GAP9 EVALUATION KIT

(GAP9_EVK)

USER MANUAL

Rel.2.5
19. July, 2024

CONTENTS

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.

PRELIMINARY NOTE - *This User Manual is intended to complement the electrical schematic of the GAP9_EVK board. In case of discrepancy between this manual and the schematic, the latter should be trusted.*

KNOWN LIMITATIONS

(1) *GAP9_EVK v1.1/v1.2* - Reset button has no effect and system reset from external JTAG probe has no effect either However, system reset from USB<>JTAG (FTDI) bridge is functional. [A hardware hack is attemptable for cases where this is a real problem (e.g. if external JTAG probe is a must)].

(2) *GAP9_EVK v1.1/v1.2, till 16.Mar.22* - The FXL6408 GPIO expander may not interface properly with I2C3. This can be fixed with a small hardware hack. See section 12. Fixed on GAP9_EVK shipped after16-mar-22

(3) *GAP9_EVK v1.1/v1.2* - (a) *till 20.Mar.22* – pull of BOOT0 to Logic-1 through J9 may require a work-around. Fixed on GAP9_EVK shipped after 20-mar-22 - (b) V_MIKROE must no be set to 1V8 at boot (through J8) for correct control of BOOT1. See Section 7.

(4) On some boards, when powered from USB, there is a sensitivity that may cause a disconnection of power suply into the EVK when the ambient temperature increases – even moderately in some cases. This can be fixed by simply connecting the external power supply input VEXT (unused in that case) to ground, which is readily done by shunting the jumper between (VEXT) « + » and GND in the top right corner.

1. INTRODUCTION & QUICK START GUIDE

GAP9_EVK is an Evaluation and Development Kit for GreenWaves Technologies’s GAP9 chip, an ultra-low power application processor for Edge AI and Digital Signal Processing, delivering an outstanding performance/power ratio.

GAP9_EVK is intended for generic evaluation of GAP9 capabilities. More application-focused developments will typically require coupling GAP9_EVK with extra hardware, for instance some sensors, a low-power radio module or an audio board. Multiple expansion connectors are provisioned on GAP9_EVK to facilitate this.

To speed up evaluation and development of selected types of applications, GreenWaves or third-parties will provide off-the-shelf ‘add-on’ boards to plug directly into GAP9_EVK – including an Audio Add-On board to work on portable/personal audio use cases (a dedicated User Manual is available for this Audio Add-On board).

IMPORTANT NOTE : Despite the protections provisioned on this Eval Kit, its electronics being deliberately made very accessible, it is, more than a final product, sensitive to ESD (Electro-Static Discharge) induced damage and mechanical stress. Therefore: avoid pressing or pushing parts and, when you need to manipulate the boards,

wear an anti-static wristband and/or anti-static gloves – or as a minimum, first touch an element at earth potential (for instance, the earth terminal of a mains plug) to discharge yourself.

> Quick Start :

Refer to Fig.6 in Section 6 for pictures of jumper location and polarities.

(1) If not already done, gently insert the GAP9Mod into the M.2 format receptacle of GAP9_EVK., with a 20° tilt (details and picture at the end of Section 4). Then push GAP9Mod to bring it into a horizontal position and fasten it to GAP9_EVK's PCB with a screw at the side of GAP9Mod opposite to the M.2 connector.

(2) Close jumper **J4** with a 2.54mm jumper

(3) Short pins 1-2 or 2-3 of **J5** with a 1.27mm jumper (on-board 1.8V generated by buck DC-DC converter in former case, by linear LDO converter in latter case)

(4) Short pins 1-2 of **J6** with a jumper (power supply into GAP9's internal DC-DC) when J6 is a 3-pin header. Else (from *GAP9_EVK v2.1 to v3.1)*, just make sure J6 is closed.

(5) Set jumpers **J9** and **J10** according to your booting needs. See section 7 for details. To boot from JTAG : on GAP9_EVK v1.x, that would be : shunt J9 positions 1-2 and J10 positions 2-3 ; on GAP9_EVK v2.x, just leave those jumpers open.

(6) *only if your GAP9Mod is v2.0* (check overlay marking, also v2.0 features two 1.27mm pitch jumpers J1 & J2 rather than just J1 as v1.x) : if you intend to use the CSI-2 interface, close jumper J2 to power this interface – else keep it open

(7) Connect USB cable. A red LED (LED-1) will turn on when input power is present.

(7.b) *USB 3.0 onwards only :* switch general power on using slider switch on right side of board.

(8) Program the board through USB (default) or through external JTAG probe.
Note : by default (with the exception of a few early v1.1 models), GAP9_EVK is configured to use the on-board USB<>JTAG bridge (FTDI chip) rather than an external probe. This can be changed by swapping resistors RR1 and RR2.

This is a minimal, default configuration to be able to boot from JTAG a program executing on GAP9, with USB-delivered power. Additional or alternative jumper settings are required to use expansion boards or to optimize voltage conversion efficiency according to your use cases. Refer to the next chapters for full information (esp. Chapter 6, *Configuration Jumper Settings)*.

2. GENERAL ARCHITECTURE

Figure 1 describes the main functionalities of GAP9_EVK.

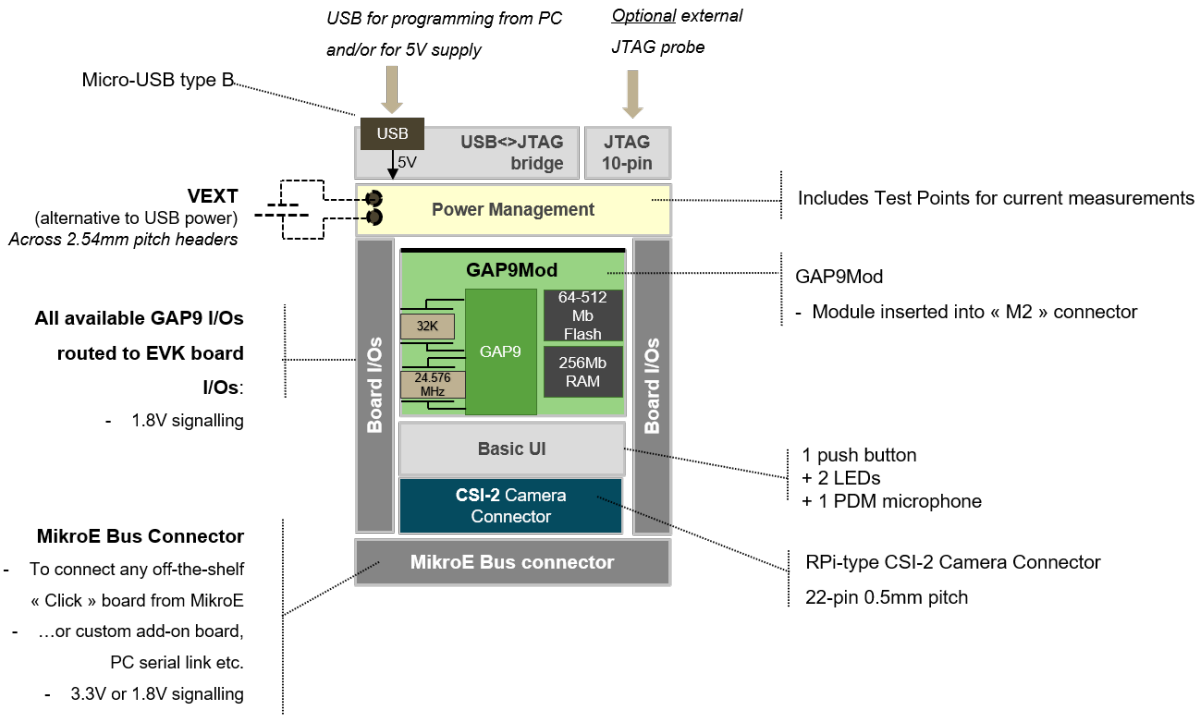


Fig. 1 – GAP9_EVK Contents

Dedicated sections further down the document provide more details. In summary :

- Power can be applied either through the USB port or through a dedicated terminal. All on-board power supplies are derived from this primary input. Means to measure current at different strategic points are provided.

- The heart of GAP9_EVK is the GAP9Mod core module, which is a system-on-module including the GAP9 chip, external Flash and RAM (to complement internal memory) and clock generation (crystals). This module presents a card edge connector compliant with the « M.2 » standard (physical specification only) to plug into a compatible receptacle on GAP9_EVK.

- Basic User Interface features are provided in the form of a general-purpose push-button and two LEDs, accessible from GAP9 through GPIOs, plus a microphone. The microphone is a digital PDM microphone that can be used not only for regular recording, but also as an ultra-low power wake-up source triggered on sound detection. It is intended for relatively basic audio applications, more sophisticated evaluations or demos would typically rely on the 'Audio Add-On board' for GAP9_EVK (sold separately).

- The CSI-2 MIPI (C-PHY) interface of GAP9 (only single lane supported by GAP9 in WLCSP package) is routed to a 15 pin FFC (Flat Flexible Cable) connector compatible with Raspberry Pi camera modules, to allow starting evaluation with readily available, off-the-shelf cameras before switching to more custom solutions.

- All free I/Os of GAP9 are made accessible on header pins, FCC connectors or Plated Through Holes (PTH). These expansion connectors may for example be used to connect satellite boards – either customer-designed, or provided by GreenWaves Technologies or by 3rd party vendors. In particular, GreenWaves Technologies will provide an « Audio Add-On Board » to complement GAP9_EVK for advanced audio-related sytems.

- Besides, a mikroBUS™ socket is provided, to interface with MikroE '*Click*' boards which offer a wide spectrum of functionalities. On-board level translation takes care of adapting signal levels to the 0-3.3V signaling employed by

mikroBUS. However, it is also possible to use 1.8V signaling to keep flexibiliy when connecting something else than a MikroE *Click* board to that socket.

3. ANATOMY OF GAP9_EVK BOARD

Figure 2 shows the main physical interfaces present on GAP9_EVK. Board dimensions(excluding protuding square tabs for mouting holes) are X=63mm, Y=68mm.

The board also comprises a number of configuration jumpers, these are documented in a specific section further down this document. Mapping of the expansion ports is also the subject of a specific focus in a next section (see ToC).

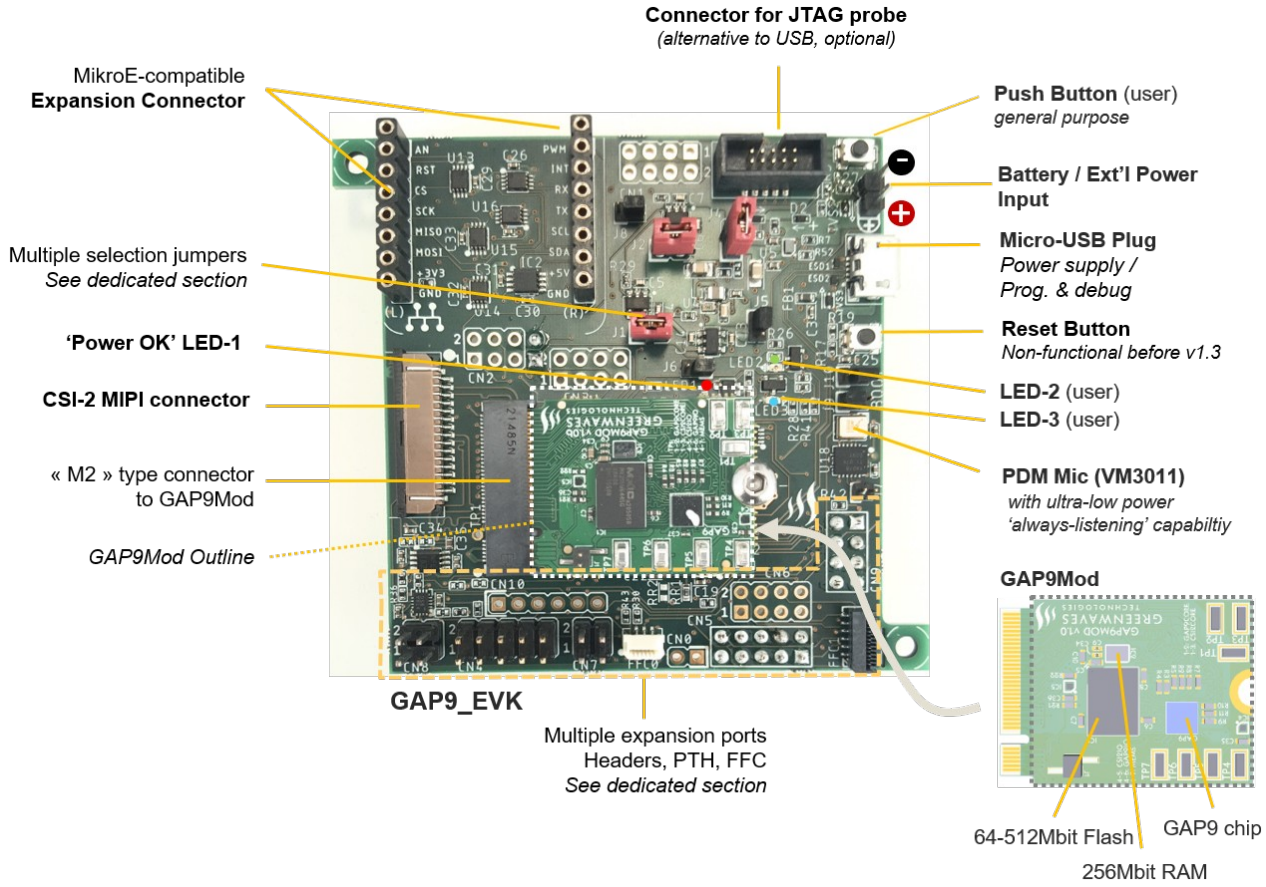


Fig. 2.a – GAP9_EVK Board Overview

Fig.2.b– GAP9_EVK 3.0 onwards : USB-C + USB power on/off switch(Top right corner)

4. GAP9Mod CORE MODULE & 'M.2' CONNECTOR

> GAP9Mod Contents

GAP9_EVK makes use of 'GAP9Mod' (**GAP9** Core **Mod**ule), a system-on-module that implements on a small PCB (22x30mm, double-sided) the mostly 'invariant' part of any application based on GAP9 – which also happens to be, in many use cases, a critical part of the overall design in terms of precautions to take regarding PCB design and layout.

Using GAP9Mod as a reusable, central module when designing GAP9-based prototype or PoC (Proof of Concept) boards helps reduce risks and secure short design cycle time.

GAP9Mod comprises the following elements (see Fig. 3.a and 3.b – components present on both side, only top side shown here).

- GreenWaves Technologies's **GAP9 chip** (in its initial WLCSP100 package version)

- an **Octal SPI NOR Flash** ('Xcella' standard): **64Mbit** Macronix MX25UW6445G initially, **512Mbit** Macronix MX25UW12845G for GAP9Mod supplied after 20.Apr.22 *[possible exceptions, subject to component availability]*

- a **256Mbit Octal SPI RAM** (*2) : APMemory APS256XXN-OBRx-WA *initially, could evolve in the future]*

- a **24.576MHz crystal** (optional use, usable to generate master reference clock) and load capacitances

- a **32.768KHz crystal** (optional use, usable as RTC/Watchdog reference clock) and load capacitances

- passives required to properly operate GAP9's internal DC-DC converter : 4.7uH inductor (*3) and various capacitors

- power supply bypass and decoupling capacitors

- sense resistors and test points for **current monitoring** at different strategic locations

- **mini-jumpers** to cut off or enable power supply (V1p8_QPS) for eFuse programming and, on GAPMod 2.x only, power supply (V0p65_CSI2) to CSI-2 interface of GAP9 – see section on jumpers further down this chapter.

These parts were selected to cover a wide spectrum of uses cases, with ample memory, optional on-board crystals, ability to handle heavy workloads and so on.

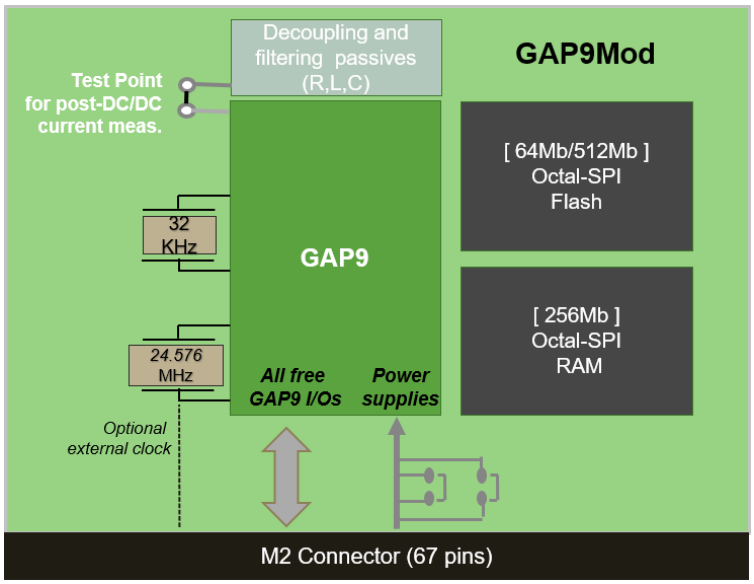


Fig. 3.a – GAP9Mod Overview

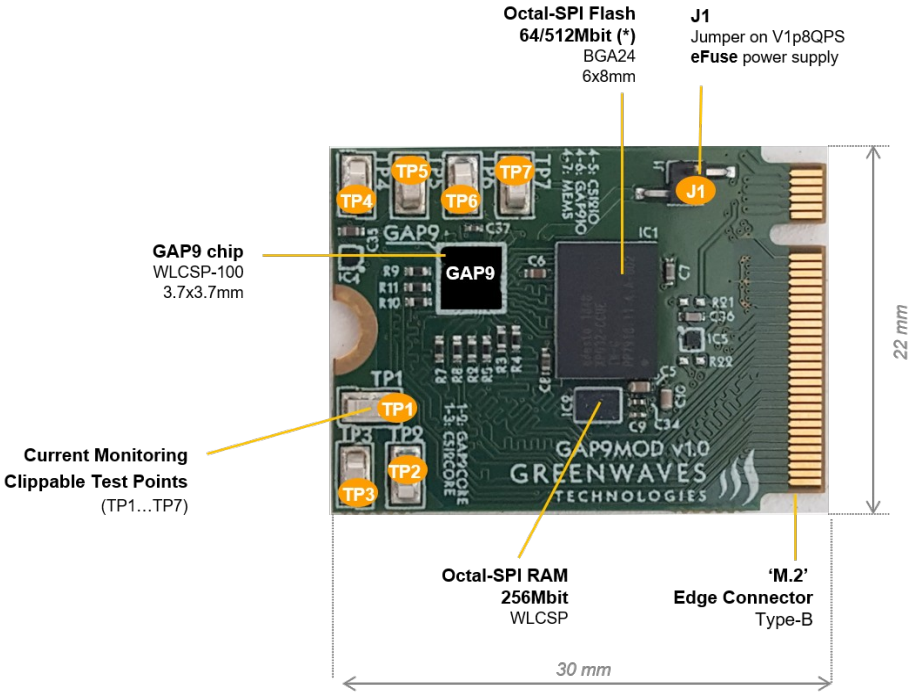


Fig. 3.b.1 – GAP9Mod v1.x PCB (Top side)

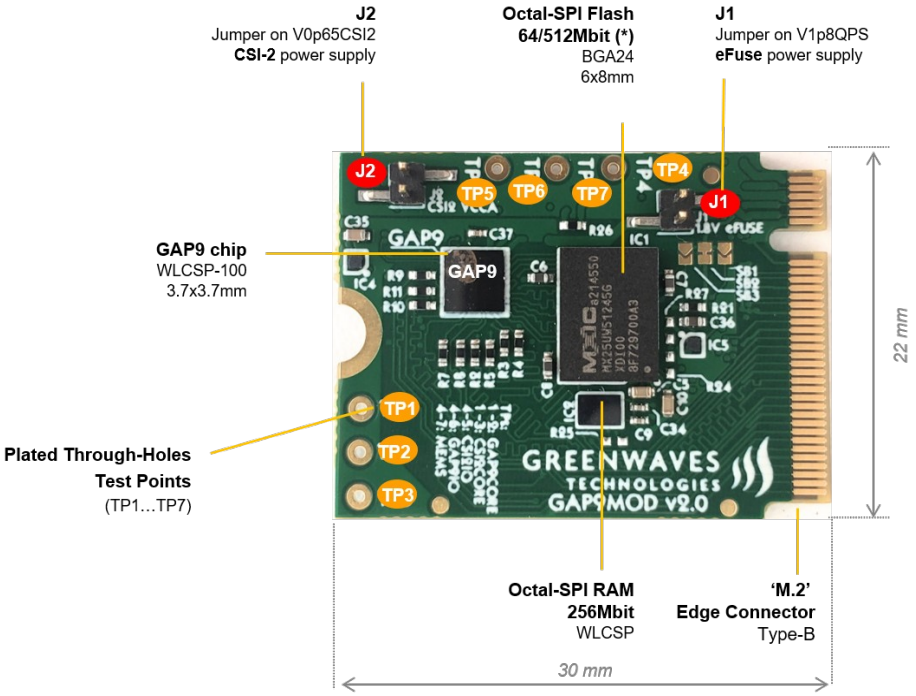


Fig. 3.b.2 – GAP9Mod v2.x PCB (Top side)

Notes :

(*1) External Flash (presence and capacity) :

The selected Flash offers ample bandwdith (Octal-SPI protocol selected here) and storage capacity – even though many applications may live with much less (or even, for very light ones, without any external Flash, relying on the 2MByte embedded non-volatile eMRAM only). Initial versions of GAP9Mod were fitted with a 64Mbit Octal-SPI Flash memory, Macronix **MX65UW6445G**, while versions provided after 20.Apr.22 are fitted with a **512Mbit** memory, Macronix MX25UW51245G (same family, only the capacity is different). Check marking on memory package if in doubt about MPN used.

The selected package is BGA24 6x8, which is somewhat bulky, but offers the advantage of being standard – which means it is possible, on the same GAP9Mod design, to swap the selected part for another one from a different vendor or with a different capacity.

(*2) External RAM (presence and capacity) :

The selected RAM (**APM APS256XXN-OBR-WA**) offers ample bandwdith (again, Octal-SPI protocol selected) and a very comfortable storage capacity (**256Mbit**) – most applications will require much less (or even just rely on the 1.5MByte embedded L2 RAM only). There is no opportunity for cross-vendor package compability (and limited opportunity for cross-capacity package compatiblity), therefore the selected package this time is a tiny WLCSP which hepls keep GAP9Mod area smalll.

(*3) Inductor for DC-DC output filtering (dimensioning) :

The output of GAP9's internal DC-DC converter needs to be filtered by an LC structure. The physical dimensions of the inductor (4.7uH effective) impact a number of key parameters. The part selected on GAP9Mod is relatively bulky to be on the safe side (selected prior to actual silicon characterization) and to cover a wide range of uses cases, be it in terms

of acceptable input voltages or maximum workload. A specific application for which those parameters are known and well defined could go for a more optimized choice.

> M.2 Connector

GAP9Mod plugs into GAP9_EVK through an 'M.2' card edge connector (Fig. 4).

The M.2 form factor (*a.k.a* 'NGFF', New Generation Form Factor) was orginally defined as an evolution of the PCI Express form factor specification, to support multiple function add-in cards /modules (WiFI, BlueTooth, NFC, SSD Devices, etc.) in space-constrained appliances such as laptops.

Here we are just relying on the physical aspects of an M.2 connector, disregarding any pin assignment and protocol aspects included in the PCI Express M.2 specification. The objective is to leverage the wide availability and low cost of M.2 receptacles in the market to implement a space-efficient and flexible solution for interfacing GAP9Mod and GAP9_EVK.

All GAP9 I/Os are made available to GAP9_EVK through that M2 connector – with the exception, obviously, of those which are consumed for interfacing GAP9 with other components present on GAP9Mod, such as the external memories for instance.

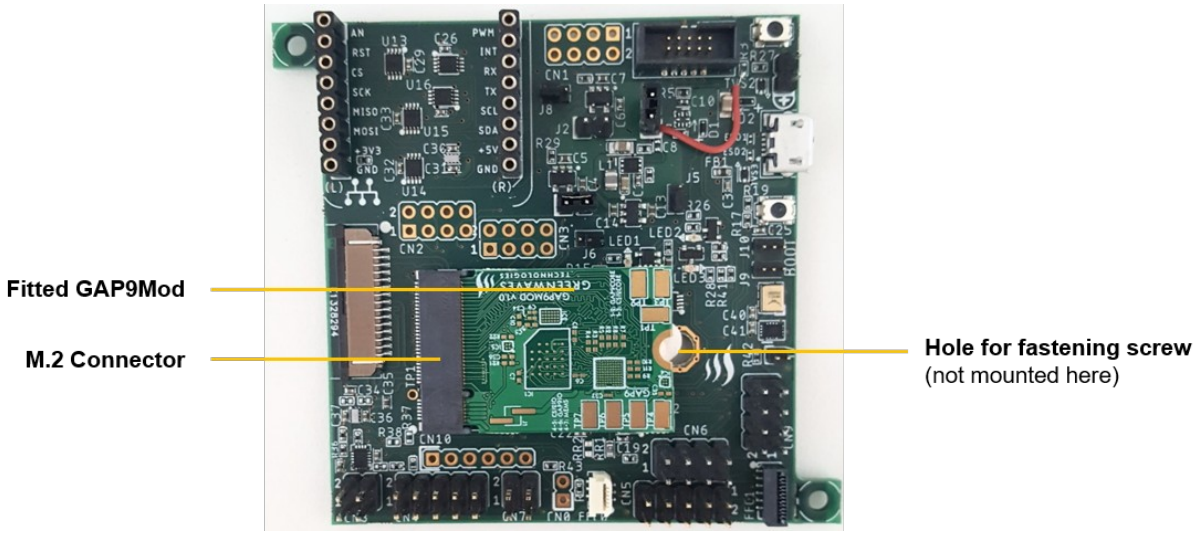


Fig. 4 – GAP9_EVK with fitted GAP9Mod *[components not yet assembled on GAP9Mod here]*

> Voltage / Current Monitoring Test Points

GAP9Mod is equipped with a number of **clippable test points (GAP9Mod v1.x)** or small **plated through-holes** (GAPMod 2.x), labeled **TP1 to TP7** on the boards, to enable measurement of current consumption by different functions.

This works as follows :

- a 0.5ohm resistor is implemented on GAP9Mod on power supplies dispatched to different functional blocks

- each terminal of the resistor connects to one of Test Points TP1 to TP7

- the voltage drop between these 2 Test Points (in mV) can then be measured in conveniently ; just multiply the obtained value by 2 to obtain the current (applying Ohm's law for a 0.5ohm resistor)

Refer to section 5, Power Supplies and Power Management for details on assignement of Test Points to each measurable power supply.

Note : The 0.5ohm value was chosen to be low enough to minimally affect the power supply voltage and current-dependent fluctutations. It is always possible to unsolder the 0.5ohm resistor (see schematic for part designator) and either replace it by some other value or use an amperemeter or other suitable measurement equipment between the 2 relevant test points. Pay attention to the voltage drop incurred by the measurement set-up, see Appendix E.

> Jumpers on GAPM9Mod

- Jumper J1 - Control of Fuse Programming Power Supply (V1p8QPS)

GAP9 embeds a number of non-volatile registers in the form of e-Fuse cells. These fuses can be programmed ('burnt') once for all to a Logic-0 or Logic-1 value. When programming an e-Fuse, it must be supplied with a 1.8V power supply. On GAP9Mod, **1.8V is supplied to GAP9's e-Fuse programming voltage pin when jumper J1 is closed** ; it is removed when J1 is open. Note this specific supply is necessary only during programming, when e-Fuses is idle or being read this supply is not required.

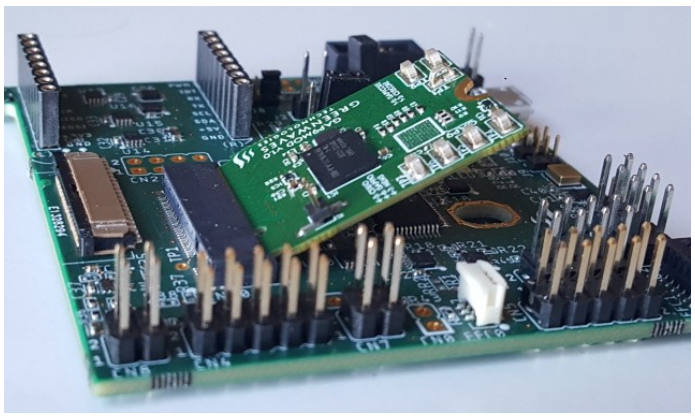
- Jumper J2 [GAP9Mod v2.x only] - Control of CSI-2 power supply (V0p65_CSI2)

The CSI-2 IP embedded in GAP9 requires 2 externally supplied power supplies, at 1.8V and 0.65V-0.8V. While the former can be cut off internally before actually reaching the CSI-2 logic, the latter cannot. Therefore, to avoid useless power consumption from the CSI-2 IP in scenarios where this functionality is not needed, an external jumper (J2) allows to externally cut off this power supply (on GAP9Mod v2.x only). **Jumper J2 must therefore be fitted to use a camera module connected to the CSI-2 connector of GAP9_EVK**

> Installing GAP9Mod into GAP9_EVK's M.2 receptacle.

The GAP9Mod module must be gently inserted **with an approximate 20° angle** into the M.2 receptacle, the B-type keying guarantees correct orientation.

Gently drive it right to the end of the receptacle. The card is then held in place with a 20° tilt. It can then be pressed down to be parallel to the GAP9_EVK carrier board and securely fastened with a screw going through the mouting hole provisioned on GAP9_EVK and a bolt on the other side. The screw and bolt can be metallic and will connect to ground of GAP9Mod and GAP9_EVK.



GAP9Mod inserted into M.2 receptacle, fastening screw not fitted (hence the 20° tilt).

5. POWER SUPPLIES, POWER MANAGEMENT & CURRENT MEASUREMENTS

Fig. 5 describes the power supply scheme implemented in GAP9_EVK and GAP9Mod.

All required power supplies are generated on-board from one of two possible external sources :

> **VEXT** : an external power supply (a battery for instance), delivering a voltage between 2.0V min. and 5.5V max. (but see note below).

> or **5V_USB** : USB master delivering 5V as per USB standard

A red LED (LED-1) will turn on when power is present (the glow might be weak when external supply voltage is under 2V or so).

A special « direct 1.8V injection » mode is also possible, with some restrictions – see below.

Also, from v3.0 onwards, there is an on/off slider switch on power from USB, on the right side of the board.

> External Power Supply VEXT Requirements

Do not connect VEXT when USB power is present.

- Maximum VEXT voltage : **5.5V**

- Minimum VEXT voltage : usage dependent, **3.45V or 2.0V/2.28V** – as follows :

> **If any peripheral connected to GAP9_EVK requires 3.3V** – for instance, a CSI-2 camera module (powered from 3V3_CAM) or Mikro-E *Click* board (powered from 3V3_PERIPH), or any add-on board relying on one of those 2 supplies – then VEXT must be at least **3.45V.**
> *[This requirements consider total current up to 300mA may be drawn on 3V3_CAM and/or 3V3_PERIPH. For applications that will always require less than that, slightly lower input voltage may be used. The user should experiment if interested.]*
>
> **> In other cases** (no 3.3V generation required),
>
> o if generating **1.8V from on-board buck DC-DC converter** (MAX38642) : VEXT must be at least **2.0V**
> *[in theory, MAX38642 can still work with VEXT as low as 1.8V. However its undervoltage lock-out threshold is 1.75V typ. and 1.8V max, which means operation at 1.8V input may be problematic if VEXT dips even very slightly, due to inrush current at power-up for instance – specifying 2V min provides some margin]*
>
> o if generating **1.8V from on-board LDO** (NCP170) : VEXT must be at least **2.28V**
> *[This requirements consider total current up to 150mA may be drawn on 1.8V and worst case operating conditions. For applications that will always require less than that, slightly lower input voltage may be used. The user should experiment if interested.]*
>
> *Note* : The above figures depend on implemented DC-DC conversion parts and may evolve if design needs to be changed – e.g. to cope with component supply shortages.

The total current required from the power source depends on the application and may range from a few mA to a few hundreds of mA (e.g. when connecting a high-resolution CSI-2 camera and/or a radio module).

> Direct 1.8V supply

If the user would prefer to inject its own, stable external 1.8V supply into the system rather than rely on the on-board 1.8V voltage converters (LDO or buck), this is possible by removing jumper J5 and injecting the 1.8V supply on the center pin.

No 3.3V peripheral cannot be used in that case since the board provides no voltage up-conversion.

> Jumper Settings for Power Supply Configuration and Manegament

Refer to section 6, « Configuration Jumper Settings » for details on how to set jumpers for configuring on-board power management, including location of jumpers on the board and their polarity.

> Global power supply h/w enable

A global on/off control signal **EXT_ON_NOFF** is provided on a header pin (CN9 pin 5, back side). It allows an external agent to switch off and on power to the full board (except JTAG and LEDs). Refer to chapters on Power Management and Expansion Connectors for details. *[Not available on V1.0 of GAP9_EVK – due to component shortage]*

> Software Enable of Selected Power Supplies

- **3V3_CAM Enable :**

The LDO that generates this power supply is switched on and off under control of :
- *GAP9_EVK v1.0 only :* GPIO63 (GAP9 pin J2), active high. *[shared with CN7 pin 3, usable as e.g. USART0_RTS)*
- *GAP9_EVK v1.1 onwards* : **GPIO0 (GAP9 pin K10)**, shared with CSI2_IO_EN (see below).
This is a 'switched' I/O (see I/O datasheet) which will go to Logic-0 when GAP9 enters sleep mode. Therefore it is not possible to keep 3V3_CAM enabled while GAP9 is in sleep mode. (However, GAP9's «light sleep» mode allows to work around this if required).

- **3V3_PERIPH Enable** :

The LDO that generates this power supply is switched on and off under control of **GPIO35 (GAP9 pin C2)**, active high. This is an always-on I/O, therefore it is possible to keep 3V3_PERIPH enabled while GAP9 is in sleep mode.

- RAM_PWR_EN :

Power supply to the RAM chip present on GAP9Mod is switched on and off under control of **GPIO88 (GAP9 pin B3),** active high. This is an always-on I/O, therefore, if so wished, it is possible to keep the external RAM powered while GAP9 is in sleep mode.
Note : this GPIO controls powers to the RAM only, the Flash present on GAP9Mod is always on (can be put in deep power down with sub-uA sleep current when needed).

- CSI2_IO_EN :

Power supply to the CSI-2 PHY (I/Os and associated mixed signal hardware) embedded in GAP9 is switched on and off under control of **GPIO0 (GAP9 pin K10)**, active high. This is a 'switched' I/O (see I/O datasheet) which will go to Logic-0 (thus disabling CSI-2 I/Os) when GAP9 enters sleep mode.
*From GAP9 EVK v1.1 onwards : t*his same GPIO0 also enables/disables 3V3_CAM (see above).

> Current Measurements

Current measurements can be performed at multiple strategic locations on the board.

Here are some features worth noticing :

- The full application current can be measured (at J4) excluding JTAG bridge (and associated level shifters) as well as LEDs – since these are not representative of what would happen in a real product. Then many sub-blocks can be measured independently, too.

- The location of measurement points allows to measure the conversion efficiency of DC-DC converters (GAP9's internal as well as external Ics) – comparing power in front of and behind the converter.

- Power to CSI-2 camera (3V3_CAM) and MikroE peripheral boards (3V3_PERIPH) – and to any add-on hardware using these supplies – can be disabled by software under control of a GPIO, or completely cut off by removing a jumper.

- The full board may be externally switched off under control of an external agent through a header pin controlling a switch on VIN

As shown on Fig. 5, depending on locations, **current is either measured across an on-board series resistor** [red stars in the drawing] **or across an external sense resistor or ampere-meter** [blue stars in the drawing].

- In the first case (essentially true for GAP9Mod): test points are provided on each side of the on-board sense resistor (0.5 ohm). Measure the voltage across these 2 test points to obtain the current that flows, using Ohm's law.

- In the second case (essentially true for all GAP9_EVK except GAP9Mod) : remove the jumper that normally enables a given current path. Replace it by an external sense resistor (select tolerance and value according to your use case) and measure voltage drop and therefore current. Or, replace it by an amperemeter or any other suitable current measuring equipment.

Beware of the voltage drop (or 'burden voltage') introduced by your measurement equipment, which may distort results – see note at the end of this chapter and Appendix E.

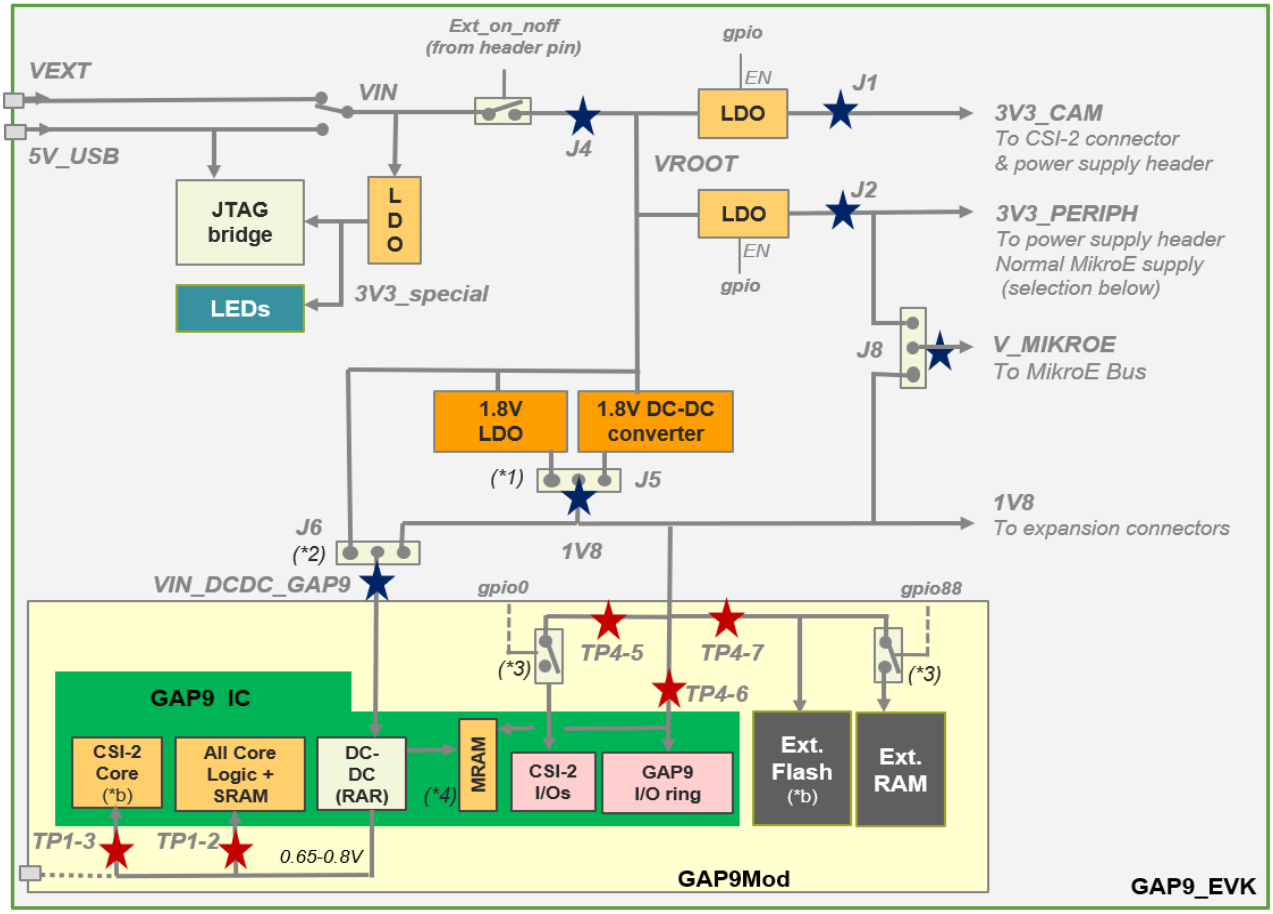
– Some extra warnings and precautions when doing current measurements –

1) 0.5ohm on GAP9Mod vs 0.65V

2) load res on J5

KEY : ★ = Current measurable across on-board 0.5ohm series resistor

★ = Remove jumper and fit external resistor or amperemeter to measure current.



*(*1): selector enables experimenting 1.8V generation by DC-DC or by LDO for different power efficiency / noise&ripple trade-offs [also usable for direct injection of external 1.8V if wished]*

*(*2): So GAP's DC-DC can operate direct from VBAT or from 1.8V (« single rail » approach)*

*(*3): Power switches controlled from GAP9's GPIOs – assignment defined by GAP9_EVK implementation*

*(*4): The eMRAM is supplied by both 1.8V (from VDDIO) and internal VSAFE itself derived from VIN_DCDC (a.k.a. VBAT)*

Fig. 5.a – Power Supply Scheme

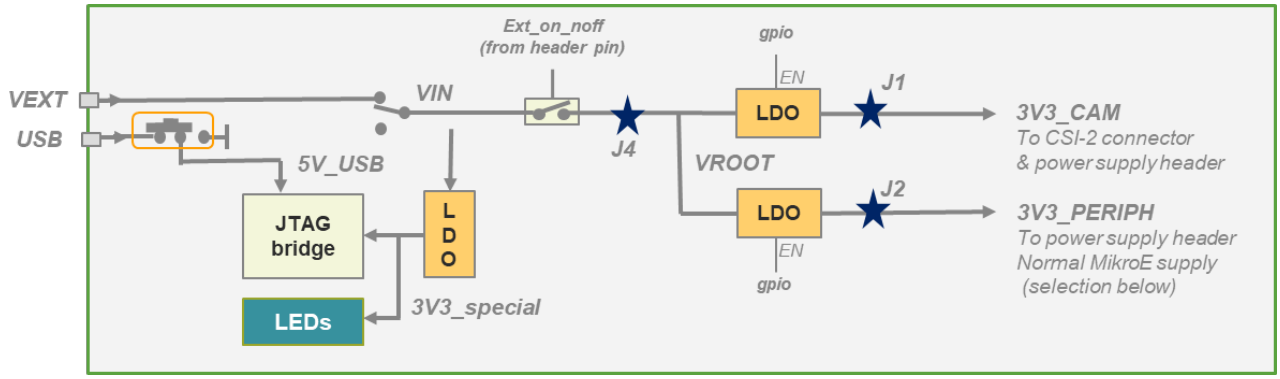


Fig. 5.b – From v3.0 onwrads : addition of a global power on/off switch on USB power

(downstream circuitry not shown here is identical to previous versions)

Assignment of jumpers and test points for power supply measurements is as follows (see also Fig.6 and Fig.3.b for a photo showing locations of test points and jumper on the boards):

> On GAP9Mod :

Measure voltage drop (in mV) between suitable test points, mutiply by 2 to get current (in mA) – as there is a 0.5ohm resistor between those test points :

- TP1-TP2 : to measure current drawn by all **GAP9 core logic and embedded memories** (on 0.65V-0.8V supply provided by internal DC-DC converter)

- **TP1-TP3** : to measure current drawn by **digital core of CSI-2 PHY**

- TP4-TP5 : to measure current drawn by **I/O part of CSI-2 PHY**

- TP4-TP6 : to measure current drawn by all **GAP9 I/Os** (except CSI-2)

- TP4-TP7 : to measure current drawn by **external Flash and RAM** chips implemented on GAP9Mod

> On GAP9_EVK board :

Remove jumper and measure current across header pins by any suitable means (for instance, voltage drop across external resistor or amperemeter in series) :

- J1 : to measure current drawn on **3V3_CAM** (typ. by CSI-2 camera, possibly by other hardware)

- J2 : to measure current drawn on **3V3_PERIPH** (typ. by 3.3V MikroE *Click* board, possibly by other hardware on expasion connectors)

- J4 : to measure current drawn by **complete board except JTAG bridge and LEDs** – typically used to get a meaningful measure of current needs for a total application

- J5 : usable to measure total current drawn on **1.8V power supply** (by comparing to current measured across J4, with J1 and J2 off, and converting to power, can be used to assess conversion efficiency of the 1.8V DC-DC converter implemented on GAP9_EVK)

- J6 : usable to measure current drawn by **GAP9's internal DC-DC converter** (by comparing to current measured on TP1-TP2 of GAP9Mod and converting to power, can be used to assess conversion efficiency)

- J8 : to measure current drawn on **V_MIKROE** (by *Click* board – or custom expansion board – plugged into mikroBUS connector)

> IMPORTANT NOTES on getting meaningful power supply measurements

- To get correct current/power consumption measurement, it is essential to take into account the possible impact of your current measurement equipment itself on the results obtained. **Refer to Appendix E.** When measuring current through the 0.5-ohm on-board current sense resistors, the bias is generally negligible, it may be much more significant with some type of ampere-meters for example.

- The current into GAP9 measured at J6 will depend on the efficiency of GAP9's internal DC-DC buck converter, which in turn depends on its load (efficiency drops for very low currents) and on the input to output voltage ratio (efficiency is better when input voltage is closer to output target voltage).

- Similarly, the efficiency of the on-board DC-DC converter in front of J5 will depend on the same factors. Also, in most cases, efficiency of a buck DC-DC converter is better than that of an LDO (the latter being simply the output to input voltage ratio), but the reverse may be true for very small load currents (for instance, in sleep mode possibly) or when input voltage is very close to output target voltage. In addition, the voltage produced by an LDO will normally be cleaner, with less ripple and noise, than that produced by a buck DC-DC – this may or may not be important for the target application. This is why GAP9_EVK provisions both options (LDO or buck DC-DC), selectable through J5.

- Therefore, when measuring the current of a full application or indeed at locations such as J6, J5,J4, care should be taken to select the most meaningful settings for the application : what external input voltage ? LDO or DC-DC at J5 ? 1.8V or battery voltage into GAP at J6 ?
In case of doubt, a generic starting point may be external voltage VEXT=3.6V, down-conversion to 1.8V by buck DC-DC (J5 on position 1-2) and 1.8V into GAP9's DC-DC (J6 on position 1-2). This may then be tuned to optimize results, taking into account application constraints.

- When measuring overall system current, pay attention to the current drawn on ancillary power supplies.
In sleep modes especially, make sure any chip or functionality that does not need to be enabled is not. In particular (unless this is not compatible with system usage scenario) :
. Switch off LDO supplying 3V3_PERIPH to cut quiescent current – using GPIO35 on GAP9 pin C2,
. Switch off LDO supplying 3V3_CAM to cut quiescent current and disable power supply to GAP9's CSI-2 interface to cut leakage there – using GPIO35 on GAP9 pin C2 for both purposes (GAP9_EVK v1.1 onwards)
. If use case allows, disable power supply to GAP9Mod's Octal-SPI RAM (to cut leakage there) using GPIO88 on GAP9 pn B3 [and make sure any I/Os connecting to the disabled RAM is either input or driving 0]

Finally, as already noted, to measure current drawn by a full application, it is recommended to do so at J4, thus keeping « parasitic » consumption from JTAG bridge/probe and status LEDs out of the picture – as they are not representative of what an actual produt would implement.

6. CONFIGURATION JUMPER SETTINGS

Fig. 6.a (for GAP9_EVK V1.x) and 6.b (for GAP9_EVK v2.x) show the locations of the different configuration jumpers and their possible settings.

Power Supply Measurement & Enable/Disable/Selection

J1 – 3V3_CAM

J2 – 3V3_PERIPH

J4 – VROOT

J5 –Sel.1V8 (1 2 3)

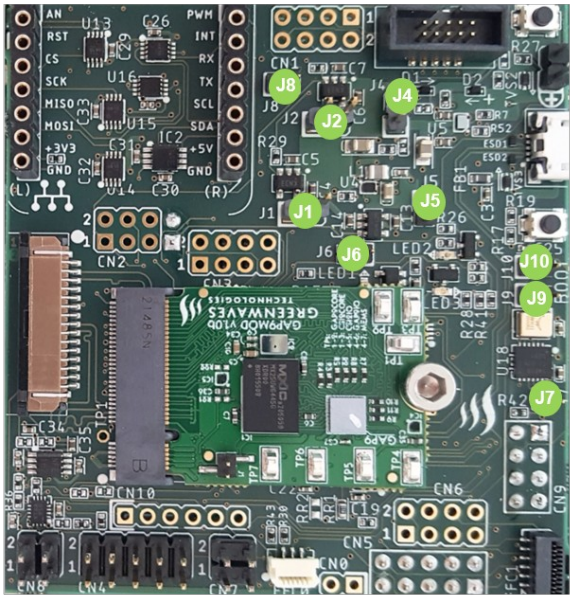
2-1: 1V8 from buck DC-DC converter
2-3: 1V8 from linear converter (LDO)
Open: inject external 1.8V (thru pin 2)

J6 –Sel.Vin-GAP9 (3 2 1)

2-1: GAP9 internal DC-DC is supplied with 1V8
2-3: GAP9 internal DC-DC is supplied with VROOT (=5V_USB or VEXT)
Open: inject external supply to GAP9 (thru pin 2).

J8 – sel.V-MikroE (1 2 3)

2-1: V-MikroE = 1.8V
- typ. to g'al purpose expansion board
2-3: V-MikroE = 3.3V
– typ. To MiroE 'Click' board
Open: No expansion bord, disable related h/w on-board



***Note**: Jumpers J1-2 and J4-6 can be replaced by, e.g., external sense resistor or amperemeter to measure current consumed on this power supply.*

Other Configurations

J7 - MICDAT
Enable output data from PDM mic to SAI1

J10 – BOOT1 (1 2 3)
2-1: BOOT1 = Logic-1
2-3 : BOOT1 = Logic-0
Open: defaults to Logic-0

J9 – BOOT0 (1 2 3)
2-1: BOOT0 = Logic-1
2-3 : BOOT0 = Logic-0
Open: defaults to Logic-0

Fig. 6.a – Configuration Jumpers – valid for GAP9_EVK v1.x

Power Supply Measurement & Enable/Disable/Selection

J1 – 3V3_CAM

J2 – 3V3_PERIPH

J4 – VROOT

J5 –Sel.1V8 (1 2 3)

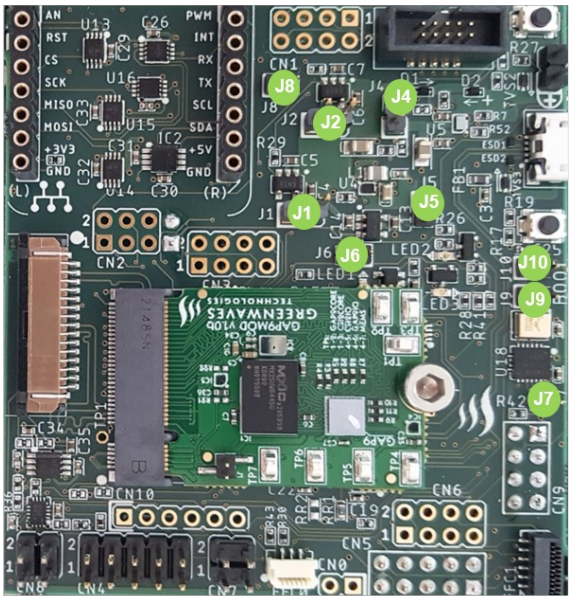
2-1: 1V8 from buck DC-DC converter
2-3: 1V8 from linear converter (LDO)
Open: inject external 1.8V (thru pin 2)

J6 –Sel.Vin-GAP9 (3 2 1)

2-1: GAP9 internal DC-DC is supplied with 1V8
2-3: GAP9 internal DC-DC is supplied with VROOT (=5V_USB or VEXT)
Open: inject external supply to GAP9 (thru pin 2).

J8 – sel.V-MikroE (1 2 3)

2-1: V-MikroE = 1.8V
- typ. to g'al purpose expansion board
2-3: V-MikroE = 3.3V
– typ. To MiroE 'Click' board
Open: No expansion bord, disable related h/w on-board



***Note**: Jumpers J1-2 and J4-6 can be replaced by, e.g., external sense resistor or amperemeter to measure current consumed on this power supply.*

Other Configurations

J7 - MICDAT
Enable output data from PDM mic to SAI1

J10 – BOOT1 (*) (2 1)
Open (not shunted): BOOT1= Logic-0
Closed: BOOT1 = Logic-1

J9 – BOOT0 (*) (2 1)
Open (not shunted): BOOT0= Logic-1
Closed: BOOT0 = Logic-0

(*) Default - J10 and J9 open:
→ BOOT[1:0] = '01' i.e. boot from JTAG
Closing jumper inverts BOOT pin value.

Fig. 6.b – Configuration Jumpers – valid for GAP9_EVK v2.x

Below is some further information on configuration jumper usage :

J1 - 3V3_CAM:
Can be used to monitor current consumption on 3V3_CAM (replacing jumper by sense resistor or amperemeter), or to cut off power on 3V3_CAM

J2 - 3V3_PERIPH:
Can be used to monitor current consumption on 3V3_PERIPH (replacing jumper by sense resistor or amperemeter), or to cut off power on 3V3_ PERIPH

J4 - VROOT:
Can be used to monitor current consumption on VROOT, which is the consumption of the full application except JTAG bridge (and associated level shifters) as well as LEDs – the latter contributors are excluded as not representative of what would happen in a real product.

J5 - Sel.1V8:
- Selects if the 1.8V power supplied to external memories, GAP9 I/Os, etc., and possibly GAP9's VBAT input (see description of power supply scheme above) is generated by on-board LDO, on-board buck DC-DC converter or perhaps supplied from external source. This allows to experiment with different power supply noise/ripple/conversion efficicency trade-offs..
- Can also be used to monitor current consumption on this supply.

J6 – Sel.Vin-GAP9

Prior to GAP9_EVK v2.1 and starting again from v3.2 :

- Selects if GAP9's internal DC-DC buck converter is supplied from VROOT (which is either 5V coming from USB or external power supply – typically, battery) or from the on-board generated 1.8V. This selection leads to different converison efficiency trade-offs. In the former case, only one conversion is involved in going from battery voltage to GAP9 core voltage (0.65V-0.8V generated by GAP9's internal DC-DC). In the latter case, two successive conversion are involved, from battery voltage to 1.8V first, then from 1.8V to 0.65V-0.8V. The former case is likely to be more efficient, however (and keeping in mind conversion efficiency of the internal DC-DC degrades as input voltage it sees increases) the second case may be interesting if the first conversion is extremely efficient, as it will place the second conversion in a more favorable configuration with a lower input voltage.
- These header pins can also be used to monitor current supplied to GAP9's internal DC-DC buck regulator.

From GAP9_EVK v2.1 to v3.1 :

- J6 is a 2-pin header to be shunted in normal operation [from V2.1, GAP9's internal DC-DC converter is always fed with 1.8V from on-board 1.8V DC-DC regulator – LDO or buck, see J5]. J6 can be replaced by a sense resistor or some test equipment to measure the current supplied to GAP9's internal DC-DC converter.

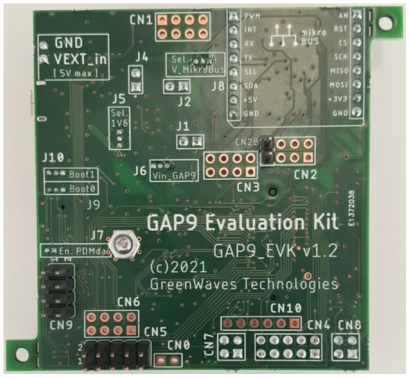
J7 - MICDAT:
- When present, this jumper will cnnect the PDM data output of the on-board digital microphone to SAI1 (SAI1_SDO). Refer to the section of this document dedicated to the on-board microphone.

J8 – Sel.V-MIKROE:
- Selects if V_MIKROE provided on the 3V3 pin of the mikroBUS connector is 3V3_PERIPH or 1V8 instead; this also defines if all digital signals present on the mikroBUS connector employ 0-3.3V signaling or 0-1.8V signaling. The former case is the normal setting when a MikroE *Click* board is in use (inserted in the mikroBUS connector). The second case is supported to also allow using the mikroBUS connector as a generic expansion connector, which may more conveniently employ 0-1.8V signals.

Note : the I2C interface of this expansion connector shoud not be used if 1.8V signaling is selected – because a level shifter that can't operate at 1.8V is involved.
- These header pins can also be used to monitor the current consumed by any add-on board connected to the mikroBUS connector and associated level shifters.

J9, J10 – BOOT0, BOOT1:
- Select the values assigned to GAP9 pins BOOT1 and BOOT0. These can be used as an alternative or complement to eFuses to define from which source GAP9 will boot. If these jumpers are not populated, GAP9_EVK will by default rely on eFuse settings, which will mean boot from JTAG if no specific eFuse programming was applied.
Refer to specific section on BOOT pins for details.

Note that the back side of the PCB provides the names of headers/jumpers and connectors, with some indications on their roles.



7. BOOT PINS

GAP9 can boot from the following sources : JTAG, embedded MRAM, HyperFlash on external memory interface, SPI Flash on external memory interface (single SPI to Octal SPI Flash are supported).

[Note : for initial accesses to external SPI memories, the hard-coded boot code always defaults to single SPI at moderate speed (50MHz, tbc) – then the secondary boot code stored in external memory or eMRAM would typically switch to increased bus width and higher speed].

Selection of boot source is done :

- either through the setting of **specific eFuse bits**

- or through the setting of 2 **specific pins** that are examined at reset

Those pins are **BOOT1 = GPIO87** and **BOOT0 = GPIO86**. They can be weakly pulled with jumpers **J10** and **J9** to 1.8V or to 0V in order to define their reset value (pulled to GND if jumpers not fitted) – refer to the paragraph on configuration jumper setting in this document. They are both sampled at exit from reset; if both are Logic0 then boot source is decided by eFuse bits, else those pins determine boot source. It is nevertheless possible to disable consideration of BOOT pins by burning an appropriate eFuse.

Decoding of BOOT pins is as follows :
BOOT[1:0] =

00 (binary) → eFuse-defined boot source
01 → boot from JTAG
10 → boot from HyperFlash
11 → boot from eMRAM

[Note : booting from OctalSPI Flash is obtained either on secondary boot with primary boot from eMRAM, or with adequately eFuse-defined direct boot]

The jumpers used to set BOOT pin values are different between versiojns V1.x and V2.x of GAP9_EVK. The configurations required to set them to Logic-0 or Logic-1 are explicited in Figures 6;a and 6.b, jumpers J9 and J10.

WARNINGS:

- *on GAP9_EVK v1.1/1.2 provided before March 20th*, *2022* : jumper J9 may not reliably pull BOOT0 to Logic-1 (as seen from GAP9) – no problem to set BOOT0 to Logic-0. This may be worked around either by pressing button S3 (top right corner of board) during boot (this firmly drives BOOT0 to Logic-1) or by removing resistor R27. GAP9_EVK boards provided after March 2àth, 2022, already implement this fix.

- setting V_MIKROE (using J8) to 1V8 at power-up – rather than 3V3_PERIPH (normal setting for standard mikroBUS usage) or 0V (J8 open) – may cause INT signal of mikroBUS to interfere witht setting of BOOT1 and should be avoided. Once boot is done, V_MIKROE (J8) can be freely set, unless there is a need to use the on-board LED (see Section 9).

8. PROGRAMMING & DEBUG (JTAG + debug UART)

The code to be executed by GAP9 can be downloaded into its memories (eMRAM and/or external Flash) through GAP9's JTAG interface. At board level, this interface is accessible by two means :

- either the on-board 10-pin JTAG connector (this requires an external JTAG probe) -

- or, through USB (used as serial communication port)

The choice between using the on-board USB-JTAG conversion or an external probe is made through a **PCB assembly option :-**

- on-board USB <> JTAG(+UART) conversion if 0-ohm resistor RR1 is fitted and RR2 is not *(default setting on GAP9_EVK v1.2)*,

- external probe if RR2 is fitted and RR1 is not

GAP9's SDK comes with **OpenOCD** tools to handle communication through USB with the JTAG probe or the conversion chip.

***Warning** : due to worlwide component shortage badly affecting the FTDI chip used for on-board USB<>JTAG conversion, some units of GAP9_EVK may not support on-board USB<>JTAG conversion and mandate the use of an external probe.*

> On-board USB <> JTAG (+UART) conversion (RR1fitted, RR2 not mounted) **:**

In this case, code is downloaded from PC through a USB cable and USB-to-JTAG translation occurs on-board (using an FTDI USB-to-serial chip – FT2232D on GAP9_EVK v1.x, or the faster FT2232H on GAP9_EVK V2.x).

In that mode, the conversion chip also handles UART-to-USB conversion : data can be received from / sent to a serial port terminal running on PC, using USB as serial communication port (VCOM).

The on-board bridge owns the JTAG bus and will drive JTAG_TDO and GAP9's hardware reset; therefore, those pins must not be actively driven elsewhere and no external JTAG probe should be connected.

It will also own the signals used as UART interface, **with some arbitration w.r.t mikroBUS™** which shares the same signals, as follows :

- signals from GAP9 pins G5(GPIO45) and K4(GPIO44**)** are made available to the USB bridge for use as **UART** signals, **using the remap feature** of selected GAP9 I/Os (see I/O datasheet), with **G5(GPIO45) usable as Uart_Tx** from GAP9 and **K4(GPIO44) usable as Uart_Rx to GAP9**.

- the conversion chip takes G5(GPIO45/Uart_Tx) as input and normally drives K4(GPIO44/Uart_Rx)

- **however, if power supply V_MikroE is enabled** (*i.e.*, J8 fitted – taken as an indication that a MikroE *Click* board is potentially in use), **then the *Click* board takes precedence** and the USB bridge releases signal K4(GPIO44/Uart_Rx to GAP9) to let the *Click* board drive it. On the other hand, signal G5(GPIO45/Uart_Tx from GAP9) is dispatched to both the mikroBUS and the USB bridge – which means it is still possible to *receive* information from GAP9 using a serial com terminal on PC, assuming the plugged *Click* board doesn't actually exploit Uart_Tx from GAP9 (else data may get mixed up).

~~- **Optional Flow Control -** In addition, from GAP9_EVK v3.1 onwards : h/w flow control is supported, with **GPIO40 remappable as uart2_cts#** to GAP9 (exclusive usage with I2C of MikroE due to h/w sharing) and **GPIO67 remappable as uart2_rts** from GAP9 (exclusive usage with PWM to MikroE due to h/w sharing) .~~ [bug in implementation]

> External JTAG Probe (RR1 not mounted, RR2 fitted) **:**

In this case, code is downloaded from PC through an exernal JTAG probe such as, for instance, Olimex ARM-USB-OCD-H or Segger's J-Link. The probe must be compatible with 1.8 I/O levels. It must also provide the (sometimes optional) JTAG_NTRST pin (beware, some JTAG-20<>JTAG-10 adapters do not propagate this signal).

No UART connects to the external JTAG probe. A separate, external USB(VCOM)<>UART converter module connecting to a UART interface on GAP9_EVK expansion headers is required to send/receive data to/from a serial port terminal running on PC.

9. USING THE ON-BOARD LEDs & PUSH-BUTTONS

> Power-OK LED

LED-1 is a red LED that will turn on when power is present at board input, either when an active USB cable is plugged or when power VEXT is supplied through J3 (the glow might be weak when external supply voltage is under 2V).

> RESET Push Button

The push-button in the top corner of GAP9_EVK controls GAP9's hardware reset (asserted when button is pushed, active low).

> User LEDs

There are 2 general purpose LEDs on the board that can be controlled by user software :

- **LED-2 is a green LED controlled by GPIO86** (GAP9 ball D6). It is ON if the GPIO is driven to Logic-1, OFF otherwise. *Beware* : GPIO86 is shared with push-button S3, so if the push-button is used, GPIO86 should be in input mode and cannot control the LED (which will reflect the state of S3, see below).

- **LED-3 is a blue LED** controlled not directly through a GAP9 GPIO, but through a GPIO provided by an I2C controlled GPIO expander IC present on the board. The expander is powered from 3V3_CAM so this power supply must be enabled (J1 fitted and voltage greater than 3.3V at board input) for LED-3 to be usable. **LED-3 is driven from pin 6 (GPIO4) of IC4, FXL6408UMX, a GPIO expander chip controllable through I2C3 of GAP9** – refer to GAP9_EVK board schematic and FXL6408UMX datasheet for full details.

All LEDs are powered from a dedicated 3.3V supply branch so that they can be used without impacting the full application current measured at J4, which does not include this branch.

> User Push Button

The status of user push-button S3 can be captured through **GPIO86** configured as floating input.

However, GPIO86 is shared with LED-2 and is also boot control pin BOOT0 (see Section 7). This puts some constraints on push button (S3) usage, as follows :

1) LED-2 is controlled by the same GPIO86 as S3, therefore, when the user configures GPIO86 as input to monitor the push-button, LED-2 cannot be software contolled and it will reflect the state of push-button S3 (OFF when pushed, else ON).

2) BOOT0 sharing :

- on GAP9_EVK v1.x : pressing the push-button drives GPIO86 to Logic-1, when not pressed GPIO86 takes the value assigned to BOOT0 (through jumper J9) – therefore in practice push button is usable only when BOOT0 is Logic-0, which may be a limitation

- on GAP9_EVK v2.0 : pressing the push-button drives GPIO86 to Logic-0 by default, but this can be changed by acting on 'solder jumpers' SB1 and SB2 – so the push button can be used whatever the default value (button not pressed) set on GPIO86/BOOT0 through J9.

10. USING THE ON-BOARD MICROPHONE

Prelimary Note : *the microphone present on GAP9_EVK board is intended for relatively simple use cases. For rich audio applications, with mutiple microphones, audio out, etc., a separate Audio Add-On board is available from GreenWaves Technologies.*

To avoid interfering with other microphones possibly connected to SAI1 through expansion connectors and to keep the on-board microphone out of power consumption measurements when no mic is actually required by the application, the on-board microphone is isolated from the system by default. That is, it is not powered and all its I/Os are disconnected.

To power and enable the on-board microphone, the user should **bridge pin 1 (1V8_MIC) and pin 2 (1V8) of connector CN9** [beware CN9 is on bottom side and pins 1,2 there correspond to pins 7,8 of CN9 on Audio Add-On]

In addition, to connect the PDM output of the on-board Vesper microphone to the SAI1 interface, thus allowing GAP9 to capture the digitized sound, **jumper J7 must be closed**. By not closing J7, it is possible to enable the on-board microphone without then outputting PDM data from the microphone onto SAI1, *e.g.* to avoid interfering with other digital microphones using SAI1 on some add-on board. With the VM3011 in particular, this makes it possible to use it in wake-up mode only.

> *Until GAP9_EVK V3.1* – Vesper VM3011 'Always Listening' PDM Microphone

Vesper's VM3011 is a « low-noise PDM digital piezoelectric MEMS microphone with adaptive ZeroPower™ listening ». In addition to operating like any standard digital PDM output microphone, it can also operate in a virtually zero power 'always-listening' mode from which it will exit upon detection of a sound above background noise level. Upon wake-up it can raise a signal on a specific output pin to wake-up a full system.

When enabled, the VM3011 microphone will output its wake-up signal **VM3011_WUP on pin 4 of connector CN6** of GAP9_EVK – it is up to the user to then bridge this pin to any suitable GAP9 input signal available on a header pin. I

> *Starting with GAP9_EVK V3.2* – Infineon IM69D128 PDM Microphone

From GAP9_EVK v3.2, the VM3011 having been obsoleted by Vesper, it has been replaced by an IM69D128 from Infineon. This is a regular PDM microphone, without any interrupt output nor I2C programmability.

The PDM data output of this microphone is RC filtered (time constant ~3.3ms) and then provided on pin 4 of connector CN6 for any possible use, in place of the wake-up signal from VM3011 available in previous versions.

11. GPIO EXPANDER

A GPIO expander is implemented on GAP9_EVK, it provides extra programmable I/Os with 0-3.3V signaling levels. The power supply is 3V3_CAM (therefore the latter must be enabled by the selected power management settings, see section on Power Management).

Warning : On GAP9_EVK 1.1/1.2 provided before March 16th, 2022, there is a small bug that may prevent the GPIO expander from interfacing properly with I2C3 (VDD of the GPIO expander is set to 3.3V whereas it should have been 1.8V). This can be worked around by pulling I2C3 to a value slightly higher than 1.8V, which can be done by implementing one of those 2 hardware changes :
- populate R37 and R38 (not fitted by default) with 47K to 100K SMD resistors (0402 format),
or :
- connect a resistor (*e.g.,* of the standard discrete axial type) of value 47K to 100K (for instance, 68K) between I2C3_SCL taken on CN10 pin 5 and 3V3_CAM taken on CN3 pin 1, as well as between I2C3_SDA taken on CN10 pin 6 and, again, 3V3_CAM taken on CN3 pin 1.

The GPIO expander IC is On Semi's FXL6408UMX (IC4 in the schematic). It is controlled from GAP9 through I2C3. It provides 8 fully progammable I/Os plus an interrupt request output. Refer to GAP9_EVK schematic for full details. Here are some salient points :

- Four of those programmable I/Os (GPIO0, GPIO1, GPIO2, GPIO3 of the expander IC) are assigned to the CSI-2 MIPI connector.

- GPIO4 of the GPIO expander (signal GPIO-A_3V3CAM in the schematic) controls the blue LED-3 present on GAP9_EVK.

- GPIO5 to GPIO7 of the expander IC, plus its interrupt request output (INT#), are routed to header pins. It is up to the user to make appropriate connections between selected header pins to ultimately connect the GPIO/IRQ she wants to use on expander side with a GPIO of GAP9.

Note – Sharing of GPIO-D_3V3CAM (header + IRQ) :

Signal GPIO-D_3V3CAM, available on a header pin, connects to both GPIO7 of the expander IC and its INT# output. It is up to the user to decide whether she wants to use this header pin as a general purpose IO (in which case she should disable the INT# outout of the expander) or as an interrupt request (in which case she should disable GPIO7 of the expander).

- Tips for using the GPIO Expander -

> Do not forget to enable 3V3_CAM on GAP9_EVK by *(a)* closing jumper J1 and *(b)* setting GPIO0, which enables/disables the 3.3V LDO, to Logic-1 (enable)
> When writing your I2C3 configuration code for GAP9, pay attention to the fact that I2C3 is available as 'Alternate 2' function on the I/Os rather than as default (Alt.0) function.

12. CSI-2 / MIPI CONNECTOR

– IMPORTANT NOTE – *On GAP9Mod v2.x,* ***mini-jumper J2 of GAP9Mod must be fitted*** *to use the CSI-2 interface. Refer to Section 4, paragraph 'Jumpers on GAPM9Mod'.*

GAP9_EVK features a CSI-2 MIPI connector (C-PHY) for a 15-pin, 0.5mm pitch FFC cable, to connect a camera module. **With GAP9 in WLCSP package, only one MIPI lane is supported**, up to 1.5Gbit/s *[tbc]*.

Its pinout is compatible with that of Raspberry Pi camera modules (Fig. 7). The objective is to allow usage of off-the-shelf camera modules to start with. Possible vendors include :

- the Raspberry Pi foundation - *www.raspberrypi.org/products/camera-module-v2*
- Arducam - *www.arducam.com/docs/cameras-for-raspberry-pi/*
- Vision Components - *www.vision-components.com/en/products/oem-embedded-vision-systems/mipi-camera-modules/*
- etc.

Make sure the camera you select can be operated in single lane mode (incidentally, the latest 8Mpix+ 'official' RPi camera appears to support only 2 or 4 lanes).

Example off-the-shelf camera modules directly compatible with GAP9_EVK's CSI-2 connector include :

- the 'official' RaspberryPi camera **v1 :** 5Mpix, colour, based on Omnivision's OV5647
  [Rpi camera v2, based on Sony IMX219 is not suitable as it does not support single-lane MIPI operation]
- Arducam's OV9281 module : global shutter, monochrome, 1Mpix

Alternatively, customers may decide to build/source their own camera board/module, optimized for their needs, with a compatible FFC interface. They may also pick a non-natively compatible 3rd party solution and build/source a suitable adapter to comply with GAP9's (RPi-compatible) FFC pinout.

The connector MPN selected on GAP9_EVK is Amphenol's SFW15R-1STE1LF; this is a horizontal entry (*a.k.a* right angle) bottom contact FFC/FPC connector.

Fig. 7 – CSI-2 MIPI Connector on GAP9_EVK

> Additional details on GAP9_EVK's CSI-2 connector pinout :

- pins 13 and 14 controlling the camera (SCCB or I2C bus) are mapped to I2C bus #3 (I2C3_SCL and I2C3_SDA on balls J5 and H5 of GAP9 WLCSP).

- the power supply is 3V3_CAM (which must be enabled by closing J1), all I/O signals present on the connector are referenced to this supply

- CAM_IO0 and CAM_IO1 of the Raspberry PI standard CSI-2 pinout are normally dedicated to Power Enable ***(*)*** and LED Indicator on the RPi pinout,. They can be used freely on a custom module. On GAP9_EVK these GPIOs are provided through a GPIO expander IC (IC4 in the schematic, pins 12 and 11), see section on GPIO expander in this document.
(*) **: in that configuration, pin CAM_IO0 must therefore be set to Logic-1 to power up the camera module** . CAM_IO0 is controlled using the GPIO Expander (see section 11 above)

- CAM_IO2 and CAM_IO3 are additional GPIOs that replace the second MIPI lane normally present on the standard Raspberry Pi CSI-2 connector, which cannot be used with GAP9 in WLCSP package anyway. This would be useful only with custom camera modules, as Raspberry Pi compatible modules would regard those pins as the second MIPI lane. Again, these GPIOs are provided through the same GPIO expander IC (IC4 in the schematic, pins 7 and 8). To avoid any electrical conflict in case a dual-lane capable camera tries to output something on these I/Os, it is recommended to use IO2 and IO3 as inputs to the GPIO expander.

> Practical Tips for using a Raspberry-Pi-compatible CSI-2 module

As noted above, the CSI-2 connector implemented on GAP9_EVK follows the standard pinout of the 15-pin FFC cable employed by RPI cameras. Those cameras typically use IO-0 of the connector as Power Enable to the

camera module. OnGAP9_EVK, IO-0 of the connector is controlled through a GPIO expander (because all native GPIOs of GAP9 are already used up elsewhere).

Following is an example sequence to access such a camera from GAP9 :

- make sure J1 is fitted on GAP9_EVK (so 3V3_CAM is propagated across the board)

- set GPIO0 of GAP9 to Logic-1 (to enable the on-board LDO delivering 3V3_CAM)
[Wait at least 550us before any subsequent access to camera or GPIO expander – to leave time for LDO output to stabilize at 3.3V]

- set up GAP9's I2C3, used to access the GPIO expander (I2C address=0x44 expressed on 7 bits, excluding R/W flag MSB) and CSI-2 camera – beware, I2C3 functionality is provided as Alternate 3 (Alt.0 is I3C, not to be used here).

- set IO-0 of the GPIO expander to Logic-1 by suitable I2C accesses (this sets CAM_IO-0 input of CSI-2 camera, i.e. asserts *Power Enable* input of camera)

This may be described with the following pseudo-code :

```
void enable_csi-2_camera(void)
{
//  NOTE –  make sure J1 is fitted on the board
      configure_gap9pin(  PAD_046, Alt.2 ) ;      // configure pin as  I2C3_SDA
      configure_gap9pin(  PAD_047, Alt.2 ) ;      // configure pin as I2C3_SCL
      configure_gap9pin(  PAD_000, Alt.1 ) ;       // configure pin as GPIO0
      set_gap9gpio_value (GPIO0, 1) ;             // drive GPIO0 of GAP9 to Logic-1 – to Enable 3V3_CAM LDO
      wait_us(500) ;                              // to make sure 3.3V from LDO has stabilized
      configure_i2c3() ;                          // set baud rate etc.
      // Now, perform I2C accesses to GIO Expander FXL6408
      //  Note -  I2C device address =0x88 on 8 bits here, check what your API actually expects (7 or 8 bits)
      write_i2c3( 0x88, 0x1, 0x1) ;               // reset GPIO expander
      write_i2c3( 0x88, 0x3, 0x1) ;               // GPIO expander's IO-0 direction : output
      write_i2c3( 0x88, 0x5, 0x1) ;               //  GPIO expander's IO-0 state : Logic-1
      write_i2c3( 0x88, 0x7, 0x0) ;               // output state follows reg.0x05 (i.e. exit high-Z)
// At this stage CSI-2 camera is powered on and ready to use
}
```

13. EXPANSION CONNECTORS (incl. MikroE)

Note : some headers on top side of GAP9_EVK in its v1.0 (prototype) release have migrated to the bottom side in subsequent releases of the board : CN5, CN9, part off CN2.

A number of connectors and plated through-holes (PTH) are provided to enhance the generic Evaluation Kit with additional hardware such as add-on boards, third-party Dev Kits or other prototyping material. They carry I/Os to or from GAP9 as well as a selection of power supplies generated on GAP9_EVK. All I/Os employ 0-1.8V signaling, with the exception of those provided through the mikroBUS™ connector which can be either 0-1.8V or level-shifted to 0-3.3V.

Fig. 8 shows their location. Most connectors are standard 2.54mm pitch headers, however there are a few FFC/FPC connectors for Flat Flexible Cables. The latter are provided to carry signals which, in some use cases, may run faster than 5-10MHz or so and for which stacked board to board connectors or flying wires may not be suitable. All signals present on FFC connectors are also present on PTH or headers, for simpler usage when only slow speeds are required.

More information on exact connector coordinates is provided in Appendix, to help people who would like to design their own expansion board.

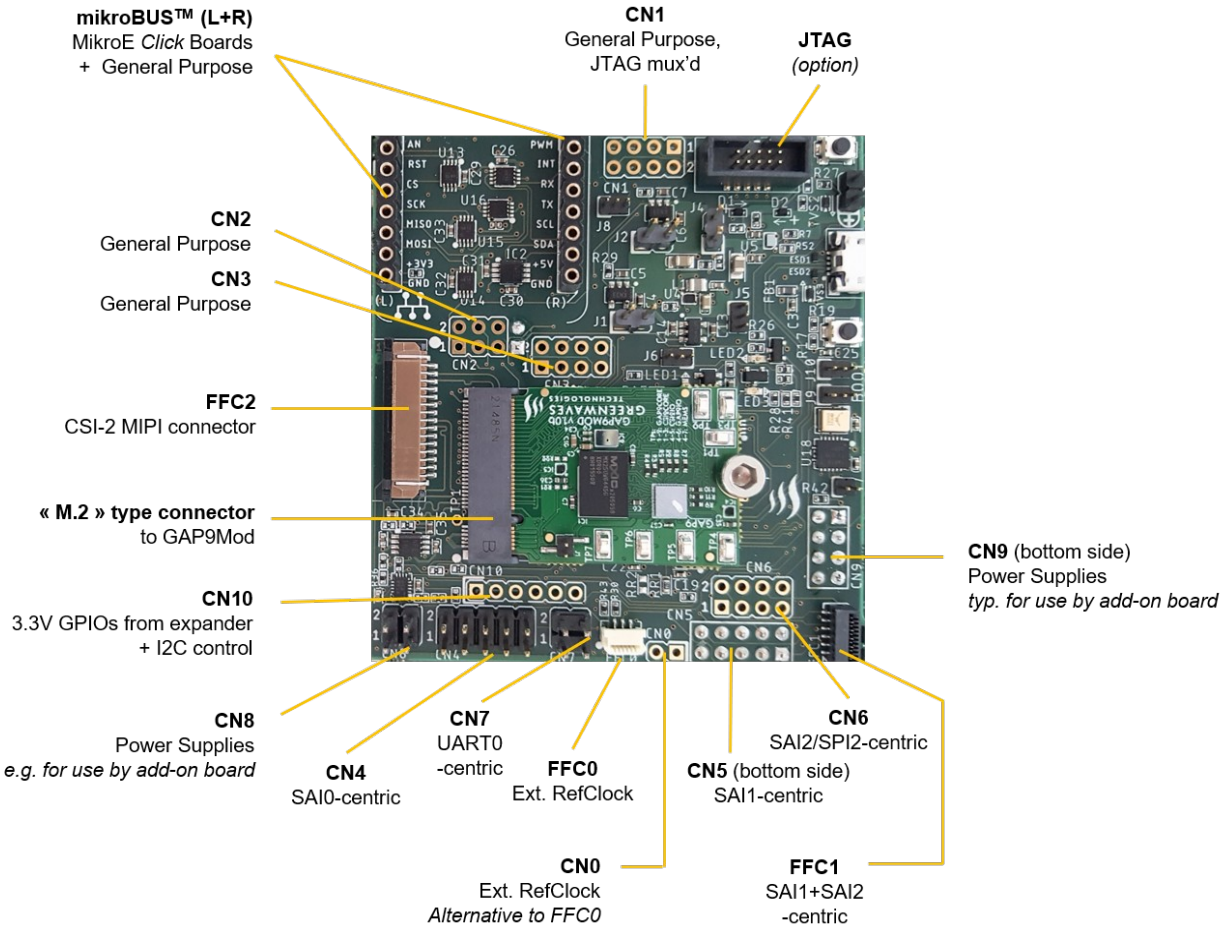
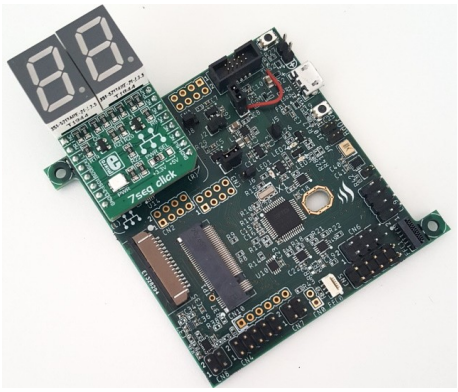


Fig. 8 – Expansion Connectors

> mikroBUS™ Expansion Connector – typ. for MikroE *Click* Boards

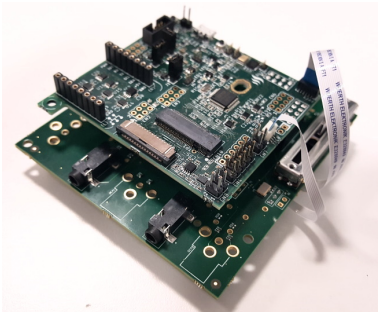


The pair of 8-pin receptacles on the top left of the board follows the mikroBUS™ standard and is intended to mate with MikroE *Click* boards (see chapter 6 on configuration jumper settings to properly power the *Click* board) . An example of such an assembly is shown opposite. Of course, it is still possible to use these headers for other purposes (with I/Os either at 3.3V or 1.8V).

Note that, while most of the GAP9 I/O signals assigned to the mikroBUS interface natively offer the required functionality (*e.g.* SPI_SCK, I2C_SDA, *etc*.) as one of their alternate functions, UART functionality is obtained by using the I/O remapping feature on GAP9 pins K4 (GPIO44) and G5 (GPIO45) – see Table 1 below.

Pay attention to the settings of jumpers J2 and J8 to enable 3V3_PERIPH as well as V_MIKROE and properly supply power to the expansion board, refer to Section 6.

> Audio Add-On Board



To enhance the generic GAP9_EVK and target specific applications, GreenWaves Technologies will provide application-focused add-on boards designed to mate with selected GAP9_EVK's connector. A first board of this type is an « audio add-on board » that adds the features typically required in TWS earbuds, headphones and other personal/portable audio applications (sound capture, D/A audio conversion and amplication, etc.).

This board essentially makes use of serial audio interfaces SAI1 and SAI2 on connectors CN5, CN6 and FFC1, as well as external reference clock on CN0/FFC0 and power supplies on CN8. It is slightly larger than GAP9_EVK and is intended to be stacked underneath it.

A full Audio Add-On Board User Manual is available.

> Signal Mapping

Please refer to the full schematic for an exhaustive description of signal assignments to connector pins.

As a complement, tables 1 to 3 below list what signals are available on which connectors. Pay attention to the fact that a given signal may be mapped to multiple header pins. In some cases, this is to allow sharing a bus between different expansion boards (for example, I2C or SPI used by a MikroE C*lick* board may also be used elsewhere, playing with Chip Address or Chip Select) ; in other cases this is just for ease of access at different locations on the board.

All signals I/Os employ 0-1.8V signaling, except those provided on the mikroBUSTM which are level-shifted to V_MikroE, which can be either 3.3V (as required by MikroE *Click* boards) or 1.8V (for possible other uses).

Refer to GAP9's I/O Datasheet for full details on any I/O you want to use.

- GAP9 digital I/Os available on expansion connectors :

Ball Name	I/O Function				Availability on Headers / PTH		Availability on FFC		Availability on MikroBus™	Notes
	Alternate 0	Alternate 1	Alternate 2	Alternate 3						
D4	RESETN	-	-	-		CN9 Pin 4				GAP9 hardware reset (active low)
B2	SPI1_SCK	GPIO33	USART3_CLK	-	CN2 Pin 5				(L) Pin4(*) – Output only	(*)= Level-shifted to V_MikroE (3.3V or 1.8V)
E4	SPI1_CS0	GPIO34	-	-	CN2 Pin 4				(L) Pin 3(*) – Output only	(*)= Level-shifted to V_MikroE
A1	SPI1_SDO	GPIO38	-	-	CN2 Pin 1				(L) Pin 6(*) – Output only	(*)= Level-shifted to V_MikroE
E3	SPI1_SDI	GPIO39	-	-		CN3 Pin 5			(L) Pin 5(*) – Input only	(*)=Level-shifted from V_MikroE
B1	I2C0_SDA	GPIO40	-	-		CN3 Pin 4			(R) Pin 6(*)	(*)=L shifted, but usable only if V_MikroE=3.3V
D2	I2C0_SCL	GPIO41	-	-		CN3 Pin 8			(R) Pin 5(*)	(*)=L shifted, but usable only if V_MikroE=3.3V
F4	I2C1_SDA	GPIO42	-	-	CN4 Pin 10	CN5 Pin 9				For CN5 : Beware – this is seen from bottom side.
C1	I2C1_SCL	GPIO43	-	-	CN4 Pin 8	CN5 Pin 7				For CN5 : Beware – this is seen from bottom side.
K4	I2C2_SDA	GPIO44	-	-					(R) Pin 3(*) – Input only	(*)=Level-shifted from/to V_MikroE, remap...
G5	I2C2_SCL	GPIO45	-	-					(R) Pin 4(*) – Output only	...I/O to use as mikroBus UART_RX/TX
H5	I3C_SDA	GPIO46	I2C3_SDA	SPI0_SDIO2	CN10 Pin 6					If CSI-2 used, shared I2C bus
J5	I3C_SCL	GPIO47	I2C3_SCL	SPI0_SDIO3	CN10 Pin 5					If CSI-2 used, shared I2C bus
G6	SAI0_SCK	GPIO48	USART2_CLK	-	CN4 Pin 5					
G7	SAI0_WS	GPIO49	-	-	CN4 Pin 7					
K5	SAI0_SDI	GPIO50	-	-	CN4 Pin 6					
H6	SAI0_SDO	GPIO51	-	-	CN4 Pin 2					
J6	SAI1_SCK	GPIO52	-	-		CN5 Pin 6	FFC0 Pin 6	FFC1 Pin 10		For CN5 : Beware – this is seen from bottom side.
H7	SAI1_WS	GPIO53	SPI2_CS1	(*) -		CN5 Pin 10		FFC1 Pin 7		For CN5 : Beware – this is seen from bottom side.
K6	SAI1_SDI	GPIO54	SPI2_CS2	(*) SPI2_SDIO2		CN5 Pin 5	FFC0 Pin 4	FFC1 Pin 12		For CN5 : Beware – this is seen from bottom side.
K7	SAI1_SDO	GPIO55	SPI2_CS3	(*) SPI2_SDIO3		CN5 Pin 1		FFC1 Pin 9		For CN5 : Beware – this is seen from bottom side.
H1	SAI2_SCK	GPIO56	SPI2_SCK	-	CN6 Pin 3			FFC1 Pin 1		
H2	SAI2_WS	GPIO57	SPI2_CS0	(*) -	CN6 Pin 7			FFC1 Pin 3		
H3	SAI2_SDI	GPIO58	SPI2_SDI	-	CN6 Pin 6			FFC1 Pin 4		
H4	SAI2_SDO	GPIO59	SPI2_SDO	-	CN6 Pin 2			FFC1 Pin 6		
J4	USART0_RX	GPIO60	-	-	CN7 Pin 1					
J3	USART0_TX	GPIO61	-	-	CN7 Pin 4					
J1	USART0_CTS	GPIO62	-	-	CN7 Pin 2					
J2	USART0_RTS	GPIO63	-	-	CN7 Pin 3					
K1	FAST_REF_CK/U	GPIO64	-	-	CN0 Pin 2		FFC0 Pin 2			On EVK V1.0, also used as 3V3_CAM Enable. Usable as external reference clock to GAP9
E5	PWM0	GPIO67	-	-	CN2 Pin 6				(R) Pin 1 – Output only	Level-shifted to V_MikroE [CN2.8 on EVK v1.0]
D3	PWM1	GPIO68	-	-		CN5 Pin 4				For CN5 : Beware – this is seen from bottom side.
G4	JTAG_TCK	GPIO81	USART4_CLK	-		CN1 Pin 8				
E2	JTAG_TDI	GPIO82	USART4_RX	-		CN1 Pin 6				
F3	JTAG_TDO	GPIO83	USART4_TX	-		CN1 Pin 2				
D1	JTAG_TMS	GPIO84	USART4_CTS	-		CN1 Pin 3				
F2	JTAG_NTRST	GPIO85	USART4_RTS	-		CN1 Pin 7				
D6	WUP_SPI2_SCK	GPIO86/BOOT0	-	-					(L) Pin 2(*) – Output only	(*)= Level-shifted to V_MikroE (3.3V or 1.8V)
D5	WUP_SPI2_SDI	GPIO87/BOOT1	-	-		CN3 Pin 7			(R) Pin 2(*) – Input only	(*)= Level-shifted to V_MikroE (3.3V or 1.8V)
A2	WUP_SPI2_CS0	GPIO89	-	-	CN4 Pin 1	CN5 Pin 2				For CN5 : Beware – this is seen from bottom side.

Table 1 : Digital I/Os availability on expansion connectors

Known Limitations -

> CN3 Pin5 (SPI1_SDI/GPIO39) : signal E3(GPIO39) in the schematic, provided on pin 5 of CN3, is also driven by level shifter U12 when E4(GPIO34) is low, resulting in **possible electrical conflict.** This is typically the case when using CN2/CN3 to carry SPI1, with SPI1_SDI=E3(GPIO39) and SPI_CS0=E4(GPIO34) : in this case SPI1 read accesses (data into GAP9) fail because SPI1_CS0 gets asserted to perform the access and this causes corruption of SPI1_SDI.
A **work-around** is to use SPI1_CS1 rather than SPI1_CS0, remapped to a pin available on one connector. For instance, CN2 pin6 bears signal E5(GPIO67) and the GAP9 I/O which drives it can be remapped to provide SPI1_CS1 on this pin.

> GPIO87 on CN3 pin7 is BOOT1 at boot , then it can be used as a GPIO but, due to the way it is shared with the mikroBUS, the following limitation apply : **to freely use GPIO87, J8 must be open so that V-mikroE is low** – and therefore usage of GPIO87 as GPIO is exclusifev with usage of the mikroBUS.

Notes :

(1) Some GAP9 I/Os are made available on different connectors. Most of these should normally be used at only one connector pin to avoid any conflict, however a few are part of a sharable bus (such as I2C or SPI) and may be used at different places, with some precautions (e.g., for SPI, keeping independent Chip Selects).

(2) The signals made available on the two mikroBUS connectors do not connect directly to the indicated GAP9 I/O. Because MikroE *Click* boards use 0-3.3V signaling while GAP9 I/Os are 0-1.8V, level shifters are present between these 2 connectors and GAP9 I/Os. These are unidirectional level shifters, whose direction is that of the mikroBUS signals they carry (even though they may also be used to connect non-MikroE hardware). Only the two I2C lines are bidirectional; however they can only be used if V_MikroE is set to 3.3V – their level shifters won't work if V_MikroE is set to 1.8V (through J8), contrarily to the unidirectional shifters.

(3) Appendix C provides additional information on mapping of TDM, I2S and PDM protocols onto the SAI SerialAudio Interfaces.

- Power Supplies and Power Management Signals mapped to expansion connectors :

Note : Also refer to the chapter dedicated to Power Management inside this document.

Power Supply / Mgmt Signal	Availability on Connectors	Notes
Connector CN0		
GND	CN0 Pin 1	
Connector CN1		
1V8	CN1 Pin 1	*Always-on 1.8V, derived from VEXT or 5V_USB through LDO or buck DC-DC.*
GND	CN1 Pin 4	
Connector CN2		
1V8	CN2 Pin 2	*Always-on 1.8V, derived from VEXT or 5V_USB through LDO or buck DC-DC.*
GND	CN2 Pin 3	
Connector CN3		
3V3_CAM	CN3 Pin 1	*Switchable 3.3V, derived from VEXT or 5V_USB through LDO and J1. Also goes to CSI-2 connector.*
VROOT	CN3 Pin 2	*Either VEXT or 5V_USB, through J4*
GND	CN3 Pins 3, 6	
Connector CN4		
GND	CN4 Pins 3, 4, 9	
Connector CN5		***For CN5 : Beware – this is seen from bottom side.***
GND	CN5 Pins 3,8	
Connector CN6		
GND	CN6 Pins 1, 8	
Connector CN8		
3V3_PERIPH	CN8 Pin 1	*Switchable 3.3V, derived from VEXT or 5V_USB through LDO and J2. Also goes to mikroBUS as V_MikroE.*
1V8	CN8 Pin 2	*Always-on 1.8V, derived from VEXT or 5V_USB through LDO or buck DC-DC.*
VROOT	CN8 Pin 3	*Either VEXT or 5V_USB, through J4*
GND	CN8 Pin 4	
Connector CN9		***For CN9 : Beware – this is seen from bottom side.***
VROOT	CN9 Pin 7	*Either VEXT or 5V_USB, passes through J4*
3V3_PERIPH	CN9 Pin 8	*Switchable 3.3V, derived from VEXT or 5V_USB through LDO and J2.*
EXT_ON_NOFF	CN9 Pin 5	*Logic input : Enable/disable of all on-board supplies (active high)*
GND	CN9 Pins 3,6	
RESETN	CN9 Pin 4	*Logic input : GAP9 hardware reset (active low)*
1V8_MIC	CN9 Pin 1	*Short with 1V8 (pin 8, can use a jumper) to supply 1.8V to GAP9_EVK's on-board PDM microphone*
1V8	CN9 Pin 2	*Always-on 1.8V, derived from VEXT or 5V_USB through LDO or buck DC-DC.*
Connector CN10		
GND	CN10 Pin 4	
Connector mikroBUS™		
V_MIKROE	mikroBus (L).7	*Either 3V3_PERIPH or 1V8 or open to mikroBUS, sel. thru J8. MikroE 'Click' boards normally use 3.3V.*
GND	mikroBus (L).8	
5V_USB	mikroBus (R).7	*5 Volt from USB cable, if present*
GND	mikroBus (R).8	

Key :

Logic Input
Always-On Supply
Switchable Supply
Ground
Other

Table 2 : Power Supplies & Power Management Signals on expansion connectors

- Other GAP9_EVK signals mapped to expansion connectors :

CONNECTOR & PIN	NAME	ROLE
CN6 Pin 4	VM3011_WUP	Wake-up signal from 'always-listeining' Vesper mic. Refer to chapter dedicated to VM3011 usage.
CN9 Pin 5	EXT_ON_NOFF	'System wide' power enable. Allows external agent to switch off complete GAP9_EVK (not available in V1.0)
CN9 Pin 4	NRESET	GAP9 hardware reset (active low).
CN10 Pin 1	GPIO-D_3V3CAM	GPIO from/to I/O expander IC4 (FXL6408UMX), pin GPIO7 (0-3.3V signaiing, requires 3V3_CAM to be enabled)
CN10 Pin 2	GPIO-C_3V3CAM	GPIO from/to I/O expander IC4 (FXL6408UMX), pin GPIO6 (0-3.3V signaiing, requires 3V3_CAM to be enabled)
CN10 Pin 3	GPIO-B_3V3CAM	GPIO from/to I/O expander IC4 (FXL6408UMX), pin GPIO5 (0-3.3V signaiing, requires 3V3_CAM to be enabled)
FFC2		***CSI-2 MIPI Connector, Pinout compatible with Raspberry PI camera, restricted to 1 data lane***
FFC2 Pin 1	GND	
FFC2 Pin 2	MIPI_D0_N	Differential (negative) CSI-2 data line (lane 0)
FFC2 Pin 3	MIPI_D0_P	Differential (positive) CSI-2 data line (lane 0)
FFC2 Pin 4	GND	
FFC2 Pin 5	CAM_IO2_3V3	Usable as GPIO from camera module to GAP9 (beware of electrical conflict if camera supports 2 data lanes)
FFC2 Pin 6	CAM_IO3_3V3	Usable as GPIO from camera module to GAP9 (beware of electrical conflict if camera supports 2 data lanes)
FFC2 Pin 7	GND	
FFC2 Pin 8	MIPI_CK_N	Differential (negative) CSI-2 clock
FFC2 Pin 9	MIPI_CK_P	Differential (positive) CSI-2 clock
FFC2 Pin 10	GND	
FFC2 Pin 11	CAM_IO0_3V3	Usable as GPIO from GAP9 to camera module (e.g., Power Enable on Raspbery Pi cameras)
FFC2 Pin 12	CAM_IO1_3V3	Usable as GPIO from GAP9 to camera module (e.g., LED indicator/strobe on Raspbery Pi cameras)
FFC2 Pin 13	MIPI_I2C3_SCL	I2C clock to camera module
FFC2 Pin 14	MIPI_I2C3_SDA	I2C data to/from camera module
FFC2 Pin 15	3V3_CAM	3.3V, derived from VEXT or 5V USB through LDO and J1. Can be switched off under s/w control.

Table 3 : Other GAP9_EVK signals mapped to expansion connectors

14. TIPS FOR PRACTICAL USE OF THE EVALUATION KIT

> Increasing I/O Drive Strength -

All GAP9 digital I/Os have programmable drive strength. At reset, they are configured into their minimum drive : 1mA or 2mA for most of them (see GAP9 I/O datasheet). In some use cases, this may be too low and it may be judicious to have the software increase drive strength.

APPENDIX

APPENDIX A - Connector Coordinates on GAP9_EVK

Figure A.1 below provides the exact coordinates of Pin 1 of each expansion connector.

The connectors all use a standard 2.54mm (100mil) pitch [1mil = 1/1000 inch = 0.0254um].

In addition, all connectors on the board are placed on a 25mil grid.

Figure A.2 provides the same information on a CAD view rather than an actual photograph.

All headers and PTH use a standard 100 mil (2.54mm) intra-connector pitch.
Their pin 1 is always on a 25mil grid – i.e. connectors are spaced by multiples of 25 mil.
[1 mil = 1/1000 inch = 0.0254 mm]

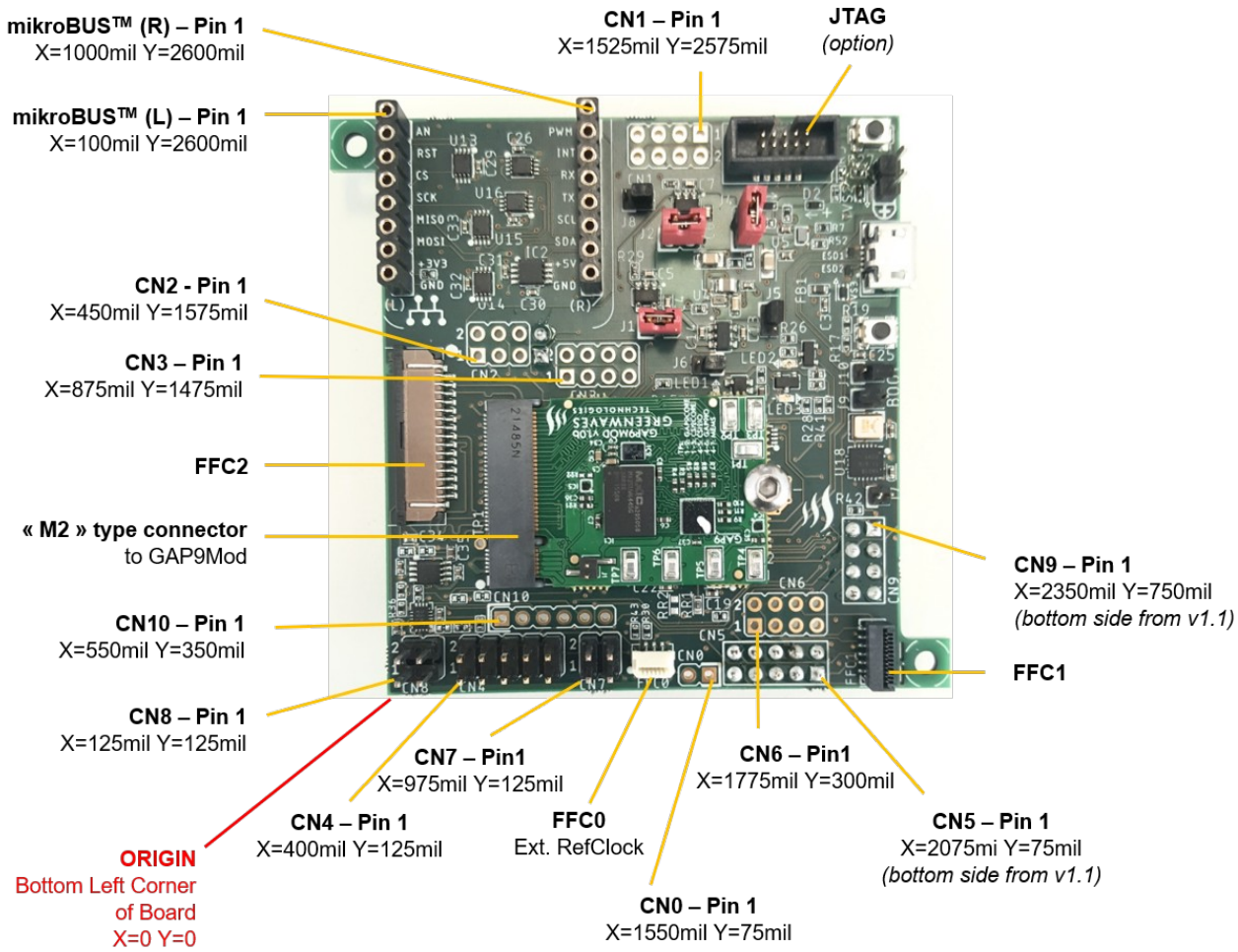


Fig. A.1 - Coordinates of connectors

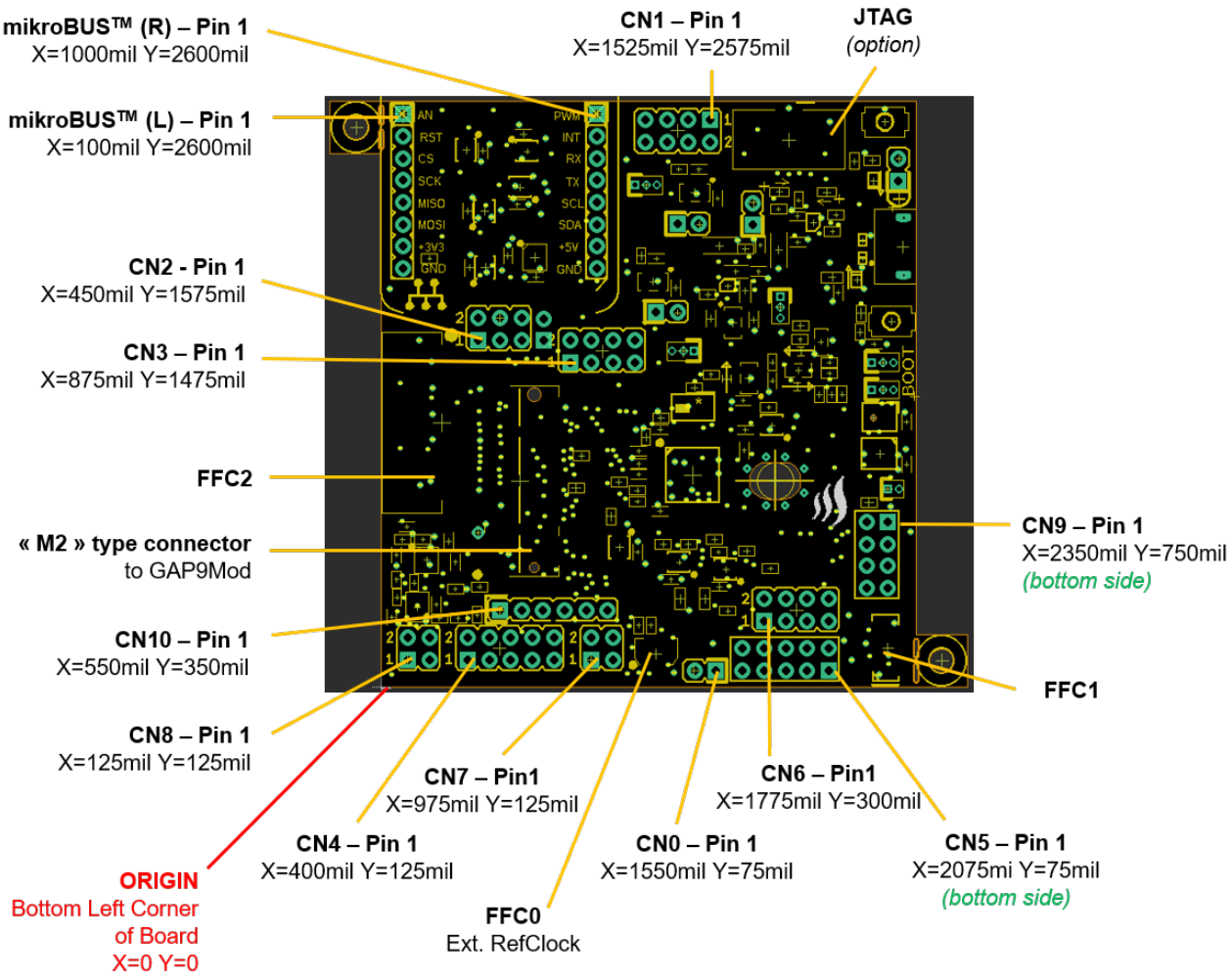


Fig. A.2 - Coordinates of connectors

APPENDIX B - GAP9_EVK Connector Usage by Audio Add-On Board

Figure B.1 below shows intended usage of selected connectors of GAP9_EVK by the Audio Add-On board

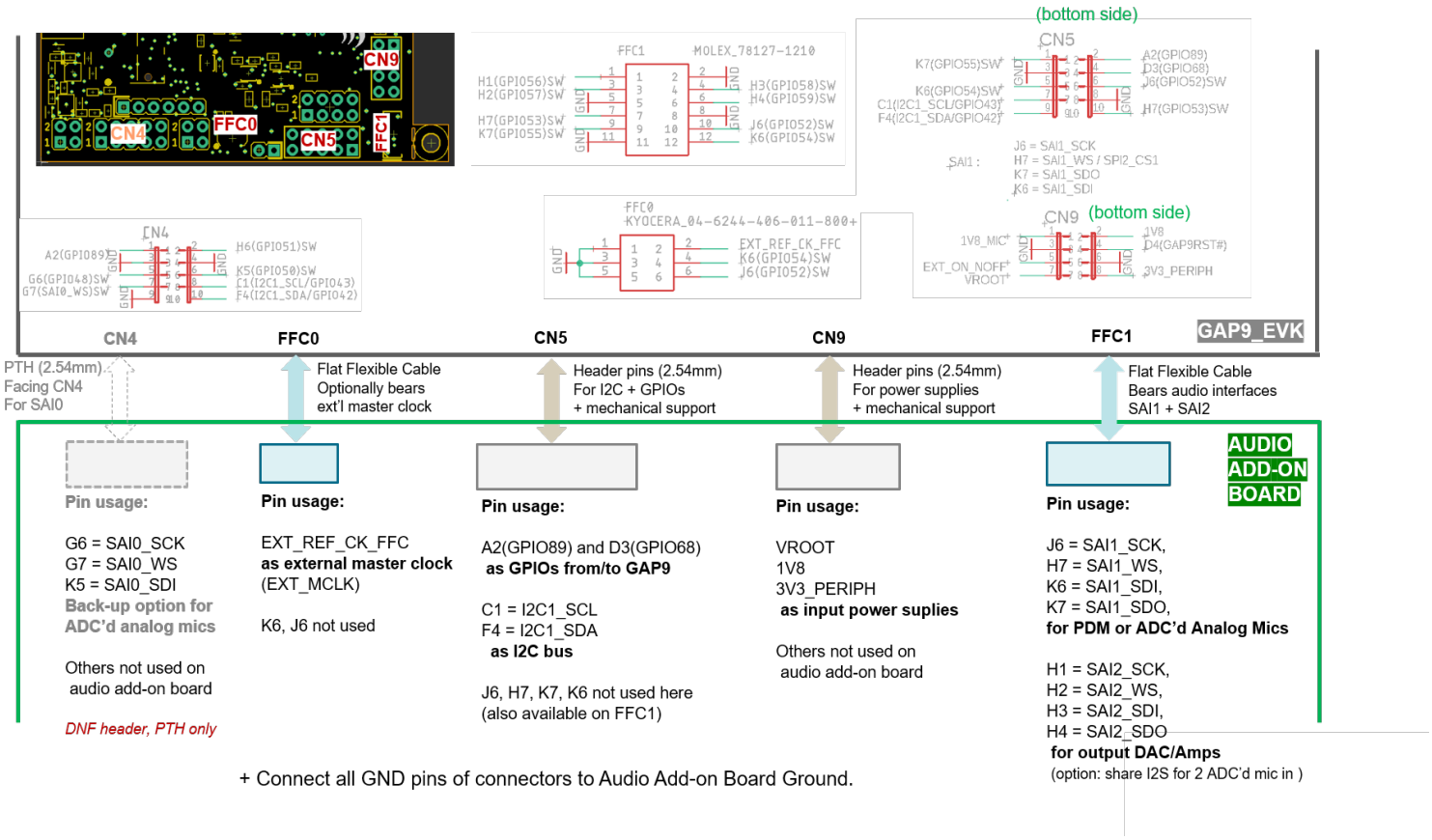


Fig. B.1 - Intended Connector Usage by Audio Add-On Board

APPENDIX C - Mapping of I2S/TDM/PDM to Serial Audio Interfaces (SAI)

GAP9 provides 3 bidirectional serial audio interfaces (SAI). Each of them can be configured to support the following :

- **I2S** (Inter-IC Sound) protocol – for transport of up to 2 **PCM** streams (i.e. stereo)

- **TDM** (Time Division Multiplex) – for transport of multiple **PCM** streams

- **PDM** (Pulse Density Modulation) – typically for connection of up to 4 digital microphones or some PDM/DSD compatible D/A converter / codec / digital amplifier

Each SAI interface involves 4 signals : SAIx_SCK (serial clock, up to 50MHz), SAIx_WS (Word Select *a.k.a* Frame Sync), SAIx_SDO (serial data out – from GAP9 point of view), SAIx_SDI (serial data in).

While the names are largely self-explanatory when mapping SAI to an I2S or TDM interface, the PDM configuration is special.

> PCM over I2S or TDM interface -

Figure C.1 outlines the connection of an I2S or TDM equipment to GAP9 using an SAI.

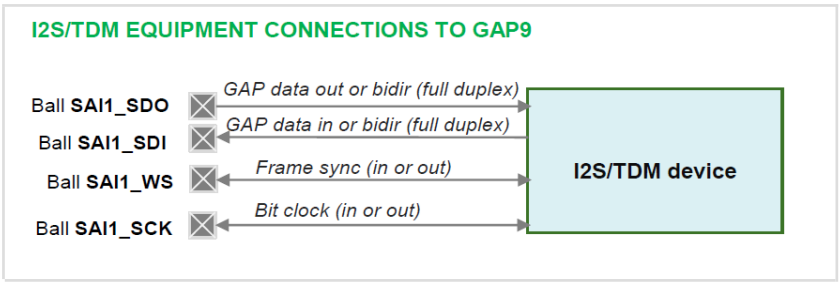


Fig. C.1. - Connection of an I2S device to, e.g., SAI1

> PDM interface -

Fig. C.2. illustrates how the internal PDM engine connects to the SAI pins.

This scheme can be used in different ways :

> PDM inputs :

- **'standard' clocked PDM mode** – with DDR (double data rate) support for L/R multiplexing, as typically used by digital microphones – so 2 data streams per PDM channel

> PDM outputs :

- **clocked PDM mode** – without DDR support i.e. 1 data stream per PDM channel

- **clockless differential mode.**

These modes are explicited in the next paragraphs.

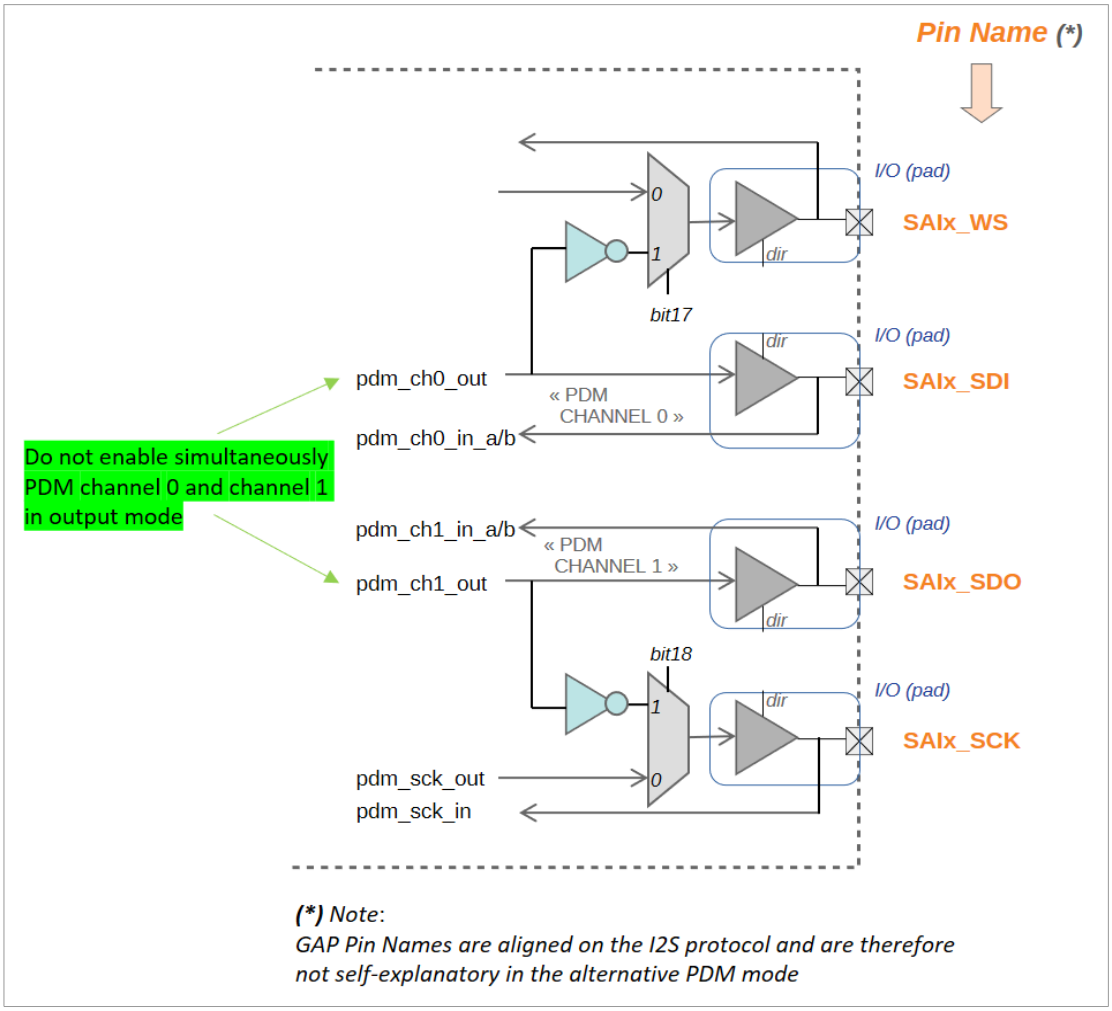


Fig. C.2 - Mapping of PDM signals to SAI interface

(1) **PDM inputs** – typ. for digital microphones

A PDM microphone typically provides its output data on one selectable clock edge of the PDM clock and tristates its data I/O on the other clock edge. This enables connecting 2 microphones, (a) and (b) (for instance, Left and Right channels) to the same data line with temporal multiplexing of the 2 channels. Fig.C.3 illustrates this.

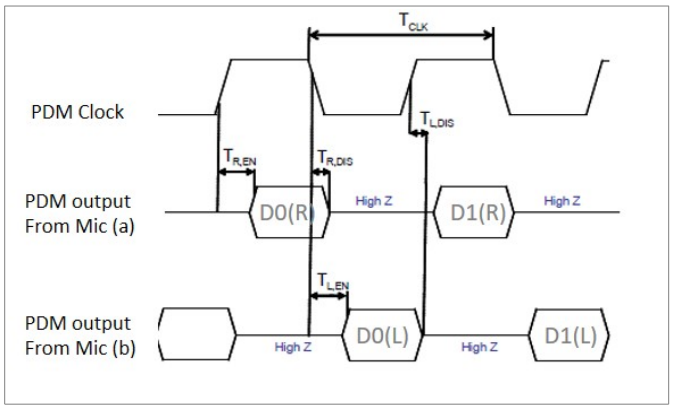


Fig. C.3 – Typical PDM clock and PDM data waveforms for a digital microphone

Two PDM input channels (each bearing 2 multiplexed data streams, so 4 data streams overall) can be born by one SAI interface of GAP9. Refer to the tables at the end of this section for pin assignment.

Fig. C.4 illustrates how up to 4 digital microphones can connect to GAP9 through one SAI interface (each microphone normally has a L/R input tied to either Logic0 or Logic1, not shown here, to decide on which phase of the PDM clock it drives data and on which phase it tri-states).

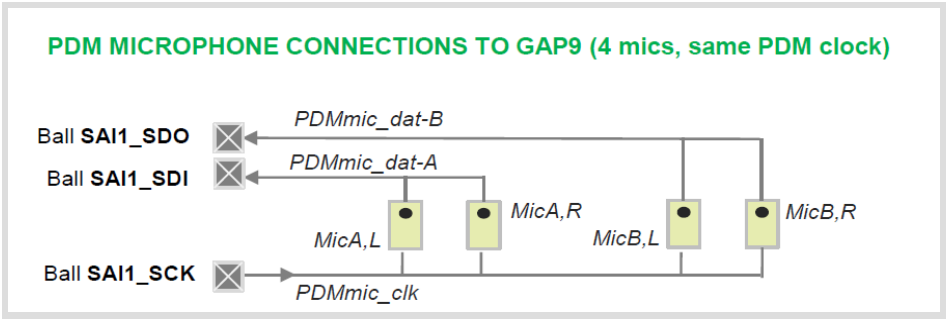


Fig. C.4. - Connection of 4 digital PDM Microphones to, e.g., SAI1

(2) PDM outputs

Two PDM output modes are supported : 'standard' clocked PDM mode (without DDR support though) and differential PDM mode.

IMPORTANT NOTE : although 2 PDM channels can technically be enabled on a given SAI channel, **only one PDM output channel should be enabled at a time**. See details in dedicated section below.

- 2.a - PDM outputs in single-ended mode

In this mode, GAP9 provides PDM ouput data on the falling edge of the PDM clock (to be sampled by the receiving device, typically an ADC, on the rising edge of the PDM clock). It does not go to high impedance state on the other clock edge nor does it perform any multiplexing. Refer to Fig.C.5. Refer to the tables at the end of this section for possible pin assignments.

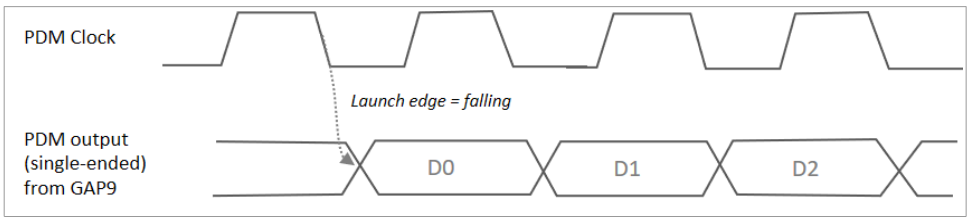


Fig. C.5 – GAP9 'standard' PDM waveforms (clock + data, single-ended)

- 2.b - PDM outputs in differential mode

'Differential PDM' can be output by GAP9 through two I/Os, one carrying a pulse-density modulated stream (sigma-delta) representative of an audio signal and the other carrying its complement. The swing is 0-1.8V, meaning common mode is 0.9V. Fig. C.6 provides an illustrative waveform.

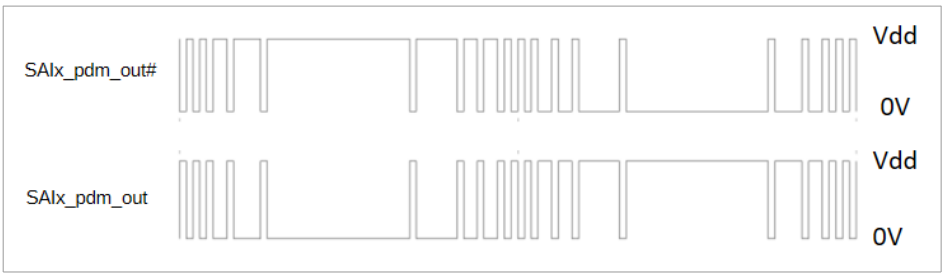


Fig. C.6– Differential PDMoutput format from GAP9

This a format specifically intended to drive a 'differential PDM Amplifier' essentially performing active analog low-pass filtering to reconstruct the analog signal encoded by the PDM pulses. One may consider the input PDM is treated as an analog differential input. No clock is required.

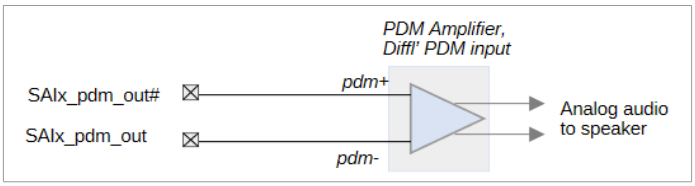


Fig. C.7 – Usage of differential PDM output

> Number of PDM channels per SAI

> In input mode, it is possible to clock in up to 2 multiplexed (e.g. Left/Right) streams, typ. from PDM microphones, on pdm_ch0_in and pdm_ch1_in – so 2 channels providing 4 PDM data streams overall after internal demultiplexing.

> In output mode, GAP9 can **output** a **single PDM channel per SAI** (be it differential or standard). Therefore, when two PDM channels need to be output (typically for stereo playback from a single GAP9), the Left channels must be output on SAIx while the Right channel must be output on SAIy, with 'x' different from 'y'.

As one PDM output channel consumes 2 lines of an SAI interface (pdm_chX_out and pdm_chX_out# in clockless differential mode, pdm_sck_out/in and pdm_chX_out in standard mode), the two other lines can still be used as PDM *inputs* (to connect e.g. 2 PDM microphones). Therefore, it is possible to map 2 PDM output channels and up to 4 PDM input channels on 2 serial audio interfaces as illustrated in Fig.C.8.a and Fig. C.8.b

One constraint though is that the PDM output and input channels sharing the same SAI must internally be clocked out/ in by the same PDM clock.

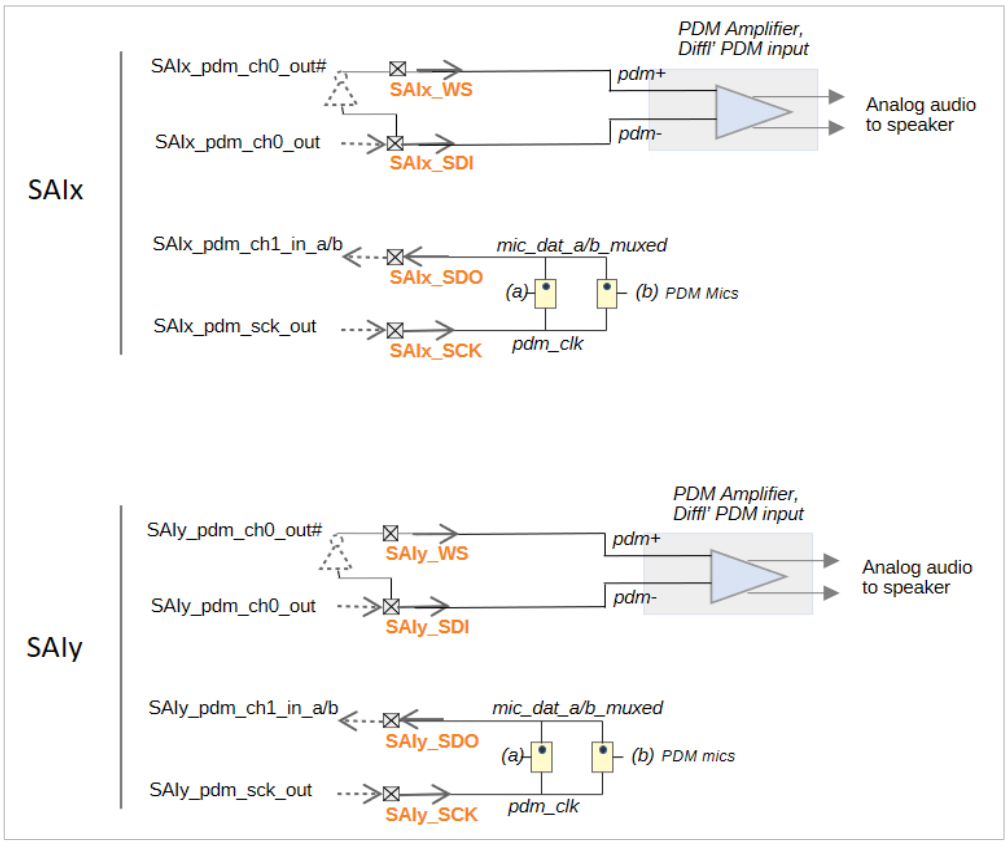


Fig. C.8.a – Mapping 2 differential PDM output channels and 4 PDM input channels onto 2 SAI of GAP9

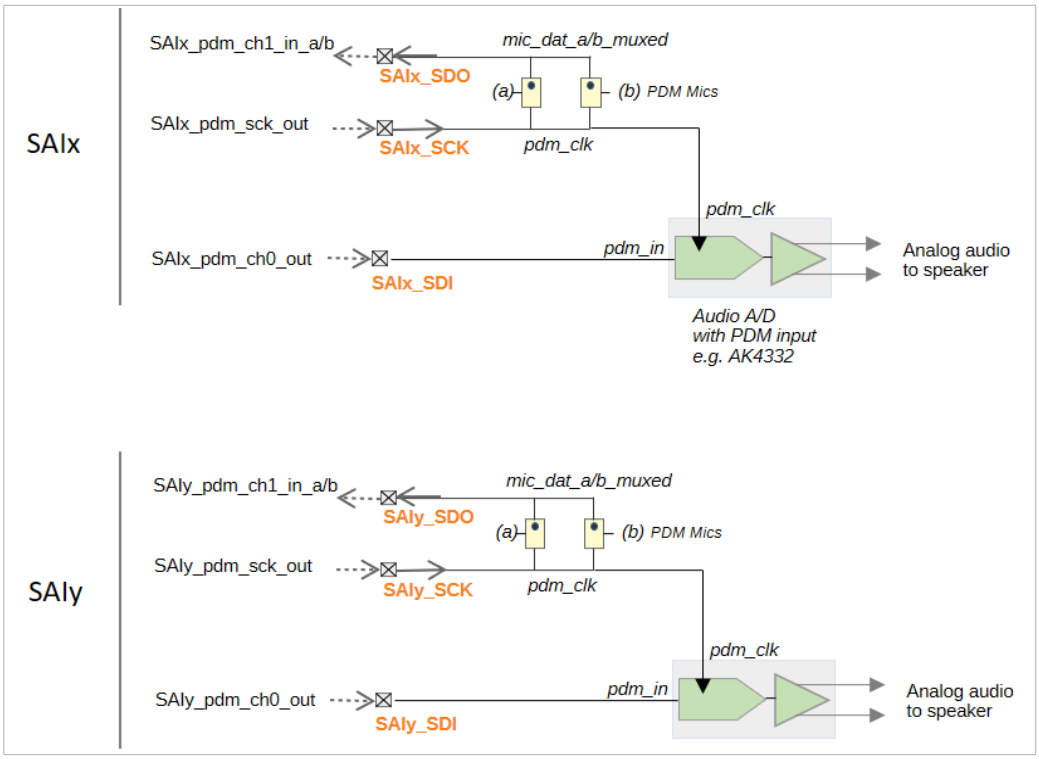


Fig.C.8.b – Mapping 2 standard PDM output channels and 4 PDM input channels onto 2 SAI of GAP9

> Summary -

The following tables recap all mapping options that **one** SAI can support :

PCM over I2S or TDM – up to 16 input or output channels (*)

GAP9 Pin Name	Pin function	Direction	Purpose
SAIx_WS	i2s-tdm_ws_in or i2s-tdm_ws_out	In (slave) or Out (master)	Audio Frame Sync
SAIx_SDI	i2s-tdm_sdi	**In**	**PCM serial data input**
SAIx_SDO	i2s-tdm_sdo	**Out**	**PCM serial data output**
SAIx_SCK	i2s-tdm_sck_in or i2s-tdm_sck_out	In (slave) or Out (master)	PCM clock

(*) up 16 TDM (or multi-channel I2S) slots supported
– with the constraint that bit clock frequency must be equal to or greater than: 'number of slots' x 'bits per slot' x 'audio sample rate'

PDM – 1, 2, 3 or 4 inputs (typ. from digital mics)

GAP9 Pin Name	Pin function	Direction	Purpose
SAIx_WS	n/a		
SAIx_SDI	pdm_ch0_in_a/b	**In**	**PDM data input(s)** channel 0, DDR support *Data_a(e.g. left)/b(e.g. right) muxed, on opposite clock phases) - typ. from 1 or 2 digital microphone(s)*
SAIx_SDO	pdm_ch1_in_a/b	**In**	**PDM data input(s)** channel 1, DDR support *Data_a(e.g. left)/b(e.g. right) muxed, on opposite clock phases) - typ. from 1 or 2 digital microphone(s)*
SAIx_SCK	pdm_sck_in *or* pdm_sck_out	In *(slave)* or Out *(master)*	PDM serial clock

PDM – 1 (single-ended) output

GAP9 Pin Name	Pin function	Direction	Purpose
SAIx_WS	n/a		
SAIx_SDI	unused		
SAIx_SDO	pdm_ch1_out	**Out**	**PDM (single-ended) output** *typ. to A/D with PDM/DSD input support*
SAIx_SCK	pdm_sck_in *or* pdm_sck_out	In *(slave)* or Out *(master)*	PDM serial clock

Alternative equivalent: swap roles of SAIx_SDI and SAIx_SDO, i.e. use **SAIx_SDI as pdm_ch0_out** and SAIx_SDO becomes unused

PDM – 1 differential output

GAP9 Pin Name	Pin function	Direction	Purpose
SAIx_WS	unused		
SAIx_SDI	unused		
SAIx_SDO	pdm_ch1_out	**Out+**	**PDM differential output** *typ. to specific 'PDM Amplifier'*
SAIx_SCK	pdm_ch1_out#	**Out-**	**PDM differential output** (complement) *typ. to specific 'PDM Amplifier'*

Alternative equivalent: swap roles of [SAIx_SCK+SAIx_SDO] and [SAIx_WS+SAIx_SDI] ,
i.e. use **SAIx_SDI as pdm_ch0_out + SAIx_WS as pdm_ch0_out#** and other pair becomes unused

PDM – 1 (single-ended) output + 1 or 2 inputs

GAP9 Pin Name	Pin function	Direction	Purpose
SAIx_WS	n/a		
SAIx_SDI	pdm_ch0_out	**Out**	**PDM (single-ended) output** *typ. to A/D with PDM/DSD input support*
SAIx_SDO	pdm_ch1_in_a/b	**In**	**PDM data inputs**, channel 1 (DDR a/b) *typ. from 1 or 2 digital microphone(s)*
SAIx_SCK	pdm_sck_in *or* pdm_sck_out	In *(slave)* or Out *(master)*	PDM serial clock *Common to input and output PDM*

Alternative equivalent: swap roles of SAIx_SDI and SAIx_SDO, i.e. use **SAIx_SDO as pdm_ch1_out** and **SAIx_SDI as pdm_ch0_in_a/b**

PDM – 1 differential output + 1 or 2 inputs

GAP9 Pin Name	Pin function	Direction	Purpose
SAIx_WS	pdm_ch0_out	**Out+**	**PDM differential output** *typ. to specific 'PDM Amplifier'*
SAIx_SDI	pdm_ch0_out#	**Out-**	**PDM differential output** (complement) *typ. to specific 'PDM Amplifier'*
SAIx_SDO	pdm_ch1_in_a/b	**In**	**PDM data inputs**, channel 1 (DDR a/b) *typ. from 1 or 2 digital microphone(s)*
SAIx_SCK	pdm_sck_in *or* pdm_sck_out	In *(slave)* or Out *(master)*	PDM serial clock *Common to input and output PDM*

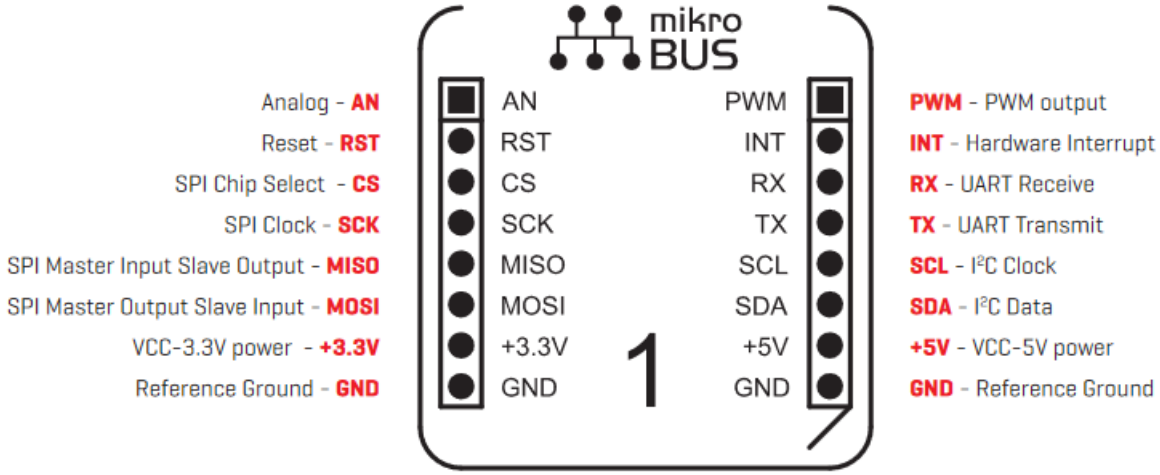
APPENDIX D - mikroBUS pin mapping

The mikroBUS™ socket is intended to plug MikroE *Click* board. It consists of 3 groups of communication pins (SPI, UART, I2C) , 6 additional pins (PWM, Interrupt, Analog Input, Reset and Chip Select) and power supplies.

All pins of the mikroBUS connect to GAP9 I/Os as per the mapping described in Table 1 of Section 13 (« Expansion Connectors »), except the analog input pin AN.

Note that UART functionality is obtained on the assigned GAP9 pins by using the IO remapping feature (see GAP9 I/O datasheet). Other functionalities (SPI, I2C…) are natively available as one of the alternate functions of the assigned GAP9 pins.

Only the analog pin, AN, of the mikroBUS connector is not connected to anything on GAP9_EVK. It can be used to connect to some external analog source or sink.



The complete mikroBUS specification can be found on MikroE's web site.

APPENDIX E - Effect of Measurement Setup on Current/Power Consumption

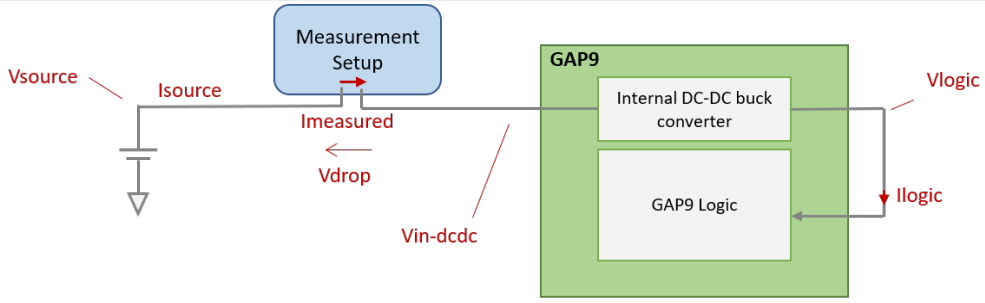
> Power Consumption at GAP9 input

Consider, as illustrated in Fig. E.1, a system where one wants to measure current (or power) at GAP9 input , using a 2-point measurement instrument (voltage source series resistor, ampere-meter, sensing resistor to measure voltage drop…). There will be some voltage drop across that measurement instrument.

Users should take care of :

- making sure the voltage at GAP9's input ($V_{in\text{-}dcdc}$) is always within its valid operating range, despite the voltage drop

- computing the effective input power at GAP9 by either removing the power lost if known (R_{sense} * I_{meas}^2 if using a sense resistor), or by measuring current and multiplying by effective $V_{in\text{-}dcdc}$ at GAP9 input, which needs to be measured too – not by the source voltage V_{source} which is not what GAP9 actually sees.



Because a DC-DC buck converter (SMPS) converts power into power ($I_{out}xV_{out}$ = Efficiency x $I_{in}xV_{in}$) :
for a given power consumption / current draw (I_{logic} above) by GAP9 core,
the current drawn by the internal DC-DC (which will be the measured current $I_{measured}$) depends on the voltage actually present at DC-DC input,
and is therefore impacted by the voltage drop across the measurement set-up.
> In other words, the *measured current* is representative of the real current consumption of GAP9 only if V_{drop} is negligible vs. V_{source} – else the measured current is over-estimated vs. what GAP9 really draws when the measurement setup is absent.
> In any case, the real *power* consumption of GAP9 is not $I_{measured}$ x V_{source}, but is $I_{measured}$ x $V_{in\text{-}dcdc}$ (again, both will be equivalent only if only if V_{drop} is negligible vs. V_{source}

Fig. E.1 – Measuring GAP9 current / power drawn from battery

Let's take look at a few numerical examples :

(1) if the voltage drop is significant w.r.t the source voltage

Assume a power source delivering 1.8V and feeding GAP9. Assume GAP9 core is running a sequence where it draws 10mA from Vlogic (0.65V-0.8V). If Vlogic is set to 0.8V, that means GAP9 core is drawing 8mW. For the sake of simplicity (this doesn't change the reasoning), assume efficiency of the internal DC-DC converter is an ideal 100 %; then the DC-DC converter will draw 8mW at its input.

In the absence of any voltage drop between the power source and measuring equipment (as would normally be the case in a real product), the voltage at DC-DC input is 1.8V and GAP9's DC-DC consuming 8mW draws 8/1.8=4.44mA from Vsource. This is GAP9's actual current/power consumption at battery level in this example use case.

However, assume now the measuring equipment causes a voltage drop of 0.3V (which is a realistic scenario for some ampere-meters) – which means the voltage drop amounts to about 17% of the source voltage . Then the voltage actually reaching the DC-DC converter is not 1.8V but 1.5V and the DC-DC consuming 8mW now draws 8/1.5 = 5.33mA, which is what the measuring equipment will indicate. In other words, the presence of the measurement equipment in these conditions causes a 20 % over-estimation of GAP9's 'real life' current consumption if the measured value is taken at face value. If looking at power rather than current : to get an accurate estimate of GAP9's power consumption, the measured value (5.33mA) must be mutiplied by the voltage actually seen by GAP9's DC-DC (5.33x1.5=8mW) and not by the source voltage (5.33x1.8mW=9.54mW which is again a 20 % over-estimation).

(2) if the voltage drop is negligible w.r.t the source voltage

Assume now a greater source voltage, at 4.2V. Assume again GAP9 core is running a sequence where it draws 10mA from Vlogic, set to 0.8V, which means GAP9 core is drawing 8mW. Assuming again an ideal 100 % efficiency of the internal DC-DC converter, it still draws 8mW at its input.

In the absence of any voltage drop between power source and measuring equipment, the voltage at DC-DC input is 4.2V and GAP9's DC-DC consuming 8mW draws 8/4.2=1.9mA from Vsource.

Assume now the measuring equipment causes a voltage drop of only 0.1V – which means the voltage drop amounts to about 2 % Then the voltage actually reaching the DC-DC converter is not 4.2V but 4.1V and the DC-DC consuming 8mW now draws a current equal to 8/4.1 = 1.95mA, which is what the measuring equipment will report. The over-estimation is now much smaller, 1.95mA instead of 1.9mA, i.e. 2.5%. If looking at power : multiplying the measured value by the source voltage (4.1V) or by the voltage actually reaching the DC-DC converter (4V) changes the result from 8.2mW to 8mW, again a 2.5% error, which may be deemed acceptable.

> Power Consumption at the output of GAP9's internal DC-DC converter – instrinsic consumption of GAP9 core

We now consider the case where we are interested in measuring the intrinsic current (or power) consumption of GAP9's core, i.e. the current which GAP9 logic core draws at the output of the internal DC-DC regulator.

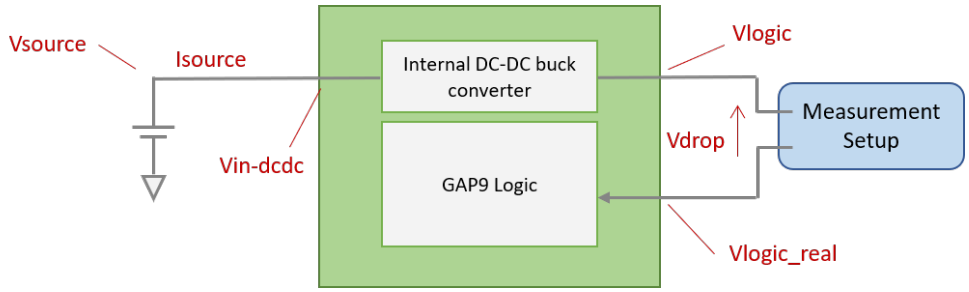


Fig. E.2 – Measuring GAP9 current / power drawn from Vlogic (supplied by internal DC-DC regulator)

If we introduce a measurement equipment between the output of GAP9's internal DC-DC regulator and the power supply pins feeding GAP9 logic core, as illustrated in Fig. E.2, we may introduce a voltage drop. This means the voltage really seen by GAP9 logic is not V_{logic} but $V_{logic\text{-}real} = V_{logic} - V_{drop}$. Therefore, it will run at a supply voltage lower than V_{logic} and will therefore consume less current than if it were operated at V_{logic} coming direct from the internal DC-DC regulator (since CMOS technology consumes less current when operated at a lower voltage, for a same operating frequency). Incidentally, this also means the maximum reachable frequency will be lower than what it would be without the voltage drop.

On GAP9Mod, 0.5-ohm sense resistors are provided across which to measure voltage drop, hence current, using Ohm's law. Considering the resulting voltage drops across the range of realistically expected currents drawn by GAP9 core, considering the measured current consumption gains for GAP9 when going from 0.8V supply to 0.65V and using linear approximation, it can be concluded that the error on measured current will remain below 5 % in most situations, and below 10 % for situations where GAP9 the highest currents from V_{logic} (approximately, above 60-80mA).

DOCUMENT HISTORY

Draft A to H – August to November,2021 (Xavier Cauchy)

Rel.1.0 – *10.Jan.2022 (XC)*
First official release
Remove prior licensing info, mark as restricted to NDA Customers

Rel.1.1 – *05.Feb.2022 (XC)*
Appendix C on SAI Mapping reworked.
- impossiblity to output 2 PDM channels onto a single SAI interface – need to dispatch 2 PDM outputs over 2 SAI
- PDM output format of GAP9 not DDR capable

Rel.1.2 – *2.March.2022 (XC)*
Add setting of BOOT pins in Quick Start guide
Fix error on CN9 pins to be shorted to get Vesper mic powered.
Add 'Known Limitations '
Add section 'Tips for practical use' and include note about I/O drive strength
Update (most) photos of GAP9_EVK to v1.2

Rel.1.3 – *4.March.2022 (XC)*
Fix pin # indicated for EXT_ON_NOFF in section 5.

Rel.1.4 – *16.Mar.2022 (XC)*
Add warning about bug on GPIO expander for CSI-2 connector I/Os + work-around

Rel.1.5 – *20.Mar.2022 (XC)*
Add warning about control of BOOT pins with J9 and J10.

Rel.1.6 – *30.Mar.2022 (XC)*
Add practical tips to use CSI-2 module and GPIO expander

Rel.1.7 – *13.Apr.2022 (XC)*
Correct mistake in pseudo-code for use of Rpi-compatible camera (section 12 - Reg.0x07 of GPIO expander)

Rel.1.8 – *20.Apr.2022 (XC)*
Indicate that pin 5 of CN3 = E3(GPIO39) cannot be used as SPIM1_SDI due to conflict with U12 which drives it too ;
Provide work-around (section 13).
Add example MIPI camera module natively compatible with GAP9_EVK's CSI-2 connector.
Specify MPN of external memories and update GAP9Mod figures regarding capacities.

Rel.1.9 – *5.May.2022 (XC)*
Discourage use of CN3 Pin5 in table 1 (due to conflict if SPI1_CS0 is asserted on U12)

Rel.2.0 – *8.Aug.2022 (XC)*
- Include information relative to GAP9_EVK v.2.0 : different FTDI chip, different BOOT jumpers
- Fix wrong pointer to reset button on Board Overview picture
- Indicate that user push button may not be usable in some cases on v1.x – fixed on v2.x

Rel.2.1 – 23.Oct.2022 (XC)
- Update GAP9Mod section for GAP9Mod 2.0 – Add setting of J2 (GAP9Mod2.0 only) in Quick Start Guide
- Add information about impact of current measurement equipment when measuring power consumption (esp.

Appendix E)
- Add recommandation to avoid using directly 5V from USB as power supply into GAP9 – rather provide 1.8V from on-board regulator.
- Add note to warn that usage of GPIO87 requires V-MikroE to be low (and is therefore exclusive with usage of mikroBUS)

Rel.2.2 – 20.Mar.2022 (XC)
- Add specificities of GAPMOd v3.0 : USB-C receptacle instead of Micro-Usb, Addition of a power on/off switch on USB power
- Rephrase recommendations for J6 and clarify it is just a shunt from v2.1 onwards.

Rel.2.3 – 13.Sept.2023 (XC)
- Indicate h/w flow control on UART to FTDI is supported from GAP9_EVK v3.1

Rel.2.3.a – 26.Feb.2024 (XC)
- Remove above feature (flow control) as there is a bug

Rel.2.4 – 7.May.2024 (XC)
- Add back option to feed GAP9 internal DC-DC with VIN to board / 5V_USB as alternative to 1.8V (stepped down on-board), through J6
- Update Fig.5 to clarify power supplies used by eMRAM

Rel.2.5 – 19.July, 2024 (XC)
- Correct inconsistencies regarding J6 (see Rel.2.4 above) – which was 3 pin header up to GAP9_EVK v2.0, then 2-pin, then again 3-pin from V3.1
- Vesper mic replaced by Infineon mic (w/o always-listening feature) from V3.2 (due to part obsolecence)

- Add known limitation : possible power failure when temperature increases