User Manual

DA728x Motherboard 359-05-B

UM-HA-001

Abstract

This document describes the functionality of the DA728x haptic driver motherboard 359-05-B and its use along with the DA728x daughterboards to allow functional demonstration and evaluation to be carried out.

Contents

Figures

Tables

1 Terms and Definitions

NOTE
"x" can denote different version of boards such as A and B variants.

359-05-x	DA728x Motherboard, containing LRA as well as the digital and analogue accelerometer circuitry
359-06-x	DA7280 Daughterboard
359-07-x	DA7281 Daughterboard
359-08-x	DA7282 Daughterboard
359-09-x	DA7283 Daughterboard
DA7280	Renesas haptic driver integrated circuit
DA7281	DA7280 variant with I2C address select
DA7282	DA7280 variant with ultra-low power shutdown
DA7283	DA7280 variant with I2C control and with ultra-low power shutdown
DUT	Device under Test
ERM	Eccentric Rotating Mass
GUI	Graphical User Interface
I^2C	Inter-Integrated Circuit Communication Standard
LRA	Linear Resonant Actuator
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
BEMF	Back Electro-Motive Force

2 References

[1] DA7280, Datasheet, Renesas Electronics.
[2] DA7281, Datasheet, Renesas Electronics.
[3] DA7282, Datasheet, Renesas Electronics.
[4] DA7283, Datasheet, Renesas Electronics.
[5] UM-HA-003, DA7280 Daughterboard User Manual, Renesas Electronics.
[6] UM-HA-004, DA7281 Daughterboard User Manual, Renesas Electronics.
[7] UM-HA-005, DA7282 Daughterboard User Manual, Renesas Electronics.
[8] UM-HA-006, DA7283 Daughterboard User Manual, Renesas Electronics.
[9] DA728x Haptics GUI software.

3 Introduction

This document describes the use and connections of DA728x Motherboard 359-05- to allow the evaluation of the DA728x family of devices. Figure 1 shows a picture of the DA728x Motherboard 359-05-B. A PCB layout is shown in Figure 2 and the details of all the jumper connections are listed in Table 1.

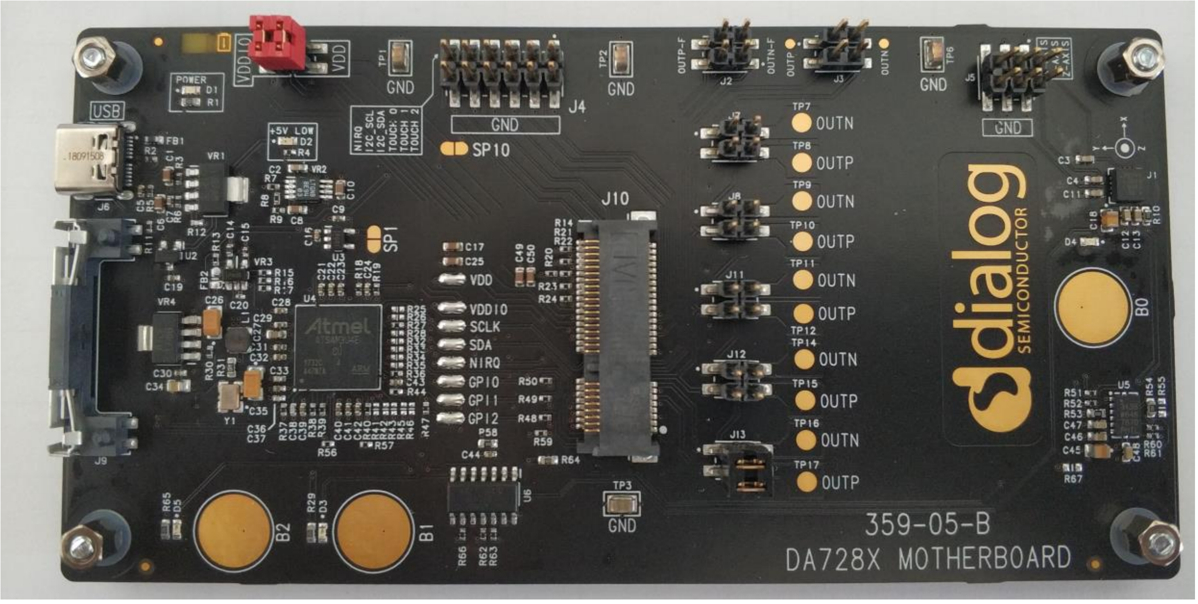

Figure 1. Bare 359-05-B motherboard

4 DA728x Motherboard 359-05-B PCB Layout and Connections

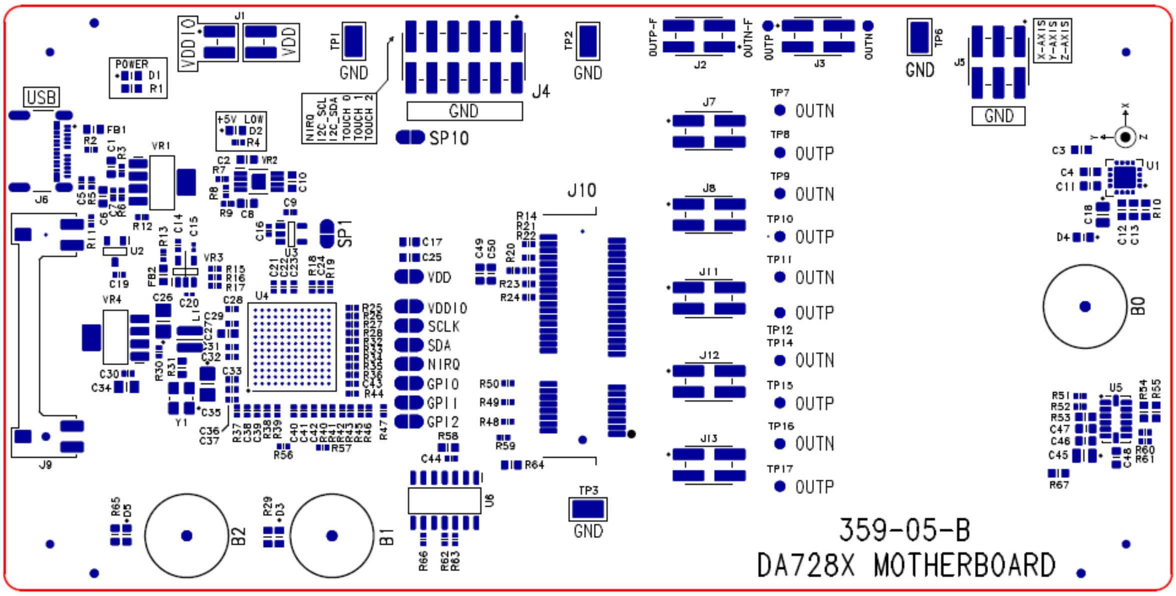

Figure 2. DA728x motherboard 359-05-B PCB layout

Table 1. Jumper connections and descriptions

Jumper/Test Point (TP)	Pin/Connection	Description
J1	VDDIO **[Default: short 1 to 3]**	VDDIO current monitoring or supply option, default level 1.8 V.
	VDD **[Default: short 2 to 4]**	VDD current monitoring or supply option, default level 3.8 V.
J2	OUTN-F	Filtered version of negative output. Bandwidth limited to 3.38 kHz for output waveform analysis through an oscilloscope (pin 1 and pin 3 are shorted on the PCB).
	OUTP-F	Filtered version of positive output. Bandwidth limited to 3.38 kHz for output waveform analysis through an oscilloscope (pin 2 and pin 4 are shorted on the PCB).
J3	OUTP	Positive output for LRA connection (pin 1 and pin 3 are shorted on the PCB. TP4 can also be used).
	OUTN	Negative output for LRA connection (pin 2 and pin 4 are shorted on the PCB. TP5 can also be used).
J4	TOUCH 2	Pin 1: touch button B2 signal.
	TOUCH 1	Pin 3: touch button B1 signal.
	TOUCH 0	Pin 5: touch button B0 signal.
	I2C_SDA	Pin 7: I2C data line (no pull-up fitted as this is on the daughtercard).
	I2C_SCL	Pin 9 I2C clock line (no pull-up fitted as this is on the daughtercard).
	NIRQ	Pin 11 interrupt trigger, open drain output (no pull-up fitted on motherboard, pull-up located on the daughtercard).
	GND	Pins 2, 4, 6, 8, and 10 = Ground reference.
J5	Z-AXIS	Pin 1: Z-axis accelerometer output.
	Y-AXIS	Pin 3: Y-axis accelerometer output.
	X-AXIS	Pin 5: X-axis accelerometer output.
	GND	Pins 2, 4, and 6 = Ground reference.
J6	USB	USB-C type connector used to supply power to the motherboard and the source of control signals through the SmartCanvas DA728x GUI.
J7	OUTN	Negative output for LRA connection (pin 1 and pin 2 should be shorted to enable TP7).
	OUTP	Positive output for LRA connection (pin 3 and pin 4 should be shorted to enable TP8).
J8	OUTN	Negative output for LRA connection (pin 1 and pin 2 should be shorted to enable TP9).
	OUTP	Positive output for LRA connection (pin 3 and 4 should be shorted to enable TP10).
J9	Mechanical lock for the DA728x daughterboard	Used to hold the daughterboard in place when fitted.
J10	Socket for the DA728x daughterboard	Electrical connections to/from the DA728x daughterboard when fitted.

Jumper/Test Point (TP)	Pin/Connection	Description
J11	OUTN	Negative output for LRA connection (pin 1 and pin 2 should be shorted to enable TP11).
	OUTP	Positive output for LRA connection (pin 3 and pin 4 should be shorted to enable TP12).
J12	OUTN	Negative output for LRA connection (pin 1 and pin 2 should be shorted to enable TP14).
	OUTP	Positive output for LRA connection (pin 3 and pin 4 should be shorted to enable TP15).
J13	OUTN **[Default – short 1 to 2]**	Negative output for LRA connection (pin 1 and pin 2 should be shorted to enable TP16).
	OUTP **[Default – short 3 to 4]**	Positive output for LRA connection (pin 3 and pin 4 should be shorted to enable TP17).
TP1, TP2, TP3, and TP6	GND	Ground reference (ideal for scope probe reference clips).

5 Hardware and Software Prerequisites

You must install the GUI including the USB driver before the DA728x motherboard hardware is connected to the USB port of PC running a Windows operating system. You should get acquainted with the DA728x motherboard hardware (Section 6) before proceeding with the configuration steps.

The DA728x GUI installation file is included on the USB stick contained within the Evaluation Board kit. To start the installation, run the `setup_DA728x_GUI_0.0.0.XXX.exe` file from the `\Software` folder – XXX depends on the version of software supplied on the USB stick.

Also, you can find the latest version of the SmartCanvas DA728x GUI on the Renesas support portal Haptic Drivers | Renesas.

5.1 Hardware configuration

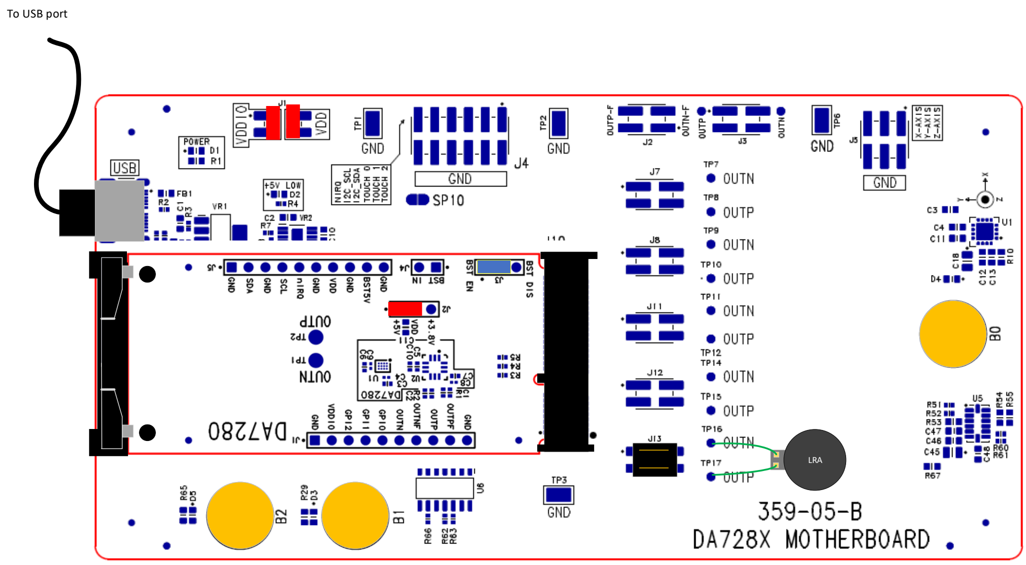

Figure 3. DA728x motherboard operation with LRA attached to DA728x motherboard

6 DA728x Motherboard Hardware Overview

The DA728x motherboard is powered and controlled from the onboard USB circuitry. The device-under-test (DUT) can be completely isolated from the USB controller circuitry in terms of power, I2C, or general-purpose input (GPI) signals if needed by de-soldering the solder bumps connections shown in Figure 4.

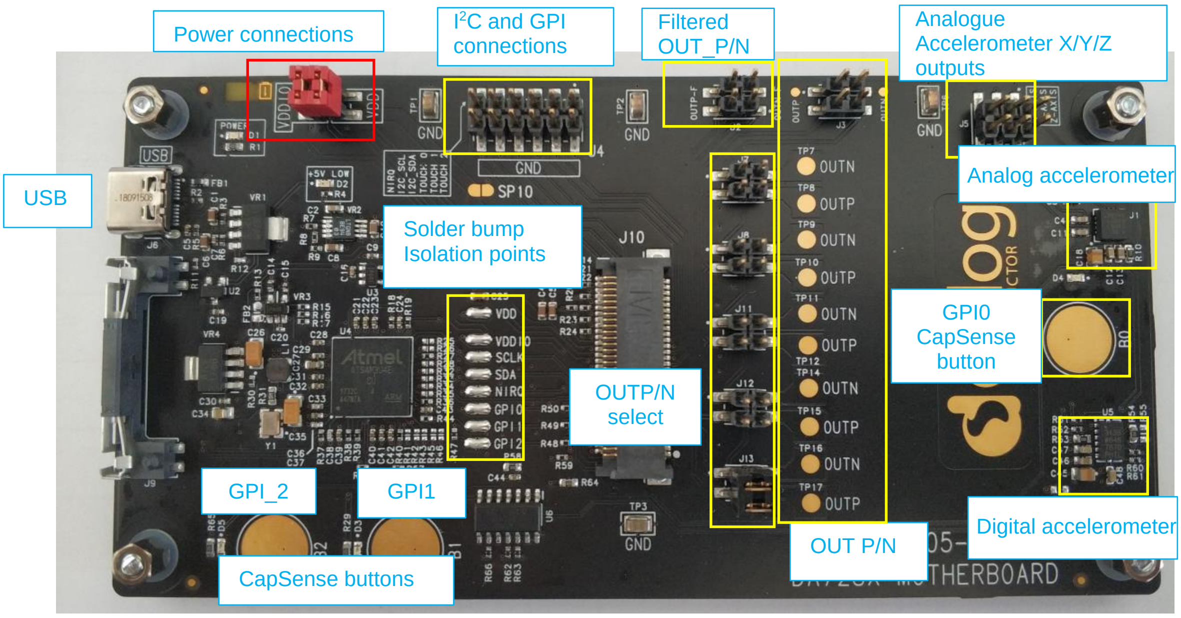

Figure 4. DA728x motherboard functionality

6.1 Power

The DA728x motherboard is powered from the +5 V USB connection (J6), the board has several regulators that supply the various power rails of the board through this +5 V supply. Additional external power supplies are not required, but there is an option to use separate supplies if needed.

VDDIO supplies the digital IO level of the DA7280 daughterboard DUT at 1.8 V through the onboard regulator. There is also an option to set the VDDIO level to 3V3 with a small modification to the board (see motherboard schematic supplied with the kit – remove R13 and place on FB2). VDDIO can also be supplied externally through a jumper as discussed further in this section.

VDD supplies the DUT VDD supply and is regulated at 3.8 V through the onboard regulator. This VDD level allows the D728x motherboard to supply the LRA load up to approximately 2.8 V pk-pk. If a higher output voltage is required, there is an option to allow the supply to be driven from the 5 V USB supply or an external VDD supply through the jumper pins.

An external VDD supply is not required when the DA7280 daughterboard is used as there is an onboard 5 V boost regulator which is set as the default supply, although the current is limited due to the limitation of the supply current from the USB port. This allows the 3.8 V VDD from the motherboard to be boosted to 5 V in the default setting when using the DA7280 daughterboard. When using the DA7281/2/3 boards, then VDD is supplied at 3.8 V only.

The DA728x devices can drive 250 mA into the LRA load through the VDD supply, so the USB power is adequate for most applications. The DA7280/1/2 devices can also drive up to 500 mA when double output current range is enabled in the register settings, therefore it is recommended that an external VDD supply is used in this mode so that the USB supply is not overloaded.

To externally power the DA728x motherboard for accurate current measurements or supply at a different VDD or VDDIO voltage, the jumpers on J1 can be removed to directly power VDD (J1 pin 4) and VDDIO (J1 pin 3). Alternatively, the jumpers can be removed and replaced with ammeters to measure the current flow through VDD or VDDIO to the DA7280 daughterboard connected to J10.

6.2 I²C communications

The SmartCanvas DA728x GUI communicates to the daughterboard connected to J10 through the USB-to-I2C interface. The DA7280/2 slave address is 0x4A, when adding the R/W bit the I2C read = 0x94 and the I2C write = 0x95.

When the DA7281 daughterboard is used, one of four I2C addresses can be selected using the onboard jumpers so that four DA7281 devices can be selected on the same I2C bus, allowing multiple LRAs to be driven. The address that can be selected with the jumper configurations are 0x48, 0x49, 0x4A, or 0x4B.

To use external I2C communication, desolder the SDA (Shorting point 5 – Marked SP5) and SCL (Shorting point 4 – Marked SP4) solder bumps and connect the external I2C wires to J4, SCL (J4 pin 9), and SDA (J4 pin 7). See **I2C and GPI connections** in Figure 4. If external I2C signaling is used, ensure that the VDDIO voltage level is the same as the external I2C signal voltage level.

6.3 Plug-in daughterboards

The DA728x motherboard supports several plug-in daughterboards included with the kit which can be easily inserted into the PCB. The daughterboards are inserted into the J10 connector of the DA728x motherboard and should be inserted so that the gold fingers of the daughterboard align with the gold pins of the J10 connector, the DUT on the daughterboard should be visible facing upwards. The daughterboard should be inserted at an angle of approximately 15 to 30 degrees into the J10 connector, then be pushed towards the connector so that the gold fingers are almost completely hidden by the connector (Figure 5).

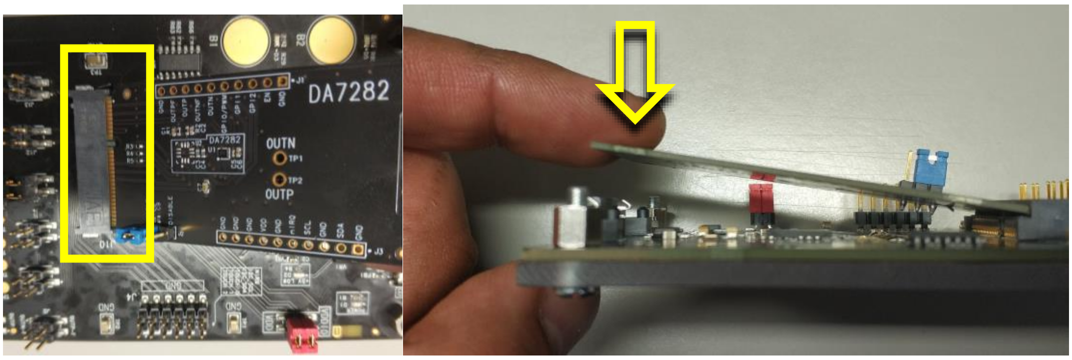

Figure 5. Daughterboard insertion

The back end of the daughterboard should then be lowered to mate with J9 on the motherboard so that the back end of the daughterboard clips into the J9 connector as shown in Figure 6.

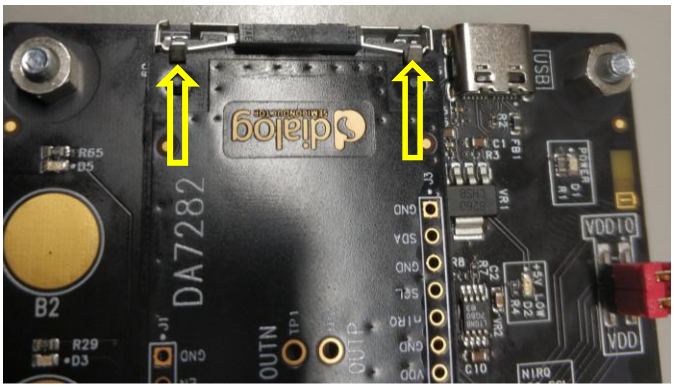

Figure 6. Daughterboard fully inserted with release tabs shown

To remove the daughterboard, press the two metal clips on J9 so that the daughterboard is released and can then be removed from J9 and pulled out of J10.

6.3.1 DA7280 daughterboard

Figure 7 shows the DA7280 daughterboard PCB. The device is located on either position U1 (WLCSP, default) or U2 (QFN) depending on the package variant on the board, the silicon in both package variants is identical.

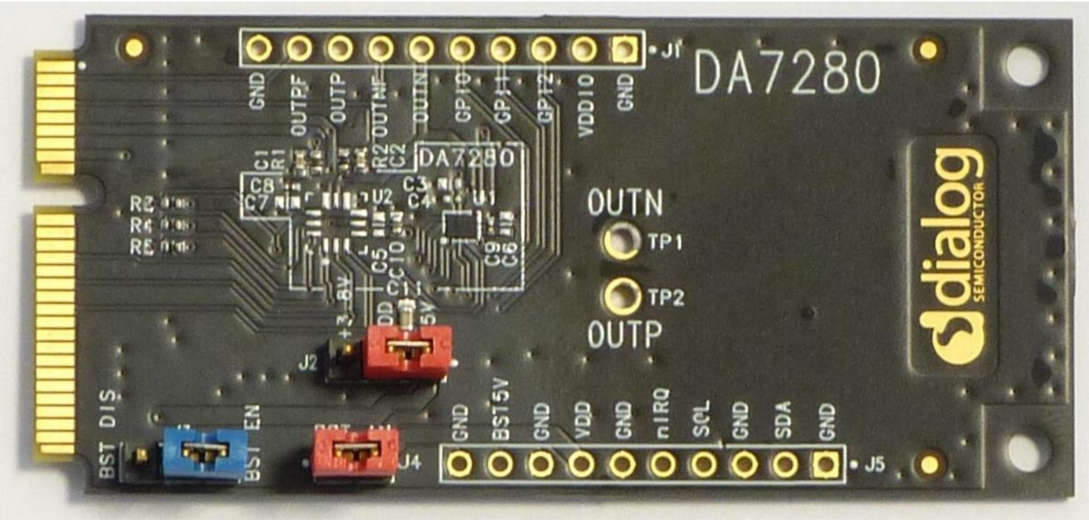

Figure 7. DA7280 daughterboard

After a power-on reset of the DA7280 daughterboard, download the LRA configuration script (see Section 9.1.1) for the supplied LRA. This ensures the correct DC parameters are driving the LRA according to the LRA's datasheet. If a different LRA is used, the settings should be adjusted according to that specific LRA’s electrical parameters.

To drive the LRA, DA7280 outputs PWM differential signals that are routed to several locations on the DA728x motherboard where an LRA can be connected:

- Solder pads labeled OUTP and OUTN (TP7 to TP12 and TP14 to TP17). A shorting jumper must be connected to J7, J8, J11, J12, or J13 [default] to select the appropriate LRA connected. The default LRA is connected to TP16 and TP17.
- J3 header pins labeled OUTP and OUTN.

Connect an LRA to only one of these positions at one time. Alternatively, LRAs can be connected from TP7 to TP12 and TP14 to TP17. A double jumper can then be connected to J7, J8, J11, J12, or J13 to select which LRA is to be used. Ensure that the correct script file is loaded for the LRA before it is selected and driven.

Filtered OUT_P and OUT_N signals from DA7280 are available for debug through the OUTP_F and OUTN_F pins (J2). Connect these signals from J2 to an oscilloscope for signal monitoring and tuning purposes (see Section 10).

When the DA7280 daughterboard inserted into the DA728x motherboard is used, it is recommended that the LRA should be placed on the motherboard for the true performance measurements of the acceleration. For more information about the hardware and connections available on the board, see the DA7280 daughterboard user manual, Ref. [4].

6.3.2 DA7281 daughterboard

Figure 8 shows the DA7281 daughterboard PCB. The device is located on either position U1 (WLCSP) or U2 (QFN, default) depending on the package variant on the board, the silicon in both package variants is identical.

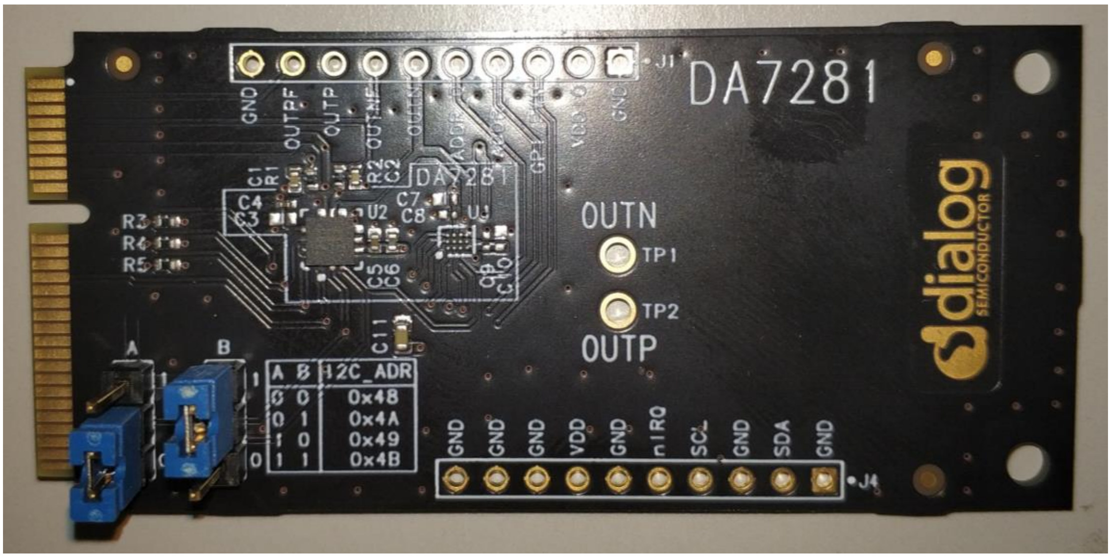

Figure 8. DA7281 daughterboard

After a power-on reset of the DA7281 daughterboard, download the LRA configuration script (see Section 9.1.1) for the supplied LRA. This ensures the correct DC parameters are driving the LRA according to the LRA's datasheet. If a different LRA is used, the settings should be adjusted according to that specific LRA's electrical parameters.

To drive the LRA, DA7281 outputs PWM differential signals which are routed to several locations on the DA728x motherboard where an LRA can be connected:

- Solder pads labeled OUTP and OUTN (TP7 to TP12 and TP14 to TP17). A shorting jumper must be connected to J7, J8, J11, J12, or J13 [default] to select the appropriate LRA connected. The default LRA is connected to TP16 and TP17.
- J3 header pins labeled OUTP and OUTN.

Connect an LRA to only one of these positions at one time. Alternatively, LRAs can be connected from TP7 to TP12 and TP14 to TP17. A double jumper can then be connected to J7, J8, J11, J12, or J13 to select which LRA is to be used. Ensure that the correct script file is loaded for the LRA used before it is selected and driven.

Filtered OUT_P and OUT_N signals from DA7281 are available for debug through the OUTP_F and OUTN_F pins (J2). Connect these signals from J2 to an oscilloscope for signal monitoring and tuning purposes (see Section 10).

When the DA7281 daughterboard is inserted into the DA728x motherboard, it is recommended that the LRA should be placed on the motherboard for the true performance measurements of the acceleration. For more information about the hardware and connections available on the board, see the DA7281 daughterboard user manual, Ref. [6].

6.3.3 DA7282 daughterboard

Figure 9 shows the DA7282 daughterboard PCB. The device is located on either position U1 (WLCSP, default) or U2 (QFN) depending on the package variant on the board, the silicon in both package variants is identical.

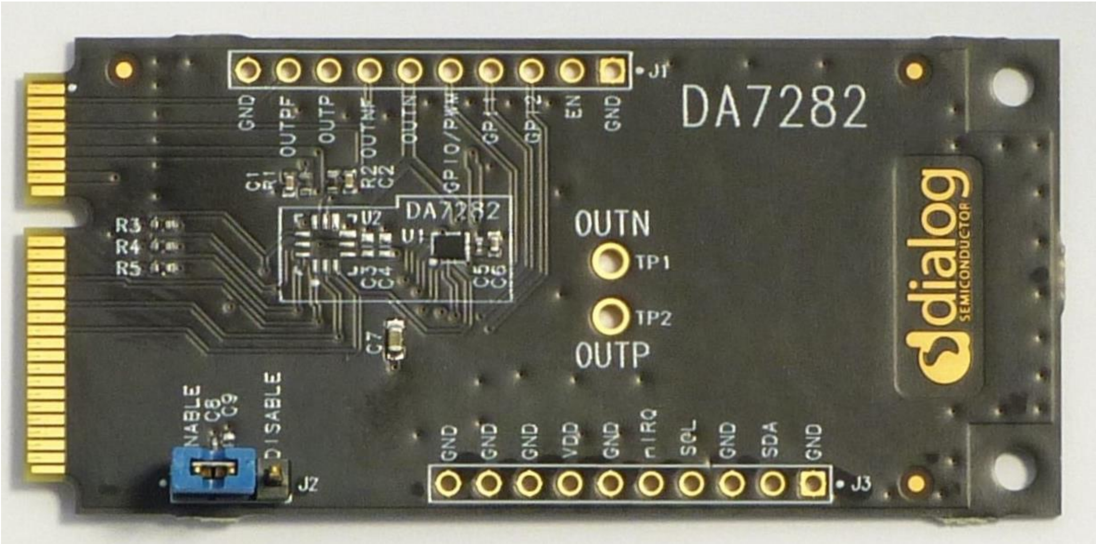

Figure 9. DA7282 daughterboard

After a power-on reset of the DA7282 daughterboard, download the LRA configuration script (see Section 9.1.1) for the supplied LRA. This ensures the correct DC parameters are driving the LRA according to the LRA's datasheet. If a different LRA is used, the settings should be adjusted according to that specific LRA's electrical parameters.

To drive the LRA, DA7282 outputs PWM differential signals which are routed to several locations on the DA728x motherboard where an LRA can be connected:

- Solder pads labeled OUTP and OUTN (TP7 to TP12 and TP14 to TP17). A shorting jumper must be connected to J7, J8, J11, J12, or J13 [default] to select the appropriate LRA connected. The default LRA is connected to TP16 and TP17.
- J3 header pins labeled OUTP and OUTN.

Connect an LRA to only one of these positions at one time. Alternatively, LRAs can be connected from TP7 to TP12 and TP14 to TP17. A double jumper can then be connected to J7, J8, J11, J12, or J13 to select which LRA is to be used. Ensure that the correct script file is loaded for the LRA being used before it is selected and driven.

Filtered OUT_P and OUT_N signals from DA7282 are available for debug through the OUTP_F and OUTN_F pins (J2). Connect these signals from J2 to an oscilloscope for signal monitoring and tuning purposes (see Section 10).

When the DA7282 daughterboard is inserted into the DA728x motherboard, it is recommended that the LRA should be placed on the motherboard for the true performance measurements of the acceleration. For more information about the hardware and connections available on the board, see the DA7282 daughterboard user manual, Ref. [7].

6.3.4 DA7283 daughterboard

Figure 10 shows the DA7283 daughterboard PCB. The device is located on either position U1 (WLCSP) or U2 (QFN, default) depending on the package variant on the board, the silicon in both package variants is identical.

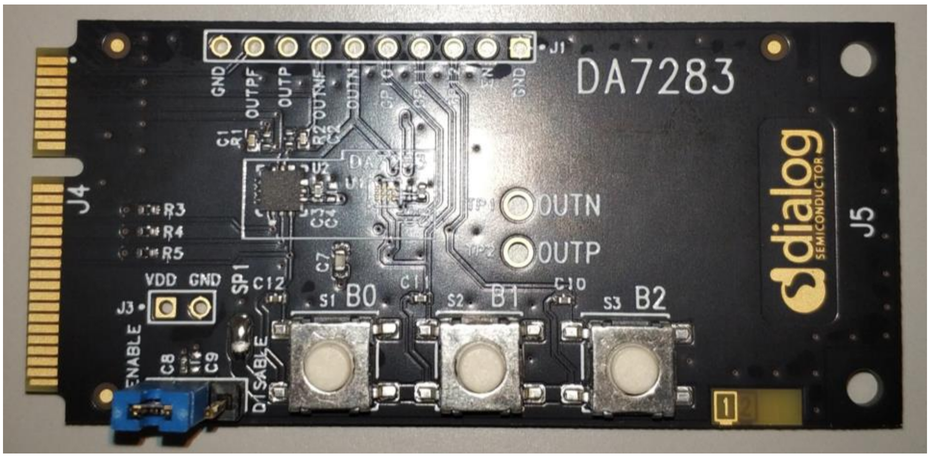

Figure 10. DA7283 daughterboard

The DA7283 device is designed to drive 95% of LRAs without configuration where a simple click or buzz is required by driving the GPI signals of the device. The DA7283 has no I2C bus for programming or tuning.

NOTE
The DA7283 supports only LRAs and does not support ERM motors.

The insertion of the DA7283 daughterboard allows the DA728x motherboard to power the daughterboard and allows the operation of the cap-sense touch buttons to operate the drive signals. The onboard push switches can also be used to operate the drive signals.

To drive the LRA, DA7283 outputs PWM differential signals which are routed to several locations on the DA728x motherboard where an LRA can be connected:

- Solder pads labeled OUTP and OUTN (TP7 to TP12 and TP14 to TP17). A shorting jumper must be connected to J7, J8, J11, J12, or J13 [default] to select the appropriate LRA connected. The default LRA is connected to TP16 and TP17.
- J3 header pins labeled OUTP and OUTN.

Connect an LRA to only one of these positions at one time. Alternatively, LRAs can be connected from TP7 to TP12 and TP14 to TP17. A double jumper can then be connected to J7, J8, J11, J12, or J13 to select which LRA is to be used.

When selecting a new LRA, ensure that the motherboard is powered down and up again as the DA7283 performs a calibration cycle to determine the drive parameters only when first driving after a first power-up. Connecting a new LRA without powering down the device results in incorrect drive signals.

Filtered OUT_P and OUT_N signals from DA7283 are available for debug and are accessed from the OUTP_F and OUTN_F pins (J2). Connect these signals from J2 to an oscilloscope for signal monitoring and tuning purposes (see Section 10).

When the DA7283 daughterboard is inserted into the DA728x motherboard, it is recommended that the LRA should be placed on the motherboard for the true performance measurements of the acceleration.

No interaction with the SmartCanvas DA728x GUI is possible when the DA7283 daughterboard is used as there is no I2C bus. For more information about the hardware and connections available on the board, see the DA7283 daughterboard user manual, Ref. [8].

6.4 Analog accelerometer

A high-precision analog accelerometer (U1) is fitted to the DA728x motherboard to capture the acceleration profile of the LRA (see Figure 4). This is used to evaluate and tune the LRA's performance (see Section 10) by using an oscilloscope to look at the X, Y, or Z movement.

6.5 Digital accelerometer

An additional high-precision digital accelerometer (U5, Figure 4) is fitted to the DA728x motherboard to capture the acceleration profile of the LRA which can be sampled by the SmartCanvas DA728x GUI. This is used to evaluate and tune the LRA's performance (see Section 10) using the GUI alone.

6.6 Plastic mass connection

To create a 100-gram reference weight (industry standard for haptics), a tightly screwed plastic mass has been fitted to the DA728x motherboard PCB (Figure 11). This is useful for evaluating LRA performance with respect to the amount of acceleration (G) the LRA produces when attached to this reference weight.

The DA728x motherboard without the plastic mass is designed to weigh approximately 50 grams which includes the daughterboard weight and default LRA. This target weight is useful in evaluating LRAs for wearable applications as 50 grams is the industry standard weight used for wearables.

To modify the motherboard to reach a 50-gram mass target, the plastic mass can be removed by unscrewing the four screws/nuts located in each corner of the board, the screws and nuts should not be refitted as these will increase the weight to over 50 grams.

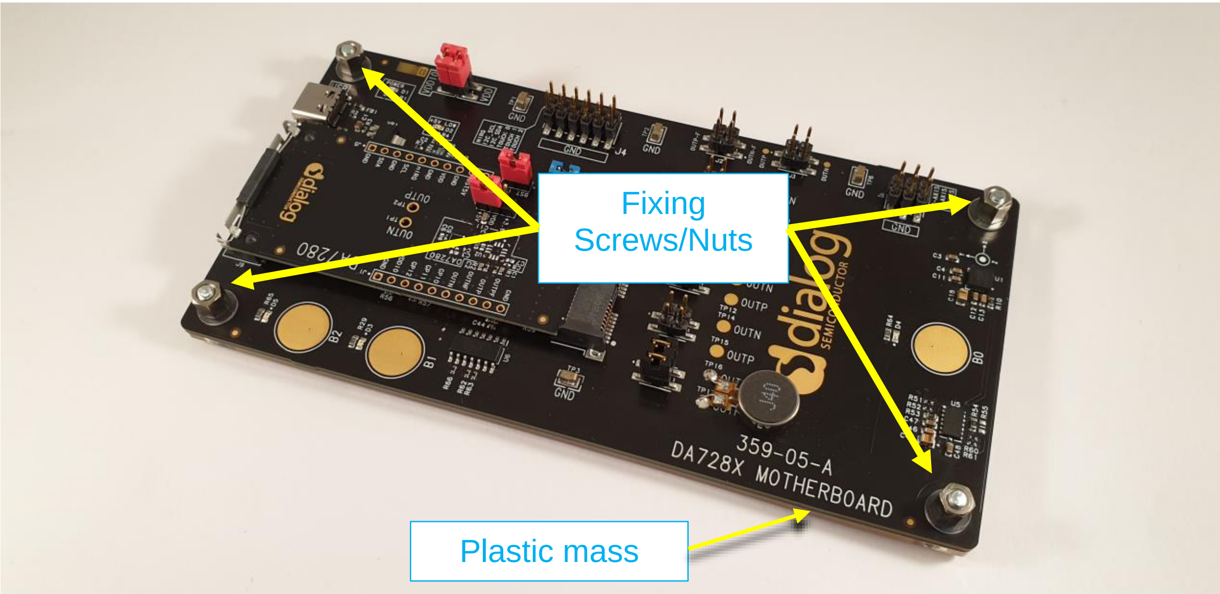

Figure 11. DA728x motherboard PCB combined with plastic mass = 100 grams

6.7 Additional LRA connection

If additional LRAs are to be attached to the DA728x motherboard PCB, it is recommended to use double sided sticky pads.

NOTE
Glue can be used but caution should be taken as it is possible that glue could enter the LRA and cause loss of performance or failure to operate.

Whichever fixing method is used, ensure that the newly added LRA is firmly connected and has no movement when driven, as movements of the LRA can result in a damping effect through loss of energy.

Connect the LRA terminals to the OUT_P and OUT_N test point pairs next to J7, J8, J11, J12, J13, or to the header J3 (see Figure 4). If the test points on OUT_P and OUT_N are used, ensure that the associated double jumper on J7, J8, J11, J12, or J13 is selected for the LRA to be used. In this way, several LRAs can be connected to the DA728x motherboard at the same time and the appropriate LRA can be easily selected by moving the double jumper link to the desired header. This can be very useful if several LRAs need to be quickly evaluated at one time.

Ensure that only one LRA is physically wired at a time when driving the LRA and analyzing the performance of each LRA. It is possible to drive two LRAs in parallel so long as the LRAs are wired in the same orientation and there is enough current to drive both LRAs. This, however, does not give the best performance as each LRA will have different characteristics which will affect the tracking algorithm.

If several LRAs are placed on the motherboard at one time, then the unconnected LRAs can absorb some of the energy created by the driven LRA. This can result in lower acceleration levels seen and unexpected resonance effects due to the undriven LRAs absorbing some of the energy and releasing them at a later point. Therefore, it is recommended that when creating performance waveforms, tuning or scripts for a particular LRA, then only one LRA should be mounted on the motherboard in this case.

7 Configuration Steps

1. Install the SmartCanvas DA728x GUI (see Section 8).
2. Plug the DA7280 daughterboard into the J10 connector of the DA728x motherboard and ensure that it is fully inserted and the clips on J9 secure the daughterboard in place (Section 6.3).
3. Connect the USB cable from the computer to the connector J6 on the DA728x motherboard and ensure that it is fully inserted into the connector.
   a. If this is the first time a DA728x board is connected to the computer, Windows automatically installs the USB drivers. If this fails to install, then see manual driver instructions.
4. If the LRA is attached to the DA728x motherboard as shown in Figure 3, ensure that the jumper is attached to J11 on the DA728x motherboard so that the DA7280 daughterboard is connected to the attached LRA. This is the recommended method of LRA attachment if performance measurements are being made with the onboard accelerometer.
5. For the best feedback performance, the DA728x motherboard should be isolated from hard surfaces as they can dampen the haptic effect. Placing the DA728x motherboard on a layer of foam is a good solution (Figure 12).
6. Open the SmartCanvas DA728x GUI and click **Download Jahwa 1040 defaults** as shown in Figure 22. This sets up the DA7280 daughterboard to work with the CapSense buttons (B0, B1, and B2) on the DA728x Motherboard.
7. Touching B0, B1, or B2 CapSense buttons on the DA728x motherboard triggers the programmed haptic sequences.

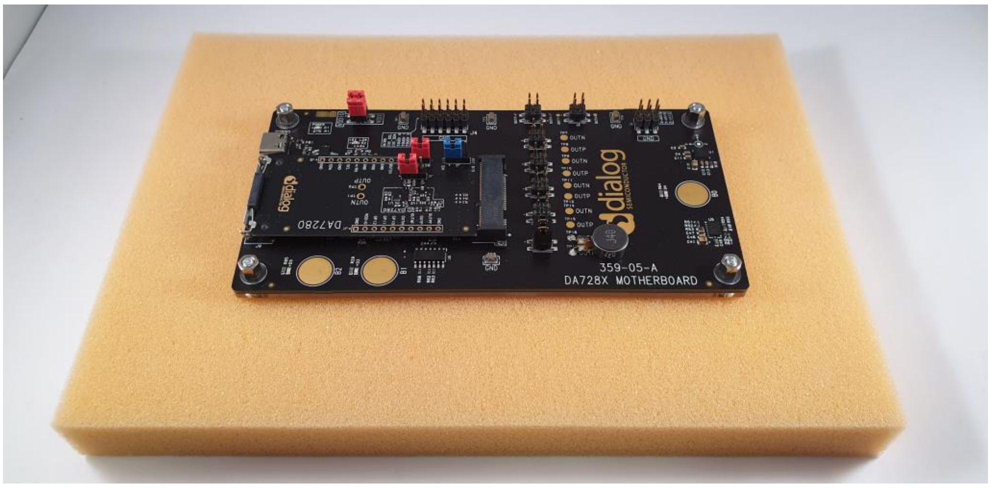

Figure 12. Placement of DA728x motherboard for optimal haptic feedback measurements

8 SmartCanvas DA728x GUI Setup

To set up SmartCanvas DA728x GUI, follow the steps in the setup wizard. Note version 0.30 is shown in the installation example but the version supplied may be newer.

1. Run the exe file located on the memory stick or downloaded from the website.
2. Click the **I accept the agreement** option and then click **Next** (Figure 13).

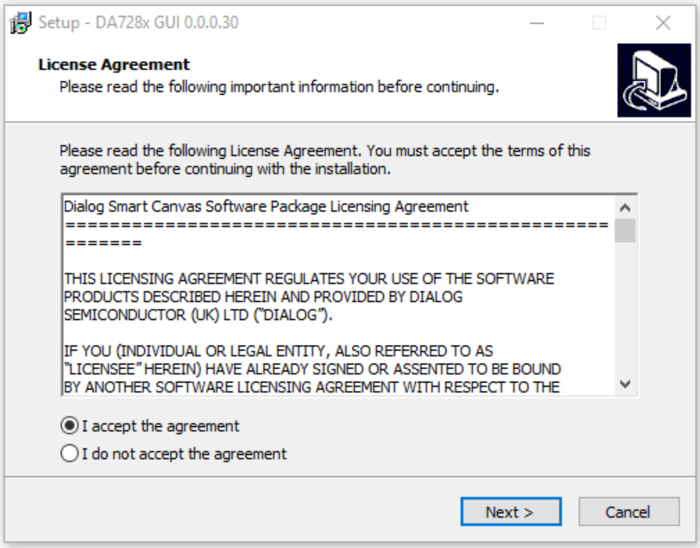

Figure 13. License agreement

3. Select the destination location for the SmartCanvas DA728x GUI to be installed and click **Next** (Figure 14).

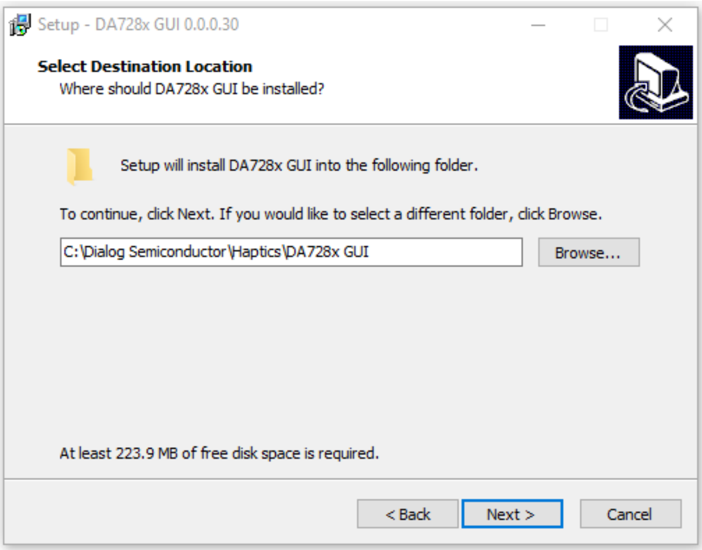

Figure 14. Select installation location

4. To create a shortcut for the SmartCanvas DA728x GUI in the Start menu, click **Next** (Figure 15).

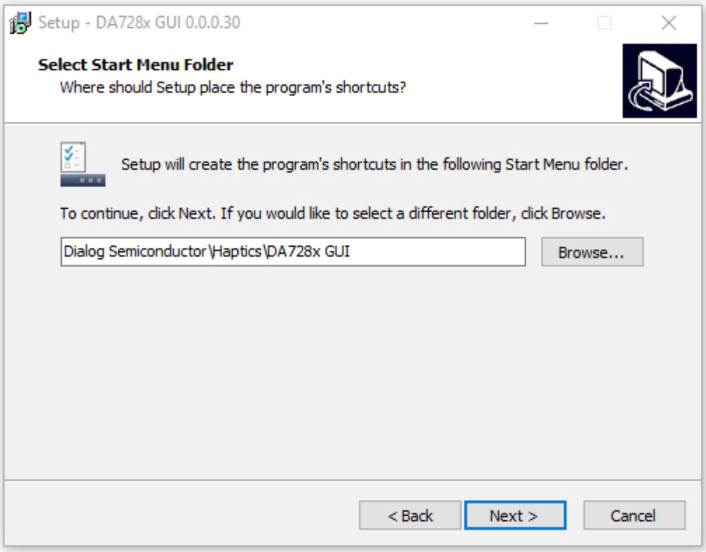

Figure 15. Select start menu folder

5. To create a desktop item for the SmartCanvas DA728x GUI, select the **Create a desktop shortcut** checkbox, and click **Next** (Figure 16).

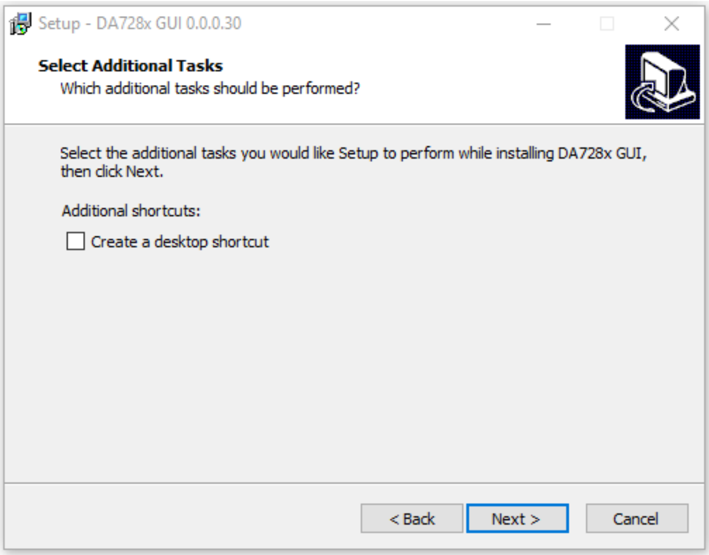

Figure 16. Create a desktop item

6. Click **Install**.

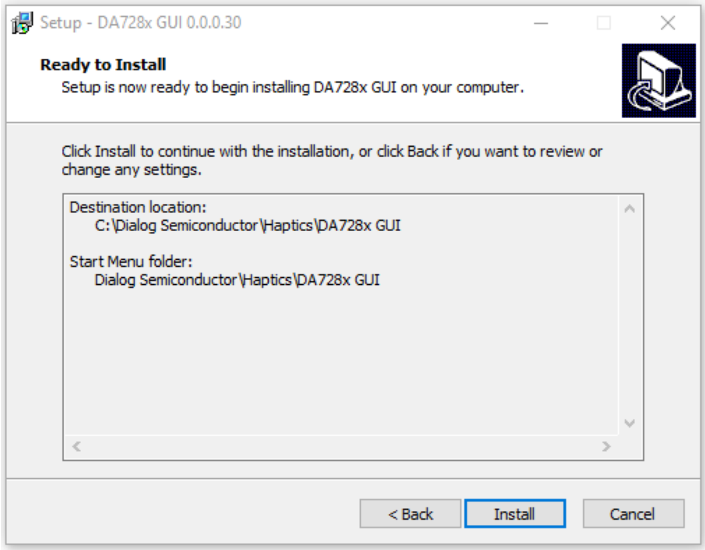

Figure 17. Install the SmartCanvas DA728x GUI

7. Clear the **Launch DA728x GUI** checkbox and click **Finish**.

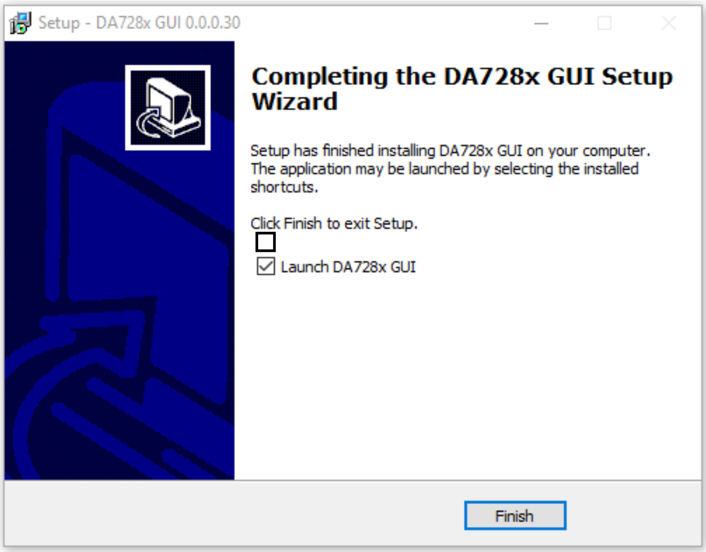

Figure 18. Finish setup

8. Connect the USB cable from the DA728x motherboard to the computer. Windows will now install the required driver (Figure 19).

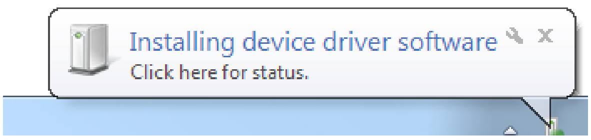

Figure 19. Installing driver windows notification

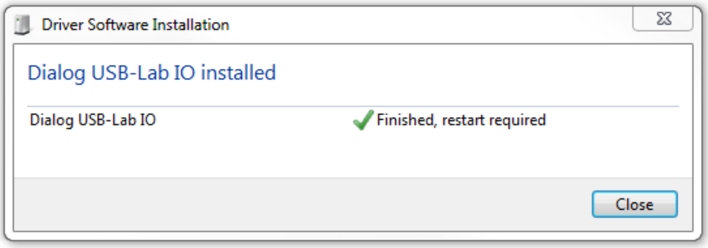

Figure 20. Installation complete

9. After the USB driver is installed (Figure 20), to run the DA728x GUI, double-click the desktop icon (Figure 21). If the driver does not install successfully, follow the manual driver installation instructions in Section 8.1.

Figure 21. Desktop icon

8.1 Manual driver installation

If there are any issues with the USB-IO driver installation, such as a failed installation or the driver has been accidently removed or deleted, you can easily reinstall the USB driver by going to `C:\Dialog Semiconductor\Haptics\DA728x GUI\driver` and double-clicking **Install USB-IO_driver** which starts the installation process.

9 Software Control

When the SmartCanvas DA728x GUI is started and the DA728x motherboard is connected to a PC through the USB cable, a dialog appears as shown in Figure 22. Click the **Download Jahwa 1040 defaults** option to set DA7280 in the edge triggered mode, so the CapSense buttons trigger haptic sequences.

NOTE
The **Download Jahwa 1040 defaults** option is only designed to be used with the default supplied LRA. When other LRAs are being evaluated, the load file IO should be used to load the correct script or the script should be developed if not available.

When other daughterboards such as DA7281 (359-07) or DA7282 (359-08) are connected to the DA728x motherboard, the diagram shown in Figure 22 changes according to the connected daughterboard. No diagram is displayed for the DA7283 daughterboard (359-09), as it has no I2C access and is therefore not able to communicate with the SmartCanvas DA728x GUI.

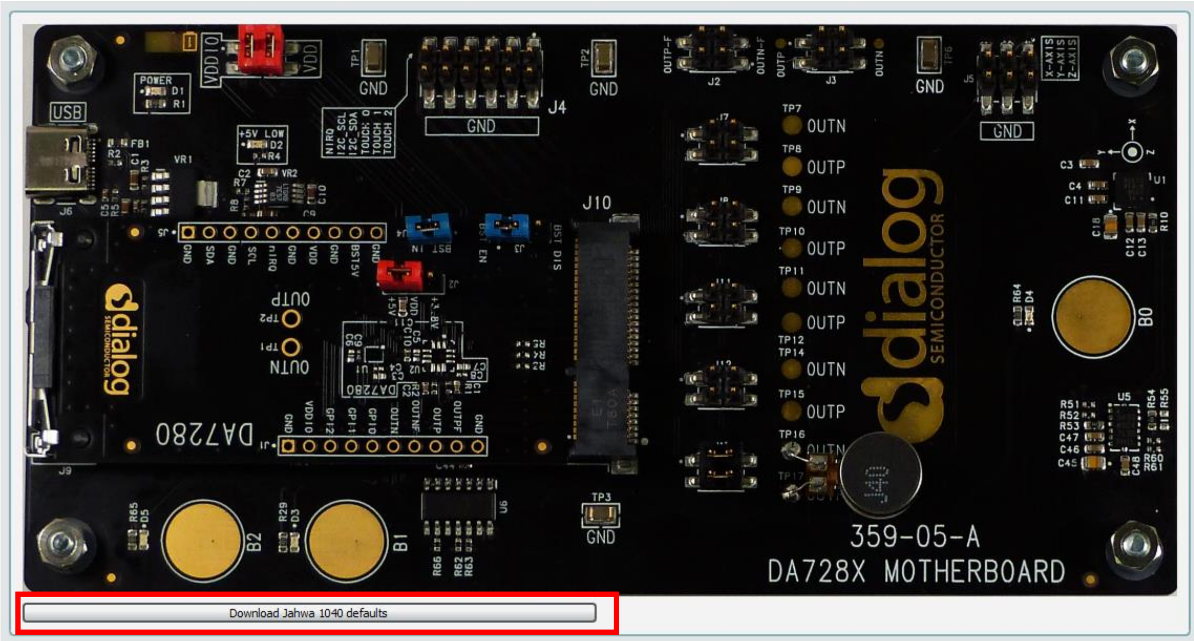

Figure 22. DA728x motherboard board setup with DA7280 daughterboard connected

The main window of the SmartCanvas DA728x GUI is shown in Figure 23, showing the status of the following indicators:

- **Bus communication** and **USB connection** LEDs are green, confirming that the SmartCanvas DA728x GUI has established communications with the DA728x motherboard.
- Under **Enable/Disable Polling**, the green button states **Enabled.**

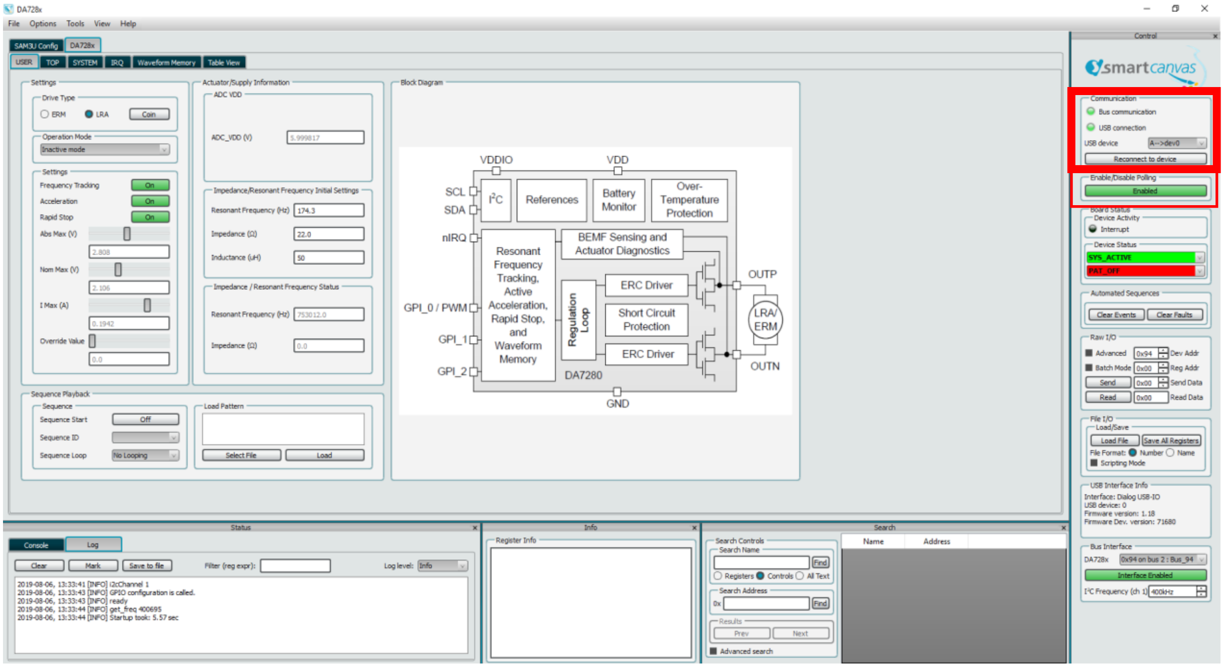

Figure 23. Main SmartCanvas DA728x GUI window

9.1 LRA configuration scripts

Four configuration scripts are bundled with the SmartCanvas DA728x GUI for the supplied JAHWA 1040 LRA. These scripts include the specific LRA configuration parameters, DA728x feature configuration parameters, and waveform memory data that are described in Section 9.1.4.

In addition to the Jahwa 1040 LRA which is supplied with the kit, in newer versions of the kit there are two new actuators included from TITAN Haptics, that are described in Section 9.2.

There are 260+ scripts covering various other LRA vendors with different driving methods, such as Frequency Tracking On/Off, Acceleration On/Off, Rapid Stop On/Off and sine wave mode. Ensure that the loaded script matches the connected LRA for the new LRA when changing from one LRA to another before starting sequences or operations.

9.1.1 Load LRA configuration scripts

1. In the **File I/O** section (Figure 24), click the **Load File** button.

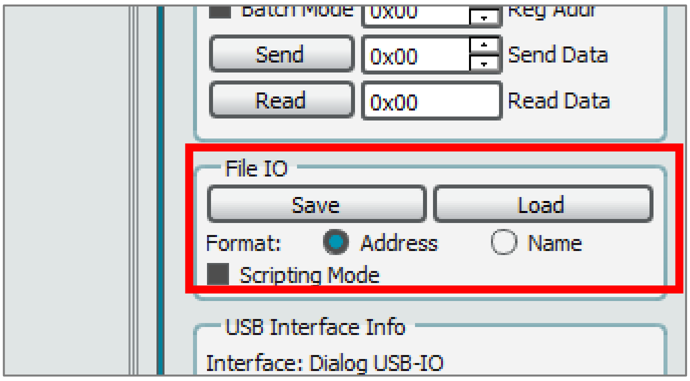

Figure 24. File I/O

2. Select the required `.txt` download script for the JAHWA LRA from the **Load File** dialog (Figure 25). For the supplied LRA only download scripts with "**JAHWA_1040_FREQ_TRACK**" prefix.

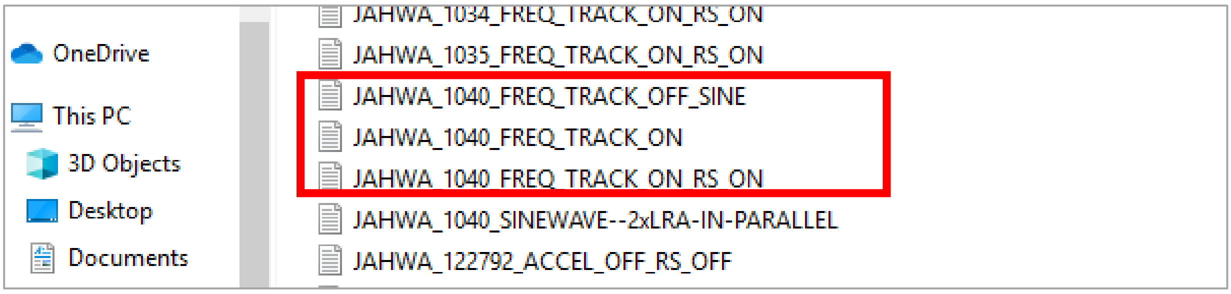

Figure 25. Load file I/O data

9.1.2 Save all registers

To save the status of DA728x registers:

1. In the **File I/O** section (Figure 24), select the **File Format** type to be **Number** (default) or **Name**. This refers to the register numbers or the register names.
2. Click **Save All Registers**.
3. In the **Save Register Dump** dialog, enter the file name and select the file type that you want. The default file type is `.txt`, however, `.csv` format also saves the IRQ status registers. The `.csv` format can be useful for debugging purposes.

9.1.3 Legacy SmartCanvas DA728x GUI script files

NOTE
The format of scripts differs between the current SmartCanvas DA728x GUI software and previous versions (pre .030 versions) which only supported the DA7280 device.

If the older dedicated DA7280 GUI is used, the scripts use "WRITE **DA7280**" (Figure 26).

The newer version of the GUI uses "WRITE **DA728x**". The current SmartCanvas DA728x GUI supplied with the DA728x motherboard supports both formats but older GUIs only support "WRITE DA7280".

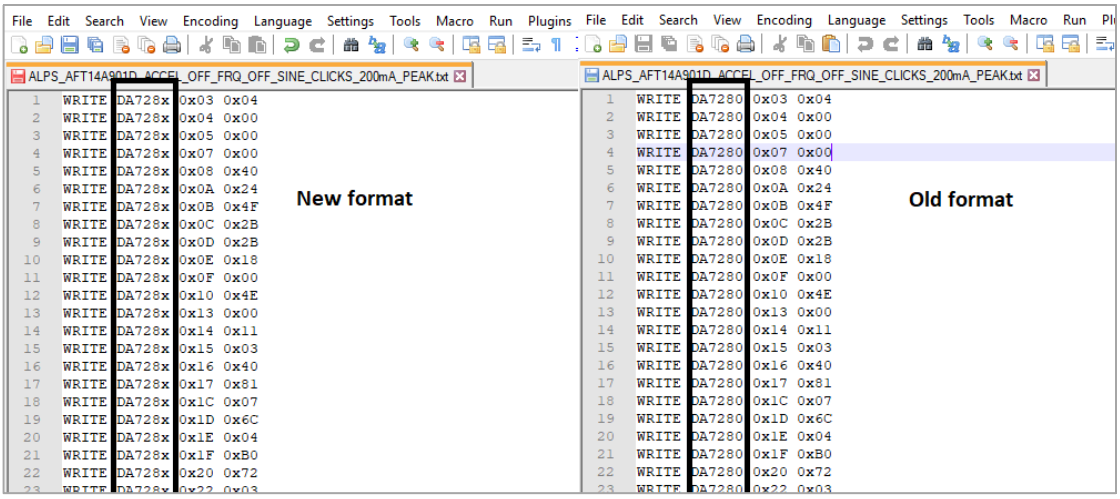

Figure 26. Old and new format script files

9.1.4 Configuration scripts for supplied LRA

The JAHWA 1040 LRA configuration scripts include specific JAHWA configuration parameters, DA728x features configuration parameters, and waveform memory data. This LRA model type is the 1040. It has fast rise time, meaning that the acceleration feature of DA728x does not need to be enabled.

This JAHWA 1040 LRA has the following specifications:

- Rated Voltage = 2.5 V(rms)
- Frequency = 170 Hz ± 10 Hz
- Vibration = 2.0 G
- Rated current = 170 mA
- Noise = 35 dB (max)
- Rise time (50%) = max 10 ms
- Fall time (50%) = max 50 ms
- Measured impedance = 13 Ω
- Measured inductance = 353 μH

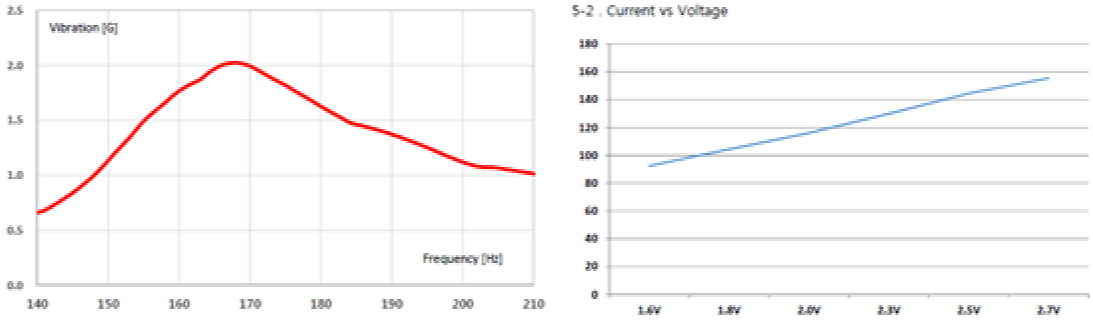

Figure 27. JAHWA 1040 LRA characteristics graph

In the following sections, you can read details regarding the four configuration scripts that are provided with the SmartCanvas DA728x GUI for the default supplied LRA. To load these scripts, see Section 9.1.1.

Table 2. JAHWA 1040 configuration script details

Section number	File name	Details
0	JAHWA_1040_FREQ_TRACK_ON.txt	Frequency Tracking is enabled, Acceleration and Rapid Stop are not active
9.1.4.2	JAHWA_1040_FREQ_TRACK_ON_RS_ON.txt	Frequency Tracking and Rapid Stop are enabled, Acceleration is not active
9.1.4.3	JAHWA_1040_FREQ_TRACK_OFF_SINE.txt	Sine wave mode - Frequency Tracking, Acceleration and Rapid Stop are not active. In sine wave mode the LRA can be driven in wide band mode, which allows the frequency to be anywhere between 25 Hz and 1 kHz.
9.1.4.4	JAHWA_1040_FREQ_TRACK_ON (2xLRA-IN-PARALLEL).txt	Double current mode - Frequency tracking enabled, Acceleration and Rapid Stop are not active. Note that this mode can only be used when two JAHWA 1040 LRAs are connected in parallel. A warning is displayed when loading this script as if only one LRA is connected when using it, the LRA is likely to be damaged due to the higher drive current used.

9.1.4.1 JAHWA_1040_FREQ_TRACK_ON

The `JAHWA_1040_FREQ_TRACK_ON.txt` script sets up the DA728x to enable the features shown in Table 3.

Table 3. JAHWA_1040_FREQ_TRACK_ON script features

Feature	Status
Frequency tracking	ON
Acceleration	OFF
Rapid stop	OFF

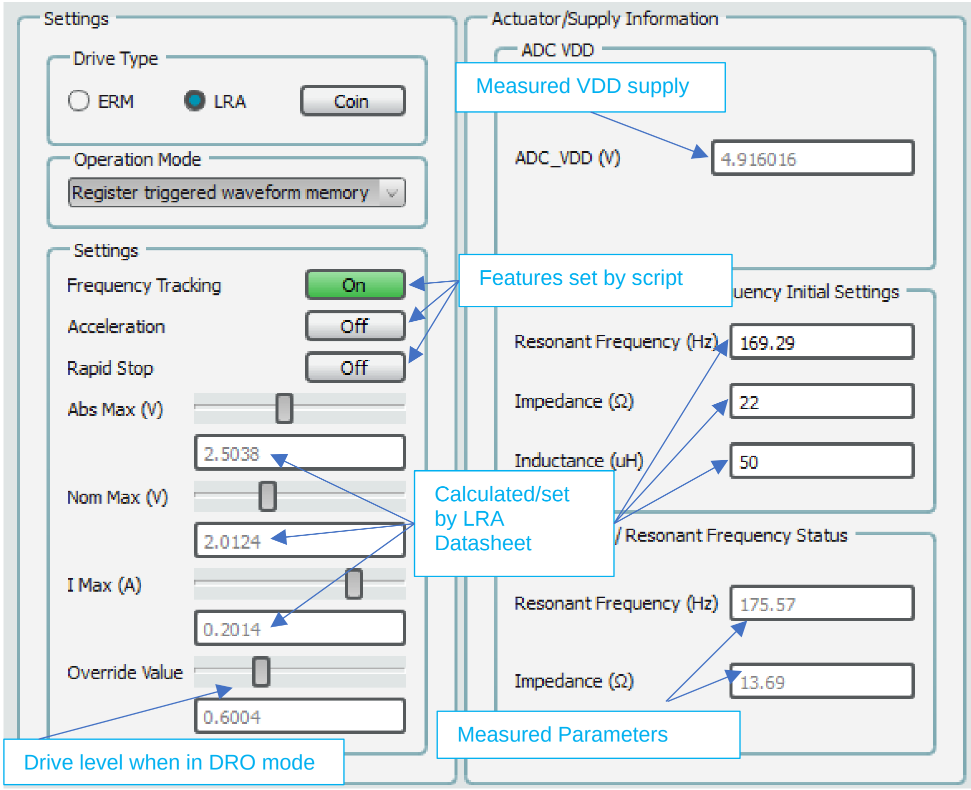

Figure 28. JAHWA_1040_FREQ_TRACK_ON parameters

The measured resonant frequency and impedance of the supplied LRA are shown in the DA728x GUI (Figure 28).

The **Abs Max** and **Nom Max** voltages for the LRA are calculated and set in the SmartCanvas DA728x GUI by the script as follows:

- Abs Max = (I MAX × 1.1) × measured impedance = 2.51 V
- Nom Max = (measured impedance × I MAX) × 0.707 = 2 V

NOTE
• These settings follow the LRA manufacturer's recommendations. It may be possible to overdrive above these levels for short periods but must be verified with the LRA manufacturer. • After downloading the LRA configuration scripts, the impedance and inductance values are not updated, but the underlying registers in DA728x are set correctly for the LRA.

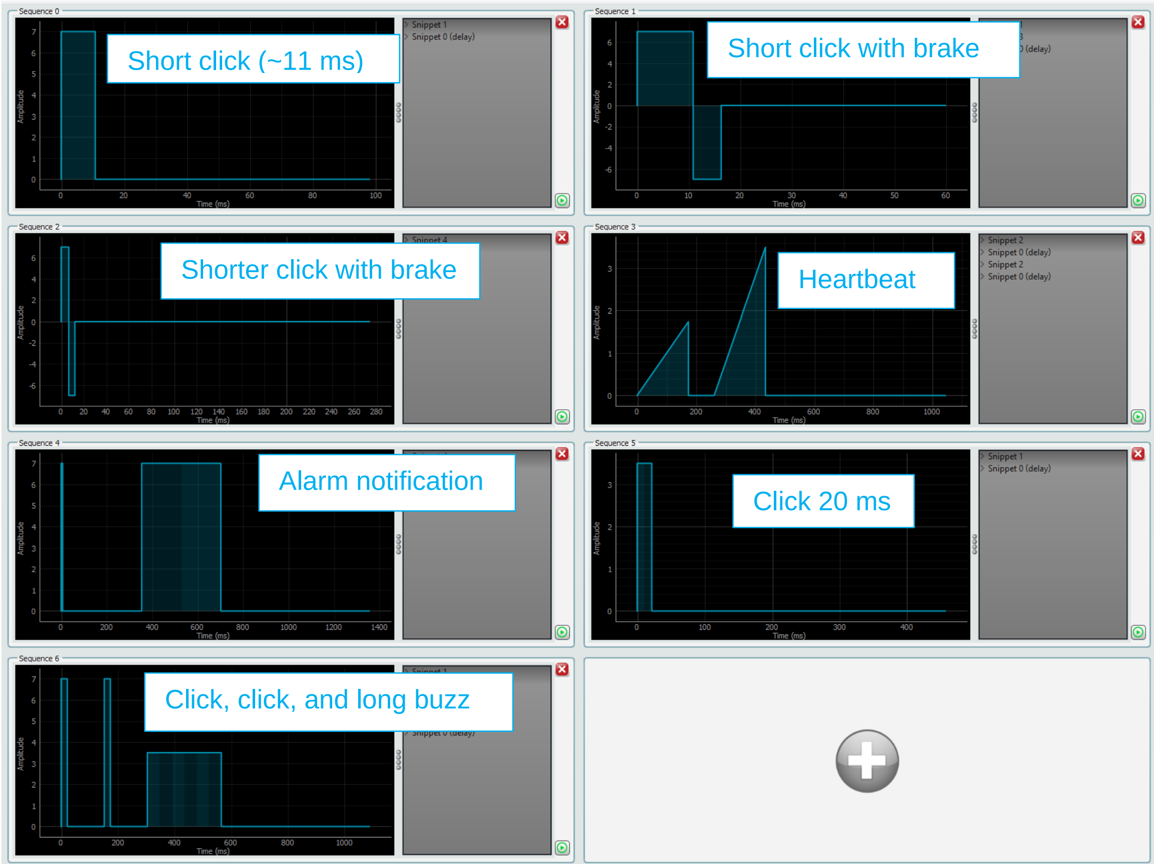

Figure 29. `JAHWA_1040_FREQ_TRACK_ON` sequences

The haptic sequences loaded from the configuration scripts are displayed in the DA728x GUI (see Section 9.4). For the `JAHWA_1040_FREQ_TRACK_ON` script, the sequences are shown in Figure 29.

9.1.4.2 JAHWA_1040_FREQ_TRACK_ON_RS_ON

The `JAHWA_1040_FREQ_TRACK_ON_RS_ON.txt` script sets up the DA728x to enable the features shown in Table 4.

Table 4. `JAHWA_1040_FREQ_TRACK_ON_RS_ON` script features

Feature	Status
Frequency tracking	ON
Acceleration	OFF
Rapid stop	ON

After the script is loaded, the SmartCanvas DA728x GUI shows the parameter settings applied by the configuration script (Figure 30).

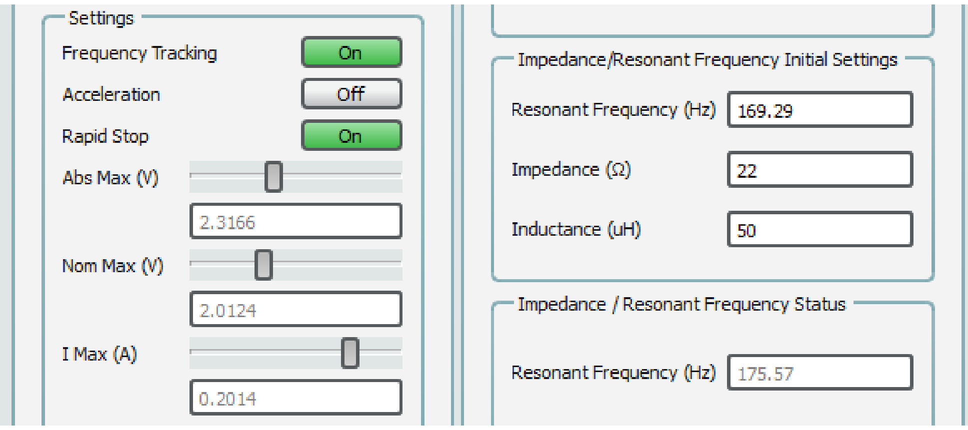

Figure 30. `JAHWA_1040_FREQ_TRACK_ON_RS_ON` parameters

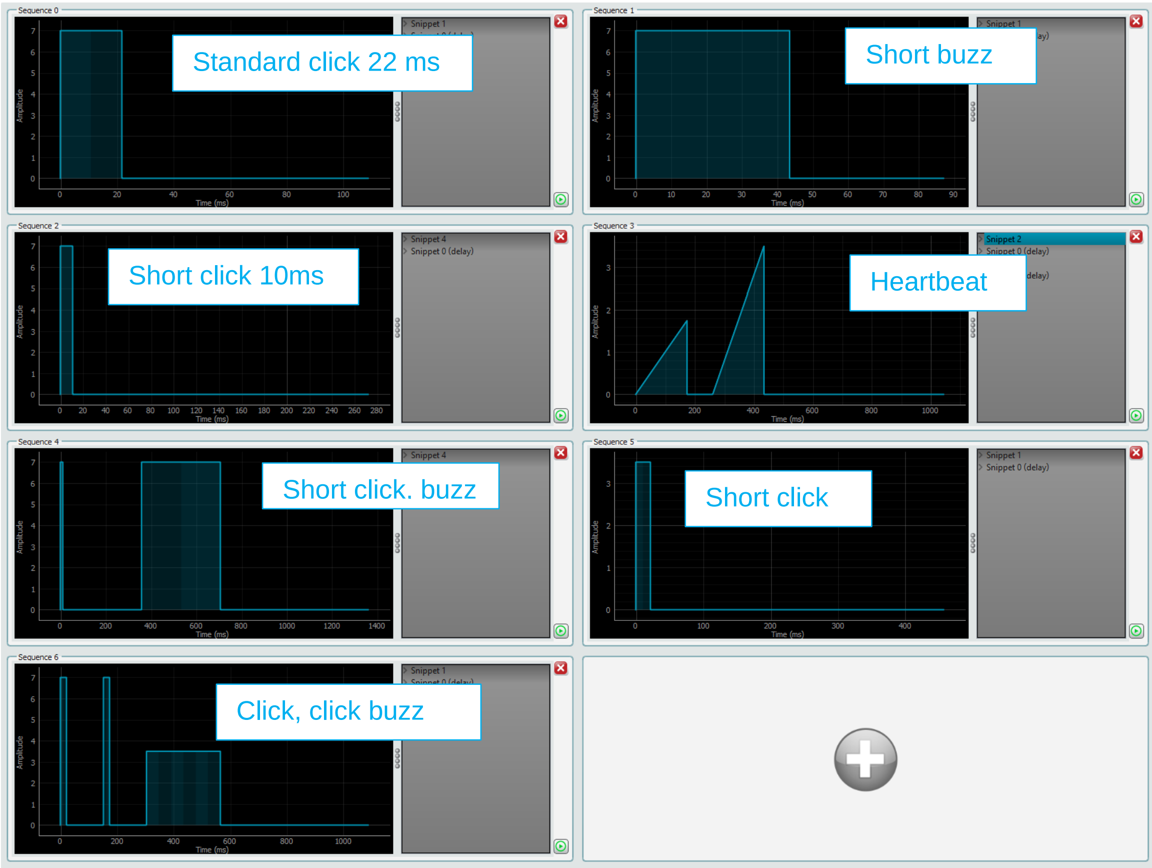

Figure 31. `JAHWA_1040_FREQ_TRACK_ON_RS_ON` sequences

The haptic sequences loaded from the configuration scripts are displayed in the DA728x GUI (Section 9.4). For the `JAHWA_1040_FREQ_TRACK_ON_RS_ON` script, the sequences are shown in Figure 31. The script uses Rapid Stop which is a feature where the DA728x attempts to automatically brake to stop the LRA as fast as possible by driving the LRA in the opposite phase after the waveform sequence is completed.

NOTE
The stop performance varies slightly on each attempt as the driver actively measures the BEMF to adjust the output to stop the LRA as quickly as possible.

9.1.4.3 JAHWA_1040_FREQ_TRACK_OFF_SINE

This script drives the output in sinewave mode. This mode reduces the LRA's audibility, but frequency tracking is not possible using this output drive. In this mode, the LRA is driven in wide band operation mode and the output frequency can be set from 25 Hz to 1 kHz. This mode can be useful to drive dual-axis LRAs where a different resonant frequency in the LRA design allows movements of the LRA in more than one direction.

Table 5 shows the features of this script and we can see that, besides Acceleration and Rapid Stop, Frequency Tracking is also off.

Table 5. JAHWA_1040_FREQ_TRACK_OFF_SINE script features

Feature	Status
Frequency tracking	OFF
Acceleration	OFF
Rapid stop	OFF

After the script is loaded, the SmartCanvas DA728x GUI shows the parameter settings applied by the configuration script (Figure 32).

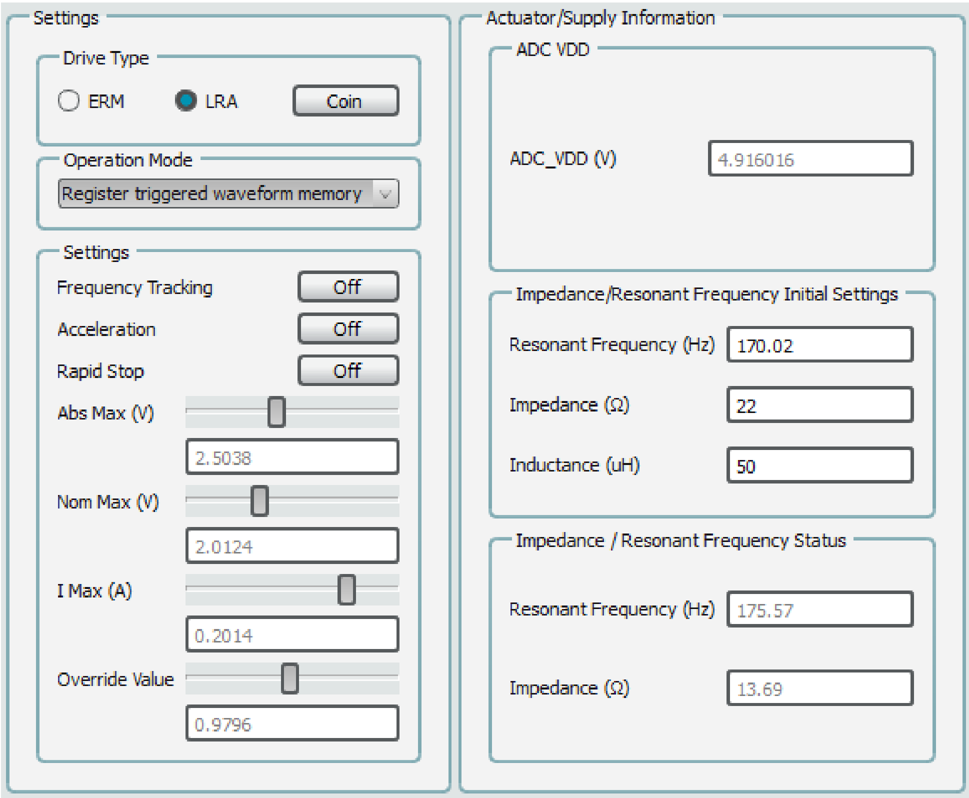

Figure 32. JAHWA_1040_FREQ_TRACK_OFF_SINE parameters

Figure 33. `JAHWA_1040_FREQ_TRACK_OFF_SINE` sequences

The haptic sequences loaded from the configuration scripts are displayed in the SmartCanvas DA728x GUI (see Section 9.4). For the `JAHWA_1040_FREQ_TRACK_OFF_SINE` script, the sequences are shown in Figure 33.

The Waveform Memory sequences can specify the drive frequency to allow wideband haptic effects, the last sequence "3 tone alert audio example" in the waveform memory demonstrates the use of this where the LRA is used to play simple tones. This can be useful to gain basic audio signaling capability in a system without using a speaker.

9.1.4.4 JAHWA_1040_FREQ_TRACK_ON - 2 x LRA IN PARALLEL

The `JAHWA_1040_FREQ_TRACK_ON(2xLRA-IN-PARALLEL).txt` script sets up the DA728x to enable the features shown in Table 3.

Table 6. `JAHWA_1040_FREQ_TRACK_ON` script features

Feature	Status
Frequency tracking	ON
Acceleration	OFF
Rapid stop	OFF

The measured inductance and impedance when the two LRAs are connected have been configured in the script file. The IMAX value has been adjusted such that there is enough current to supply both

LRAs at once. To achieve this, special double current mode has been set in the script so that the current supply can be doubled, this is reflected in the IMAX value.

The **Abs Max** and **Nom Max** voltages have been reduced slightly for the parallel LRAs configuration.

Ensure that this script is only used when two JAHWA LRAs are connected in parallel to the outputs as driving a single LRA will result in damage to the LRA.

NOTE
● After downloading the LRA configuration scripts, the impedance and inductance values are not updated, but the underlying registers in DA728x are set correctly for the LRA.

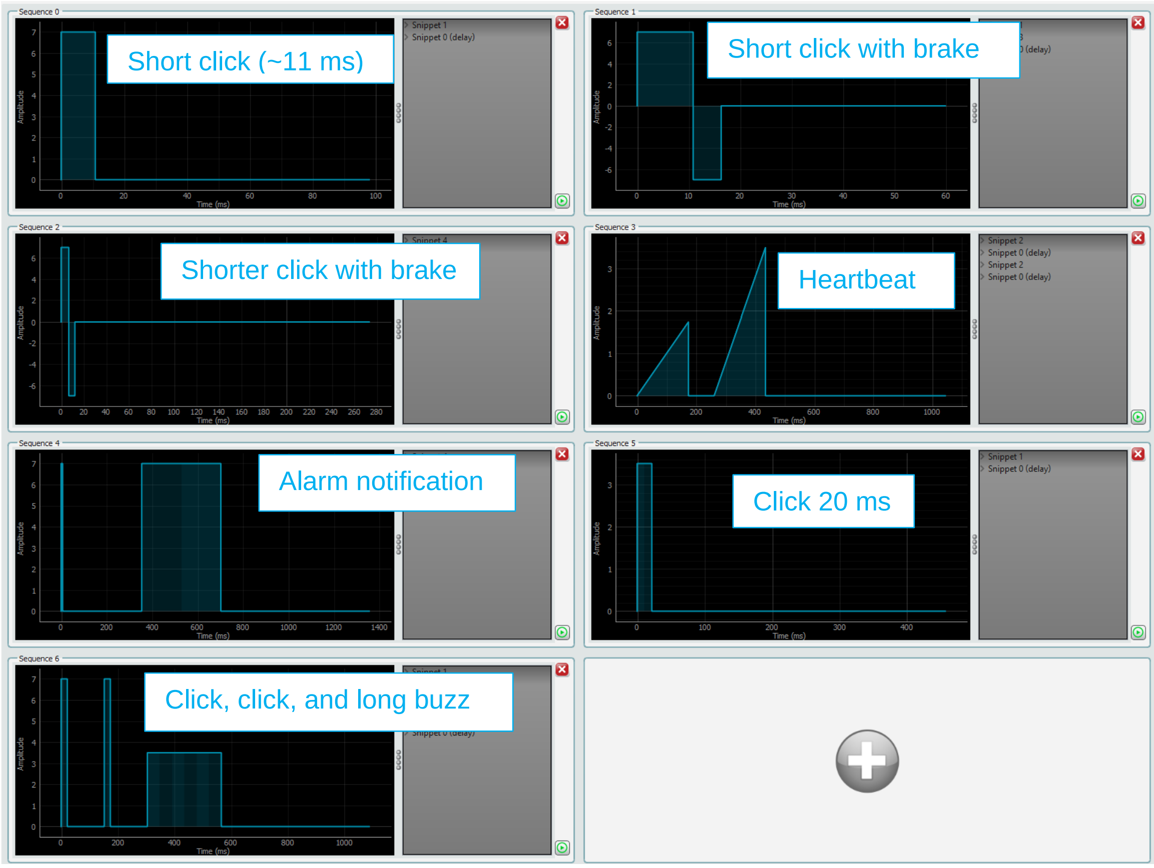

Figure 34. JAHWA_1040_FREQ_TRACK_ON(2xLRA-IN-PARALLEL) sequences

The haptic sequences loaded from the configuration scripts are displayed in the DA728x GUI (see Section 9.4). For the JAHWA_1040_FREQ_TRACK_ON2xLRA-IN-PARALLEL) script, the sequences are shown in Figure 29.

9.2 TITAN haptics actuators

Two additional actuator types have been supplied with newer evaluation kits to demonstrate the haptics technology from TITAN Haptics with their Tachammer DRAKE range of Linear Magnetic Ram (LMR) actuators. The DRAKE LF (low frequency) and HF (high frequency) actuators are included in the evaluation kit and can be easily connected to allow demonstration and evaluation of the products.

To evaluate and test the actuators:

1. Select which one of the two actuators (LF or HF) you would like to connect.
   The LF actuator is marked with red tape in the middle region and the HF actuator is marked with white tape in the middle region as shown in Figure 35. Each of the actuators has internal parts that move back and forth along the length of the actuator which results in movement and vibration on this axis.

Figure 35. TITAN haptics LF and HF actuators

2. Remove the white double-sided tape from the underside of the actuator as shown in Figure 36.

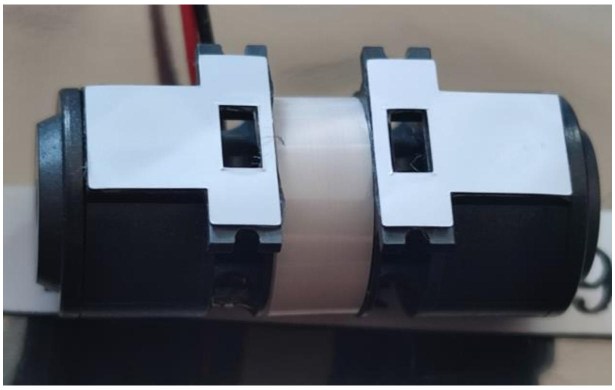

Figure 36. White tape shown on the underside of HF and LF actuators

3. Ensure the wires can reach the two left side pins of either J7, J8, J11 or J12 leaving enough length on the actuator cable to allow them to be comfortably connected and disconnected using the 1 pin sockets as shown in Figure 37.
   Note that the black wire should connect to the top left pin which is OUTN and the red wire should connect to the bottom left pin which is OUTP.
4. You can mount the actuator in one of the ways:
   - Mounting in the up/down position (Figure 37) results in the vibration being able to be measured on the X axis of the accelerometer.
   - Mounting in the left/right orientation results in the vibration being able to be measured on the Y-axis of the accelerometer.

NOTE
• Ensure that the actuator placement is in line with the board edge angle and pressed firmly down on the DA728x evaluation board to allow the actuator to stick.

NOTE
● Ensure that the black double jumper which is normally connected to the default actuator on J13 is removed and remove any other actuator which is connected to OUTN/OUTP as only one motor can be connected at a time to the outputs for the best performance.

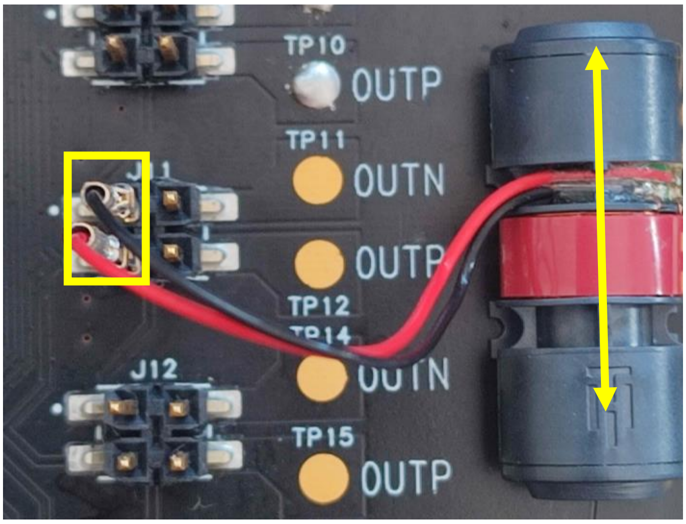

Figure 37. Connection to left hand pins of J11, orientation vibration for x-axis (arrow)

5. Several scripts have been included within the GUI/memory stick to allow the DRAKE actuators to be driven with various example waveforms. To load the script, click **Load file**, select the appropriate script for the motor selected (HF or LF), the scripts are listed as "titan-drake-XXXX", for example "titan-drake-LF-FRQ_TRK_ON_RS_ACC_OFF-EVB.txt".

NOTE
Scripts for the MF (medium frequency) and LFi (low frequency impact) are also included but these actuators are not included within the evaluation kit and need to be purchased separately. The MF and LFI scripts script will not work correctly with the LF and HF actuators and should not be loaded when using the HF and LF actuator variants.

6. After the script has been loaded, go to **Tools** > **Waveform memory editor**, then click **Read from device**. Various waveforms are encoded within the script and then decoded.

NOTE
Ensure that the same process is followed when loading a new script, particularly the step with clicking the **Read from device** button as this ensures that the waveforms displayed are the correct ones decoded from the script which was last loaded.

7. Click the various play buttons for the sequences to feel the vibration examples as shown in Figure 38, these include heartbeat, Cat Purr, Machine gun, Bounce, Danger Alarm, bouncing ball and other waveform sequence by scrolling down the sequence window.

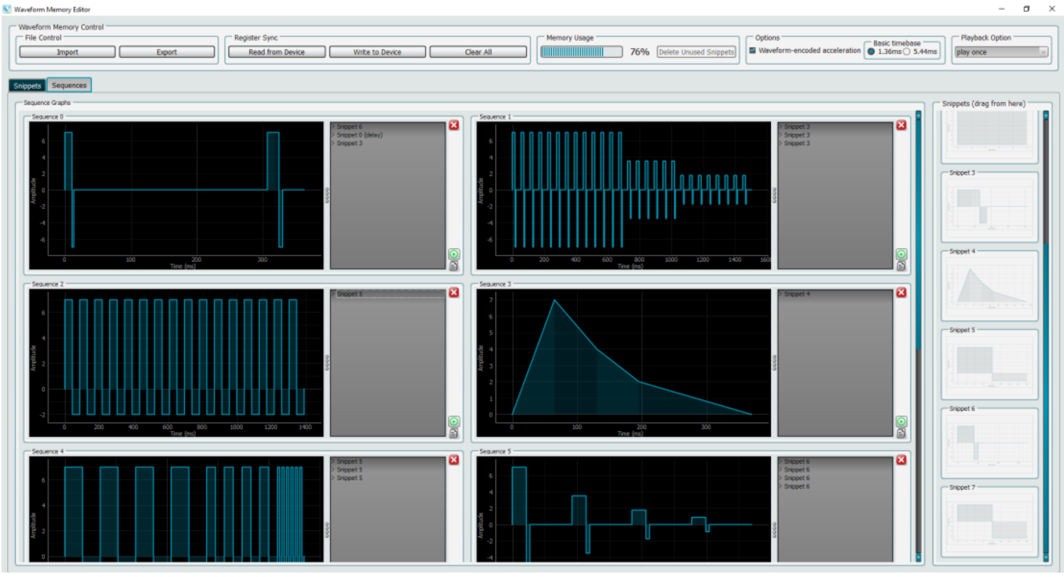

Figure 38. TITAN DRAKE LF example waveforms

9.3 Operation modes

You can select different modes for driving the LRA using the DA728x GUI (Figure 39) and each operation mode is described in detail in the following sub-sections:

- Direct register override, see Section 9.3.1.
- Register triggered waveform memory, see Section 9.3.2.
- Playback from PWM data source, see Section 9.3.3.
- Edge triggered waveform memory, see Section 9.3.4.

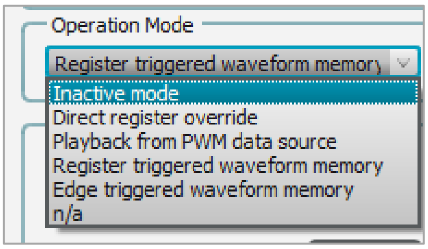

Figure 39. Operation mode

9.3.1 Direct register override

In the direct register override (DRO) mode, the haptic sequences are streamed to DA728x through the I2C input. The drive level of the output is set through **Override Value** in Figure 41.

To use the DRO mode:

1. In DA728x GUI, go to **DA728x** > **USER** > **Settings**, under **Drive Type**, select **LRA**.

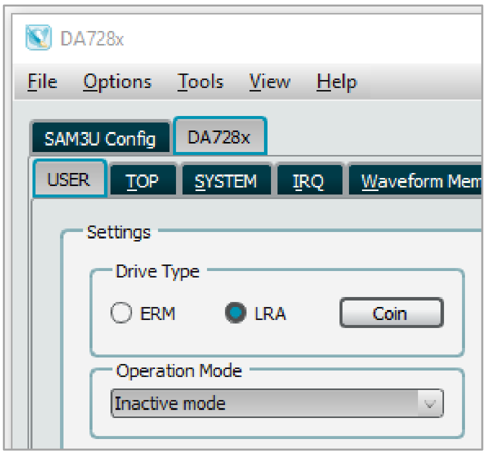

Figure 40. Drive type

2. At the lower left side, in the **Settings** section, set the value in **Override Value** (Figure 41).
3. In the **Operation Mode** list, select **Direct register override**.

First setting the override value and then setting the operation mode to DRO ensures the optimal latency of the LRA.

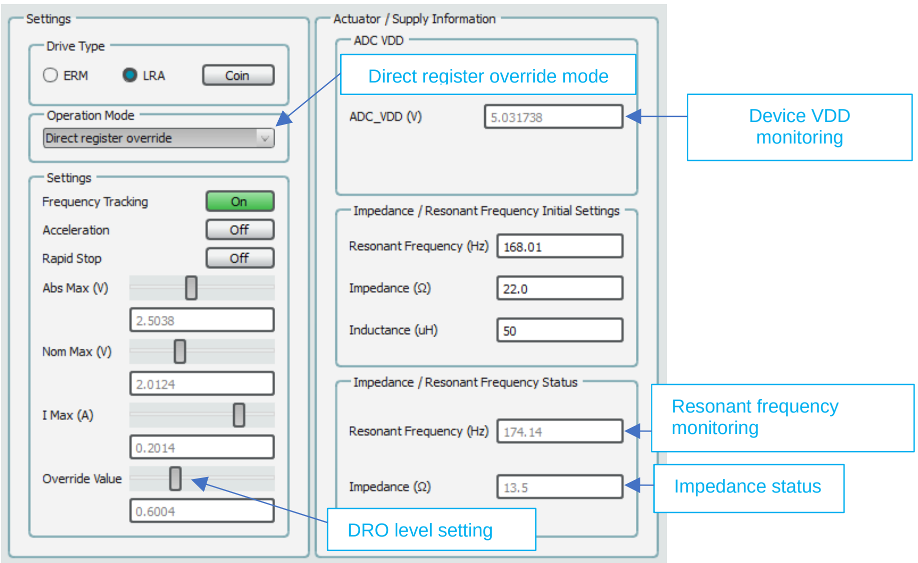

Figure 41. SmartCanvas DA728x GUI in DRO mode

9.3.2 Register triggered waveform memory

The register triggered waveform memory (RTWM) is a mode that allows you to trigger the haptic sequences stored in the Waveform Memory by using the I2C writes to DA728x.

1. In DA728x GUI, go to **DA728x** > **USER** > **Settings**, under **Drive Type**, select **LRA** (Figure 42).
2. In the **Operation Mode** list, select **Register triggered waveform memory**.

3. In the lower left side, in the **Sequence Playback** section, in the **Sequence ID** list, select a sequence. Programmed sequences are described in Section 9.4.
4. In the **Sequence Playback** section, in the **Sequence Loop** list, select the number of times that the sequence will repeat. Note that "No Looping" = 1 sequence; "1 loop" = the sequence repeated once after first playback, and so on.
5. To trigger the chosen sequence, click the **Sequence Start** button.

The haptic sequences in the waveform memory can be edited, created, and triggered from the Waveform memory editor (see Section 9.4).

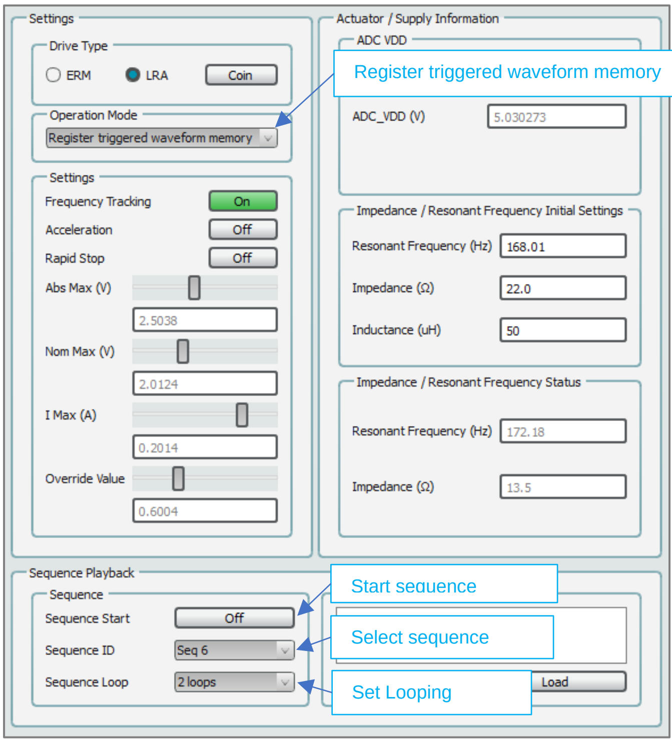

Figure 42. SmartCanvas DA728x GUI in RTWM mode

9.3.3 Playback from PWM data source

Pulse Width Modulation (PWM) mode is used to stream the haptic sequences to DA728x through the GPI_0/PWM input pin where the output drive level is determined by the duty cycle of the PWM signal. The PWM signal only contains envelope information and the input PWM frequency is unrelated to the output PWM frequency or the resonant frequency setting. The duty cycle is interpreted differently depending on whether acceleration mode is turned ON or OFF in DA728x registers (see the *Pulse Width Modulation Mode* and *General Data Format* sections in Ref. [1] for more details).

To drive the LRA in the **Playback from PWM Data Source** mode:

1. Set the input PWM frequency to the lower end of the allowable range, for example, 10 kHz to 250 kHz.
2. Actively drive the PWM signal through the GPI_0/PWM pin.
3. In the DA728x GUI, in the **Impedance/Resonant Frequency Initial Settings** section, set the resonant frequency and impedance of the LRA.
4. In the **Operation Mode** list, select **Playback from PWM Data Source**.

NOTE
• Before this mode is selected, the PWM signal must be actively driving the GPI_0/PWM pin, otherwise, DA728x will generate an IRQ which must be cleared to allow device operation.

NOTE
● Ensure that while the PWM signal is being driven, the signal does not reach 100% ON or 100% OFF as the device expects a constant PWM stream to always be present when in PWM mode. If the PWM signal stays high or low for longer than expected, the device counts this as a loss of signal or stuck high signal and triggers an IRQ which must be cleared to allow device operation. ● To stop the LRA output by using the PWM signal, a PWM signal of less than the brake threshold level (FULL_BRAKE_THR value in TOP_CFG2, the default value is 6.66%) must be driven. ● It is recommended that if the operational mode is to be set back to idle from PWM mode, the PWM signal should be reduced to lower than the brake threshold level first before exiting PWM mode. Otherwise, there is a risk that the last buffered PWM level will be seen at the output for one pulse when PWM mode is started again, even though the PWM signal may be below the brake threshold value. ● Stopping the PWM input signal before exiting PWM mode will also result in an IRQ fault which must be cleared manually before the output can be enabled again.

9.3.4 Edge triggered waveform memory

In edge triggered waveform memory (ETWM) mode, you can trigger playback of sequences stored in waveform memory by changing the transition voltage seen on the inputs of the GPI pins, the minimum pulse duration needed to trigger a GPI transition needs to be at least 25 µs.

The transition for each GPI can be set separately and can be configured as either rising, falling, or both rising and falling edges for GPI_0/PWM, GPI_1, and GPI_2 pins (see Section 9.4). The GPI can also be set to ignore a transition by changing the polarity value to 3 if there is a need to disable that input at any time.

Set the GPI registers (0x29, 0x2A, and 0x2B) by going to the **SYSTEM** tab in SmartCanvas DA728x GUI shown in Figure 43, see also Section 9.7. When the ETWM mode is selected in operational mode, it is also still possible to trigger sequence playbacks through I2C without changing the operational mode back to the RTWM mode.

If GPIs are not used in the design of a schematic, they should be grounded in the schematic; as if left floating, they could increase the leakage current if the pin voltage crosses the threshold of 0>1 or 1>0.

When the DA7281 daughterboard is used with the DA728x motherboard, only one GPI is available on this device (GPI 0) as the other GPIs are used to determine the I2C address.GPI 1 and 2 are still shown in the GUI when using the DA7281 but cannot be configured.

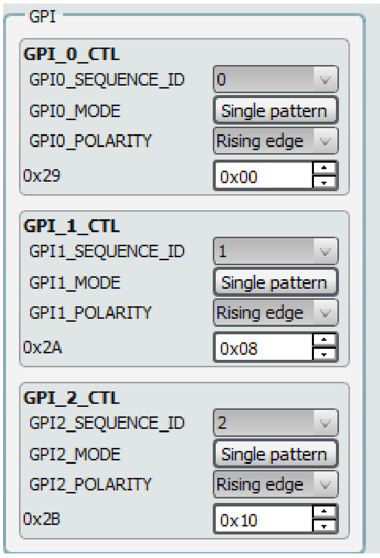

Figure 43. GPI registers

9.4 Waveform memory editor

The waveform memory editor is a powerful and intuitive tool for creating haptic effects stored in memory. You can modify, view, and play the haptic effects in waveform memory by selecting **Tools** > **Waveform Memory Editor** in the DA728x GUI (Figure 44).

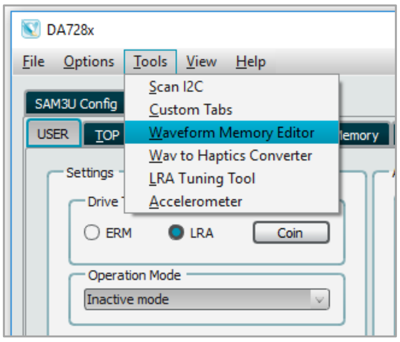

Figure 44. Starting the waveform memory editor in DA728x GUI

To load any waveform memory data in the device or proceeding the loading of a script, in the **Waveform Memory Editor**, under **Register Sync**, select **Read from Device** (Figure 45) to read back the waveform memory from DA728x to ensure that the display is synchronized with the device memory.

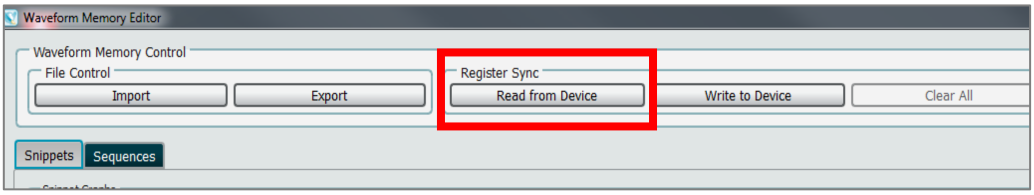

Figure 45. Register syncing from DA728x memory

9.4.1 Snippets and sequences

Snippets and sequences form the basis of creating complex waveforms, clicks, and sequences, which use very little memory to create long and complex haptic effects. This is due to the waveforms being described as shapes instead of storing actual amplitude samples.

The waveform memory in the DA728x consists of only 100 bytes which means it can be easily updated very quickly over I2C if more sequences are needed. The waveform memory is used to store snippets that describe simple waveforms and describe sequences of snippets which form the actual waveform that is played back.

You can create up to 15 snippets and 15 sequences within the 100 bytes of memory depending on the complexity of the snippets and sequences.

You can create, delete, or modify snippets using the **Snippets** tab in the Waveform Memory Editor, see Section 9.4.2. The created snippets appear on the **Sequences** tab as waveforms and you can drag snippets into the window to create more complex sequences of snippets that are played one after the other, see Section 9.4.3.

NOTE
Snippets cannot be played directly, because the haptic effects can only be triggered when the snippet is added to the sequence memory. See Ref. [1] for more details.

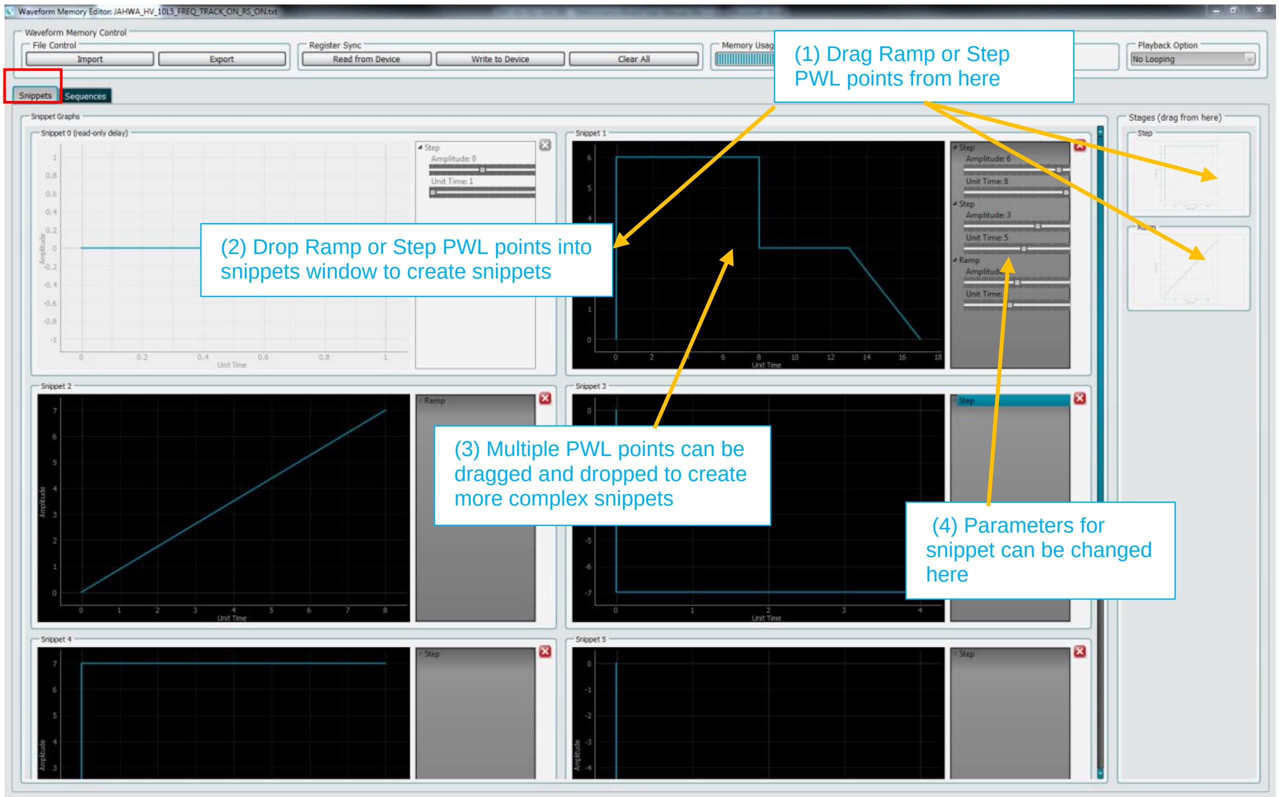

Figure 46. Waveform memory editor Snippets tab

You can play the created haptic sequences by clicking the green play buttons (see Figure 47). You can also loop the played sequence by using the **Playback Option** in the upper-right corner.

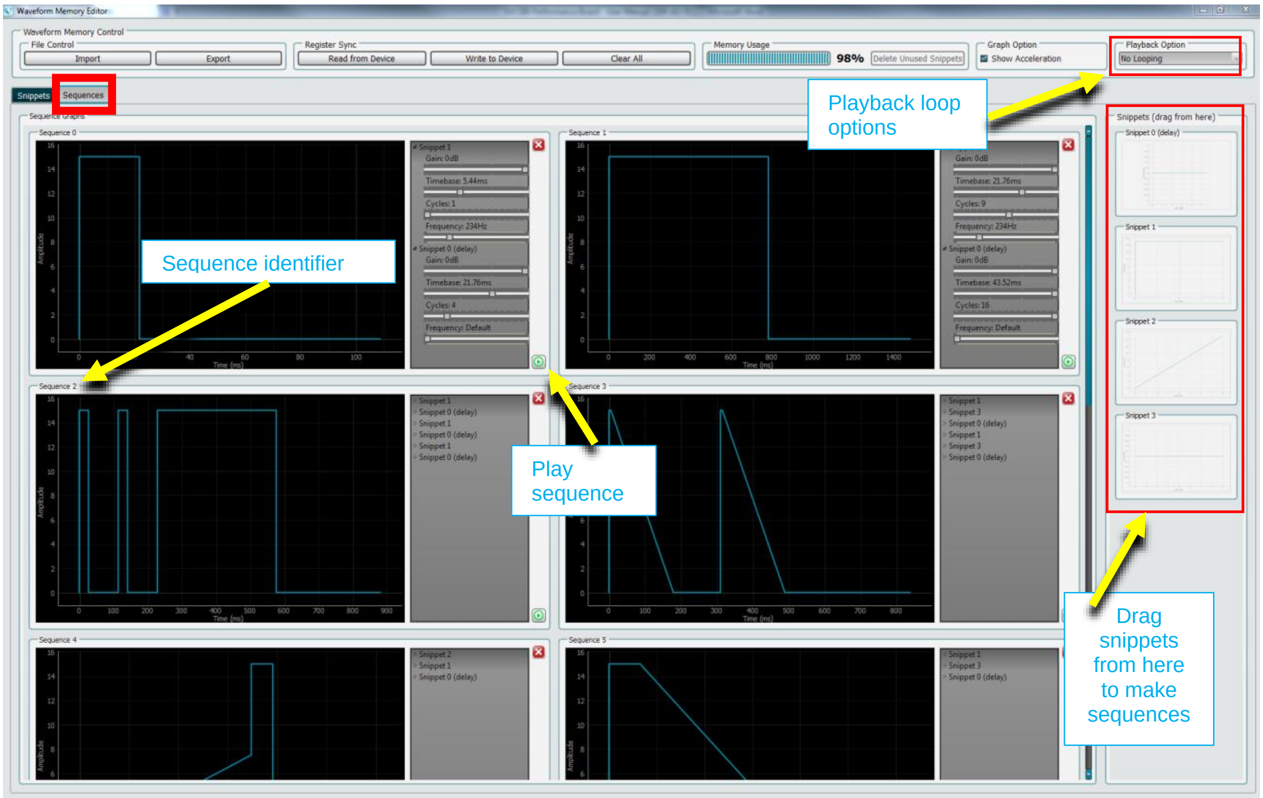

Figure 47. Waveform memory editor Sequences tab

9.4.2 Create snippets

To create and construct a snippet, drag the basic **Step** or **Ramp** piecewise-linear (PWL) snippets from the **Stages** window that is on the right side of the **Snippets** window as shown in Figure 48.

Steps are described as set levels of amplitude over several units of time.

Ramps are described as a change from an existing amplitude level of the previous signal to a new amplitude level over several unit times.

You can adjust the time and amplitude parameters of each snippet using the sliders in the GUI.

Figure 48. Snipper creation

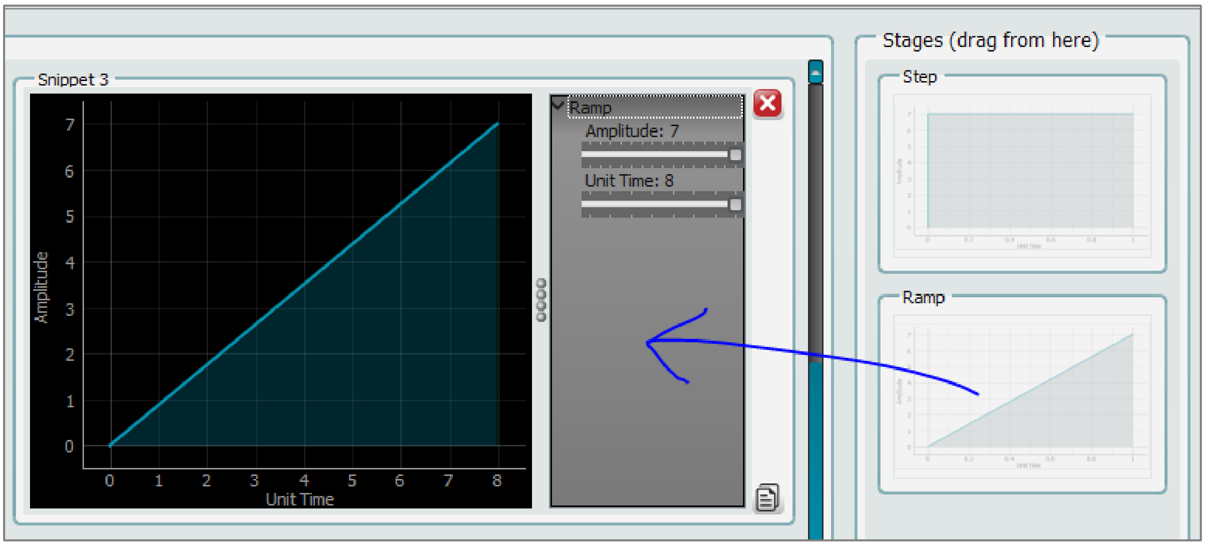

Figure 49. Second snippet waveform addition

The amplitude is between +/-7, where +7 is the full forward amplitude level, 0 is stationary (LRA not driving) and -7 is full reverse amplitude (output phase of AC signal changes), but only when acceleration is turned off. Driving in reverse can be used to create certain vibration effects or to stop the LRA quicker than the natural stopping time if the driver output is simply turned off.

These steps and ramps can be connected one after the other in any number or order (depending on memory left) to describe a waveform which is stored in the device and known as a snippet. For example, in Figure 50, one snippet at level 7 is played for one unit of time, then a snippet at level 3 is played for one unit of time, followed by a ramp going down to level 0 over eight unit of time.

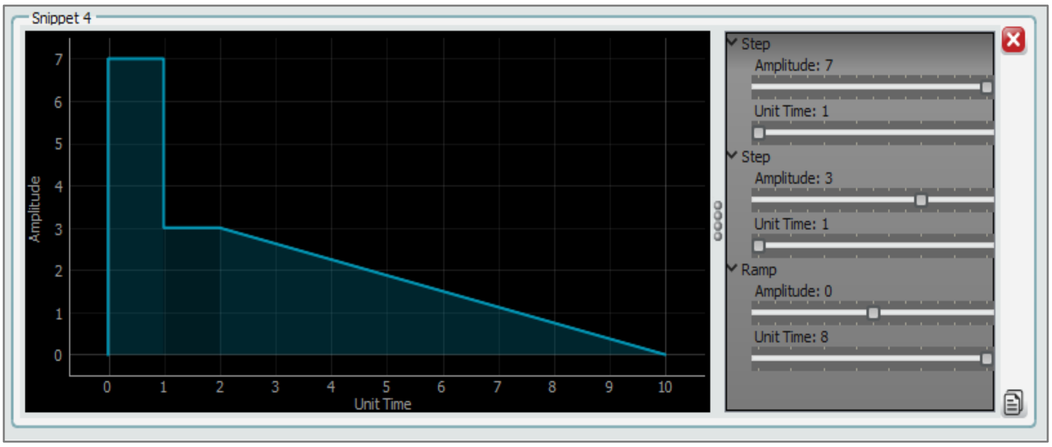

Figure 50. Complex snippet creation

9.4.3 Create sequences from snippets

After the snippets are created using the **Snippets** tab, you can then bundle them into sequences by using the **Sequences** tab in the waveform editor. Sequences describe a list of snippets played to define a more complex waveform that can be played back and manipulated in different ways.

Snippets created in the snippet's editor appear on the right side of the sequence editor. They can then be dragged into the sequences as a list of snippets to create more complex waveforms. An example of this shows the previous snippet being dragged into a sequence, along with a second snippet that is dragged afterward to form a complex sequence shown in Figure 51.

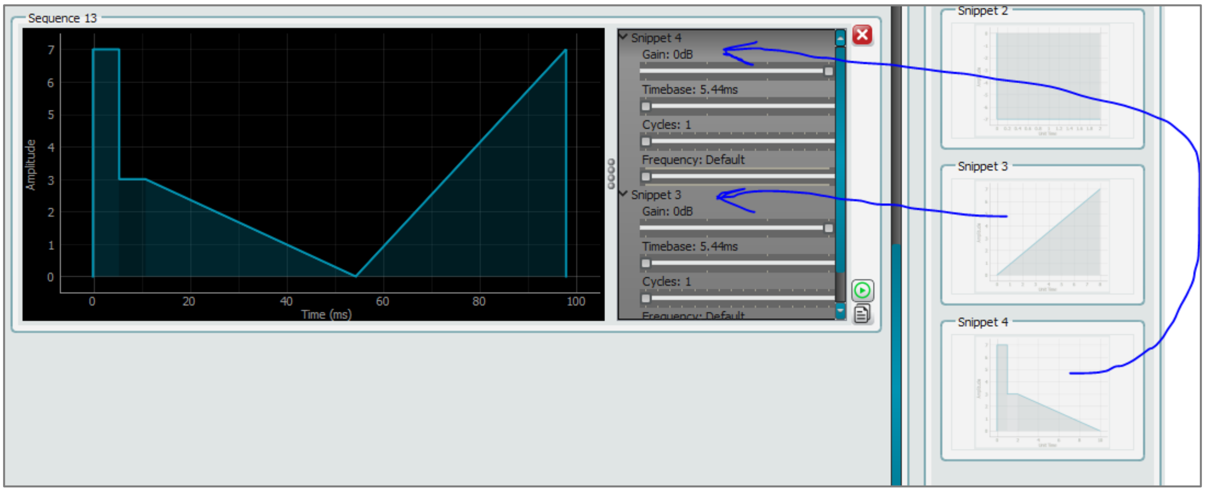

Figure 51. Sequence creation

For example, in Figure 52, a sequence for **Short-Click** is made up of **Snippet 1** and **Snippet 0**. The snippets are shown in detail in Figure 53.

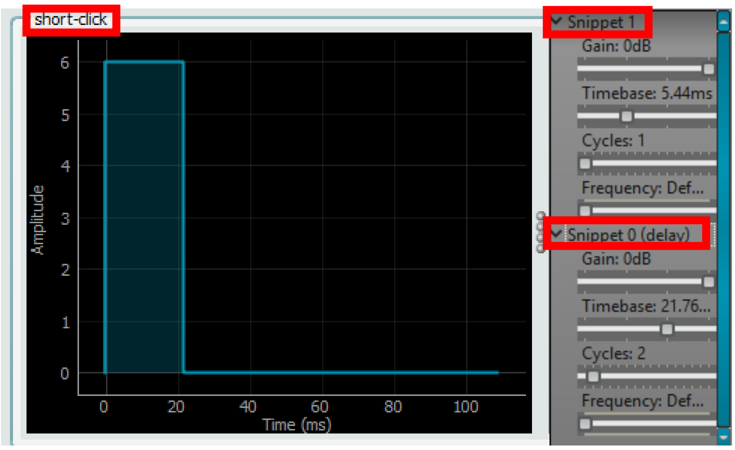

Figure 52. Two snippet sequence example

- Snippet 1 has an amplitude of 7 and a unit time of 4. The overall time is:

$$4\ (\text{unit time}) \times\ 5.44\ \text{ms}\ \ (\text{timebase in sequence definition}) \times\ 1\ (\text{cycles})\ =\ 21.76\ \text{ms}$$

- Snippet 0 has an amplitude of 0 and a unit time of 2. This is silence for an overall time of:

$$2\ (\text{unit time}) \times\ 21.76\ \text{ms}(\text{timebase in sequence definition}) \times\ 2\ (\text{cycles})\ =\ 87.04\ \text{ms}$$

More details on the fine details of time base are covered in Section 9.4.5 but the basic overall time of each sequence = Unit Time × Cycles × Timebase.

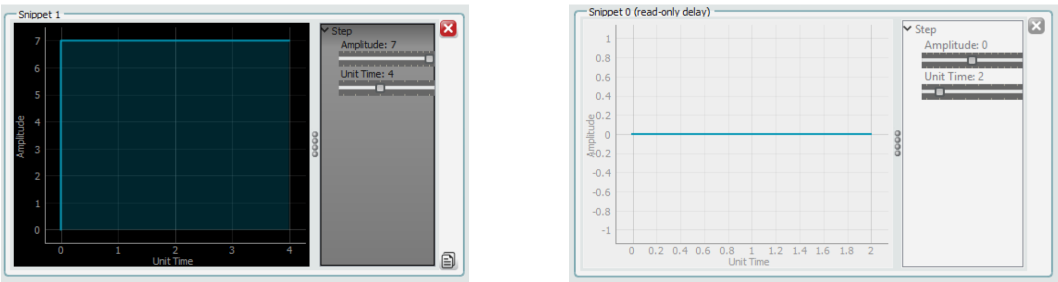

Figure 53. Snippets for two-snippet sequence example, 1 and 0

When snippets are dragged into the sequences window, you can also control four snippet parameters by expanding each snippet using the down arrow on the left side of the snippet name:

- The gain parameter allows the snippet gain to be adjusted between 0 dB and -18 dB in -6 dB steps.
- The frequency parameter allows the waveform to be played at a certain resonant frequency other than the "default" which is the current resonant frequency. If switching away from the default, the frequency can be anywhere between 25 Hz and 1 kHz. This is ideal when used in wideband mode (Sine wave), with frequency tracking off, to play tones. However, care must be taken when frequency tracking is turned on, as this could result in an IRQ being generated when outside the resonant frequency range of the LRA as the device will always try to track within +/-25% of the set frequency and measurement from the LRA.
    - Setting the frequency when tracking is on should not normally be done; but with very short "click" type waveforms, sometimes it is useful to set the resonant frequency of the first snippet to be close to the resonant frequency of the LRA. The last snippet or delay should also have the frequency set in this case and should be set to the resonant frequency ±2 Hz.
    - The reason for this is that there are sometimes not enough drive cycles for the frequency tracking algorithm to measure the data from driving on such a short waveform, due to the BEMF taking a few drive cycles to build up. The result of this is that when a short sequence is

played multiple times, there is sometimes a drift in resonant frequency tracking seen on some LRAs, which can eventually cause an IRQ.

- This can be avoided by setting the initial start frequency for very short snippets so that the tracking algorithm has a good start location for tracking every time the sequence is played.

  The reason that the last snippet or delay in the sequence is set ±2 Hz from the original is that if the value is set to the same as the previous value, then the write is ignored. The device assumes the resonant frequency has not changed if it was set to the same previous value (if the start and end frequencies were set to the same value). Instead, it must be set to a different value in case it has drifted when measured.

- Time base**,** the unit time of a snippet depends on the FREQ_WAVEFORM_TIMEBASE bit of the register SEQ_CTL1. It can be of (5.44, 21.76, 43.52, 87.04) ms or (1.36, 5.44, 21.76, 43.52) ms. For more details, see Section 9.4.5.
- Cycles allow the snippet to be played between 1 and 16 times based on the current time base selection. For more details, see Section 9.4.5.

NOTE
Care should be taken when changing parameters such as frequency, cycles, and so on. These all take up additional bytes from the waveform memory, so will result in less sequences or snippets being able to be created due to the waveform memory taken up.

9.4.4 Edit sequences

To change the sequence name, double-click the sequence identifier.

To change the order of the snippets played in the sequence editor and snippets editor, right-click the snippet in the sequence list and select **Move Up** or **Move Down**. To remove snippets, right-click the snippet in the sequence list and select **Remove**.

The order of the two snippets is changed as shown in Figure 54. The resulting sequence is shown on the right after the change. New snippets or delays can be dragged to the sequence window, then moved up or down or deleted to arrange the sequence in any order. Each of the snippets can also be adjusted within the sequence.

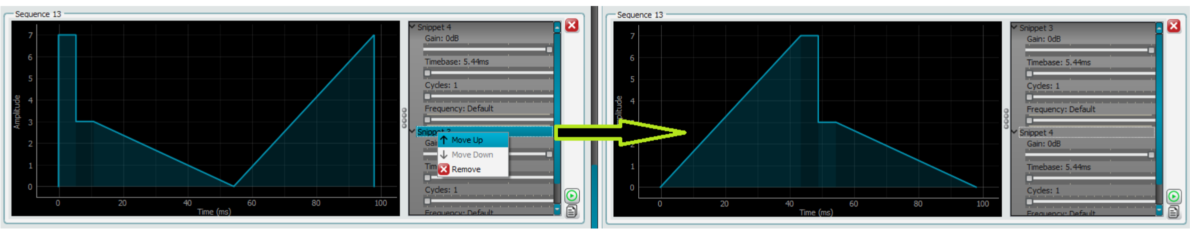

Figure 54. Modifying snippet order in a sequence

9.4.5 Detailed information on time-base and unit time

As mentioned, snippets are created and edited based on "unit times" which are basically slots of allocated time to describe the waveform. These unit time slots are based on the value in register 0x24Time base FREQ_WAVEFORM_TIMEBASE (bit 2) which sets one of two-time base values.

- Setting to 0 results in 5.44 ms based time-base units and setting to 1 results in 1.36 ms based Time base units. This means that if a snippet which has been created to have 4 * Time base units and if the Time base register is set to 0 (5.44 ms), then, when played back, the waveform is approximately 4 * 5.44 ms =21.76 ms. If the time-base register is set to 1 (1.36 ms), then, when played back, the waveform is approximately 4 * 1.36 ms = 5.44 ms.
- The Time base slider, seen within the sequence editor for each individual snippet, can be increased too. When the Time base register is set to 1.36 ms based values, the slider allows the selected snippets Time base to be increased at set steps of 1.36, 5.44, 21.76, or 43.52 ms.
- When the Time base register is set to 5.44 ms based values, the slider allows the selected snippets Time base to be increased at set steps of 5.44, 21.76, 43.52, or 87.04 ms.

For example, if a snippet is 4-unit times, the time base is set to 1.36 ms time base setting and within the sequence the slider is increased to 21.76 ms. This results in the waveform time played for approximately 4 * 21.76 ms = 87.04 ms.

Setting the cycles slider within each snippet in the sequence editor allows the selected snippet time base to be cycled several times to customise the time for playback.

For example, if the snippet last mentioned (4 units * 21.76 ms = 87.04 ms) is played with the cycle set to 1, then the waveform is approximately (1 * (4 * 21.76)) = 87.04 ms. If the cycle slider is then set to 4, then the waveform is played four times, resulting in the playback being approximately 4 * 87.04 = 261.12 ms.

Setting the time base unit to 1.36 ms or 5.44 ms is purely down to preference, but it is better to set to 1.36 ms, where more fine control is needed for short clicks, or when the resonant frequency of the LRA is quite high and therefore you need more control on the number of half periods generated. If not generating clicks and when longer waveforms and less accurate sequences are needed, then the time base setting of 5.44 ms is more efficient, as longer sequences can be created with less bytes used of the waveform memory.

Care should be taken to create waveforms in an efficient manner, as there are several ways to create the same time settings, but some will result in more waveform memory being used than others.

For example, in creating a 21.76 ms delay within a sequence by dragging the delay snippet to the sequence when the Time base setting is set to 5.44 ms can be done in two ways.

- The first method would be to change the cycles to 4 (4 * 5.44 = 21.76 ms).
- A second method would be to change the Time base setting of that snippet to 21.76 ms, by using the second method this results in one byte less being used in waveform memory.

9.4.6 Real playback time V approximate playback time

"Approximately" has been used several times to describe the playback time. The reason for this is that the device must interpret the waveform sequence in real time during playback and try to fit this playback slot to take account of the current resonant frequency setting in terms of the amount of half periods (pulses at half resonant frequency). This can result in some unexpected behavior when creating waveforms and looking at the resultant sequence on an oscilloscope. This behavior is expected and is purely the result of the quantisation of the time base versus the current resonant frequency.

For example, the resonant frequency is 170 Hz, this is 5.88 ms for one full frequency cycle (one pulse on OUTP, then one pulse on OUTN), so the half period is 2.94 ms (half waveform cycle).
If we request playback of a snippet which has 1 unit of time at a time base setting of 1.36 ms, this is 1.36 ms. When playing this sequence at a resonant frequency of 170 Hz, nothing is seen at the output, as 1.36 ms is under the threshold needed to trigger playback of one half of a half period which is the minimum time that can be played. In this case the threshold is 2.94 ms (half of 1/170), which is half of a half period at the current resonant frequency.

If we add another time unit to the snippet (2 * 1.36 ms = 2.72 ms), then play the waveform again, this is slightly shorter than a half period and still nothing is seen at the output. Adding a third time unit (3*1.36 ms = 4.08 ms) and click play on the snippet, then one-half period can be seen at the output.

The difference between the real playback length (2.94 ms) and the time allocated for the snippet (2.72 ms) is 2.94-2.72 = 0.22 ms which is quantized out as it is not possible to play a half period in this time.

When playing several snippets in a sequence, these time differences between one snippet and another can accumulate and cause another half period in the sequence to either be quantized out or appear as an extra drive pulse. This can be a little confusing at first, when creating short clicks and sequences, and can sometimes lead to undesired effects, but this is less of a problem for longer sequences. Care must be taken when creating short clicks and sequences because of this quantisation effect and accumulated differences between desired timeslot and resonant frequency.

The device is tracking the resonant frequency of the LRA in the actual system in real time when frequency tracking is turned on. This can mean that there is part-to-part variation and load dependency on how the product is physically held, which means the resonant frequency can change depending on these factors. As the output timing is always being quantized for the time available that is dependent on the actual resonant frequency half period, this also means that the output can change slightly as the device is always tracking the resonant frequency in real time. These timing thresholds can result in pulses appearing, or not depending on the current resonant frequency being measured and how many pulses can be fitted into the allocated time.

If, for example, the resonant frequency measured becomes higher (less load of the motor or LRA variation), then the time period is shorter for a half period so more pulses at the output could be seen. On the other hand, if the resonant frequency shifts to a lower value (more load on the motor or LRA variation), then fewer half periods can be fitted into the time slot, so you may see less pulses at the output.

9.5 Wav to haptics converter

The **Wav to Haptics converter** tool allows `.wav` files to be converted to haptic patterns and the audio and haptic effect can be played back simultaneously. This should only be used to create haptics effects to go along with music and long samples, the tool cannot be used to create short haptics effects such as clicks.

The tool looks at the envelope (volume ramp) of the sound file, these levels are quantized into regular larger time frames to determine the average level of "buzz" for each of the steps in relation to the time of the sample. The frequency of the "buzz" is not changed so it is not possible to output the actual audio frequencies to the actuator, instead the resonant frequency setting is used to set the current drive frequency, or a set frequency can be used if tracking is not turned on.

To use the tool, in the SmartCanvas DA728x GUI (Figure 55), select **Tools** > **Wav to Haptics Converter**, and then click the **Load File** button (Figure 56).

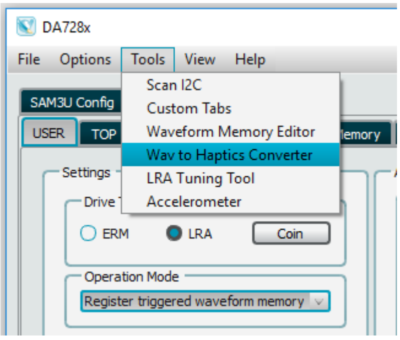

Figure 55. Launching Wav to haptics converter

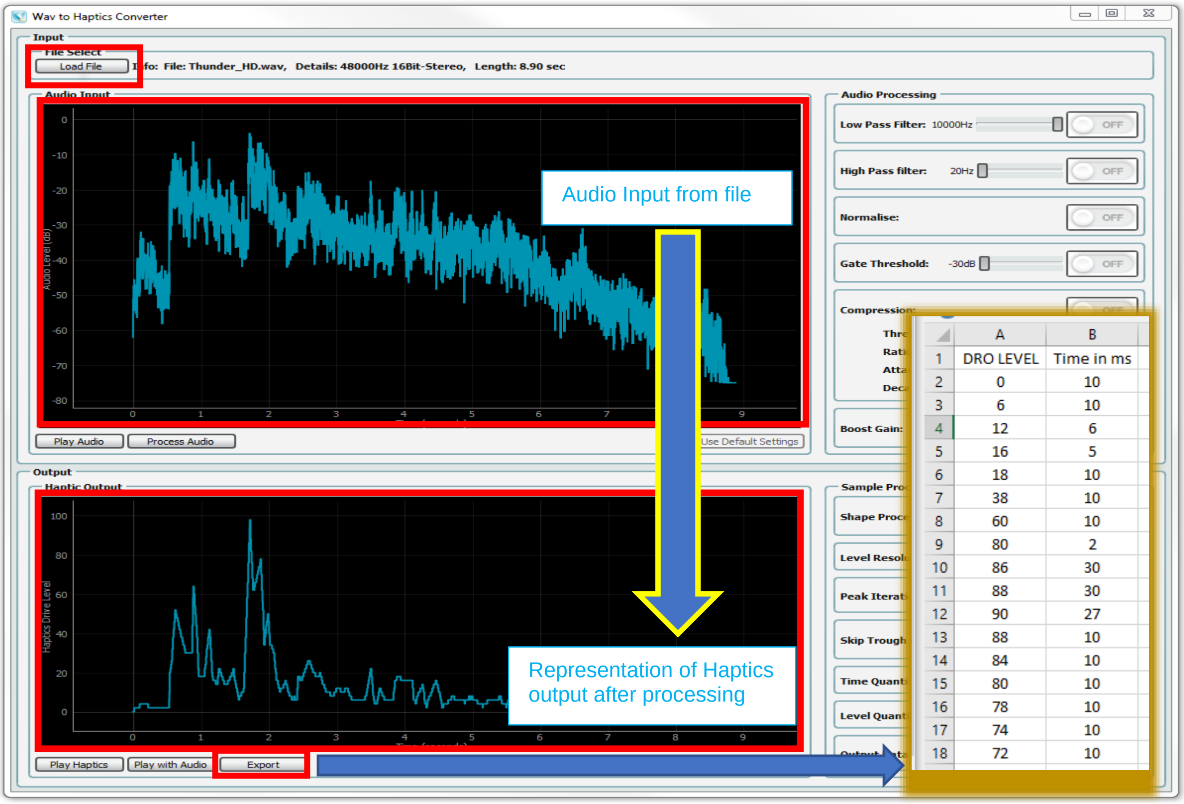

Figure 56. Wav to haptics converter tool

You can use the following options in **Wav to Haptics Converter**:

- **Load File**: navigates you to the `test_audio` folder bundled with the GUI. A number `.wav` files of sound effects are included for demonstration purposes (Figure 57).

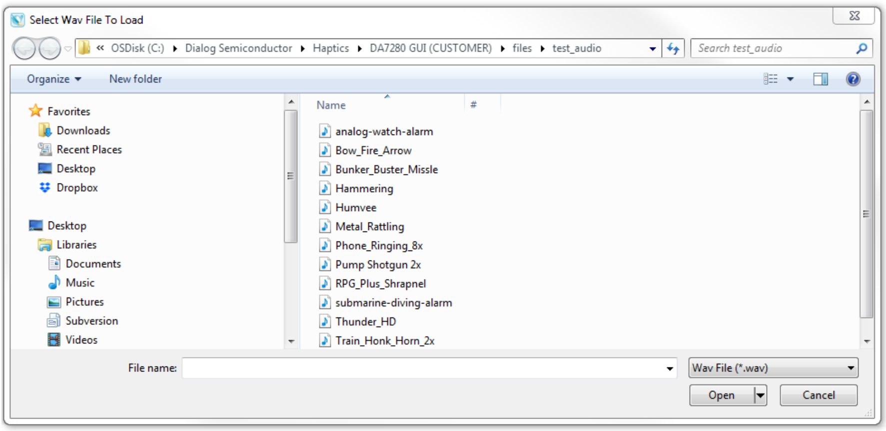

Figure 57. Included sound effect files

- **Play Audio**: plays the audio track through the computer's soundcard.
- **Process Audio**: the audio is processed with the tool basic filters, including **low-pass filter**, **high-pass filter**, **Normalise**, **Gate Threshold**, **Compression**, and **Boost Gain** in the **Audio Processing** block. These filters create an envelope of the audio file that is then used to determine the level and length of vibration to represent the audio.
- **Play Haptics**: plays the haptic sequence when the audio has been processed.
- **Play with Audio**: plays the haptic sequence and audio simultaneously.

- **Export**: exports the haptic sequence as `.txt`, `.csv`, or `.bin` file. The produced output data is a list of "buzz" levels at the resonant frequency along with the durations in ms of these vibrations. This data can be used and processed within a customer's system to drive the DA7280 in DRO mode, sending the correct vibration level to the timing calculated from the saved data.

NOTE
Pointing to the sliders shows detailed descriptions of each control (Figure 58).

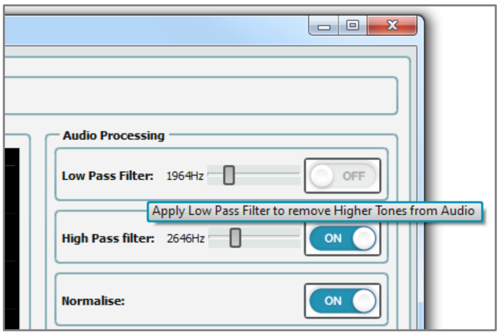

Figure 58. Detailed descriptions of controls

9.6 IRQ and faults

Interrupt requests can be viewed on the **IRQ** tab of the DA728x GUI (Figure 59). Interrupt requests can be events, warnings, and faults occurring in normal operations.

Warnings allow the output to still be driven but certain faults stop the operation of the device driving. To allow the LRA to continue to be driven again, these faults and IRQs must be cleared from the DA728x GUI which clears the values stored in the corresponding IRQ registers and allows the outputs to be restarted again.

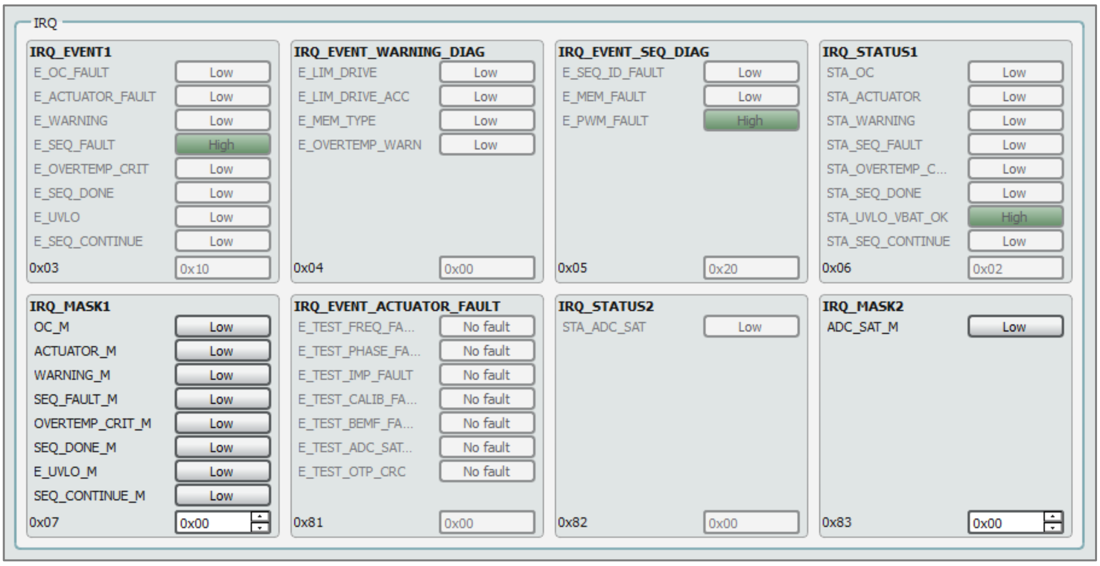

Figure 59. IRQ tab

To clear IRQ events, warnings, and faults, click the **Clear Events** and **Clear Faults** buttons in the SmartCanvas DA728x GUI (see Figure 60). Any faults or warning are shown as a "High" in value for the corresponding registers, clicking the "high" value writes a 1 into the corresponding bit to clear the register bits manually.

The **IRQ** tab also contains IRQ status registers that show real time status of some events and mask registers that allow some IRQ events to be masked from causing the IRQ pin to transition low in the event of an IRQ fault.

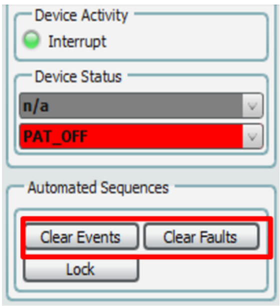

Figure 60. Clear IRQ events and faults

NOTE
In the **EMBEDDED_MODE** operation accessed through the **TOP** tab in the DA728x GUI (Figure 61), IRQs automatically clear themselves when a playback ends. This is useful for systems with minimum interaction between the host and DA728x but should not be used in cases where the part is not configured correctly for the LRA being used.

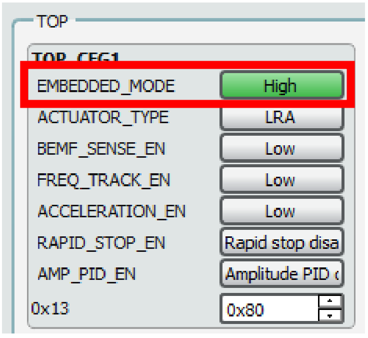

Figure 61. Embedded mode

9.7 CapSense setup

You can configure the USB to I2C controller to ensure the USB does not interfere with CapSense:

1. In the SmartCanvas DA728x GUI (see Figure 62), on the **Sam3U Config** tab, set the **IO Mode** in each port to **Input: High-Z**. It is also possible to use the SmartCanvas DA728x GUI to set the state of the GPIs if edge triggering control is required from the SAM3U microcontroller.

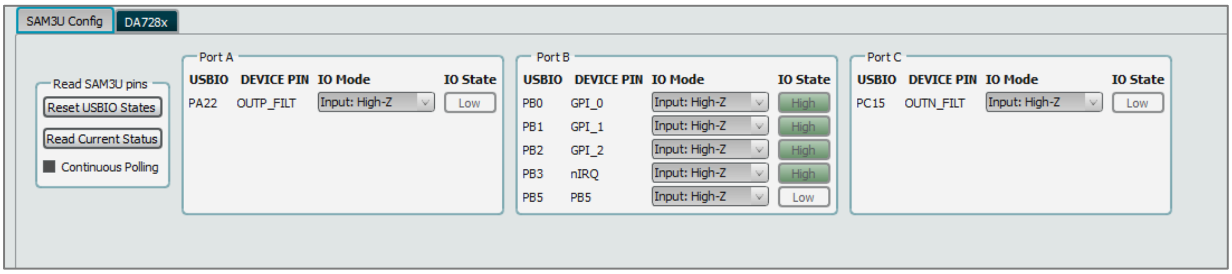

Figure 62. SAM3U configuration

2. Set the operation mode to ETWM mode (see Figure 63).

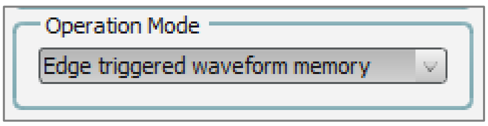

Figure 63. ETWM mode

3. Set the GPI controls as required for user defined operation. In this case **GPI0_POLARITY** is set to **Falling edge** triggered, and **GPI0_SEQUENCE_ID** is set to **0** (see Figure 64). On receiving a falling edge on **GPI_0** pin the first haptic sequence in the Waveform Memory will be triggered.

NOTE
The voltage levels of the CapSense buttons are high by default; clicking a CapSense button pulls the line to ground, triggering GPI 0 on the falling edge. Triggering on the rising edge means that the sequence will be triggered when the CapSense button is released. A third option is to trigger on either a rising or a falling edge. If the **Multi sequence** mode is chosen in the list for GPI_0_CTL, the rising edge triggers the sequence denoted by SEQUENCE_ID, while the falling edge triggers the sequence located at SEQUENCE_ID + 1.

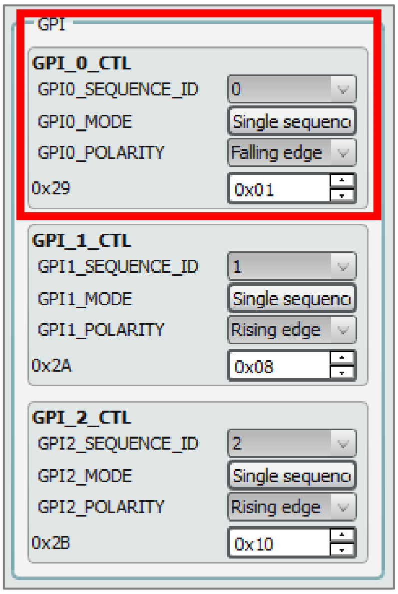

Figure 64. GPI_0_CTL settings

9.7.1 GPI triggering from GUI

You can control the GPIs from the **SAM3U Config** tab. You can change each GPI by setting the **IO Mode** to **Output: PushPull**, then change the IO state by clicking the **IO State** button (see Figure 65).

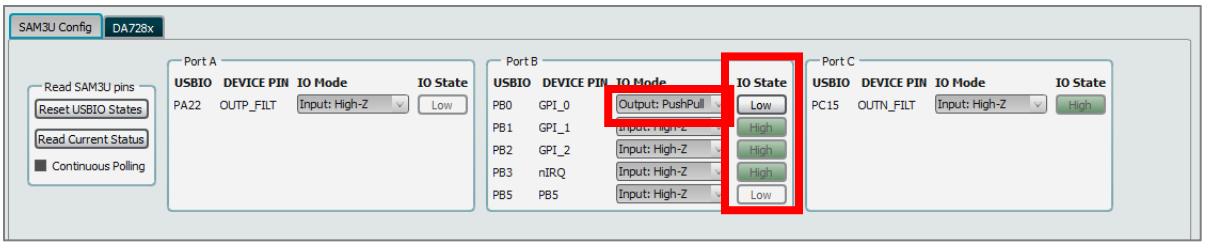

Figure 65. Controlling GPIs from GUI

10 Monitor Drive Signals and Measuring Acceleration

10.1 Analogue accelerometer

To use the analogue accelerometer to evaluate the system performance of the haptic driver and the LRA and to do general tuning, go through the following procedure:

1. Place the DA728x motherboard on a foam as shown in Figure 12 to isolate it from hard surfaces. This reduces damping of the LRA's motion.
2. Connect the oscilloscope channels 1 and 2 (yellow and green trace) to header J2 to monitor the filtered OUTP_F and OUTN_F signals, ensuring that channel coupling is set to DC. The filter cut-off is set at approximately 3.3 kHz.
3. The unfiltered OUTP and OUTN signals accessed through header J3 are PWM signals at 187.5 kHz. Checking these makes little sense for analysis purposes.
4. Connect oscilloscope channels 3 (blue trace) to header J5, making sure that the oscilloscope channel is set to AC coupling to remove gravity induced offsets. Choose a suitable accelerometer axis which is dependent on the type of actuator. The supplied actuator with the 359-05-X DA728x motherboard is a Z-AXIS LRA so the Z-AXIS pin should be used for analysis.
5. Set the oscilloscope to be triggered on the rising edge of channel 1. Now drive haptic patterns and observe the acceleration. You can also feel the movement of the LRA with your finger but be aware that this can change the visible effect of loading the LRA with a different mass.
6. The acceleration measured will be around 360 mV/G, as the supply voltage to the accelerometer is set to 3.6 V. Each of the outputs are low pass filtered to a bandwidth of 50 Hz. For more information on the accelerometer, please refer to https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL335.pdf.

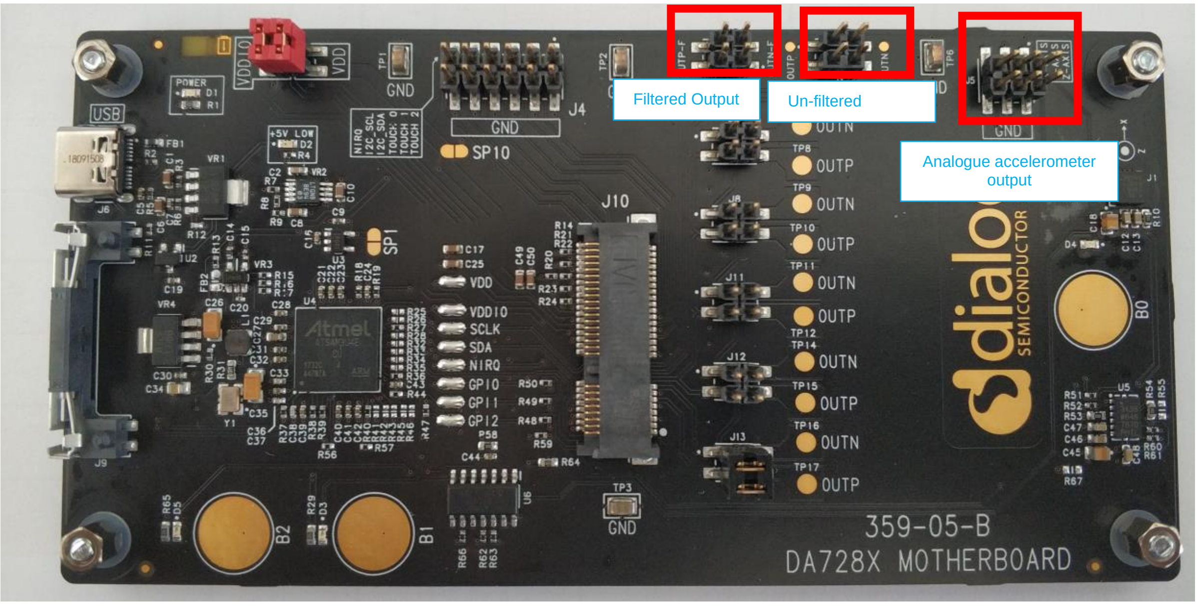

Figure 66. Monitoring signals

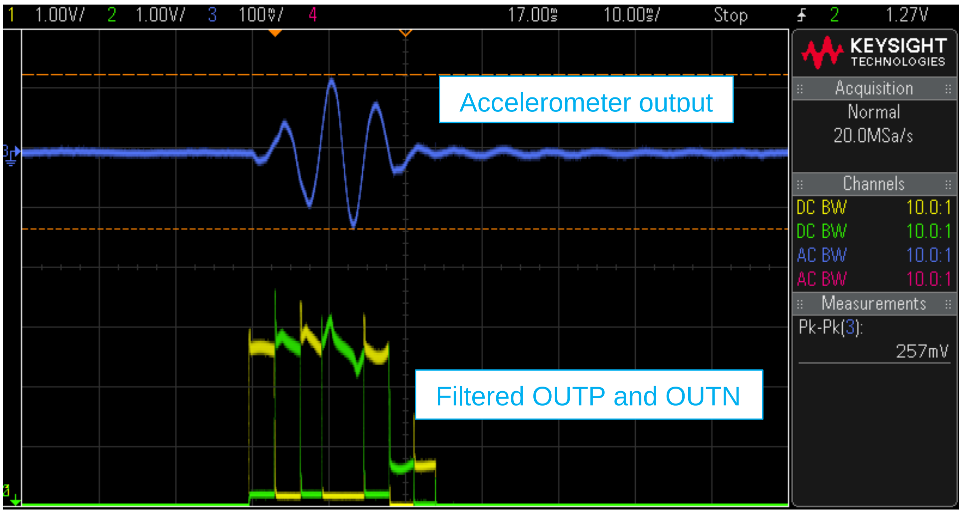

Figure 67. Measured output signals and acceleration profile

10.2 Digital accelerometer

The onboard digital accelerometer can be used in the same way as the analogue accelerometer to evaluate the system performance of the haptic driver and the LRA and for general tuning in the following procedure:

1. Place the DA728x motherboard on a foam as shown in Figure 12 to isolate it from hard surfaces. This reduces damping of the LRA's motion.
2. Connect the oscilloscope channels 1 and 2 (yellow and green trace) to header J2 to monitor the filtered OUTP_F and OUTN_F signals. The filter cut-off is set at approximately 3.3 kHz.
3. The unfiltered OUTP and OUTN signals accessed through header J3 are PWM signals at 187.5 kHz. Checking these makes little sense for analysis purposes.
4. In the DA728x GUI, select **Tools** > **Accelerometer** as shown in Figure 68.

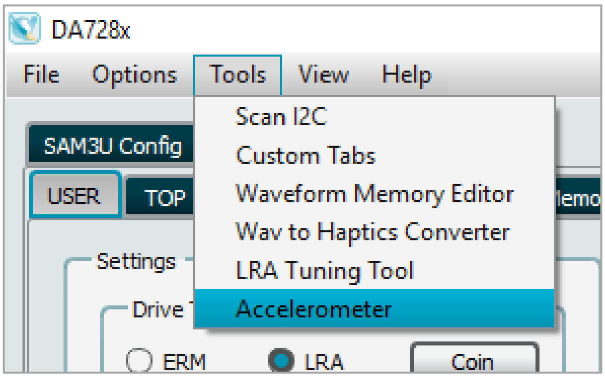

Figure 68. Digital accelerometer

5. In the **Accelerometer** window (Figure 69), in the **WM Seq ID** list, select the waveform memory sequence ID and then to play the sequence, click **REG TRIGGER**.

The resulting data from the digital accelerometer is sampled from the **Axis to sample** selection in the GUI. The data is then displayed in a graph which is useful for analyzing the output acceleration when an oscilloscope is not available.

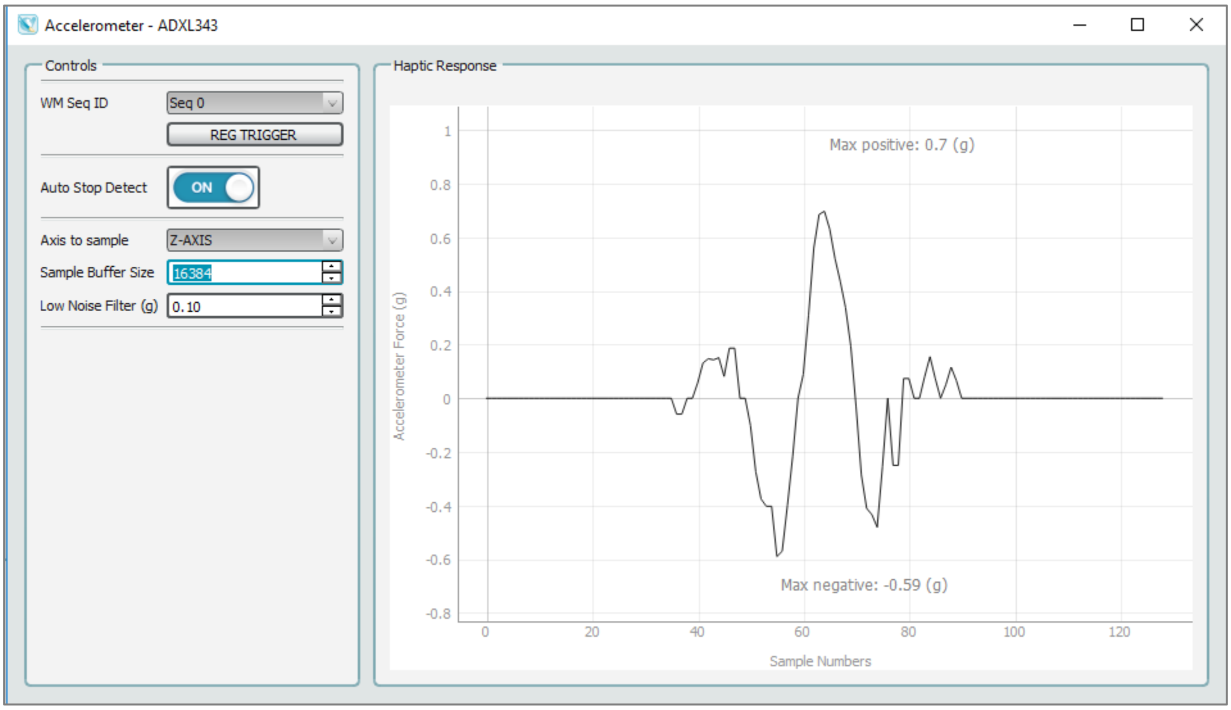

Figure 69. Digital accelerometer GUI

11 Measure Current on VDD and VDDIO

To measure the current taken by the VDDIO and VDD supplies routed to the daughterboards, the jumpers on J1 can be removed as shown in Figure 70.

The default VDD supply voltage is set to 3.8 V and is used to power the VDD level for the DA728x daughterboards except for the case when the onboard boost is enabled on the DA7280 daughterboard. When the onboard boost is enabled on the DA7280 daughterboard, the 3.8 V VDD is boosted to 5 V with the onboard boost circuit located on the underside of the DA7280 daughterboard. When the boost is disabled, the VDD input to the board is then used directly to power VDD at 3.8 V. There is no boost option on other variants of daughterboard.

To use other VDD supply levels or to power the DA728x daughterboards externally, LNK2 must be removed from J1 on the motherboard. The positive rail of a power supply should be connected to pin 4 of J1 and the negative rail should be connected to any GND point on the DA728x motherboard. To measure the current though VDD, LNK2 can be removed from J1 and replaced with an ammeter.

There is also an option to allow the VDD to alternatively be connected directly to the USB 5 V supply as the default level. To use this option the VDD jumper (LNK2) on J1 must be removed, A solder short can then be placed on SP10 which connects the VDD rail to the 5 V USB. This can be useful when needing a higher VDD supply but the current available to the DA728X daughterboards is limited to only what current the USB can supply along with the power consumption taken from other circuitry on the motherboard such as the microcontroller and accelerometers.

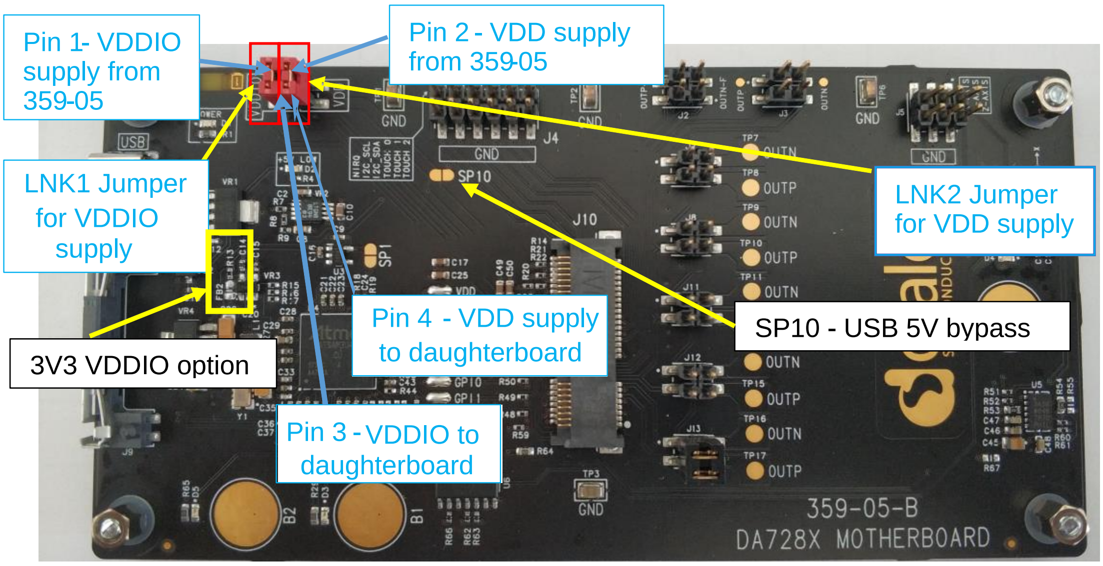

Figure 70. Current measurement and external supply connections

The default VDDIO supply voltage is set to 1.8 V and is directly supplied to the daughterboard VDDIO which also contains pull-ups such as I2C and nIRQ which are tied to this supply. To measure the current though VDDIO, LNK1 can be removed from J1 and replaced with an ammeter.

When measuring standby and shutdown currents, connect a suitable meter capable of measuring currents of nA range. Ensure that the currents for any pull-ups are taken into consideration, such as I2C, GPIs, and nIRQ lines which are located on the daughterboard. Also ensure that **Polling Enabled/Disabled** in the SmartCanvas DA728x GUI is set to disabled so that the device is not receiving/sending I2C information which affects the VDDIO current measurement.

To use other VDDIO supply levels or power the DA728x motherboard externally, LNK1 must be removed from J1. The positive rail of a power supply should be connected to pin 3 of J1 and the negative rail should be connected to and GND point on the DA728x motherboard.

Alternatively, there is an option to set VDDIO to 3V3 by removing R13 and placing it on FB2 which is shown in Figure 70. See the motherboard schematic and layout on the supplied memory stick.

12 Measure Current Going through LRA

You can measure current going through an LRA by only populating one side of the double jumper used to select the LRA and replacing the other jumper with a short length of wire. A current probe can then be clipped onto the loop of wire that allows the precise current flowing through the LRA to be measured using an oscilloscope. See Figure 71 and Figure 72.

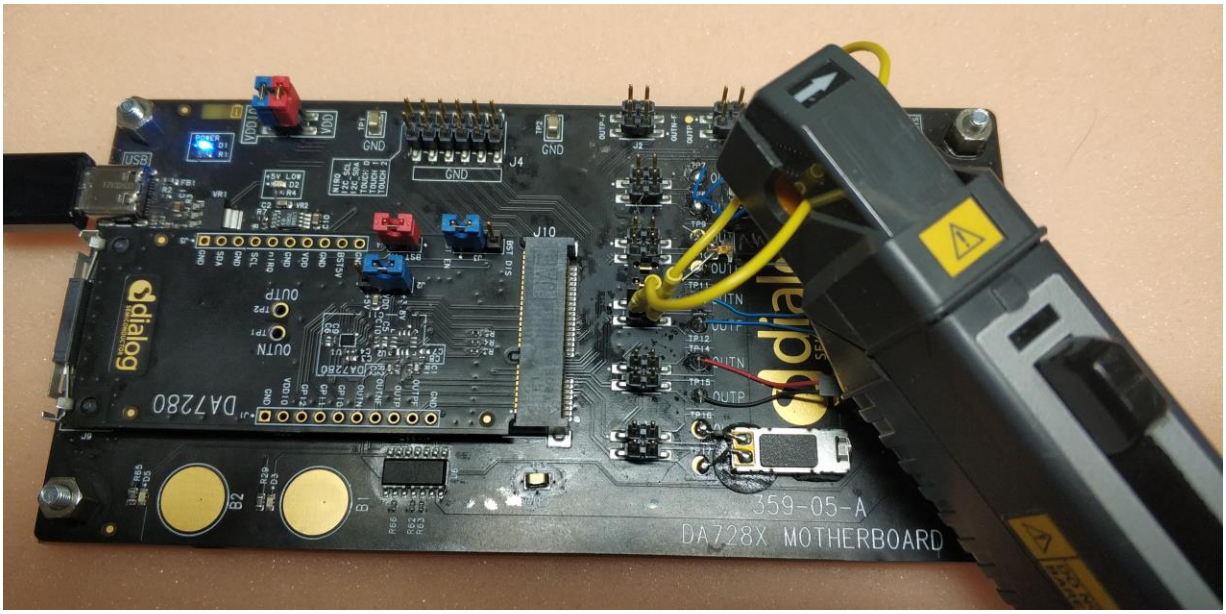

Figure 71. Current measurement of drive current into LRA

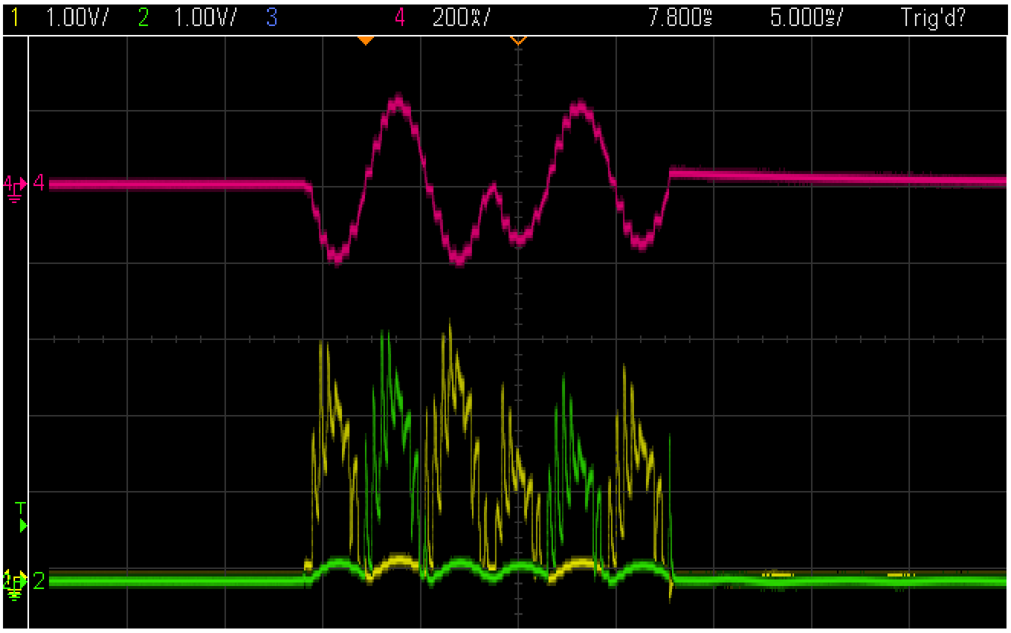

Figure 72. Green/yellow drive signals (sine wave mode), magenta = current probe output

13 Connect Multiple Devices on the Same I2C Bus

The DA7281 is a haptics device in the DA728x family that supports multiple I2C addresses (for detailed information, see Ref. [2]). An example of two DA7281 connected to the same I2C bus is shown in Figure 73. This shows the DA728x motherboard with a DA7281 daughterboard plugged into J10 and an additional DA7281 daughterboard wired externally sharing the same I2C bus. It is also possible to set up this example in a similar way with the DA7280 board and the DA7281 board.

In this hardware configuration, you can set up and control two different LRAs. SmartCanvas DA728x GUI can communicate with the DA7281 daughterboard plugged into J10 of the DA728x motherboard using the slave address 0x4A and with the other externally wired DA7281 daughterboard using the slave address 0x48. Both these slave addresses are configured using the J2 and J3 jumpers that are located on the DA7281 daughterboards as shown in Figure 73.

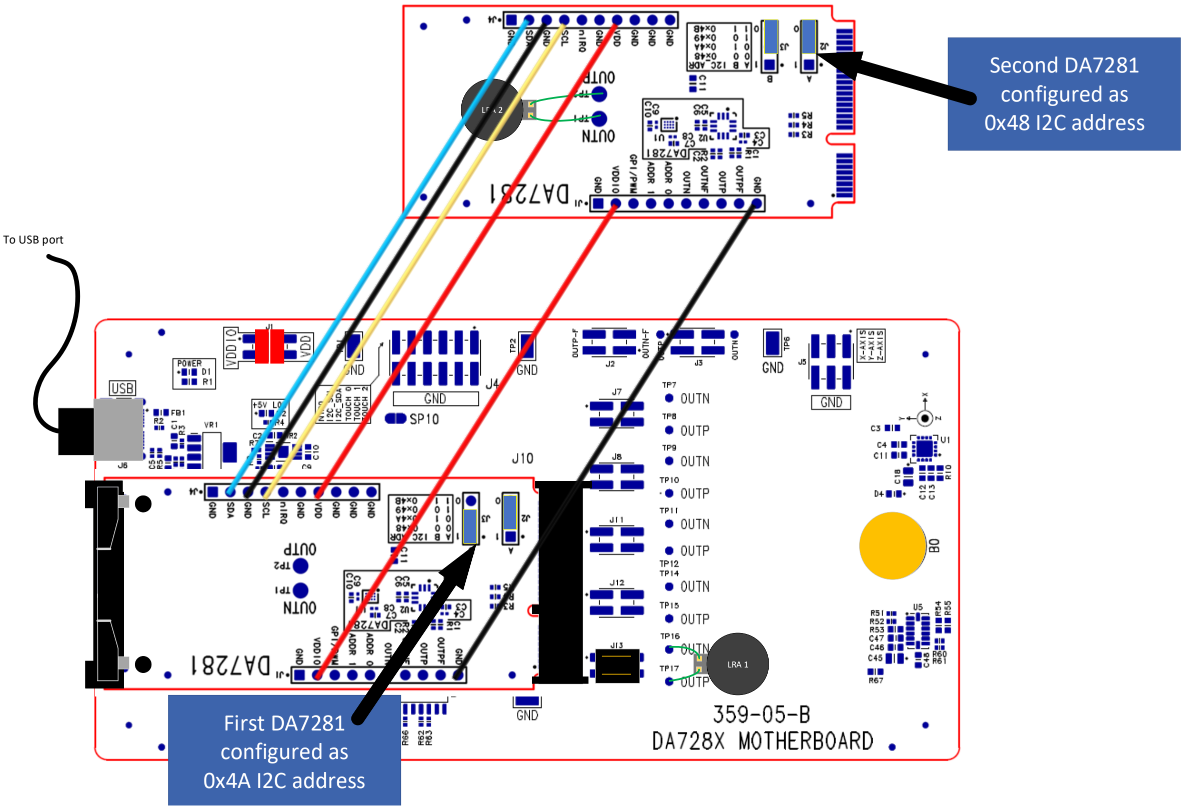

Figure 73. DA728x motherboard configuration for multiple DA7281 daughterboards

By wiring ADDR_0 and ADDR_1 high or low (daughterboard jumpers J2 and J3), the DA7281 daughterboards can have I2C addresses of 0x48, 0x49, 0x4A, or 0x4B. This 7-bit base address is left shifted by one bit and the R/W bit is inserted into the new LSB for the address to become eight bits in length. This gives a bus address of 0x90, 0x92, 0x94, and 0x96 for performing an I2C read command depending on the configuration of the ADDR_0 and ADDR_1 lines for each device.

To support multiple DA7281 devices, you can select which bus address is used by one device in the **Bus Interface** list of the SmartCanvas DA728x GUI as shown in Figure 74.

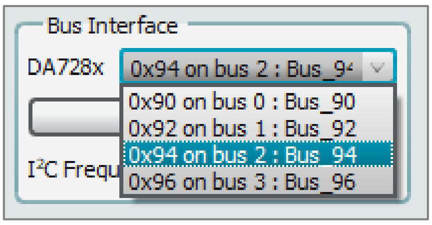

Figure 74. Changing I2C address settings

The default I2C address for DA7280 is 0x4A that translates into a bus address of 0x94 for an I2C read or 0x95 for an I2C write. The SmartCanvas DA728x GUI handles this when the bus address is set. The ADDR_0 and ADDR_1 pins of the DA7281 directly control bits 0 and 1 of the I2C (7-bit addressing) address bus. This is shown in Table 7 and describes the 7-bit and 8-bit equivalent address. The bits, which are determined by the ADDR_0 and ADDR_1, are indicated by the brackets in the binary values.

Table 7. I2C hex/binary settings for DA7281

	8-Bit Addressing **0 Padded to LSB for Read or 1 padded to LSB for Write** **Values for Addr_0 & 1 pins shown in (xx) – (pulled high or low)**			
	Read		**Write**	
	Hex address	**Bin**	**Hex address**	**Bin**
	0x90	10010(00)0	0x91	10010(00)1
	0x92	10010(01)0	0x93	10010(01)1
default address for DA7280 0x4A = (7 bit) 1001010b	0x94	10010(10)0	0x95	10010(10)1
	0x96	10010(11)0	0x97	10010(11)1

Note:

- The SmartCanvas DA728x GUI does not automatically update this address, so you should set this accordingly which address you want to communicate with.
- For the hardware configuration shown in Figure 73, the Bus Interface should be set to 0x94 to control the DA7281 daughterboard connected to J10 of the DA728x motherboard and 0x90 for the externally wired DA7281 daughterboard. When you switch Bus Interface addresses, ensure **Polling** is enabled so that the SmartCanvas DA728x GUI refreshes the current state of the haptic driver.
- Consider the supply current needed for VDD, as driving multiple LRAs simultaneously could result in not enough current to supply more than one DA728x device. It is recommended that an external VDD supply is used so that there is always enough current available.
- It is possible to stack up to four DA7281 daughterboards on one DA728x motherboard by using long tail stackable header pins such as https://www.digikey.co.uk/products/en?keywords=SAM1206-02-ND (see Figure 75).
- Performance measurements using the onboard accelerometer is not recommended when multiple daughterboards are stacked on one motherboard, as there may be mechanical movements of the boards and wires which affect the accelerometer output.
- It is possible to connect up to 128 DA7281 devices to one I2C bus by actively changing the I2C base addresses of the DA7281 devices in groups of four devices. This requires each bank of four DA7281 daughterboards to be enabled through a GPI to power the VDDIO supply so that four devices can be selected at a time for I2C communication. For more deails, see Ref. [2].

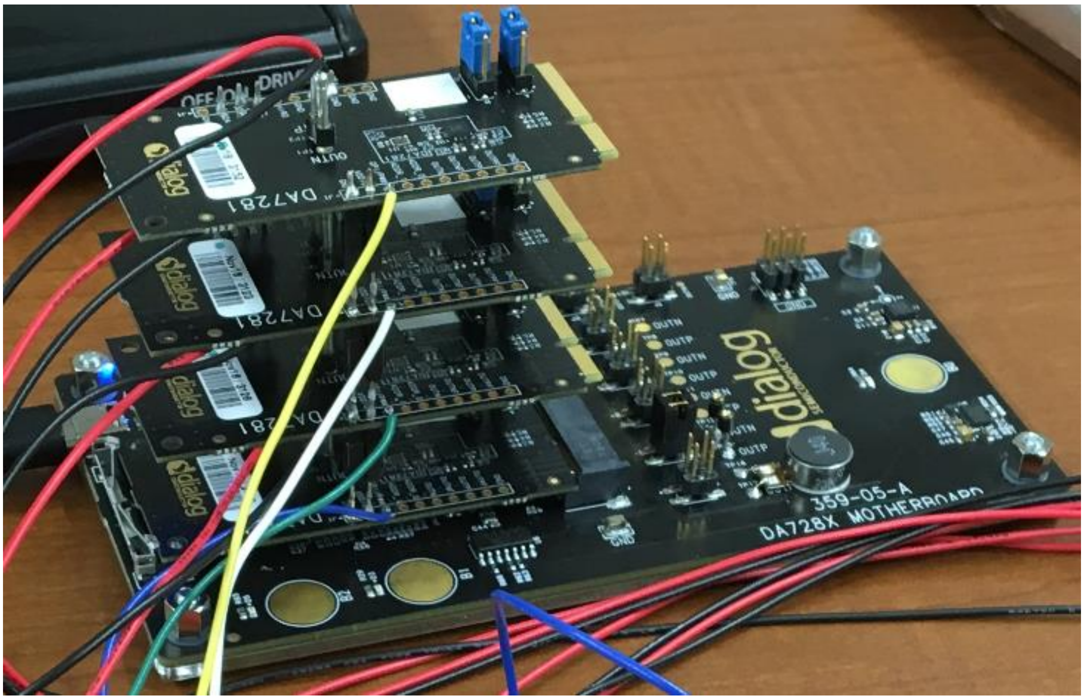

Figure 75. DA728x motherboard with four DA7281 daughterboards stacked

13.1 I2C scan tool

The SmartCanvas DA728x GUI has a useful I2C scanning tool that can help debug the available I2C addresses when multiple DA7281 and other I2C devices are sharing the same bus. To access this tool, select **Tools** > **Scan I2C** and the **I2C Bus Scan** window appears (Figure 76).

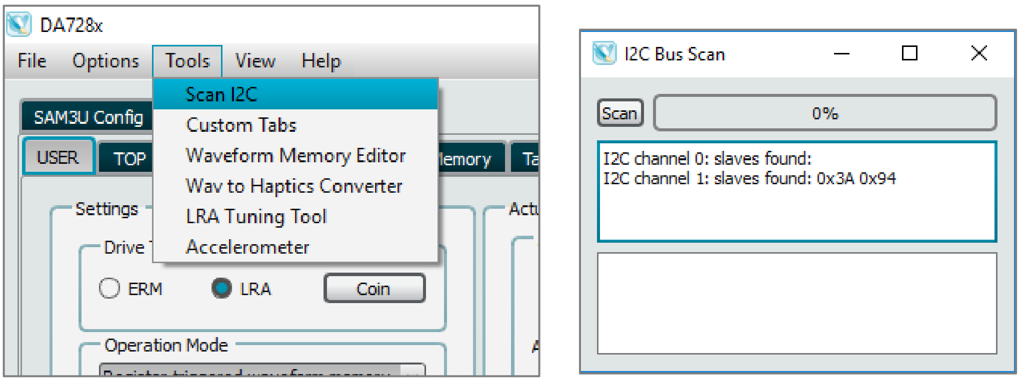

Figure 76. I2C scan tool

Clicking the **Scan** button interrogates the I2C bus and shows a list of data with the addresses of all devices found.

Revision History

Revision	Date	Description
1.2	Mar 29, 2024	Added Titan haptics actuator information. Editorial changes.
1.1	June 08, 2022	Rebranded to Renesas.
1.0	May 5, 2020	Initial version.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.