Neo900 Infrared Subsystem

Jörg Reisenweber*, Werner Almesberger†

December 22, 2016

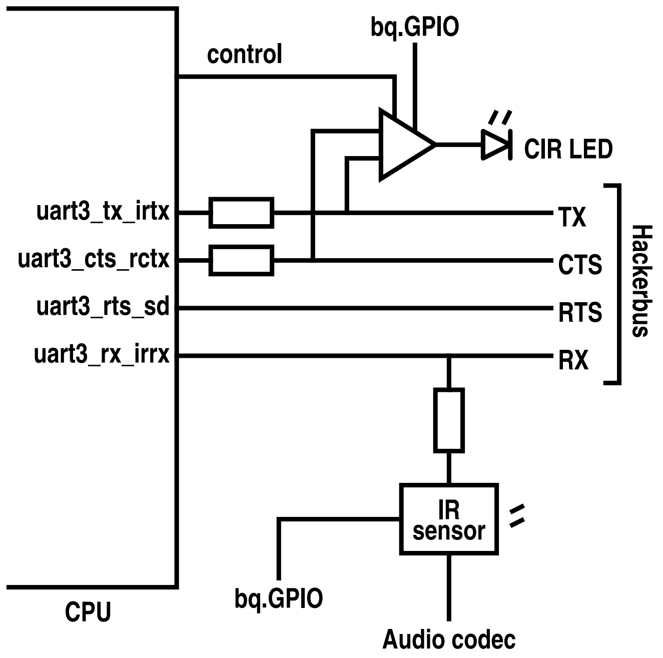
This document describes the infrared (IR) system of Neo900. It is intended to be used both as tentative specification and as documentation of actual functionality, and may be revised as focus shifts over time from requirements to implementation.

*Concept and design.
†Specification details and illustrations.

1 System overview

The IR subsystem consists of a high-powered transmitter and a receiver. It connects to UART3 of the DM3730 CPU, where hardware-assisted IrDA[1] (SIR) and CIR[2] ("Consumer IR") functions are available.

The IR subsystem also connects to the "Hackerbus" which can use the signals to communicate with the CPU, be it as UART or as GPIOs, or to access the IR system. For advanced signal processing, the analog output of the IR sensor connects to an unused input of the audio codec. The following drawing shows the various components:



The signal named bq.GPIO is not only a control input but it also determines the configuration after CPU reset. It is described in section 3.1.

1.1 IR modes

The IR subsystem supports both CIR and IrDA operation. In CIR mode, it can receive incoming IR signals and thus act as a "learning" remote control. Furthermore, the IR transceiver can be configured to send and receive unmodulated data from the UART. We call this "IR-UART" mode.

These characteristics mean that some design conflicts are unavoidable, e.g., because CIR and IrDA operate at different wavelengths. Where a conflict arises, highest priority is given to CIR transmission, followed by CIR reception, and finally IrDA.

[1] http://en.wikipedia.org/wiki/Infrared_Data_Association
[2] http://en.wikipedia.org/wiki/Consumer_IR

Likewise, since the IR data path is shared between CPU and Hackerbus, conflicting configurations are possible. Their resolution is described in sections 4 and 5.

Note that we don't consider IR-UART reception and IrDA to be very important features. If their implementation should prove to be overly troublesome, they could therefore be omitted.

2 Physical layer

The following sections describe the main characteristics of the physical layers of CIR, IrDA, and IR-UART, and then define a set of hardware characteristics that achieve compatibility with all these modes of operation.

2.1 CIR

While there is no common standard for CIR, the characteristics of most IR remote controls typically lie in the following ranges[3], with amplitude-shift keying (ASK) being the most common form of modulation, although other types of modulation are possible:

Parameter	Value	Unit	
Wavelength	870, 930–950	nm	
Data rate	4–120	bps	
Carrier	33-60	kHz	
Beam angle	≥ 50	$^\circ$	(estimate for IR remote)

2.2 IrDA

IrDA (SIR) uses a pulse modulation with UART-like framing and has the following characteristics:[4]

Parameter	Value	Unit	
Wavelength	850–900	nm	
Data rate	9.6-115.2	kbps	
Duty cycle	3/16		“0“ bit
	0/16		“1“ bit
Pulse width	1.6–19.5	μs	
Beam angle	≥ 15	$^\circ$	

2.3 IR-UART

Last but not least, IR-UART mode uses on-off keying (OOK), and we can define the following characteristics:

[3] Based on the following Wikipedia article: `http://en.wikipedia.org/w/index.php?title=Consumer_IR&oldid=615137350#Technical_information`

[4] Based on the following Wikipedia article: `http://en.wikipedia.org/w/index.php?title=Infrared_Data_Association&oldid=615436526#IrPHY`

Parameter	Value	Unit
Data rate	9.6–115.2	kbps (typical)
Duty cycle	9.1–91	% [5]

2.4 Combined characteristics

Combining the specifications from the preceding sections, we obtain the following common characteristics for transmitter and receiver:

Parameter	Value	Unit	
Wavelength	940	nm	transmit
	850–950	nm	receive
Frequency	0.9–312	kHz	
Beam angle	≥ 50	$^\circ$	

2.5 Transmit power

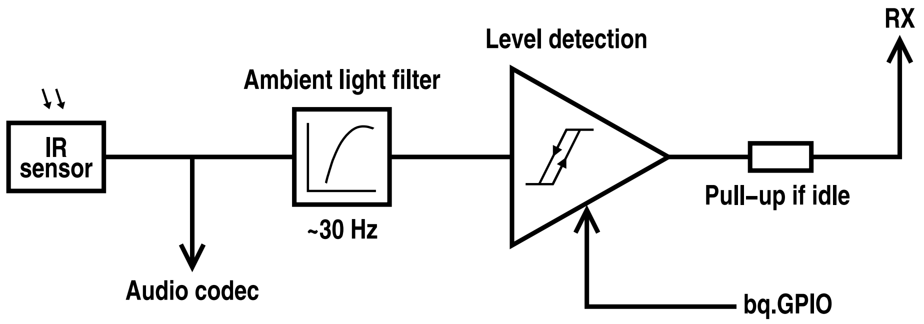
While we do not know the radiant flux density required at the IR receivers of consumer devices and therefore cannot specify the minimum radiant intensity of the transmitter, we expect performance of the IR transmitter to be on par with the average TV remote control. For comparison, the N900 has a range of only approximately 2 m, which we would consider too short.

2.6 Received signal processing

On the receiving side, a high-pass filter suppresses the effects of ambient light.[6] A Schmitt-trigger then detects the strong pulses generated by the remote control and sends the resulting digital signal to the RX line. The following drawing shows the general structure of the receiver circuit. For details, please refer to section 8.

[5] The duty cycle range of 9.1–91% (1/11 to 10/11) is for 8 bits with parity. The duty cycle range would be 10–90% for the more common 8 bits without parity format.

[6] The receiver is designed to be insensitive to constant ambient light like strong sunlight, as well as common incandescent and fluorescent lamps. LED lamps may create interference when they are PWM-controlled and create a sufficient brightness level at the receiver diode.



The series resistor ensures that CPU and Hackerbus can override the IR receiver. When the receiver is not illuminated or when it is disabled, the circuit outputs a high level and thus acts as a pull-up for the RX line.

The unfiltered analog signal should also be sent to an unused input of the audio codec. Further details on the circuit can be found in section 8.

3 Configurations

The IR subsystem can be configured for several different modes of operation. The following sections detail the various configurations.

In each case the configuration of both the transmitter and the receiver is shown. In many real-life situations, only one direction will be used at a time while the other is deactivated. A deactivated transmitter or receiver should behave as described in section 3.2.

3.1 Reset

After a reset, the configuration of the IR system is determined by the state of the GPIO pin of the bq27200 battery fuel gauge. We call this pin "bq.GPIO" in the rest of the document. The fuel gauge chip is directly connected to the battery and retains its state during normal system reset.

Note that the bq27200 does not preserve the entire state of bq.GPIO across power-on reset: while the direction of the pin is determined by EEPROM content and is therefore retained across power-cycling, the open-collector output is always set to "1", i.e., it is high-Z.

If bq.GPIO is high-Z and the CPU's control signals are in their reset state, both transmitter and receiver shall be active and operate in IR-UART mode. The purpose of this mode is to allow the CPU's ROM boot loader to try to boot from UART3 via infrared.

If bq.GPIO is "0" and the CPU's control signals are in their reset state, the IR subsystem shall be deactivated as described in section 3.2.

When the CPU has left its reset state and can actively control the IR subsystem, it is also able to change the state of bq.GPIO. bq.GPIO can therefore be used (or ignored) as the implementation sees fit.

The following table summarizes the IR configuration in power-down and after CPU reset:

bq.GPIO	CPU.controls	IR configuration	
		TX	RX
"1" (High-Z)	reset state	IR-UART	on
	active	defined by controls	on
"0"	reset state	off	off
	active	defined by controls	off

Use of the data signals in IR-UART mode is further explained in section 3.5. Note that Hackerbus can override the receiver (see section 3.6 for details) and the CPU can therefore boot from (wired) UART even if the IR-UART configuration is selected.

3.2 IR deactivated

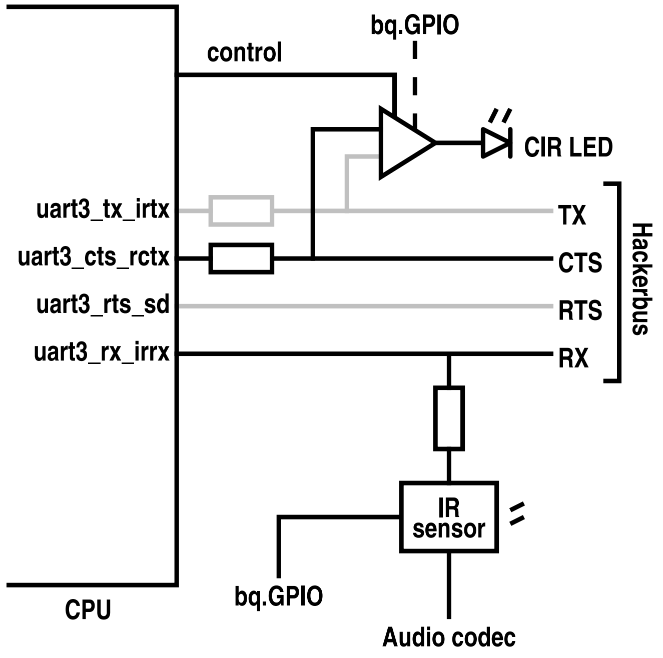
The CPU can deactivate the transmitter and receiver, either both together or one at a time,[7] through the control signals to the IR subsystem.

When deactivated, the IR subsystem must draw as little current as possible. The transmitter must ignore its data inputs, e.g., TX and CTS, and under no circumstance activate the IR LED. The receiver must not drive the RX line and use as little power as possible for any filters and/or amplifiers.

3.3 CIR mode

In CIR mode the CPU uses CTS as an active-high output[8] for transmit data. The CIR module in the CPU could also use RTS for configuration purposes, but we seem to have no use for this and the IR circuit must ignore its state. The use of RX in CIR mode is not clearly specified and it seems safe to use it similar to IrDA mode. TX is not used and can be assigned to other tasks. The IR circuit must therefore ignore the state of TX when in CIR mode.

CIR mode configuration is illustrated in the following drawing:

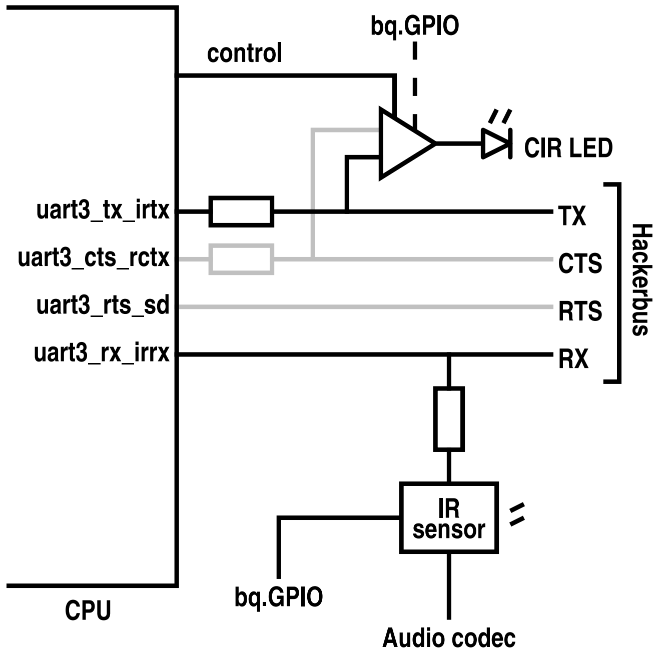


[7] See sections 7.1 and 8.
[8] Note that CTS is an input at the CPU when UART3 is used as regular UART.

3.4 IrDA mode

In IrDA (SIR) mode, the CPU uses TX as active-high output and RX as either[9] active-low or active-high input. RTS could again be used for configuration but is not relevant in our application. CTS is not used in IrDA mode. The IR circuit must therefore ignore the state of RTS and CTS when in IrDA mode.

IrDA mode configuration is illustrated in the following drawing:



3.5 IR-UART mode

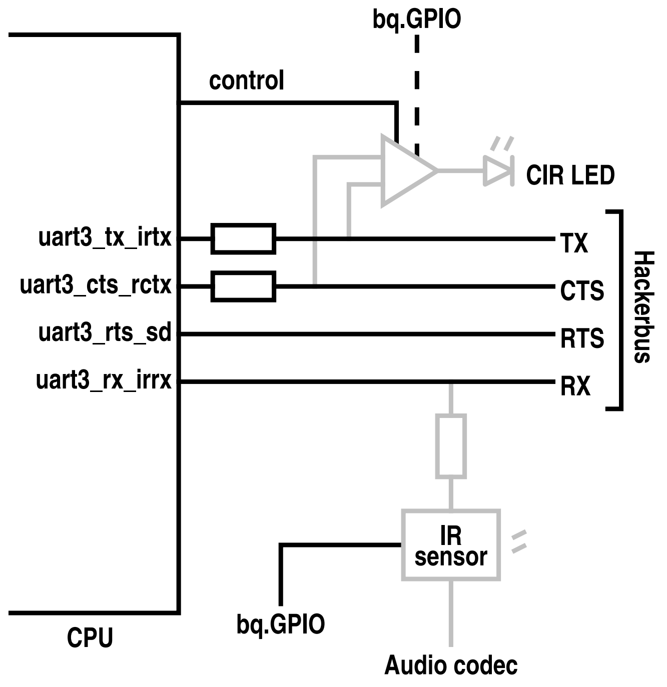
In IR-UART mode, TX and RX both are active-low. RTS and CTS are not used. The routing of the TX and RX signals is the same as in IrDA mode.

3.6 UART mode

The IR subsystem is inactive in UART mode and must behave as described in section 3.2. While we call it “UART mode”, this configuration applies to any setup where the UART signals are used for some form of communication with the Hackerbus that does not involve IR.

UART mode configuration is illustrated in the following drawing:

[9] Depending on the setting of MDR2_REG.RRXINVERT.



Please note that, in order to avoid conflicts with use of the RX line, at least one of the following must be true:

- the IR sensor is disabled by setting bq.GPIO to “0”,
- it is not illuminated by a light source that excites the receiver, or
- a pull-up resistor overriding R7 (see section 8) is added to the RX line.

3.7 Mode summary

The following table summarizes the configuration of the I/O pins at the CPU in the various IR modes:

Signal	IR subsystem mode				
	Off	CIR	IrDA	IR-UART	UART
TX	—	—	$\overset{\sqcap}{\longrightarrow}$	$\overset{\sqcup}{\longrightarrow}$	$\overset{\sqcup}{\longrightarrow}$
CTS	—	$\overset{\sqcap}{\longrightarrow}$	—	—	$\overset{\sqcup}{\longleftarrow}$
RTS	—	—	—	—	$\overset{\sqcup}{\longrightarrow}$
RX	—	$\overset{\sqcup/\sqcap}{\longleftarrow}$	$\overset{\sqcup/\sqcap}{\longleftarrow}$	$\overset{\sqcup}{\longleftarrow}$	$\overset{\sqcup}{\longleftarrow}$

$\longrightarrow$ indicates an output from the CPU, $\longleftarrow$ an input. An active-high signal (i.e., light is emitted or received on the signal being “1”) is marked with $\sqcap$ while an active-low (light on ”0”) signal is marked with $\sqcup$. A signal where the CPU can be configured for either polarity is marked with $\sqcup/\sqcap$. An unused signal is marked with —.

4 Conflict resolution

When a line is driven as output from multiple sources, the following conflict resolution strategy shall be applied:

Line	Dominant	Subordinate	Applicable IR system states
TX	Hackerbus	CPU	IrDA, IR-UART
CTS	Hackerbus	CPU	CIR
RX	Hackerbus or CPU	IR sensor	CIR, IrDA, IR-UART

Conflicts between CPU and Hackerbus on RTS or RX result in undefined behaviour and should be avoided. Conflict increases overall power consumption and should be avoided during normal system operation.

Precedence is established by series resistors next to the CPU for TX and CTS, and a series resistor on the digital output of the IR receiver, before entering the RX line.

5 Undriven signals

The IR circuit must be tolerant to a UART signal being driven by neither CPU, Hackerbus, or the IR receiver. Whether this tolerance is achieved by pulling the signal to a safe state or by applying a more general protection against incorrect inputs is left as an implementation choice.

Control signals that float after CPU reset and that are not ignored by the IR subsystem must be pulled to a safe state.

6 Optoelectronics

The following sections discuss the selection of the optoelectronic devices, and their physical placement.

6.1 Possible component selection

Taking into account the requirements specified in section 1.1, available space (see section 6.2), and general component availability, we consider the Vishay VSMB2948SL[10] IR LED and the Vishay VEMD10940F[11] photodiode suitable for our purpose.

Parameter	Requirement	Unit	Transmitter VSMB2948SL	Receiver VEMD10940F
Wavelength	850–950	nm	940	780–1050
Beam angle	≥ 50	°	50	150
Peak frequency	≥ 0.3	MHz	23	≈ 0.2

Note that the peak frequency of 0.2 MHz for the receiver is an estimate based on the specified rise and fall times of 100 ns each, which are specified for a reverse voltage of 10 V and a load resistance of 1 kΩ whereas we operate at 1.8 V and with a load of 22 kΩ. (See section 8.)

The characteristics of the daylight filter integrated in the IR window of the N900 case are unknown, our design will add multiple internal light sources that may affect the photodiode, and there is also the possibility of outside light reaching the diode through the spacer frame. We therefore consider it desirable for the photodiode to contain its own filter and not have to rely on external filtering and shielding.

We considered the use of a phototransistor, but found that the readily available SMT parts all had rise and fall times in the order of 15 μs, which would limit the maximum signal frequency to only about 30 kHz. They do not allow improving response time by biasing the transistor.

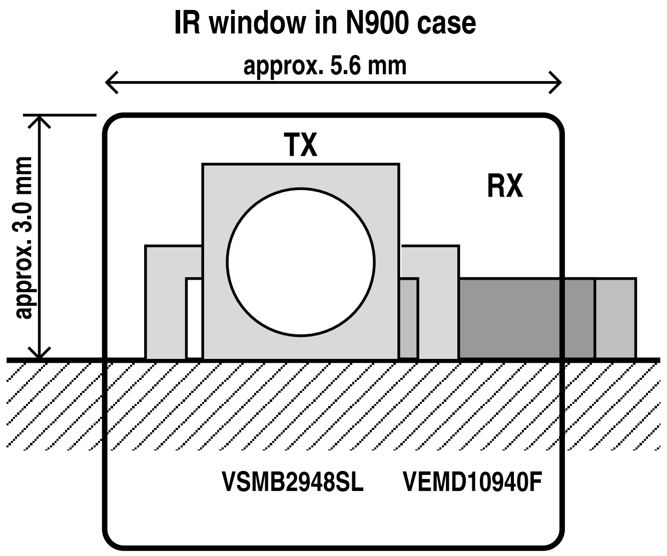
6.2 Component placement

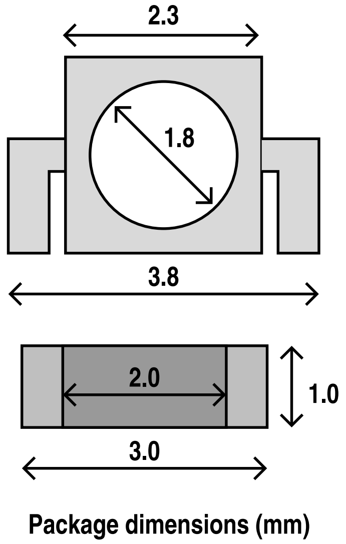
The N900 case has a window with an integrated IR filter of unknown characteristics. The window is approximately 5.6 mm × 5.3 mm.

The window is designed for a single IR LED but we will have a LED and a photodiode. Since IR components are generally large, it won't be possible to place both "nicely" side by side. However, as the following drawing illustrates, both components can "see" the window if placed behind each other:

[10] `http://www.vishay.com/docs/83498/vsmb2948sl.pdf`

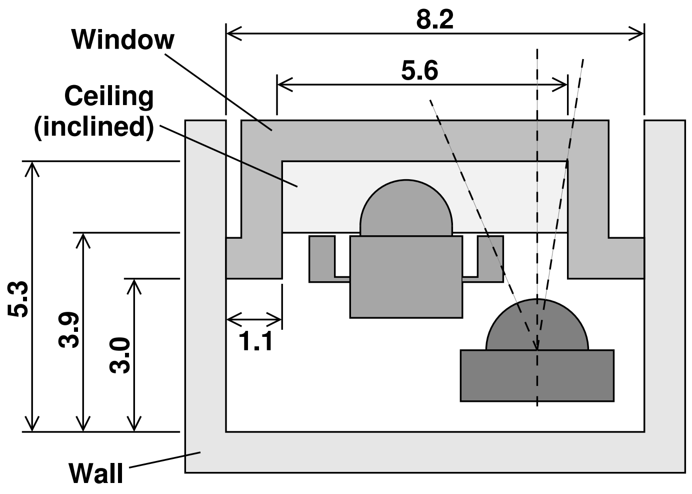
[11] `http://www.vishay.com/docs/83493/vemd2023slx01.pdf`





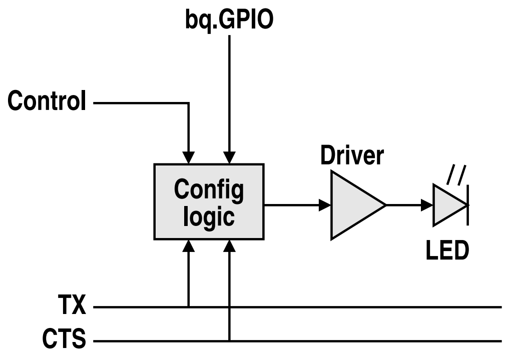
The transmitter should be closer to the window since its performance is more important than that of the receiver.

The following drawing shows a **bottom view** (i.e., we look from the PCB at the bottom of the components and into the plastic case) of the chamber that contains the IR diodes. The indicated placement maintains a clearance of at least 5 mm between components and between components and walls.



7 Transmitter circuit

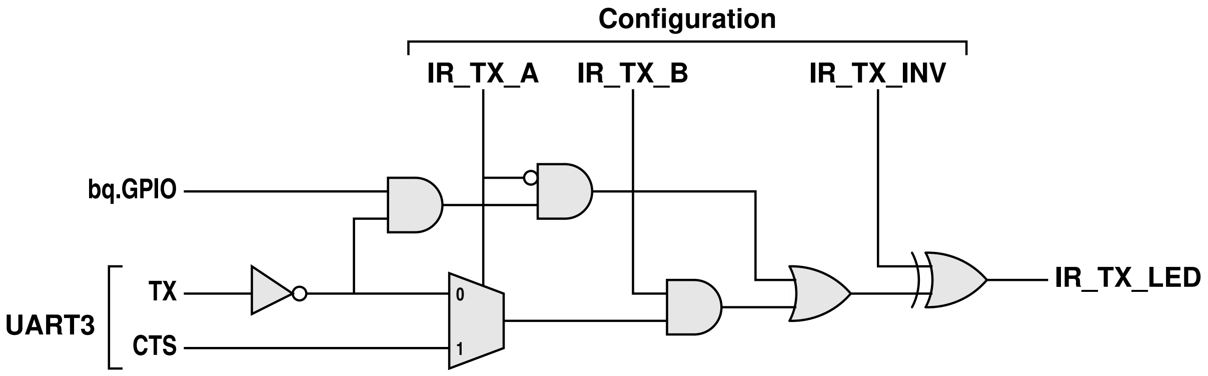
The transmitter circuit consists of the configuration logic that chooses which signal to output depending on the configuration requested by the CPU and the bq.GPIO input, the LED driver that amplifies the output of the configuration logic, and finally the LED.



7.1 Configuration logic

The configuration logic selects either TX, CTS, or "off" depending on the mode of operation (section 3). The polarity of the TX signal depends on whether we operate in IR-UART or IrDA mode (section 3.7). Finally, if the configuration inputs are in their reset state, the mode is determined by bq.GPIO (section 3.1).

With the assumption that the configuration inputs default to "0" after reset, we can accomplish all this with the following circuit:



The table below shows how the various modes are selected:

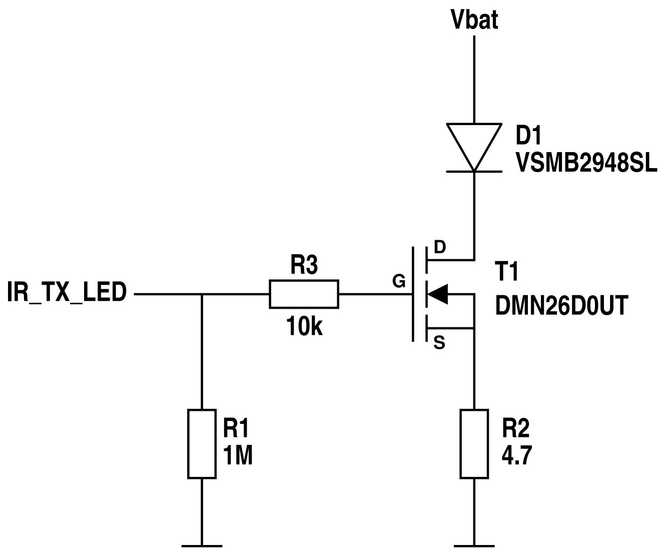
System state	IR_TX_...			bq.GPIO	IR_TX_LED	Mode
	A	B	INV			
Reset	0	0	0	0	0	Off or UART
	0	0	0	1	¬TX	IR-UART
Configured	1	0	0	X	0	Off or UART
	1	1	0	X	CTS	CIR
	0	1	0	X	¬TX	IR-UART
	0	1	1	X	TX	IrDA

Since the circuit is too complicated to be implemented efficiently with discrete logic, we use a Silego SLG46533 mixed-signal array [1]. This is the same chip also used for power selection in the Neo900 SIM switch [2].[12]

7.2 LED driver

We drive the transmit LED through an n-FET in the current-limiting configuration shown below.

The underlying concept is that the diode current flowing through R2 raises the voltage at the source (S) of T1 until V_{GS} is at the level where the FET admits exactly the desired current.



The maximum continuous forward current of the IR LED (VSMB2948SL) is 100 mA. We design the circuit to operate at no more than 90 mA, so that even a LED accidently left turned on permanently will not overheat.

The DMN26D0UT n-FET[13] is rated at 300 mW, a continuous I_D current of 230 mA, and admits 90 mA when V_{GS} is at about 1.4 V (figure 2 in the data sheet).

[12] Since this chip can also communicate over I^2C, we will to use that instead of dedicated external configuration signals.

[13] `http://www.diodes.com/_files/datasheets/ds31854.pdf`

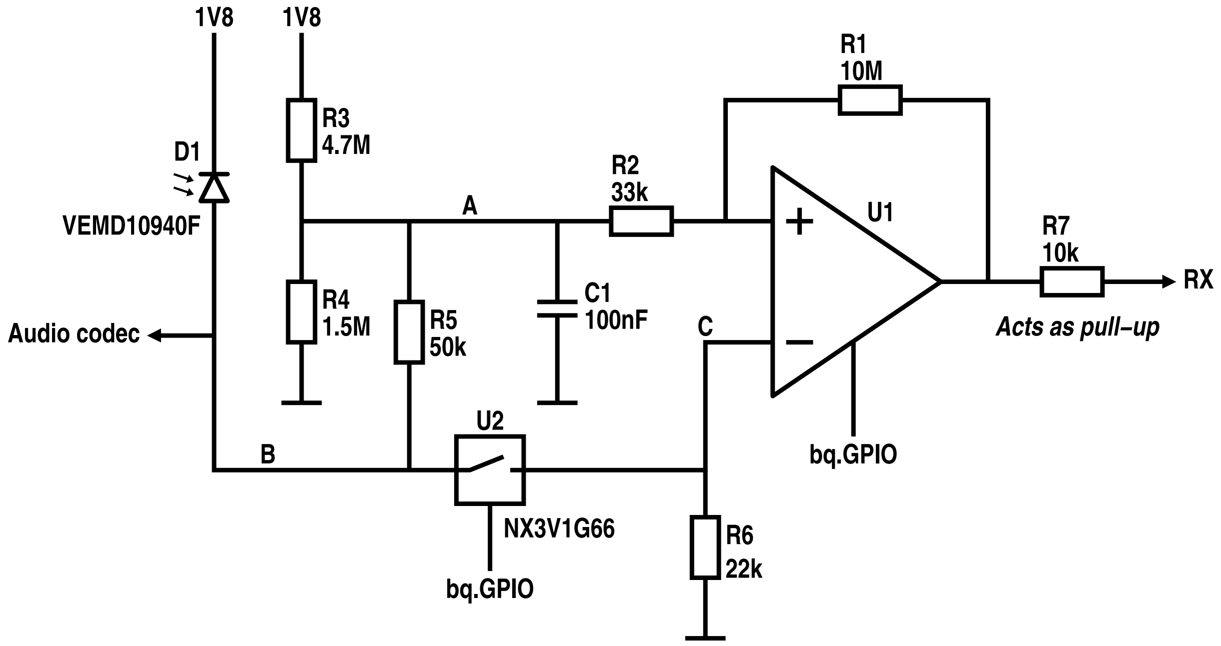
With a voltage of $V_{\mathrm{OH}} = 1.8$ V on IR_TX_LED, resistor R2 has to drop $V_{\mathrm{OH}} - V_{\mathrm{GS}} = 0.40$ V to reach the desired equilibrium. This corresponds to a resistance of 4.44 Ω. We therefore use a part with a nominal resistance of 4.7 Ω.

Power dissipation in R2 is ${I_{\mathrm{LED}}}^2 \cdot \mathrm{R2} = 38$ mW and thus requires no special consideration. With battery voltage $V_{\mathrm{BAT}} = 4.3$ V and LED forward voltage $V_{\mathrm{F}} = 1.35$ V, the voltage drop across the FET is $V_{\mathrm{DS}} = V_{\mathrm{BAT}} - V_{\mathrm{F}} - I_{\mathrm{LED}} \cdot \mathrm{R2} = 2.53$ V, and the power dissipation 227 mW.

Resistor R3 acts with the FET gate capacitance as low-pass filter to control the slew rate. Using $C_{\mathrm{ISS}} \approx 15$ pF at $V_{\mathrm{DS}} \approx 4$V, we get a cut-off frequency of about 1 MHz with R = 10 kOhm. (See also section 2.4.)

8 Receiver circuit

The following diagram shows the IR receiver circuit:



When the receiver is enabled (analog switch U2 is closed), D1 is reverse-biased through R6. R6 pulls the signal at **B** close to ground when the diode is not illuminated, and lets the voltage raise when D1 is illuminated and the reverse current through it increases.

The high-pass filter is implemented by comparing the signal from the sensor (**B** and **C**) with the same signal after passing the low-pass filter formed by R5 and C1 (**A**).

When the diode is not illuminated, R5 also ensures that the voltage at **A** is higher than at **C**, thus adding a threshold for signal detection.

The comparator U1 operates as inverting Schmitt-trigger, with R1 and R2 determining the hysteresis.

When the receiver is disabled (analog switch U2 is open) and the sensor is not illuminated, the voltage divider formed by R3 and R4 (with some contribution from R1) lets C1 charge to the expected operating point. R6 pulls **C** to ground in this case.

R7 ensures that CPU and Hackerbus can override the IR receiver, especially when disabled or not illuminated.

The analog comparator is part of the mixed-signal array. (See section 7.1.)

9 References

[1] Silego Technology, Inc. *SLG46533 - GreenPAK Programmable Mixed-signal Matrix*, Rev 1.06, October, 2016. `http://www.silego.com/uploads/Products/product_482/SLG46533r106_10202016.pdf`

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