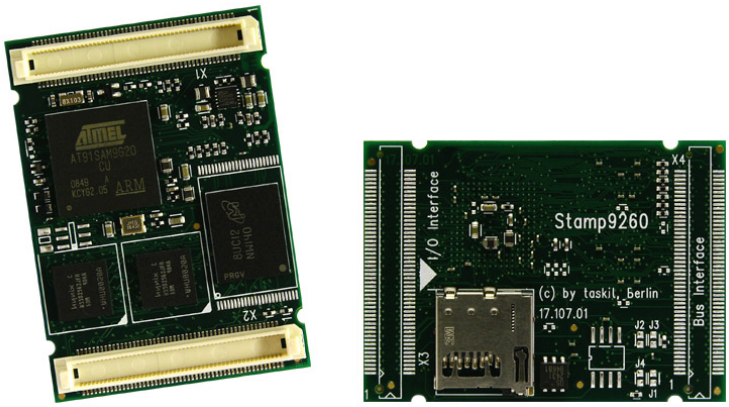

Stamp9G20

Technical Reference

Stamp9G20: Technical Reference

This document was generated on 2012-10-17T12:10:37+02:00.

Table of Contents

List of Figures

List of Tables

1. Introduction

The Stamp9G20 is intended to be used as a small size "intelligent" CPU module as well as a universal Linux CPU card. It can be used anywhere where restricted energy and space requirements play a role. The design of the Stamp9G20 is limited to the processors core needs like SDRAM and Flash, thus giving the customer a wide-ranged choice of configurations of the peripherals and environment.

The Stamp9G20 has all the necessary interfaces to support a huge variety of peripheral devices. Equipped with a 16-Bit parallel bus it gives fast access to a number of chips and additional devices.

The ARM architecture as a modern and widely supported processor architecture is currently the platform of choice for medium performance embedded devices. Almost all major processor manufacturers have ARM products in their portfolio.

The availability of the widespread operating system "Linux" for the ARM platform opens access to a broad range of software, including tools, drivers, and software libraries. Programs written for ARM can easily be employed on the PC platform for testing and debugging.

Examples of actual or potential applications are: protocol converters, measuring and test equipment, data-logging, as well as any simple or more complex control and automation tasks.

2. Scope

This document describes the most important hardware features of the Stamp9G20. It includes all informations necessary to develop a customer specific hardware for the Stamp9G20. The Operating System Linux is described in a further document.

The manual comprises only a brief description of the AT91SAM9G20 processor, as this is already described in depth in the manual of the manufacturer Atmel. Descriptions of the ARM core ARM926EJ-S are available from Atmel and also at *http://www.arm.com*. It is much recommended to have a look at these documents for a thorough understanding of the processor and its integrated peripherals.

3. Overview of Technical Characteristics

3.1. CPU

Atmel AT91SAM9G20 Embedded Processor featuring an ARM926EJ-S™ ARM® Thumb® Core

- CPU Clock 396 MHz
- 32KB Instruction Cache
- 32KB Data Cache
- Memory Management Unit (MMU)
- 3.3V Supply Voltage, 1.8V Memory Bus Voltage, 1.0V Core Voltage

3.2. Memory

- 128 MB NAND flash memory (optional more)
- 64 MB SDRAM (optional 128 MB)
- 2 x 16 KB SRAM
- 128 Bytes EEPROM
- Serial Dataflash optional possible

3.3. Interfaces and external signals

- 2x 100-pin fine-pitch low-profile Connectors (Hirose FX8)
- Ethernet 10/100 Mbit MAC
- Dual USB 2.0 Full Speed (12 MBit/s) Hosts
- USB 2.0 Full Speed (12 MBit/s) Device
- Six USARTs
- One UART
- One Synchronous Serial Controller (SSC, I^2S)
- Two Serial Peripheral Interfaces (SPI)
- One Two Wire Interface (TWI, I^2C)
- Two MultiMedia Card Interfaces
- JTAG debug port

- Digital Ports - up to 80 available
- Control Signals: IRQs, BMS, SHDN, WKUP
- 4 Programmable Clocks
- External RTC (optional)
- Image Sensor Interface
- Analog-to-Digital Converter
- 16-Bit parallel CPU-Bus

Some of the various functions are realized by multiplexing connector pins; therefore not all functions may be used at the same time (see Appendix D, *Stamp9G20 Pin Assignment*)).

3.4. Miscellaneous

- Four 16-Bit Timer/Counter
- Real Time Timer (RTT), with battery backup support
- Periodic Interval Timer (PIT)
- Watchdog Timer (WDT)
- Unique Hardware Serial Number

3.5. Power Supply

- 3.3V power supply
- 3V backup power supply, e.g. from a lithium battery

3.6. Dimensions

- Dimensions: 53x38x6.1 mmm (WxDxH)

4. Hardware Description

4.1. Mechanics

The Stamp9G20 was designed as a flexible CPU-Module, which can be connected to base boards via 2x 100-pin fine pitch low profile Hirose® FX8 connectors

The size of the Stamp9G20's PCB is only 53x38x6.1 mm fitting it in even the smallest design. While having implemented the sensible CPU, SDRAM and Flash design it still exports almost all possible CPU-Pins on it's connectors to allow a flexible design on base boards

The Stamp9G20 has a on-board Micro SD-Card slot, thus supporting even large memories needs in its compact design

4.2. AT91SAM9G20 Processor Core

The AT91SAM9G20 runs at 396 MHz with a memory bus frequency of 132 MHz.

Here are some of the most important features of the SAM9G20 ARM926EJ-S core:

- 32 Kbyte Data Cache, 32 Kbyte Instruction Cache, Write Buffer
- 32 Bit Data Bus
- ARM v4 and v5 Memory Management Unit (MMU)
- ARM v5 32-bit Instruction Set, ARM Thumb 16-bit Instruction Set supported
- DSP Instruction Extensions
- ARM Jazelle® Technology for Java® Acceleration
- EmbeddedICE™ Debug Communication Channel Support

Some of these features - like Jazelle - are currently not supported by the operating system of the product.

4.3. Memory

The Stamp9G20 is equipped with 32-Bit CPU-bus. Only a 16-Bit bus is exported on the interface connectors of the Stamp9G20. The memory bus voltage is 1.8 V and runs at 132 MHz. The memory bus voltage is different from normal operating voltage, which is 3.3 V. This has to be considered, when designing additional peripherals connected to the memory bus. Eventually buffer chips are necessary.

4.3.1. NAND Flash

The Stamp9G20 is equipped with a 128 MB NAND flash with 100000 erase and write cycles.It is organized in 128KB blocks. Customer specific adaptations are possible up to 512 MB on-board NAND flash. It is connected to chip select three (NCS3) of the microcontroller.

NAND flash has a different organisation of transistors than the common used NOR flash. While it allows a much higher density and thus increases the storage amount, it leads to some differences which need to be kept in mind.

Typically NAND flash is organized in pages and blocks, similar to hard disks. Pages are 512, 2048 or 4096 bytes in size, typical block sizes are 16, 128, 256 or 512 KB. Reading and programming are performed on page basis. Programming can only be done sequently in one block.

Additionally NAND flash requires bad block management, either by the driver software or by a seperate controller chip. Most NAND devices are shipped with bad blocks. These are identified and marked according to a specified bad block strategy. Further bad blocks may be detected during runtime. They are detected via an ECC. If a bad block is detected the data is written to a different, good block, and the bad block table is updated. So the overall memory capacity gradually shrinks as more and more blocks are marked bad.

This error detection is done by software like U-boot and Linux. Additionally NAND flash is subject to limited write and erase cycles. These are typically 100.000 cycles per block. So it is highly recomended to use wear levelling filesystems.

4.3.2. SDRAM

The Stamp9G20 is equipped with 64MB SDRAM. Customer specific adaptations allow configurations up to 128MB. The SDRAM is connected to chip select one (NCS1) of the micrcontroller.

SDRAM allows random access to any of its memory area and is volatile memory. SDRAM has synchronous interface, that means it waits for a clock signal before responding to control inputs and is therefore synchronized with the CPU bus. The clock is used to drive a final state machine in the chip, which allows to accept new instructions, before the previous one has finished executing.

4.3.3. EEPROM

The Stamp9G20 is equipped with a 128 bytes EEPROM, connected to the Dallas™ 1 wire bus.

EEPROM stands for Electrically Erasable Programmable Read-Only Memory and is non-volatile memory, which is used to store small amounts of data like calibration or configuration data. EEPROMS are byte-wise erasable, thus allowing true random access.

4.3.4. SRAM

The Stamp9G20's microcontroller is equipped with 2 x 16 KB internal SRAM. The internal SRAM can be accessed in one bus cycle and may be used for time critical sections of code or interrupt handlers.

4.3.5. DataFlash

The Stamp9G20 can optionally be equipped with DataFlash™.

DataFlash is a Atmel® proprietary interface and is compatible to the SPI standard. Similar to other flash chips it can be addressed page orientated and is available in sizes up to 8MB. Furthermore it is a possible boot media for the microcontroller.

4.4. Bus Matrix

The bus matrix of AT91SAM-controllers allows many master and slave devices to be connected independly of each other. Each master has a decoder and can be defined specially for each master. This allows concurrent access of masters to their slaves (provided the slave is available).

The bus matrix is thus the bridge between external devices connected to the EBI, the microcontroller's embedded peripherals and the CPU core.

Master 0	ARM926™ Instruction
Master 1	ARM926™ Data
Master 2	PDC
Master 3	ISI Controller
Master 4	Ethernet MAC
Master 5	USB Host DMA

Table 4.1. Bus Matrix Masters

Slave 0	Internal SRAM0 16KB
Slave 1	Internal SRAM1 16KB
Slave 2	Internal ROM / USB Host User Interface
Slave 3	External Bus Interface (EBI)
Slave 4	Internal Peripherals

Table 4.2. Bus Matrix Slaves

4.5. Advanced Interrupt Controller (AIC)

The Advanced Interrupt Controller can handle up to 32 internal or external interrupt sources. The AIC integrates an 8-level priority controller. Interrupt sources can be programmed to be level sensitive or edge triggered. The polarity can be programmed for all external interrupt sources.

Moreover, all PIO lines can be used to generate a PIO interrupt. However, the PIO lines can only generate level change interrupts, that is, positive as well as negative edges will generate an interrupt. The PIO interrupt itself (PIO to AIC line) is usually programmed to be level-sensitive. Otherwise interrupts will be lost if multiple PIO lines source an interrupt simultaneously.

On the Stamp9G20 IRQ0, IRQ2 and the FIQ are available. The list of peripheral identifiers, which are used to program the AIC can be found in Table B.1, “Peripheral Identifiers”

4.6. Battery Backup

The following parts of the AT91SAM9G20 Processor can be backed-up by a battery:

- Slow Clock Oscillator
- Real Time Timer
- Reset Controller
- Shutdown Controller
- General Purpose Backup Registers

It is recommended to always use a backup power supply (normally a battery) in order to speed up the boot-up time and to avoid reset problems.

4.7. Reset Controller (RSTC)

The embedded microcontroller has an integrated Reset Controller which samples the backup and the core voltage. The presence of a backup voltage (VDDBU) when the card is powered down speeds up the boot time of the microcontroller.

4.8. Serial Number

Every Stamp9G20 has a unique 48-bit hardware serial number chip which can be used by application software. The chip is a Dallas® one-wire-chip. A Linux driver is provided. Additionally it functions as the 128 Byte EEPROM.

4.9. Clock Generation

4.9.1. Processor Clocks

The embedded microcontroller generates its necessary clocks based on two crystal oscillators: One slow clock (SLCK) oscillator running at 32.768 KHz and one main clock oscillator running at 18.432 MHz. It has furthermore a internal slow clock oscillator, which can be battery backed-up.

From the main clock oscillator, the microcontroller generates two further clocks by using two PLLs. PLLA provides the processor clock (PCK) and the master clock (MCK). Typically PLLB provides the 48 MHz USB Clock and is normally used only for this purpose. The clocks of most peripherals derive from MCK. These include EBI, USART, SPI, TWI, SSC, PIT and TC.

Some peripherals like the programmable clocks and the timer counters (TC) can also run on SLCK. The real time timer (RTT) runs always on SLCK.

Clock	Frequency	Source
PCK (Processor Clock)	396 MHz	PLLA
MCK (Master Clock)	132 MHz	PCK/3
USB Clock	48 MHz	PLLB
Slow Clock	32.768 KHz	Slow Clock Oscillator

Table 4.3. AT91SAM9G20 Clocks

4.9.2. Programmable Clocks

The programmable clocks can be individually programmed to derive their input from SLCK, PLLA, PLLB and Main Clock. Each PCK has a divider of 2, 4, 8, 16, 32 or 64.

The Stamp9G20 features two programmable clocks PCK0, PCK1.

4.10. Power Management Controller (PMC)

4.10.1. Function

The PMC has a Peripheral Clock register which allows enabling or disabling of the clocks of all integrated peripherals individually using their "Peripheral Identifier" (see Table B.1, “Peripheral Identifiers”). The System Clock register allows enabling or disabling of each of the following clocks individually:

- Processor Clock
- ISI Clock
- USB Host Clock (common for both channels)
- USB Device Clock
- Programmable Clocks

The PMC status register provides "Clock Ready" or, respectively, "PLL Lock" status bits for each of these clocks. An interrupt is generated when any of these bits changes from 0 to 1. The PMC provides status flags for the

- Main Oscillator
- Master Clock
- PLLA
- PLLB
- Programmable Clocks

The Main Oscillator frequency can be measured by using the PMC Main Clock Frequency register. The SLCK is used as reference for the measurement.

4.10.2. Power Management

Using power management can dramatically reduce the power consumption of an Embedded Device. Via the PMC various clocks can be disabled or their speed reduced:

- stopping the PLLs (PLLA and / or PLLB)
- stopping the clocks of the various peripherals

- reducing the clock rates of peripherals, especially by changing MCK.

The PMC supports the following power-saving features: Idle mode and power-down mode. Please note, that not every operating system supports these modes.

- **Idle Mode.** In idle mode, the processor clock will be re-enabled by any interrupt. The peripherals, however, are only able to generate an interrupt if they still have a clock, so care has to be taken as to when a peripheral can be powered down.

- **Power-down Mode.** In many cases a system waits for a user action or some other rare event. In such a case, it is possible to change MCK to SLCK. Any external event which changes the state on peripheral pins (not the USB) can then be detected by the PIO controller or the AIC.

  It should also be taken into account that when a PLL is stopped it will take some time to restart it. Changing the PLL frequencies or stopping them can therefore be done only at a moderate rate. If short reaction times are required, this is not a choice.

Additionally further measurements can reduce power consumption considerably:

- switching off the TFT supply voltage

- putting peripheral chips like Ethernet controller and / or PHY or serial driver devices in power down mode

- putting the SDRAM into self-refresh mode

4.11. Real-time Timer (RTT)

The Real-time Timer is a 32-bit counter combined with a 16-bit prescaler running at Slow Clock (SLCK = 32768 Hz). As the RTT keeps running if only the backup supply voltage is available, it is used as a Real-time clock.

The RTT can generate an interrupt every time the prescaler rolls over. Usually the RTT is configured to generate an interrupt every second, so the prescaler will be programmed with the value 7FFFh.

The RTT can also generate an alarm if a preprogrammed 32-bit value is reached by the counter.

4.12. Timer Counter (TC)

The Stamp9G20 features two blocks of timer counters with three counters each. Due to multiplexing four timer counters may be used with external signals.

The TC consists of three independent 16-bit Timer/Counter units. They may be cascaded to form a 32-bit or 48-bit timer/counter. The timers can run on the internal clock sources MCK/2, MCK/8, MCK/32, MCK/128, SLCK or the output of another timer channel. External clocks may be used as well as the counters can generate signals on timer events. They also can be used to generate PWM signals.

4.13. Periodic Interval Timer (PIT)

The PIT consists of a 20-bit counter running on MCK / 16. This counter can be preloaded with any value between 1 and 2^{20}. The counter increments until the preloaded value is reached. At this stage it rolls over and generates an interrupt. An additional 12-bit counter counts the interrupts of the 20 bit counter.

The PIT is intended for use as the operating system's scheduler interrupt.

4.14. Watchdog Timer

The watchdog timer is a 12-bit timer running at 256 Hz (Slow Clock / 128). The maximum watchdog timeout period is therefore equal to 16 seconds. If enabled, the watchdog timer asserts a hardware reset at the end of the timeout period. The application program must always reset the watchdog timer before the timeout is reached. If an application program has crashed for some reason, the watchdog timer will reset the system, thereby reproducing a well defined state once again.

The Watchdog Mode Register can be written only once. After a processor reset, the watchdog is already activated and running with the maximum timeout period. Once the watchdog has been reconfigured or deactivated by writing to the Watchdog Mode Register, only a processor reset can change its mode once again.

4.15. Peripheral DMA Controller (PDC)

The Peripheral DMA Controller (PDC) transfers data between on-chip serial peripherals and the on- and/or off-chip memories. The PDC contains unidirectional and bidirectional channels. The full-duplex peripherals feature unidirectional channels used in pairs (transmit only or receive only). The half-duplex peripherals feature one bidirectional channel. Typically full-duplex peripherals are USARTs, SPI or SSC. The MCI is a half duplex device.

The user interface of each PDC channel is integrated into the user interface of the peripheral it serves. The user interface of unidirectional channels (receive only or transmit only), contains two 32-bit memory pointers and two 16-bit counters, one set (pointer, counter) for current transfer and one set (pointer, counter) for next transfer. The bidirectional channel user interface contains four 32-bit memory pointers and four 16-bit counters. Each set (pointer, counter) is used by current transmit, next transmit, current receive and next receive.

Using the PDC removes processor overhead by reducing its intervention during the transfer. This significantly reduces the number of clock cycles required for a data transfer, which improves microcontroller performance. To launch a transfer, the peripheral triggers its associated PDC channels by using transmit and receive signals. When the programmed data is transferred, an end of transfer interrupt is generated by the peripheral itself. There are four kinds of interrupts generated by the PDC:

- End of Receive Buffer
- End of Transmit Buffer

- Receive Buffer Full
- Transmit Buffer Empty

The "End of Receive Buffer" / "End of Transmit Buffer" interrupts signify that the DMA counter has reached zero. The DMA pointer and counter register will be reloaded from the reload registers ("DMA new pointer register" and "DMA new counter register") provided that the "DMA new counter register" has a non-zero value. Otherwise a "Receive Buffer Full" or, respectively, a "Transmit Buffer Empty" interrupt is generated, and the DMA transfer terminates. Both reload registers are set to zero automatically after having been copied to the DMA pointer and counter registers.

4.16. Debug Unit (DBGU)

The Debug Unit is a simple UART which provides only RX/TX lines. It is used as a simple serial console for Firmware and Operating Systems.

4.17. JTAG Unit

The JTAG unit can be used for hardware diagnostics, hardware initialization, flash memory programming, and debug purposes. The JTAG unit supports two different modes, namely the "ICE Mode", and the "Boundary Scan" mode. It is normally jumpered for "ICE Mode".

JTAG interface devices are available for the unit. However, the use of them is not within the scope of this document.

4.18. Two-wire Interface (TWI)

The TWI is also known under the expression "I^2C-Bus", which is a trademark of Philips and may therefore not be used by other manufacturers. However, interoperability is guaranteed. The TWI supports both master or slave mode.

The TWI uses only two lines, namely serial data (SDA) and serial clock (SCL). According to the standard, the TWI clock rate is limited to 400 kHz in fast mode and 100 kHz in normal mode, but configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.

Caveat: The TWI hardware unit has been known to be error prone in various microcontrollers, which has lead operating systems like Linux to use a bit-banging driver on the same pins instead.

4.19. Multimedia Card Interface (MCI)

The Stamp9G20 features a onboard Micro-SD-Card slot, which is connected to the MCI-A interface of the microcontroller. Please note that this is the same slot used on the Evaluation Board, so only one interface can be used at a time. Also operating systems like Linux do not necessarily support all features of the hardware unit.

The MultiMedia Card Interface (MCI) supports the MultiMedia Card (MMC) Specification V3.11, the SDIO Specification V1.1 and the SD Memory Card Specification V1.0.

The MCI includes a command register, response registers, data registers, timeout counters and error detection logic that automatically handle the transmission of commands and, when required, the reception of the associated responses and data with a limited processor overhead. The MCI supports stream, block and multi-block data read and write, and is compatible with the Peripheral DMA Controller (PDC) channels, minimizing processor intervention for large buffer transfers.

The MCI operates at a rate of up to Master Clock divided by 2 and supports the interfacing of 2 slot(s). Each slot may be used to interface with a MultiMediaCard bus (up to 30 Cards) or with a SD Memory Card. Only one slot can be selected at a time (slots are multiplexed). A bit field in the SD Card Register performs this selection.

The SD Memory Card communication is based on a 9-pin interface (clock, command, four data and three power lines) and the MultiMedia Card on a 7-pin interface (clock, command, one data, three power lines and one reserved for future use). The SD Memory Card interface also supports MultiMedia Card operations. The main differences between SD and MultiMedia Cards are the initialization process and the bus topology.

4.20. USB Host Port (UHP)

The Stamp9G20 integrates two USB host ports supporting speeds up to 12 MBit/s.

The USB Host Port (UHP) interfaces the USB with the host application. It handles Open HCI protocol (Open Host Controller Interface) as well as USB v2.0 Full-speed and Low-speed protocols.

The USB Host Port integrates a root hub and transceivers on downstream ports. It provides several high-speed half-duplex serial communication ports. Up to 127 USB devices (printer, camera, mouse, keyboard, disk, etc.) and an USB hub can be connected to the USB host in the USB "tiered star" topology.

4.21. USB Device Port (UDP)

The Stamp9G20 integrates one USB device port supporting speeds up to 12 MBit/s.

The USB Device Port (UDP) is compliant with the Universal Serial Bus (USB) V2.0 full-speed device specification. The USB device port enables the product to act as a device to other host controllers.

The USB device port can also be implemented to power on the board. One I/O line may be used by the application to check that VBUS is still available from the host. Self-powered devices may use this entry to be notified that the host has been powered off. In this case, the pullup on DP must be disabled in order to prevent feeding current to the host. The application should disconnect the transceiver, then remove the pullup.

4.22. Ethernet MAC (EMAC)

The EMAC module implements a 10/100 MBit/s Ethernet MAC compatible with the IEEE 802.3 standard using an address checker, statistics and control registers, receive and transmit blocks, and a DMA interface.

The address checker recognizes four specific 48-bit addresses and contains a 64-bit hash register for matching multicast and unicast addresses. It can recognize the broadcast address of all ones, copy all frames, and act on an external address match signal.

An individual 48-bit MAC address (ETHERNET hardware address) is allocated to each product. This number is stored in flash memory. It is recommended not to change the MAC address in order to comply with IEEE Ethernet standards.

To completely implement ethernet an additional physical layer interface is needed (PHY). A sample implementation is found on the Adaptor Board.

4.23. Universal Sychronous Asynchronous Receiver and Transmitter (USART)

The Stamp9G20 has up to six independent USARTs, not including the debug unit.

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver time-out enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

The USART supports the connection to the Peripheral DMA Controller, which enables data transfers to the transmitter and from the receiver. The PDC provides chained buffer management without any intervention of the processor.

Six different modes are implemented within the USARTs:

- Normal (standard RS232 mode)
- RS485
- Hardware Handshaking
- ISO7816 Protocol: T=0 or T=1
- IrDA

RS485. In RS485 operating mode the RTS pin is automatically driven high during transmit operations. If RTS is connected to the "enable" line of the RS485 driver, the driver will thus be enabled only during transmit operations.

Hardware Handshaking. The hardware handshaking feature enables an out-of-band flow control by automatic management of the pins RTS and CTS. The receive DMA channel must be active for this mode. The RTS signal is driven high if the receiver is disabled or if the DMA indicates a buffer full condition. As the RTS signal is connected to the CTS line of the connected device, its transmitter is thus prevented from sending any more characters.

ISO7816. The USARTs have an ISO7816-compatible mode which permits interfacing with smart cards and Security Access Modules (SAM). Both T=0 and T=1 protocols of the ISO7816 specification are supported.

IrDA. The USART features an infrared (IrDA) mode supplying half-duplex point-to-point wireless communication. It includes the modulator and demodulator which allows a glueless connection to the infrared transceivers. The modulator and demodulator are compliant with the IrDA specification version 1.1 and support data transfer speeds ranging from 2.4 kb/s to 115.2 kb/s.

Signals of the Serial Interfaces. All UARTs/USARTs have one receiver and one transmitter data line (full duplex). Not all USARTs are implemented with full modem control lines. Furthermore the available lines depend largely on the used multiplexing. Most modem control lines can be implemented with standard digital ports.

Hardware Interrupts. There are several interrupt sources for each USART:

- Receive: RX Ready, (DMA) Buffer Full, End of Receive Buffer
- Transmit: TX Ready, (DMA) Buffer Empty, End of Transmit Buffer, Shift Register Empty
- Errors: overrun, parity, framing, and timeout errors
- Handshake: the status of CTS has changed
- Break: the receiver has detected a break condition on RXD
- NACK: non acknowledge (ISO7816 mode only)
- Iteration: the maximum number of repetitions has been reached (ISO7816 mode only)

Please refer to the chapter about the DMA unit (PDC) for a description of the "Buffer Full" and "End of Receive / Transmit Buffer" events.

4.24. Synchronous Peripheral Interface (SPI)

The Stamp9G20 features two SPI ports, each of them with three chip selects.

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Master or Slave Mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the "master" which controls the data flow, while the other devices act as "slaves" which have data shifted into and out by the master.

A slave device is selected when the master asserts its NSS signal. If multiple slave devices exist, the master generates a separate slave select signal for each slave (NPCS). The SPI system consists of two data lines and two control lines:

- Master Out Slave In (MOSI): This data line supplies the output data from the master shifted into the input(s) of the slave(s).

- Master In Slave Out (MISO): This data line supplies the output data from a slave to the input of the master. There may be no more than one slave transmitting data during any particular transfer.
- Serial Clock (SPCK): This control line is driven by the master and regulates the flow of the data bits. The master may transmit data at a variety of baud rates; the SPCK line cycles once for each bit that is transmitted. The SPI baudrate is Master Clock (MCK) divided by a value between 1 and 255
- Slave Select (NSS): This control line allows slaves to be turned on and off by hardware.

Each SPI Controller has a dedicated receive and transmit DMA channel.

4.25. Synchronous Serial Controller (SSC)

The Stamp9G20 has one SSC interface available, depending on the multiplexing of the pins.

The SSC supports many serial synchronous communication protocols generally used in audio and telecom applications such as I2S, Short Frame Sync, Long Frame Sync, etc.

The SSC has separated receive and transmit channels. Each channel has a data, a clock and a frame synchronization signal (RD, RK, RF, resp. TD, TK, TF). Both a receive and a transmit DMA channel are assigned to each SSC.

4.26. Peripheral Input/Output Controller (PIO)

The Stamp9G20 has a maximum of 80 freely programmable digital I/O ports on its connectors. These pins are also used by other peripheral devices.

The Parallel Input/Output Controller(PIO) manages up to 32 programmable I/O ports. Each I/O port is associated with a bit number in the 32 bit register of the user interface. Each I/O port may be configured for general purpose I/O or assigned to a function of an integrated peripheral device. In doing so multiplexing with 2 integrated devices is possible. That means a pin may be used as GPIO, device A or device B. The PIO Controller also features a synchronous output providing up to 32 bits of data output in a single write operation.

The following characteristics are individually configurable for each PIO pin:

- PIO enable
- Peripheral A or B enable
- Output enable
- Output level
- Write Enable
- Level change interrupt
- Glitch filter: pulses that are lower than a half clock cycle are ignored

- Open-drain outputs
- Pull-up resistor

All configurations as well as the pin status can be read back by using the appropriate status register. Multiple pins of each PIO can also be written simultaneously by using the synchronous output register.

For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources 2 to 31. Refer to the PIO Controller peripheral identifier Table B.1, “Peripheral Identifiers” to identify the interrupt sources dedicated to the PIO Controllers. The PIO Controller interrupt can be generated only if the PIO Controller clock is enabled.

A number of the PIO signals might be used internally on the module. Care has to be taken when accessing the PIO registers in order not to change the settings of these internal signals, otherwise a system crash is likely to happen.

4.27. Image Sensor Interface (ISI)

The Image Sensor Interface (ISI) supports direct connection to the ITU-R BT. 601/656 8-bit mode compliant sensors and up to 12-bit grayscale sensors. It receives the image data stream from the image sensor on the 12-bit data bus. This module receives up to 12 bits for data, the horizontal and vertical synchronizations and the pixel clock. The reduced pin count alternative for synchronization is supported for sensors that embed SAV (start of active video) and EAV (end of active video) delimiters in the data stream.

The Image Sensor Interface interrupt line is generally connected to the Advanced Interrupt Controller and can trigger an interrupt at the beginning of each frame and at the end of a DMA frame transfer. If the SAV/EAV synchronization is used, an interrupt can be triggered on each delimiter event.

For 8-bit color sensors, the data stream received can be in several possible formats: YCbCr 4:2:2, RGB 8:8:8, RGB 5:6:5 and may be processed before the storage in memory. The data stream may be sent on both preview path and codec path if the bit CODEC_ON in the ISI_CR1 is one. To optimize the bandwidth, the codec path should be enabled only when a capture is required.

In grayscale mode, the input data stream is stored in memory without any processing. The 12-bit data, which represent the grayscale level for the pixel, is stored in memory one or two pixels per word, depending on the GS_MODE bit in the ISI_CR2 register. The codec datapath is not available when grayscale image is selected.

4.28. Analog Digital Converter (ADC)

The Stamp9G20 has a four ADC channels available.

The ADC is based on a Successive Approximation Register (SAR) 10-bit Analog-to-Digital Converter(ADC). It also integrates a 4-to-1 analog multiplexer, making possible the analog-to-digital conversions of 4 analog lines. The conversions extend from 0V to ADVREF.

The ADC supports an 8-bit or 10-bit resolution mode, and conversion results are reported in a common register for all channels, as well as in a channel-dedicated register. Software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) are configurable.

The ADC also integrates a Sleep Mode and a conversion sequencer and connects with a DMA channel. These features reduce both power consumption and processor intervention.

Finally, the user can configure ADC timings, such as Startup Time and Sample and Hold Time.

5. Design Considerations

5.1. Ethernet Controller (EMAC)

The emac needs an aditional PHY design. The emac supports both, MII and RMII interface.

Please take care of the specific layout requirements of the Ethernet port when designing a base board. The two signals of the transmitter pair (ETX+ and ETX-) should be routed in parallel (constant distance, e.g. 0.5mm) with no vias on their way to the RJ45-jack. The same is true for the receiver pair (ERX+ and ERX-). No other signals should be crossing or get next to these lines. If a ground plane is used on the base board, it should be omitted in the vicinity of the Ethernet signals.

A 1nF / 2kV capacitor should be connected between board ground and chassis ground (which is usually connected to the shield of the RJ45-jack).

5.2. USB Host Controller (UHP)

External Parts. A few external parts are required for the proper operation of the UHP:

- Pull-down resistors on each line of approximately 15 kΩ. These should be installed even if the UDP is not to be used at all in order to keep the signals from floating.
- Series resistors of 27 Ω (5%) on each line.
- Small capacitors (e.g. 15pF) to ground on each line (optional).
- ESD protection devices are recommended for applications which are subject to external contact. The restrictions with regard to capacitive loading have to be applied when selecting a protection device.
- A circuit to generate the 5V VBUS supply voltage.

V_{BUS} considerations for USB Host. A USB host port has to provide a supply voltage V_{BUS} of 5V +- 5% which has to be able to source a maximum of 500mA, or 100mA in case of battery operation. Please refer to the appropriate rules in the USB specification. A low ESR capacitor of at least 120µF has to be provided on V_{BUS} in order to avoid excessive voltage drops during current spikes.

V_{BUS} has to have an over-current protection. The over-current drawn temporarily on V_{BUS} must not exceed 5A. Polymeric PTCs or solid state switches are recommended by the specification. Suitable PPTCs are "MultiFuse" (Bourns), "PolyFuse" (Wickmann/Littelfuse), "PolySwitch" (Raychem/Tyco).

It is required that the over-current condition can be detected by software, so that V_{BUS} can be switched off or be reduced in power in such a case.

Layout considerations. The external resistors should be placed in the vicinity of the module's connector. The two traces of any of the differential pairs (USB-Host A+ and USB-Host A- , as well as USB-Host B+ and USB-Host B-) should not encircle large areas on

the base board, in order to reduce signal distortion and noise. The are preferably routed closely in parallel to the USB connector.

5.3. USB Device Controller (UDP)

External Parts. A few external parts are required for the proper operation of the UDP:

- Pull-down resistors on each line of approximately 330 kΩ. These should be installed even if the UDP is not to be used at all in order to keep the signals from floating.
- Series resistors of 27 Ω (5%) on each line.
- A voltage divider on the 5V USB supply voltage VBUS converting this voltage to 3.3V (1.8V), e.g. 27 kΩ / 47 kΩ, for the VBUS monitoring input (USB_CNX).
- ESD protection devices are recommended for applications which are subject to external contact. The restrictions with regard to capacitive loading have to be applied when selecting a protection device.

The USB specification (revision 2.0 of April 27, 2000, chapter 7.1.1.1) demands a cable with a characteristic impedance of 90 Ω ± 15%. Matching this impedance would require a driver output impedance of 45 Ω on both the negative and the positive driver. Nevertheless, the controller demands external resistors of 27 Ω. These would correspond to intrinsic resistances of the drivers of 18 Ω.

The USB specification demands a switchable pull-up resistor of 1.5 kΩ on USB-Device+ which identifies the UDP as a full speed device to the attached host controller. On this module, this resistor is integrated on the chip. It can be switched on or off using the "USB Pad Pull-up Control Register", which is part of the "Bus Matrix User Interface" (not the "USB Device Port User Interface", as one might expect). This pull-up resistor is required to be switchable in order not to source current to an attached but powered down host. This would otherwise constitute an irregular condition on the host. The software has to take care of this fact.

The capacitors are intended to improve the signal quality (edge rate control) in dependency of the specific design. They are not mandatory. The total capacitance to ground of each USB pin, the PCB trace to the series resistor, and the capacitor must not exceed 75pF.

Operation with V_{BUS} as a Supply. Special care has to be taken if the module is powered by the VBUS supply. Please refer to the appropriate rules in the USB specification with regard to inrush current limiting and power switching. As the module draws more than 100mA in normal mode, it is a "high-power" device according to the specification (<100mA = "low-power", 100..500mA = "high-power"). It therefore requires staged switching which means that at power-up it should draw not more than 100mA on VBUS. The capacitive load of a USB device on VBUS should be not higher than 10µF.

Layout considerations. The external resistors should be placed in the vicinity of the module's connector. The traces of the differential pair (USB-Device+ and USB-Device-) should not encircle large areas on the base board, in order to reduce signal distortion and noise. The are preferably routed closely in parallel to the USB connector.

5.4. Memory Bus

On the Stamp9G20 the memory bus is driven with 1.8V. This affects the voltages of PIOC-controller pins, they are 1.8V as well. Not affected are the ADC-Channels, which have their own ADV_{REF}. The V_{MEM} pins on the module are pin one and two of the bus interface. If pins of PIOC or the memory bus are in use on the customer's design it is highly recommended to implement buffers on both memory bus and PIOC pins.

The memory bus is used inside of the module. It can be either 1.8V or 3.3V. The V_{mem} pin of the module is powered by the module itself. Do not power this pin externally to maintain inter-product dependencies. A difference between V_{mem} and VCC may also affect the behaviour of one PIO-controller of the respective module.

To connect 3.3V chips to the memory bus or to maintain compatibility between various products it is recommended to implement buffer chips on the memory bus externally, like shown in Figure 5.1, “Buffered Memory Bus (PIOC) 1.8V - 3.3V”

To connect 5V chips the same schematics can be used with suitable buffer chips.

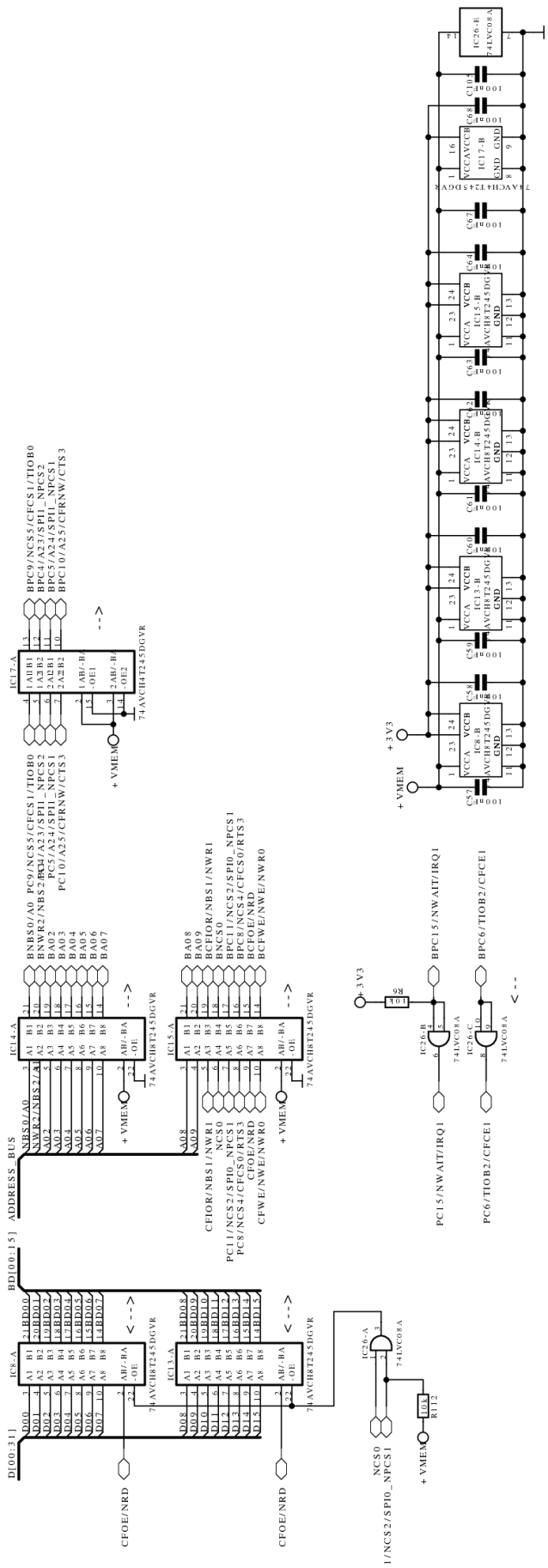

Figure 5.1. Buffered Memory Bus (PIOC) 1.8V - 3.3V

6. Stamp9G20 Starterkit

6.1. Starterkit Contents

The Stamp9G20 Starterkit contains the following components:

- Stamp9G20
- Stamp Adaptor 2: Adapter board for connecting Stamp9G20 and on Panel-Card EVB
- Panel-Card EVB: Evaluation and Prototyping Board
- Wall Adapter Power Supply, Input AC 230V, Output DC 9 to 16V, min. 400 mA
- Serial "Null-Modem" Cable with two 9-pin D-type Connectors
- Adapter cable for accessing the Debug UART
- CD with Operating System, Toolchain, and Documentation

6.2. Panel-Card EVB

The Panel-Card EVB (Evaluation Board) is designed to be both simple and universal. Some elements of the circuit board will not always be needed, but facilitate implementation for certain purposes. It was designed to serve the Panel-Card and other products as an evaluation platform.

6.2.1. First Steps

The Starter Kit board "Panel-Card EVB" makes it easy to put the module to use. The first steps involve the following:

- connecting the wall adapter to the main supply and to the board
- connecting the DBGU-Adapter and the serial cable to a COM port of a PC
- starting a terminal program for the selected COM port at 115200 baud, 8N1
- starting the module by flipping the power switch
- boot messages of the module are now expected to appear on the terminal

6.2.2. Power Supply

From an unregulated input voltage between 8 and 35V two voltages are produced:

- 3.3V for the CPU module,
- 5V for USB and optional peripherals

6.2.3. RS232 Interface

The RS232 port provides RS232 drivers/receivers for the RxD, TxD, RTS, CTS, RI and DCD signals of the one USART of the module. It is connected to X4 DSUB-9 connector on the EVB.

The RxD and TxD lines of the module's "Debug UART" are connected to X25 Debug on EVB. To use the console port of the Stamp9261, which usually is identical to the Debug UART, the serial adaptor cable has to be used. It connects the PC's TxD, RxD, and Ground lines to the appropriate pins of X25 Debug of the EVB.

6.2.4. Connectors

The following connectors are part of the Panel-Card EVB:

- Two 40-pin header connectors (0.1 inch pitch)
- Ethernet 10/100 MBit
- USB Host (dual)
- USB Device
- RS232
- MMC-Card Slot
- Matrix keyboard
- JTAG
- User Connector D-type 25pin
- User Connector 26pin header (optional)
- 2-contact terminal block for power supply
- DC connector for power supply

6.2.5. Rotary Encoder

The Panel-Card EVB includes a rotary encoder with push-button which is a "Human Interface Device" well suited for many embedded applications. Generally spoken, it is a simplified replacement for a computer mouse, touch pad, or keyboard. The rotary encoder represents a pointing device with only one dimension, while the push-button serves as a "return" key or mouse button.

The rotary encoder is accessed by two PIO ports of the module (see circuit diagram). The push-button occupies another PIO port.

6.2.6. Touch Controller

The Panel-Card EVB implements a ADS7843 Touch controller. It is connected to SPI of the module. The touch controller and the rotary encoder can only be used mutually exclusive. Connect jumper J2 to ID for the rotary encoder and to TP for touch.

6.2.7. Schematics

The following circuit diagram is intended for reference only and does not dispense the user from checking and applying the appropriate standards. No warranty can be granted if parts of the circuit are used in customer applications.

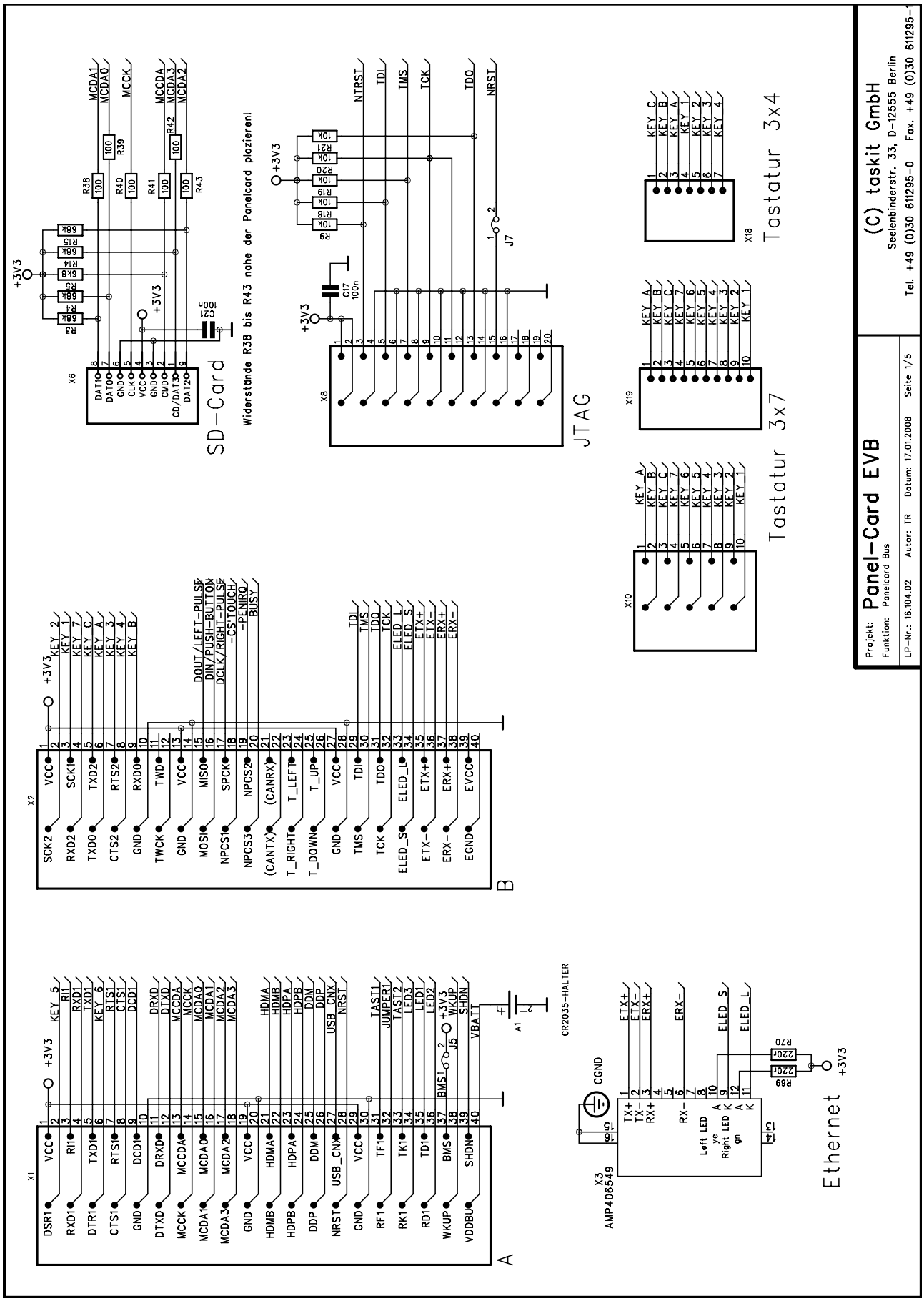

Figure 6.1. Panel-Card EVB Schematics Bus/JTAG

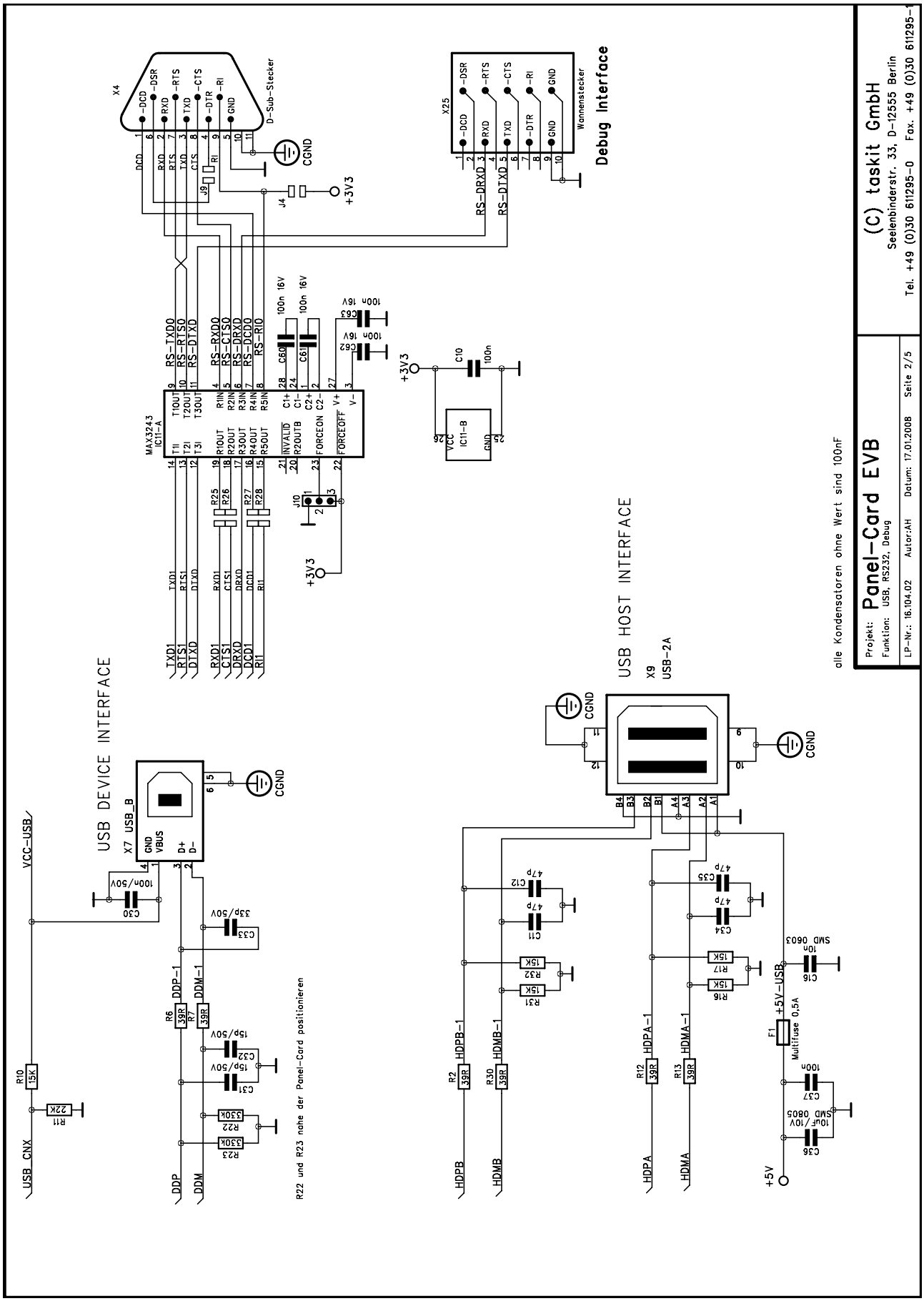

Figure 6.2. Panel-Card EVB Schematics USB/RS232

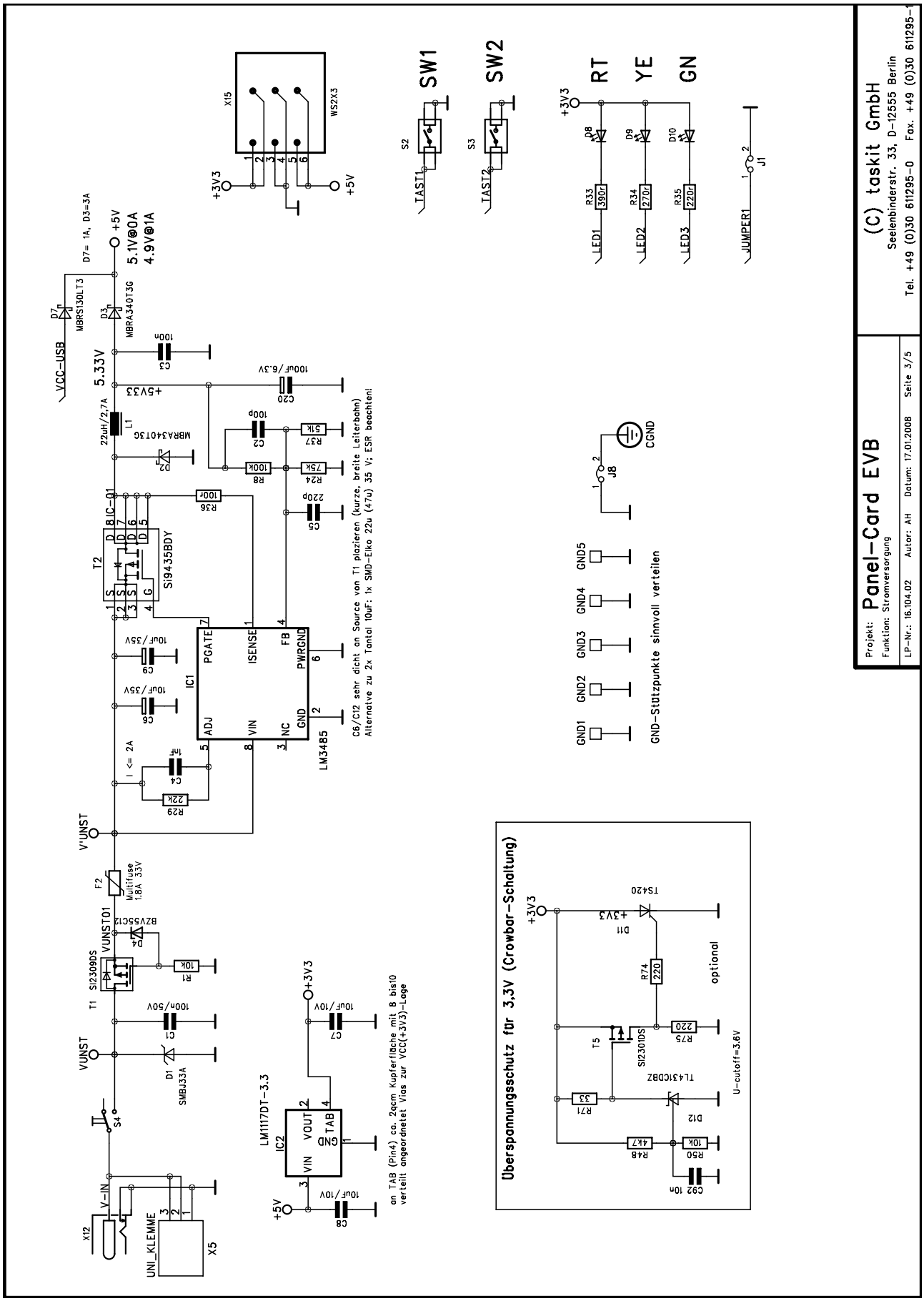

Figure 6.3. Panel-Card EVB Schematics Power Regulation

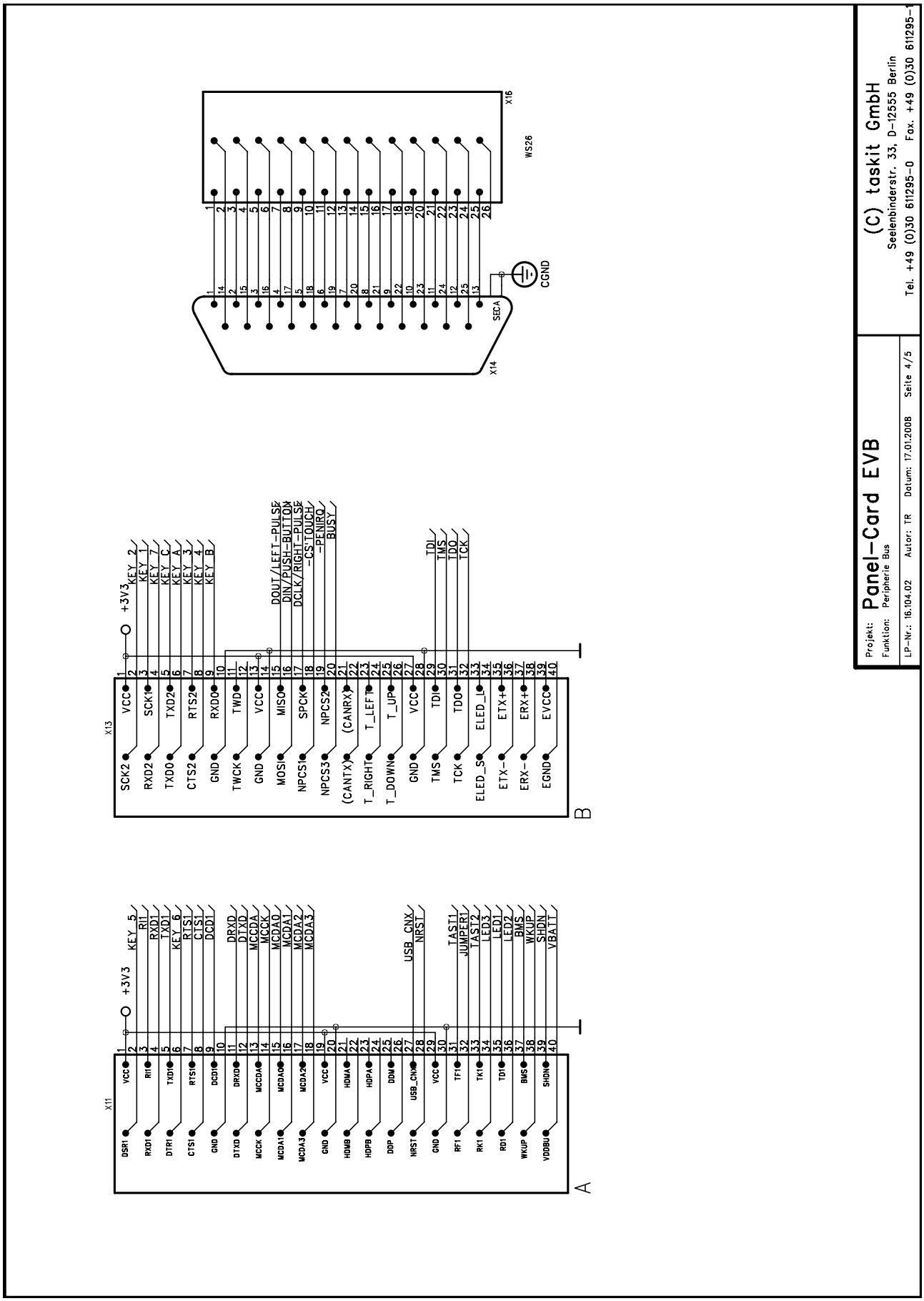

Figure 6.4. Panel-Card EVB Schematics Connectors

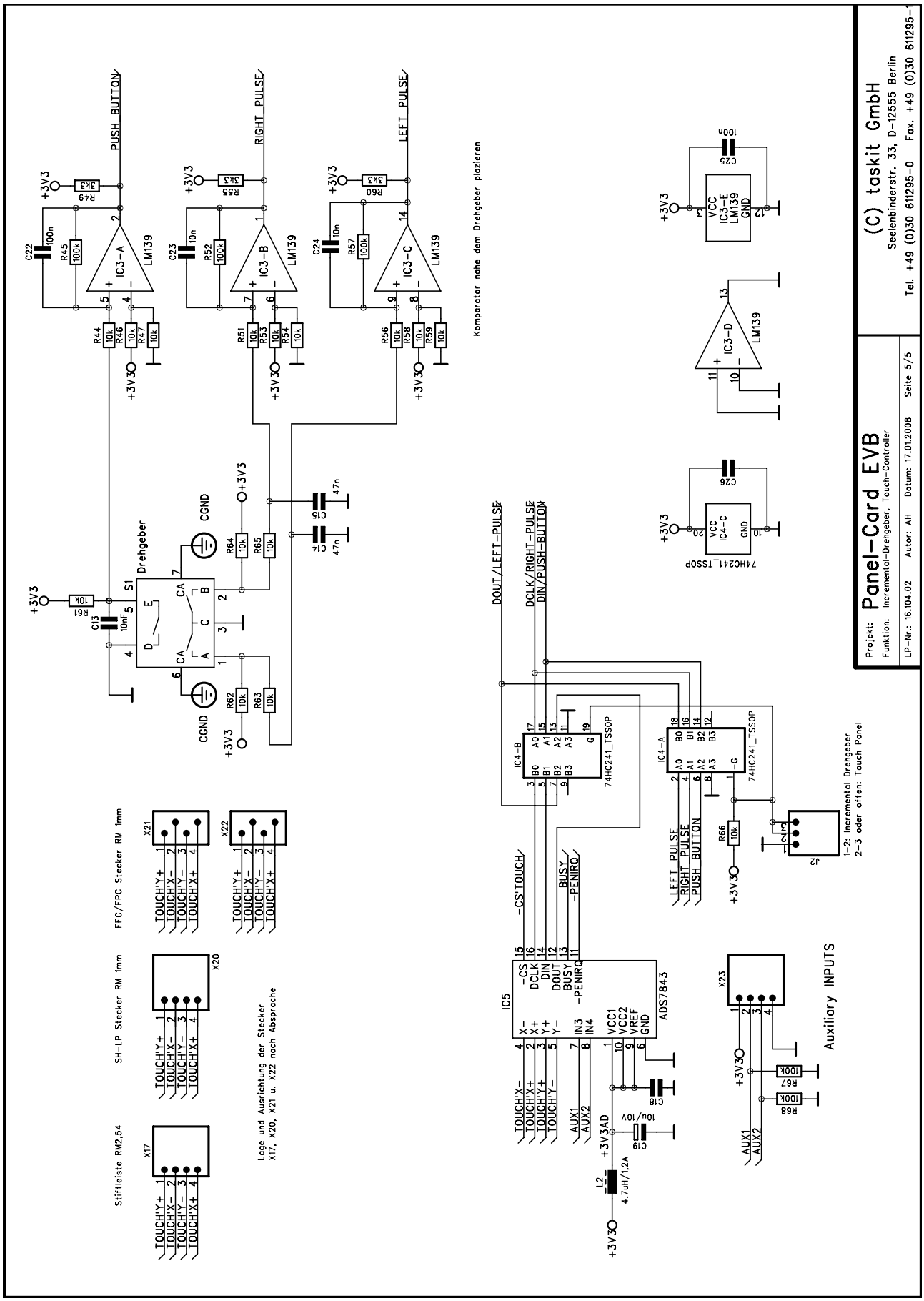

Figure 6.5. Panel-Card EVB Schematics HID

6.3. Stamp Adaptor 2

6.3.1. Description

The Stamp Adaptor 2 functions as interface board between Stamp9G20 and Panel-Card EVB. It may also be used for prototyping without the EVB. It has the connectors for Panel-Card EVB (X21, X22), a wrapfield for direct access to all pins of Stamp9G20 (X11, X12) and connectors for connecting the Stamp9G20. Additionally it implemts a Davicom® DM9161BIEP ethernet physical interface design, which may be used for customer designs as well.

6.3.2. Connector Pin Assignments

The pin assignments for the wrapfield are identical with the pins of Stamp9G20. The pin assignment for X21/X22 are below:

1	VCC			DSR0	ISI D2	PB22	2
3	PB25	ISI D5	RI0	RXD0		PB5	4
5	PB4		TXD0	DTR0	ISI D4	PB24	6
7	PB26	ISI D6	RTS0	CTS0	ISI D7	PB27	8
9	PB23	ISI D3	DCD0	GND			10
11	PB14		DRXD	DTXD		PB15	12
13	PA7		MCCDA	MCCK		PA8	14
15	PA6		MCDA0	MCDA1		PA9	16
17	PA10	ETX2	MCDA2	MCDA3	ETX3	PA11	18
19	VCC			GND			20
21	USB Host A-			USB Host B-			22
23	USB Host A+			USB Host B+			24
25	USB Device-			USB Device+			26
27	PA22	ETXER	ADTRG	/RESET			28
29	VCC			GND			30
31	PB17	TCLK4	TF0	RF0	ISI D1	PB21	32
33	PB16	TCLK3	TK0	RK0	ISI D0	PB20	34
35	PB18	TIOB4	TD0	RD0	TIOB5	PB19	36
37	BMS			WKUP			38
39	SHDN			VBATT			40

Table 6.1. Pin Assignment and Multiplexing X21

1	VCC			SCK2	RXD4	PA30	2
3	PA31	TXD4	SCK0	RXD2		PB9	4
5	PB8		TXD2	TXD1	TCLK1	PB6	6
7	PA4	MCDB2	RTS2	CTS2	MCDB1	PA5	8
9	PB7	TCLK2	RXD1	GND			10
11	PA23	ETX2	TWD	TWCK	ETX3	PA24	12
13	VCC			GND			14
15	PB0	TIOA3	SPI1MISO	SPI1MOSI	TIOB3	PB1	16
17	PB2	TIOA4	SPI1CLK	SPI1CS1	TIOA5	PB3	18
19	PA0	MCDB0	SPI1CS2		SPI1CS3	PC3	20
21							22
23							24
25							26
27	VCC			GND			28
29	TDI			TMS			30
31	TDO			TCK			32
33	ELED_L			ELED_S			34
35	ETX+			ETX-			36
37	ERX+			ERX-			38
39	POE1			POE2			40

Table 6.2. Pin Assignment and Multiplexing X22

6.3.3. Stamp Adaptor 2 Dimensions

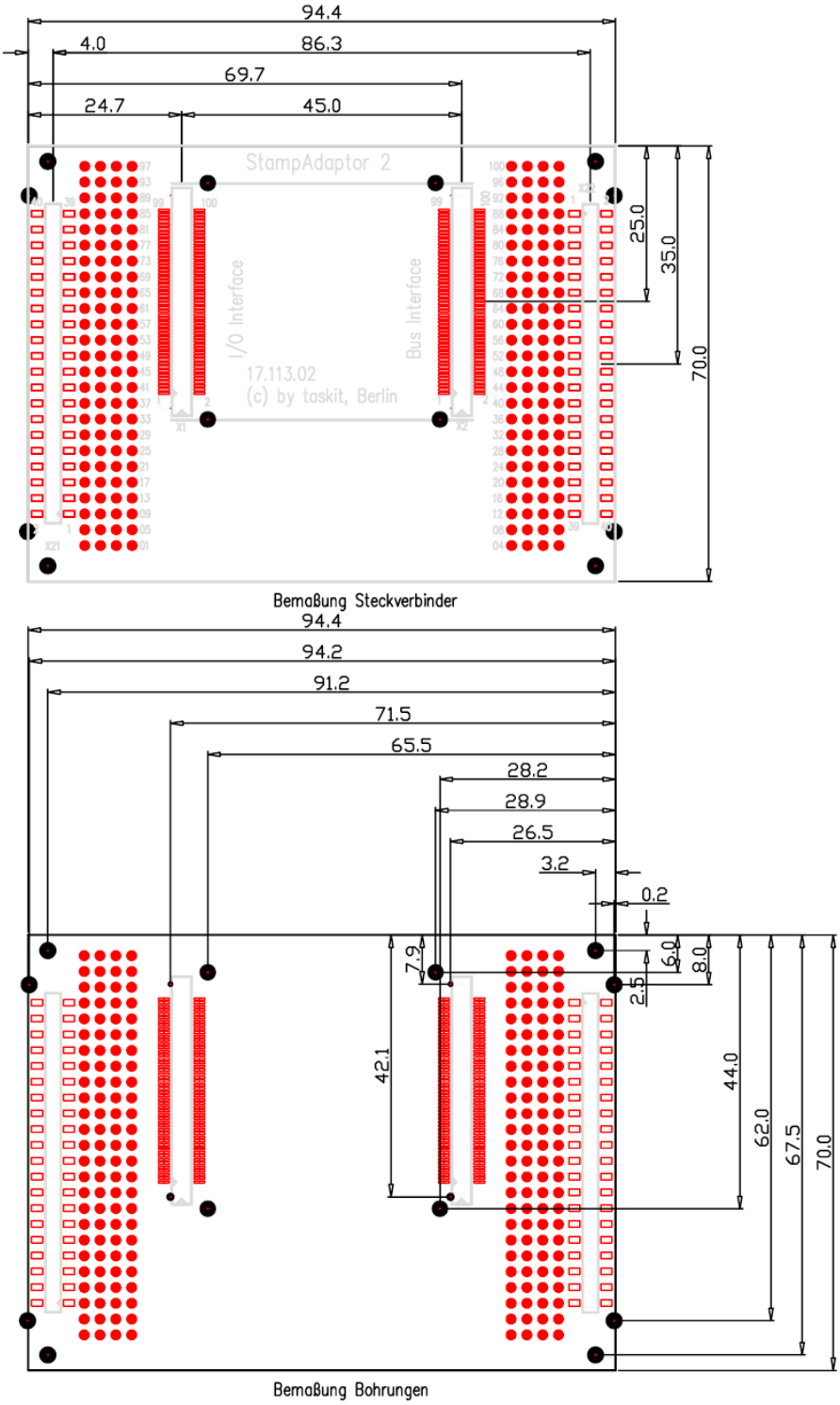

Figure 6.6. Stamp Adaptor 2 Dimensions

6.3.4. Stamp Adaptor 2 Schematics

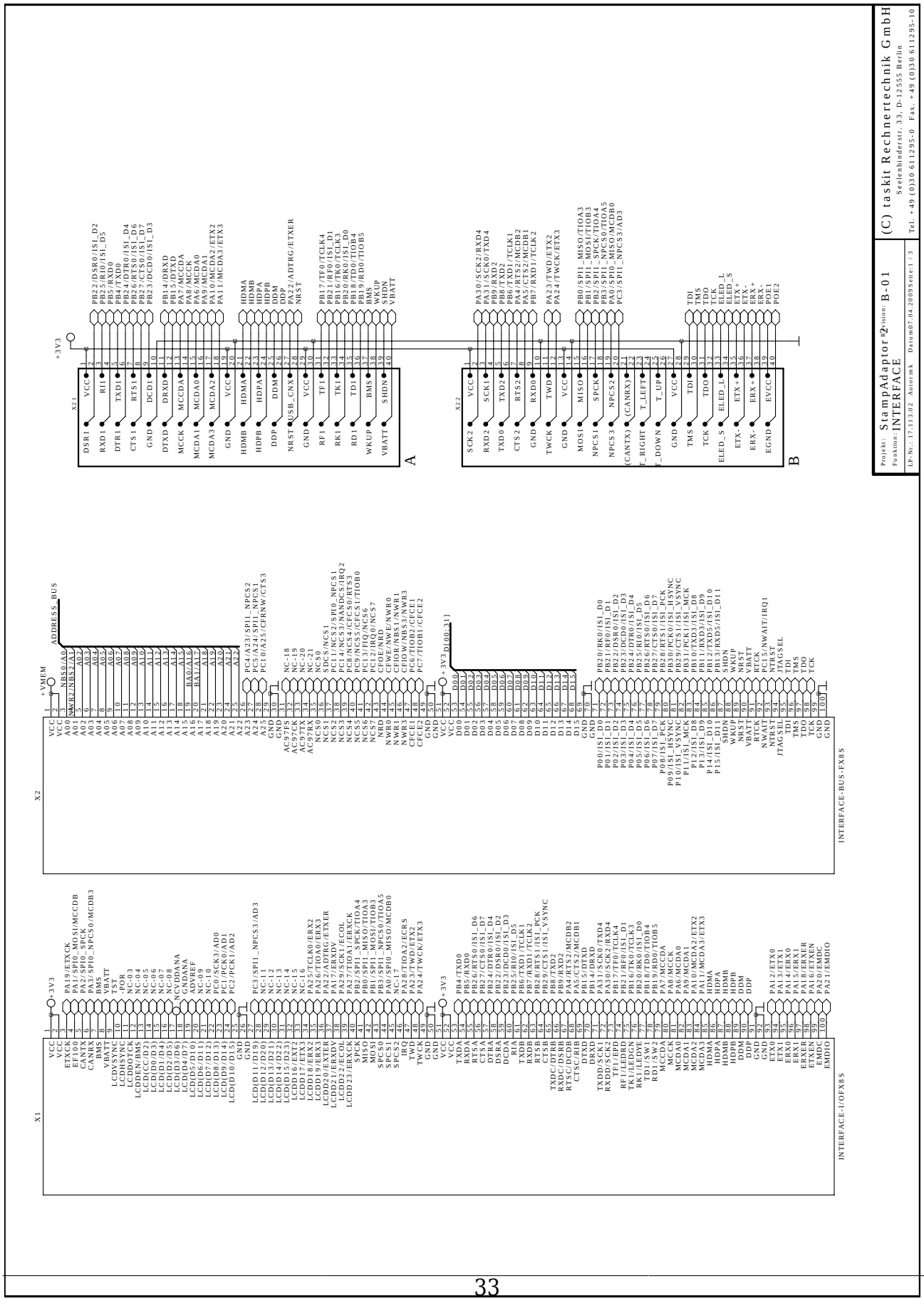

Figure 6.7. Stamp Adaptor 2 Bus Interface

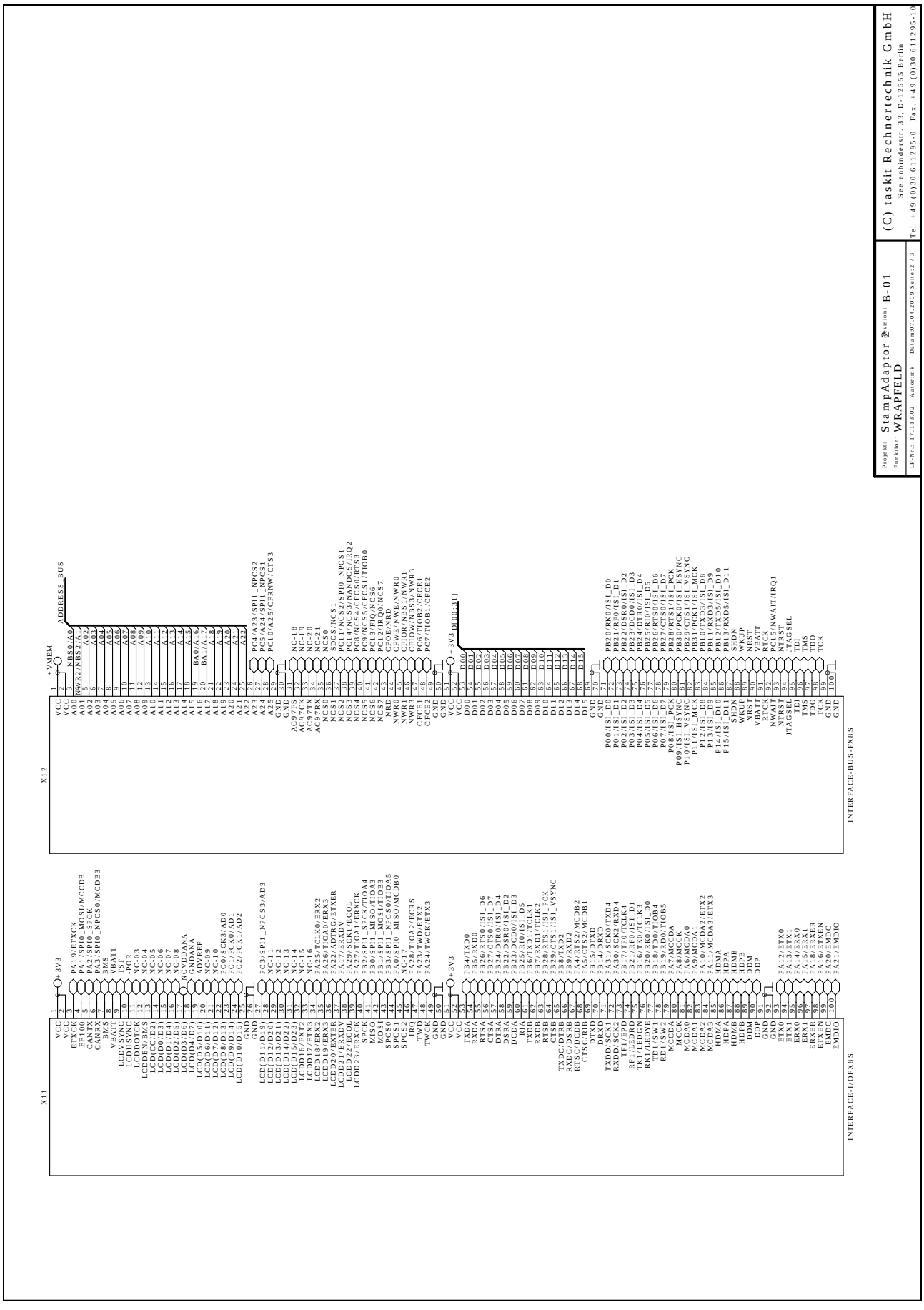

Figure 6.8. Stamp Adaptor 2 Wrapfield

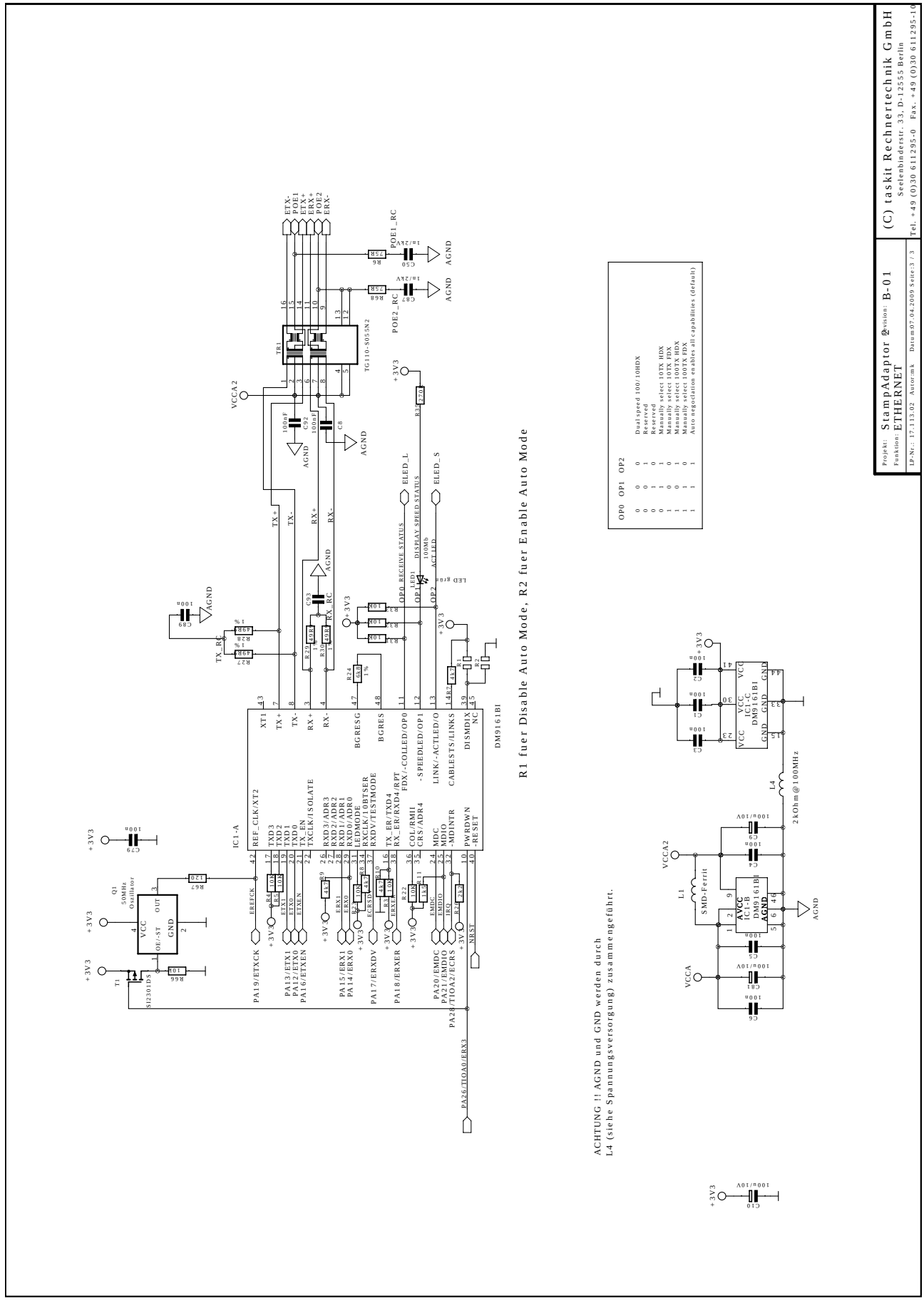

Figure 6.9. Stamp Adaptor 2 Ethernet

Appendix A. Peripheral Color Codes

This table matches the color used to identify various peripherals in tables.

Power Supply/Ground
USART
Debug UART
TWI (I^2C-Bus)
SD-Card/MMC
SPI
USB Host
USB Device
Reserved
Synhcronous Serial Controller (SSC)
JTAG
Control
Ethernet
Genral Purpose I/O Port
Programmable Clock Output
Analog-to-digital Converter
Timer Counter
Image Sensor Interface
LCD/TFT Controller Interface
Embedded Trace Macrocell
Static Memory Controller
Compact Flash Interface
Pulse Width Modulator
Touch Controller
AC97 Sound Interface

Appendix B. Peripheral Identifiers

ID	Mnemonic	Peripheral Name	External Interrupt
0	AIC	Advanced Interrupt Controller	FIQ
1	SYSC	System Controller Interrupt	
2	PIOA	Parallel I/O Controller A	
3	PIOB	Parallel I/O Controller B	
4	PIOC	Parallel I/O Controller C	
5	ADC	Analog to Digital Converter	
6	US0	USART 0	
7	US1	USART 1	
8	US2	USART 2	
9	MCI	Multimedia Card Interface	
10	UDP	USB Device Port	
11	TWI	Two-Wire Interface	
12	SPI0	Serial Peripheral Interface 0	
13	SPI1	Serial Peripheral Interface 1	
14	SSC	Synchronous Serial Controller	
15-16	-	Reserved	
17	TC0	Timer/Counter 0	
18	TC1	Timer/Counter 1	
19	TC2	Timer/Counter 2	
20	UHP	USB Host Port	
21	EMAC	Ethernet MAC	
22	ISI	Image Sensor Interface	
23	US3	USART 3	
24	US4	USART 4	
25	US5	USART 5	
26	TC3	Timer/Counter 3	
27	TC4	Timer/Counter 4	
28	TC5	Timer/Counter 5	
29	AIC	Advanced Interrupt Controller	IRQ0
30	AIC	Advanced Interrupt Controller	IRQ1
31	AIC	Advanced Interrupt Controller	IRQ2

Table B.1. Peripheral Identifiers

Appendix C. Address Map (Physical Address Space)

After the execution of the remap command the 4 GB physical address space is separated as shown in the following table. Accessing these addresses directly is only possible if the MMU (memory management unit) is deactivated. As soon as the MMU is activated the visible address space is changed completely. If absolute memory addresses should be accessed within an application, the corresponding address space has first to be mapped to the virtual address space using mmap or ioremap under Linux.

Address (Hex)	Mnemonic	Function
00 0000	Boot Memory	NCS0 or internal ROM or internal SRAM (depending on BMS and REMAP)
10 0000	ROM	Internal ROM 32 kByte
20 0000	SRAM0	Internal SRAM 16 kByte
30 0000	SRAM1	Internal SRAM 16 kByte
50 0000	UHP	USB Host Port
1000 0000	EBI NCS0	Chip Select 0
2000 0000	EBI NCS1	Chip Select 1: SDRAM
3000 0000	EBI NCS2	Chip Select 2
4000 0000	EBI NCS3	Chip Select 3: NAND
5000 0000	EBI NCS4	Chip Select 4
6000 0000	EBI NCS5	Chip Select 5
7000 0000	EBI NCS6	Chip Select 6
8000 0000	EBI NCS7	Chip Select 7
FFFA 0000	TC0, TC1, TC2	3 Timer Counter, 16-Bit
FFFA 4000	UDP	USB Device Port
FFFA 8000	MCI	Multimedia Card / SD-Card Interface
FFFA C000	TWI	Two Wire Interface (I^2C)
FFFB 0000	USART0	Synchronous or Asynchronous Serial Port #0
FFFB 4000	USART1	Synchronous or Asynchronous Serial Port #1
FFFB 8000	USART2	Synchronous or Asynchronous Serial Port #2
FFFB C000	SSC	Serial Synchronous Controller (I^2S)
FFFC 0000	ISI	Image Sensor Interface
FFFC 4000	EMAC	Ethernet Controller
FFFC 8000	SPI0	Serial Peripheral Interface #0
FFFC C000	SPI1	Serial Peripheral Interface #1
FFFD 0000	USART3	Synchronous or Asynchronous Serial Port #3
FFFD 4000	USART4	Synchronous or Asynchronous Serial Port #4
FFFD 8000	USART5	Synchronous or Asynchronous Serial Port #5
FFFD C000	TC3, TC4, TC5	3 Timer Counter, 16-Bit
FFFE 0000	ADC	Analog Digital Converter
FFFF E800	ECC	Error Correction Controller

Address (Hex)	Mnemonic	Function
FFFF EA00	SDRAMC	SDRAM Controller
FFFF EC00	SMC	Static Memory Controller
FFFF EE00	MATRIX	Bus Matrix User Interface
FFFF F000	AIC	Advanced Interrupt Controller
FFFF F200	DBGU	Debug Unit, including UART
FFFF F400	PIOA	32 Bit Parallel I/O Controller A
FFFF F600	PIOB	32 Bit Parallel I/O Controller B
FFFF F800	PIOC	32 Bit Parallel I/O Controller C
FFFF FC00	PMC	Power Management Controller
FFFF FD00	RSTC	Reset Controller, Battery Powered
FFFF FD10	SHDWC	Shutdown Controller, Battery Powered
FFFF FD20	RTT	Real-time Timer 32 Bit, Battery Powered
FFFF FD30	PIT	Periodic Interval Timer 32 Bit
FFFF FD40	WDT	Watchdog Timer
FFFF FD50	GPBR	4 General Purpose Backup Registers

Table C.1. Physical Address Space

Appendix D. Stamp9G20 Pin Assignment

Pin	GPIO	Periph. A	Periph. B	Add. Function		Add. Function	Periph. B	Periph. A	GPIO	Pin
1	VMEM								VMEM	2
3	**A0/NBS0**								**A1/NBS2/NWR2**	4
5	**A2**								**A3**	6
7	**A4**								**A5**	8
9	**A6**								**A7**	10
11	**A8**								**A9**	12
13	**A10**								**A11**	14
15	**A12**								**A13**	16
17	**A14**								**A15**	18
19	**A16/BA0**								**A17/BA1**	20
21	A18								A19	22
23	A20								**A21/NANDCLE**	24
25	**A22/NANDALE**						SPI1 NPCS2	A23	PC4	26
27	PC5	A24	SPI1 NPCS1				CTS3	A25/ CFRNW	PC10	28
29	GND								GND	30
31	Reserved								Reserved	32
33	Reserved								Reserved	34
35	NCS0								**NCS1/SDCS**	36
37	PC11	NCS2	**SPI0 NPCS1**				IRQ2	**NCS3/ NANDCS**	PC14	38
39	PC8	NCS4/ CFCS0	RTS3				TIOB0	NCS5/ CFCS1	PC9	40
41	**PC13**	FIQ	NCS6				NCS7	IRQ0	PC12	42
43	NRD/CFOE								NWR0/NWE/CFWE	44
45	**NWR1/NBS1/CFIOR**								NWR3/NBS3/CFIOW	46
47	PC6	TIOB2	CFCE1				CFCE2	TIOB1	PC7	48
49	GND								GND	50
51	VCC								VCC	52
53	**D0**								**D1**	54
55	**D2**								**D3**	56
57	**D4**								**D5**	58
59	**D6**								**D7**	60
61	**D8**								**D9**	62
63	**D10**								**D11**	64
65	**D12**								**D13**	66
67	**D14**								**D15**	68
69	GND								GND	70
71	PB20	RK0	ISI D0				ISI D1	RF0	PB21	72

Pin	GPIO	Periph. A	Periph. B	Add. Function		Add. Function	Periph. B	Periph. A	GPIO	Pin
73	PB22	DSR0	ISI D2				ISI D3	DCD0	PB23	74
75	PB24	DTR0	ISI D4				ISI D5	RI0	PB25	76
77	PB26	RTS0	ISI D6				ISI D7	CTS0	PB27	78
79	PB28	RTS1	ISI PCK				ISI HSYNC	PCK0	PB30	80
81	PB29	CTS1	ISI VSYNC				ISI MCK	PCK1	PB31	82
83	PB10	TXD3	ISI D8				ISI D9	RXD3	PB11	84
85	PB12	TXD5	ISI D10				ISI D11	RXD5	PB13	86
87	SHDN					WKUP				88
89	NRST					Reserved				90
91	RTCK						IRQ1	NWAIT	PC15	92
93	NTRST					JTAGSEL				94
95	TDI					TMS				96
97	TDO					TCK				98
99	GND					GND				100

Table D.1. Pin Assignment Bus Interface

Pin	GPIO	Periph. A	Periph. B	Add. Function		Add. Function	Periph. B	Periph. A	GPIO	Pin
1	VCC					VCC				2
3	PA19	ETXCK					MCCDB	**SPI0 MOSI**	PA1	4
5	PA2	**SPI0 SPCK**					MCDB3	SPI0 NPCS0	PA3	6
7	BMS					VBATT				8
9	TST					-POR				10
11	Reserved					Reserved				12
13	Reserved					Reserved				14
15	Reserved					Reserved				16
17	VDDANA					GNDANA				18
19	ADVREF					Reserved				20
21	Reserved					AD0	SCK3		PC0	22
23	PC1		PCK0	AD1		AD2	PCK1		PC2	24
25	GND					GND				26
27	PC3		SPI1 NPCS3	AD3		Reserved				28
29	Reserved					Reserved				30
31	Reserved					Reserved				32
33	Reserved						ERX2	TCLK0	PA25	34
35	PA26	TIOA0	ERX3				ETXER	ADTRG	PA22	36
37	PA17	ERXDV					ECOL	SCK1	PA29	38

Pin	GPIO	Periph. A	Periph. B	Add. Function		Add. Function	Periph. B	Periph. A	GPIO	Pin
39	PA27	TIOA1	ERXCK				TIOA4	SPI1 SPCK	PB2	40
41	PB0	SPI1 MISO	TIOA3				TIOB3	SPI1 MOSI	PB1	42
43	PB3	SPI1 NPCS0	TIOA5				MCDB0	SPI0 MISO	PA0	44
45	Reserved						ECRS	TIOA2	PA28	46
47	PA23	TWD	ETX2				ETX3	TWCK	PA24	48
49	GND					GND				50
51	VCC					VCC				52
53	PB4	TXD0						RXD0	PB5	54
55	PB26	RTS0	ISI D6				ISI D7	CTS0	PB27	56
57	PB24	DTR0	ISI D4				ISI D2	DSR0	PB22	58
59	PB23	DCD0	ISI D3				ISI D5	RI0	PB25	60
61	PB6	TXD1	TCLK1				TCLK2	RXD1	PB7	62
63	PB28	RTS1	ISI PCK				ISI VSYNC	CTS1	PB29	64
65	PB8	TXD2						RXD2	PB9	66
67	PA4	RTS2	MCDB2				MCDB1	CTS2	PA5	68
69	PB15	DTXD						DRXD	PB14	70
71	PA31	SCK0	TXD4				RXD4	SCK2	PA30	72
73	PB17	TF0	TCLK4				ISI D1	RF0	PB21	74
75	PB16	TK0	TCLK3				ISI D0	RK0	PB20	76
77	PB18	TD0	TIOB4				TIOB5	RD0	PB19	78
79	PA7	**MCCDA**						**MCCK**	PA8	80
81	PA6	**MCDA0**						**MCDA1**	PA9	82
83	PA10	**MCDA2**	ETX2				ETX3	**MCDA3**	PA11	84
85	HDMA					HDPA				86
87	HDMB					HDPB				88
89	DDM					DDP				90
91	GND					GND				92
93	PA12	ETX0						ETX1	PA13	94
95	PA14	ERX0						ERX1	PA15	96
97	PA18	ERXER						ETXEN	PA16	98
99	PA20	EMDC						EMDIO	PA21	100

Table D.2. Pin Assignment IO Interface

Appendix E. Stamp9G20 Electrical Characteristics

Ambient temperature 25°C, unless otherwise indicated

Symbol	Description	Parameter	Min.	Typ.	Max	Unit
V_{CC}	Operating Voltage		3.0	3.3	3.6	V
V_{MEM}	Memory Bus Voltage		1.65	1.8	1.95	V
V_{RES}	Reset Treshhold			2.9		V
T_{RES}	Duration of Reset Pulse		150		280	ms
V_{IH}	High-Level Input Voltage	3.3V	2.0		V_{CC} + 0.3	V
		(PIOC4 - PIOC31) 1.8V	1.26		2.1	V
V_{IL}	Low-Level Input Voltage	3.3V	-0.3		0.8	V
		(PIOC4 - PIOC31) 1.8V	-0.3		0.54	V
P	Normal Operation			120		mW
	Full Load	max.		200		mW
	Stand-By			80		mW
	Power-Down			35		mW
	Full Load with Ethernet			480		mW
V_{BATT}	Battery Voltage		2.0	3.0	V_{CC}	V
I_{BATT}	Battery Current	Ambient temp. = 25°C		5		µA
		Ambient temp. = 70°C			17	µA
		Ambient temp. = 85°C			22	µA

Table E.1. Electrical Characteristics

Appendix F. Stamp9G20 Clock Characteristics

Symbol	Description	Dependency	Tolerance	Typical Value	Unit
MAINCK	Main Oscillator frequency			18.432	MHz
SLCK	Slow Clock			32.768	KHz
PLLACK	PLLA Clock	MAINCK		792.000	MHz
PCK	Processor Clock	PLLACK		396.000	MHz
MCK	Master Clock	PCK		132.000	MHz
SDCK	SDRAM Clock	MCK		132.000	MHz
BCK	Baudrate Clock	MCK	1.5%	8.25(max)	MHz
PLLBCK	PLLB Clock	MAINCK		96.000	MHz
USBCK	USB Clock	PLLBCK	0.25%	48.000	MHz

Table F.1. Clock Characteristics

Appendix G. Stamp9G20 Environmental Ratings

Symbol	Description	Parameter	Operating		Storage		Unit
			Min.	Max.	Min.	Max.	
T_A	Ambient temperature		-30	85	-45	85	℃
	Relative Humidity	no condensation		90		90	%RH
	Absolute Humidity		<= Humidity@T_A = 60℃, 90%RH				
	Corrosive Gas		not admissible				

Table G.1. Environmental Ratings

Appendix H. Stamp9G20 Dimensions

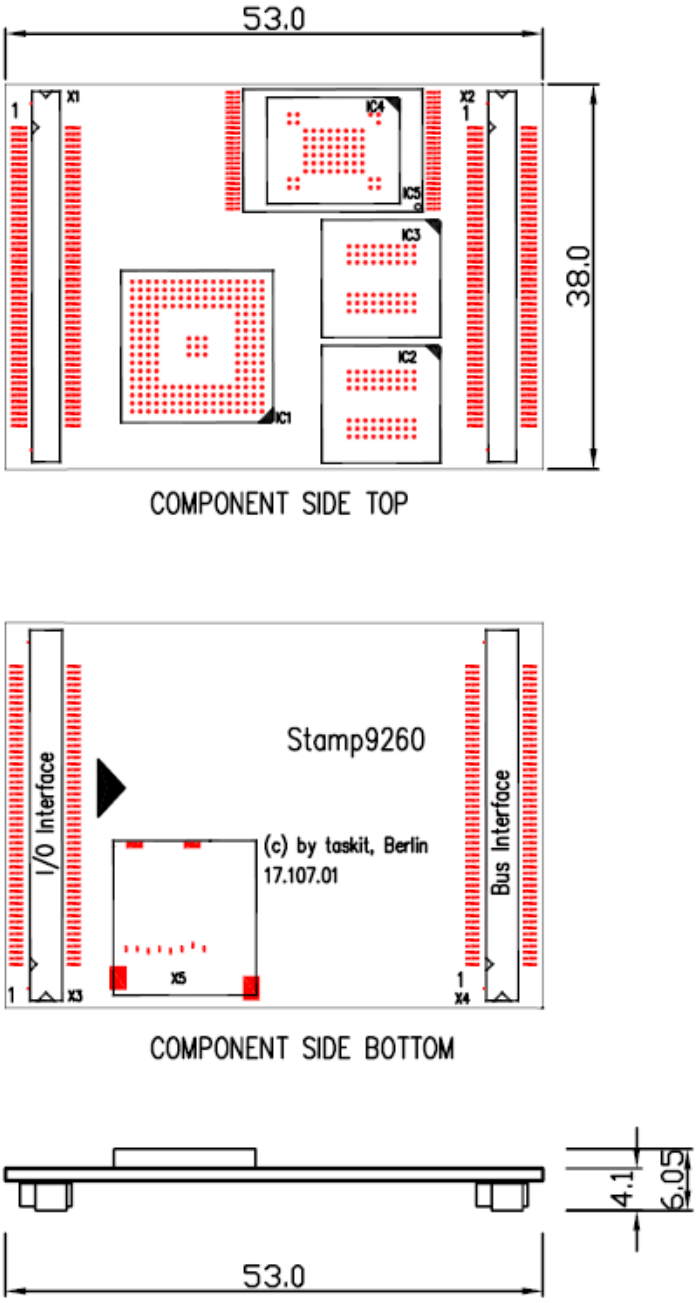

Figure H.1. Stamp9G20 Dimensions