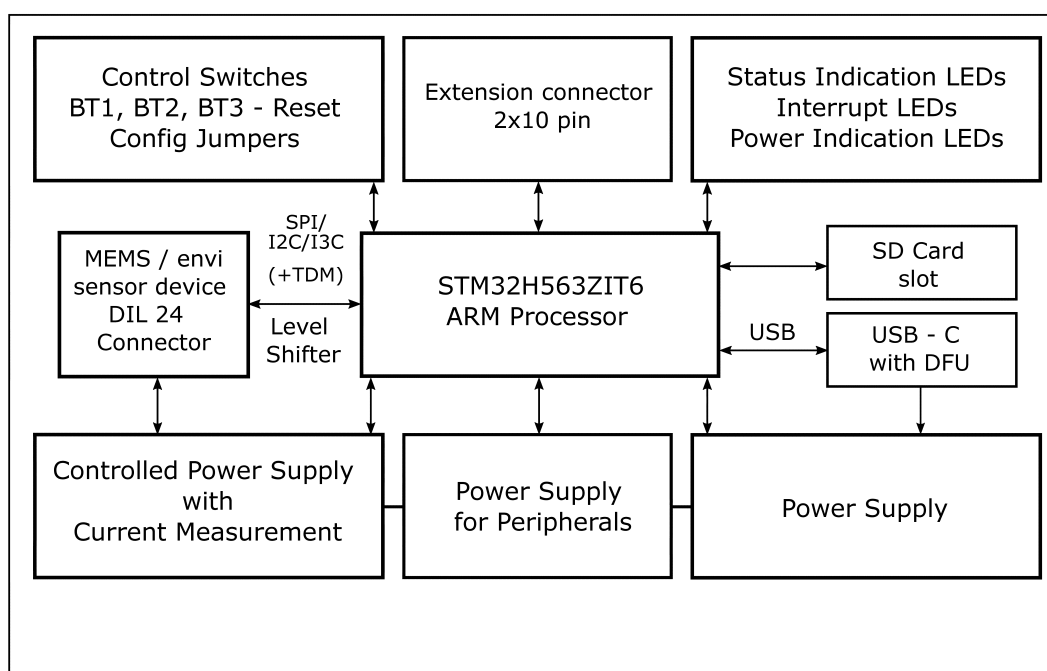


Notice: For dedicated assistance, submit a request through our online support portal at www.st.com/support.

1 Demonstration kit description

The Professional MEMS Tool is a complete demonstration kit for digital and analog MEMS sensors. Thanks to its DIL24 connector, a wide range of MEMS adapter boards can be used.

Figure 2. Demonstration board block diagram



The Professional MEMS Tool demonstration kit is based on the STM32H563ZIT6 microcontroller and can be connected to a PC via USB. Data from MEMS sensors connected to the board can be read through the PC GUI provided with the kit.

STEVAL-MKI109D also implements the device firmware upgrade (DFU) feature, so it can be reprogrammed with a new firmware release without the need to use a programmer (see www.st.com/mems).

The STEVAL-MKI109D integrates:

- LEDs:
 - four LEDs connected via FET buffers to the interrupt pins of digital adapters
 - a power/USB LED
 - eight general-purpose LEDs for firmware/hardware state indication
 - four LEDs for power and overcurrent state indication
- Three buttons:
 - two user buttons on a dedicated GPIO of the microcontroller
 - a microcontroller reset button
- Connectors feature:
 - DIL24 Socket for MEMS adapter, J8 J9 for extension module
 - Power header J4 with USB switched Vbus at top

All the MEMS adapter pins are replicated and available on board connectors J1 and J3.

Figure 3. Top silkscreen of the Professional MEMS Tool kit STEVAL-MKI109D

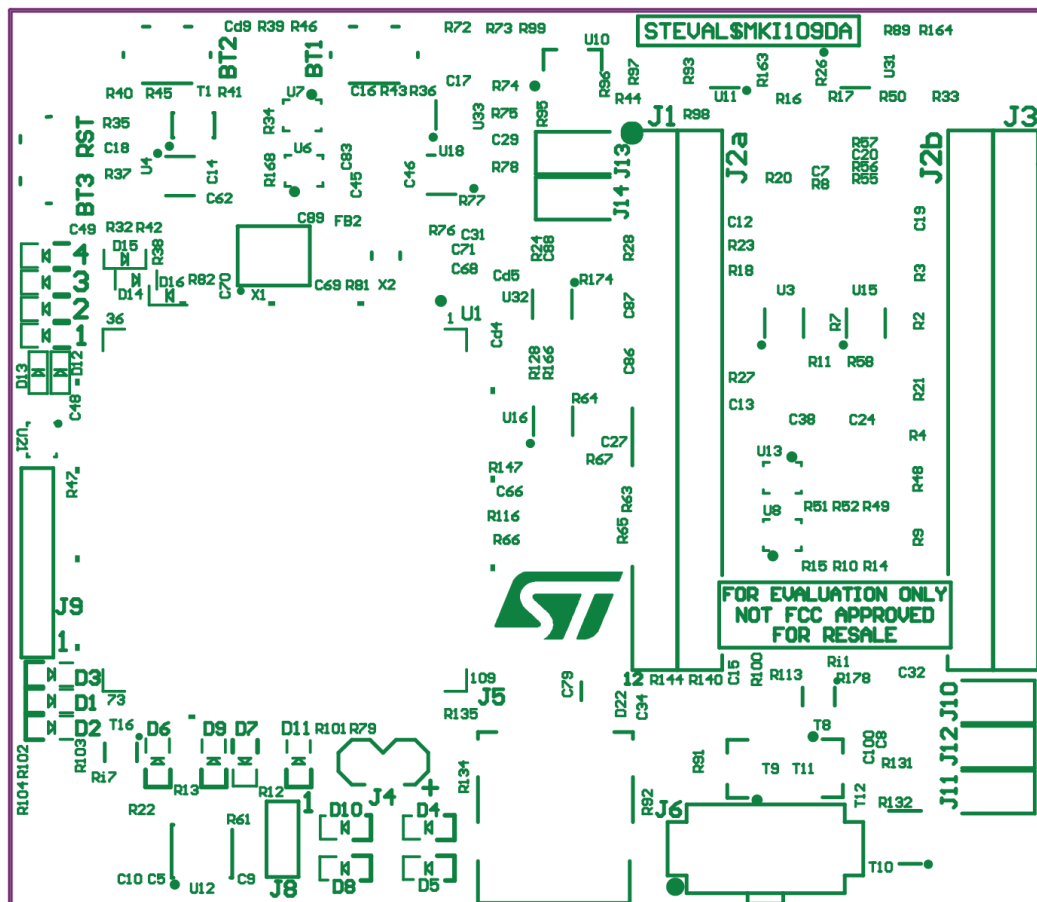


Figure 4. Top view of Professional MEMS Tool kit STEVAL-MKI109D

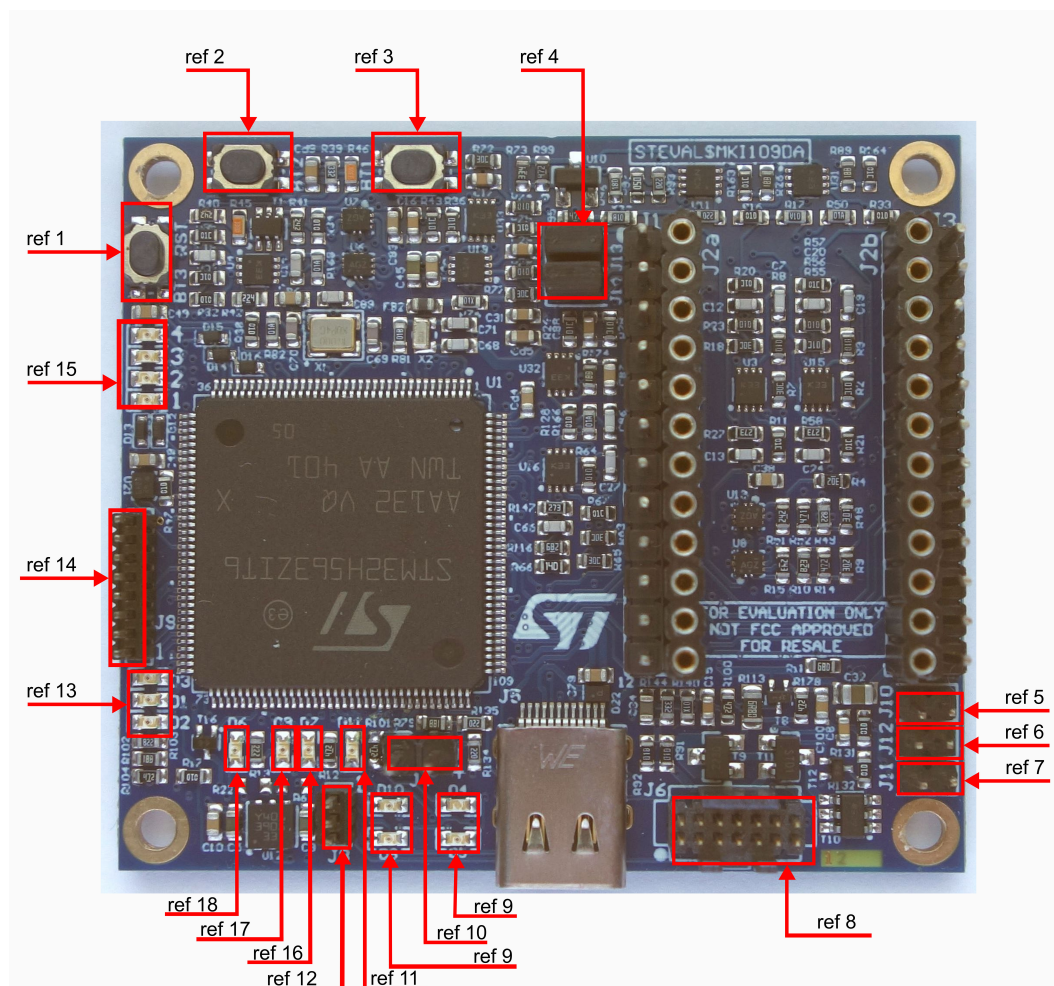


Figure 4 highlights some of the main components on the top layer of the Professional MEMS Tool kit.

1. Button BT3 is used to reset the STM32.
2. Button BT2 connected to STM32 GPIOs and available to the user. To enter DFU mode:
 - a. press buttons BT3 (Reset) and BT2 together
 - b. first release BT3 and then release BT2
3. BT1 connected to STM32 GPIOs and available to the user.
4. Jumpers J13 (VDD) and J14 (VDDIO) allow the user to measure the sensor current consumption by connecting a multimeter in series with their terminals.
5. Jumper J10 is used as a general purpose input to manually set certain features for several MEMS adapters.
6. Jumper J12 is used as a general purpose input to manually set certain features for several MEMS adapters.
7. Jumper J11 is used to enable internal power 3.6 V for Pin11 of J2a. (Needs enabling by FW).
8. J6 connector can be used to reprogram the STM32 and debug the code through the JTAG or SWD protocols
9. LEDs D4, D5, D8, D10 are used for firmware state indication: D4: USB enumeration, D5: Internal power Vin, D8: General purpose state indication D10: General purpose / SD card state indication.
10. Jumper J4 can be used to supply external boards by USB Vbus 5V. Output current is internally limited to 500mA by U26.
11. LED D11 lights up when the board is powered up and directly indicates 3.3V voltage on SD card
12. J8 connector can be used for UART RX/TX communication. Is used together with J9 (ref.14)

13. LEDs D1, D2, and D3 are general-purpose LEDs used to indicate firmware states; e.g.:
 - a. LED D3 YELLOW light up when specific firmware is selected from those available
 - b. LED D2 RED on indicates that the microcontroller is properly configured for communication with the sensor
 - c. LED D1 GREEN blinks at run state according to the sensor data rate selected
14. J9 connector can be used for general purpose SPI / I2C bus. Is used with J8 for UART (ref.12).
15. LEDs 1, 2, 3, 4 are directly indicating state of the interrupt pins of the MEMS digital adapter (if available on the sensor mount on the adapter board). All LEDs are connected via level translator IC U5.
16. LED D7 lights up when the Vbus 5V to J4 is switched ON.
17. LED D9 lights up when the overcurrent >500mA from J4 is sensed by U26.
18. LED D6 lights up when the internal power 3.6V is powered ON and can be passed to Pin 11 of tested adapter through J11 jumper.

Figure 5. Bottom view of Professional MEMS Tool kit STEVAL-MKI109D

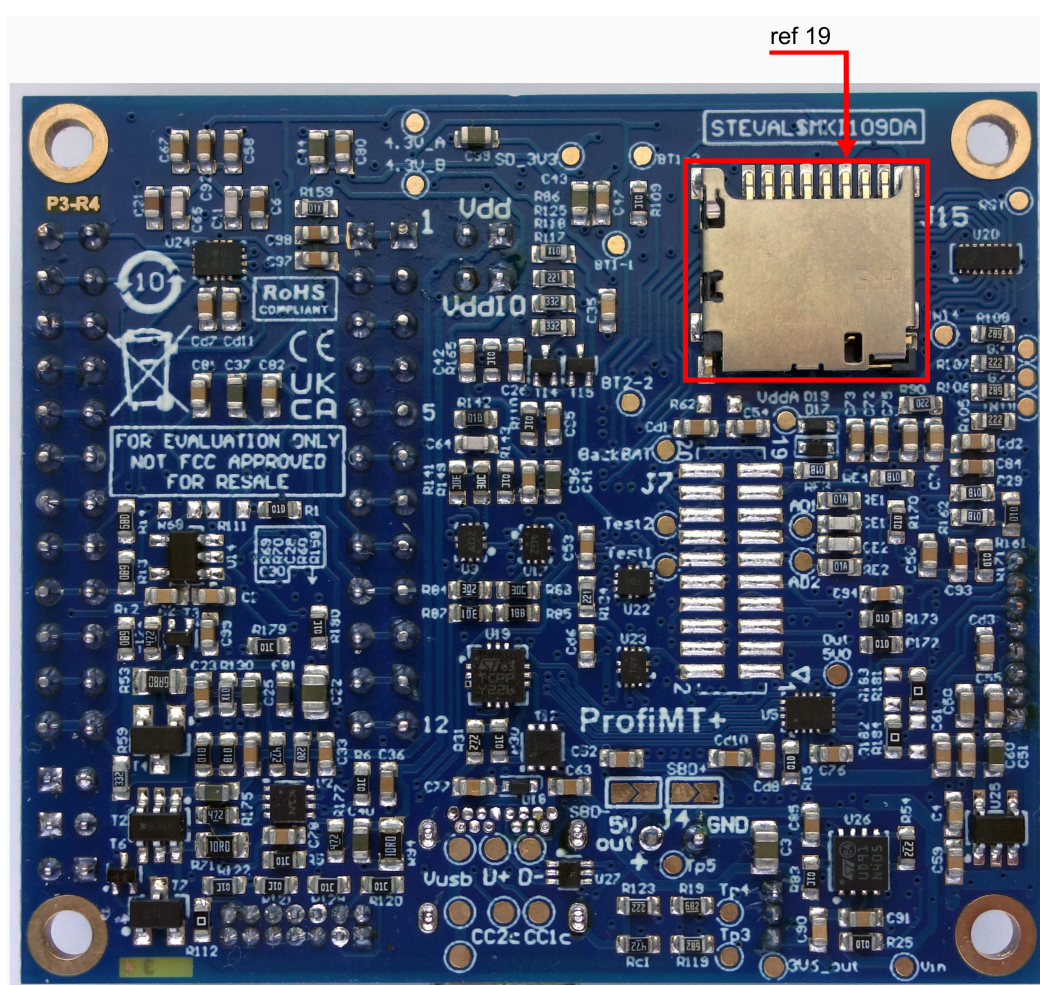


Figure 5 highlights main component on the bottom layer of the Professional MEMS Tool kit:
19. Micro SD Card slot.

Figure 6. How to plug the DIL24 adapter on STEVAL-MKI109D

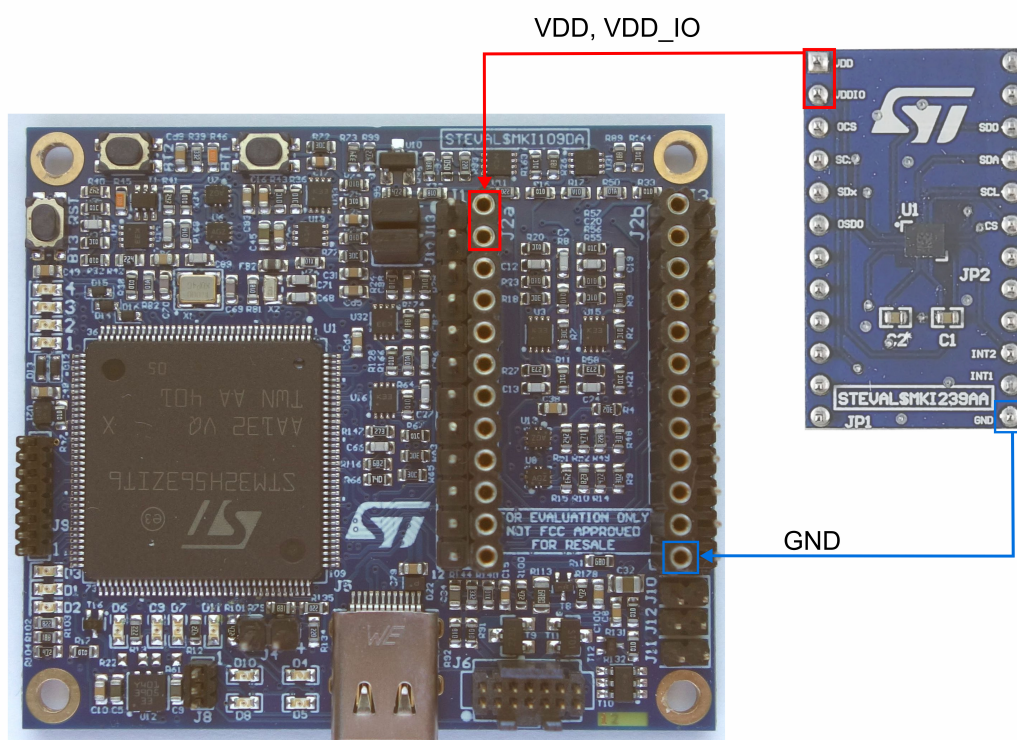


Figure 6 shows how to plug the DIL 24 adapter MEMS module on the Professional MEMS Tool STEVAL-MKI109D.

VDD and VDDIO are on the top left corner (pins 1 and 2) on J2a and GND is on the bottom right corner on J2b (pin 13 of adapter).

2 Professional MEMS tool board installation

Professional MEMS Tool board MKI109D can be used with [MEMS-Studio](#) or [Unico-GUI](#) graphical user interface. The Unico-GUI package (.zip) downloaded from [st.com](#) contains a "FIRMWARE" folder with .dfu/.bin files for firmware upgrade.

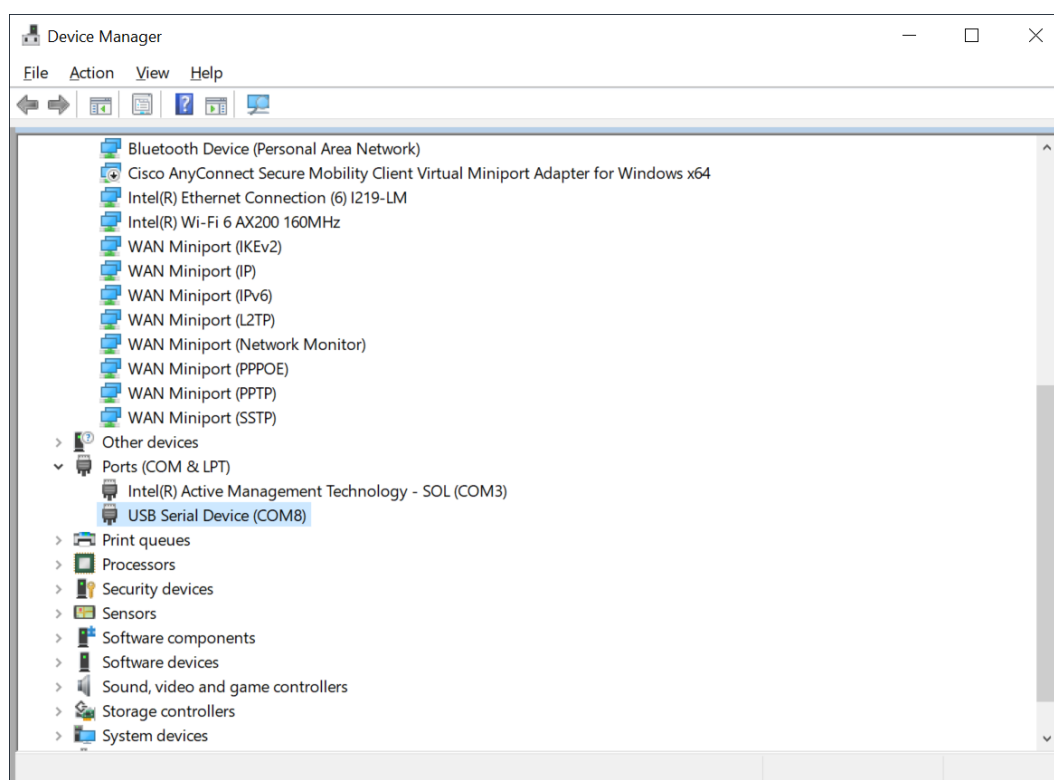
2.1 Hardware installation

- For Windows 10, Linux® and Mac OS® platforms, no driver installation is required.
- For older Windows platforms, install the STM32 virtual COM port driver from the [STSW-STM32102](#) package.

Once the driver is installed, connect the demonstration kit board to a free USB port. A confirmation message should appear.

Note: On Windows 7 platform, confirm which COM port has been assigned to the board: right click on *My Computer* and select *Manage*, then select *Device Manager* and scroll through the list to *Ports (COM & LPT)*. The STM32 virtual COM port driver for Windows 7 platform and related documents are packaged with the [STSW-STM32102](#) software downloadable at [www.st.com](#).

Figure 7. Virtual COM port assignment



2.2 Firmware upgrade

You can reprogram the STEVAL-MKI109D MEMS evaluation board both by the ST-LINK hardware programmer using J6 connector, or by simply connecting to the USB connector using the STM32CubeProgrammer software (alternatively using MEMS Studio in firmware Programming section) and using the DFU mode of the evaluation board.

The following steps show the most simple way using the STM32CubeProgrammer software and the DFU mode of STEVAL-MKI109D. The input files can be both *.bin or *.dfu for the STM32CubeProgrammer.

Note: The STM32CubeProgrammer software uses *.bin source files for firmware upgrade.

Note: In case you have installed older DfuSe software on your machine, you must uninstall it before installing the STM32CubeProgrammer software.

For all details on the use of STM32CubeProgrammer software, refer to user manual UM2237 on www.st.com.

2.2.1 Entering DFU mode

The direct reprogramming of the microcontroller is particularly suited to USB applications where the same USB connector can be used both for the standard operating mode and the reprogramming process.

To configure the Professional MEMS Tool board in DFU mode:

- press button BT2 before supplying the board and release it when LED D6 lights up
- or
 1. press BT3 (Reset) and BT2 together
 2. release BT3 followed by BT2

Led D6 lights up and the device should appear in the Windows Device manager as “STM device in DFU mode”.

It is necessary to switch the Professional MEMS Tool board in DFU mode before executing the firmware upgrade procedures described in the following sections.

2.2.2 Firmware upgrade on Windows with STM32CubeProgrammer using *.bin source files

To upgrade the firmware by using the *.bin source files, follow the steps below.

Step 1. Attach the STEVAL-MKI109D MEMS board to the USB port while you are holding the BT2 button on it (procedure described in Section 2.2.1). As alternative to this procedure you can reset board by pressing BT1 button while you are holding BT2 button.

The STEVAL-MKI109D becomes then visible in your system as a device in DFU mode.

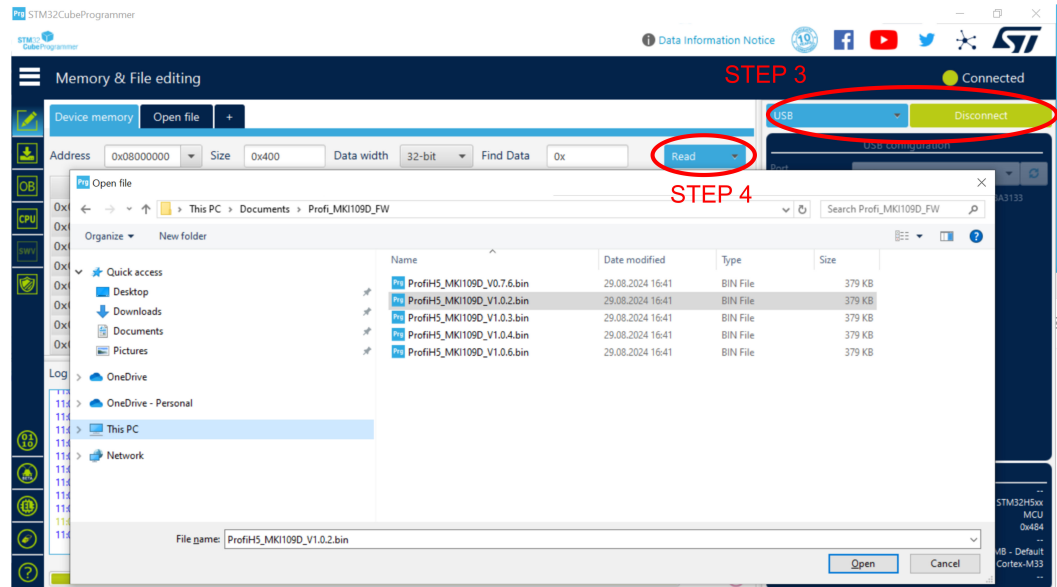
Step 2. Open the STM32CubeProgrammer software application.

Step 3. Select [USB] at the top-right corner roller blue button and then click on the [Connect] green button at the right side.

In the log window, the “Data read successfully” message should appear. The connection state indicator should show the “Connected” green state.

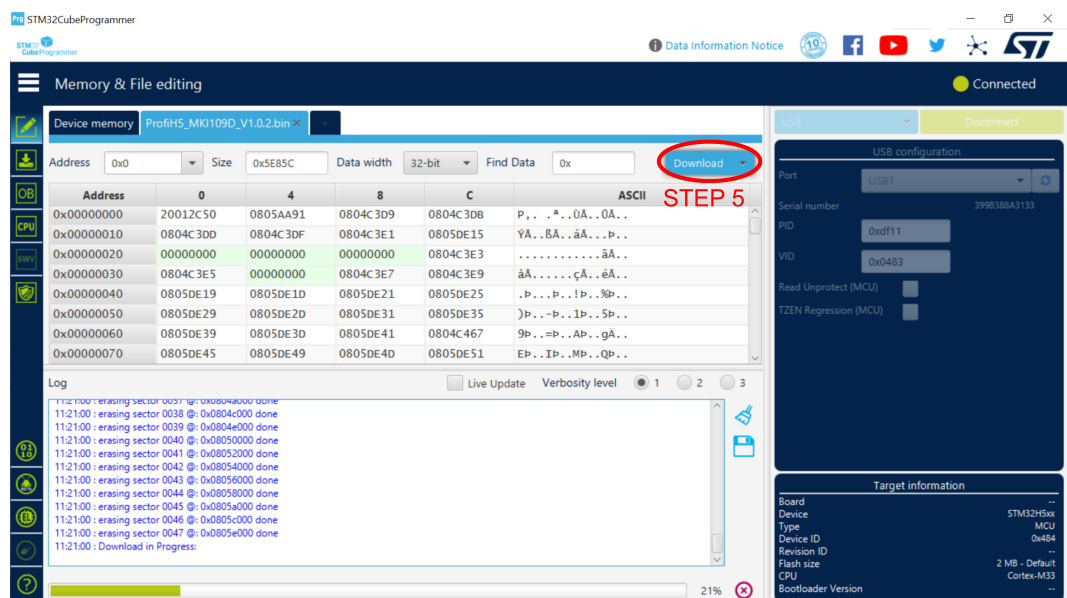
- Step 4.** Click on **[Open File]** and select the proper *.bin file that you want to download into the STEVAL-MK109D.

Figure 8. Opening the selected .bin file in STM32CubeProgrammer



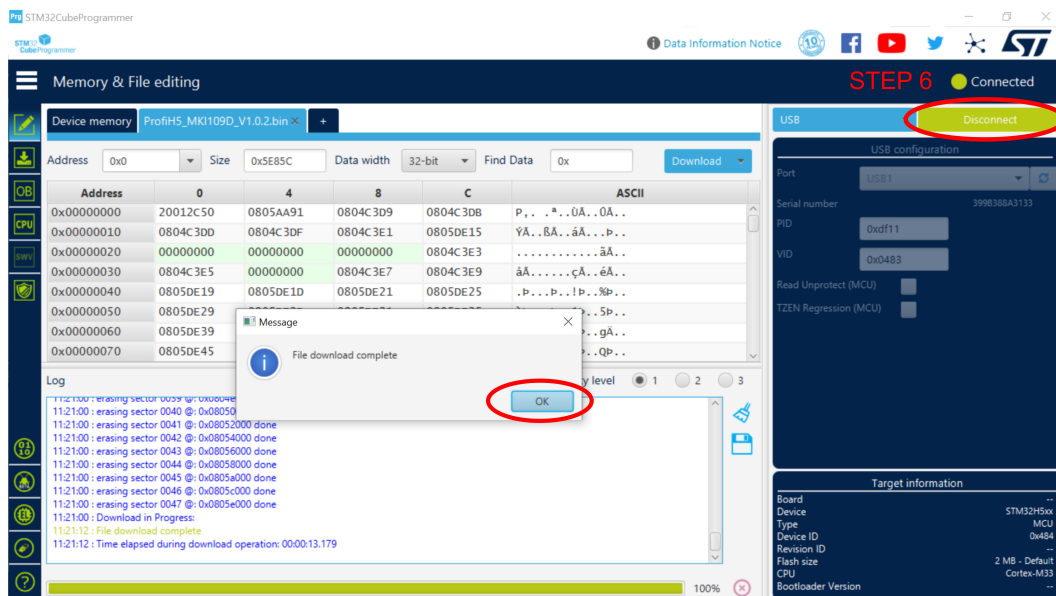
- Step 5.** After choosing the tab with the selected .bin file, click on the **[Download]** blue button at the top right (there you can check the proper starting address "0x08000000"). The programming process then starts.

Figure 9. Programming process in STM32CubeProgrammer



- Step 6.** After the successful update of the STEVAL-MK109D, the “File download complete” message should appear.
- The firmware update is then complete.
- Click on the **[OK]** button and then click on the **[Disconnect]** green button at the top right.

Figure 10. File download complete message after successful firmware update



2.2.3 Firmware upgrade using command line on Windows/Linux/Mac OS® with STM32CubeProgrammer using *.bin source files

There is possible to use STM32CubeProgrammer tool in terminal / command line. To upgrade the firmware by using the *.bin source files, follow the steps below.

- Step 1.** Attach the [STEVAL-MKI109D](#) MEMS board to the USB port while you are holding the BT2 button on it (procedure described in [Section 2.2.1](#)). Then, run the command:

```
STM32_Programmer_CLI -c port=USB1 -d . ProfiH5_MKI109D_V1.0.2.bin 0x8000000  
-v -g 0x800000
```

where

ProfiH5_MKI109D_V1.0.2.bin is the .bin file name.

3 Supported MEMS adapter boards

Table 1. List of supported MEMS adapter boards

Adapter board	Device	*setb command string
STEVAL-MET001V1	LPS22HB	001V1
STEVAL-MKI105V1	LIS3DH	105V1
STEVAL-MKI110V1	AIS328DQ	110V1
STEVAL-MKI125V1	A3G4250D	125V1
STEVAL-MKI135V1	LIS2DH	135V1
STEVAL-MKI137V1	LIS3MDL	137V1
STEVAL-MKI151V1	LIS2DH12	151V1
STEVAL-MKI153V1	H3LIS331DL	153V1
STEVAL-MKI158V1	AIS3624DQ	158V1
STEVAL-MKI163V1	LSM303C	163V1
STEVAL-MKI164V1	LIS2HH12	164V1
STEVAL-MKI165V1	LPS25HB	165V1
STEVAL-MKI166V1	H3LIS100DL	166V1
STEVAL-MKI167V1	H3LIS200DL	167V1
STEVAL-MKI168V1	IIS2DH	168V1
STEVAL-MKI169V1	I3G4250D	169V1
STEVAL-MKI170V1	IIS328DQ	170V1
STEVAL-MKI172V1	LSM303AGR	172V1
STEVAL-MKI173V1	LSM303AH	173V1
STEVAL-MKI174V1	LIS2DS12	174V1
STEVAL-MKI175V1	LIS2DE12	175V1
STEVAL-MKI178V2	LSM6DSL	178V2
STEVAL-MKI179V1	LIS2DW12	179V1
STEVAL-MKI180V1	LIS3DHH	180V1
STEVAL-MKI181V1	LIS2MDL	181V1
STEVAL-MKI182V2	ISM330DLC	182V2
STEVAL-MKI184V1	ISM303DAC	184V1
STEVAL-MKI185V1	IIS2MDC	185V1
STEVAL-MKI186V1	IIS3DHHC	186V1
STEVAL-MKI189V1	LSM6DSM	189V1
STEVAL-MKI190V1	LIS2DTW12	190V1
STEVAL-MKI191V1	IIS2DLPC	191V1
STEVAL-MKI192V1	LPS22HH	192V1
STEVAL-MKI193V1	ASM330LHH	193V1
STEVAL-MKI194V1	LSM6DSR	194V1
STEVAL-MKI195V1	LSM6DSRX	195V1
STEVAL-MKI196V1	LSM6DSO	196V1
STEVAL-MKI197V1	LSM6DSOX	197V1

Adapter board	Device	*setb command string
STEVAL-MKI198V1K	STTS751	198V1K
STEVAL-MKI199V1K	STLM20	199V1K
STEVAL-MKI200V1K	STTS22H	200V1K
STEVAL-MKI201V1K	STTS75	201V1K
STEVAL-MKI202V1K	STDS75	202V1K
STEVAL-MKI203V1K	STCN75	203V1K
STEVAL-MKI204V1K	STLM75	204V1K
STEVAL-MKI205V1	LPS33W	205V1
STEVAL-MKI207V1	ISM330DHCX	207V1
STEVAL-MKI208V1K	IIS3DWB	208V1K
STEVAL-MKI209V1K	IIS2ICLX	209V1K
STEVAL-MKI210V2K	ISM330DHCX	210V2
STEVAL-MKI211V1K	LIS25BA	211V1K
STEVAL-MKI212V1	ASM330LHHX	212V1
STEVAL-MKI213V1	LPS27HHW	213V1
STEVAL-MKI214V1	LPS33K	214V1
STEVAL-MKI215V1	LSM6DSO32	215V1
STEVAL-MKI216V1K	IIS3DHHC	216V1
STEVAL-MKI217V1	LSM6DSOX+LIS2MDL	217V1
STEVAL-MKI218V1	AIS2IH	218V1
STEVAL-MKI219V1	LPS22CH	219V1
STEVAL-MKI220V1	LPS27HHTW	220V1
STEVAL-MKI221V1	LSM6DSO32X	221V1
STEVAL-MKI222V1	LISDU12	222V1
STEVAL-MKI223V1K	ILPS28QSW	223V1
STEVAL-MKI224V1	LPS22DF	224A
STEVAL-MKI225A	LPS28DFW	225A
STEVAL-MKI226KA	AIS25BA	226KA
STEVAL-MKI227KA	LSM6DSV16X	227A
STEVAL-MKI228KA	ILPS22QS	228KA
STEVAL-MKI229A	LSM6DSO16IS	229A
STEVAL-MKI230KA	ISM330IS	230KA
STEVAL-MKI231KA	STHS34PF80	231KA
STEVAL-MKI233KA	ISM330ISN	233KA
STEVAL-MKI234KA	LSM6DSV16BX	234KA
STEVAL-MKI235KA	LIS2DUXS12	235KA
STEVAL-MKI236A	2x ASM330LHB	236A
STEVAL-MKI237KA	LSM6DSV16BX	237KA
STEVAL-MKI238A	LIS2DUX12	238A
STEVAL-MKI239A	LSM6DSV	239A
STEVAL-MKI240KA	LSM6DSV32X	240A

Adapter board	Device	*setb command string
STEVAL-MKI241KA	LSM6DSV16B	241KA
STEVAL-MKI242A	ST1VFE6AX	242A
STEVAL-MKI243A	ASM330LHHXG1	243A
STEVAL-MKI244A	2xASM330LHBG1	244A
STEVAL-MKI245KA	ISM330BX	245KA
STEVAL-MKI246KA	IIS2DULPX	246KA
STEVAL-MKI247A	LSM6DSV80X	247A
STEVAL-MKI249KA	ST1VAFE6AX	249KA
STEVAL-MKI250KA	ST1VAFE3BX	250KA

4 Supported commands

The microcontroller mounted on the Professional MEMS Tool board is equipped with dedicated firmware that allows control of the digital output MEMS sensor, and acquisition of the measured data.

The firmware also handles the communication between the board and the PC through the USB bus.

4.1 Getting started

Before using the commands supported by the firmware, the following procedure must be performed:

- Step 1.** Connect the Professional MEMS Tool to the USB port
 - Step 2.** Launch an application that allows sending commands through the virtual serial port. The remainder of this document assumes the use of Microsoft® HyperTerminal program available with the Windows XP operating system, but you can use any similar tool.
 - Step 3.** Create a new connection, enter a name (e.g. STEVAL-MKI109D), and click **OK**.
 - Step 4.** In the **Connect Using** field, select the virtual COM port to which the USB port has been mapped, and click **OK**.
 - Step 5.** In port settings, set **bits per second** to 115200, **data bits** to 8, **parity** to none, **stop bits** to 1, and **flow control** to none; click **OK**
 - Step 6.** In the HyperTerminal application window, select **files > properties > settings**, then click **ASCII Setup**.
 - Step 7.** Select **Send line ends with line feeds** and **Echo typed characters locally**
 - Step 8.** Click **OK** to close the ASCII Setup window
 - Step 9.** Click **OK** button to close the Properties window.
- Once this procedure has been completed you can use the commands described in the following sections by typing them in the "HyperTerminal" window.

4.1.1 Quick start

The basic sequence of commands (based on the LIS3DH accelerometer) to start a data communication session and to retrieve X, Y, and Z acceleration data from the demonstration kit is:

- Step 1.** Connect the Professional MEMS Tool to the USB port.
- Step 2.** Start "Microsoft® HyperTerminal" (or another similar application) and configure it as described in [Section 4.1: Getting started](#).
- Step 3.** Enter the `*setdb105v1` command in the HyperTerminal" window, (supposing the LIS3DH adapter board is used – for other adapters see the relevant datasheets to check the register configuration), enter the command `* Zoff` to enable the control of the device by the STM32F401VE microcontroller, and `*w2047` to switch on the LIS3DH and to set the data rate to 50 Hz.
- Step 4.** Send the `*debug` command to get the X, Y, and Z data measured by the sensor.
- Step 5.** Send `*stop` to end the continuous acquisition and visualization.

4.2 Supported commands

The firmware supports a wide range of MEMS adapters; the complete list of supported commands and their descriptions are given below. Commands are not case sensitive.

Table 2. List of supported commands

Command	Description	Returned value ⁽¹⁾
*setdbXXXVY	Selects firmware according to the adapter connected	Device name e.g.: LIS2DU12
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable text format	
*stop	Stops data acquisition	
*zon	Forces High impedance state. Turns off VDD and VDDIO power supply.	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply	
*dev	Device name	e.g.: LIS2DU12
*ver	Firmware version	e.g.: V1.0.2
*board	Returns board name	
*rAA	Accelerometer register read	e.g.: RAAhDDh
*wAADD	Accelerometer register write	
*grAA	Gyroscope register read	e.g.: GRAAhDDh
*gwAADD	Gyroscope register write	
*mrAA	Magnetometer register read	e.g.: MRAAhDDh
*mwAADD	Magnetometer register write	
*prAAx	Pressure sensor register read	e.g.: PRAAhDDh
*pwAADD	Pressure sensor register write	
*hrAA	Humidity sensor register read	e.g.: HRAAhDDh
*hwAADD	Humidity sensor register write	
*trAA	Temperature sensor register read	e.g.: TRAAhDDh
*twAADD	Temperature sensor register write	
*single	It gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	It gets a single point data acquisition in readable text	
*list	Prints the list of MKIs supported	e.g.: MKI001V1→LPS22HB MKI105V1→LIS3DH MKI110V1→ AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ ...
*echoon	Activates the write verbose mode	e.g.: RAAhDDh
*echooff	Deactivates the write verbose mode	
*fiforst	Accelerometer "Reset mode" enable	st XH XL YH YL ZH ZL IR FC FS
*fifomde	Accelerometer "FIFO mode" enable	st XH XL YH YL ZH ZL IR FC FS
*fifostr	Accelerometer "FIFO stream" enable	st XH XL YH YL ZH ZL IR FC FS
*fifostf	Accelerometer "Stream to FIFO" enable	st XH XL YH YL ZH ZL IR FC FS
*fifobtf	Accelerometer "Bypass to FIFO" enable	st XH XL YH YL ZH ZL IR FC FS

Command	Description	Returned value ⁽¹⁾
*fifobts	Accelerometer "Bypass to stream" enable	st XH XL YH YL ZH ZL IR FC FS
*fifodstr	Accelerometer "Dynamic stream" enable	st XH XL YH YL ZH ZL IR FC FS
*gfiforst	Gyroscope "Reset mode" enable	st XH XL YH YL ZH ZL IR FC FS
*gfifomde	Gyroscope "FIFO mode" enable	st XH XL YH YL ZH ZL IR FC FS
*gfifostr	Gyroscope "FIFO stream" enable	st XH XL YH YL ZH ZL IR FC FS
*gfifostf	Gyroscope "Stream to FIFO" enable	st XH XL YH YL ZH ZL IR FC FS
*gfifobtf	Gyroscope "Bypass to FIFO" enable	st XH XL YH YL ZH ZL IR FC FS
*gfifobts	Gyroscope "Bypass to stream" enable	st XH XL YH YL ZH ZL IR FC FS
*gfifodstr	Gyroscope "Dynamic stream" enable	st XH XL YH YL ZH ZL IR FC FS
*pfiforst	Pressure sensor "Reset mode" enable	st XH XL YH YL ZH ZL IR FC FS
*pfifomde	Pressure sensor "FIFO mode" enable	st XH XL YH YL ZH ZL IR FC FS
*pfifostr	Pressure sensor "FIFO stream" enable	st XH XL YH YL ZH ZL IR FC FS
*pfifostf	Pressure sensor "Stream to FIFO" enable	st XH XL YH YL ZH ZL IR FC FS
*pfifobtf	Pressure sensor "Bypass to FIFO" enable	st XH XL YH YL ZH ZL IR FC FS
*pfifobts	Pressure sensor "Bypass to stream" enable	st XH XL YH YL ZH ZL IR FC FS
*pfifodstr	Pressure sensor "Dynamic stream" enable	st XH XL YH YL ZH ZL IR FC FS
*rmAA ₁ NN	Multiple read of NN Accelerometer successive registers	RMAA ₁ hNNhDD ₁ hDD ₂ ...DD _{NN} h
*mutli-rAA ₁ AA ₂ AA ₃ ...	Accelerometer registers multiple read	MULTI-RAA ₁ hDD ₁ h AA ₂ hDD ₂ h.... AAnDDn
*grmAA ₁ NN	Multiple read of NN Gyroscope successive registers	GRMAA ₁ hNNhDD ₁ hDD ₂ ...DD _{NN} h
*multi-grAA ₁ AA ₂ AA ₃ ...	Gyroscope registers multiple read	MULTI-GRAA ₁ hDD ₁ h AA ₂ hDD ₂ h.... AAnDDn
*mrmAA ₁ NN	Multiple read of NN Magnetometer successive registers	MRMAA ₁ hNNhDD ₁ hDD ₂ ...DD _{NN} h
*multi-mrAA ₁ AA ₂ AA ₃ ...	Magnetometer registers multiple read	MULTI-MRAA ₁ hDD ₁ h AA ₂ hDD ₂ h.... AAnDDn
*prmAA ₁ NN	Multiple read of NN Pressure sensor successive registers	PRMAA ₁ hNNhDD ₁ hDD ₂ ...DD _{NN} h
*multi-prAA ₁ AA ₂ AA ₃ ...	Pressure sensor registers multiple read	MULTI-PRAA ₁ hDD ₁ h AA ₂ hDD ₂ h.... AAnDDn
*hrmAA ₁ NN	Multiple read of NN Humidity sensor successive registers	HRMAA ₁ hNNhDD ₁ hDD ₂ ...DD _{NN} h
*multi-hrAA ₁ AA ₂ AA ₃ ...	Humidity sensor registers multiple read	MULTI-HRAA ₁ hDD ₁ h AA ₂ hDD ₂ h.... AAnDDn
*trmAA ₁ NN	Multiple read of NN Temperature sensor successive registers	TRMAA ₁ hNNhDD ₁ hDD ₂ ...DD _{NN} h
*multi-trAA ₁ AA ₂ AA ₃ ...	Temperature sensor registers multiple read	MULTI-TRAA ₁ hDD ₁ h AA ₂ hDD ₂ h.... AA _n DD _n h
*odr [param]	Sets ODR speed	

1. RP: Reference pressure XLSB.MSB, IR: interrupt byte; FC: FIFO control register; FS: FIFO source register

Table 3. List of supported commands for power supply control and I/V measurement

Command	Description	Returned value
*power_on	Turns on VDD and VDDIO power supply	
*power_off	Turns off VDD and VDDIO power supply	
*set_vddaV.VVV	Sets VDD voltage value "V.VVV" Volts. Value defined by 1 or 2 decimal places is possible. e.g. 1.8; 1.86; 1.825 are all valid inputs	
*set_vddioV.VVV	Sets VDDIO voltage value "V.VVV" Volts. Value defined by 1 or 2 decimal places is possible. e.g. 1.8; 1.86; 1.825 are all valid inputs	
*get_i [param]	Measures IDD, IDDIO with automatic / semiautomatic current ranges toggling and measurement method selection. Output values format selected by parameters [F / B / A] [S] [C] [L]	e.g. param = A ida Idd = 152 μ A IddIO = 0.092 μ A
*get_v [param]	Measures VDD, VDDIO Output values format selected by parameters [F/B/A]	e.g. param = A vda Vdd = 1.807 V VddIO = 1.806 V
*ranges_auto	Turns on automatic IDD/IDDIO ranges selection (default)	
*ranges_man	Turns off automatic IDD/IDDIO ranges selection (manual IDD/IDDIO ranges selection possible)	
*range_ma [param]	Selects current range for IDD	
*range_mb [param]	Selects current range for IDDIO	
*adc_run_time [param]	Sets time period during which ADC samples the measured current waveforms	
*adc_uni_stop	Stops all measurements	

The commands listed in Table 3 work regardless of whether an adapter is physically inserted in the DIL24 socket or not, and also whether the STEVAL-MKlxxxxx of the adapter is selected or not (using the *Setdb command).

4.2.1 *setdbXXXVY

This command selects the part of the firmware able to handle the adapter board sensor connected to the board. For example, *setdb105V1 selects the firmware for the LIS3DH.

The D3 LED (yellow) switches on automatically.

4.2.2 *start

This command initiates continuous data acquisition. When sent, the device returns a string of bytes (plus carriage return and line feed) like `st OUT1 OUT2 OUT3 IR STP BT`.

The first two bytes are always the ASCII char `s` and `t` which correspond to the hexadecimal values {73h 74h}.

OUT1, OUT2, and OUT3 contain the values measured at device outputs; if the output data is represented in more than 8 bits, OUT1, OUT2, and OUT3 are split into high byte (e.g., XH) and low byte (e.g., XL). In case of 24-bit resolution for some sensors, there is also an extra-low byte (e.g., pressure data: PXL PL PH).

IR (INT1 INT2) contains the interrupt bytes and BT SW1|SW2 contains the bytes that describe the state of the buttons integrated on the board.

Specifically, bit#0 of the SW1|SW2 data corresponds to the status of the SW1 button on the demonstration kit board: it is set to 1 when the SW1 is pressed (otherwise 0). Bit#1 has the same behavior but is dedicated to the SW2.

STP (STPL STPH) contains the step counter bytes for the internal device step counter value.

The string is ended with the carriage return (`\r`) and line feed (`\n`) bytes.

Before sending the *start command, the device must be out of 3-state (high impedance) and some registers must be configured according to user needs. Therefore, *start must be preceded by a *zoff and some Register Write commands.

As data is continuously acquired, LED D1 (green) blinks according to sensor data rate selected.

Table 4. Returned values for *start command shows the format of the string returned for each device when a *start command is sent. Similar byte strings are returned for groups of commands related to FIFO, as is shown in Table 5. Digital output accelerometers: supported commands list, Section 4.4: Digital output gyroscopes: supported commands, Section 4.5: Digital output magnetometers: supported commands, Table 8. Digital output pressure sensor: supported commands list, Table 9. Digital output temperature sensor: supported commands list and Table 10. Analog output temperature sensor: supported commands list.

Table 4. Returned values for *start command

STEVAL # (device)	Returned value ⁽¹⁾
STEVAL-MKI105V1 (LIS3DH) STEVAL-MKI110V1 (AIS328DQ) STEVAL-MKI125V1 (A3G4250D) STEVAL-MKI135V1 (LIS2DH) STEVAL-MKI151V1 (LIS2DH12) STEVAL-MKI158V1 (AIS3624DQ) STEVAL-MKI164V1 (LIS2HH12) STEVAL-MKI168V1 (IIS2DH) STEVAL-MKI170V1 (IIS328DQ) STEVAL-MKI179V1 (LIS2DW12) STEVAL-MKI180V1 (LIS3DHH) STEVAL-MKI186V1 (IIS3DHHC) STEVAL-MKI191V1 (IIS2DLPC) STEVAL-MKI206V1 (AIS2DW12) STEVAL-MKI216V1 (IIS3DHHC) STEVAL-MKI218V1 (AIS2IH)	st XH XL YH YL ZH ZL int1 int2 sw1 sw2 \r \n
STEVAL-MKI175V1 (LIS2DE12)	st X Y Z int1 int2 sw1 sw2 \r \n
STEVAL-MKI166V1 (H3LIS100DL) STEVAL-MKI167V1 (H3LIS200DL)	st X 0 Y 0 Z 0 int1 int2 sw1 sw2 \r \n
STEVAL-MKI208V1K (IIS3DWB)	st XH XL YH YL ZH ZL int1 sw1 sw2 \r \n
STEVAL-MKI174V1 (LIS2DS12)	st XH XL YH YL ZH ZL int1 int2 stpL stpH 0 sw1 sw2 \r \n
STEVAL-MKI137V1 (LIS3MDL) STEVAL-MKI181V1 (LIS2MDL) STEVAL-MKI185V1 (IIS2MDC)	st M_XH M_XL M_YH M_YL M_ZH M_ZL int1 sw1 sw2 \r \n
STEVAL-MKI169V2 (I3G4250D)	st G_XH G_XL G_YH G_YL G_ZH G_ZL G_int1 G_int2 sw1 sw2 \r \n
STEVAL-MKI159V1 (LSM9DS1)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL M_XH M_XL M_YH M_YL M_ZH M_ZL A_int1 G_int2 G_int3 0 sw1 sw2 \r \n
STEVAL-MKI178V2 (LSM6DSL) STEVAL-MKI182V1 (ISM330DLC) STEVAL-MKI189V1 (LSM6DSM)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL Int1 Int2 StpL StpH 0 sw1 sw2 \r \n
STEVAL-MKI163V1 (LSM303C)	st A_XH A_XL A_YH A_YL A_ZH A_ZL M_XH M_XL M_YH M_YL M_ZH M_ZL A_intG_int sw1 sw2 \r \n
STEVAL-MKI172V1 (LSM303AGR)	st A_XH A_XL A_YH A_YL A_ZH A_ZL M_XH M_XL M_YH M_YL M_ZH M_ZL A_int1Aint2 M_int sw1 sw2 \r \n

STEVAL # (device)	Returned value ⁽¹⁾
STEVAL-MKI173V1 (LSM303AH) STEVAL-MKI184V1 (ISM303DAC)	st A_XH A_XL A_YH A_YL A_ZH A_ZL M_XH M_XL M_YH M_YL M_ZH M_ZL A_int1A_int2 M_int StpL StpH sw1 sw2 \r \n
STEVAL-MKI142V1 (LPS25H) STEVAL-MKI165V1 (LPS25HB)	st PXL PL PH TL TH REF_PXL REF_PL REF_PH P_int1sw1 sw2 \r \n
STEVAL-MET001V1 (LPS22HB) STEVAL-MKI183V1 (LPS33HW) STEVAL-MKI192V1 (LPS22HH) STEVAL-MKI205V1 (LPS33W) STEVAL-MKI213V1 (LPS27HHW) STEVAL-MKI214V1 (LPS33K) STEVAL-MKI219V1 (LPS22CH) STEVAL-MKI220V1 (LPS27HHTW) STEVAL-MKI224V1 (LPS22DF) STEVAL-MKI225A (LPS28DFW)	st PXL PL PH TL TH REF_PXL REF_PL REF_PH P_int1 sw1 sw2 \r \n
STEVAL-MKI190V1 (LIS2DTW12) STEVAL-MKI222V1 (LISDU12) STEVAL-MKI238A (LIS2DUX12) STEVAL-MKI246KA (IIS2DULPX)	st A_XH A_XL A_YH A_YL A_ZH A_ZL Int1 Int2 TL TH sw1 sw2 \r \n
STEVAL-MKI198V1K (STTS751)	st TH TL Int1 sw1 sw2 \r \n
STEVAL-MKI199V1K (STLM20)	st TH TL sw1 sw2 \r \n
STEVAL-MKI200V1K (STTS22H) STEVAL-MKI201V1K (STTS75) STEVAL-MKI202V1K (STDS75) STEVAL-MKI203V1K (STCN75) STEVAL-MKI204V1K (STLM75)	st TH TL Int1 sw1 sw2 \r \n
STEVAL-MKI194V1 (LSM6DSR) STEVAL-MKI196V1 (LSM6DSO) STEVAL-MKI215V1 (LSM6DSO32) STEVAL-MKI221V1 (LSM6DSO32X)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL Int1 Int2 FSMLC_INT FSM1 FSM2 StpL StpH TL TH sw1 sw2 \r \n
STEVAL-MKI211V1K (LIS25BA) STEVAL-MKI226A (AIS25BA)	st XH XL YH YL ZH ZL sw1 sw2 \r \n
STEVAL-MKI195V1 (LSM6DSRX) STEVAL-MKI197V1 (LSM6DSOX) STEVAL-MKI207V1 (ISM330DHCX) STEVAL-MKI212V1 (ASM330LHHX) STEVAL-MKI243A (ASM330LHHXG1)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL Int1 Int2 FSMLC_INT FSM1 FSM2 StpL StpH MLC0 MLC1 MLC2 MLC3 MLC4 MLC5 MLC6 MLC7 TL TH sw1 sw2 \r \n
STEVAL-MKI209V1 (IIS2ICLX)	st A_XH A_XL A_YH A_YL Int1 Int2 FSMLC_INT FSM1 FSM2 StpL StpH MLC0 MLC1 MLC2 MLC3 MLC4 MLC5 MLC6 MLC7TL TH sw1 sw2 \r \n
STEVAL-MKI217V1 (LSM6DSOX + LIS2MDL hub)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL M_XH M_XL M_YH M_YL M_ZH M_ZL Int1 Int2FSMLC_INT FSM1 FSM2 MGN_INT StpL StpH MLC0 MLC1 MLC2 MLC3 MLC4 MLC5 MLC6 MLC7 0 sw1 sw2 \r \n

STEVAL # (device)	Returned value ⁽¹⁾
STEVAL-MKI228KA (ILPS22QS)	stPXL PL PH TL TH 0 REF_PL REF_PH SGNB QXL QL QH INT1 sw1 sw2 \r \n
STEVAL-MKI193V1 (ASM330LHH) STEVAL-MKI229A (LSM6DSO16IS) STEVAL-MKI230KA (ISM330IS)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL Int1 Int2TL TH sw1 sw2 \r \n
STEVAL-MKI231KA (STHS34PF80)	st Tobj_H Tobj_L Tobj_comp_H Tobj_comp_L Tambient_H Tambient_L Tpresence_H Tpresence_L Tmotion_H Tmotion_L Tambient_shock_H Tambient_shock_L Int1 Int_status sw1 sw2 \r \n
MKI234KA (LSM6DSV16BX) MKI241KA (LSM6DSV16B) MKI245KA (LSM5DSV16BX) MKI245KA (ISM330BX)	Ulstring: s t A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL QH QL Q_FH Q_FL A_Int1 A_Int2 FSMLC_INT FSM1 StpL StpH DT1 DT2 DT3 DT4 TL TH sw1 sw2 \r \n TDMstring: t d A_XH A_XL A_YH A_YL A_ZH A_ZL 'T' \r \n SFLPstring: s t A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL Q0_L Q0_H Q1_L Q1_H Q2_L Q2_H gbiasX_H gbiasX_L gbiasY_H gbiasY_L gbiasZ_H gbiasZ_L gravX_H gravX_L gravY_H gravY_L gravZ_L gravZ_H \r \n Ulstring + time: s t A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL QH QL Q_FH Q_FL A_Int1 A_Int2FSMLC_INT FSM1 StpL StpH DT1 DT2 DT3 DT4 TL TH sw1 sw2 TS_XL TS_L TS_H TS_XH \r \n TDMstring+time: t d A_XH A_XL A_YH A_YL A_ZH A_ZL 'T' TS_XL TS_L TS_H TS_XH \r \n SFLPstring+time: s t A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL Q0_L Q0_H Q1_L Q1_H Q2_L Q2_H gbiasX_H gbiasX_L gbiasY_H gbiasY_L gbiasZ_H gbiasZ_L gravX_H gravX_L gravY_H gravY_L gravZ_L gravZ_H TS_XL TS_L TS_H TS_XH \r \n
MKI236A (2x ASM330LHB)	st A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL A2_XH A2_XL A2_YH A2_YL A2_ZH A2_ZL 2G_XH G2_XL G2_YH G2_YL G2_ZH G2_ZL A_Int1 A_Int2 FSMLC_INT FSM1 FSM2 StpL StpH DT1 DT2 DT3 DT4DT5 DT6 DT7 DT8 TL TH sw1 sw2 \r \n
MKI239A(LSM6DSV)	Ulstring: s t A_XH A_XL A_YH A_YL A_ZH A_ZL G_XH G_XL G_YH G_YL G_ZH G_ZL A_Int1 A_Int2 FSMLC_INT FSM1 StpL StpH DT1 DT2 DT3 DT4 TL TH sw1 sw2 'U' \r \n OISstring: s t A_XH A_XL A_YH A_YL A_ZH A_ZL 'O' \r \n
MKI232A (LSM6DSO16ISN) MKI233KA (ISM330ISN) MKI249 MKI250	Outputstring is defined by Unico / MEMS studio. See proper documentation of device setting with .UCF file

1. XH: X-axis output high byte (same for Y axis, Z axis, G gyroscope, M Magnetometer, P pressure, H humidity, T temperature, Stp step counter). XL: X-axis output low byte (same for Y axis, Z axis, G gyroscope, M Magnetometer, P pressure, H humidity, T temperature, Q Qvar data, Stp step counter). FSMx: Finite state machine bytes. MLCx: Machine Learning core source registers bytes. DTx : data bytes. TSxy:time stamp data bytes. sw1|sw2: User buttons BT1, BT2 state.

4.2.3

*debug

This command starts continuous data acquisition in debug mode. When sent to the board, it returns the output values measured by the device formatted in a readable text format.

4.2.4 ***stop**

This command interrupts any acquisition session that has been started with either the `*start` or `*debug` commands.

4.2.5 ***zon and *zoff**

These commands put the STM32F401VE microcontroller on the demonstration kit in 3-state (high impedance). They allow the isolation of the sensor from the microprocessor and let the user interact with the sensor in a purely analog fashion.

When the kit is first turned on, the lines are in 3-state (high impedance) mode and you must send the `*zoff` command to allow communication between the sensor and the microcontroller.

Note that the `Zoff` command also internally calls `*power_on` command which enables sensor Vdd and Vddio at default level defined internally in FW. In case user wants to change Vdd and Vddio levels he has to use `*set_vdda` and `*set_vddio` commands with proper values. See also example below.

After this command has been executed, LED D2 (Red) is turned on. If the `*zoff` command has not been launched, the firmware ignores any other command sent to sensor.

Correct example of `*zoff` command folowed by different supply voltage setting is following :

```
*setdb MKI181V1
*zoff // Default Vdd=2.5V Vddio = 2.5V are set
*set_vdda3.6 //settling Vdda to 3.6V
*set_vddio2.8 /settling Vddio to 1.8V
...
*gr4f // read whoami to check correct sensor is present
...
```

4.2.6 ***dev**

This command retrieves the name of the adapter connected to the demonstration kit; e.g., LIS3DH.

4.2.7 ***ver**

This command returns the version of the firmware loaded in the microprocessor; e.g., V1.5.2.

4.2.8 ***rAA**

This command reads the contents of the accelerometer registers in the demonstration kit board. The hexadecimal value `AA` written in upper case represents the address of the register to be read.

Once the read command is issued, the board returns `RAAhDDh`, where `AA` is the address sent by the user and `DD` is the data present in the register.

For example, to read the register at address 0x20, the user issues the command `*r20`, which would return a result like `R20hC7h`.

4.2.8.1 ***rmAA1NN**

This command allows the contents of multiple accelerometer registers in the demonstration kit board to be read in single data block. Once this command is issued, the board returns a set of `NN` values starting with `RMAA1hNNhDD1hDD2h... DDNNh` where `AA1` is the starting address set by user and `DD1` is the data present in this register and so on for next registers.

For example, `*rm2006` reads six registers starting from 0x20, which would return a result like `RM20h06h27h00h00h00hA0h0Bh`.

4.2.8.2 ***multi-rAA1AA2AA3...AAN**

This command reads multiple accelerometer registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of `N` values starting `RAA1hDD1h... AANhDDNh` where `AA1` is the starting address set by user and `DD1` is the data present in this register.

For example, `*multi-r202425292B2D` reads six register starting from 0x20, which would return a result like `MULTI-R20h27h24hA0h25h0Bh29hE0h2Bh3Fh2Dh90h`.

4.2.9 ***wAADD**

This command writes the contents of the accelerometer registers in the demonstration kit board. The hexadecimal upper case values **AA** and **DD** represent the address of the register and the data to be written, respectively.

For example, ***w20C7** writes 0xC7 to the register at address 0x20

4.2.10 ***grAA**

This command allows the contents of the gyroscope registers in the demonstration kit board to be read. The hexadecimal, upper case **AA** represents the address of the register to be read.

Once the read command is issued, the board returns **GRAAhDDh**, where **AA** is the address sent by the user and **DD** is the data present in the register.

For example, ***gr20** reads the register at address 0x20, which would return a result like **GR20hC7h**.

4.2.10.1 ***grmAA1NN**

This command allows the contents of multiple gyroscope registers in the demonstration kit board to be read in single data block. Once this command is issued, the board returns set of **NN** values starting with **GRMAA₁hNNhDD₁h... DD_{NN}h** where **AA₁** is the starting address set by user and **DD₁** is the data present in this register and so on.

For example, ***grm2006** reads six registers starting from 0x20, which would return a result like **GRM20h06h27h00h00h00hA0h0Bh**.

4.2.10.2 ***multi-grAA1AA2AA3...AAN**

This command reads multiple gyroscope registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of **N** values starting with **MULTI-GRAA₁hDD₁h... AA_NhDD_Nh** where **AA₁** is the starting address set by user and **DD₁** is the data present in this register and so on.

For example, ***multi-gr202425292B2D** reads six registers starting from 0x20, which would return a result like **MULTI-GR20h27h24hA0h25h0Bh29hE0h2Bh3Fh2Dh90h**.

4.2.11 ***gwAADD**

This command writes the contents of the gyroscope registers in the demonstration kit board. The hexadecimal, upper case. **AA** and **DD** represent the address of the register and the data to be written, respectively.

For example, ***gw20C7** writes 0xC7 to the register at address 0x20.

4.2.12 ***mrAA**

This command allows the contents of the magnetometer registers in the demonstration kit board to be read. The hexadecimal, upper case **AA** represents the address of the register to be read.

Once the read command is issued, the board returns **MRAAhDDh**, where **AA** is the address sent by the user and **DD** is the data present in the register.

For example, ***mr00** reads the register at address 0x00, which would return a result like **MR00h10h**.

4.2.12.1 ***mrmAA1NN**

This command readss the contents of multiple magnetometer registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of **NN** values starting with **RMAA₁hNNhDD₁hDD₂h... DD_{NN}h** where **AA₁** is the starting address set by user and **DD₁** is the data present in this register.

For example, ***mrm2006** reads six register starting from 0x20, which would return a result like **MRM20h06h27h00h00h00hA0h0Bh**.

4.2.12.2 ***multi-mrAA1AA2AA3...AAN**

This command reads multiple magnetometer registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of **N** values starting with **MULTI-MRAA₁hDD₁h... AA_NhDD_Nh**, where **AA₁** is the starting address set by user and **DD₁** is the data present in this register, and so on.

For example, ***multi-mr202425292B2D** reads six registers starting from 0x20, which would return a result like **MULTI-MR20h27h24hA0h25h0Bh29hE0h2Bh3Fh2Dh90h**.

4.2.13 ***mwAADD**

This command writes the contents of the magnetometer registers in the demonstration kit board. Hexadecimal, upper case AA and DD represent the address of the register and the data to be written, respectively.

For example, *mw0120 writes 0x20 to the register at address 0x01.

4.2.14 ***prAA**

This command reads the contents of the pressure sensor registers in the demonstration kit board. The hexadecimal, upper case AA represents the address of the register to be read.

Once the read command is issued, the board returns PRAAhDDh, where AA is the address sent by the user and DD is the data present in the register.

For example, *pr20 reads the register at address 0x20, which would return a value like PR20h10h.

4.2.14.1 ***prmAA1NN**

This command reads the contents of multiple pressure sensor registers in the demonstration kit in a single data block. Once this command is issued, the board returns set of NN values starting with PRMAA₁hNNhDD₁hDD₂h... DD_{NN}h where AA₁ is the starting address set by user and DD₁ is the data present in this register, and so on...

For example, *prm2006 reads six registers starting from 0x20, which would return a value like PRM20h06h27h00h00h00hA0h0Bh.

4.2.14.2 ***multi-prAA1AA2AA3...AAN**

This command reads multiple pressure sensor registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of N values starting with MULTI-PRAA₁hDD₁h... AA_NhDD_Nh where AA₁ is the starting address set by user and DD₁ is the data present in this register, and so on.

For example, *multi-pr202425292B2D reads six registers starting from 0x20, which would return a result like MULTI-PR20h27h24hA0h25h0Bh29hE0h2Bh3Fh2Dh90h.

4.2.15 ***pwAADD**

This command writes the contents of the pressure sensor registers in the demonstration kit board. The hexadecimal, upper case AA and DD represent the address of the register and the data to be written, respectively.

For example, *pw20C7 writes 0xC7 to the register at address 0x20.

4.2.16 ***hrAA**

This command reads the contents of the humidity sensor registers in the demonstration kit board. The hexadecimal, upper case AA represents the address of the register to be read.

Once the read command is issued, the board returns HRAAhDDh, where AA is the address sent by the user and DD is the data present in the register.

For example, *hr20 reads the register at address 0x20, which would return a result like HR20h10h.

4.2.16.1 ***hrmAA1NN**

This command reads the contents of multiple humidity sensor registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of NN values starting with HRMAA₁hNNhDD₁hDD₂h... DD_{NN}h where AA₁ is the starting address set by user and DD₁ is the data present in this register, and so on.

For example, *hrm2006 reads six registers starting from 0x20, which would return a result like HRM20h06h27h00h00h00hA0h0Bh.

4.2.16.2 ***multi-hrAA1AA2AA3...AAN**

This command reads multiple humidity sensor registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of N values starting with MULTI-HRAA₁hDD₁h... AA_NhDD_Nh where AA₁ is the starting address set by user and DD₁ is the data present in this register, and so on.

For example, *multi-hr202425292B2D reads six registers starting from 0x20, which would return a result like MULTI-HR20h27h24hA0h25h0Bh29hE0h2Bh3Fh2Dh90h.

4.2.17 ***hwAADD**

This command writes the contents of the humidity sensor registers in the demonstration kit board. The hexadecimal, upper case **AA** and **DD** represent the address of the register and the data to be written, respectively. For example ***hw20C7** writes 0xC7 to the register at address 0x20.

4.2.18 ***trAA**

This command reads the contents of the temperature sensor registers in the demonstration kit board. The hexadecimal, upper case **AA** represents the address of the register to be read. Once the read command is issued, the board returns **TRAAhDDh**, where **AA** is the address sent by the user and **DD** is the data present in the register. For example, ***tr20** reads the register at address 0x20, which would return a result like **TR20h10h**.

4.2.18.1 ***trmAA1NN**

This command reads the contents of multiple temperature sensor registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of **NN** values starting with **TRMAA₁hNNhDD₁hDD₂h... DD_{NN}h** where **AA₁** is the starting address set by user and **DD₁** is the data present in this register, and so on. For example, ***trm2006** reads six registers starting from 0x20, which would return a result like **TRM20h06h27h00h00h00hA0h0Bh**.

4.2.18.2 ***multi-trAA1AA2AA3...AAN**

This command reads multiple temperature sensor registers in the demonstration kit board in a single data block. Once this command is issued, the board returns set of **N** values starting with **MULTI-TRAA₁hDD₁h... AANhDDN_h** where **AA₁** is the starting address set by user and **DD₁** is the data present in this register, and so on. For example, ***multi-tr202425292B2D** reads six registers starting from 0x20, which would return a result like **MULTI-TR20h27h24hA0h25h0Bh29hE0h2Bh3Fh2Dh90h**.

4.2.19 ***twAADD**

This command writes the contents of the temperature sensor registers in the demonstration kit board. The hexadecimal, upper case **AA** and **DD** represent the address of the register and the data to be written, respectively. For example ***tw20C7** writes 0xC7 to the register at address 0x20.

4.2.20 ***single**

This command may be used to read just one set of data. It returns the read values of one data sample if the sensor is configured properly.

4.2.21 ***sindebug**

This command returns a single point data acquisition in readable text ASCII format.

4.2.22 ***list**

The command returns the list of the MKI adapters and devices supported by the firmware in ASCII format.

4.2.23 ***listdev**

This command returns the list of devices supported by the firmware in ASCII format.

4.2.24 ***echoon**

This command is used to activate the write command verbose mode so that the firmware automatically reads the contents of a register that has just been written to check if the write was successful.

For example, ***echoon** launched after ***w2027** returns **R2027**.

4.2.25 ***echooff**

This command stops the write command verbose mode.

- 4.2.26 *fiforst**
This command enables the accelerometer FIFO reset mode. For more details see application note AN3308 on www.st.com.
- 4.2.27 *fifomde**
This command enables the accelerometer FIFO mode. For more details see application note AN3308 on www.st.com.
- 4.2.28 *fifostr**
This command enables the accelerometer FIFO stream mode. For more details see application note AN3308 on www.st.com.
- 4.2.29 *fifostf**
This command enables the accelerometer Stream-to-FIFO mode. For more details see application note AN3308 on www.st.com.
- 4.2.30 *fifobtf**
This command enables the accelerometer Bypass-to-FIFO mode.
- 4.2.31 *fifobts**
This command enables the accelerometer Bypass-to-Stream mode.
- 4.2.32 *fifodstr**
This command enables the accelerometer Dynamic Stream mode.
- 4.2.33 *gfiforst**
This command enables the gyroscope FIFO reset mode.
- 4.2.34 *gfifomde**
This command enables the gyroscope FIFO mode.
- 4.2.35 *gfifostr**
This command enables the gyroscope FIFO stream mode.
- 4.2.36 *gfifostf**
This command enables the gyroscope Stream-to-FIFO mode.
- 4.2.37 *gfifobtf**
This command enables the gyroscope Bypass-to-FIFO mode.
- 4.2.38 *gfifobts**
This command enables the gyroscope Bypass-to-Stream mode.
- 4.2.39 *gfifodstr**
This command enables the gyroscope Dynamic Stream mode.
- 4.2.40 *pfiforst**
This command enables the pressure sensor FIFO reset mode.
- 4.2.41 *pfifomde**
This command enables the pressure sensor FIFO mode.

- 4.2.42 *pfifostr**
This command enables the pressure sensor FIFO stream mode.
- 4.2.43 *pfifostf**
This command enables the pressure sensor Stream-to-FIFO mode.
- 4.2.44 *pfifobtf**
This command is used to enable the pressure sensor Bypass-to-FIFO mode.
- 4.2.45 *pfifobts**
This command is used to enable the pressure sensor Bypass-to-Stream mode.
- 4.2.46 *pfifodstr**
This command enables the pressure sensor Dynamic Stream mode.
- 4.2.47 *set_vddaV.VVV and *set_vddioV.VVV**
These commands set the power supply VDD and VDDIO voltage values of device adapter.
For example, `*set_vdda3.6` sets 3.6V on VDDA. Up to 3 decimal places can be used for set of V.VVV voltage value. If `*set_vdda` and `*set_vddio` have not been sent before the `*power_on` command, the setting is the default voltage defined in the device datasheet selected during the `*setdb` initialization. Please refer to device datasheet regarding specified VDD and VDDIO values and their relationship (i.e., which of these can be higher, etc.).
As the maximum voltages are defined also by the `*setdb` command, you cannot set values above these limits.
- 4.2.48 *power_on and *power_off**
These commands are used to switch on and to switch off the VDD and VDDIO power supplies of the device adapter together. The proper power-on sequence of both voltages is handled internally by firmware.
Note that command `*power_on` must be sent after commands `*setvddaV.VVV` and `*setvddioV.VVV` otherwise it sets voltages to 0V (default).
Correct example of `*power_on` and `power_off` commands is following:

```
*setdb181V1
*zoff // Default Vdd=2.5V Vddio = 2.5V are set
*power_off // physical switch-off of sensor
*set_vdda3.6 //settling Vdda to 3.6V
*set_vddio2.8 /settling Vddio to 1.8V
*power_on // physical switch-on of sensor
...
*gr4f // read whoami to check correct sensor is present
...
```
- 4.2.49 *odr [p1]**
This command is used to set ODR speed when data acquisition is driven by TIMER (analog sensors).
 - P1** ODR in mHz: 1 to 2000000
- 4.2.50 *get_i [p1] [p2] [p3] [p4]**
This command is used to get average IDD and IDDIO during set/calculated time interval. Current ranges toggling and measurement method selection is done automatically as default setup.
 - P1** form of output data:
 - F** or **f** - floating-point (1 µA base)
 - B** or **b** - fixed-point (1 nA base)
 - A** or **a** - ASCII (readable text)

- **P2** status bits:
 - **S** or **s** - Two bytes containing information (overflow, underflow, current range index) will be added at the end of the IDD, IDDIO data
- **P3** continuous (repetitive) measurement:
 - **C** or **c** - firmware will send measured values periodically
- **P4** Long Time Enable:
 - **L** or **l** firmware will wait for waveform edges or Drdy pulses until it appears or until another *get_i or *adc_uni_stop command interrupts it

4.2.51 ***get_v [p1]**

This command is used to get average VDD and VDDIO VddIO (precise measurement with calibrations).

- **P1** form of output data:
 - **F** or **f** - floating-point (1 V base)
 - **B** or **b** - fixed-point (1 mV base)
 - **A** or **a** - ASCII (readable text)

4.2.52 ***ranges_auto and *ranges_man**

These commands are used to toggle between automatic and manual current ranges selection. The default command is *ranges_auto.

4.2.53 ***range_ma [p1] and *range_mb [p1]**

These commands are used to select current ranges manually.

- *range_ma ... IDD, *range_mb ... IDDIO
- **P1** current range: 5, 10, 25, 50, 100, 200, 500, 1000, 2500, 5000, 10000, 20000, 50000, 100000
- Full scale of the selected range = 300000 / P1 [μA]

4.2.54 ***adc_run_time [p1]**

This command is used to set the time interval during which ADC samples the measured current waveforms.

- **P1** time in milliseconds: 1 to 50000 (default is 500 ms)

4.2.55 ***adc_uni_stop**

This command is used to stop all measurements.

4.3 Digital output accelerometers: supported commands

Table 5. Digital output accelerometers: supported commands list

Command	Description	Returned value ⁽¹⁾
*setdbXXXVY	Selects firmware according to the adapter connected	
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable text format	
*stop	Stops data acquisition	
*zon ⁽²⁾	Forces High impedance state. Turns off VDD and VDDIO power supply	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply, configures IO communication type and sets special registers if needed	
*dev	Device name	e.g.: LIS2DU12
*ver	Firmware version	e.g.: V1.0.2
*rAA	Accelerometer register read	e.g.: RAAhDDh
*rmAA ₁ NN	Multiple accelerometer registers read	e.g.: RMAA ₁ hNNhDD ₁ h...DD _{NN} h
*multi-rAA ₁ .. AA _N	Multiple accelerometer registers read	e.g.: MULTI-RAA ₁ hDD ₁ h...AA _N hDD _N h
*wAADD	Accelerometer register write	
*single	It gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	It gets a single point data acquisition in readable text format	
*list	Prints the list of MKIs supported	e.g.: MKI001V1→LPS22HB MKI105V1→ LIS3DH MKI110V1→ AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ ...
*echoon	Activates the write verbose mode	e.g.: RAAhDDh
*echooff	Deactivates the write verbose mode	
*fiforst ⁽³⁾	Accelerometer "Reset mode" enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*fifomde ⁽³⁾	Accelerometer "FIFO mode" enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*fifostr ⁽³⁾	Accelerometer "FIFO stream" enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*fifostf ⁽³⁾	Accelerometer "Stream-to-FIFO" enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*fifobtf ⁽³⁾	Accelerometer "Bypass-to-FIFO" enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*fifobts ⁽³⁾	Accelerometer "Bypass-to-Stream" enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*fifodstr ⁽³⁾	Accelerometer "Dynamic Stream" enable	st XH XL YH YL ZH ZL IR FC FS \r \n

1. IR: interrupt bytes; FC: FIFO control register; FS: FIFO source register

2. Correctly turns off sensor, deinitializes communication with sensor, forces High impedance state and turns off VDD and VDDIO power supply

3. Available only for devices with embedded FIFO

4.4 Digital output gyroscopes: supported commands

The table below lists the commands supported by the devices/demonstration boards including a digital output gyroscope.

Table 6. Digital output gyroscopes: supported commands list

Command	Description	Returned value ⁽¹⁾
*setdbXXXVY	Selects FW according to the adapter connected	
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable text format	
*stop	Stops data acquisition	
*zon	Forces High impedance state. Turns off VDD and VDDIO power supply	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply.	
*dev	Device name	e.g.: I3G4250D
*ver	Firmware version	e.g.: V1.0.2
*grAA	Gyroscope register read	e.g.: GRAAhDDh
*grmAA ₁ NN	Multiple gyroscope registers read	e.g.: GRMAA ₁ hNNhDD ₁ h...DD _{NN} h
*multi-grAA ₁ .. AA _N	Multiple gyroscope registers read	e.g.: MULTI-GRAA ₁ hDD ₁ h...AA _N hDD _N h
*gwAADD	Gyroscope register write	
*single	It gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	It gets a single point data acquisition in readable text format	
*list	Prints the list of MKIs supported	e.g.: MKI001V1→LPS22HB MKI105V1→LIS3DH MKI110V1→ AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ ...
*echoon	Activates the write verbose mode	e.g.: GRAAhDDh
*echooff	Deactivates the write verbose mode	
*gffiforst ⁽²⁾	Gyroscope “Reset mode” enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*gffifomde ⁽²⁾	Gyroscope “FIFO mode” enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*gffifostr ⁽²⁾	Gyroscope “FIFO stream” enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*gffifostf ⁽²⁾	Gyroscope “Stream to FIFO” enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*gffifobtf ⁽²⁾	Gyroscope “Bypass to FIFO” enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*gffifobts ⁽²⁾	Gyroscope “Bypass to stream” enable	st XH XL YH YL ZH ZL IR FC FS \r \n
*gffifodstr ⁽²⁾	Gyroscope “Dynamic stream” enable	st XH XL YH YL ZH ZL IR FC FS \r \n

1. IR: interrupt bytes; FC: FIFO control register; FS: FIFO source register

2. Available only for devices with embedded FIFO

4.5 Digital output magnetometers: supported commands

Table 7. Digital output magnetometer: supported commands list

Command	Description	Returned value ⁽¹⁾
*setdbXXXVY	Selects firmware according to the adapter connected	
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable text format	
*stop	Stops data acquisition	
*zon	Forces High impedance state. Turns off VDD and VDDIO power supply.	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply.	
*dev	Device name	e.g.: LIS2MDL
*ver	Firmware version	e.g.: V1.0.2
*mrAA	Magnetometer register read	e.g.: MRAAhDDh
*mrmAA ₁ NN	Multiple magnetometer registers read	e.g.: MRMAA ₁ hNNhDD ₁ h...DD _{NN} h
*multi-mrAA ₁ ...AA _N	Multiple magnetometer registers read	e.g.: MULTI-MRAA ₁ hDD ₁ h...AA _N hDD _N h
*mwAADD	Magnetometer register write	
*single	It gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	It gets a single point data acquisition in readable text format	
*list	Prints the list of MKIs supported	e.g.: MKI001V1→LPS22HB MKI105V1→LIS3DH MKI110V1→AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ...
*echoon	Activates the write verbose mode	e.g.: MRAAhDDh
*echooff	Deactivates the write verbose mode	

1. IR: interrupt bytes; FC: FIFO control register; FS: FIFO source register

4.6 Digital output pressure sensor: supported commands

Table 8. Digital output pressure sensor: supported commands list

Command	Description	Returned value ⁽¹⁾
*setdbXXXVY	Selects firmware according to the adapter connected	
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable text format	
*stop	Stops data acquisition	
*zon	Forces High impedance state. Turns off VDD and VDDIO power supply.	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply	
*dev	Device name	e.g.: LPS33K
*ver	Firmware version	e.g.: V1.0.2
*prAA	Pressure sensor register read	e.g.: PRAAhDDh
*prmAA ₁ NN	Multiple pressure sensor registers read	e.g.: PRMAA ₁ hNNhDD ₁ h...DD _{NN} h
*multi-prAA ₁ .. AA _N	Multiple pressure sensor registers read	e.g.: MULTI-PRAA ₁ hDD ₁ h...AA _N hDD _N h
*pwAADD	Pressure sensor register write	
*single	It gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	It gets a single point data acquisition in readable text format	
*list	Prints the list of MKIs supported	e.g.: MKI001V1→LPS22HB MKI105V1→LIS3DH MKI110V1→ AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ ...
*echoon	Activates the write verbose mode	e.g.: PRAAhDDh
*echooff	Deactivates the write verbose mode	
*pfiforst ⁽²⁾	Pressure sensor "Reset mode" enable	st PXL PL PH TL TH IR FC FS \r \n
*pfifomde ⁽²⁾	Pressure sensor "FIFO mode" enable	st PXL PL PH TL TH IR FC FS \r \n
*pfifostr ⁽²⁾	Pressure sensor "FIFO stream" enable	st PXL PL PH TL TH IR FC FS \r \n
*pfifostf ⁽²⁾	Pressure sensor "Stream to FIFO" enable	st PXL PL PH TL TH IR FC FS \r \n
*pfifobtf ⁽²⁾	Pressure sensor "Bypass to FIFO" enable	st PXL PL PH TL TH IR FC FS \r \n
*pfifobts ⁽²⁾	Pressure sensor "Bypass to stream" enable	st PXL PL PH TL TH IR FC FS \r \n
*pfifodstr ⁽²⁾	Pressure sensor "Dynamic stream" enable	st PXL PL PH TL TH IR FC FS \r \n

1. IR: interrupt bytes; FC: FIFO control register; FS: FIFO source register

2. Available only for devices with embedded FIFO

4.7 Digital output temperature sensor: supported commands

The table below lists the commands supported by the devices/evaluation boards including a digital output temperature sensor.

Table 9. Digital output temperature sensor: supported commands list

Command	Description	Returned value
*setdbXXXVY	Selects firmware according to the adapter connected	
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable textformat	
*stop	Stops data acquisition	
*zon	Forces High impedance state. Turns off VDD and VDDIO power supply.	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply.	
*dev	Device name	e.g.:STTS22H
*ver	Firmware version	e.g.: V1.0.2
*trAA	Temperature sensor register read	e.g.:TRAAhDDh
*trmAA ₁ NN	Multiple temperature sensor registers read	e.g.: TRMAA ₁ hNNhDD ₁ h...DD _{NN} h
*multi-trAA ₁ .. AAN	Multiple temperature sensor registers read	e.g.:MULTI-TRAA ₁ hDD ₁ h...AA _N hDD _N h
*twAADD	Temperature sensor register write	
*single	It gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	It gets a single point data acquisition in readable text format	
*list	Prints the list of MKIs supported	e.g.: MKI001V1→LPS22HB MKI105V1→LIS3DH MKI110V1→ AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ ...
*echoon	Activates the write verbose mode	e.g.: TRAAhDDh
*echooff	Deactivates the write verbose mode	

4.8 Analog output temperature sensor: supported commands

The table below lists the commands supported by the devices/evaluation boards that include an analog output temperature sensor.

Table 10. Analog output temperature sensor: supported commands list

Command	Description	Returned value
*setdbXXXVY	Selects firmware according to the adapter connected	
*odr [param]	Sets ODR speed (parameter: ODR in mHz)	
*start	Starts continuous data acquisition	(see Table 4. Returned values for *start command)
*debug	Returns the output data in readable text format	e.g.: Temp. = 22.93 [°C]
*stop	Stops data acquisition	
*zon	Forces High impedance state. Turns off VDD and VDDIO power supply.	
*zoff	Exits from High impedance state. Turns on VDD and VDDIO power supply.	
*dev	Device name	e.g.: STLM20
*ver	Firmware version	e.g.: V3.7.30
*single	Gets a single point data acquisition	(see Table 4. Returned values for *start command)
*sindebug	Gets a single point data acquisition in readable text format	e.g.: Temp. = 22.93 [°C]
*list	Prints the list of STEVAL-MKIxxxxx evaluation boards supported	e.g.: MKI001V1→LPS22HB MKI105V1→LIS3DH MKI110V1→ AIS328DQ
*listdev	Prints the list of devices supported	e.g.: LPS22HB LIS3DH AIS328DQ ...

4.9 Configuration of FIFO mode on combined devices

For the configuration of combined sensors as iNEMO class LSM6xxx and similar there are no dedicated commands for the Professional MEMS tool intended for device simple setup.

As devices are becoming more complex in each generation and with dedicated functionalities it is not possible to implement general dedicated commands. Refer to the respective device data sheet for its exact configuration and, particularly, for FIFO configuration. The meaning of data in FIFO registers then depends on complex FIFO setup defined by the end user.

5 Schematic diagrams

Figure 11. STEVAL-MKI109D circuit schematic (1 of 11)

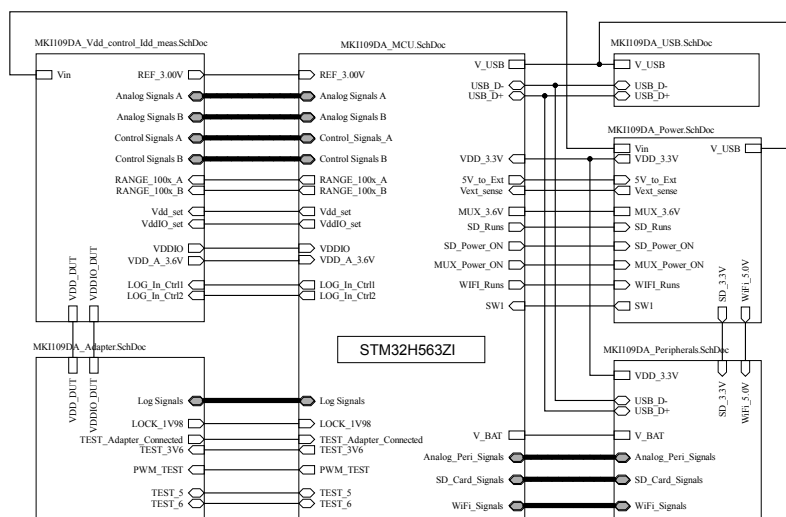


Figure 12. STEVAL-MKI109D circuit schematic (2 of 11)

Power & Reference for Idd measurement

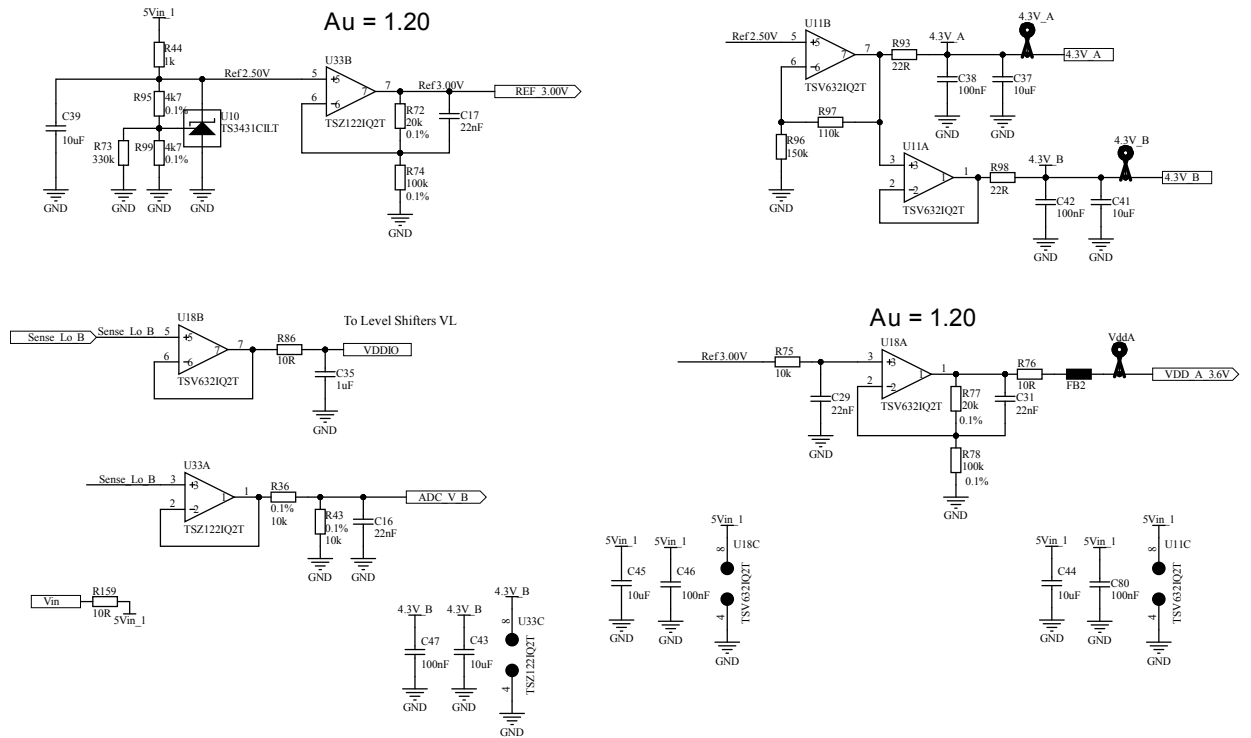


Figure 13. STEVAL-MKI109D circuit schematic (3 of 11)

V1 - Dual Channel I_{dd} measurement - lin

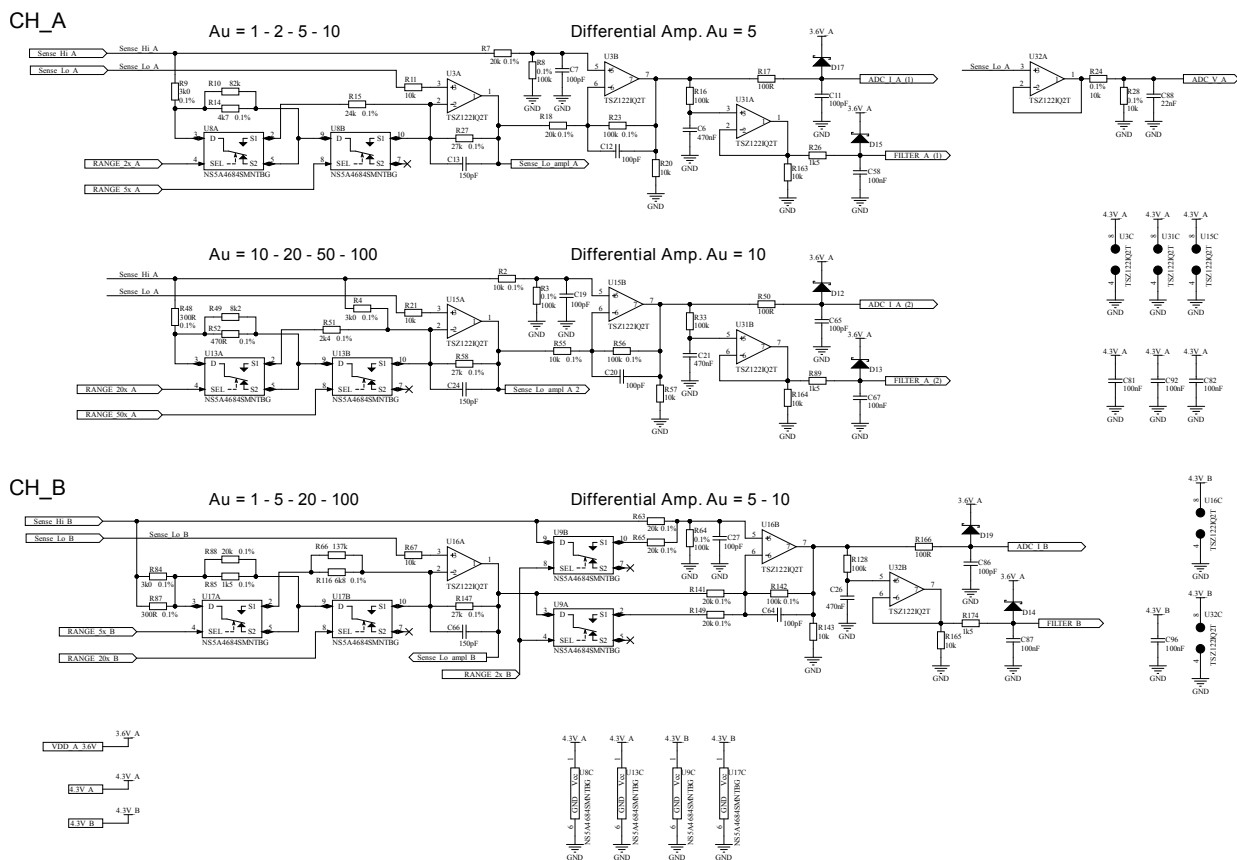
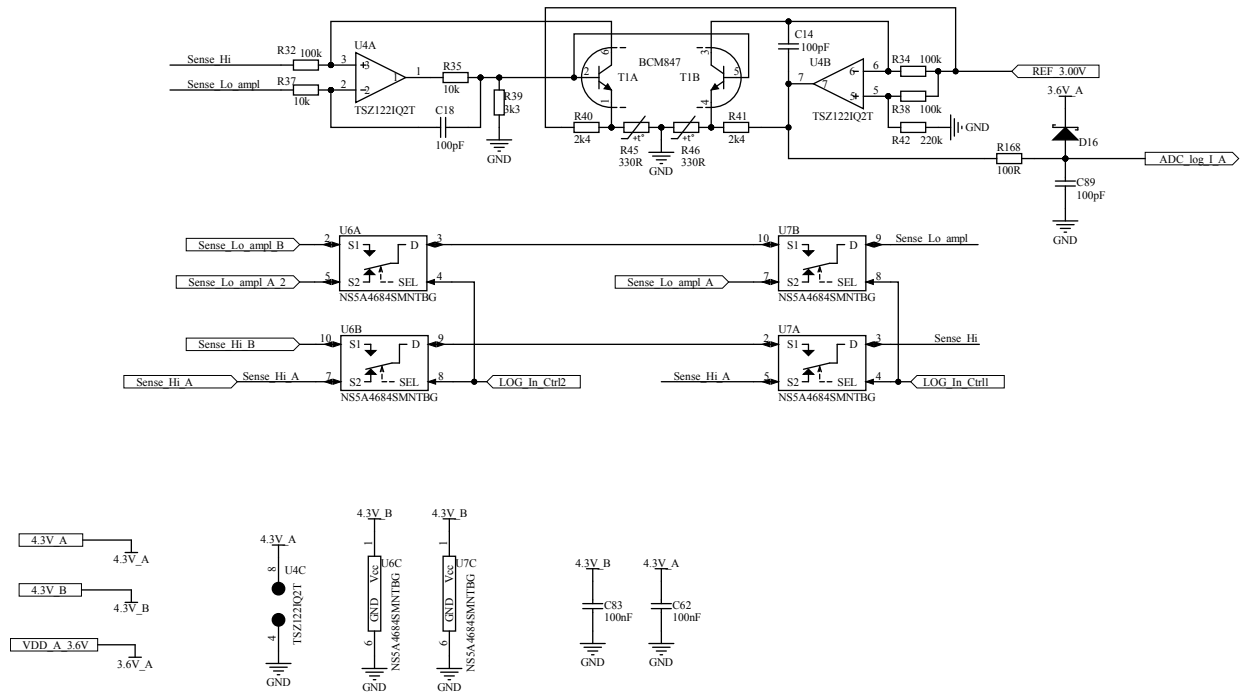


Figure 14. STEVAL-MKI109D circuit schematic (4 of 11)

I_{dd} measurement - log



DIL24 Device Adapter:

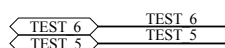


Figure 16. STEVAL-MKI109D circuit schematic (6 of 11)

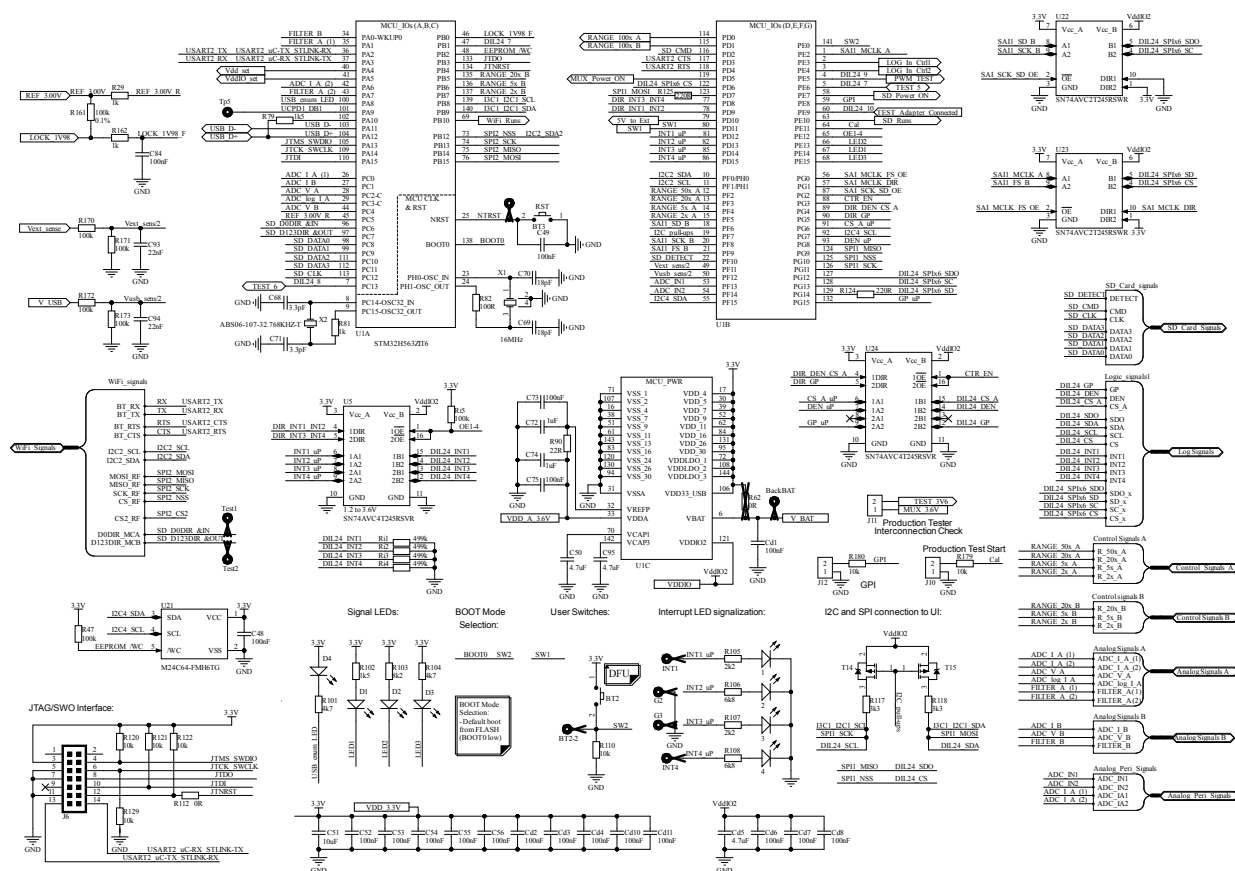


Figure 17. STEVAL-MKI109D circuit schematic (7 of 11)

USB Connection:

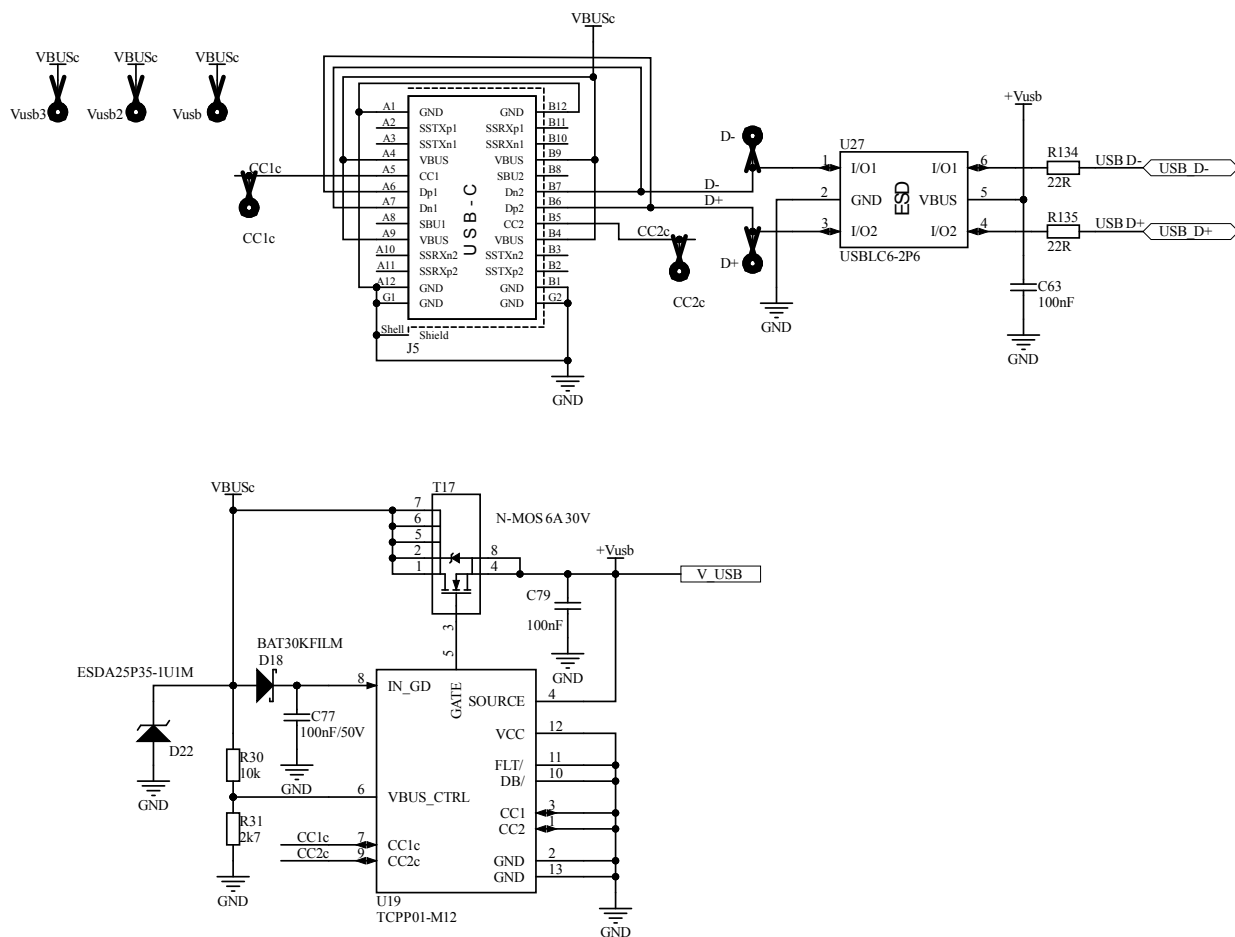


Figure 18. STEVAL-MKI109D circuit schematic (8 of 11)

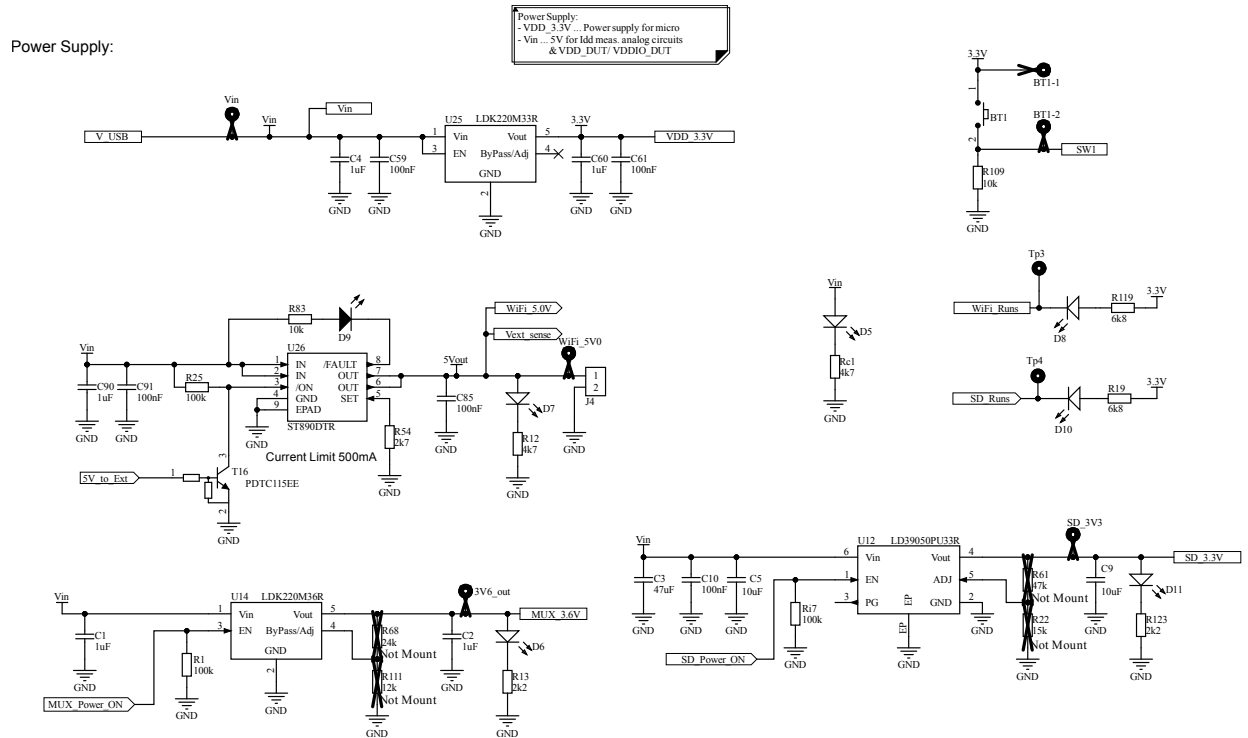


Figure 19. STEVAL-MKI109D circuit schematic (9 of 11)

Dual Channel Controlled Power Supply

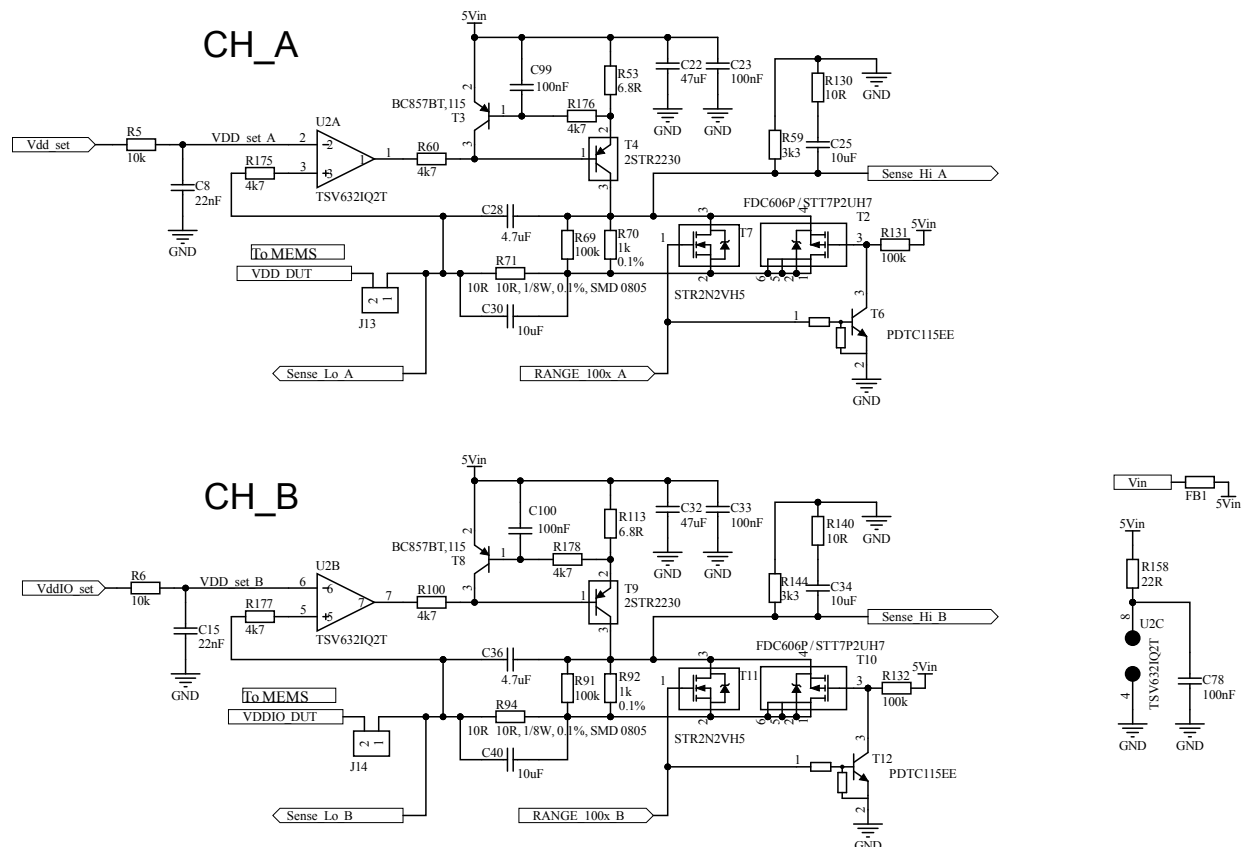


Figure 20. STEVAL-MKI109D circuit schematic (10 of 11)

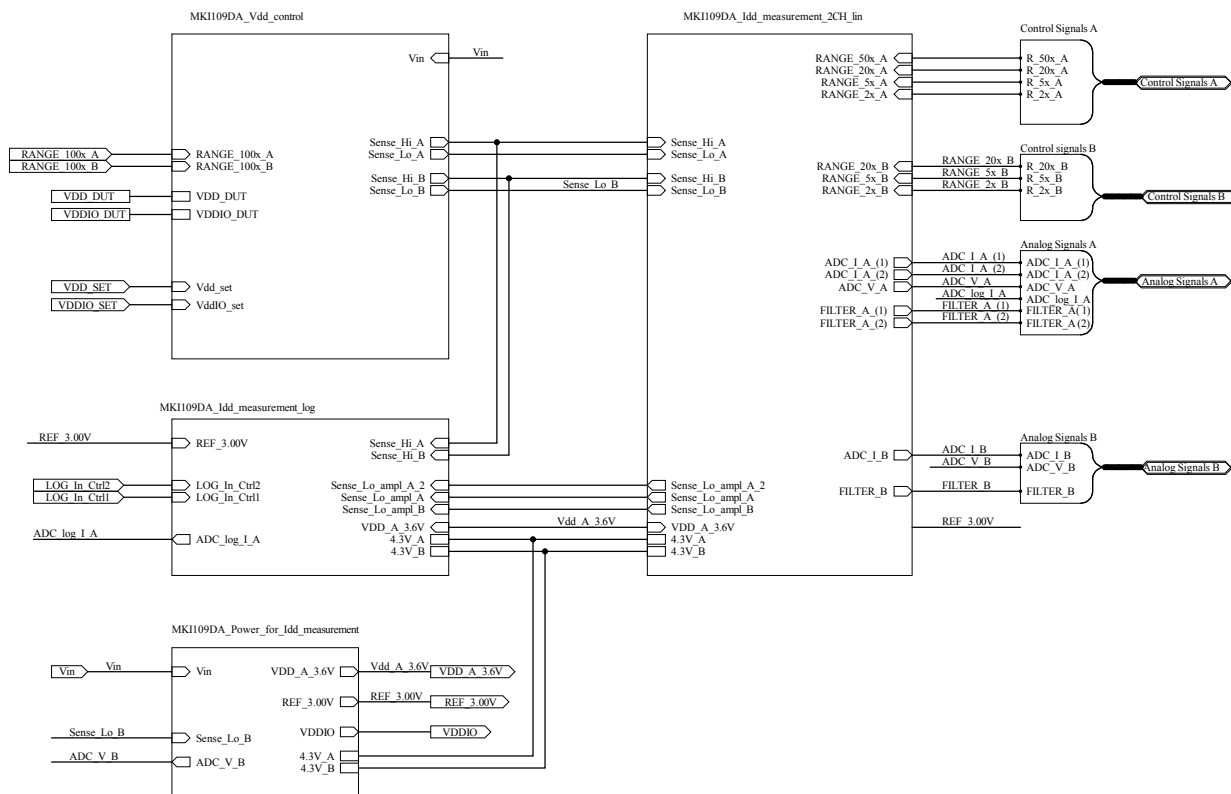
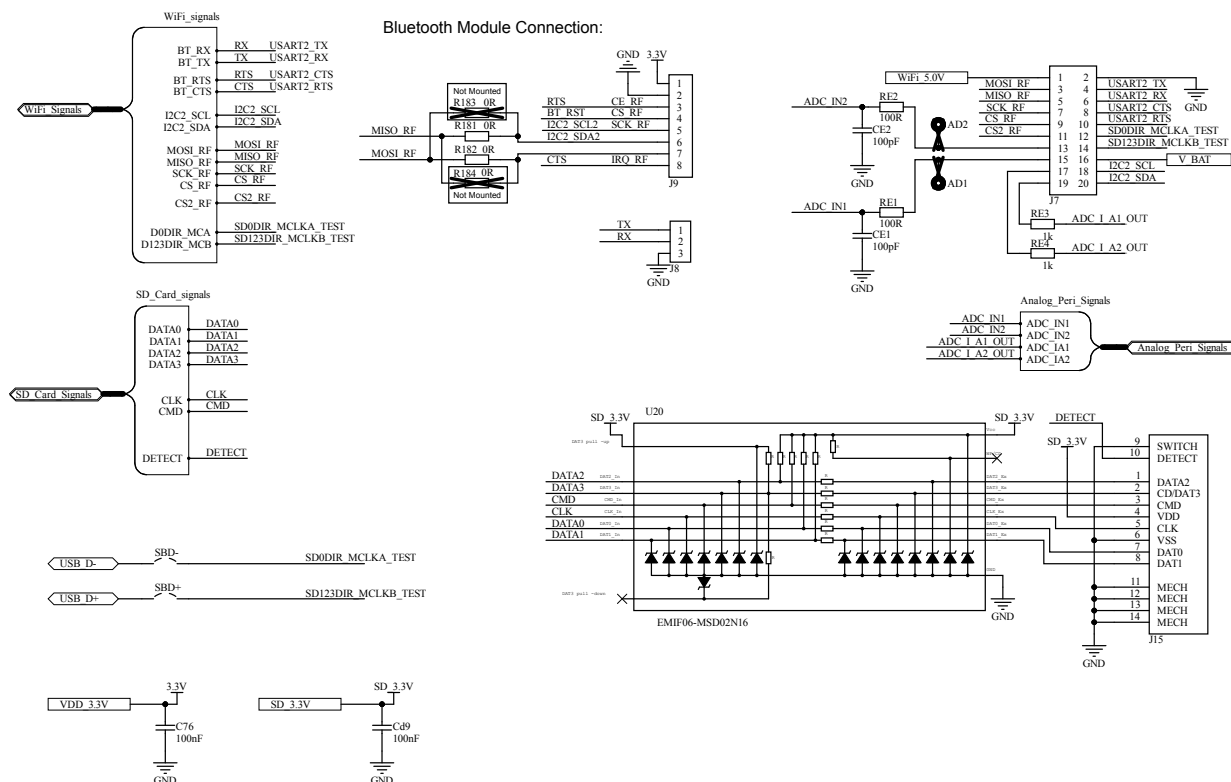


Figure 21. STEVAL-MKI109D circuit schematic (11 of 11)



6 Bill of materials

Table 11. STEVAL-MKI109D bill of materials

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
1	7	1, 3, D1, D5, D6, D7, D11	50mcd, SMD_LED_G_0603_REFLOW_S	LED GREEN 0603 50mcd 120°	KINGBRIGHT	KPT-1608CGCK
					WURTH ELEKTRONIK (1)	150060VS75000
2	5	2, 4, D3, D8, D10	150mcd, SMD_LED_Y_0603_REFLOW_S	LED YELLOW 0603 150mcd 120°	KINGBRIGHT	KPT-1608SYCK
					WURTH ELEKTRONIK (1)	150060YS75000
3	3	BT1, BT2, BT3	160 gf, SMD_KMR211G_PTS810_S	Button PTS810_and SKRPA	C&K COMPONENT S	PTS810SJM250SMTRLFS
					WURTH ELEKTRONIK (1)	434133025816
4	13	C5, C9, C25, C30, C34, C37, C39, C40, C41, C43, C44, C45, C51	10uF, SMD_C_0603_REFLOW_S	CAP CER 0603 10uF 6.3V X5R +/-20%	MULTICOMP	MC0603X106M6R3CT
					WURTH ELEKTRONIK (1)	885012106006
5	48	C10, C23, C33, C38, C42, C46, C47, C48, C49, C52, C53, C54, C55, C56, C58, C59, C61, C62, C63, C67, C73, C75, C76, C77, C78, C79, C80, C81, C82, C83, C84, C85, C87, C91, C92, C96, C99, C100, Cd1, Cd2, Cd3, Cd4, Cd6, Cd7, Cd8, Cd9, Cd10, Cd11	100nF, SMD_C_0603_REFLOW_S	CAP CER 0603 100nF 50V X7R +/-10%	MULTICOMP	MC0603B104K500CT
					WURTH ELEKTRONIK (1)	885012206095
6	8	8C1, C2, C4, C35, C60, C72, C74, C90	1uF, SMD_C_0603_REFLOW_S	CAP CER 0603 1uF 25V X5R +/-10%	MULTICOMP	MC0603X105K250CT
					WURTH ELEKTRONIK (1)	885012106030
7	3	C6, C21, C26	470nF, SMD_C_0603_REFLOW_S	CAP CER 0603 470nF 16V X5R +/-10%	MULTICOMP	MC0603X474K160CT

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
8	14	C7, C11, C12, C14, C18, C19, C20, C27, C64, C65, C86, C89, CE1, CE2	100pF, SMD_C_0603_RE FLOW_S	CAP CER 0603 100pF 50V C0G/NP0 +/-5%	MULTICOMP	MC0603N101J500CT
					WURTH ELEKTRONIK (1)	885012006057
9	11	C8, C15, C16, C17, C29, C31, C88, C93, C94, C97, C98	22nF, SMD_C_0603_RE FLOW_S	CAP CER 0603 22nF 50V X7R +/-10%	MULTICOMP	MC0603B223K500CT
					WURTH ELEKTRONIK (1)	885382206003
10	3	C13, C24, C66	150pF, SMD_C_0603_RE FLOW_S	CAP CER 0603 150pF 50V C0G/NP0 +/-5%	MULTICOMP	MC0603N151J500CT
					WURTH ELEKTRONIK (1)	885012006058
11	3	C3, C22, C32	47uF, SMD_C_0805_RE FLOW_S	CAP CER 0805 47uF 6.3V X5R +/-20%	MURATA	GRM21BR60J476ME1 5L
					WURTH ELEKTRONIK (1)	885012107006
12	5	C28, C36, C50, C95, Cd5	4.7uF, SMD_C_0603_RE FLOW_S	CAP CER 0603 4.7uF 10V X5R +/-10%	KEMET	C0603C475K8PAC78 67
13	2	C68, C71	3.3pF, SMD_C_0603_RE FLOW_S	CAP CER 0603 3.3pF 100V C0G/NP0 +/-5%	KYOCERA AVX	06031A3R3CAT2A
					WURTH ELEKTRONIK (1)	885012006090
14	2	C69, C70	18pF, SMD_C_0603_RE FLOW_S	CAP CER 0603 18pF 50V C0G/NP0 +/-5%	MULTICOMP	MC0603N180J500CT
15	2	D2, D9	230mcd, SMD_LED_R_060 3_REFLOW_S	LED RED 0603 230mcd 120°	KINGBRIGHT	KPT-1608SURCK
					WURTH ELEKTRONIK (1)	150060RS75000
16	1	D4	100mcd, SMD_LED_B_060 3_REFLOW_S	LED BLUE 0603 100mcd 120°	KINGBRIGHT	KPT-1608QBC-D
					WURTH ELEKTRONIK (1)	150060BS75000
17	8	D12, D13, D14, D15, D16, D17, D18, D19	0.3A, SMD_SOD523_R EFLOW	BAT30KFILM	ST	BAT30KFILM
18	1	D22	22V, SMD_DFN2_1.6x 1.0_REFLOW	ESDA25P35-1U1 M	ST	ESDA25P35-1U1M
19	1	FB1	0.35ΩDC, 600Ω/ 100MHz, 400mA, SMD_R_0603_RE FLOW_S	2506036017Y0	FAIR-RITE	2506036017Y0
					WURTH ELEKTRONIK (1)	74279265

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
20	1	FB2	1.5ΩDC, 2.2kΩ/ 100MHz, 100mA, SMD_R_0603_RE FLOW_S	HZ0603A222R-10	LAIRD TECHNOLOG IES	HZ0603A222R-10
					WURTH ELEKTRONIK (1)	742792695
21	1	J1	TRH_HDR1X12_0 1_TO_12	Header 1x12 pins, 2.54mm, 1-12	WURTH ELEKTRONIK	61301211121
22	2	J2a, J2b	TRH_HDR_1X12 _FEMALE	Receptacle 1x12 pins, 2.54mm	E-TEC	BL1-036-G-700-01
23	1	J3	TRH_HDR1X12_1 3_TO_24	Header 1x12 pins, 2.54mm, 13-24	WURTH ELEKTRONIK	61301211121
24	1	J4	TRH_HDR1X02	Header 1x2 pins, 2.54mm	HARWIN	M20-9990246
					WURTH ELEKTRONIK (1)	61300211121
25	1	J5	24 pins, SMD_TRH_USB- C	USB 3.1 TYPE C, RECEPTACLE, SMD_TRH	WURTH ELEKTRONIK	632723100011
26	1	J6	TRH_HDR2X07_1 .27MM_BOX	Header 2x7 pins, 1.27mm	SAMTEC	FTSH-107-01-L-D-K
27	1	J7	SMD_RECEP_2x 10_50mil	Receptacle 2x10 pins, 1.27mm, straight, SMD	Samtec	SFMC-110-02-L-D
28	1	J8	TRH_HDR1X03_1 .27MM	Header 1x3 pins, 1.27mm, straight	SAMTEC	TMS-103-01-G-S
29	1	J9	TRH_HDR1X08_1 .27MM	Header 1x8 pins, 1.27mm, straight	SAMTEC	TMS-108-01-G-S
30	5	J10, J11, J12, J13, J14	TRH_HDR1X02_2 MM	Header 1x2 pins, 2mm, straight	HARWIN	M22-2510205
					WURTH ELEKTRONIK (1)	62000211121
31	1	J15	SMD_MICRO_SD _DM3D-SF	Micro SD Card Slot HRS - DM3D- SF	Hirose	DM3D-SF
32	6	R2, R24, R28, R36, R43, R55	10k, SMD_R_0603_RE FLOW_S	RES 0603 10k 0.1% 1/16W	PANASONIC	ERA3AEB103V
33	11	R3, R8, R23, R56, R64, R69, R74, R78, R91, R142, R161	100k, SMD_R_0603_RE FLOW_S	RES 0603 100k 0.1% 1/16W	PANASONIC	ERA3AEB104V
34	3	R4, R9, R84	3k0, SMD_R_0603_RE FLOW_S	RES 0603 3k0 0.1% 1/16W	PANASONIC	ERA3AEB302V

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
35	24	R5, R6, R11, R20, R21, R30, R35, R37, R57, R67, R75, R83, R109, R110, R120, R121, R122, R129, R143, R163, R164, R165, R179, R180	10k, SMD_R_0603_RE FLOW_S	RES 0603 10k 1% 1/16W	MULTICOMP	MC0063W0603110K
36	9	R7, R18, R63, R65, R72, R77, R88, R141, R149	20k, SMD_R_0603_RE FLOW_S	RES 0603 20k 0.1% 1/16W	PANASONIC	ERA3AEB203V
37	1	R10	82k, SMD_R_0603_RE FLOW_S	RES 0603 82k 1% 1/16W	MULTICOMP	MC0063W0603182K
38	3	R14, R95, R99	4k7, SMD_R_0603_RE FLOW_S	RES 0603 4k7 0.1% 1/16W	PANASONIC	ERA3AEB472V
39	1	R15	24k, SMD_R_0603_RE FLOW_S	RES 0603 24k 0.1% 1/10W	PANASONIC	ERA3AEB243V
40	17	R1, R16, R25, R32, R33, R34, R38, R47, R128, R131, R132, R170, R171, R172, R173, Ri5, Ri7	100k, SMD_R_0603_RE FLOW_S	RES 0603 100k 1% 1/16W	MULTICOMP	MC0063W06031100K
41	7	R17, R50, R82, R166, R168, RE1, RE2	100R, SMD_R_0603_RE FLOW_S	RES 0603 100R 1% 1/16W	MULTICOMP	MC0063W06031100R
42	4	R19, R106, R108, R119	6k8, SMD_R_0603_RE FLOW_S	RES 0603 6k8 1% 1/16W	MULTICOMP	MC0063W060316K8
43	5	R26, R79, R89, R102, R174	1k5, SMD_R_0603_RE FLOW_S	RES 0603 1k5 1% 1/16W	MULTICOMP	MC0063W060311K5
44	3	R27, R58, R147	27k, SMD_R_0603_RE FLOW_S	RES 0603 27k 0.1% 1/10W	PANASONIC	ERA3AEB273V
45	6	R29, R44, R81, R162, RE3, RE4	1k, SMD_R_0603_RE FLOW_S	RES 0603 1k 1% 1/16W	MULTICOMP	MC0063W060311K
46	2	R31, R54	2k7, SMD_R_0603_RE FLOW_S	RES 0603 2k7 1% 1/16W	MULTICOMP	MCWR06X2701FTL
47	5	R39, R59, R117, R118, R144	3k3, SMD_R_0603_RE FLOW_S	RES 0603 3k3 1% 1/16W	MULTICOMP	MC0063W060313K3
48	2	R40, R41	2k4, SMD_R_0603_RE FLOW_S	RES 0603 2k4 1% 1/16W	MULTICOMP	MC0063W060312K4

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
49	1	R42	220k, SMD_R_0603_RE FLOW_S	RES 0603 220k 1% 1/16W	MULTICOMP	MC0063W06031220K
50	2	R45, R46	330R, SMD_R_0603_RE FLOW_S	Thermistor PTC 0603 330R 1% 75mW	VISHAY	TFPT0603L3300FV
51	2	R48, R87	300R, SMD_R_0603_RE FLOW_S	RES 0603 300R 0.1% 1/16W	PANASONIC	ERA3AEB301V
52	2	R49, R103	8k2, SMD_R_0603_RE FLOW_S	RES 0603 8k2 1% 1/16W	MULTICOMP	MC0063W060318K2
53	1	R51	2k4, SMD_R_0603_RE FLOW_S	RES 0603 2k4 0.1% 1/10W	PANASONIC	ERA3AEB242V
54	1	R52	470R, SMD_R_0603_RE FLOW_S	RES 0603 470R 0.1% 1/16W	PANASONIC	ERA3AEB471V
55	2	R53, R113	6.8R, SMD_R_0805_RE FLOW_S	RES 0805 6R8 1% 1/8W	WELWYN	ASC0805-6R8FT5
56	10	R12, R60, R100, R101, R104, R175, R176, R177, R178, Rc1	4k7, SMD_R_0603_RE FLOW_S	RES 0603 4k7 1% 1/16W	MULTICOMP	MC0063W060314K7
57	1	R66	137k, SMD_R_0603_RE FLOW_S	RES 0603 137k 1% 1/10W	VISHAY DRALORIC	CRCW0603137KFKE A
58	2	R70, R92	1k, SMD_R_0603_RE FLOW_S	RES 0603 1k 0.1% 1/16W	SUSUMU	RR0816P-102-B-T5
59	2	R71, R94	10R, SMD_R_0805_RE FLOW_S	RES 0805 10R 0.1% 1/8W	YAGEO	RT0805BRD0710RL
60	1	R73	330k, SMD_R_0603_RE FLOW_S	RES 0603 330k 1% 1/16W	MULTICOMP	MC0063W06031330K
61	5	R76, R86, R130, R140, R159	10R, SMD_R_0603_RE FLOW_S	RES 0603 10R 1% 1/16W	MULTICOMP	MC0063W0603110R
62	1	R85	1k5, SMD_R_0603_RE FLOW_S	RES 0603 1k5 0.1% 1/16W	SUSUMU	ERA3AEB152V
63	6	R90, R93, R98, R134, R135, R158	22R, SMD_R_0603_RE FLOW_S	RES 0603 22R 1% 1/16W	MULTICOMP	MC0063W0603122R
64	1	R96	150k, SMD_R_0603_RE FLOW_S	RES 0603 150k 1% 1/16W	MULTICOMP	MC0063W06031150K
65	1	R97	110k, SMD_R_0603_RE FLOW_S	RES 0603 110k 1% 1/16W	MULTICOMP	MCWR06X1103FTL
66	4	R13, R105, R107, R123	2k2, SMD_R_0603_RE FLOW_S	RES 0603 2k2 1% 1/16W	MULTICOMP	MC0063W060312K2

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
67	3	R112, R181, R182	0R, SMD_R_0603_RE FLOW_S	RES 0603 0R0 1% 1/16W	MULTICOMP	MC0063W06030R
68	1	R116	6k8, SMD_R_0603_RE FLOW_S	RES 0603 6k8 0.1% 1/10W	PANASONIC	ERA3AEB682V
69	2	R124, R125	220R, SMD_R_0603_RE FLOW_S	RES 0603 220R 1% 1/16W	MULTICOMP	MC0063W06031220R
70	4	Ri1, Ri2, Ri3, Ri4	499k, SMD_R_0603_RE FLOW_S	RES 0603 499k 1% 1/16W	VISHAY DRALORIC	CRCW0603499KFKE A
71	1	T1	0.1A, 45V, SMD_SOT363_R EFLOW_S	BCM847BS	NXP	BCM847BS,115
72	2	T2, T10	P-MOS 6A -12V, SMD_SOT23-6_R EFLOW	FDC606P_b	ON Semi / ST	FDC606P / STT7P2UH7
73	2	T3, T8	PNP, 0.1A, -65V, SMD_SOT416-3_ REFLOW	BC857BT PNP 65V 0.1A	Onsemi	BC857BTT1G
74	2	T4, T9	PNP, 1.5A, -30V, SMD_SOT23-3	2STR2230_b	ST	2STR2230
75	3	T6, T12, T16	0.1A, -50V, SMD_SOT416-3_ REFLOW	PDTC115EE NPN+R 50V 0.1A	NXP	PDTC115EE
76	2	T7, T11	N-MOS, 2.3A, -20V, SMD_SOT23-3	STR2N2VH5_b	ST	STR2N2VH5
77	2	T14, T15	P-MOS, 0.9A, 20V, SMD_SOT416-3_ REFLOW	NTA4151PT1G	ON Semi	NTA4151PT1G
78	1	T17	N-MOS, 6A, 30V, SMD_PowerFLAT _2x2	STL6N3LLH6	ST	STL6N3LLH6
79	1	U1	STM32H563ZIT6, SMD_LQFP144_R EFLOW	STM32H563ZIT6	ST	STM32H563ZIT6
80	3	U2, U11, U18	TSV632IQ2T, SMD_DFN8_2x2_ REFLOW	TSV632IQ2T_ver1	ST	TSV632IQ2T
81	7	U3, U4, U15, U16, U31, U32, U33	TSZ122IQ2T, SMD_DFN8_2x2_ REFLOW	TSZ122IQ2T_ver1	ST	TSZ122IQ2T
82	2	U5, U24	SN74AVC4T245R SVR, SMD_UQFN16_2. 6x1.8_TI_REFLO W	SN74AVC4T245R SVR	TI	SN74AVC4T245RSVR
83	6	U6, U7, U8, U9, U13, U17	NS5A4684SMNTB G, SMD_QFN10L_O NSEMI_1.8x1.4_R EFLOW	NS5A4684SMNTB G	ON SEMICONDU CTOR	NS5A4684SMNTBG
84	1	U10	TS3431CILT, SMD_SOT23-3	TS3431CILT	ST	TS3431CILT

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
85	1	U12	LD39050PU33R, SMD_DFN6_3x3_ EP_STBC08_REF LOW	LD39050PU33R	ST	LD39050PU33R
86	1	U19	TCPP01-M12, QFN12_TCPP01- M12_STM	TCPP01-M12	ST	TCPP01-M12
87	1	U20	EMIF06- MSD02N16, SMD_QFN40P350 X120X50-17L	EMIF06- MSD02N16	ST	EMIF06-MSD02N16
88	1	U21	M24C64- FMH6TG, SMD_DFN5_1.7x 1.4_0.4_REFLOW	M24C64-FMH6TG	ST	M24C64-FMH6TG
89	2	U22, U23	SN74AVC2T245R SWR, SMD_UQFN10_1. 8x1.4_TI_REFLO W	SN74AVC2T245R SWR	TI	SN74AVC2T245RSW R
90	1	U25	LDK220M33R, SMD_SOT23-5_H AND	LDK220M33R	ST	LDK220M33R
91	1	U26	ST890DTR, DFN8L-3x3_STM	ST890DTR	ST	ST890DTR
92	1	U27	USBLC6-2P6, SMD_SOT666_R EFLOW	USBLC6-2P6	ST	USBLC6-2P6
93	1	Um1	LDK220M36R, SMD_SOT23-5_R EFLOW	LDK220M36R	ST	LDK220M36R
94	1	X1	16MHz, SMD_X_ABM8	Q ABM8-16MHz SMD	ABRACOM	ABM8-16.000MHZ-B2- T
					WURTH ELEKTRONIK (1)	830055951
95	1	X2	ABS-06-10, 32.768KHz, SMD_X_2.0x1.2_ S	ABS06-107-32.76 8kHz-T 4pF	ABRACON	ABS06-107-32.768KH Z-T
					WURTH ELEKTRONIK (1)	830107326401
96	2	SJ13, SJ14	----	Jumper Shunt 1x2 pins, 2mm	MULTICOMP	SPC20180

1. Alternative manufacturer.

7 Certified MEMS evaluation boards/kits

The following MEMS evaluation boards are certified as per following standards:

- FCC CFR 47 Part 15 Subpart B
- IEC 61000-6-1:2016 – IEC 61000-6-3:2020
- EN IEC 61000-6-1:2019 – EN IEC 61000-6-3:2021
- CISPR 32:2015 + A1:2019 / EN 55032:2015 + A1:2020 + A11:2020
- CISPR 35:2016 / EN 55035:2017 + A11:2020
- IEC 61000-3-2:2018 + A1:2020
- EN IEC 61000-3-2:2019 + A1:2021
- IEC 61000-3-3:2013 + A1:2017 + A2: 2021
- EN 61000-3-3:2013 + A1:2019 + A2: 2021
- ICES-003 Issue 7 (2020)

Table 12. List of certified MEMS evaluation boards/kits

Board name (link)	Device
STEVAL-MET001V1	LPS22HB
STEVAL-MKI089V1	LIS331DLH
STEVAL-MKI105V1	LIS3DH
STEVAL-MKI106V1	LSM303DLHC
STEVAL-MKI107V2	L3GD20
STEVAL-MKI110V1	AIS328DQ
STEVAL-MKI122V1	LSM330DLC
STEVAL-MKI123V1	LSM330D
STEVAL-MKI125V1	A3G4250D
STEVAL-MKI127V1	L3GD20
STEVAL-MKI133V1	LSM303D
STEVAL-MKI134V1	LIS3DSH
STEVAL-MKI135V1	LIS2DH
STEVAL-MKI136V1	L3GD20H
STEVAL-MKI137V1	LIS3MDL
STEVAL-MKI142V1	LPS25H
STEVAL-MKI151V1	LIS2DH12
STEVAL-MKI153V1	H3LIS331DL
STEVAL-MKI154V1	LSM9DS0
STEVAL-MKI158V1	AIS3624DQ
STEVAL-MKI159V1	LSM9DS1
STEVAL-MKI160V1	LSM6DS3
STEVAL-MKI164V1	LIS2HH12
STEVAL-MKI165V1	LPS25HB
STEVAL-MKI166V1	H3LIS100DL
STEVAL-MKI167V1	H3LIS200DL
STEVAL-MKI168V1	IIS2DH
STEVAL-MKI169V1	I3G4250D

Board name (link)	Device
STEVAL-MKI170V1	IIS328DQ
STEVAL-MKI172V1	LSM303AGR
STEVAL-MKI173V1	LSM303AH
STEVAL-MKI174V1	LIS2DS12
STEVAL-MKI175V1	LIS2DE12
STEVAL-MKI176V1	LSM6DS3H
STEVAL-MKI177V1	LPS35HW
STEVAL-MKI178V1	LSM6DSL
STEVAL-MKI178V2	LSM6DSL
STEVAL-MKI179V1	LIS2DW12
STEVAL-MKI180V1	LIS3DHH
STEVAL-MKI181V1	LIS2MDL
STEVAL-MKI182V1	ISM330DLC
STEVAL-MKI182V2	ISM330DLC
STEVAL-MKI183V1	LPS33HW
STEVAL-MKI184V1	ISM303DAC
STEVAL-MKI185V1	IIS2MDC
STEVAL-MKI186V1	IIS3DHHC
STEVAL-MKI188V1	L20G20IS
STEVAL-MKI189V1	LSM6DSM
STEVAL-MKI190V1	LIS2DTW12
STEVAL-MKI191V1	IIS2DLPC
STEVAL-MKI192V1	LPS22HH
STEVAL-MKI193V1	ASM330LHH
STEVAL-MKI194V1	LSM6DSR
STEVAL-MKI195V1	LSM6DSRX
STEVAL-MKI196V1	LSM6DSO
STEVAL-MKI197V1	LSM6DSOX
STEVAL-MKI198V1K	STTS751
STEVAL-MKI199V1K	STLM20
STEVAL-MKI200V1K	STTS22H
STEVAL-MKI201V1K	STTS75
STEVAL-MKI202V1K	STDS75
STEVAL-MKI203V1K	STCN75
STEVAL-MKI204V1K	STLM75
STEVAL-MKI205V1	LPS33W
STEVAL-MKI206V1	AIS2DW12
STEVAL-MKI207V1	ISM330DHCX
STEVAL-MKI208V1K	IIS3DWB
STEVAL-MKI209V1K	IIS2ICLX
STEVAL-MKI210V2K	ISM330DHCX

Board name (link)	Device
STEVAL-MKI210V2K	ISM330DHCX
STEVAL-MKI211V1K	LIS25BA
STEVAL-MKI212V1	ASM330LHHX
STEVAL-MKI213V1	LPS27HHW
STEVAL-MKI214V1	LPS33K
STEVAL-MKI215V1	LSM6DSO32
STEVAL-MKI216V1K	IIS3DHHC
STEVAL-MKI217V1	LSM6DSOX+LIS2MDL
STEVAL-MKI218V1	AIS2IH
STEVAL-MKI219V1	LPS22CH
STEVAL-MKI220V1	LPS27HHTW
STEVAL-MKI221V1	LSM6DSO32X
STEVAL-MKI224V1	LPS22DF
STEVAL-MKI225A	LPS28DFW
STEVAL-MKI227A	LSM6DSV16X
STEVAL-MKI228KA	ILPS22QS
STEVAL-MKI229A	LSM6DSO16IS
STEVAL-MKI230KA	ISM330IS
STEVAL-MKI231KA	STHS34PF80
STEVAL-MKI233KA	ISM330ISN
STEVAL-MKI234KA	LSM6DSV16BX
STEVAL-MKI235KA	LIS2DUXS12
STEVAL-MKI236A	2x ASM330LHB
STEVAL-MKI237KA	LSM6DSV16BX
STEVAL-MKI238A	LIS2DUX12
STEVAL-MKI239A	LSM6DSV
STEVAL-MKI241KA	LSM6DSV16B
STEVAL-MKI243A	ASM330LHHXG1
STEVAL-MKI244A	2xASM330LHBG1
STEVAL-MKI245KA	ISM330BX
STEVAL-MKI246KA	LIS2DUXS12
STEVAL-MKI247A	LSM6DSV80X
STEVAL-MKI249KA	ST1VAFE6AX
STEVAL-MKI250KA	ST1VAFE3BX

8 Regulatory compliance information

Note: *This regulatory compliance information applies to all the boards listed in [Section 7: Certified MEMS evaluation boards/kits](#).*

Note: *The evaluation kit with order code STEVAL-MKlxxxY contains the board whose finished good is STEVAL\$MKlxxxYA (x is a number and Y is a letter).*

Notice for US Federal Communication Commission (FCC)

FCC NOTICE - This kit is designed to allow:

- (1) Product developers to evaluate electronic components, circuitry, or software associated with the kit to determine whether to incorporate such items in a finished product and
- (2) Software developers to write software applications for use with the end product.

This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required FCC equipment authorizations are first obtained. Operation is subject to the condition that this product not cause harmful interference to licensed radio stations and that this product accept harmful interference. Unless the assembled kit is designed to operate under part 15, part 18 or part 95 of this chapter, the operator of the kit must operate under the authority of an FCC license holder or must secure an experimental authorization under part 5 of this chapter 3.1.2.

Notice for Innovation, Science and Economic Development Canada (ISED)

For evaluation purposes only. This kit generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to Industry Canada (IC) rules.

À des fins d'évaluation uniquement. Ce kit génère, utilise et peut émettre de l'énergie radiofréquence et n'a pas été testé pour sa conformité aux limites des appareils informatiques conformément aux règles d'Industrie Canada (IC).

Notice for European Union

This device is in conformity with the essential requirements of the Directive 2014/30/EU (EMC) and of the Directive 2015/863/EU (RoHS).

Notice for United Kingdom

This device is in compliance with the UK Electromagnetic Compatibility Regulations 2016 (UK S.I. 2016 No. 1091) and with the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment Regulations 2012 (UK S.I. 2012 No. 3032).

Revision history

Table 13. Document revision history

Date	Revision	Changes
02-Oct-2024	1	Initial release.
11-Dec-2024	2	Updated Section Introduction , Section 3: Supported MEMS adapter boards and Section 7: Certified MEMS evaluation boards/kits . Minor text changes.
07-Mar-2025	3	Updated Section 6: Bill of materials .
02-Apr-2025	4	Updated item number 20 of Section 6: Bill of materials .

Contents

1	Demonstration kit description.....	2
2	Professional MEMS tool board installation	7
2.1	Hardware installation	7
2.2	Firmware upgrade	8
2.2.1	Entering DFU mode	8
2.2.2	Firmware upgrade on Windows with STM32CubeProgrammer using *.bin source files. . . .	8
2.2.3	Firmware upgrade using command line on Windows/Linux/Mac OS® with STM32CubeProgrammer using *.bin source files	11
3	Supported MEMS adapter boards	12
4	Supported commands	15
4.1	Getting started	15
4.1.1	Quick start.	15
4.2	Supported commands.....	16
4.2.1	*setdbXXXVY	18
4.2.2	*start	18
4.2.3	*debug	21
4.2.4	*stop	22
4.2.5	*zon and *zoff	22
4.2.6	*dev	22
4.2.7	*ver	22
4.2.8	*rAA	22
4.2.9	*wAADD	23
4.2.10	*grAA	23
4.2.11	*gwAADD	23
4.2.12	*mrAA	23
4.2.13	*mwAADD	24
4.2.14	*prAA	24
4.2.15	*pwAADD	24
4.2.16	*hrAA	24
4.2.17	*hwAADD	25
4.2.18	*trAA	25
4.2.19	*twAADD	25
4.2.20	*single	25
4.2.21	*sindebug	25
4.2.22	*list	25
4.2.23	*listdev	25

4.2.24	*echoon	25
4.2.25	*echooff	25
4.2.26	*fiforst	26
4.2.27	*fifomde	26
4.2.28	*fifostr	26
4.2.29	*fifostf	26
4.2.30	*fifobtf	26
4.2.31	*fifobts	26
4.2.32	*fifodstr	26
4.2.33	*gfiforst	26
4.2.34	*gfifomde	26
4.2.35	*gfifostr	26
4.2.36	*gfifostf	26
4.2.37	*gfifobtf	26
4.2.38	*gfifobts	26
4.2.39	*gfifodstr	26
4.2.40	*pfiforst	26
4.2.41	*pfifomde	26
4.2.42	*pfifostr	27
4.2.43	*pfifostf	27
4.2.44	*pfifobtf	27
4.2.45	*pfifobts	27
4.2.46	*pfifodstr	27
4.2.47	*set_vddaV.VVV and *set_vddioV.VVV	27
4.2.48	*power_on and *power_off	27
4.2.49	*odr [p1]	27
4.2.50	*get_i [p1] [p2] [p3] [p4]	27
4.2.51	*get_v [p1]	28
4.2.52	*ranges_auto and *ranges_man	28
4.2.53	*range_ma [p1] and *range_mb [p1]	28
4.2.54	*adc_run_time [p1]	28
4.2.55	*adc_uni_stop	28
4.3	Digital output accelerometers: supported commands	29
4.4	Digital output gyroscopes: supported commands	30
4.5	Digital output magnetometers: supported commands	31
4.6	Digital output pressure sensor: supported commands	32
4.7	Digital output temperature sensor: supported commands	33
4.8	Analog output temperature sensor: supported commands	34

4.9	Configuration of FIFO mode on combined devices	34
5	Schematic diagrams	35
6	Bill of materials	44
7	Certified MEMS evaluation boards/kits	51
8	Regulatory compliance information	54
	Revision history	55
	List of tables	59
	List of figures	60

List of tables

Table 1.	List of supported MEMS adapter boards	12
Table 2.	List of supported commands	16
Table 3.	List of supported commands for power supply control and I/V measurement	18
Table 4.	Returned values for *start command.	19
Table 5.	Digital output accelerometers: supported commands list	29
Table 6.	Digital output gyroscopes: supported commands list.	30
Table 7.	Digital output magnetometer: supported commands list.	31
Table 8.	Digital output pressure sensor: supported commands list	32
Table 9.	Digital output temperature sensor: supported commands list	33
Table 10.	Analog output temperature sensor: supported commands list	34
Table 11.	STEVAL-MKI109D bill of materials	44
Table 12.	List of certified MEMS evaluation boards/kits.	51
Table 13.	Document revision history	55

List of figures

Figure 1.	Professional MEMS tool kit STEVAL-MKI109D	1
Figure 2.	Demonstration board block diagram	2
Figure 3.	Top silkscreen of the Professional MEMS Tool kit STEVAL-MKI109D	3
Figure 4.	Top view of Professional MEMS Tool kit STEVAL-MKI109D	4
Figure 5.	Bottom view of Professional MEMS Tool kit STEVAL-MKI109D	5
Figure 6.	How to plug the DIL24 adapter on STEVAL-MKI109D	6
Figure 7.	Virtual COM port assignment	7
Figure 8.	Opening the selected .bin file in STM32CubeProgrammer	9
Figure 9.	Programming process in STM32CubeProgrammer	9
Figure 10.	File download complete message after successful firmware update	10
Figure 11.	STEVAL-MKI109D circuit schematic (1 of 11)	35
Figure 12.	STEVAL-MKI109D circuit schematic (2 of 11)	36
Figure 13.	STEVAL-MKI109D circuit schematic (3 of 11)	37
Figure 14.	STEVAL-MKI109D circuit schematic (4 of 11)	38
Figure 15.	STEVAL-MKI109D circuit schematic (5 of 11)	39
Figure 16.	STEVAL-MKI109D circuit schematic (6 of 11)	40
Figure 17.	STEVAL-MKI109D circuit schematic (7 of 11)	41
Figure 18.	STEVAL-MKI109D circuit schematic (8 of 11)	42
Figure 19.	STEVAL-MKI109D circuit schematic (9 of 11)	42
Figure 20.	STEVAL-MKI109D circuit schematic (10 of 11)	43
Figure 21.	STEVAL-MKI109D circuit schematic (11 of 11)	43

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2025 STMicroelectronics – All rights reserved