Application note

DA7219 - Advanced Accessory Detection widget and configuration

AN-AU-063

Abstract

This application note describes the usage of the Advanced Accessory Detect widget with SmartCanvas DA7219 GUI - it also describes configuration issues regarding the AAD.

Contents

Abstract 1

Contents 2

Figures 2

1 Terms and definitions 3

2 References 3

3 Introduction 4

4 System setup 5

5 Jack insertion and impedance detection routine 11

5.1 Jack insertion and no impedance detection routine 13

6 Jack removal 14

7 Conclusions 15

Appendix A 16

A.1 Android button specification (v1.1) 16

A.2 4-pole jack signal configurations 16

Revision history 17

Figures

Figure 1: DA7219 Performance Board 5
Figure 2: Dialog USB-Lab IO 5
Figure 3: USB Drivers 6
Figure 4: Sound settings for digital audio at 48000 Hz sample rate 6
Figure 5: SmartCanvas Communication Widget 7
Figure 6: SmartCanvas Plugin menu 8
Figure 7: Advanced Accessory Detection Widget - disabled mode 8
Figure 8: ‘DAC to HP Path Enabled’ no insertion 9
Figure 9: 4-pole jack inserted 9
Figure 10: Button A Released 10
Figure 11: SmartCanvas Polling Enabled 10
Figure 12: AAD Widget with Impedance Detection Enabled 11
Figure 13: AAD sequence on jack insertion with impedance detection enabled 11
Figure 14: AAD Widget with Impedance Detection Disabled 13
Figure 15: AAD sequence with no Impedance Detection routine 13
Figure 16: Jack removal timing capture 14

1 Terms and definitions

AAD	Advanced Accessory Detect
CTIA	Configures the jack as MIC-GND-RIGHT-LEFT (see- 16)
DA7219	Dialog Semiconductor's audio CODEC with Advanced Accessory Detection
OMTP	Configures the jack as GND-MIC-RIGHT-LEFT (see- 16)
EVB	Evaluation Board containing DA7219
SmartCanvas GUI	Dialog Semiconductors graphical user interface for controlling DA7219
SAM3U	USB controller and audio soundcard with custom Dialog Semiconductor firmware. Windows defines this as Dialog USB-Lab IO in Control Panel – Hardware and Sound - Devices and Printers (details here 3).

2 References

[1] DA7219 Datasheet, Dialog Semiconductor

3 Introduction

The Advanced Accessory Detect (AAD) plugin for SmartCanvas DA7219 GUI allows the user to demonstrate the AAD features on DA7219. This application note will describe how to setup the system for this demonstration and also the configuration required to run AAD, while also the requirements of the host controller for the DA7219 with respect to sequencing register writes to the CODEC.

The AAD has the following features:

- Signal that the jack is being inserted
- Signal that the jack is fully inserted
- Signal jack has been removed
- Determine if the jack has 3 or 4 poles
- Determine if the jack is CTIA or OTMP configuration
- Determine the impedance of the attached headphone/headset/Line output
- Fully configurable Android Wired Spec 1.1 compliant 4-button detection

4 System setup

The SmartCanvas GUI runs on a Windows PC that is connected with a USB cable to the SAM3U on the DA7219 EVB. SmartCanvas acts as the host controller (master) in this system, while the DA7219 CODEC is the slave audio device. To setup the system the following is required.

1. Install the GUI 'setup_DA7219_Evaluation_GUI_1.1.0.x.exe' but do not run the program just yet.
2. Connect the DA7219 EVB to the computer via the USB cable. Windows will install drivers if this is the first time that the EVB is connected to the computer.

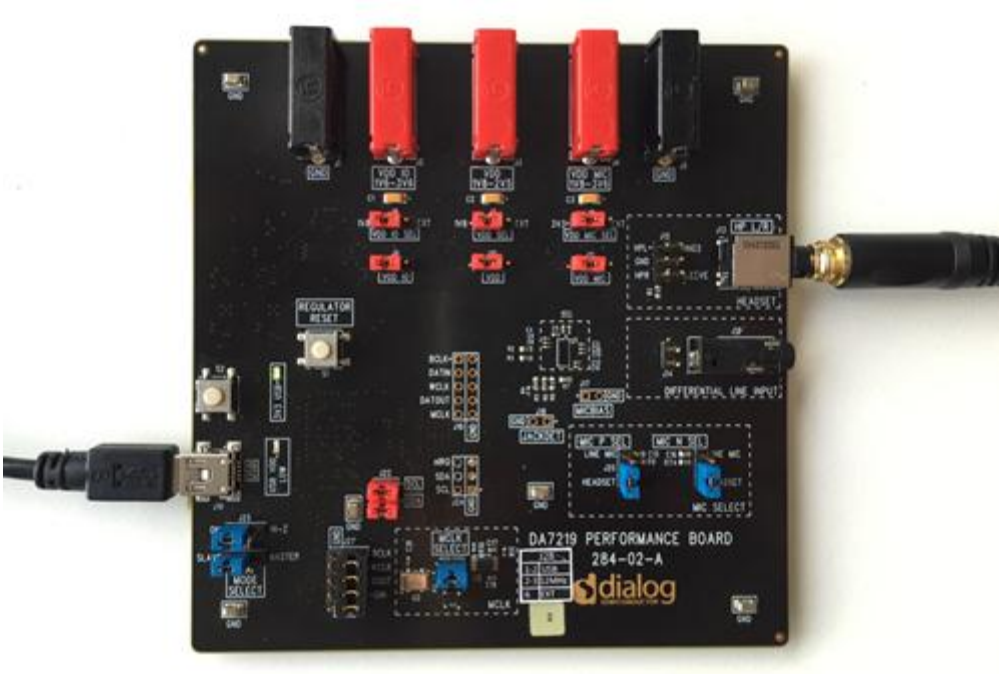


Figure 1: DA7219 Performance Board

3. All 3 drivers shown below should be installed automatically. The SAM3U device is called Dialog USB-Lab IO in in Control Panel – Hardware and Sound - Devices and Printers.

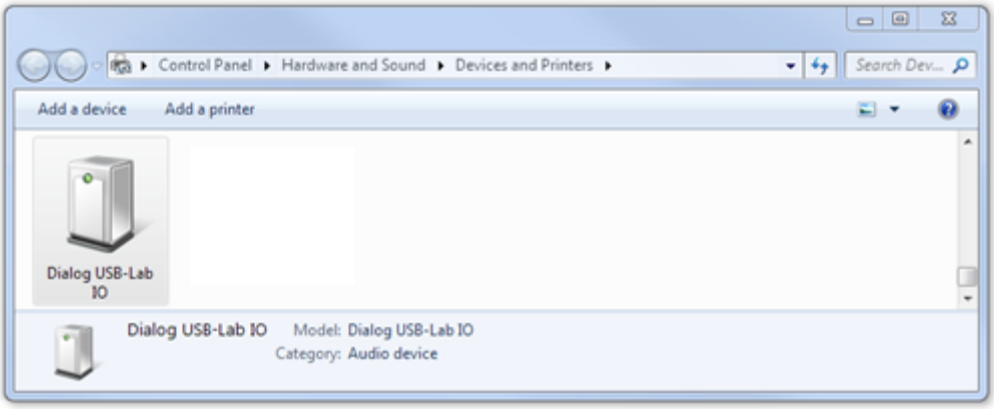


Figure 2: Dialog USB-Lab IO

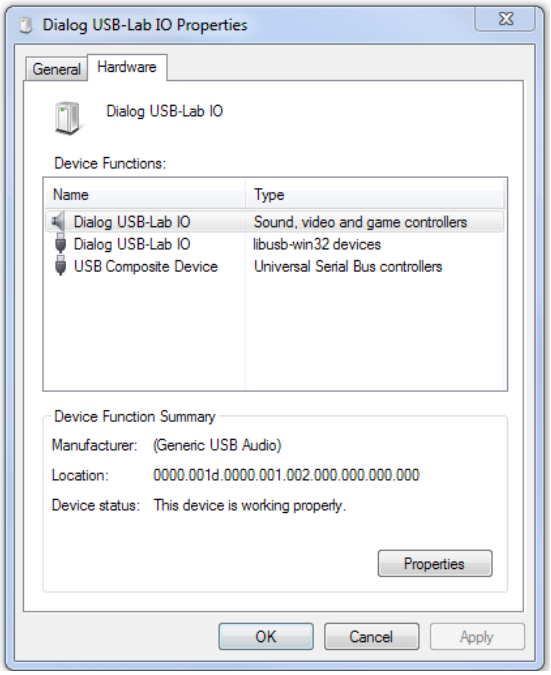


Figure 3: USB Drivers

4. Stream audio to the USB device on the EVB, for example use Windows Media Player.

    a. Ensure the sample rate is set to 48000 Hz in the Windows Sound, Playback, Advanced tab, see Default Device below.

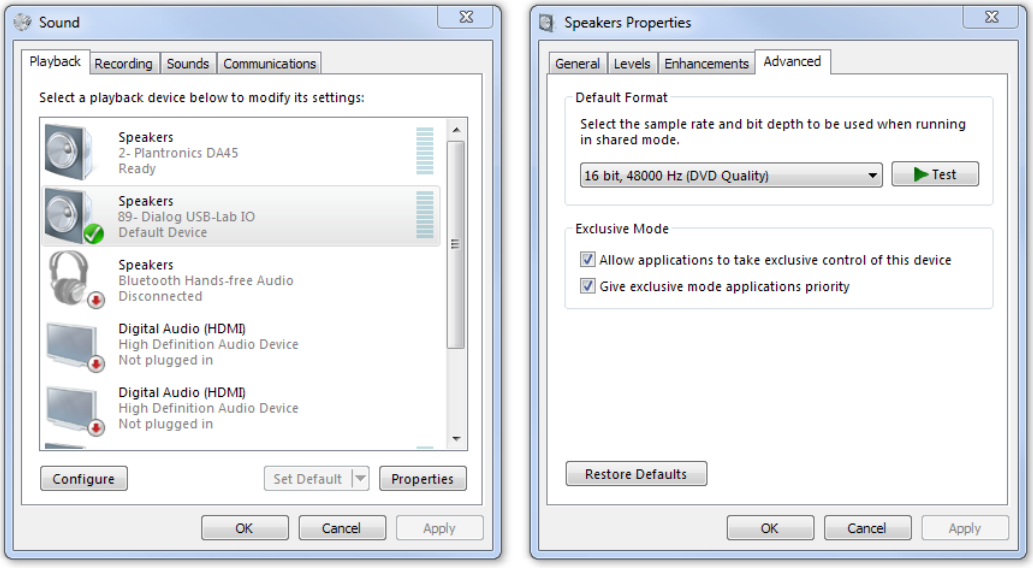


Figure 4: Sound settings for digital audio at 48000 Hz sample rate

5. Connect headphones or headset to the EVB - note the button detection is setup for standard android headsets (see section - A.1)

6. Launch the GUI, the GUI will indicate that Communication to the I2C and USB are active by illuminating the LEDS in a green colour. Flashing on and off in a red colour indicates there is an issue with communications between software and hardware.

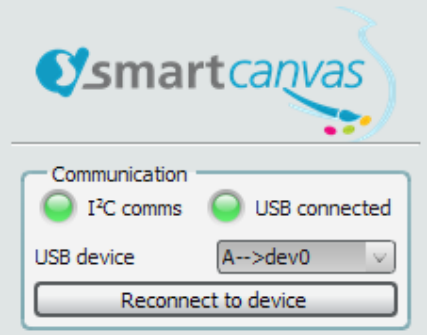


Figure 5: SmartCanvas Communication Widget

7. From the Plugins menu select the '**Adv Acc Det**' widget. The Advanced Accessory Detection Widget is shown below and on start-up its functionality is disabled until pressing the **'Enable DAC to HP Path**' button.

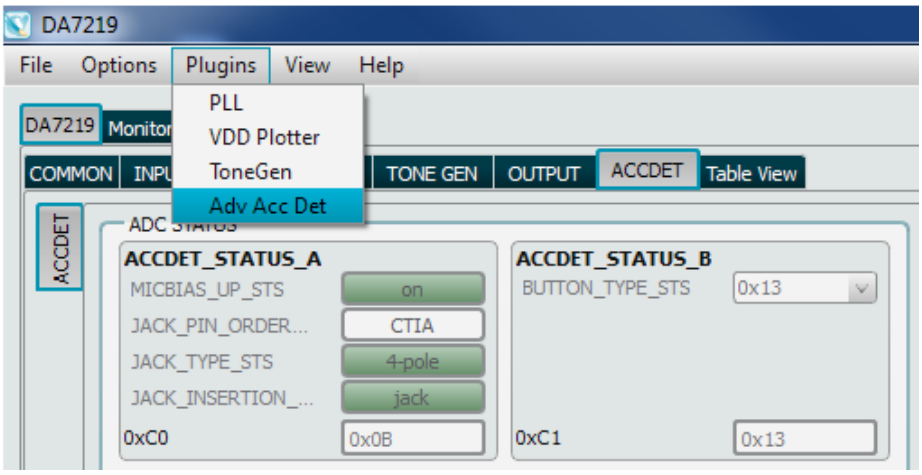


Figure 6: SmartCanvas Plugin menu

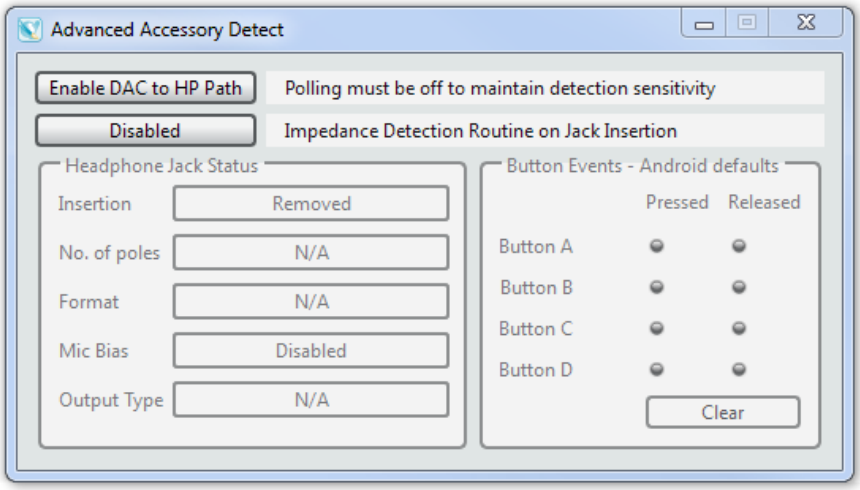


Figure 7: Advanced Accessory Detection Widget - disabled mode

8. Press '**Enable DAC to HP Path**' button. You should now hear the audio playing through the headset/headphones.

   The GUI sets up the following:

   a. The Polling of the CODEC's register map is disabled to improve latency.

   b. The software sets up the CODEC's audio path to accept 48 kHz I2S audio from SAM3U.

   c. DAC to Headphone path is configured.

   d. The system is continually polling for interrupts from the AAD. On reception of AAD interrupt the nIRQ line driven from DA7219 connected to GPIO on the SAM3U is pulled low.

9. Remove the jack from the EVB and listen to the transition of the removal of this jack - you will not hear any nasty pops, clicks or distortion caused by this removal of the jack due to the AAD and careful muting of the headphones output amplifier.

   a. On receiving an interrupt from the AAD, the system quickly mutes the headphones amplifier so that you hear a clean transition as the jack is removed.

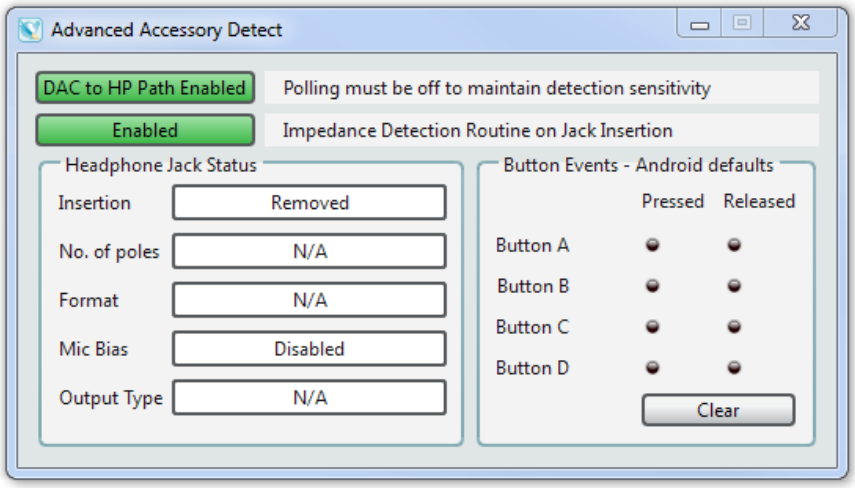


Figure 8: 'DAC to HP Path Enabled' no insertion

10. Inserting the jack you will hear a clean transition as the AAD determines if the jack is headphones or line output impedance, after this you will hear the audio unmuted without any pops or clicks.

    a. On receiving a jack insertion interrupt, the system determines the impedance of the headset/headphones plugged into the jack.

    b. It then unmutes the headphones to give a clean transition from no audio to audio.

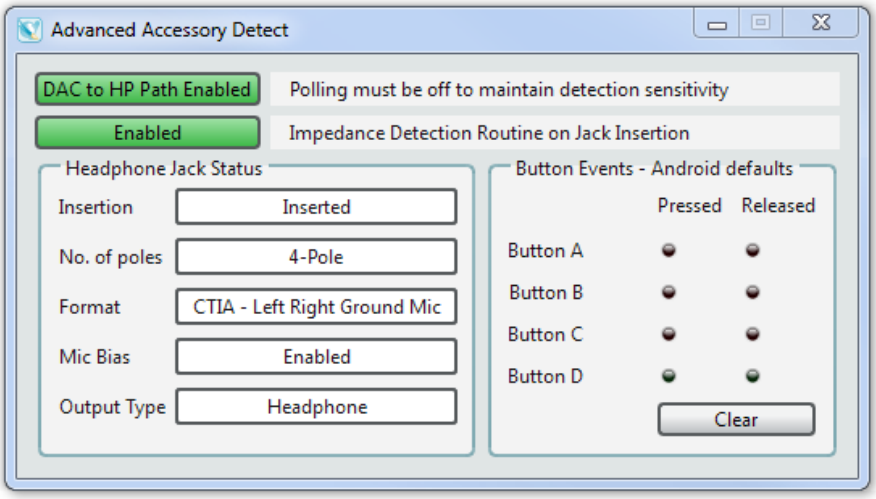


Figure 9: 4-pole jack inserted

11. The GUI will indicate the headset/headphones specifics such as number of poles, configuration type and output type. Hovering over **Format** and the **Button Event** group box will show graphically these specifications.

12. Pressing the buttons on the headset will show that the button is been **Pressed** when held down, on release it indicates that the button has been **Released**.

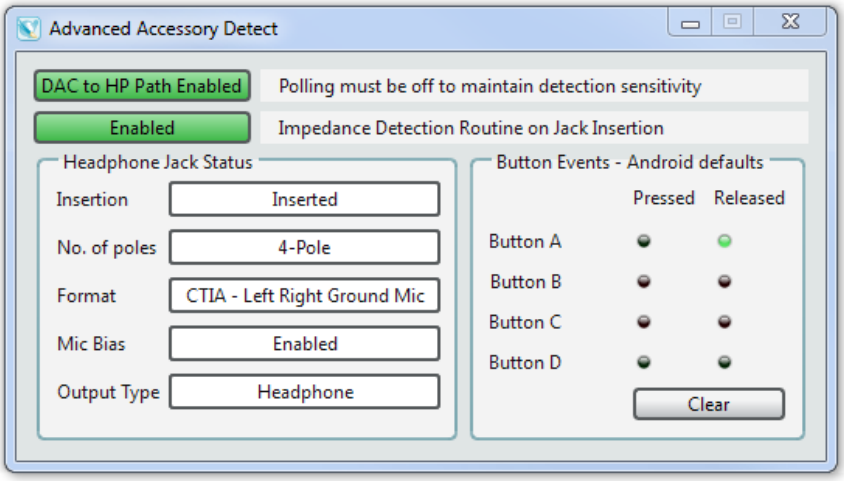


Figure 10: Button A Released

13. Closing the plugin will reset the Polling state of SmartCanvas GUI to its previous state before launching the AAD plugin. The rest of the internal registers of the audio CODEC will remain untouched.
14. If you do not hear a clean transition inserting or removing the jack headphone/headset the following could be the reason:
    a. The AAD plugin is not running.
    b. The **Enable DAC to HP Path** button is not pressed and turned green displaying **DAC to HP Path Enabled**.
    c. The **Enable/Disable** Polling in SmartCanvas is **Enabled**, as shown below.

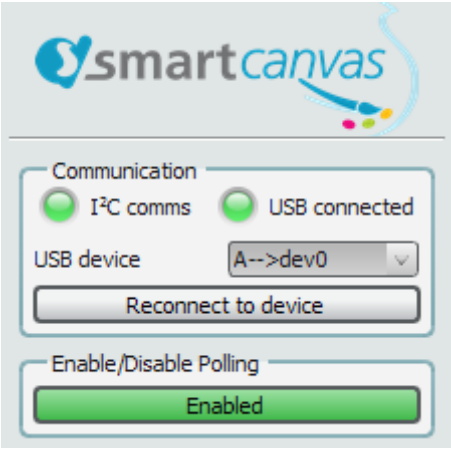


Figure 11: SmartCanvas Polling Enabled

    d. Computer/laptop is in power saving mode which affects the response time of the GUI to complete I2C writes. See section - Jack removal.

5 Jack insertion and impedance detection routine

On insertion of the jack the following events occur - as shown in Figure 13: AAD sequence on jack insertion with impedance detection enabled.

1. AAD determines if the jack is 3 or 4 pole.
2. AAD determines if the 4 pole jack is CTIA or OMTP configuration.
3. If the 'Impedance Detection Routine' button is enabled, the routine to determine this impedance is run.
4. The headphone amplifiers are unmuted by the system controller and audio starts playing around 370ms.

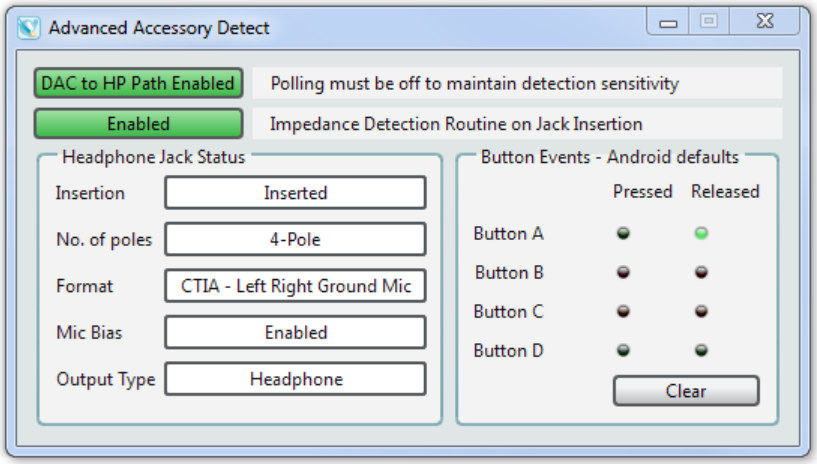


Figure 12: AAD Widget with Impedance Detection Enabled

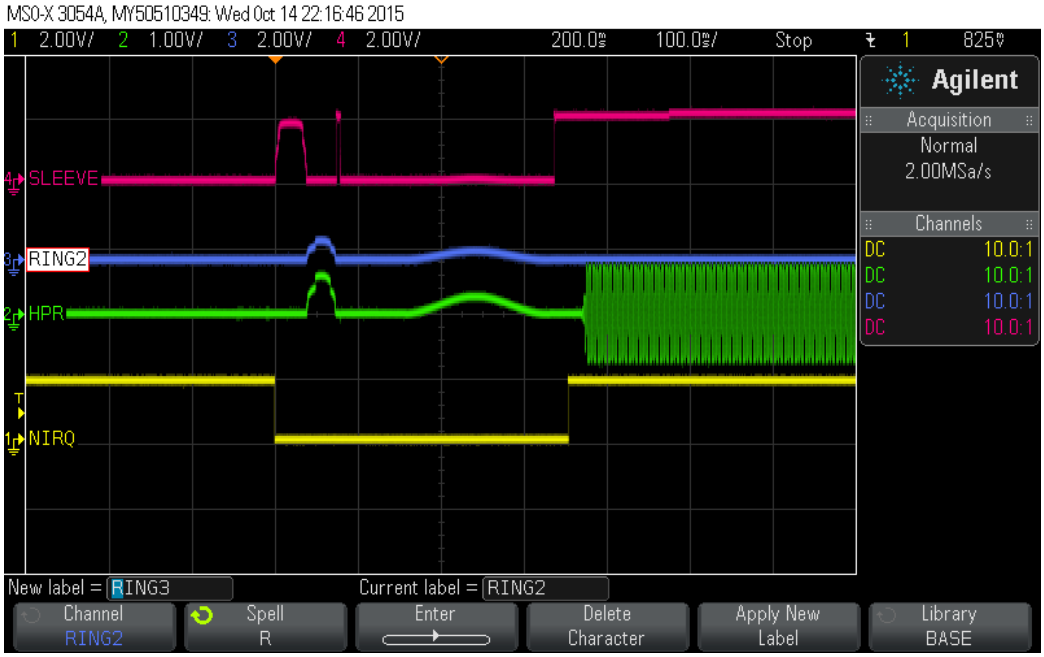


Figure 13: AAD sequence on jack insertion with impedance detection enabled

On insertion of the jack the **ACCDET_IRQ_EVENT_A** register value 0xC2 indicates the various states while inserting the jack. When bit 2 of the **ACCDET_IRQ_EVENT_A** register **E_JACK_DETECT_COMPLETE_[2]** is set - this indicates that a jack is fully inserted and we can begin the impedance detection routine. The following are the necessary writes to the CODEC required to determine the impedance of the attached headphones/headset/Line output.

Note: The hex value is the value that should be written to the register named accordingly.

1. 0x0B, 'MICBIAS_CTRL' // Enable MICBIAS
2. 0xD7, 'ACCDET_CONFIG_1' // Sets up and enabled AAD
3. 0x00, 'ACCDET_CONFIG_2' // Sets defaults for detection
4. 0x01, 'ACCDET_CONFIG_8 // Enables HP Test 300mV signal, impedance threshold of 1000 Ohms
5. 0x00, 'TONE_GEN_CFG1' // Disable Tone Generator
6. 0x00, 'DAC_FILTERS1' // Disable the high pass filter
7. 0x53, 'TONE_GEN_CFG2' // Set tone gen gain to -15dB and sramp function
8. 0x3F, 'TONE_GEN_ON_PER' // Continuous tone gen settings for ON
9. 0x3F, 'TONE_GEN_OFF_PER' // Continuous tone gen settings for OFF
10. 0x07, 'TONE_GEN_CYCLES' // Infinite cycles i.e. always ON
11. 0x23, 'TONE_GEN_FREQ1_L' // OP frequency = ~6 Hz
12. 0x00, 'TONE_GEN_FREQ1_U‘ // OP frequency = ~6 Hz '
13. 0x11, 'DIG_ROUTING_DAC' // Setup the tone gen to drive the DAC’S
14. 0x39, 'HP_L_GAIN' // Set for 0dB gain
15. 0x39, 'HP_R_GAIN' // Set for 0dB gain
16. 0x01, 'MIXOUT_L_SELECT' // DAC L as output
17. 0x01, 'MIXOUT_R_SELECT’ // DAC R as output
18. 0x80, 'MIXOUT_L_CTRL' // Enable MIXOUT L amplifier
19. 0x80, 'MIXOUT_R_CTRL' // Enable MIXOUT R amplifier
20. 0xE8, 'HP_L_CTRL' // Enabled, muted, ramped, driven
21. 0xE8, 'HP_R_CTRL' // Enabled, muted, ramped, driven
22. Delay 30ms // Allow amps to switch on
23. 0xA8, 'HP_L_CTRL' // Enabled, un-muted, ramped, driven
24. 0xA8, 'HP_R_CTRL' // Enabled, un-muted, ramped, driven
25. 0x80, 'TONE_GEN_CFG1' // Tone gen enabled
26. Delay 80ms // Allow tone gen to run at least one cycle - this value could be reduced
27. Read 'ACCDET_CONFIG_8' // 0x01= Imp > 1000 Ohms (Line) , 0x11 = imp < 1000 Ohms (headphones)
28. 0x00, 'TONE_GEN_CFG1' // Disable tone generator
29. Delay 80ms // Allow time for signal to return to 0v - this value could be reduced
30. 0xE0, 'HP_L_CTRL' // Un-drive headphone amps
31. 0xE0, 'HP_R_CTRL' // Un-drive headphone amps
32. 0x10, 'ACCDET_CONFIG_8' // Disable HP Test
33. 0x32, 'DIG_ROUTING_DAC' // Setup the DAI to drive the DAC’S
34. 0x80, 'DAC_FILTERS1' // Enable the high pass filter

5.1 Jack insertion and no impedance detection routine

If the Impedance Detection Routine is disabled as shown below, the impedance detection routine is not run and the time to hear audio is around 120ms.

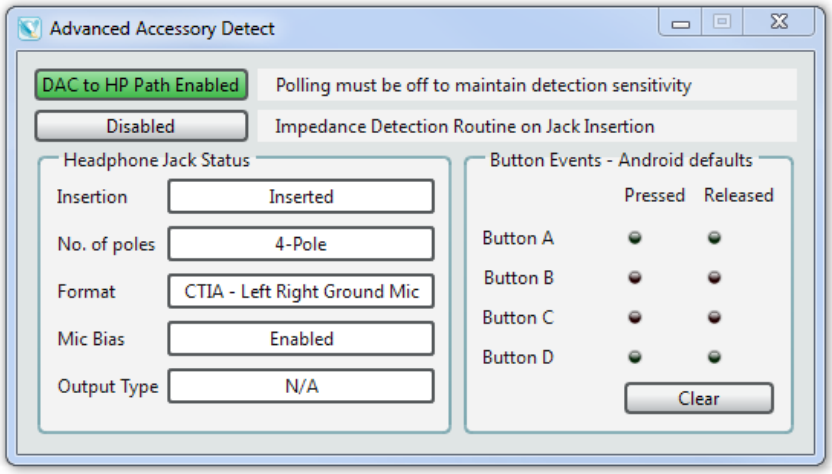


Figure 14: AAD Widget with Impedance Detection Disabled

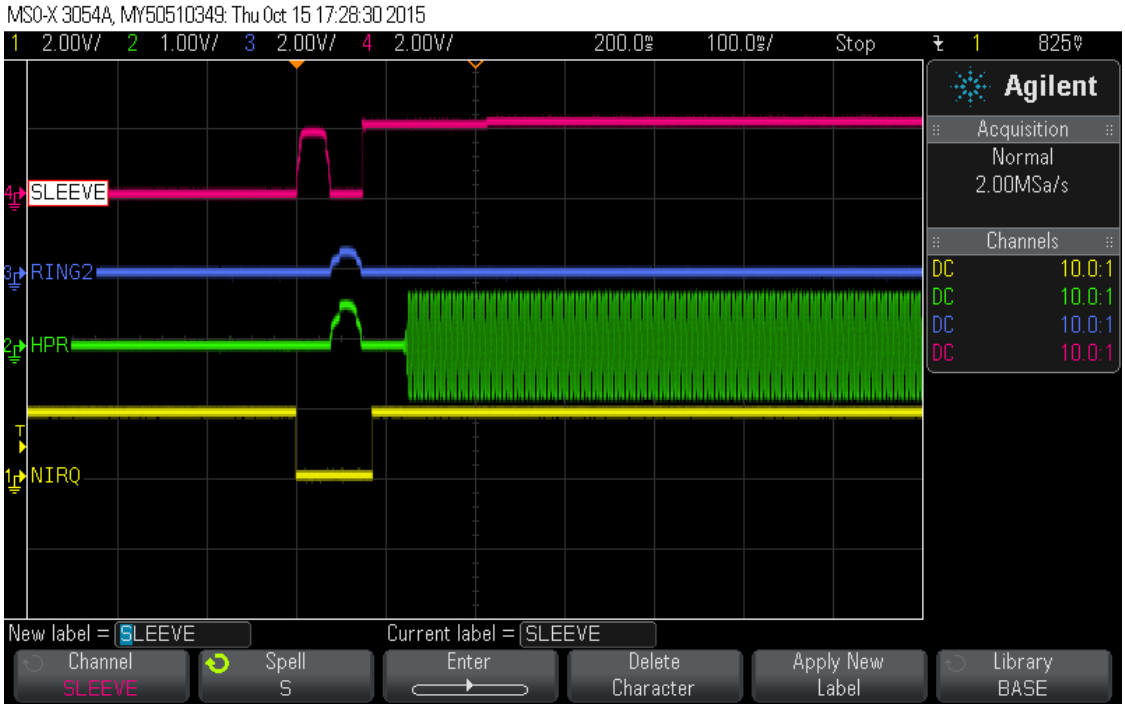


Figure 15: AAD sequence with no Impedance Detection routine

6 Jack removal

On removal of the jack the **ACCDET_IRQ_EVENT_A** register (0xC2) indicates the various states while removing the jack. When bit 1 **E_JACK_REMOVED_[1]** is set, this indicates that a jack is not present and we can mute the headphones.

When the nIRQ line goes low (channel 1 – yellow trace) the host controller has to disable the headphones to avoid the listener hearing any pops or general unwanted noise.

The following times are testing conditions found by experimenting with removal of the jack in the lab - it simply means that dependanat on how fast the user removes the jack from the socket the headphone amplifier must be muted before the specified removal time has elapsed.

- Medium removal time = 35ms
- Fast removal time = 19ms
- Very fast removal time = 11ms

Shown below in this testing instance, the muting for headphone right output channel (HPR) register write happens before 4ms, after this the gain of the HPR is ramped down, you hear a clean muting of the headphones.

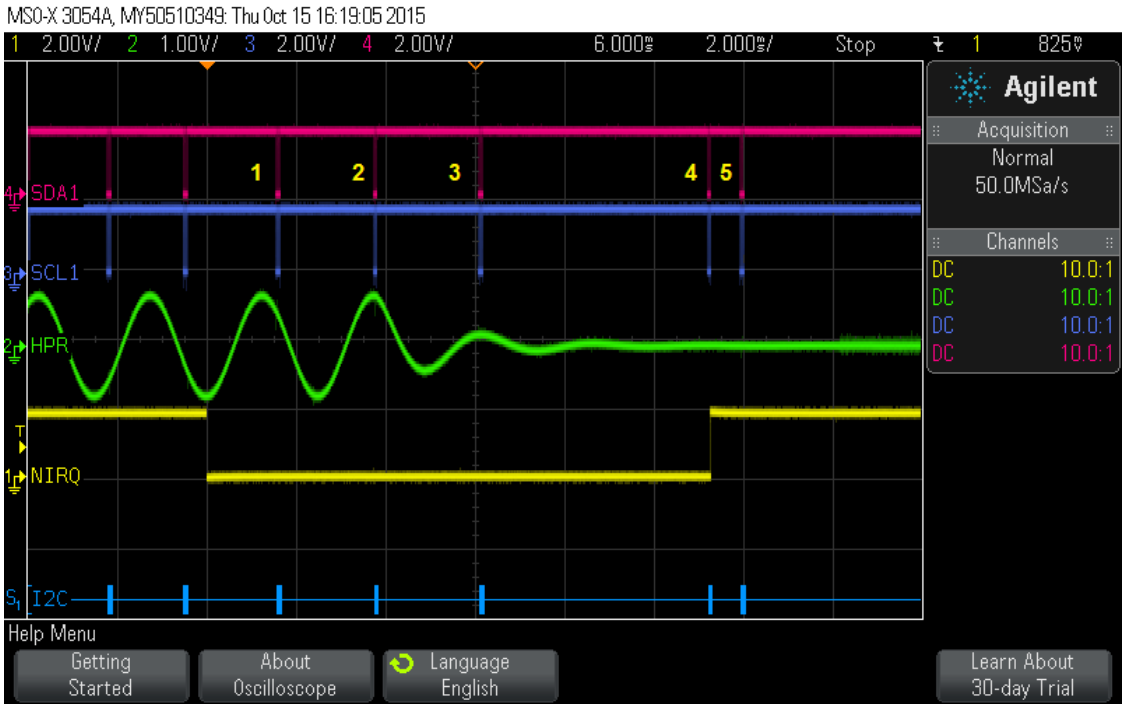


Figure 16: Jack removal timing capture

Note: In an embedded system the responses times for doing the sequence below would be considerably shorter.

Sequence of events after nIRQ goes low:

1. Read the **ACCDET_IRQ_EVENT_A** register to determine what has occurred, in this case **E_JACK_REMOVED_[1]** is set.
2. Write 0xC2 to '**SYSTEM_MODES_OUTPUT**' this sequences the mutes for both headphone amplifiers.
    - This happens < 4ms after nIRQ goes low.
    - The HPR signal as you can see is muted after this write.
3. Read-back of previous write by SmartCanvas – not needed.
4. Write 0xFF to '**ACCDET_IRQ_EVENT_A**' to clear the interrupts.

5. SmartCanvas automatically reads back the previous write to **'ACCDET_IRQ_EVENT_A**' - this is not necessary.

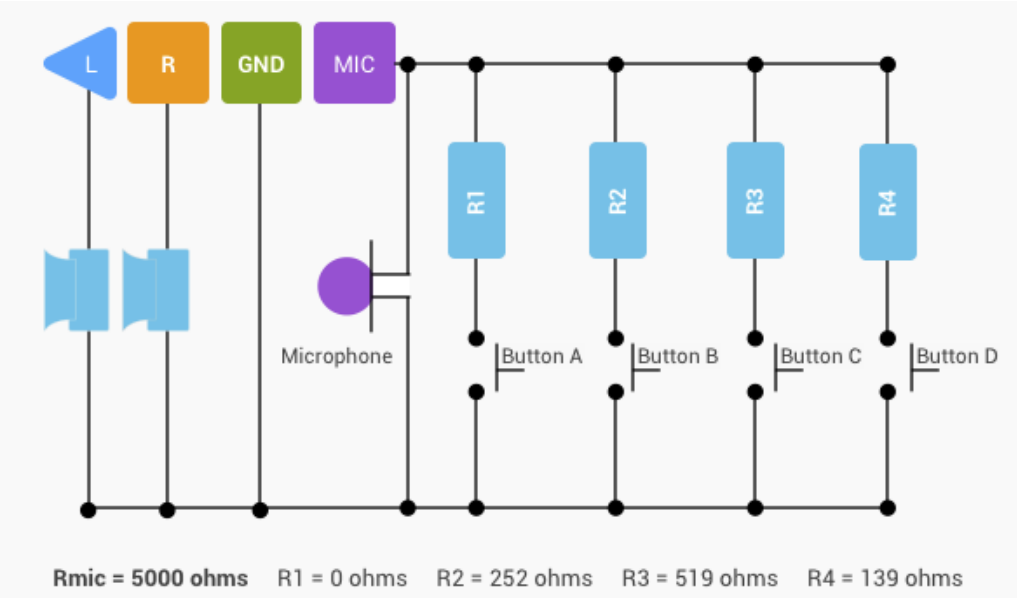
7 Conclusions

The setup of the DA7219 EVB to enable real time listening of the AAD features using SmartCanvas DA7219 GUI has been discussed in detail. On insertion and removal of the jack a detailed explanation of what is required by the host controller was also discussed providing relevant details should the host controller not have access to the Linux driver and a more programmatic method of simple register writes is required.

Appendix A

A.1 Android button specification (v1.1)

DA7219 button detection thresholds are by default setup according to the Android wired audio headset specification v1.1 shown below. Headsets that follow this specification will require no changing of the default settings for button threshold detection, headsets that are not compliant will require changing the thresholds for the specific headsets button impedances.



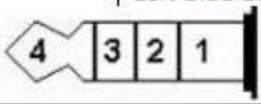
A.2 4-pole jack signal configurations

The following configurations are the current standards.

- Commonly referred as TIP (position 4), RING1 (position 3), RING2 (position 2) and SLEEVE (position 1).

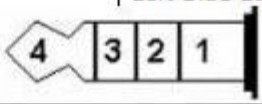
OMTP

PIN	Description
1	Ground
2	Microphone
3	Right Side Earpiece
4	Left Side Earpiece



CTIA

PIN	Description
1	Microphone
2	Ground
3	Right Side Earpiece
4	Left Side Earpiece



Revision history

Revision	Date	Description
1.0	12/10/2015	Initial version.
1.1	28/10/2015	Minor changes to text.
1.2	27/01/2022	Updated logo, disclaimer, copyright.

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.

RoHS Compliance

Dialog Semiconductor complies to European Directive 2001/95/EC and from 2 January 2013 onwards to European Directive 2011/65/EU concerning Restriction of Hazardous Substances (RoHS/RoHS2).
Dialog Semiconductor's statement on RoHS can be found on the customer portal https://support.diasemi.com/. RoHS certificates from our suppliers are available on request.