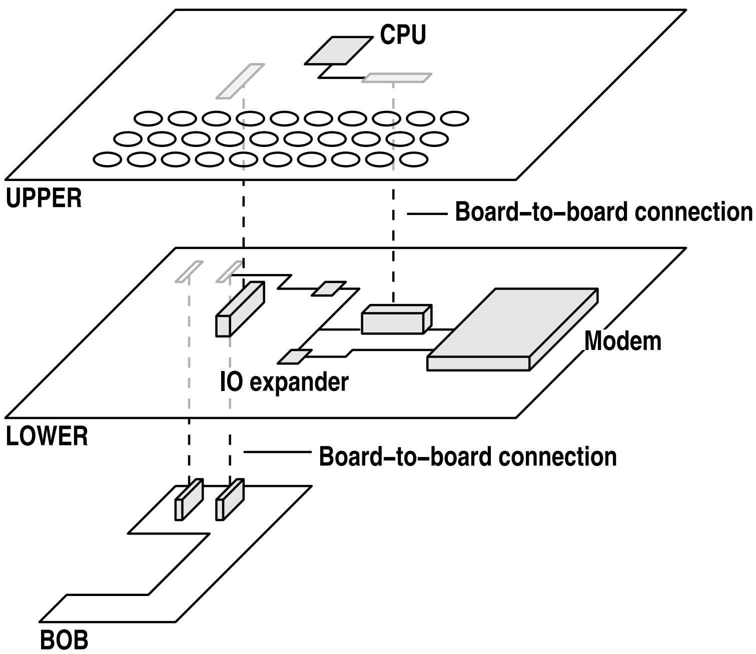
Neo900 IO Expanders

Jörg Reisenweber*, Werner Almesberger†

December 31, 2016

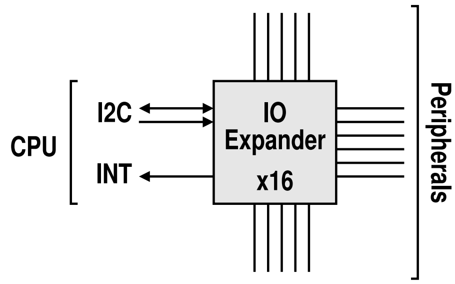
The circuits of Neo900 are distributed over three principal boards called UPPER, LOWER, and BOB (for "Break-Out Board"). LOWER contains the modem, most of the sensors and switches, the audio subsystem, etc. UPPER contains the keyboard contacts, the CPU, memories, etc. BOB contains the memory card holder, Hackerbus, etc. The three boards are interconnected with board-to-board (B2B) connectors, as shown in the following stylized drawing:



Neo900 uses a large number of control and interrupt signals that are handled by ordinary GPIO (for "General-Purpose Input/Output") pins. The number of GPIO pins available on the CPU is limited, and any signal traveling from the CPU to a peripheral on LOWER or BOB also needs to cross the UPPER-LOWER B2B connector.

*Design requirements.
†Specification details and illustrations.

In order to reduce the number of CPU pins and B2B contacts needed, we employ so-called IO expander chips:

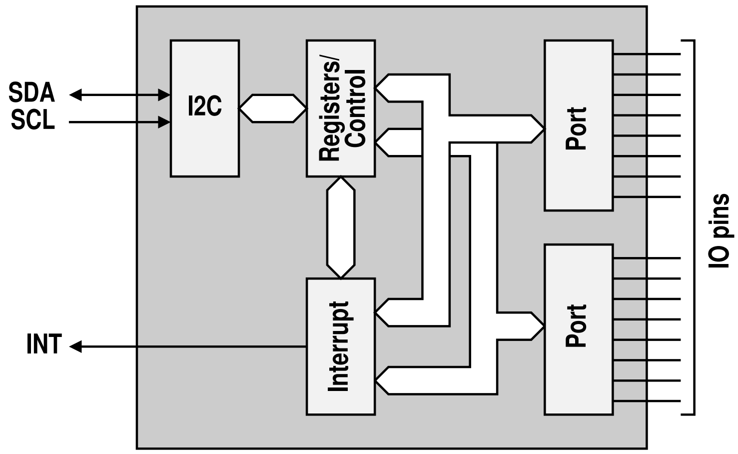


An IO expander communicates with the CPU through I^2C and typically has one dedicated interrupt line. The IO expander provides a number of IO pins that connect to peripherals.

In the following sections we briefly describe common features of IO expander chips, examine a few examples of commonly used chips, and then analyze the characteristics of the signals that connect to GPIOs of the CPU or an IO expander.

1 IO expander characteristics

The type of IO expander we consider here consists of the function blocks shown below: I^2C interface gives access to a number of internal registers, through which the functions of the chip are controlled. The port drivers connect to the registers and also to the interrupt logic.

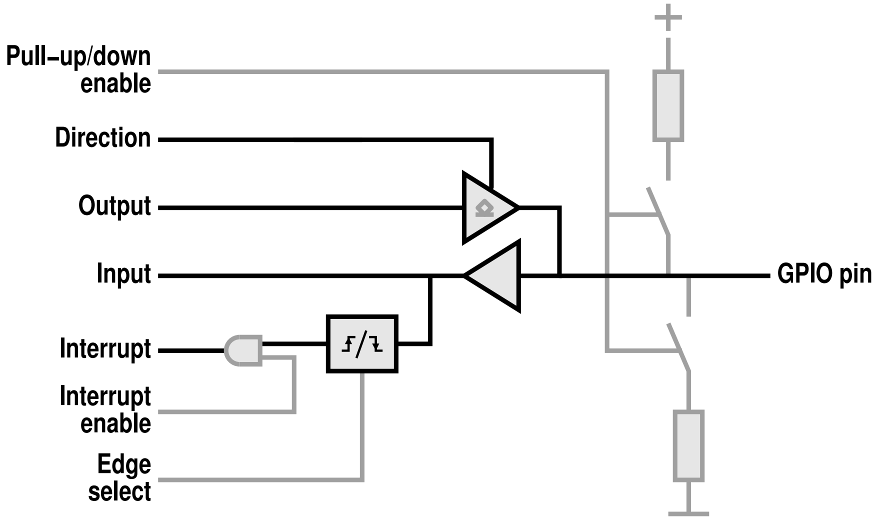


1.1 Pin circuit

The following diagram illustrates the structure of the circuit at each GPIO pin, with features that are only found in some of the chips we looked at shown in grey: an output driver that can be individually enabled when the pin is used for output. In most chips, this driver has a push-pull configuration while some only have open collector outputs. There is normally also a configurable pull-up resistor, but only few chips have also pull-down resistors.

The input driver connects to a register from which the pin status can be read, and also connects to the interrupt logic. In all the chips we considered, interrupts are edge-triggered. Some chips trigger on both edges while others allow selection of rising, falling, or both edges.

Some but not all IO expander chips allow interrupts to be enabled on a per-pin basis. Some chips indicate pending interrupts through a register, while others require the CPU to read the pin state (which means that fast pulses and glitches may be missed). All interrupts are or-ed to produce the interrupt signal to the CPU.



The GPIO functions are thus very similar to that in a CPU, except that interrupt handling is less flexible, and also that GPIO chips usually lack pull-downs, which have become common in microcontrollers over the last years.

Additional capabilities may include multiple voltage domains (e.g., for the host interface and for IO), allowing a GPIO expander to act as level shifter.

1.2 Access timing

We consider only IO expander chips that connect through I^2C. The I^2C interface imposes some constraints on access timing. The two system-wide I^2C busses in Neo900 operate at 400 kHz (I2C#2) and 100 kHz (I2C#3). We assume that all GPIO expanders connect to I2C#2.

A read or write operation to a byte-wide register typically consists of three bytes: the 7-bit I^2C device address, the direction bit, a register number, and the value read or written. Each byte is followed by an acknowledge bit. Furthermore, each read or write operation has one start and one stop bit. The total number of bits per access is therefore $2 + 3 \times (8 + 1) = 29$. At 400 kHz, this yields an access time of 72.5 μs, and a maximum rate of 13 793 operations per second.

A slow I^2C device may also delay acknowledgement, extending the duration of a transfer.

Since the I^2C bus is shared among many devices, it may be occupied at the time an access is attempted, and the operation will have to be deferred until the on-going transfer terminates. Given that transfers between the CPU and NFC or the RDS section of the FM transceiver may take in excess of 1000 bit times, such competing transfers may impose latencies in the order of several milliseconds.

Worse, should an I^2C transfer be attempted to a unresponsive device, the kernel driver applies a hard-coded timeout of 1000 ms.[1]

[1] `OMAP_I2C_TIMEOUT` in `drivers/i2c/busses/i2c-omap.c`

We should therefore use IO expanders only for signals that change infrequently, where a typical latency of at least 10 ms can be tolerated, where variations of this latency are acceptable,[2] and occasionally much longer latency does not cause severe malfunction.

1.3 I^{2}C addressing

Each device on an I^{2}C bus has a 7-bit address. This address must be unique on that bus. Most I^{2}C-capable chips can be configured to use one of several hard-wired addresses. This configuration is typically accomplished by connecting a number of address selection pins to ground, VCC, or in some cases also SDA or SCL.

The IO expander chips we consider support between one (i.e., not configurable) and 32 different addresses.

[2] I.e., a consistent high latency at a human interface may be perceived as more agreeable than a latency that is usually very short but occasionally jumps to a large value.

2 IO expander chips

This section examines the characteristics of several IO expander chips available on the market.

2.1 Catalog search

We searched the Digi-Key catalog for suitable IO expander chips in the category "Integrated Circuits (ICs)", sub-category "Interface – I/O Expanders", with 1 474 entries.[3] We then refined the query as follows:

Parameter	Value	Parts
Mounting Type	Surface Mount	1 428
Interface	$I^2C\star$	1 277
Packaging:	¬Digi-Reel, ¬Tape & Reel	532
Interrupt Output	Yes	469
Voltage – Supply	$V_{min} \leq 1.8$ V	155
Number of I/O	≥ 16	79
Package	¬SIOC, $\neg \star$ SOP	45
Frequency	≥ 400 kHz	44

Of these, 35 were in stock. Sorted by unit price for 1000 units, and ignoring anything larger than 4×4 mm or more expensive than USD 1.50, we get this list:

[3] As of 2016-06-05.

Manufacturer	Part name	Package	Size (mm)	Unit price (USD)
Exar	XRA1201P	24-QFN	4×4	0.56
	XRA1201	24-QFN	4×4	0.56
NXP	PCAL6416A	24-BGA	3×3	0.78
		24-QFN	4×4	0.78
STM	STMPE1600	24-QFN	4×4	0.80
	STMPE1801	25-BGA	2×2	0.82
Semtech	SX1503	28-QFN	4×4	0.84
NXP	PCA9539A	24-QFN	4×4	0.86
	PCAL6416A	24-BGA	2×2	0.86
	PCA6416A	24-QFN	4×4	0.86
	PCAL6416A	24-BGA	2×2	0.86
	PCA9535A	24-QFN	4×4	0.86
TI	TCA6416A	24-QFN	4×4	0.89
STM	STMPE1601	25-BGA	3×3	0.89
NXP	PCA9575	24-QFN	4×4	0.90
TI	TCA1116	24-QFN	4×4	0.95
	TCA9539	24-QFN	4×4	0.95
	TCA9555	24-QFN	4×4	0.95
	TCA6416A	24-BGA	3×3	0.96
Exar	XRA1203	24-QFN	4×4	1.00
	XRA1207	24-QFN	4×4	1.00
TI	TCA6418	25-BGA	2×2	1.02
	TCA9535	24-QFN	4×4	1.07
Microchip	MCP23018	24-QFN	4×4	1.14
Maxim	MAX7325	24-QFN	4×4	1.19
Semtech	SX1509B	28-QFN	4×4	1.22
	SX1509QB	28-QFN	4×4	1.22
STM	STMPE2401	36-BGA	3.5×3.5	1.46

Except for the STMPE1801 and the TCA6418, which have 18 GPIOs, all the above expander chips have 16 GPIO pins.

2.2 Current consumption

The following table shows the supply current for “idle” and “active” states, according to the respective data sheet. We define “idle” as the absence of I^2C or any other IO activity, and “active” as some amount of traffic on the I^2C bus, operating at $f_{SCL} = 400$ kHz. Conditions for published “active” state consumptions vary widely, and the values should therefore be considered as indicators (or absence thereof) for characteristics that may need further examination when a given chip is considered.

Manufacturer	Part name	Current consumption (μA)				Voltage (V)
		Idle		Bus active		
		Typ	Max	Typ	Max	
Exar	XRA1201/P	—	1	—	50	1.8
	XRA1203	—	1	—	50	1.8
	XRA1207	—	1	—	50	1.8
Maxim	MAX7325	0.9	1.9	23	55	3.3
Microchip	MCP23018	—	1	—	1000	1.8–5.5
NXP	PCA6416A	0.5	1.7	4	9	1.65-2.3
	PCA9535A	0.5	1.7	4	9	1.65–2.3
	PCA9539A	0.5	1.7	4	9	1.65–2.3
	PCA9575	0.25	2	135	200	3.6
	PCAL6416A	0.5	1.7	4	9	1.65–2.3
Semtech	SX1503	—	2	—	7	< 2
	SX1509B/QB	3	9	?	?	3.3
STM	STMPE1600	0.25	1	135	200	1.8–3.3
	STMPE1801	—	0.5	28	55	1.8
TI	TCA6416A	0.5	1.7	4	9	1.65–2.3
	TCA6418	—	13	—	25	1.65–3.6
	TCA9535	0.4	2.2	5	11	1.95
	TCA9539	0.4	2.2	5	11	1.95
	TCA9555	0.5	1	5	11	1.95

We choose operating conditions that include an environmental temperature of 25 °C. We prefer specifications for exactly 1.8 V, but the published values may apply to a wide range or a higher voltage, as indicated in the "Voltage" column, above.

The above table does not include chips that the manufacturer declared as obsolete or for which only insufficient documentation exists.

2.3 Products by vendor

IO expanders made by the same company usually share a common set of basic features, or show some form of design evolution. We revisit the aspect of progressive design improvements in section 2.4.

2.3.1 Exar

All Exar chips have push-pull outputs (despite Digi-Key claiming that some are only open drain) and each pin has an individually programmable pull-up resistor. Interrupts can be generated on the raising or falling edge, or on both edges, and each pin has an individual interrupt mask bit.

Part name	I^2C addresses	Pull-up on reset	Other
XRA1201	32	off	
XRA1201P	32	on	
XRA1203	16	off	reset input
XRA1207	4	off	reset input, level shifter

Exar also indicate that their IO expanders are "pin and software compatible" with the following chips by other manufacturers:

Exar	NXP	TI	Maxim
XRA1201P	PCA9555	TCA9555	MAX7311, MAX7318
XRA1201	PCA9535	TCA9535	MAX7312
XRA1203	PCA9539	TCA9539	—
XRA1207	—	TCA6416	—

We can therefore refer to the NXP section (2.3.4) for further details.

2.3.2 Maxim

The MAX7325 has a number of rather eccentric features: one port is push-pull and output-only while the other port is open drain with pull-up resistors whose configuration depends on the I^2C address (!) selection. Likewise, the reset level (i.e., high or low) of the output-only ports is determined by the address selection. Registers are selected not in the usual way of transmitting a register number, but through the I^2C address (this chip occupies two device addresses), and position-dependent semantics of multi-byte reads or writes.

2.3.3 Microchip

The MCP23018 has only open drain outputs, supports I^2C speeds of up to 3.4 MHz, and features one interrupt line per port. Otherwise, it is similar to the XRA1203.

Part name	I^2C addresses	Pull-up on reset	Other
MCP23018	8	off	reset input, two interrupt lines

2.3.4 NXP

The NXP PCA series chips PCA6416A, PCA9535A, and PCA9539A have push-pull outputs. There are no integrated pull-up or -down resistors. Interrupts are generated on either edge, and cannot be masked. The CPU has to determine the interrupt status by reading the inputs – there is no interrupt status register.

The PCA9575 has pull-up and -down resistors, where pull resistors are enabled per port but the direction (i.e., up or down) can be selected per pin. The PCA9575 also has interrupt mask and status registers.

Part name	I^2C addresses	Pull on reset	Other
PCA6416A	2	—	reset input, level shifter
PCA9535A	8	—	
PCA9539A	4	—	reset input
PCA9575	1	off	reset input, level shifter[4]

The PCAL series is an update of the PCA series, adding improved pull resistor and interrupt handling, like in the PCA9575, but with per-pin pull enable.

2.3.5 Semtech

The SX1503 has pull-up and -down resistors, interrupt mask, and edge selection (rising, falling, both). All of these features can be individually set per pin. As an unusual extra, some GPIOs can be configured to act as simple programmable logic devices (PLD).

SX1509B and SX1509QB seem to differ only in part name and package marking. They are similar to the SX1503, but have a LED controller and a keyboard controller instead of PLD.

Part name	I^2C addresses	Pull on reset	Other
SX1503	1	off	reset input, level shifter[4], PLD
SX1509B/QB	4	off	reset input, level shifter[4], LED, keyboard

2.3.6 STM

The STMPE1600 and the STMPE1801 are similar to the PCA family, except that they have an interrupt mask. The STMPE1801 also allows edge selection, has 18 GPIOs instead of the usual 16, and features a keyboard controller. The STMPE1600 has no pull resistors. The STMPE1801 has individually programmable pull-up resistors in GPIO mode, and an additional set of pull-down resistors only used in keyboard mode. The data sheet does not mention the reset state of the pull-ups.

The STMPE1601 and STMPE2401 found in the catalog search are not recommended for new designs.

Part name	I^2C addresses	Pull on reset	Other
STMPE1600	8	—	
STMPE1801	1	?	reset input, keyboard

2.3.7 TI

The chips TCA6416A, TCA9535, and TCA9539 seem to be functionally identical to their NXP counterparts, PCA6416A, PCA9535A, and PCA9539A, respectively.

[4] Each port (8 bits) has individual IO voltage.

The TCA1116 seems to be an obsolete design that is similar to the TCA9539. The data sheet is truncated and says that a full data sheet is available only on request.

The TCA6418 has per-pin pull-down resistors (but no pull-up), an interrupt mask, per-pin selectable edge polarity, and supports an I^{2}C speed of up to 1 MHz.

The TCA9555 is identical to the TCA9535, except that it contains hard-wired pull-up resistors.

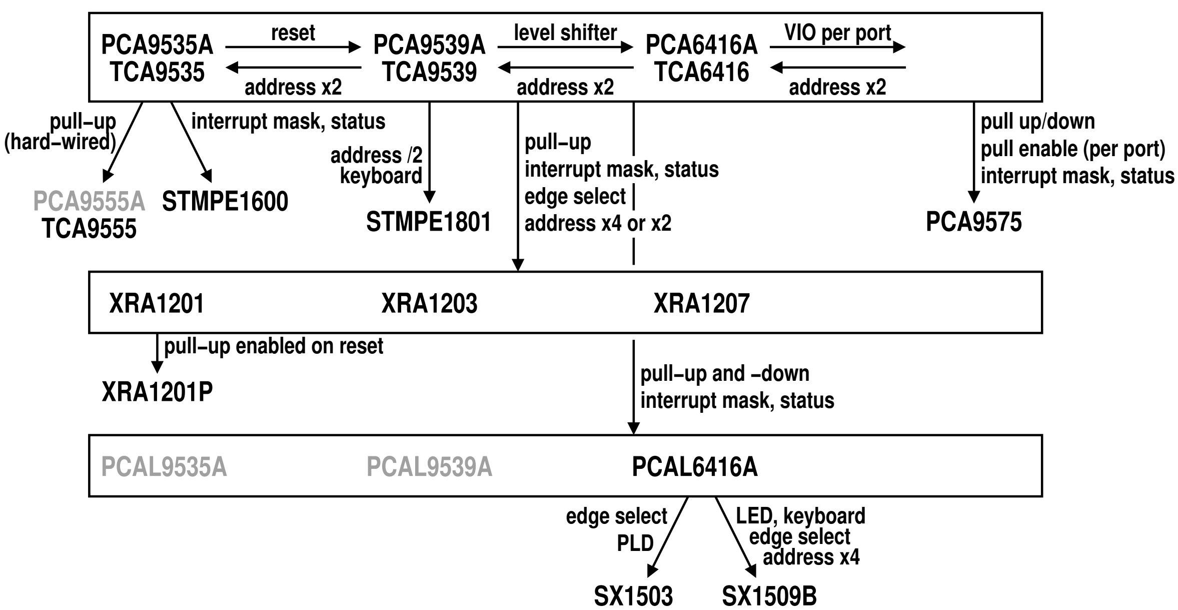
Part name	I^{2}C addresses	Pull on reset	Other
TCA6418	1	off	reset input
TCA9555	8	up	

2.4 Evolution

There are two clear development trends in the PCA/TCA type of chip families: within a technology generation, different chip variants trade I^{2}C address selection pins for a reset input and a separate IO voltage input (level shifting).

There is a big technological leap from the PCA/TCA generation to the compatible chips by Exar, adding pull-up resistors, interrupt masks, interrupt edge selection, and also making more efficient use of the I^{2}C address selection pins by adding the option to connect them to SDA or SCL, in addition to VCC or ground.

The NXP PCAL series is an update of the PCA series, with improvements similar to those by Exar plus pull-down resistors, but still lacks edge selection and does not use the improved address selection technique.



The PCA9575 is somewhere between generations, adding some improvements but lacking others. A few more chips are relatively small variations of members of the major IO expander chip families.

We left out the few chips that would not fit nicely into this pattern, but which are also of less interest. Chips with difficult availability are shown in grey.

3 Signal classification

The following sections contain a classification of all signals that can connect either to an IO expander or the CPU. The names of function blocks and signals are those used in the Neo900 block diagram [1] unless noted otherwise. Signal names are given for block diagram and schematics.

Signal types are as seen by the CPU or IO expander, i.e., an "input" is a signal sent from a peripheral towards the CPU.

We classify speed requirements in the following thee categories:

slow for signals that can tolerate latencies in the order of hundreds of milliseconds. This would typically be enable or reset signals for peripherals that are expected to take some time to initialize, or any buttons that request major device configuration changes, such as opening the display slider.

medium for signals that can tolerate latencies of tens of milliseconds. When interacting with a device, the shortest response delays and delay variations a human user is able to perceive are in this range.

fast for signals where any delay is unwelcome. An example would be interrupts from fast peripherals such as WLAN, where latency directly impacts throughput. Also for signals related to performing any sort of emergency shutdown, minimum latency may be desirable even if the peripheral effecting the shutdown adds substantial delays of its own.

Furthermore, some "fast" signals may have a high rate of change, which would increase occupancy of the I^2C bus and delay other operations.

We consider "slow" signals as generally suitable for use with an IO expander, and "fast" signals as generally unsuitable. "Medium" signals have to be decided on a case-by-case basis.

3.1 Power

Battery charger and fuel gauge have a small number of configuration and status signals, all of which are low-speed.

Function	Signal (block)	(schematics)	Type	Speed
Batt charger	INT	CHG_INT	interrupt	slow
	OTG	CHG_OTG	output	slow
Fuel gauge	GPOUT	BQ_GPOUT	input/output	slow

3.2 Human input sensors

There is a large number of buttons and sensors that detect actions of the user. Most of them should tolerate a moderate amount of latency. Exceptions are discussed below.

Function	Signal (block)	(schematics)	Type	Speed
Lock	SCREEN_LCK	SCREEN_LOCK	interrupt	slow
Slide mag. sensor	SLIDE_SW	SLIDE_SW	interrupt	slow
Capture	CAM_CAP[0]	CAM_CAP_1	interrupt	slow
	CAM_CAP[1]	CAM_CAP_2	interrupt	medium
Cam cover	cam_d11	CAM_COVER_INT	interrupt	slow
Stylus	stylus	STYLUS_INT	interrupt	slow
Kbd scan	KEYIRQ	KEYIRQ	interrupt	medium[5]
3.5 mm	HEADPH_IND	HEADPH_IND	interrupt	slow
	present	MIC_nPRESENT	interrupt	slow
Batt. lid mag.	BATT_LID	BATT_LID	interrupt	slow
uSD card	CD	SD_CD	interrupt	slow
Touch scrn ctrl	TSC_RST	TSC_RST	output	slow[6]
	PEN_INT	PEN_INT	interrupt	medium[7]
Main flex connector	PROXY	PROXY	interrupt	slow

We consider keyboard and touch screen to be medium-speed interrupt sources. The capture button has two levels: the first initiates camera configuration (focus, etc.) and is usually not very timing-critical. The second level releases the shutter, which we consider medium-speed.

Related items that connect to other peripherals and are therefore not considered in this document:

Function	Signal (block)	(schematics)	Connected to ...
Vol +/−	—	VOL_UP	Keyboard scanner
	—	VOL_DOWN	Keyboard scanner
Power	POWERON	POWERON	Companion chip

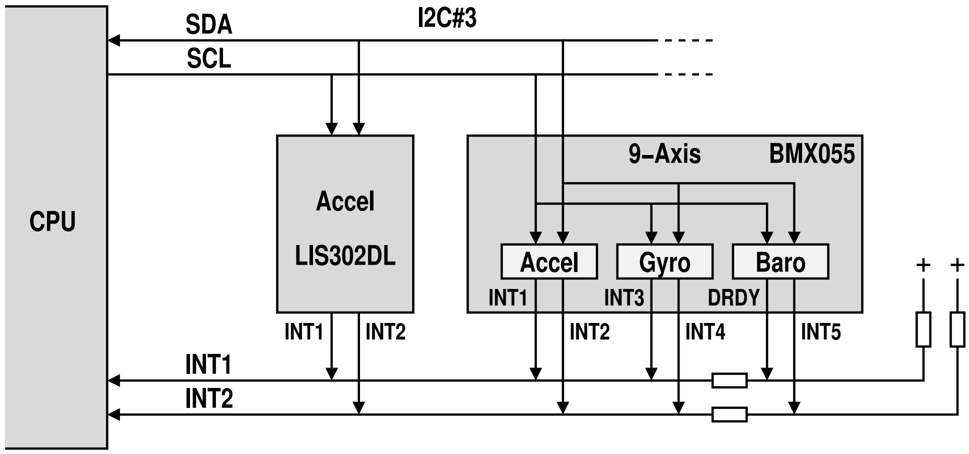
The keyboard controller requires a reset pulse with a minimum duration of 120 μs. The time to perform a reset is also 120 μs (section 6.7 of [3]). We use nRESWARM (instead of a dedicated reset signal), which has a default duration of about 183 μs.

3.3 Environmental sensors

Function	Signal (block)	(schematics)	Type	Speed
Main flex connector	ALS_INT	ALS_INT	interrupt	slow[8]
Accel	INT1	SENS_INT1	interrupt	fast
	INT2	SENS_INT2	interrupt	fast
9-Axis	INT1	SENS_INT1	interrupt	fast
	INT2	SENS_INT2	interrupt	fast

[5] The minimum debounce time is 25 ms (section 8.6.2.15 of [3]), the maximum debounce time is 60 ms (section 6.9).
[6] Reset time is 13 ms (section A.7 of [4]). The minimum duration of the reset pulse is not specified.
[7] The highest sampling rate of the resistive touch screen is 10 points per second (section 4.2.3.4 of [4]. Neo900 does not use the chip's capacitive system.
[8] Maximum interrupt rate is at most once (see page 16 of [5] per integration time (13.7, 101, or 402 ms, table 6).

The accelerometer and the 9-axis sensor have a total of eight different interrupt outputs. The following drawing illustrates how these outputs are merged into the two signals that go to the CPU:



The resistors at the barometer side are needed because the interrupts of the barometer pins have push-pull outputs while all other interrupts can be configured to work as open collector. We therefore let the barometer outputs act as pull-up for the other interrupt lines. The additional pull-ups (on the right side) are need in case the barometer interrupts are disabled, which sets them high impedance.

The LIS302DL ("Accel" [6]) can generate interrupts for the following events:

- availability of sample data (up to 400 samples per second),
- free fall detection, and
- click detection.

Free fall and click detection presumably operate at a granularity determined by the sample rate. Any of the three event types can be assigned to either interrupt line.

The BX055 9-axis sensor [7] contains an accelerometer module that can provide up to 10 ksamples/s (section 5.2.1), a gyroscope module with up to 2 ksamples/s (section 7.3), as well as additional modules operating at lower rates, and various filtering options. Accelerometer and gyroscope each have two interrupt outputs, which are connected to the two interrupt lines in Neo900.

3.4 Audio

Function	Signal (block)	(schematics)	Type	Speed
Mic/TV	TVOUT_EN	TVOUT_EN	output	slow
Headphone amplifier	HEADPH_EN	HEADPH_EN	output	slow
Stereo audio codec	CODEC_RST	CODEC_nRESET	output	slow[9]

[9] Minimum duration is 10 ns (page 22 of [8]), with no upper limit.

We assume that users will expect only "slow" responses when setting these audio-related configuration and reset signals. The chips have no unusual or otherwise noteworthy timing requirements.

Related items that connect to other peripherals and are therefore not considered in this document:

Function	Signal (block)	(schematics)	Connected to ...
Speaker amplifier	en	SPEAKER_EN	Companion chip (GPIO7)

3.5 ECI

There is no publicly accessible specification of the ECI protocol and its timing. The best available resource appears to be an MSc thesis by Jussi Hannula [9].

A patch adding ECI support [10] has been submitted for inclusion into the mainline Linux kernel but was apparently never accepted. In any case, this driver assumes an unknown ECI controller device and therefore does not reveal any details of the ECI wire protocol.

The kernel for N-series Nokia devices accesses ECI through an ACI component in the TWL5031 chip.[10] TWL5031 may be an alias for TPS65951 [11]. No further public information about ACI on TWL5031 or TPS65951 seems to exist.

Function	Signal (block)	(schematics)	Type	Speed
ECI	ECI[0]	ECI0	input	fast
	ECI[1]	ECI1	input	fast
	ECI[2]	ECI_OUT	output	fast
	ECI_EN	ECI_EN	output	slow

Given that [9] mentions a close similarity between I^2C and ECI,[11] and that the ECI implementation presented in the paper operates at 400 kHz, we conservatively assume that ECI may run (even though empirical evidence suggesting that ECI may operate at low rates is said to exist) at similar speeds and thus qualifies as "fast".

3.6 WLAN and Bluetooth

Given that the transmission time of 802.11n frames is measured in tens of microseconds, it is safe to consider interrupt latency requirements of the WLAN/BT module to be in the "fast" category.

[10] `https://github.com/pali/linux-n900/blob/v2.6.32-nokia/drivers/mfd/twl5031-aci.c`

[11] Section 3.2 begins with "*From the electronics point of view, the ECI-bus is very much like the I^2C-bus protocol. [...] Basically, a single I^2C-bus protocol's signal line is excluded from the design.*"

Function	Signal (block)	(schematics)	Type	Speed
WLAN/BT	WLAN_EN	WLAN_EN	output	slow[12]
	WLAN_IRQ	WLAN_IRQ	interrupt	fast
	BT_EN	BT_EN	output	slow[13]

3.7 FM

Function	Signal (block)	(schematics)	Type	Speed
FM/TX	FM_nINT	FM_nINT	int./output	medium
	FM_nRST	FM_nRST	output	slow

The Si4721 uses interrupts mainly to indicate the following types of events:

- command completion (for all commands, no matter how long or short),
- the crossing of a signal strength threshold, and
- RDS FIFO state changes.

In all cases, the use of interrupts is optional. Among the interrupt sources in the transceiver, we assume the RDS FIFO to be the source of the most timing-critical type of event (i.e., to prevent overruns). The documentation [13, 14] is inconclusive regarding FIFO characteristics.

However, programming examples mention transfer of one RDS group while another is being received. According to [15], RDS "A" groups can arrive at a rate of up to 11.4 groups per second in pre-2.0 RDS. Without further background research, we take this as our reference rate and thus assume that interrupt latency should be significantly below one "A" group time, i.e., 87.7 ms.

The GPO2/nINT pin of the FM transceiver acts as configuration input (!) at reset, and must be held low to select the I^2C interface.

[12] WLAN_EN acts as reset signal, with a power-up time of 5 ms (figure 5.3 in section 5.22.3 of [12]. WLAN_EN also needs to be deasserted at least 10 μs before the VBAT and VIO voltage rails drop, lest the device be damaged (section 5.22.2). This timing requirement requires dedicated hardware and is outside the scope of this document.

[13] BT_EN acts as reset signal, with a maximum initialization time of 100 ms (figure 5-5 in section 5.22.5 of [12].

3.8 Modem

Function	Signal (block)	(schematics)	Type	Speed
Modem monitor	ALERT	INA231_INT	interrupt	fast
	EN	MODEM_EN	output	fast
Modem	EMERG_OFF	MODEM_EMERG	output	fast
	RING	RING	interrupt	slow
	PWR_IND	PWR_IND	interrupt	slow
	LC_IND	LC_IND	interrupt	slow[14]
	STATUS	STATUS	interrupt	slow[15]
	WAKEUP	MODEM_WAKEUP	interrupt	medium
	MODEM_IGT	MODEM_IGT	output	slow[16]
	VMIC	MODEM_VMIC_SENSE	interrupt	slow
Modem TX monitor	—	CELL_DETECT_IRQ	interrupt	fast
USB PHY	RESETB	—	output	slow

We classify all modem monitor signals and EMERG_OFF as "fast", assuming that upon detection of unexpected activity, corrective action (e.g., a shutdown) may be initiated without delay, even if not all of these signals may cause immediate cessation of a potentially undesired operation.

Related items that connect to other peripherals and are therefore not considered in this document:

Function	Signal (block)	(schematics)	Connected to ...
Modem TX monitor	—	ADC1 (analog)	Companion chip
	—	ADC2 (analog)	Companion chip

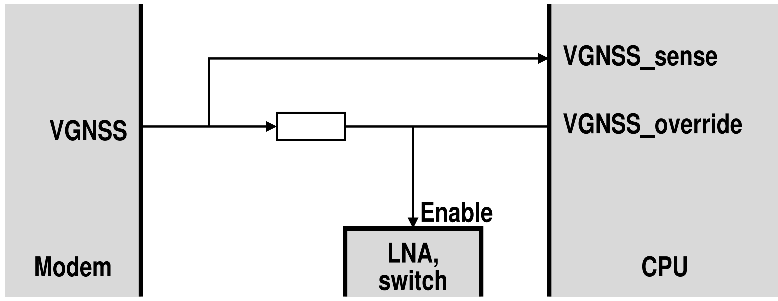
3.9 GPS

The modem uses its VGNSS output to enable the amplifier for the GPS antenna signal. We add two signals controlled by the CPU that allow it to detect whether the modem is trying to use the GPS antenna, and to override the modem.

[14] Use case is unclear – we already have independent current monitoring through the modem monitor.

[15] STATUS is not only an on/off indication but can also indicate additional details by blinking at 1 Hz with duty cycles ranging from 1% (10 ms) to 50% (section 18.5 of [16]).

[16] Minimum pulse width is 100 ms for activation (section 3.3.1 of [17]), 2.1 s if used for deactivation (section 3.3.4). Figure 7 of section 3.3.4 seems to suggest that the impulse used for activation should not exceed one second, but there is no mention of such a limit elsewhere.



If the CPU tri-states VGNSS_override, the modem controls the LNA and the GPS kill switch. The CPU "kills" GPS by driving VGNSS_override low. In either case, VGNSS_sense tracks which configuration the modem requests.

VGNSS_override (CPU)	VGNSS (Modem)	Enable (LNA)	VGNSS_sense (CPU)
Z	H	H	H
Z	L	L	L
L	H	L	H
L	L	L	L

Neither detection nor intervention are particularly timing-critical since VGNSS_override can simply be kept low unless GPS use is authorized.

Function	Signal (block)	(schematics)	Type	Speed
GPS kill	—	VGNSS_SENSE	interrupt	slow
	—	VGNSS_OVERRIDE	output	slow

3.10 LED drivers

Function	Signal (block)	(schematics)	Type	Speed
Fancy RGB LED driver	RGB_INT	RGB_INT	interrupt	slow
	RGB_CTRL_EN	RGB_CTRL_EN	output	slow

The minimum delay after activating RGB_CTRL_EN is 500 μs (section 7.4.1 of [18]). Note that EN does not affect the I^2C interface, and the functionality of the EN line is also available through the CHIP_EN register bit, with resulting internal enable signal being EN && CHIP_EN (section 7.4.1).

3.11 NFC

Function	Signal (block)	(schematics)	Type	Speed
NFC	rst	NFC_nRESET	output	slow
	int	NFC_INT	interrupt	fast
	swd[0]	NFC_SWD_DIO	input/output	slow
	swd[1]	NFC_SWD_CLK	output	slow

NFC operates at data rates of up to 848 kbps.[17] While the dedicated microcontroller can perform timing-critical operations on behalf of the CPU, the ability to respond rapidly to NFC events may still be desirable.

The SWD signal are for use during development and are not timing-critical.

3.12 Second SIM

Function	Signal[18] (block)	(schematics)	Type	Speed
SIM switch	—	MUX_STROBE	output	slow[19]
	—	MUX_SEL	output	slow
	—	MUX_CPU_nMODEM	output	slow
	—	CPU_3V_n1V8	output	slow
	—	CPU_PWR_EN	output	medium
	—	CPU_SIM_RST	output	medium
	—	CPU_SIM_CLK	output	fast
	—	CPU_SIM_IO	input/output	fast
SIM #2	cd	CPU_CD_2	interrupt	medium

According to sections 5.2.3 and 8.3 of [21], the SIM CLK frequency must be at least 1 MHz and can be as high as 20 MHz. The data rate at the IO pin can be negotiated, with a maximum of $f_{max} \times D/F = 860$ kHz with $D = 64$, $F = 372$, and $f_{max} = 5$ MHz.

CPU_PWR_EN, RST, and CPU_CD_2 should operate at least at “medium” speed in order to facilitate clean shutdown on card removal.

[17] See section 7.1.2 of [19].
[18] Names as defined in sections 3.1 and 3.2 of [20].
[19] The SIM switch circuit stretches the STROBE signal to at least 500 ms.

3.13 Main camera

Function	Signal (block)	(schematics)	Type	Speed
Main cam conn	cam_d3	CAM_MAIN_SHDN	output	slow[20]
Cam switch	CAM_B_EN	CAM_B_EN	output	slow
Flash LED driver	en	FLASH_EN	output	slow

The following two signals connect to the CPU but are not used as GPIOs:

Function	Signal (block)	(schematics)	Connected to ...
Flash LED driver	int	FLASH_INT	Camera function block of CPU
	strobe	FLASH_STROBE	Camera function block of CPU

3.14 Hackerbus

Function	Signal[21] (block)	(schematics)	Type	Speed
Hackerbus	—	HB_A	input/output	fast
	—	HB_B	input/output	fast
	—	HB_C	input/output	fast
	—	HB_D	input/output	fast
HB USB PHY	RST	—	output	slow

For a maximum of flexibility, all Hackerbus GPIOs are connected – through level shifters – to the CPU, thus allowing them to be used for input, output, and interrupts.

[20] cam_d3 connects to the XSHUTDOWN pin of the camera module: `http://natisbad.org/N900/ref/VS6555.pdf`

[21] Names are either from the Hackerbus white paper [22] or the block diagram [1].

4 GPIO overview

The following tables gives an overview of the GPIOs described in this document. The compatibility column indicates the GPIO number of the corresponding signal in the N900. "—" indicates that a signals is new in Neo900 or has no obvious N900 counterpart. GPIOs that are used for different function blocks in Neo900 are shown in parentheses.

4.1 UPPER board

Function	Signal	Type	Speed	N900
Slide mag. sensor	SLIDE_SW	interrupt	slow	(gpio_71)
Stylus	STYLUS_INT	interrupt	slow	—
Kbd scan	KEYIRQ	interrupt	medium	—
Touch scrn ctrl	TSC_RST	output	slow	gpio_104
	PEN_INT	interrupt	medium	gpio_100
Main flex connector	PROXY	interrupt	slow	(gpio_89)
	ALS_INT	interrupt	slow	gpio_99
Accel	SENS_INT1	interrupt	fast	(gpio_180)
	SENS_INT2	interrupt	fast	(gpio_181)
9-Axis	SENS_INT1	interrupt	fast	shared
	SENS_INT2	interrupt	fast	shared
USB PHY	RESETB	output	slow	—
Fancy RGB LED driver	RGB_INT	interrupt	slow	gpio_55
	RGB_CTRL_EN	output	slow	gpio_41
Main cam conn	CAM_MAIN_SHDN	output	slow	gpio_102
Cam switch	CAM_B_EN	output	slow	gpio_97
HB USB PHY	RST	output	slow	—

4.2 LOWER board

Function	Signal	Type	Speed	N900
Batt charger	CHG_INT	interrupt	slow	gpio_7
	CHG_OTG	output	slow	—
Fuel gauge	BQ_GPOUT	input/output	slow	—
Lock	SCREEN_LOCK	interrupt	slow	gpio_113
Capture	CAM_CAP_1	interrupt	slow	gpio_68
	CAM_CAP_2	interrupt	medium	gpio_69
3.5 mm	HEADPH_IND	interrupt	slow	(gpio_177)
	MIC_nPRESENT	interrupt	slow	—
Mic/TV	TVOUT_EN	output	slow	gpio_40
Headphone amplifier	HEADPH_EN	output	slow	gpio_98
Stereo audio codec	CODEC_nRESET	output	slow	gpio_58
ECI	ECI0	input	fast	gpio_61

Function	Signal	Type	Speed	N900
	ECI1	input	fast	gpio_62
	ECI_OUT	output	fast	(gpio_182)
	ECI_EN	output	slow	—
WLAN/BT	WLAN_EN	output	slow	gpio_87
	WLAN_IRQ	interrupt	fast	gpio_42
	BT_EN	output	slow	—
FM/TX	FM_nINT	int./output	medium	gpio_43
	FM_nRST	output	slow	gpio_163
Modem monitor	INA231_INT	interrupt	fast	—
	MODEM_EN	output	fast	—
Modem	MODEM_EMERG	output	fast	—
	RING	interrupt	slow	—
	PWR_IND	interrupt	slow	—
	LC_IND	interrupt	slow	—
	STATUS	interrupt	slow	—
	MODEM_WAKEUP	interrupt	medium	—
	MODEM_IGT	output	slow	—
	MODEM_VMIC_SENSE	interrupt	slow	—
Modem TX monitor	CELL_DETECT_IRQ	interrupt	fast	—
GPS kill	VGNSS_SENSE	interrupt	slow	—
	VGNSS_OVERRIDE	output	slow	—
NFC	NFC_nRESET	output	slow	—
	NFC_INT	interrupt	fast	—
	NFC_SWD_DIO	input/output	slow	—
	NFC_SWD_CLK	output	slow	—
SIM switch	MUX_STROBE	output	slow	—
	MUX_SEL	output	slow	—
	MUX_CPU_nMODEM	output	slow	—
	CPU_3V_n1V8	output	slow	—
	CPU_PWR_EN	output	medium	—
	CPU_SIM_RST	output	medium	—
	CPU_SIM_CLK	output	fast	—
	CPU_SIM_IO	input/output	fast	—
SIM #2	CPU_CD_2	interrupt	medium	—
Flash LED driver	FLASH_EN	output	slow	(gpio_88)

4.3 BOB

Function	Signal	Type	Speed	N900
Cam cover	CAM_COVER_INT	interrupt	slow	gpio_110
Batt. lid mag.	BATT_LID	interrupt	slow	gpio_160
uSD card	SD_CD	interrupt	slow	—
Hackerbus	HB_A	input/output	fast	—
	HB_B	input/output	fast	—

Function	Signal	Type	Speed	N900
	HB_C	input/output	fast	—
	HB_D	input/output	fast	—

5 References

[1] Neo900 project. *Neo900 Block Diagram*, Work in progress 2016-05-23. http://neo900.org/stuff/block-diagrams/neo900/neo900.html

[2] Golden Delicious Computers GmbH&Co. KG. *Neo900 schematics*, https://neo900.org/stuff/werner/tmp/Neo900-toc.pdf

[3] Texas Instruments Incorporated. *TCA8418 - I2C Controlled Keypad Scan IC*, SCPS215E, November 2015. http://www.ti.com/lit/ds/symlink/tca8418.pdf

[4] NXP/Freescale Semiconductor. *CRTouch Data Sheet*, CRTOUCHDS, Rev. 3, April 2013. http://cache.freescale.com/files/32bit/doc/data_sheet/CRTOUCHDS.pdf

[5] Texas Advanced Optoelectronic Solutions Inc. *TSL2562, TSL2563 Low-Voltage Light-to-Digital Converter*, TAOS066N, August 2010. http://ams.com/eng/content/download/250126/975581/142958

[6] STMicroelectronics. *LIS302DL - MEMS motion sensor*, Rev 4, October 2008. http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00135460.pdf

[7] Bosch Sensortec. *BMX055 - Small, versatile 9-axis sensor module*, BST-BMX055-DS000-02, Revision 1.1 November 2014. https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMX055-DS000-02.pdf

[8] Texas Instruments Incorporated. *TLV320AIC34 - Four-Channel, Low-Power Audio Codec for Portable Audio/Telephony*, SLAS538A, November 2007. http://www.ti.com/lit/ds/symlink/tlv320aic34.pdf

[9] Hannula, Jussi. *Microphone Line Used for Data Transfer*, Tampere University of Technology, August 2011. https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/20669/hannula.pdf

[10] Vihuri, Tapio. *dd support for ECI (multimedia) accessories*, December 12, 2010. http://thread.gmane.org/gmane.linux.alsa.devel/80260

[11] Texas Instruments Incorporated. *TPS65951 - Integrated Power Management/Audio Codec*, SWCS053F, May 2012. https://media.digikey.com/pdf/Data%20Sheets/Texas%20Instruments%20PDFs/TPS65951.pdf

[12] Texas Instruments Incorporated. *WL18x7MOD WiLinkTM 8 Dual-Band Industrial Module*, SWRS170H, October 2015. http://www.ti.com/lit/ds/symlink/wl1837mod.pdf

[13] Silicon Laboratories. *Si4720/21-B20 - Broadcast FM Radio Transceiver for Portable Applications*, Rev. 1.0, February 2008. http://www.silabs.com/Support%20Documents/TechnicalDocs/Si4720-21-B20.pdf

[14] Silicon Laboratories. *AN332 - Si47xx Programming Guide*, Rev. 1.0, September 2014. http://www.silabs.com/Support%20Documents/TechnicalDocs/AN332.pdf

[15] Wikipedia. *Radio Data System*, https://en.wikipedia.org/wiki/Radio_Data_System

[16] Cinterion. *PHS8-P AT Command Set*, PHS8-P_ATC_V02.003, Version 02.003, July 2012.

[17] Cinterion. *PHS8-E Hardware Interface Description*, PHS8-E_v03.001, Version 03.001, December 2012.

[18] Texas Instruments Incorporated. *LP55231 - Programmable 9-Output LED Driver*, SNOSCR5A, October 2014. http://www.ti.com/lit/ds/symlink/lp55231.pdf

[19] Almesberger, Werner. *Neo900 NFC Subsystem*, Draft, December 2015. https://neo900.org/stuff/papers/nfc-draft.pdf

[20] Reisenweber, Jörg; Almesberger, Werner. *Neo900 SIM Switch*, June 2016. https://neo900.org/stuff/papers/simsw.pdf

[21] ISO/IEC 7816-3. *Identification cards – Integrated circuit cards – Part 3: Cards with contacts – Electrical interface and transmission protocols*, ISO/IEC 7816-3:2006(E), Third edition, November 2006.

[22] Reisenweber, Jörg; Almesberger, Werner. *Neo900 Hackerbus*, November 2016. http://neo900.org/stuff/papers/hb.pdf