Application Note

Restricted to customers under NDA

GAP9-based TWS/Headphones Hardware Architectures

with Implementation Guidelines

Rel.1.0

Nov. 3rd, 2023

Disclaimer

This information is subject to change without notice.
Information on this document is provided “as is” without any warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, suitability for a particular purpose, or non-infringement. The information provided in this document is intended for informational purposes only. Information may be changed or updated without notice.

Table of contents

1. DOCUMENT NATURE AND PURPOSE

This document presents hardware architecture options for TWS (True Wireless Stereo) earbuds and Bluetooth-connected Headphones – with single or dual GAP9. These are essentially examples, which can be departed from according to specific constraints or preferences.

Also provided are implementation guidelines in the form of notes pertaining to specific aspects of these architecture.

The intent is to help system architects and PCB designers integrate GAP9 into their target 'hearables' product in an efficient manner, quickly getting to a viable schematic.

Obvisouly, architecture choices should also be made in the light of implications on software complexity.

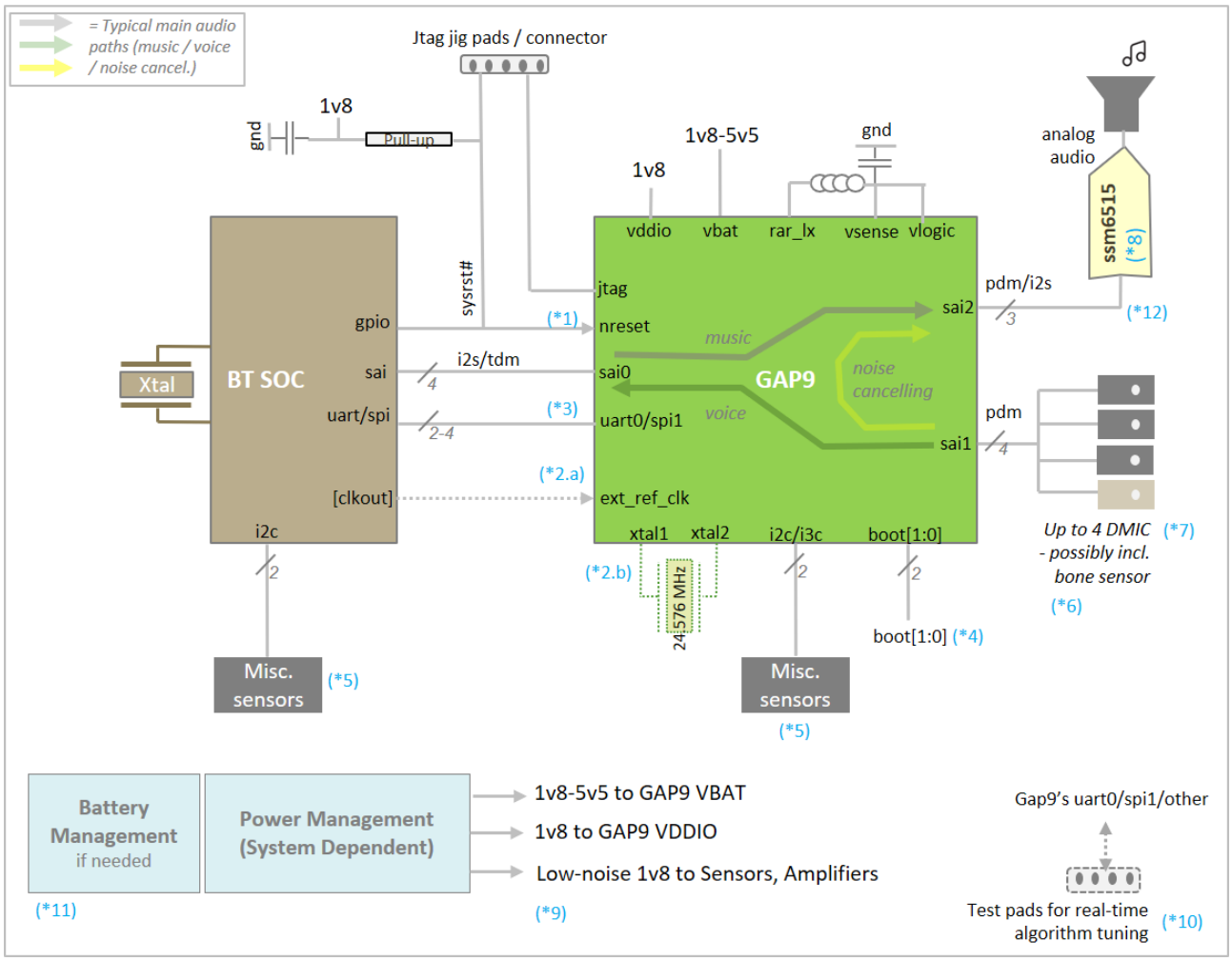
When going to the implementation phase, it is important to read the present document in conjunction with the two Applications Notes referenced at the end of this document : '*GAP9 Hardware Integration Guide'* and *'Audio Clocks for GAP9 – H/W Options and System Implications'.*

Note : Besides the type of product family (TWS or Headphones) and allocated hardware resources and their partitioning (single or dual GAP9, symmetrical architecture or not), an important decision that impacts required interfaces between BT-SOC and GAP9 is whether the audio codec (SBC/AAC/LC3…) is left to the BT-SOC or is running on GAP9. There is a significant energy benefit in running compute-intensive, state-of-the-art audio codecs on GAP9; it may also open up support of BT-SOC models with no readily available I2S/TDM interface and it enables connecting more microphones to GAP9.

In the following, we introduce baseline scenarios where the audio codecs are running on the BT SOC., plus variants where the audio codec runs on GAP9.

2. EXAMPLE TWS ARCHITECTURE

2.1. BLOCK DIAGRAM



Notes numbered « *(*xx)* » in this figure provide additional explainations and implementation guidelines. Please refer to the following section.

2.2. IMPORTANT NOTES

*- relative to the aspects denoted by « (*xx) » within the former figure.*

(*1) GAP9 reset control – For FOTA update (Firmware OverThe Air), BT SOC would typically need to be able to reset GAP9. GAP9 hardware reset also needs to be controllable from JTAG probe. Check ***'GAP9 h/w Integration Guide'*** *[Ref.1]*, section 10.

(*2) GAP9 reference clock – GAP9's reference clock should preferably be a power-of-2 mutiple of 48KHz (or, possibly, of 44.1KHz) – typically, 24.576MHz. If the BT SOC is able to deliver such a clock, then option 2.a saves one crystal. Otherwise, an external crystal (2.b) or a dedicated clock oscillator (smaller, lower power, but a bit more costly) would typically be required. Further explainations in Application Note ***'Audio Clocks for GAP9 – Hardware Options and System Implications'*** *[Ref.2].*

(*3) BT SOC <> GAP9 Control / Auxiliary Data Channel – A channel for control information and non-audio data is normally required between BT SOC and GAP9. It can be implemented over UART (with optional flow control) or, if no uart resource is available, over SPI (GAP9 preferably Master, but Slave mode is supported too, at the expense of sligthly more complex firmware). A GPIO/IRQ line alongside the SPI bus may be useful so the Slave can tell the Master it has some data to send.

(*4) Boot pins – Pins BOOT1 and BOOT0 must be properly set to boot from the right source (JTAG, eMRAM, etc.), refer to '*GAP9 h/w Integration Guide*', section 4. In a product, one option is to drive these pins from BT SOC (then BT SOC must be able to tristate when boot pins need to be forced externally, typ. during GAP9 programming). Another option is to tie them to Logic-0, *i.e*. boot according to eFuses, defaulting to JTAG as long as fuses are not programmed - then program Fuses on production line to boot from required source, *e.g.* eMRAM. Or, enhancing the latter solution: weakly pull BOOT pins to a suitable default value and allow external forcing from test/program jig.

(*5) Sensors – Can connect either to BT SOC or to GAP9, typically through I2C (sensor dependent). There may be system implications to consider; for instance an IMU intended for head tracking may better belong to GAP9 to reduce latency. Also, an important factor to consider may be the availability of relevant software drivers on the target processor.

(*6) Bone Sensor a.k.a Voice Accelerometer a.k.a Voice Sensor – There are different models with different output interfaces that have more or less the same usage. When the output is PDM similar to what a MEMS DMIC provides, then the sensor can share the same SAI interface (in PDM mode) as other DMICS – provided it supports a high enough PDM clock frequency (not always the case). If the output is TDM then it may be possible to multiplex it on SAI0 if the BT SOC<>GAP9 interface can accomodate this (BT SOC MPN and system dependent). Alternatively, it could share SAI2 with SSM6515 provided the latter is operated in PCM input mode (which may degrade latency). Other ad-hoc solutions can be envisaged.

(*7) DMICS – The digital mics (PDM ouput) typically connect to the main PCB through wires/FFC of some length. It is recommended to provision termination resistors (e.g. 10-20 ohm to start with) on clock pin (GAP9 side) and data pins (mic side) to limit reflections and EMI. Note also these mics typically go to deep sleep when no clock is applied, so a power switch on their power supply is not necessarily required. See '*GAP9 h/w Integration Guide*' for details on multiple DMICs shraing one SAI.

Microphones intended as feed-forward and feed-back microphones for ANC should be carefully selected, especially regarding SNR, AOP and group delay. State of the art digital microphones offer a group delay no greater than 7-8us at 1KHz.

(*8) ssm6515 – The SSM6515 is an example suitable Class-D DAC/Amplifier. Other parts may be used, pay special attention though to the input-output delay they introduce as this parameter should be kept very small (a few us) when Active Noise Cancellation (ANC) is required.

(*9) Power Management – The overall power management scheme depends on system architecture and specific chips being used. GAP9 requires a input voltage (pin VBAT) between 1.8V and 5.5V to feed its internal DC-DC regulator, plus an I/O power supply (VDDIO) at 1.8V nom. A 'common rail' approach may be suitable, with both supplies at 1.8V. In any case, pay attention that any voltage conversion between battery and GAP9 supplies (e.g. to go from 3.8V nominal Li-Ion voltage to 1.8V) had better be performed with high efficiency (incl. at very light loads down to a few 100 uAs), to obtain the best battery life at system level. Also, if sharing 1.8V with sensitive mixed signal parts or sensors, pay attention to the acceptable level of power supply noise.

(*10) Test Pads for Tuning – This is to provide real-time access to internal data as this may be required to tune some specific algorithms (e.g., ANC) in a prototype or product with (almost) final form factor. Interfaces of choice to export this data (which may require significant bandwidth) would be SPI or UART (preferably with flow control). This interface might be shared with BT SOC <> GAP9 control channel, with some precautions and system level checks.

(*11) Battery management – A fuel gauge and a battery charging block are typically required. These blocks are often included in the BT SOC, if they are not, external IC would typically be required. A battery protection IC specific to the type of battery being used (e.g., Li-Ion coin cell) may be required, sometimes this protection is already included in the battery pack (e.g., in most LiPo batteries)

(*12) ssm6515 interfacing – The SSM6515 (assuming this is the selected Class-D amp, although other options are possible) can be fed with either PDM data or PCM data (through I2S or TDM protocol in the latter case). It is anyway recommended to connect SAIx_SCK, SAIx_WS and SAIx_SDO from GAP9 to the SSM6515 as this provides support at bo extra cost for both PDM and I2S/TDM modes -- to support PDM only, SAIx_WS could be dropped but it is better to keep the flexibility of dual-mode support. PCM mode supports up to 768KHz audio sampling rate while PDM mode supports PDM clock up to 12.288MHz. PDM at 12.288MHz will provide better input-to-ouput delay (whcih is good for ANC performance in particular), however, in some use cases, other options might be preferable

2.3. VARIANT – WITH AUDIO CODEC RUNNING ON GAP9

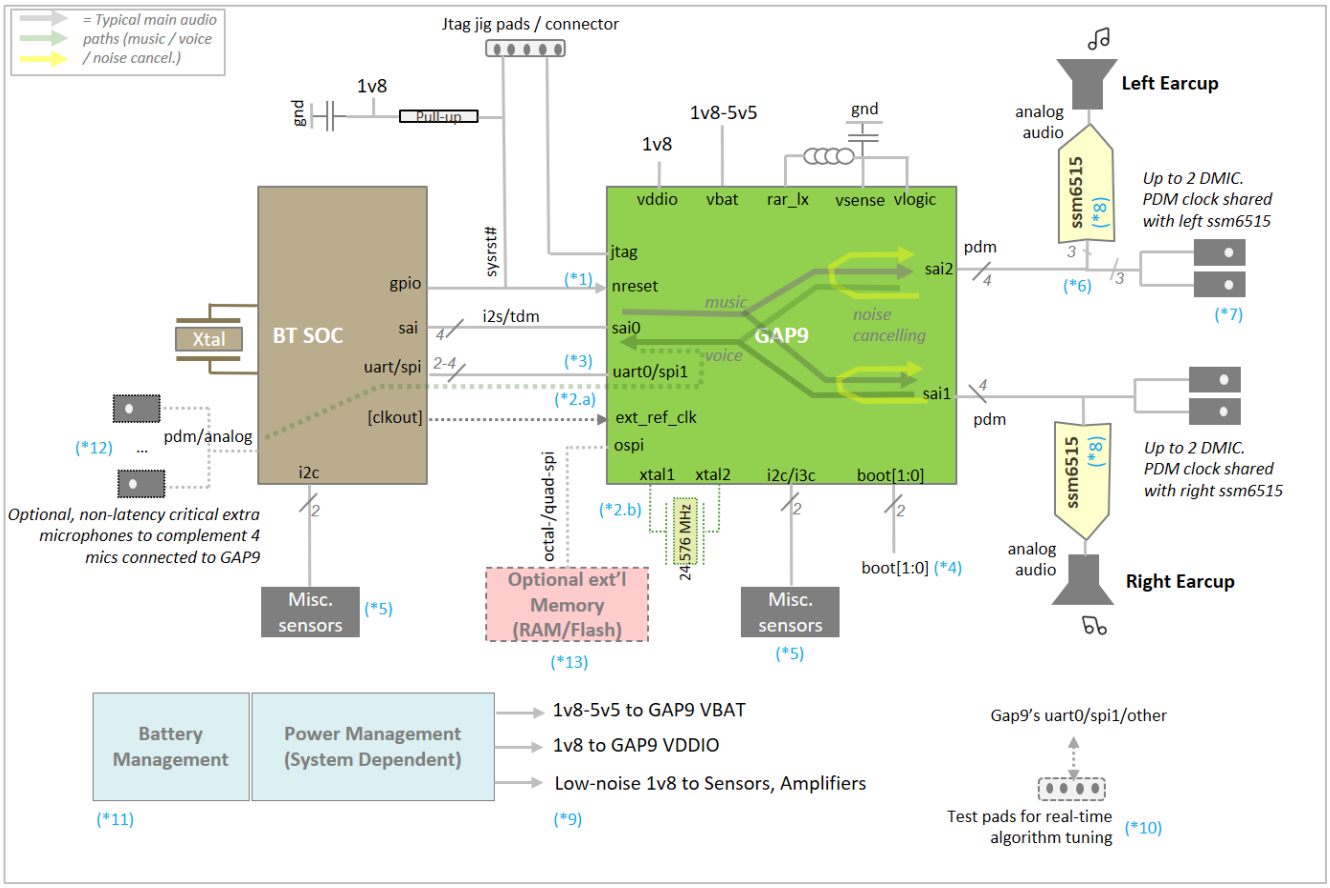
The above scenario considers the audio codec (SBC, AAC…) is executed inside the BT SOC and uncompressed audio streams are exchanged between BT-SOC and GAP9. Nevertheless, if so wished, it is also possible to have GAP9 running the audio codec – see note in 'Introduction' chapter.

In this case, compressed audio packets (typically, ACL or SCO packets), rather than uncompressed audio bit streams, are exchanged between the BT-SOC and GAP9. As a result, the Serial Audio Interface dedicated to BT-SOC<>GAP9 uncompressed audio bit stream in the former figure (SAI0) is no longer required; compressed audio data packets are typically exchanged over SPI or UART, using the Bluetooth HCI (Host Controller Interface) protocol.

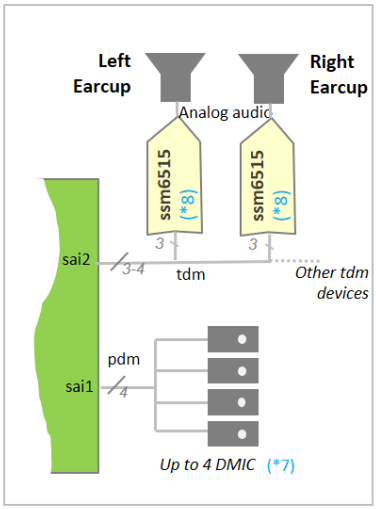
The impact on SAI assignment is the same as that described and graphically illustrated in Section 4 ('Headphones Architecture, Single GAP9, Audio Codec on GAP9') further down this document.

3. EXAMPLE HEADPHONES ARCHITECTURE, SINGLE GAP9

3.1. BLOCK DIAGRAM



Notes numbered « *(*xx)* » in this figure provide additional explainations and implementation guidelines. Please refer to the following section.



ALTERNATIVE AUDIO OUT ARRANGEMENT

(*14)

3.2. IMPORTANT NOTES

*- relative to the aspects denoted by « (*xx) » within the former figure.*

(*1) to (*5) – Same reamrks as in TWS case

(*6) PDM sharing by DMIC and DAC/Amp ; Impact on SSM6515 latency – GAP9 cannot provide 2 PDM *outputs* on a single SAI interface. Therefore the 2 SSM6515 (or other DAC/Amp solution), when configured to receive PDM, must connect to 2 different SAIs. When doing so, it is possible to share each SAI interface between one SSM6515 and 2 digital microphones, whereby they all share the same PDM clock and SSM6515 uses one data line while the microphones use the other data line (see '*GAP9 h/w integration guide*', section 9). Note however that, because the maximum PDM clock frequency of digital microphone is typically 4+MHz, an SSM6515 sharing the same clock will not be operable at its maximum frequency (12.288MHz) but rather at the classic 3.072MHz – translating as 4.5us to 7.5us increase in input-output delay of the SSM6515, see SSM6515 datasheet, Table 1.

(*7) to (*11) – Same remarks as in TWS case

(*12) Extra voice mics – In the proposed architecture, 4 microphones can be accomodated on GAP9's SAI1 and SAI2 (with SAI0 dedicated to BT SOC<>GAP9 interfacing). To accomodate more (voice) microphones, one option is to attach them to the BT SOC, which may forward the captured data (possibly after some pre-processing) to GAP9 in TDM format (provided TDM – or multi-channel I2S – is supported by BT SOC of course), multiplexing it with the regular audio stream.

Although perfectly feasible, one must be aware, however, that adding microphones on the BT SOC side in this architecture also adds some complexity to the system architecture, especially on the software side, in particular to manage timings between the audio front-end of the BT SOC and GAP9.

Also, in that scheme, latency from these microphones to GAP9 will be significantly longer than latency from those directly attached to GAP9 (on SAI1 or SAI2); therefore, latency critical mics, such as feed-forward and feedback mics for ANC, should be those connected to GAP9, while extra mics connecting to BT SOC should be less latency critical voice mics. Note also the extra voice mics connecting to BT SOC must not create too much traffic on the TDM interface (else that interface won't offer enough bandwidth to accomodate all required traffic); 20 or 24 bits at 16Khz can be a suitable trade-off (system dependent, do your own calculations).

An alternative solution, in the same spirit, is to connect the extra digital mics to a PDM-to-TDM concentrator IC and then, again, share as required the TDM interface on SAI0 (provided the BT SOC properly implements TDM and tri-states on slots that do not belong to it). It might also possible to share SAI1 or SAI2 with an SSM6515 if the latter is operating in PCM input mode (rather than PDM), see note (*14).

(*13) Optional external memory – While most audio algorithms can run satisfactorily using only GAP9's embedded RAM (1.5MByte) and eMRAM (2MByte, non-volatile), more advanced ones may benefit from extra memory space. To that aim, it is possible to add external RAM and/or Flash attached to GAP9 through an octal-SPI interface (or quad-SPI if any interest). The external memory option is not envisaged on the TWS example architectue as space is so limited in that case; the situation is a bit less tight in headphones.

(*14) Alternative audio output arrangement – An alternative arrangement to that described in (*6) can be to use the two SSM6515 in TDM mode and have them share the SAI1 interface (possibly with some other TDM devices, too), with up to 4 digital microphones sharing SAI2 in PDM mode. However, pay attention to the fact that bit clock of SSM6515 in TDM mode is limited to 24.576MHz, therefore having 2 SMM6515 on one SAI requires the SSM6515 to operate at no more than 368KHz if sample bit depth is greater than 16 bits, or reduce bit depth to 16 bits to run each SSM6515 at 768KHz audio sampling rate. Of course, sharing this SAI with more devices will further reduce the bandwidth available to each SSM6515. Therefore, this audio ouput arrangement may beneficial in some situations (as it saves a PCM-to-PDM modulation stage in GAP9 SFU and enables plugging in additional devices) but is not an option when highest quality audio and/or ultra-low latency is required (typ. for ANC) from the two SSM6515, for the reasons just listed.

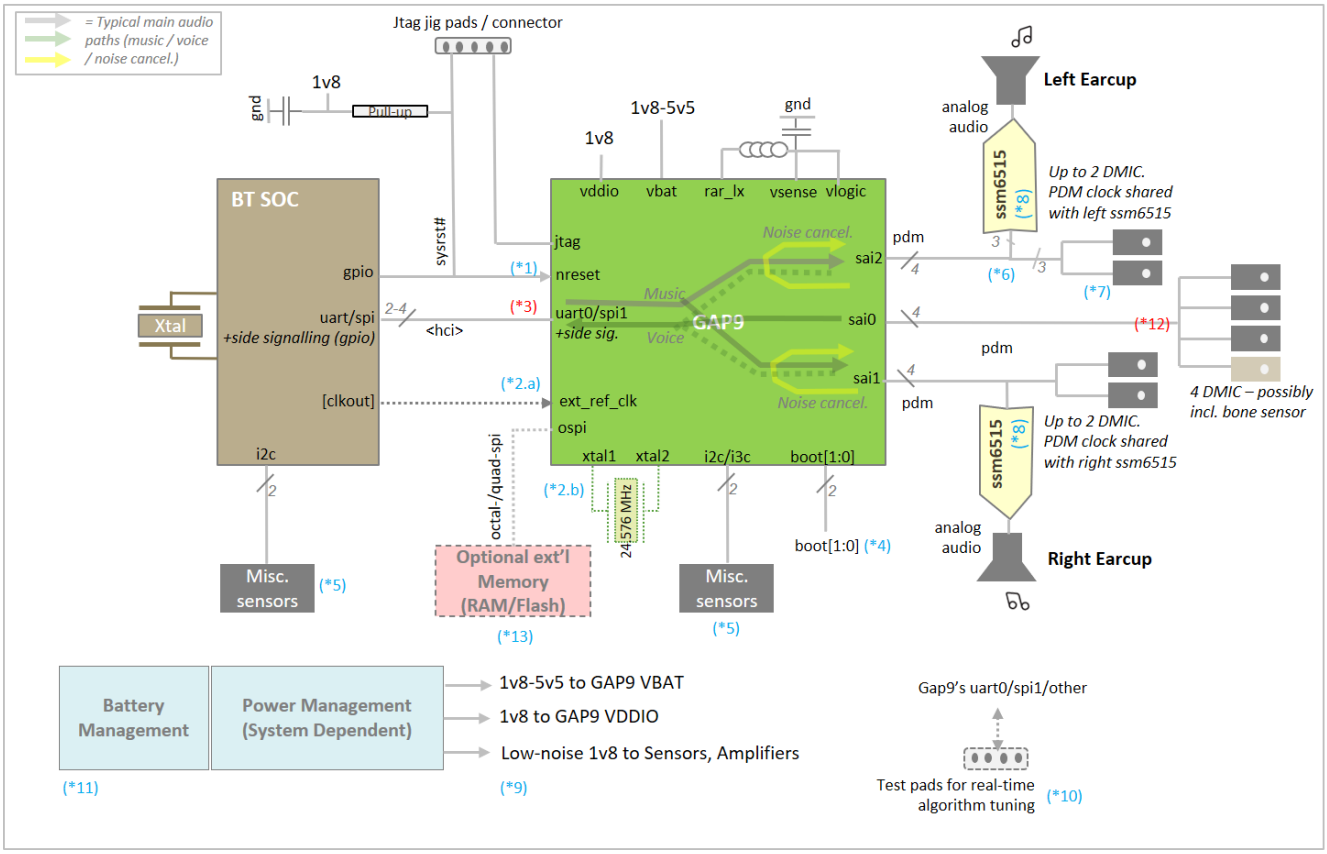
4. EXAMPLE HEADPHONES ARCHITECTURE, SINGLE GAP9, AUDIO CODEC ON GAP9

4.1. BLOCK DIAGRAM

In this scenario, instead of having the audio codec (SBC, AAC, LC3…) executed inside the BT SOC, we have the audio codec running on GAP9 side – see note in 'Introduction' chapter.

In this case, compressed audio packets (typically, ACL or SCO packets) are exchanged between the BT-SOC and GAP9, rather than uncompressed audio bit streams. This also means one SAI is freed up (see notes below) and can be used to attach more digital microphones to GAP9.

Below is an example harwdare architecture geared towards that scenario.



Notes numbered « *(*xx)* » in this figure provide additional explainations and implementation guidelines. Please refer to the following section.

4.2. IMPORTANT NOTES

*- relative to the aspects denoted by « (*xx) » within the former figure.*

(*1) and (*2) – Same as TWS and single GAP9 Headphones cases

(*3) BT SOC <> GAP9 Audio interfacing through HCI (compressed audio packets over SPI or UART) – In this scenario, compressed audio packets are exchanged between the BT SOC and GAP9, typically using the HCI (Host Controller Interface) defined in the Bluetooth standard. The transport interface can be either UART or SPI (possibly with some side signaling, e.g. for an SPI slave to tell the master it has some data ready to be pulled). GAP9 receives compressed audio data packets and must implement the audio codec (e.g. LC3, AAC…) on its side. There are also some Bluetooth synchronization implications that must be dealt with at software level.

(*4) to (*11) – Same as single GAP9 Headphones cases

(*12) Four extra (voice) mics – Since the BT SOC now exchanges audio data through SPI or UART (HCI interface), one serial audio interface (SAI0 on figure) of GAP9 is freed up and can be used to connect up to 4 PDM microphones, with one common clock and 2 data lines multiplexing 2 mics each, as detailed in the *GAP9 h/w Integration Guide*.

One might also want to connect a bone sensor (a.k.a voice vibration sensor) instead of one of the extra mics, provided it presents a compatible PDM interface– with high enough PDM clock frequency to share with regular DMICs (not always the case).

(*13) and (*14) – Same as single GAP9 Headphones cases

5. EXAMPLE HEADPHONES ARCHITECTURE, DUAL GAP9, SYMMETRICAL

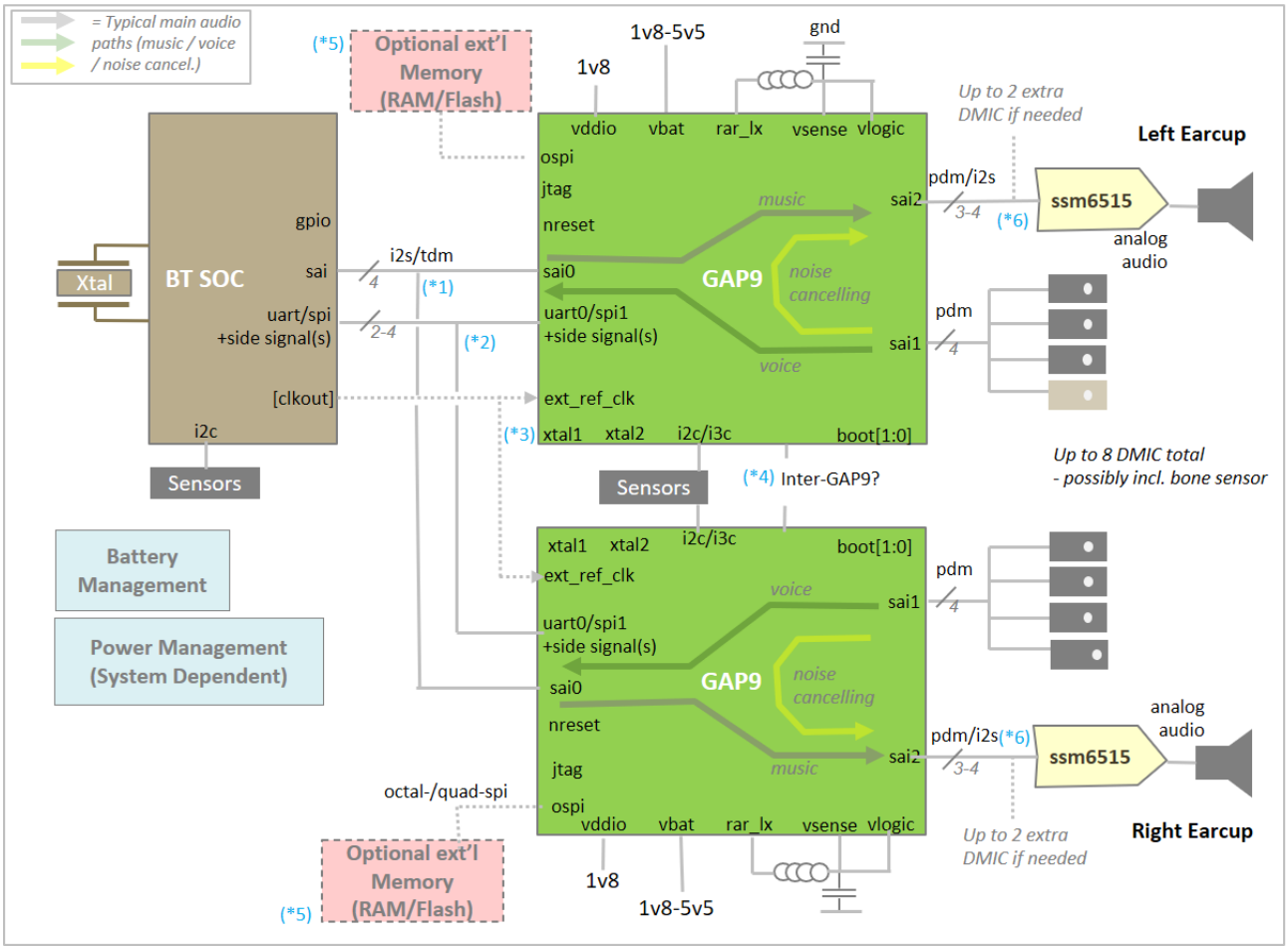
5.1. BLOCK DIAGRAM

This dual-GAP architecture enhances the single-GAP9 headphones architecture to, in particular, support more microphones directly attached to GAP9.

Please check the single GAP9 architecture first, including implementation notes, most of which are still relevant but not repeated here.

Connections to some pins such as BOOT, JTAG or NRESET, explicited in former architecture diagrams, are not detailed again here. One additional requirement is that some now need to be shared between the two GAP9 ; different solutions are envisageable and left to the reader to design.

Notes numbered « *(*xx)* » in this figure provide additional explainations and implementation guidelines. Please refer to the following section.



5.2. IMPORTANT NOTES

*- relative to the aspects denoted by « (*xx) » within the former figure.*

(*1) Audio interface between BT SOC and multiple GAP9 – BT SOC sends stero, received by both 'Left' and 'Right' GAP9. If BT SOC supports TDM, then the interface can be used in TDM mode and the GAP9's can send their data to BT SOC on 2 different time slots. If BT SOC only supports I2S, GAP9 can still be configured for 2-channel TDM mode so that 'Left' GAP9 provides its data on the first time slot and tri-states on the second time slot, while the 'Right' GAP9 provides its data on the second time slot and tri-states on the first time slot.

(*2) Control / Auxiliary Data channel between BT SOC and GAP9(s) – Different strategies are possible to have the BT SOC communicating with the GAP9's :

- One option is to have the BT SOC Communicating with only one GAP9, which would act as a relay to the 'Right' GAP9 (see *4).

- Another option is to have the BT SOC on one end of a UART or SPI interface and both GAP9 on the other side, sharing the interface. This obviously requires some precautions to make sure both GAP9 are not driving a line at the same time. Different technical solutions can be envisaged; provisioning a Chip Select (-like) signal (natively part of SPI interface / can be emulated with a GPIO and some software in case of UART interface) may help manage this.

(*3) GAP9 reference clocks – Similarly to the single GAP9 cases (see note *2 in those scenarios), the most cost- and area-efficient approach is to have the BT SOC delivering a suitable reference clock to both GAP9's, if it is able to do so.

Otherwise, either fit each GAP9 with a suitable crystal (but note, in this case, this means the GAP9's will be asynchronous, which might have some implications if GAP9-to-GAP9 communication is required), or use a single dedicated oscillator chip (such as SiTime's SiT8021) feeding both GAP9, which will be more optimized area-wise and, possibly, cost-wise.

(*4) Inter-GAP9 communication – if required – Depending on system choices, inter-GAP9 communication may or may not be required, besides simple GPIO based flags. If it is definitely required (adding some more complexity to the sytem) :

- If low bit rate is required, I2C may be sufficient. One GAP9 could be I2C Master and the other could be I2C Slave.

- For higher bit rates, UART or SPI may be relevant. UART without flow control can support of the order of 100Kbps in general (and up to 3Mbit/s for fixed lengths bursts) and over 3 Mbit/s with flow control, while SPI may run up to a few 10s of Mbit/s (instananeous bit rate). UART0 is available if not already used for BT SOC/GAP9 communication, else another UART can be remapped to specific pins to be made available for inter-GAP communication. SPI1 is available if not already used for BT/GAP9 SOC communication, one GAP9 would be master and the other would be slave. Else SPI1 could possibly be shared, with one GAP9 acting as Master and the other GAP9 as well as BT SOC acting as slaves, addressing a specific slave through asserting the right Chip Select pin. A GPIO/IRQ line alongside the SPI bus may be useful so the Slave can tell the Master it has some data to send.

- Finally, if a more audio-oriented interface is required between both GAP9's, SAI0 could possibly be used both as BT-SOC<>GAP9 audio interface and as inter-GAP9 interface, provided it is configured in TDM Mode (assuming BT SOC supports this mode). Then it is a matter of assigning different slots to different usages. Pay attention, though, to the fact that total bandwidth over SAI0 will be limited by the maximum bit clock frequency acceptable to GAP9 (normally equal to the reference clock frequency, for instance 24.576 MHz – unless FLL are involved, which is not recommended

for audio clocks) and to the BT-SOC. For instance, if achievable bit clock frequency is say 12.288MHz and BT SOC <> GAP9 audio interface is 32-bit stereo at 48KHz (which amounts to 3.072Mbps), then 9.216Mbit/s could be allocated to inter-GAP communication, suitable for, *e.g.*, a stereo stream at 192KHz, 24-bit per sample – but not higher bit rate at same bit depth nor higher bit depth at same audio sampling rate.

(*5) Optional external memories – For the most advanced audio algorithms, additional external memory (RAM and/or Flash) may be useful (see note *12 in single GAP9 headphones architecture). It may or may not be doable to have only one of the two GAP9 fitted with external memory, depending on software partitioning.

(*6) SSM6515 interfacing – In this setup, when minimum input-to-output delay is desirable (typ. for ANC), the SSM6515's can be operated at their maximum PDM clock of 12.288MHz.
However, in the case where the total of 8 DMICs would still not be sufficient and the SSM6515's SAI interface would have to be shared with (up to 2) extra DMICs, then the frequency of the PDM clock, common to mics and SSM6515, would likely be limited by the microphones, whose clock will in general not be able to exceed 4MHz or so.

6. EXAMPLE HEADPHONES ARCHITECTURE, DUAL GAP9, ASYMMETRICAL

6.1. BLOCK DIAGRAM

Compared to the former symmetrical dual-GAP9 architecture, this asymmetrical architecture brings the following :

- enables having 8 DMIC plus one free SAI available for more audio devices

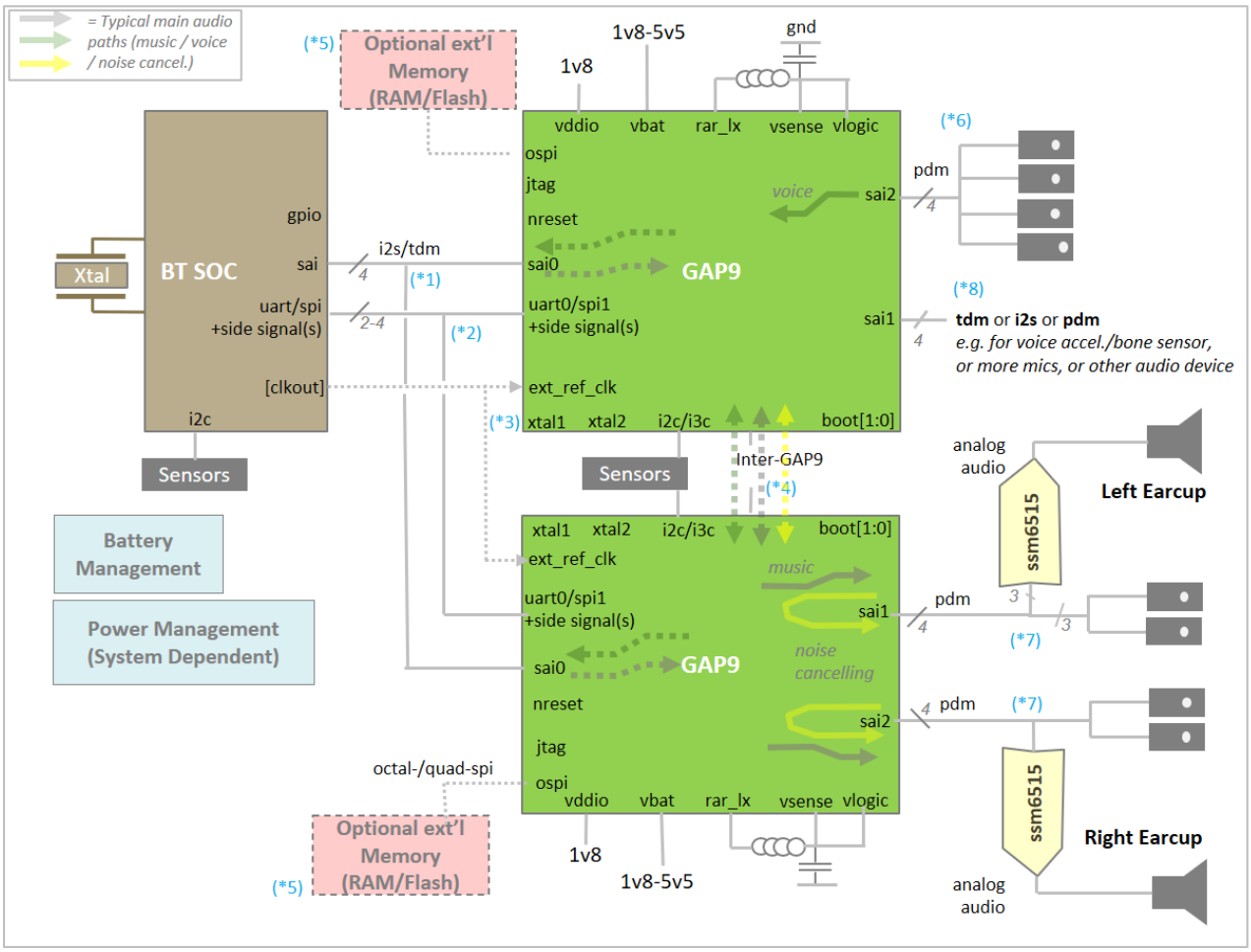
- enables different software partioning and optimisations,

- as a result, in use cases where external memory is useful, may make it easier to have only one GAP9 fitted with external memory

- is a superset of single GAP9 headphones architecture, so one may envision a low-end and a high-end platform with many commonalities.

Possible drawbacks include the unability to operate the SSM6515 at their lowest possible latency (see notes) and the more likely need to provision moderate-to-high bandwidth communication between both GAP9, somewhat complexifying the required software.

This asymmetrical dual-GAP9 architecture is a variant of the symmetical dual-GAP9, which itself enhanced the single-GAP9 headphones architecture. Therefore, **please check the single GAP9 and symmetrical dual-GAP9 architectures first, including implementation notes, most of which are still relevant but not repeated here.**



Notes numbered « *(*xx)* » in this figure provide additional explainations and implementation guidelines. Please refer to the following section.

6.2. IMPORTANT NOTES

(*1) to (*5) – same as for symmetrical dual GAP9 headphones architecture

(*6) DMICS – The digital mics (PDM ouput) typically connect to the main PCB through wires/FFC of some length. It is recommended to provision termination resistors (e.g. 10-20 ohm to start with) on clock pin (GAP9 side) and data pins (mic side) to limit reflections and EMI. Note also these mics typically go to deep sleep when no clock is applied, so a power switch on their power supply is not necessarily required. See *GAP9 h/w Integration Guide* for details on multiple DMICs sharing one SAI.

Microphones intended as feed-forward and feed-back microphones for ANC should be carefully selected, especially regarding SNR, AOP and group delay. State of the art digital microphones offer a group delay no greater than 7-8us at 1KHz.

(*7) PDM sharing by DMIC and DAC/Amp – Impact on SSM6515 latency – GAP9 cannot provide 2 PDM *outputs* on a single SAI interface. Therefore the 2 SSM6515 (or other DAC/Amp solution), when configured to receive PDM, must connect to 2 different SAIs. When doing so, it is possible to share each SAI interface between one SSM6515 and 2 digital microphones, whereby they all share the same PDM clock and SSM6515 uses one data line while the microphones use the other data line (see GAP9 h/w integration guide, section 9). Note however that, because the maximum PDM clock frequency of digital microphone is typically 4MHz or so, the SSM6515 sharing the same clock will not be operable at its maximum frequency (12.288MHz) but rather at the classic 3.072MHz – translating as 4.5us to 7.5us increase in input-output delay of the SSM6515, see datasheet Table 1.

(*8) Free SAI interface – With this arrangement of microphones and DAC/Amplifier, up to 8 digital microphones can be fitted and one SAI interface is still free (SAI1 of the first GAP9 in the former figure). This can be used for a variety of purposes, such as:

- connecting a Voice Accelerometer with a TDM interface

- connecting auxiliary Audio Devices – typ. through an I2S or TDM interface

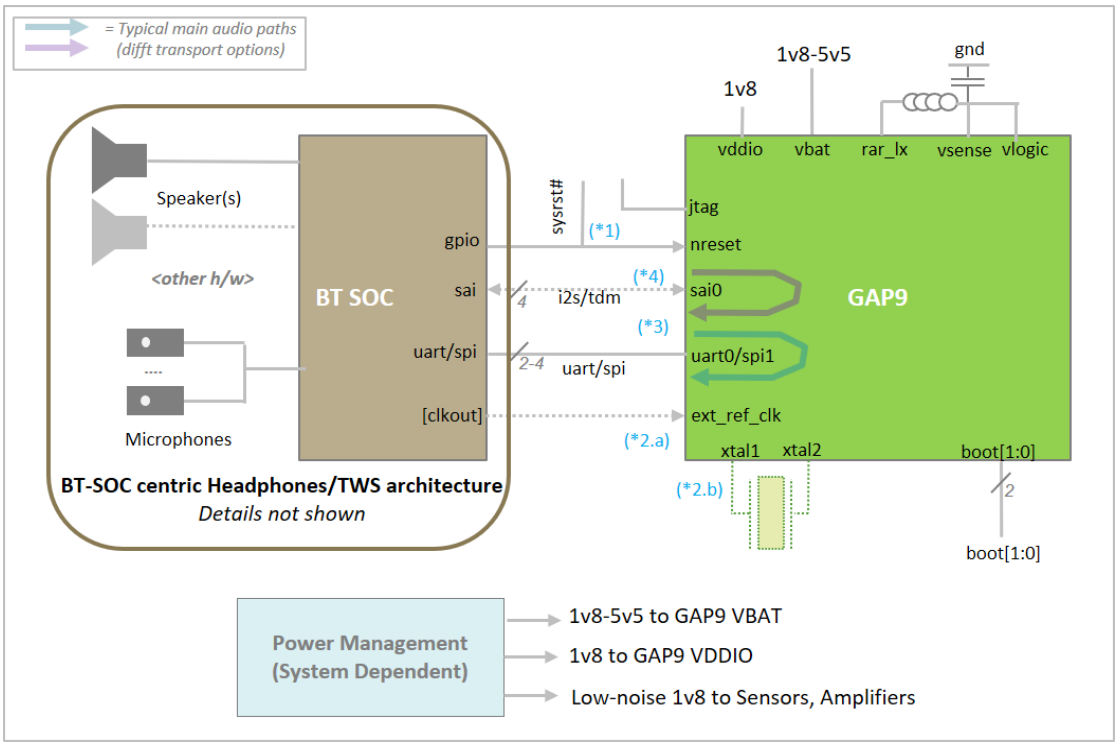
- connecting even more PDM microphones or bone sensor(s)

etc.

7. GAP9 AS PURE CO-PROCESSOR

When very low processing latency is not required (which in particular discards high performance ANC), it is also enviseagable to enhance a classic BT-SOC based TWS or Headphones architecture with GAP9 used as pure co-processor,. Heavy DSP and/or Neural Network tasks would be offloaded to GAP9 to reach performance the BT SOC on its own would never reach.

7.1. BLOCK DIAGRAM



7.2. IMPORTANT NOTES

(*1) GAP9 reset control – For FOTA update, BT SOC would typically need to be able to reset GAP9. GAP9 hardware reset also needs to be controllable from JTAG probe. See '*GAP9 h/w integration guide', section 10.*

(*2) GAP9 reference clock – Here there are fewer constraints on respective BT SOC and GAP9 reference clocks. Further explainations can be found in Application Note *'Audio Clocks for GAP9 – Hardware Options and System Implications*'. In general, it should be possible to obtain a suitable reference clock from the BT-SOC (as auxiliary clock output or even PWMout signal), for instance equal to BT-SOC's own reference clock, this is option 2.a. Otherwise, an external crystal (2.b) or a dedicated clock oscillator (smaller, lower power, but a bit more costly) would be required.

(*3) BT SOC <> GAP9 Control / Auxiliary Data Channel – A channel for control information and non-audio data is likely to be required between BT SOC and GAP9, if only for FOTA code updates. It can be implemented over UART (with optional flow control) or, if no uart resource is available, over SPI (GAP9 preferably Master, but Slave mode is supported too at the expense of sligthly more complex firmware). A GPIO/IRQ line alongside the SPI bus may be useful so the Slave can tell the Master it has some data to send.

(*4) BT SOC <> GAP9 interface – When audio needs to be exchanged between GAP9 and the BT-SOC, TDM or I2S is an option of choice. However, with GAP9 used as coprocessor and essentially exchanging buffers of data with the BT-SOC, tight coupling between the data samples and a real time audio clock (bit clock, frame sync) may not be mandatory. In that case, an SPI or even UART (+flow control) link may be sufficient.

REFERENCES

[1] GAP9 Hardware Integration Guide –
GreenWaves Technologies web site : GAP9 Resources Restricted Access Web Page (NDA Customers only),
GAP9 Documentation > GAP9 IC Integration
URL at the time of writing : *greenwaves-technologies.com/product/gap9-resources/*

[2] Audio Clocks for GAP9 – H/W Options and System Implications –
GreenWaves Technologies web site : GAP9 Resources Restricted Restricted Access Web Page (NDA Customers only),
GAP9 Documentation > GAP9 IC Integration
URL at the time of writing : *greenwaves-technologies.com/product/gap9-resources/*

DOCUMENT HISTORY

Drafts A&B – *Oct. 2023 (Xavier Cauchy)*

Rel.1.0 – *3. Nov. 2023 (XC)*
Initial release