Let's build a small 2-layer PCB design as a break-out board for the calypso/iota/rita based GTM900-B module.

It should have the following key features:

- on-board power supply for (5V?) input
- over-voltage + polarity protection
- 2.5mm jack for the UART (+ TVS array)
- expose all pins of the module on 2.54mm header
- LED for power and modem (is there one?)
- power button and jumper in parallel (auto-start)
- reset button
- SIM card slot (2FF) with TVS array

One option would also be to directly put a CP2102 or even CP2105 on the board to attach to one (or both) UARTs. This way we'd also have the option to control reset and/or power via GPIOS or serial status lines, like tsaitgaist used to do..

History

#1 - 05/28/2019 10:22 PM - laforge

The shortest 40pos 0.5mm pitch FPC available from Digikey seems to be https://www.digikey.com/product-detail/en/molex/0151660425/WM16424-ND/3281306 at 29.97mm.

#2 - 06/05/2019 07:56 PM - mschramm

GTM900 exposes no LED at all, so the breakout only will get a power-on LED.

Should the modem reside on the break-out PCB like on the mPCIe adapter, e.g. with hex spacer, or will this board be much smaller for only carrying the additional functionality and not the modem itself?

#3 - 06/05/2019 08:03 PM - fixeria


#4 - 06/05/2019 08:31 PM - laforge

- Priority changed from Normal to Low

the modem should be mounted with spacers on the board. I typically want to have a mechanically stable configuration, without any pigtails or other fragile constructs hanging off it. That's why we have the SMA sockets on the mpcie breakout board, and why I believe we should have a similar construct here.

#5 - 06/05/2019 08:51 PM - mschramm

mschramm wrote:

GTM900 exposes no LED at all,
this might be not true, there are two pins with funny names and ambiguous descriptions, so likely one of them (or both?) might drive a LED (plus additional driver):

Pin 13 is VDD (output), and its function "signal indication of normally started GTM900". The other is pin 32, LPG (output), "mainly controls the status of the indicator".

mschramm wrote:
Pin 13 is VDD (output), and its function "signal indication of normally started GTM900". The other is pin 32, LPG (output), "mainly controls the status of the indicator".

LPG is "LED pulse generation", basically dedicated timer circuitry to let an LED blink. Currently it is configured to have the LED always on, which works on the GTM900 on the existing breakout-board from China.

just committed an approach for a GTM900 breakout last week, and first (internal) feedback came from laforge, who wanted the CP210x and a decent audio connection w/o pin header contacts, - which both are now in place too. Some routing is not yet done because of open questions.

- I left in a mirrored version of the FPC receptacle representing the modem connector; it will get removed.
- level translation on that board is made by the CP2105 USB dual UART for both serial ports (now bus-powered)
- do we want to care about Vbackup? e.g. bringing this signal to the aux/ expansion header?
- RESET now is a tact switch, and PWRON a jumper. Do we want momentary switches on both?
- the initial AUX header now is gone; now we only would need it for the 2nd audio channel and the ADC input. Do we need either/both/none? THT header or test pads?

remarks on the audio section:

- schema is similar to that in the chinese datasheet
- I prefer the differential design, but it makes ESD design little more complicated (more TVS; TVS tbd)
- electret biasing is derived by VCC which originates from the modem; we hence might want an LC lowpass towards the mic
- the audio jack now is a 3.5mm TRRS, whereas the datasheet shows a RJ10 modular jack on that position - which receptacle do we want? If TRS or TRRS: what pin assignment (cause there are more than one concurrent...) ?

Let me know your feedback!
I actually thought a CP2102 (single-port) only for one of the two UARTS, while the other one is on the normal osmocom style 2.5mm jack. But putting both UARTs on a CP2105 also seems fine to me, to be honest.

- do we want to care about Vbackup? e.g. bringing this signal to the aux/ expansion header?

I think it may make sense if one wanted to play with the power saving features. At least have a connector, or possibly even have a footprint for a DNP battery holder?

- RESET now is a tact switch, and PWRON a jumper. Do we want momentary switches on both?

I think reset as a button (taster) and PWRON on a jumper (default: on) looks good to me.

I think tsaitgaist once had a "DTR to power on calypso" device, so maybe we also have a jumper (and transistor?) to enable this mode of operation?

- the initial AUX header now is gone; now we only would need it for the 2nd audio channel and the ADC input. Do we need either/both/none? THT header or test pads?

In general I would put all unused signals on test pads.

- the audio jack now is a 3.5mm TRRS, whereas the datasheet shows a RJ10 modular jack on that position - which receptacle do we want? If TRS or TRRS: what pin assignment (cause there are more than one concurrent...)?

I would definitely go for one (or two) 3.5mm jacks for the audio, assuming that people can use the kind of headsets that you would attach to audio jacks of a PC or laptop. If a 4pin 3.5mm is used, let's use what laptop jacks e.g. of lenovo tend[ed?] to use.

#14 - 04/28/2020 04:10 PM - laforge

- Checklist item [ ] UART A added
- Checklist item [ ] UART B added
- Checklist item [ ] Audio on 3.5mm jack[s] added
- Checklist item [x] power supply added
- Checklist item [ ] mechanical mounting of GTM module added
- Checklist item [ ] power button control via modem handshake (optional via jumper) added
- Checklist item [x] mounting holes of breakout board added

#15 - 04/28/2020 06:45 PM - falconia

FYI, I already made my own breakout/adapter board for GTM900 modules (called MMTB1) several months ago:

https://www.freecalypso.org/pipermail/community/2020-January/000729.html

But my MMTB1 is much simpler than what you are doing here; your version is more complex with built-in USB-serial and taking 5V in rather than ready-made VBAT. My MMTB1 brings out the same interfaces as FCDEV3B, but it looks like your preferences are different.

In technical terms, I see only two problems with your current version, based on your PDF schematics posted in this ticket:

1) You have the GTM900 modem's UART_RI line connected to CP2105 SUSPEND/RI_SCI. The modem's UART_RI line is actually Calypso GPIO 0, but Huawei's official firmware configures this Calypso GPIO as an output, and my FreeCalypso firmware does likewise. But according to CP2105 datasheet, pin 1 (SUSPEND/RI_SCI) acts as the SUSPEND output by default, unless you switch it to RI_SCI or some other function via OTP configuration. Having CP2105 and Calypso outputs fight seems like a bad idea to me, and I would be concerned about possibly damaging one of the chips. So if you are going to keep this wiring, you should do the appropriate OTP configuration on the CP2105 to change that pin function. Yes, it sucks that CP2105 configuration is OTP instead of EEPROM: you only get one chance to program it, and if you mess it up, you have to either throw away the board or try to replace the OFN chip, which would be a major pita. For this reason, if you are going to keep the wiring that calls for OTP configuration change, you should do the programming at your factory, rather than leave it to end users. (This OTP nonsense is one of the many reasons why I am sticking with FT2232D for my projects and have no desire to switch to CP2105, but it is obviously not my place to tell you what to do in your business.)

2) MIC- and MIC+ microphone input signals on the GTM900 FPC interface are NOT raw Iota MICIN & MICIP, instead GTM900 already incorporates capacitive coupling and ECM bias injection circuits inside the module. Therefore, your external ECM bias injection circuit is wrong and should be removed. The circuit depicted in the Chinese document resides inside the GTM900 module itself, it is not something you need to build externally. The module's VDD output is the internal V-IO power rail, produced by Iota regulator VRIO - it is a digital (thus noisy) supply rail, not for anything analog. For the ECM bias the Iota chip has a dedicated MICBIAS analog voltage output, but this Iota output is not brought to the outside of GTM900, instead that whole circuit is implemented inside the module.

#16 - 04/29/2020 06:50 AM - laforge

Dear Mychaela,
many thanks for your input, it is - as always - much appreciated.

On Tue, Apr 28, 2020 at 06:45:39PM +0000, falconia [REDMINE] wrote:

FYI, I already made my own breakout/adapter board for GTM900 modules (called MMTB1) several months ago:

I am aware of this board, but indeed I found it somewhat lacking in features for the kind of use case I have in mind. It seemed more like a 'raw breakout', i.e. a mechanical adapter fo the FPC to other pins/connectors, than to try provide as much as possible in terms of an environment that's easy to use without building further hardware.

This is not a complaint at all. It's just that our goals are (as it seems customary by now) somewhat different :) 

In technical terms, I see only two problems with your current version, based on your PDF schematics posted in this ticket:

1) You have the GTM900 modem's UART_RI line connected to CP2105 SUSPEND/RI_SCI. The modem's UART_RI line is actually Calypso GPIO 0, but Huawei's official firmware configures this Calypso GPIO as an output, and my FreeCalypso firmware does likewise. But according to CP2105 datasheet, pin 1 (SUSPEND/RI_SCI) acts as the SUSPEND output by default, unless you switch it to RI_SCI or some other function via OTP configuration.

Thanks for pointing that out. I would assume we can do the OTP programming - but we'd have to do it in a safe way, i.e. at a time before the GTM900B is connected.

Factory pre-programming the CP2105 is not really an option for us, the quantities are too low to rectify that. But we can of course do the programming after component placement before ever placing the modem on the board, and long before it ever gets into the hand of a user.

As a "plan B safeguard" I suggest we put a 0-ohm resistor into the line, so if for some reason we are in trouble, we can interrupt the signal without having to resort to cutting copper traces with a scalpel.

2) MIC- and MIC+ microphone input signals on the GTM900 FPC interface are NOT raw Iota MICIN & MICIP, instead GTM900 already incorporates capacitive coupling and ECM bias injection circuits inside the module. Therefore, your external ECM bias injection circuit is wrong and should be removed.

Interesting, thanks. I'll leave it to @weef to investigate and act accordingly.

Regards,
Harald

#17 - 04/29/2020 07:27 AM - falconia

laforge wrote:

I am aware of this board, but indeed I found it somewhat lacking in features for the kind of use case I have in mind. It seemed more like a 'raw breakout', i.e. a mechanical adapter fo the FPC to other pins/connectors, than to try provide as much as possible in terms of an environment that's easy to use without building further hardware.

I would not call my MMTB1 a "raw breakout", instead it is an intermediate form between a raw breakout and what you are building. A true raw breakout is what Songbosi sent me originally: all 40 pins of the FPC connector wired 1-to-1 to 2.54 mm header pins. That raw breakout was truly difficult to work with, so I built my MMTB1 as a replacement. MMTB1 has an on-board SIM socket and on-board pushbuttons for PWON and RESET, so it is not just going from FPC to other connectors. And I wasn't building any further hardware, instead I simply reused the same power supply and dual UART adapter which I already had for FCDEV3B.

This is not a complaint at all. It's just that our goals are (as it seems customary by now) somewhat different :) 

Yes, of course we have different goals as usual, but if you do produce this breakout board you are currently building and make complete GTM900-B kits available in your webshop, then it would help me too: it would provide one more low-entry-barrier path for potential users to play with my FreeCalypso firmware.
Thanks for pointing that out. I would assume we can do the OTP programming - but we'd have to do it in a safe way, i.e. at a time before the GTM900B is connected.

Yes, that was my thinking as well.

Factory pre-programming the CP2105 is not really an option for us, the quantities are too low to rectify that.

I didn't mean Silabs chip factory preprogramming, instead I meant your own Sysmocom small hw factory - just like my FreeCalypso hw factory is a converted bedroom where I have my CMU200 and other production test gear, I assumed that you have something similar.

But we can of course do the programming after component placement before ever placing the modem on the board, and long before it ever gets into the hand of a user.

Good, this is exactly what I was asking for.

As a "plan B safeguard" I suggest we put a 0-ohm resistor into the line, so if for some reason we are in trouble, we can interrupt the signal without having to resort to cutting copper traces with a scalpel.

Seconded!

[MIC interface]

Interesting, thanks. I'll leave it to @weef to investigate and act accordingly.

Just to reiterate, the MIC- and MIC+ pins that come out of the GTM900 on the FPC interface are meant for direct connection of an electret condenser microphone. On my MMTB1 I have these two signals routed to a 2-pin 2.54 mm header without any extra glue components, and I can connect an ECM to those two pins exactly the same as on FCDEV3B. In the case of FCDEV3B I have copied the microphone interface circuit (ECM bias injection and capacitive coupling to Iota MIC inputs) from TI's Leonardo schematics, but GTM900 already has this exact same circuit built-in inside the module.

#18 - 04/29/2020 09:29 AM - laforge

On Wed, Apr 29, 2020 at 07:27:18AM +0000, falconia [REDMINE] wrote:

Factory pre-programming the CP2105 is not really an option for us, the quantities are too low to rectify that.

I didn't mean Silabs chip factory preprogramming, instead I meant your own Sysmocom small hw factory - just like my FreeCalypso hw factory is a converted bedroom where I have my CMU200 and other production test gear, I assumed that you have something similar.

It still is too much effort. That would mean we'd have to have a ZIF socket for this specific QFN package, and remove every chip from the tape/reel, then re-reel it before passing it to the SMT house. Even if a tray package was available, it's too much work and no real benefit over the post-production-programming.

[MIC interface]

Interesting, thanks. I'll leave it to @weef to investigate and act accordingly.

Actually, I meant mschramm

Just to reiterate, the MIC- and MIC+ pins that come out of the GTM900 on the FPC interface are meant for direct connection of an electret condenser microphone. [...] understood.

#19 - 05/04/2020 07:47 PM - mschramm

- Checklist item [x] UART A set to Done
  Checklist item [x] UART B set to Done
  Checklist item [x] Audio on 3.5mm jack[s] set to Done

05/16/2020
Checklist item [x] mechanical mounting of GTM module set to Done
Checklist item [x] power button control via modem handshake (optional via jumper) set to Done
- File gtm900-audio-io-asymm.png added
- File gtm900-audio-io-symm.png added
- File lenovo-combo-audio-jack.jpg added
- File PWON_chinese-ds.png added
- File PWON_english-ds.png added
- % Done changed from 70 to 80

Thanks for remarks and input!

Audio I/O:

@falconia: after I had some translated phrases on what is in section 4.6.3 of the Chinese datasheet, it became clear that this is *internal* circuitry...!

But this means that we

- either have an audio-out as shown in gtm900-audio-io-symm.png (incompatible with a Lenovo Combo audio ("AHJ/CTIA") requested by laforge; in this case we should strip the TRRS jack and bring those four wires to a pin header)),
- or try an asymmetrical approach (gtm900-audio-io-asymm.png), which might work on a CTIA TRRS headset (in this case the TVS array should be reduced to two discrete diodes)

  For the latter, we might want to introduce a AGND area covering the receptacle, connected with a ferrite or a 10R.. to GND.

SUSPEND/RI SCI on modem's UART, RI:

The CP2105 can be programmed before the FPC towards the modem is mounted / inserted into both receptacle. Question remains whether the CP2105 needs the VIO input from the modem during programming.

Also, as we don't need the VDD_3V45 from the CP2105, we could strip the two buffer caps there - or just leave them in place and bring VDD_3V45 to another test pad.

PWON control via modem handshake (optional via jumper):

The English and the Chinese datasheet differ in the display on how PWON needs to be used (see images PWON_chinese-ds.png and PWON_english-ds.png): for powering on, a 10ms LOW on that line switches on, while bringing it LOW for 2 to 3 seconds, switches off the modem.

We hence should not short this signal with a jumper to GND. I now used a tact switch as for RESET, and a non-inverting buffer with a tristate output fed by the UART_DTR signal (I should add a 0R resistor on that buffer's input, or maybe the mentioned jumper... ;)

mechanical mounting of GTM module:

Out of the four drills in the GTM900 we should only use three (one is a halved on an edge), drills take 2mm screws. We should take hex spacers for 2mm and screws similar to those on the M.2/NGFF PCBA.

#20 - 05/04/2020 08:22 PM - mschramm
- File deleted (GTM900-breakout-schematic.pdf)

#21 - 05/04/2020 08:22 PM - mschramm
- File GTM900-breakout-schematic.pdf added

(schematic pdf updated)

#22 - 05/05/2020 05:17 AM - falconia

Hi mschramm,

@falconia: after I had some translated phrases on what is in section 4.6.3 of the Chinese datasheet, it became clear that this is *internal* circuitry...!

Glad we are on the same page now.

But this means that we

- either have an audio-out as shown in gtm900-audio-io-symm.png (incompatible with a Lenovo Combo audio ("AHJ/CTIA") requested by laforge; in this case we should strip the TRRS jack and bring those four wires to a pin header)),

This is the approach I would favor, including the pin header part - it is the approach implemented on my MMTB1, simple and trustworthy.
or try an asymmetrical approach (gtm900-audio-io-asymm.png), which might work on a CTIA TRRS headset (in this case the TVS array should be reduced to two discrete diodes)

I don’t know enough audio-fu to tell if this approach will work or not
- proceed at your own risk.

For the latter, we might want to introduce a AGND area covering the receptacle, connected w/ a ferrite or a 10R.. to GND.

Ditto regarding the unknown.

The CP2105 can be programmed before the FPC towards the modem is mounted / inserted into both receptacle.

Yes, this is what I am hoping you will do on your end before shipping these boards to webshop customers.

Question remains whether the CP2105 needs the VIO input from the modem during programming.

To be honest, I rather dislike the entire idea of using the modem’s V-IO output to power the VIO supply on the USB-serial converter side. Instead the proper approach would be to implement a dedicated LDO regulator on the USB side, powered from USB VBUS and putting out 2.8V, and use that non-modem-sourced regulator to power the USB-serial chip’s VIO supply. If you are going to keep your current arrangement of abusing Calypso V-IO for powering CP2105 I/O buffers, then you are going to have the following two potential problems:

1) You won’t be able to use any CP2105 GPIO outputs to command a modem power-on: while the VRPC block in the Iota chip in the modem is in the switched-off state, the chip’s VRIO regulator is fully off, and there is no voltage put out on the V-IO rail - a chicken and egg problem.

2) If the Calypso+Iota chipset is allowed to go into superdeep sleep (my made-up term for the sleep mode in which not only is the Calypso put into deep sleep with the VCXO stopped, but the Iota ABB sleep mode is also activated), the maximum current that can be drawn from the VRIO regulator is reduced to just 1 mA, in contrast with the 100 mA allowed in Active mode. The modem’s internal circuits are designed so that this 1 mA sleep mode limit is never exceeded, but I can only reason that your CP2105 can easily exceed that very tight limit.

PWON control via modem handshake (optional via jumper):

The GTM900 modem’s PWON control input is wired directly to the same-named signal on the Iota ABB chip, thus its behaviour is governed 100% by Ti’s Calypso+Iota chipset design, not by anything from Huawei. The only way to understand it properly is to read and thoroughly understand section 4.10 (Voltage Reference and Power Control) of Ti’s TWL3025 datasheet:


Pages 40 through 44 of this datasheet cover the VRPC block.

The english and the chinese datasheet differ in the display on how PWON needs to be used (see images PWON_chinese-ds.png and PWON_english-ds.png):

Independent of Huawei’s recommendations, the actual hardware reality inside the Iota chip is that when the VRPC state machine is in the OFF state (the initial power-on condition after application of VBAT to module pins 1-5), any instantaneous sampling of a LOW on the PWON line will initiate the switch-on sequence, causing the Calypso processor to boot. Regardless of what happens afterward (PWON being quickly released or staying low forever), any further state transitions can only be commanded by Calypso firmware.
for powering on, a 10ms LOW on that line switches on, while bringing it LOW for 2 to 3 seconds, switches off the modem.

These are firmware conventions, not hardware - there aren't any such timers anywhere in that modem hw.

We hence should not short this signal with a jumper to GND.

At least with FreeCalypso fw (not sure about Huawei's original fw), having PWON permanently grounded with a jumper is OK but suboptimal: the firmware is smart enough to recognize that no power-off request is being made in this situation, but the downside is that Iota sleep mode (what I call superdeep sleep) cannot be entered: a LOW on PWON is an unconditional wakeup request to Iota VRPC.

On my FCDEv3B I provide a PWON pushbutton switch and a two-pin header wired in parallel with it: using the pushbutton switch is preferable, but a jumper may also be installed for unattended operation where no human is available to press the button.

I now used a tact switch as for RESET.

This one is good. Please note that your pull-up resistor R11 is unnecessary, although harmless - PWON is internally pulled up to VBAT inside the Iota chip itself.

and a non-inverting buffer w/ a tristate output fed by the UART_DTR signal

This one is more problematic: Huawei's UART_DTR signal is wired to Calypso GPIO3 internally (following TI's C-Sample and D-Sample reference boards where GPIO3 was wired as DTR), TI's TCS211 reference fw implements logic for CSD call hang-up per AT&D, and I assume that Huawei's fw does likewise. Users should be able to wiggle DTR however they like to play with AT&D logic in the firmware, without accidentally commanding a power-off or preventing the modem from going into superdeep sleep. Please pick a different CP2105 GPIO output: you have all of the GPIOs associated with the other UART currently unused.

I would also strongly recommend following Openmoko's reference for how to control PWON with a GPIO coming an application processor:

ftp://ftp.freecalypso.org/pub/GSM/GTA02/GTA02_Schematic_MB_A5_1220.pdf

See page 16, the circuit around Q1000.

Two additional concerns:

1) The RST signal which GTM900 provides on pin 31 is a very non-trivial beast. As I discovered by reverse eng of this GTM900 module, it turns out that Huawei replicated the XDS_RESET circuit from TI's Leonardo development board:


Look on page 3, the JTAG block depiction along the top of the page, the transistor circuit (Q308A and Q308B) between Iota nTESTRESET and JTAG connector pin 2. This Q308 dual transistor is also present on the GTM900, and Huawei's FPC connector pin 31 is wired exactly the same way as JTAG connector pin 2 on TI's reference boards (and on my own FCDEv3B too, which also has this exact same circuit). Some while ago I wrote a detailed document explaining all of these quirks:

https://www.freecalypso.org/hg/freecalypso-docs/file/tip/Calypso-test-reset

Given that the RST signal you are manipulating is really XDS_RESET, your pull-up to VBAT (R12) is wrong - it would leak the much higher VBAT voltage into the modem's V-IO rail through the internal resistors built into the prebiased transistor package. Having a pushbutton switch shorting XDS_RESET to GND is OK (although not ideal because of contact bounce, as explained in my article linked above) - but because the reset signal is XDS_RESET rather than raw Iota nTESTRESET, you
could drive it with another GPIO from your CP2105.

2) What are you doing with the Calypso SIM_CD line which GTM900 brings out on pin 24? Right now it looks like you are leaving it unconnected, which would be bad as it would be a floating CMOS input on the Calypso. On TI's Leonardo reference schematics, on Openmoko GTA02 and on my own FCDEV3B this SIM_CD signal is connected to Calypso V-IO, and I recommend that you do the same. Alternatively, connecting it to GND will probably be fine too - it looks like most firmwares ignore it, as long as it isn't left completely floating.

I hope that my feedback will be taken to heart - while I prefer my own MMTB1 for my own playing with GTM900 modems, I am interested in your competing GTM900 breakout board because if you end up offering it in your webshop and it is designed correctly, it would provide one more low-entry-barrier route for people to play with my FreeCalypso firmware on cheap and readily available hw. But it would be a real shame if you design your breakout board in such a way that it would be usable only with your primitive OBB firmware and not with richer firmwares like FreeCalypso or Huawei's original.