GAPPoc A

DOCUMENTATION

Rel.1.3
15-oct-2019

Table des matières

Disclaimer

This information is preliminary and subject to change.

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.

The GAPPoc design files and associated documentation are provided under the Solderpad Hardware License Version 2.0, a copy of which can be obtained at: http://solderpad.org/licenses/SHL-2.0/

1. Introduction

GAPPoc is an acronym for GAP8-centric Proof Of Concept. GAPPoc comes in different flavors for different classes of applications. The objective of GAPPoc boards is to build credible demonstrators, representative of a final application in terms of features, performance, power consumption and form factor (as far as possible, as board size and shape cannot be fully optimized independently from final product constraints). Design choices are intended to shorten time to functional POC and to ease redesign of GAPPoc to go from target application X to target application Y.

Any GAPPoc is architecture around a GAPmod *[Ref.1]* core module (which itself essentially embeds a GAP8 chip, external memory (Flash + RAM), a 32.768KHz crystal and power management for core power and crystal oscillator supplies), augmented with functionalities in line with the target applications (especially sensors and radio), on-board generation of all power supplies, plus various expansion/monitoring connectors.

2. Board Description

GAPPoc A (Fig. 1) includes:

> GAPmod1.x core module with 64Kbit RAM and 512Kbit Flash on HyperBus,
> ON Semiconductor MT9V034 (mid/high-end) Image Sensor with WVGA/VGA resolution, QVGA and QQVGA through x2/x4 down-sampling (binning), arbitrary size by cropping (FOV impacted)
> uBlox NINA-B102 Bluetooth Low Energy (BLE) module with integrated antenna and pre-Flashed firmware to support GATT profiles through AT commands and a Serial Port Service (“wireless UART”).
> Slot for a through-hole PIR *(GAPPoc A V0.1)* or ultra-low-profile connector for clipping a small satellite board controlled through SPI/I2C/UART/GPIO *(GAPPoc A V0.2 only)*
> 2/3A cylindrical battery holder (battery voltage: 3.3V min, 3.6V max, able to supply 100mA peak)
> 2.54mm pitch expansion female connectors, dual-entry
> Multiple test points to monitor on-board generated voltages and currents
> 10-pin, 1.27mm pitch male JTAG connector.

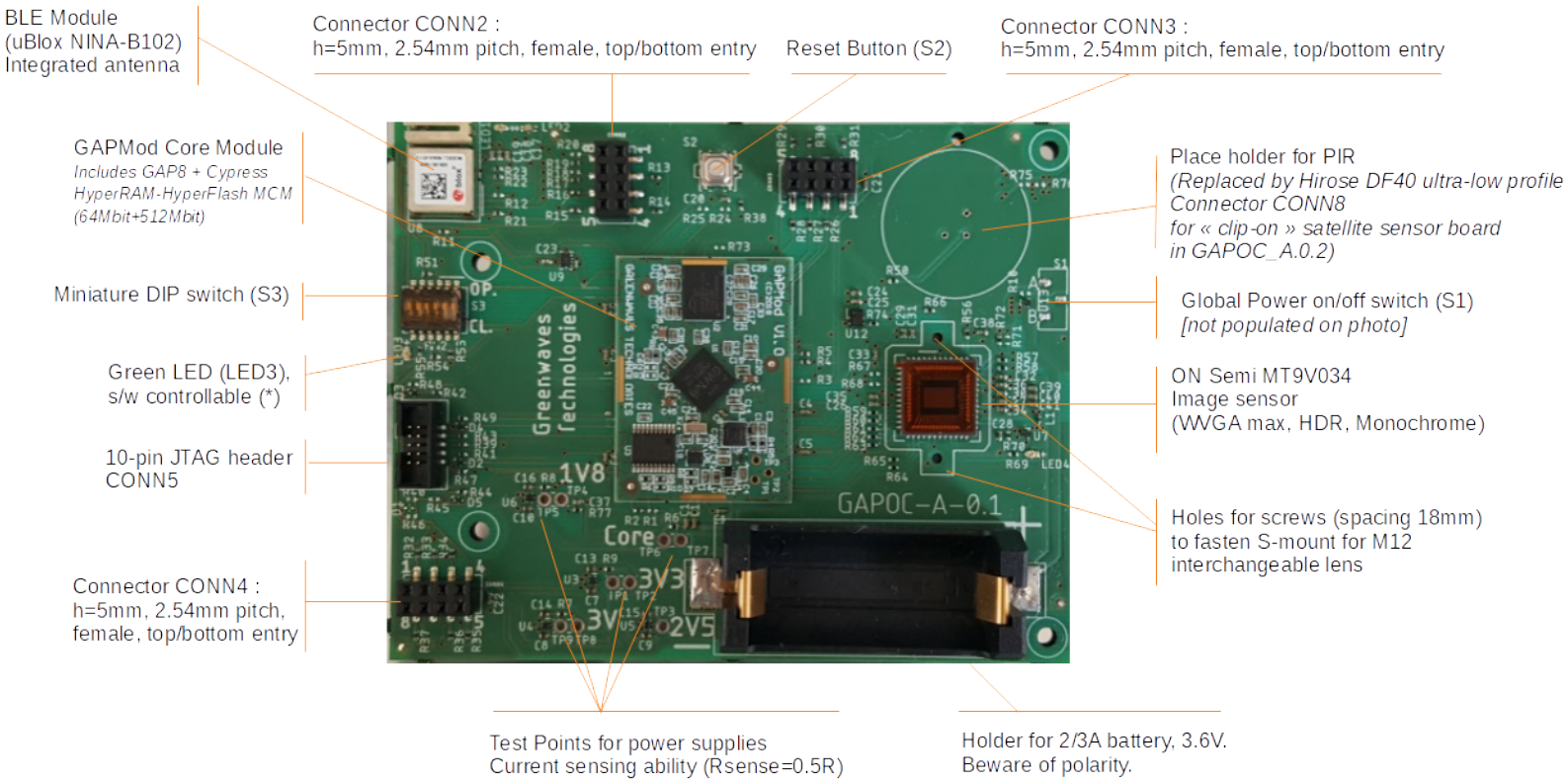

Fig. 1 – GAPPoc A Board View (V0.1) – 10cmx8cm

Programming of GAPPoc boards is achieved using an OLIMEX or compatible external probe (such as their ARM-USB-OCH-H model), plus an adapter JTAG-20-pin/2.54mm pitch to 10-pin/1.27mm pitch (ARM-JTAG-20-10) *[Ref.5]*.

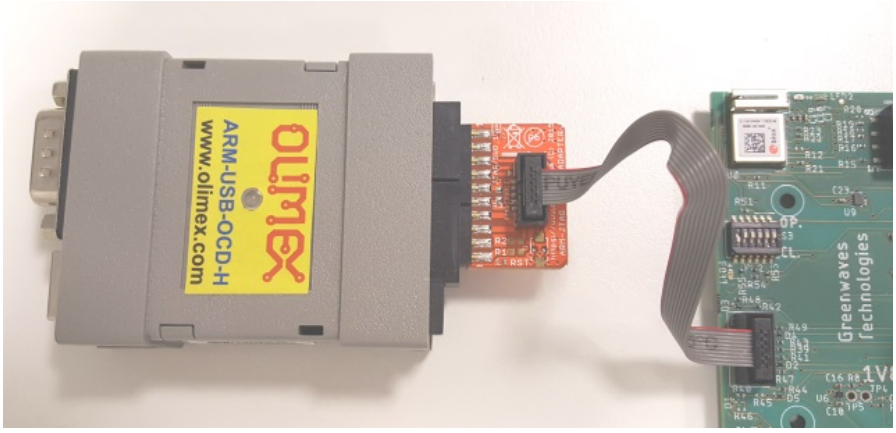

Fig. 2 – OLIMEX probe with JTAG-20-10 adapter

3. Mechanical Switches and Status LED

Manual ON/OFF switch S1:
S1 is a slider switch that completely powers on/off the board. It acts on the power supply straight out of the on-board battery. Position "I" (or 'A' on GAPPoc A V0.1) closes the switch (In service/power on), position "O" (or 'B') opens the switch (Off service/power OFF).
When battery is inserted in the holder, make sure S1 is OFF when board is not in use to maximize battery lifetime (unless GAP8 firmware is already optimized for ultra-low power sleep mode)

Manual Reset button S2:
When S2 is pushed, the global board reset NRESET is pulled low (reset asserted).

Green Status ("Heartbeat") LED3:
LED3 is a green LED that is available to the application running on GAP8. To use it, micro-switch #6 (USER_LED_CTRL) on the miniature DIP switch S3 (see above) must be closed. Then LED3 is on when pin A3 is configured as GPIO and driving a logic 1; the LED consumes in the 2-4mA range. LED3 is off pin A3 is either driving a logic 0 or high-Z.

Miniature DIP Switch S3:
The default setting (open) of all 6 individual switches of miniature DIP switch S3 is suitable for typical use cases. They may however be altered when the following options are required:

- using the on-board status LED (at the expense of some extra power consumption)
- making GAP8's UART available for external usage rather than dedicated to NINA BLE module
- allowing blowing of GAP8's embedded programmable fuses
- changing boot mode of NINA BLE module

> Pos. #1: NINA_LED_EN *[GAPPoc A V0.2 only] (*)*
Open (default): NINA_LED_EN = 0; Closed: NINA_LED_EN = 1
NINA_LED_EN controls whether status LED associated to the NINA BLE module are enabled (if 1) or always off (if 0). Refer to NINA B112 datasheet.

> Pos. #2: SEL_UART (*)
Open (default): SEL_UART = 0; Closed: SEL_UART = 1
SEL_UART controls the UART multiplexing scheme. Refer to section 6.1 for details.

> Pos. #3: 2V5_FUSE_PWRON (*)
Open (default): 2V5_FUSE_PWRON = 0; Closed: 2V5_FUSE_PWRON = 1
2V5_FUSE_PWRON controls whether 2V5 generated on-board is fed to GAP8 (needed for fuse programming). Also refer to section 6.2.
In case of conflict on 2V5_FUSE_PWRON due to incompatible settings of Pos.3 and Pos.4, the setting defined by position 4 has priority

> Pos. #4: 2V5_FUSE_PWRON (*)

Open (default): 2V5_FUSE_PWRON = 0; Closed: 2V5_FUSE_PWRON = value present on GAP8 pin B12

2V5_FUSE_PWRON controls whether 2V5 generated on-board is fed to GAP8 (needed for fuse programming). Also refer to section 6.2.

In case of conflict on 2V5_FUSE_PWRON due to incompatible settings of Pos.3 and Pos.4, the setting defined by position 4 has priority

> Pos. #5: NINA_SW2

Open (default): NINA_SW2 = 0; Closed: NINA_SW2 = 1

NINA_SW2 controls behavior of the NINA BLE module at boot. Refer to its datasheet.

> Pos. #6: USER_LED_CTRL (*)

Open (default): USER_LED_CTRL = High-Z; Closed: USER_LED_CTRL = GAP8 pin A3.

USER_LED_CTRL is the value provided to the anode (+) of the on-board green status (or "heartbeat") LED. When it's high-Z the LED stays off (no current consumption). In the other position, the LED is controlled by GAP8 pin A3; it lights up and consumes about 1-3mA when pin A3 is driven to logic 1.

Note that A3 is also available on expansion connectors – take care to avoid usage conflicts.

(*) Also refer to Section 8 on Resource Sharing

4. Generation, control and monitoring of on-board power supplies

The following figure describes how power supplies are generated from the primary battery, how some can be switched off and what sub-domains this impacts.

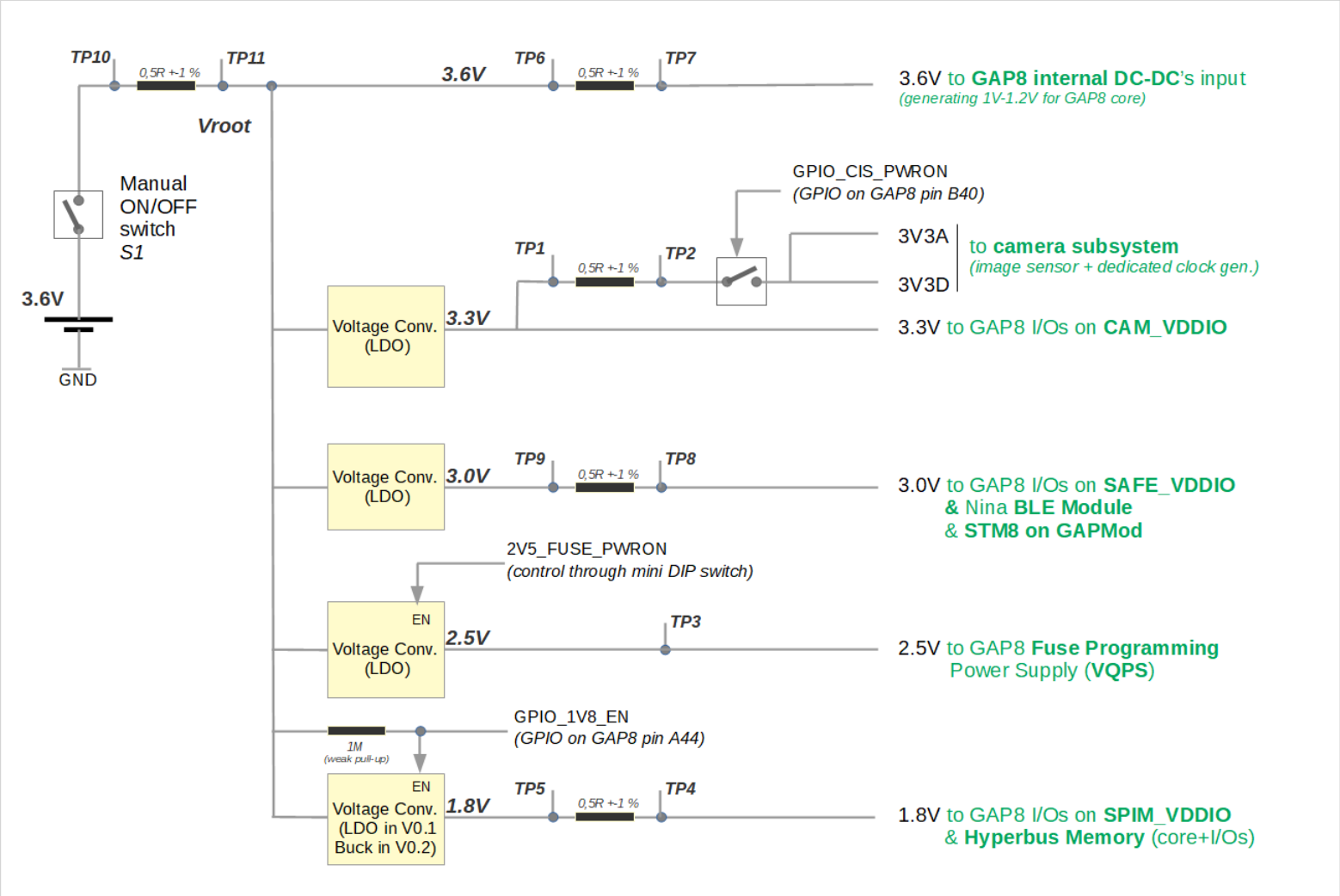

Fig. 3 – On-board power supply generation and control (notional)

– Main Power Source –

> Power can be provided to the board from a single 2/3A battery inserted in the battery holder, providing a voltage within the **3V-3.6V range**, and at least 100mA sustained current.

> Alternatively, the battery can be replaced by an external power supply with same characteristics hooked to the battery holder terminal.

> Finally, it is also possible to connect an external power supply with the same characteristics to position #4 of connector CONN4 (see section 12) for the positive side, to position #7 for the negative

(GND) side. In that case, NOTHING MUST BE CONNECTED TO THE BATTERY HOLDER TERMINALS and NO BATTERY MUST BE PRESENT in the holder. (*Note*: up to GAPOC A v0.3, the global power switch S1 does not cut power injected through CONN4 (it only cuts power from the battery holder).

– Test Points / Power Supply Monitoring –

Several test points are provided to monitor supply voltages or supply currents. To monitor current, Test Points are arranged in pairs with a 0.5ohm (+-1%) sense resistor on the current path between each element of the pair. This means the current going through the sense resistor (in mA) is twice (2x the value of the voltage drop (in mV) measured across the sense resistor (I=U/R). The sense resistor is deliberately small to avoid severe supply voltage fluctuations when there are significant instantaneous current changes.
The test points are 2.54mm apart plated holes to allow usage of standard header pins and ease measurements.

> TP1 and TP2: for on-board **3.3V** monitoring.
- power supply at output of on-board 3.3V LDO
- sense resistor between TP1 and TP2 measures current going to the Image Sensor Subsystem. GAP8 I/Os on 3.3V are excluded.

> TP3: for on-board **2.5V** monitoring.
- power supply at output of 2.5V LDO
- no current sense resistor here – as 2.5V is only used on the board for punctual GAP8 fuse programming, else disconnected all the time

> TP4 and TP5: for on-board **1.8V** monitoring.
- power supply at output of on-board 1.8V LDO (GAPPoc A V0.1) or buck DC-DC converter (V0.2)
- sense resistor between TP4 and TP5 measures current going to the full 1.8V sub-domain: HyperBus memory and GAP8 I/Os on 1.8V (notably but not restricted to HyperBus interface pins)

> TP6 and TP7: for monitoring voltage **VCore** and current provided to GAP8's internal DC-DC
- measured voltage should be that of the battery
- sense resistor between TP6 and TP7 measures current going to GAP8's internal DC-DC in charge of providing power to GAP8's core logic ("SOC" or "FC" part + "Cluster" part). It is therefore a measure of the total current drawn from the battery by the GAP8 chip (factoring in GAP8's internal DC-DC conversion efficiency), excluding its I/Os. Note that it is also possible to measure current after DC-DC conversion, drawn by SOC part or drawn by Cluster part, on the GAPmod core module itself.

> TP8 and TP9: for on-board **3V** monitoring.
- power supply at output of on-board 3V LDO (LDO fed from battery)
- sense resistor between TP8 and TP9 measures current going to the full 3V sub-domain: BLE Module (NINA), GAPMod components and GAP8 I/Os on SAFE_VDDIO=3V (notably JTAG, NRESET, I2C0).

> TP10 and TP1: for monitoring voltage **Vbat** and current provided to full GAPPoc board
- measured voltage should be that of the battery

- sense resistor between TP1 and TP10 measures total current supplied by the battery to the complete GAPPoc board. It is a measure of the total power consumption of the application.

– Software-controllable Power Switches –

> 1V8 power supply switch (GPIO_1V8_EN) :
GAP8's pin **A44** connects to enable signal GPIO_1V8_EN that can switch on (when GPIO on pin A44 is driven to 1) and off (when driven to 0) the full 1.8V domain (comprising of the HyperBus Memory and those GAP8's I/Os referenced to 1.8V on the GAPmod core module). Note – if 1.8V is switched off, all code after this operation must be fetched from internal memory only, 1.8V must be enabled back before fetching from external memory).
Note also driving the 1.8V enable signal low (1.8V off) draws about 3.6uA, due to a weak pull-up on this signal. On the other hand, an alternative option to cut power drawn from the 1.8V supply is to put the external memory in deep power-down. These factors should be taken into consideration to decide the best option.

> 2V5 power supply switch:
The 2.5V power supply is normally off. It can be turned on to program fuses in GAP8 (or possibly because a satellite board connected to CONN8 is requiring a 2.5V supply) by means of the miniature DIP switch S3 (see above).

> Image sensor's 3.3V power supply switch (GPIO_CIS_PWRON):
GAP8's pin **B40** connects to enable signal GPIO_CIS_PWRON that can switch on (when GPIO on pin B40 is driven to 1) and off (when driven to 0) the 3.3V Image Sensor sub-domain. This comprises of the 'analog' and 'digital' power supplies provided to the MT9V034 image sensor (core + I/Os) and to its dedicated cock generator (U12). However, this does **not** cover GAP8's I/Os that are on 3.3V. This allows to switch off the camera subsystem and continue using the multiple GPIOs of GAP8 that, although on the same I/O power supply as the camera interface, are typically used independently.

5. GAPmod Core Module

GAPmod is a module architectured around GreenWaves Technologies' GAP8 ultra-low power Application Processor for the IoT. GAPmod packs on a small board (*ca.* 26mmx36mm) the "invariant" part of any application based on GAP8 (GAP8 chip, external Flash + RAM memory, associated decoupling caps, crystal oscillator, passives for internal DC-DC, etc.).
GAPmod is SMT assembled on GAPPoc. GAPmod gives to the motherboard (this GAPPoc A in the present case) full and transparent access to the GAP8 chip it bears.
See *[Ref.1]* for a full GAPmod datasheet.

6. Camera (On Semiconductor MT9V034 Image Sensor)

The on-board Camera Image Sensor is the MT9V034 from ON Semiconductor in its monochrome version. It is a mid/high-end sensor with High Dynamic Range (HDR) capability. Refer to its datasheet for details *[Ref.2]*.
It is configured from GAP8 using **I2C1** interface on pins **D1** and **B34**. Pixels are exchanged over GAP8's **CPI** (Camera Parallel Interface). The MT9V034 can operate in **Master mode**, whereby it is free-running and generates by itself HSYNC and VSYNC. It can also operate in **Snapshot mode**, whereby it shoots one picture upon occurrence of a rising edge on its EXPOSURE input pin. On GAPPoc A, this pin is controlled from signal **GPIO_CIS_EXP**, driven by GAP8 GPIO available **on pin B1**.

GAPPoc is intended to be fitted with an **M12 lens mounted on an S-mount**. This solution allows lenses to be interchanged, for example to experiment with different fields of view (optical angles). GAPPoc requires an S-mount with mounting screws 18mm apart, which will go through 2 holes placed symmetrically on two sides of the image sensor. The user just must select her lens of choice and a suitable S-mount which can then be screwed onto the PCB.

Notes for writing MT9V034 software :

> The MT9V034 presents a peculiarity that is not document in its datasheet : while in non-binning mode, it provides pixel data and synchronization signals on the falling edge of the pixel clock (for sampling on pixel clock rising edge at the other end of the camera parallel interface – as expected by GAP8 in our case), in (at least some) binning modes, it provides these signals on the rising edge of the pxiel clock by default. There is a configuration bit in MT9V034 registers that allows to **invert polarity of the pixel clock**: since GAP8's camera parallel interface (CPI) samples all data on the rising edge, this bit should be used to adjust MT9V034 clock polarity when required (binning modes).

> On GAP8, data captured by the CPI is normally transferred to some memory buffer using the uDMA. Those transactions are **limited to 128KBytes**. At the time of writing, the SDK does not allow "hiding" this limitation by splitting large transactions into smaller one. Consequently, the application should take care to keep CPI-to-memory uDMA transaction size lower than 128KBytes. This is not an issue when dealing with, for instance, QVGA monochrome pictures. However, larger formats such as VGA represent over 128K pixels and therefore currently cannot be transferred with a single uDMA sequence.

7. BLE Module Control (uBlox NINA-B112)

The Bluetooth Low Energy (BLE) module implemented on GAPPoc A is a module from uBlox with integrated antenna a pre-flashed firmware supporting AT commands: NINA-B112. Refer to its datasheet for details *[Ref.3].*
> GAP8 can communicate with the NINA module through the UART interface available on pins **A7 (GAP UART Tx)** and **B6 (GAP UART Rx)**. Hardware flow control is not supported.
In addition, GAP8 can interact with NINA through the following GPIOs:
> GAP8 GPIO on pin **A2** (signal GPIOGAP_NINARST# on schematic):
can drive pin 19 = **RESET_N on NINA** module. When asserted, the module is held in reset.

> GAP8 GPIO on pin **B2** (signal LED_G/GPIOGAP_SW1 on schematic):
can drive pin 7 = **SWITCH_1 on NINA** module. The value of this pin at boot defines boot behavior, refer to NINA datasheet.
> GAP8 GPIO on pin **B13** (signal GPIOGAP_NINA17 on schematic):
can drive pin 17 = **UART_DSR on NINA** module. This input can be used for several purposes (definable through AT Commands), notably putting the NINA module in deep power-down (lowest power consumption, significantly lower than in reset).

8. Satellite Sensor Board

This section pertains to GAPPoc A v0.2 only.

A high-density (0.4mm pitch) 24-pin female (receptacle) Hirose DF40 connector is present on the board. This connector, when mated with its male counterpart, displays a **stacking height of just 1.5mm with a very firm fastening** of the two parts.

A selection of signals is available on this connector, notably supporting the following interfaces: **I2C, I2S, SPI, UART, Timer, GPIO/IRQ**. The intent is to allow enhancing a GAPPoc A V0.2 board for a specific application, by clipping on a small, custom satellite board with supplementary sensors fit for that target application – for example inertial module, microphone, time-of-flight, PIR presence detector, ambient light sensor and so on, which are based on simple interfaces such as those just listed.

The full set of signals available on the connector is listed in section 11, connector CONN8.

9. Programming / Debugging

> GAP8:
Code can be downloaded into GAP8 internal RAM or into the HyperBus Memory present on GAPmod through its JTAG interface. This is done using the **standard GAP8 SDK**, please refer to its documentation *[Ref.4].*
However, contrary to the GAPuino board also supported by the SDK, GAPPoc does not include a USB-to-JTAG converter chip. Instead, GAPPoc features a 10-pin, 1.27mm pitch JTAG interface. From a hardware point of view, this just means using an external probe. The SDK works with Olimex's ARM-USB-OCD-H. This probe is equipped with a 20-pin 2.54mm pitch connector, therefore a 20-pin to 10-pin, 2.54mm to 1.27mm pitch adapter is required *[Ref.5].*
Once this hardware is in place, the only thing to do is instructing the SDK we are using an external Olimex probe instead of an on-board FTDI chip. This can be done by simply adding the following instruction in the Makefile: `PLPBRIDGE_FLAGS += -ftdi`

> NINA BLE Module:
The NINA-B112 module comes pre-loaded with a firmware that supports AT commands, which can be used to select a proprietary "Serial Port Service" (a kind of wireless UART replacement) or standard

BLE profiles, or to define and use custom GATT profiles. **This use model does not require any firmware download into the NINA module.**
However, if desired, it is also possible to reprogram (and debug) the firmware inside NINA. This can be done using the SWD interface of the Nordic BLE chip inside NINA, which is made available on pins of connector CONN2. Alternatively, it is also possible to instruct the NINA module to download code from UART at boot by positioning specific bootstrap pins of NINA (either through expansion connectors or GAP8 GPIOs). Refer to NINA datasheet for details.

10. Multiplexed Resources – How to use them

10.1. UART Multiplexing Scheme

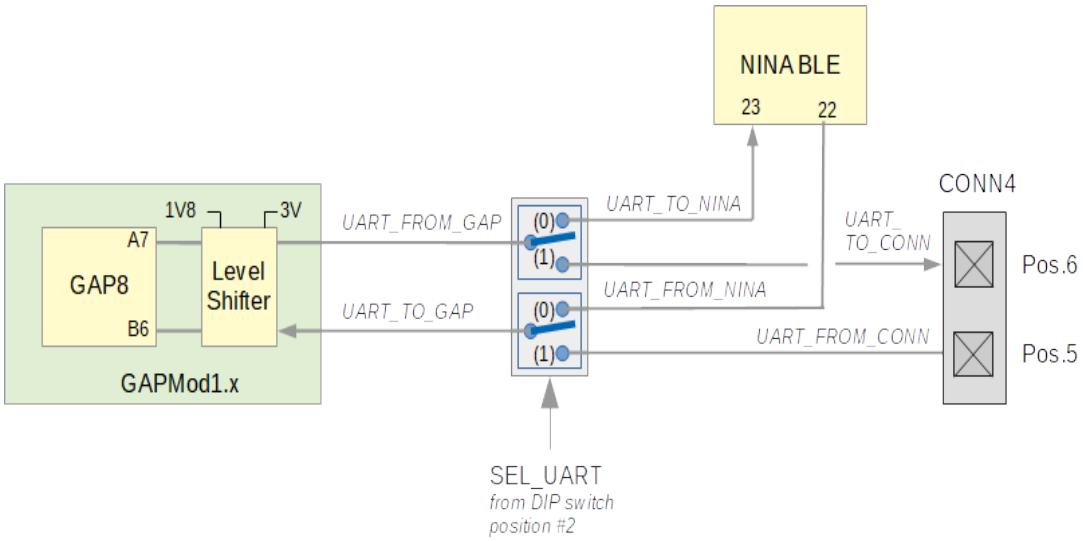

Fig. 4.a – UART Sharing scheme on GAPPoc A V0.1

*On GAPPoc A **V0.1:***
> when signal SEL_UART=0 (as defined by position of DIP switch #2): GAP8 and NINA can communicate through UART
> when signal SEL_UART=1 (as defined by position of DIP switch #2): GAP8 and a device connected to CONN4 can communicate through UART

*On GAPPoc A **V0.2:***

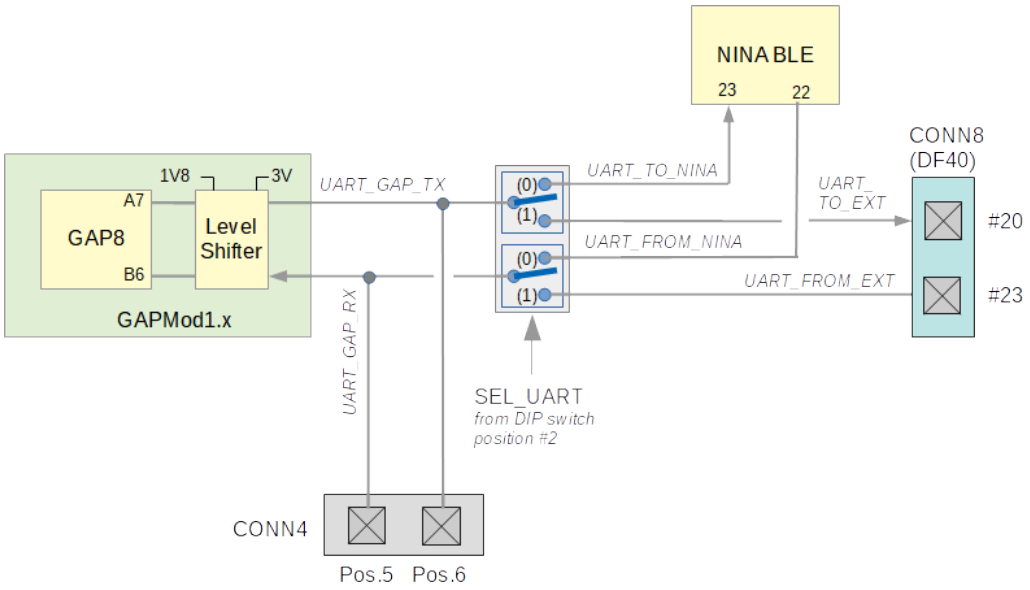

Fig. 4.b – UART Sharing scheme on GAPPoc A V0.2

> when signal SEL_UART=0 (as defined by position of DIP switch #2) – NINA selected :
- GAP8 and NINA can communicate through UART
- CONN4 can be used to monitor (e.g. with scope) the UART signals
- Alternatively, if GAP8 I/Os on (UART) pins A7 and B6 are set to high-impedance and therefore don't interfere, it is also possi

> when signal SEL_UART=1 (as defined by position of DIP switch #2) – EXT selected
- GAP8 and a satellite daughter board connected to CONN8 (positions 20 and 23) can communicate through UART. This requires that nothing is driving the UART signals on CONN4 positions 5 and 6.
- alternatively, GAP8 can communicate with an external device (such as a PC in serial terminal mode or an oscilloscope) on connector CONN4 positions 5 and 6. This requires that nothing is driving the UART signals on CONN8 positions 20 and 23.

IMPORTANT NOTES:
> There is a bi-directional level shifter on the GAPmod core module to make the UART (or GPIO alternates) signals as seen outside the core module use **0-3V levels.**
> Because of the structure of this level shifter (which auto-detects signal directions) it is **not allowed to use pull-up or pull-down resistors on input I/Os** of devices connected to the expansion connector pins involved this scheme.

10.2. Enabling 2.5V Supply for Fuse Programming

An on-board LDO generates a 2.5V power supply which can be fed to GAP8 for fuse programming. Enabling this 2.5V supply is under control of signal 2V5_FUSE_PWRON, which depends on the setting of miniature DIP switches and, optionally, an external control signal. The following figure details the control scheme.

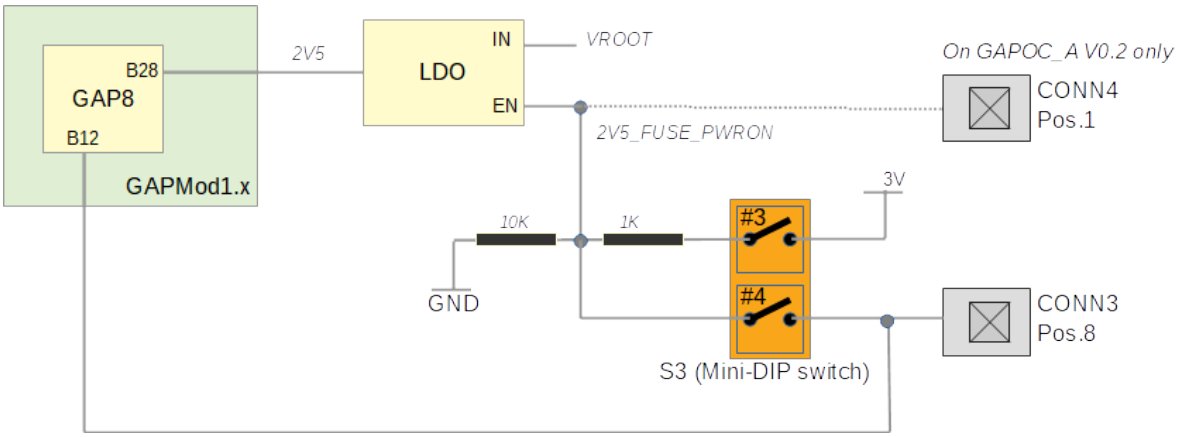

Fig. 5 – Control scheme for 2.5V power supply

> When positions 3 and 4 of the miniature DIP switch are both open 2V5_FUSE_PWRON is pulled low so 2.5V supply generation is disabled.
> To enable 2.5V power, either switch #3 or switch #4 should be closed, not both – as follows:
> If position 3 of the miniature DIP switch is closed: 2V5_FUSE_PWRON is pulled high so 2.5V supply generation is enabled.
> If position 4 of the miniature DIP switch is closed: 2V5_FUSE_PWRON can be controlled either from a GPIO on pin B12 of GAP8 (do not drive CONN3 Pos.8 in that case) or from CONN3 Pos.8 (make GAP8 pin B12 high-Z in that case)

> *On GAPPoc A V0.2 only:* the enable signal can be controlled directly from CONN4 Pos.1. This leaves B12 available on CONN3 Pos. 8 for other uses.
==> DIP switch #4 is redundant in that case and should be kept open to eliminate any risk of conflict between CONN 4 Pos.1 and CONN3 Pos. 8.

10.3. Using the Status LED (Green LED, LED3)

The on-board green status LED can be controlled from the GPIO available on GAP8 pin A3. However:
> by default, this feature is disabled, and the LED is not driven. In that mode, pin A3 can be freely used to interact with the outside world through position 2 of connector CONN3.
> If the user closes position 6 of the miniature switch S3, then LED3 is driven from GAP8 pin A3. In this mode, the user should not connect anything to position 2 of connector 3 to avoid any risk of conflict.

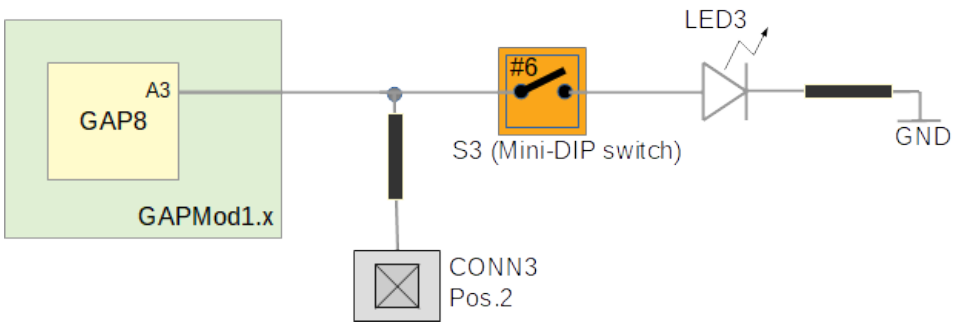

Fig.6 – Sharing of GAP8 pin A3: external connector vs. on-board green LED

11. Board I/Os: Connector Pin Assignments

Note*: different I/Os may use different logic levels, depending on what partition of GAP8's I/O ring they belong to. See GAP8's and GAPmod's datasheets [Ref.1].*

Connector CONN4:
Pos. #1 – **2V5_FUSE_PWRON** control signal (*)
Pos. #2 – GAP8 pin **A25** [I2C – *see GAP8 datasheet*]
Pos. #3 – GAP8 pin **B22** [I2C]
Pos. #4 – **VROOT** typ.3.6V from Battery to GAP8 and on-board DC-DC converters
Pos. #5 – GAP8 pin **B6** [UART_GAP_RX/GPIO], voltage translated for 0-3V swing (*)
Pos. #6 – GAP8 pin **A7** [UART_GAP_TX/GPIO], voltage translated for 0-3V swing (*)
Pos. #7 – **GND**
Pos. #8 – **PB2** [*see GAPmod datasheet – Not Connected on GAPMod1.2*]
(*) Also refer to section xx on Resource Sharing

Connector CONN3:
Pos. #1 – **3.3V** from on-board LDO
Pos. #2 – GAP8 pin **A3** [**SPI/GPIO** – *see GAP8 datasheet*]
Pos. #3 – GAP8 pin **B3** [**SPI/GPIO**]
Pos. #4 – GAP8 pin **A4** [**SPI/GPIO**]
Pos. #5 – GAP8 pin **B4** [**SPI/I2C/GPIO**]
Pos. #6 – GAP8 pin **A5** [**SPI/I2C/GPIO**]
Pos. #7 – **GND**
Pos. #8 – GAP8 pin **B12** [**TIMER/GPIO**]

Connector CONN2:
This connector is mostly dedicated to NINA BLE module debug/diagnosis
Pos. #1 – **3V** from on-board LDO
Pos. #2 – **NINA_SWCK**
Pos. #3 – **GND**
Pos. #4 – **NINA_IO3**
Pos. #5 – **NINA_SWDIO**
Pos. #6 – **NINA_RST#**
Pos. #7 – **NINA_IO2**
Pos. #8 – **NINA_IO5**

JTAG Connector CONN5:
Pos. #1 – **3V** from on-board LDO
Pos. #2 – **JTAG_TMS** *(on-board 10K pull-down)*

Pos. #3 – **GND**
Pos. #4 – **JTAG_TCK** *(on-board 10K pull-down)*
Pos. #5 – **GND**
Pos. #6 – **JTAG_TDO**
Pos. #7 – **SWIM** [*see GAPmod datasheet – Not Connected on GAPMod1.2*]
Pos. #8 – **JTAG_TDI** *(on-board 10K pull-down)*
Pos. #9 – **NC**
Pos. #10 – **NRESET** [system reset]

On GAPPoc A V0.2 only -
Connector CONN8:
Pos. #1 – **GND**
Pos. #2 – GAP8 pin **A3** [**SPI/GPIO** – *see GAP8 datasheet*]
Pos. #3 – GAP8 pin **B3** [**SPI/GPIO**]
Pos. #4 – **NC** (Reserved for Future Use)
Pos. #5 – GAP8 pin **A4** [**SPI/GPIO**]
Pos. #6 – GAP8 pin **B12** [**TIMER/GPIO**]
Pos. #7 – **GND**
Pos. #8 – **1V8** from on-board DC-DC converter
Pos. #9 – GAP8 pin **A5** [**SPI/I2C/GPIO**]
Pos. #10 – GAP8 pin **B4** [**SPI/I2C/GPIO**]
Pos. #11 – **3V** from on-board LDO
Pos. #12 – **GND**
Pos. #13 – **GND**
Pos. #14 – **2V5** from on-board LDO – *if signal 2V5_FUSE_PWRON is set (*)*.
Pos. #15 – GAP8 pin **B23** [**I2S**]
Pos. #16 – GAP8 pin **A26** [**I2S**]
Pos. #17– GAP8 pin **A24** [**I2S**]
Pos. #18 – **3V3** from on-board LDO
Pos. #19 – GAP8 pin **A13** [**TIMER/GPIO**]
Pos. #20 – GAP8 pin **A7** [**UART/GPIO**], voltage translated for 0-3V swing (*)
Pos. #21 – GAP8 pin **A25** [**I2C**]
Pos. #22 – GAP8 pin **B22** [**I2C**]
Pos. #23 – GAP8 pin **B6** [**UART/GPIO**], voltage translated for 0-3V swing (*)
Pos. #24 – GND
(*) Also refer to section xx on Resource Sharing

On GAPPoc A V0.1 only -
3 thru-holes for Panasonic PIR sensor EKMB series.

REFERENCES

[Ref.1] - GAPmod Datasheet and Schematic – Available from GreenWaves Technologies
GAP8 Datasheet "*GAP8 Hardware Reference Manual*" – available from *greenwaves-technologies.com*

[Ref.2] - ON Semiconductor MT9V034 Datasheet
"1/3-Inch Wide-VGA CMOS Digital Image Sensor"
https://www.onsemi.com/pub/Collateral/MT9V034-D.PDF

[Ref.3] - uBlox NINA-B112 Datasheet
"NINA-B1 Series – stand-alone Bluetooth low energy modules – data sheet"
https://www.u-blox.com/sites/default/files/NINA-B1_DataSheet_%28UBX-15019243%29.pdf

[Ref.4] - GreenWaves GAP8 SDK & Manuals: greenwaves-technologies.com/en/sdk/

[Ref.5] - OLIMEX ARM-USB-OCD-H USB to JTAG/RS232 converter
https://www.olimex.com/Products/ARM/JTAG/ARM-USB-OCD-H/
ARM JTAG 20-pin/2.54mm pitch to 10-pin/1.27mm pitch converter (multiple vendors)

DOCUMENT HISTORY

Draft A to C – Jan-2019 (XC/MC):
Initial draft followed by cosmetic changes

Draft D – 22-jan-19 (XC)
Clarified that GAP UART is also available on connector CONN4

Draft E – 01-Feb-19 (XC)
Added precautions for camera interface s/w in section on MT9V034

Rel 1.0 – 23-Apr-2019 (XC)
Changed markings from Draft to Rel.1.0

Rel. 1.1 – 5-Jun-2019 (XC)
Added note in STM8 section about behavior of blank (never programmed) STM8.

Rel. 1.2 – 2-Jul-2019 (XC)
Added information about power sources in section 4.
Fixed a mistake in UART muxing scheme description.

Rel. 1.3 – 15-oct-2019 (XC)
Remove information related to older GAPMod (pre-1.2)

