SYS TEC ELECTRONIC

ECUcore-9G20

Hardware Manual

PCB Version: 4258.0

Edition March 2010

Status / Changes

Status: released

Date/ Version	Section	Change	Editor
L-1255e_01		initial version	K.Otto

In this manual are descriptions for copyrighted products, which are not explicitly indicated as such. The absence of the trademark (©) symbol does not infer that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, SYS TEC electronic GmbH assumes no responsibility for any inaccuracies. SYS TEC electronic GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. SYS TEC electronic GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages, which might result.

Additionally, SYS TEC electronic GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. SYS TEC electronic GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

	WORLDWIDE
Address:	SYS TEC electronic GmbH August-Bebel-Str. 29 D-07973 Greiz GERMANY
Ordering Information:	+49 (3661) 6279-0 info@systec-electronic.com
Technical Support:	+49 (3661) 6279-0 support@systec-electronic.com
Fax:	+49 (3661) 62 79 99
Website:	http://www.systec-electronic.com

1st Edition March 2010

Index of figures and tables

1 Introduction

This manual describes the function and technical data for the ECUcore-9G20, but not for the microcontroller Atmel AT91SAM9G20 or any other supplemental products. Please refer to the corresponding manuals and documentation for any other products you may use.

Low-active signals are denoted by a „/“ in front of the signal name (i.e. “/RD”). The representation “0” indicates a logical-zero or low-level signal. A “1” is the synonym for a logical one or high-level signal.

2 Ordering Information and Support

Order Number	Version
4001016	ECUcore-9G20
KIT-164	Development Kit ECUcore-9G20

The ECUcore-9G20 standard version features:

- Atmel AT91SAM9G20 MCU with 400 MHz
- 16MiB FLASH
- 32MiB SDRAM
- FPGA with 6000 LUTs and Configuration-EEPROM
- Ethernet PHY
- CAN controller
- Real time clock
- Reset/Watchdog IC
- Temperature sensor
- Single power supply 3,3V (all other voltages are derived onboard)
- ESD Handling Instructions (printed version)

The Development Kit includes the ECUcore module and a Development Board which allows for rapid application development.

The Development Board features:

- Power supply connector 9-24VDC and an external power supply
- Dipswitch (8pos.), some Keys and LEDs to test IO functionality
- Potentiometer for using the analog input
- Connectors for RS232, CAN, USB and Ethernet interfaces
- EEPROM to test SPI
- microSD card socket
- Boot and Reset button
- Battery for buffering RTC
- ICE and JTAG interfaces for CPU and FPGA
- Expansion connector for developing user periphery

Software Support:

- Integrated Development Environment with complete GNU toolchain for ARM architecture
- uboot bootloader (pre-installed)
- Linux
- plc-firmware (PLCcore-9G20)
- CANopen protocol stack (limited function, obj-code library)
- CANopen Configuration Suite (Evaluation Version)
- CAN-Report CAN-bus monitor (Evaluation Version)
- OD-Builder

3 Properties of the ECUcore-9G20

3.1 Overview

The ECUcore-9G20 belongs to SYS TEC's ECUcore family. The ECUcore-9G20 combines all elements of a microcontroller system on one board. Thanks to most modern SMD packages and the multilayer design, the module was designed at minimum size.

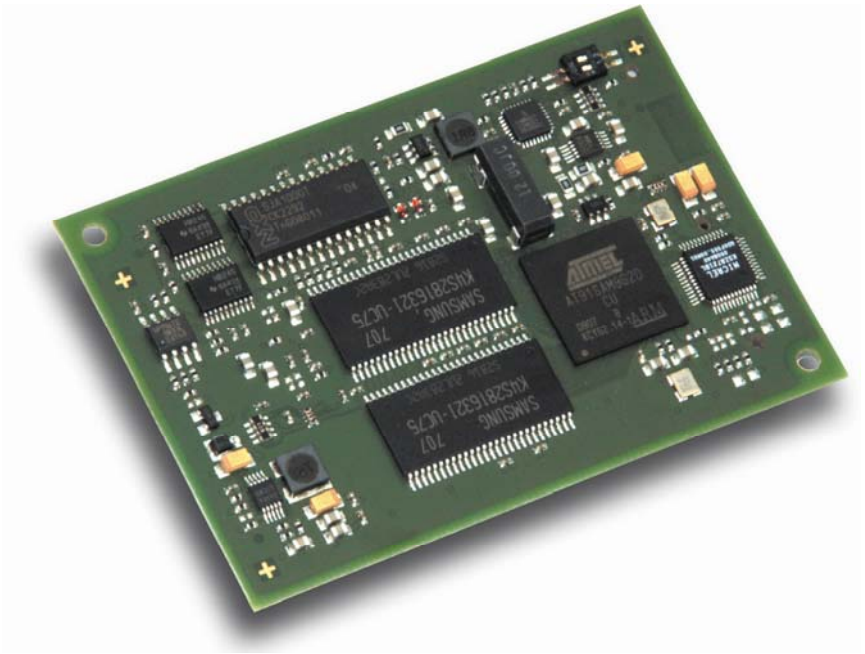


Figure 1: ECUcore-9G20

The dimensions of the board are 78mm x 54mm and with two included board connectors it is multifunctional in embedded systems.

The ECUcore-9G20 features an Atmel AT91SAM9G20 microcontroller. It is a highly-integrated 32-bit microprocessor based on the ARM926EJ-S architecture.

The interconnection to a baseboard is possible through a pair of low-density (2mm pitch) connectors with 200 pins in total.

The ECUcore-9G20 brings about the following features:

- Internal features of the Atmel AT91SAM9G20:
    - Internal 400MHz CPU-clock from external 3-20MHz
    - 32kiB Data- / 32kiB instruction cache, MMU
    - 64k ROM, 32k RAM internal with single cycle access
    - External bus interface for
      SDRAM, Static memory, NAND- and Compact Flash
    - USB2.0 full speed 1x device, 2x host
    - 1 Fast Ethernet controller MAC
    - Six 32-bit-layer bus matrix
    - Peripheral DMA controller channels
    - System controller with Reset, Shutdown, Battery backup, Clock generator, Power management, Watchdog and Realtime timer
    - 4x USART, 2x UARTs
    - 2 Master/Slave SPI with 4 chip selects
    - 1 I2C interface up to 400kbits
    - 2x 3channel 16bit Timer/Counter
    - 1x 4channel ADC 10bit
    - 1x two-slot SDcard/MMC interface
    - 1x SSC
    - EmbeddedICE interface
    - JTAG Boundary Scan on all digital pins
    - 217-pin LFBGA package

- Memory configuration:
    - default 16MiB 16bit NOR-Flash (4-64MiB)
    - default 32MiB 32bit SDRAM (16-128MiB)
- Communication features:
    - 1 Ethernet interface
    - 3 USARTs and 2 UARTs as LVTTL
    - 1 CAN as TTL
    - 1 USB device full speed
    - 2 USB host full speed
    - 2 SDcard/MMC
    - 1 SSC
    - SPI with 3 chip select
    - I²C interface
- Other board-level features:
    - FPGA with default 6000 (up to 21000 LUTs) with own configuration device
    - Ethernet PHY
    - CAN controller
    - Power sequencer with Reset- and Watchdog function
    - Power failure recognition
    - Battery-buffered Real time clock
    - Temperature sensor
    - JTAG/Embedded ICE interface
    - 18,432MHz main crystal, 32,768kHz low speed crystal
    - 3.3V operating voltage
      onboard generated voltage for CPU and FPGA core and CAN controller
    - Industrial temperature range (-40°C to +85°C)
    - RoHS compliant

3.2 Block Diagram

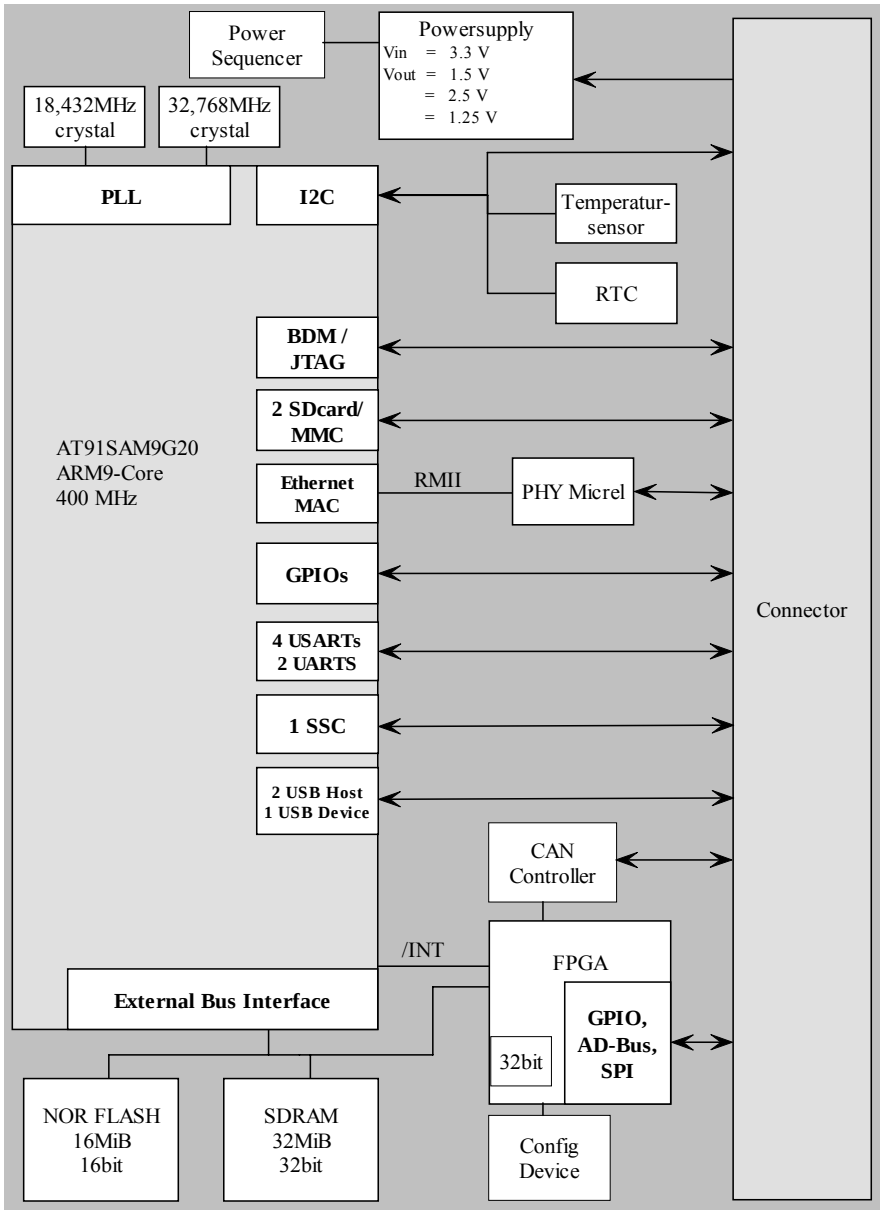


Figure 2: Block Diagram ECUcore-9G20

4 Component Descriptions

The functions of the on-board components are explained in the following sections.

4.1 Connectors

The ECUcore-9G20 has two board connectors. Each of the SMT socket strips consists of 100 contacts divided into double rows. In total the module has 200 contacts. For better emc-properties, 20% of the pins are GND.
A third connector at the front side is for connecting debug interfaces of the CPU, Power sequencer and FPGA. It is not mounted by default.

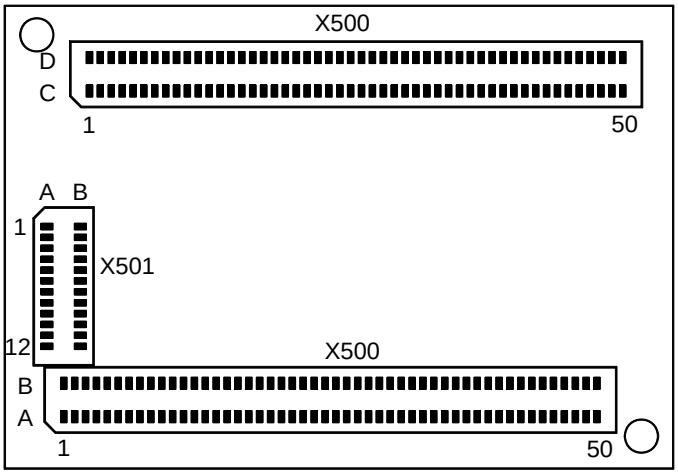


Figure 3: Pinout (top view)

The picture shows the module from top view which means that you look from the top through the module. The connectors are placed accordingly to the ones on the baseboard.

The board connectors are equipped with the common and durable 1,27mm pitch. The type of socket stripes used on the ECUcore-9G20 is '6060'-series provided by "W+P PRODUCTS" with a hight of 3,6mm.

The series matches (e.g.) with "W+P PRODUCTS" strip line series '7072' or '7073'. Please refer to the datasheet and their electrical specifications.
Connectors:
Module: W+P 6060-100-36-00-00-00-PPST (2x50pol. socked))
W+P 6060-024-36-00-00-00-PPST (2x12pol.)
Baseboard: W+P 7072-100-10-00-10-PPST (2x50pol. header)
W+P 7072-024-10-00-10-PPST (2x12pol.)

The following table defines the pinout.

Signal	Pin	Pin	Signal		Signal	Pin	Pin	Signal
GND	A01	B01	GND		GND	C01	D01	+2V5_EPHY
/BOOT	A02	B02	/MR		ETH0_TX-	C02	D02	GND
WKUP	A03	B03	/RESET		ETH0_TX+	C03	D03	ETH_SPEED
SHDN	A04	B04	/PFI		ETH0_RX+	C04	D04	ETH_LINK/ACT
BMS	A05	B05	WDI		ETH0_RX-	C05	D05	GND
GND	A06	B06	PS_IO		GND	C06	D06	AD0
DRXD	A07	B07	GND		ADTRG	C07	D07	AD1
DTXD	A08	B08	RTS0		ADVREF	C08	D08	AD2
DSR0	A09	B09	CTS0		GND	C09	D09	GND
DTR0	A10	B10	RTS1		SD_MCDA0	C10	D10	SD_MCDB0
DCD0	A11	B11	CTS1		SD_MCDA1	C11	D11	SD_MCDB1
GND	A12	B12	GND		SD_MCDA2	C12	D12	SD_MCDB2
TXD0	A13	B13	TXD1		SD_MCDA3	C13	D13	SD_MCDB3
RXD0	A14	B14	RXD1		SD_MCCK	C14	D14	SD_MCCDA
TXD2	A15	B15	TXD3		GND	C15	D15	SD_MCCDB
RXD2	A16	B16	RXD3		SCK0	C16	D16	GND
GND	A17	B17	TXD5		SCK1	C17	D17	TIOA1
USB_HDPA	A18	B18	RXD5		SCK2	C18	D18	TIOB1
USB_HDMA	A19	B19	GND		PCK1	C19	D19	TIOA2
USB_HDPB	A20	B20	USB_DDP		RK0	C20	D20	TIOB2
USB_HDMB	A21	B21	USB_DDM		TK0	C21	D21	TD0
GND	A22	B22	GND		RF0	C22	D22	RD0
I2C_DATA	A23	B23	CAN_TXD		TF0	C23	D23	GND
I2C_CLK	A24	B24	CAN_RXD		GND	C24	D24	FPGA_IO0
GND	A25	B25	CAN_VCC		FPGA_IO1	C25	D25	FPGA_IO2
FPGA_IO44	A26	B26	GND		FPGA_IO3	C26	D26	FPGA_IO4
FPGA_IO46	A27	B27	FPGA_IO45		FPGA_IO5	C27	D27	FPGA_IO6
FPGA_IO48	A28	B28	FPGA_IO47		FPGA_IO7	C28	D28	GND
FPGA_IO50	A29	B29	FPGA_IO49		GND	C29	D29	FPGA_IO8
FPGA_IO52	A30	B30	FPGA_IO51		FPGA_IO9	C30	D30	FPGA_IO10

Signal	Pin	Pin	Signal		Signal	Pin	Pin	Signal
GND	A31	B31	FPGA_IO53		FPGA_IO11	C31	D31	FPGA_IO12
FPGA_IO54	A32	B32	GND		FPGA_IO13	C32	D32	FPGA_IO14
FPGA_IO56	A33	B33	FPGA_IO55		FPGA_IO15	C33	D33	GND
FPGA_IO58	A34	B34	FPGA_IO57		FPGA_IO17	C34	D34	FPGA_IO16
FPGA_IO60	A35	B35	FPGA_IO59		GND	C35	D35	FPGA_IO18
FPGA_IO62	A36	B36	FPGA_IO61		FPGA_IO19	C36	D36	FPGA_IO20
GND	A37	B37	FPGA_IO63		FPGA_IO21	C37	D37	FPGA_IO22
FPGA_IO64	A38	B38	GND		FPGA_IO23	C38	D38	FPGA_IO24
FPGA_IO66	A39	B39	FPGA_IO65		FPGA_IO25	C39	D39	GND
FPGA_IO68	A40	B40	FPGA_IO67		FPGA_IO27	C40	D40	FPGA_IO26
FPGA_IO70	A41	B41	FPGA_IO69		GND	C41	D41	FPGA_IO28
FPGA_IO72	A42	B42	FPGA_IO71		FPGA_IO29	C42	D42	FPGA_IO30
GND	A43	B43	FPGA_IO73		FPGA_IO31	C43	D43	FPGA_IO32
FPGA_IO74	A44	B44	GND		FPGA_IO33	C44	D44	FPGA_IO34
FPGA_IO76	A45	B45	FPGA_IO75		FPGA_IO35	C45	D45	GND
FPGA_IO78	A46	B46	FPGA_IO77		FPGA_IO37	C46	D46	FPGA_IO36
FPGA_IO80	A47	B47	FPGA_IO79		GND	C47	D47	FPGA_IO38
VBAT	A48	B48	FPGA_IO81		FPGA_IO39	C48	D48	FPGA_IO40
GND	A49	B49	GND		FPGA_IO41	C49	D49	FPGA_IO42
+3V3	A50	B50	+3V3		FPGA_IO43	C50	D50	GND

Table 1: Pinout connector X500

Pin function:

Name	Function
/BOOT	for selecting software-boot-sequence going till uboot or linux image
/MR	manual reset input of module
/RESET	reset output signal of reset-ic and CPU
/PFI	Power fail input for watching external power supply
WDI	watchdog input
WKUP	wakeup-pin for leaving shutdown mode
SHDN	shutdown-output for signaling shutdown mode
BMS	boot mode select of CPU boot from NOR-Flash or USB
PS_IO	Power sequencer IO-Pin (not yet used)
DRXD, DTXD	Debug UART (LV-TTL-level)
TXD0, RXD0, DSR0, DTR0, RTS0, CTS0, DCD0, SCK0	USART 0 with handshake signals
TXD1,2,3,5; RXD1,2,3,5; SCK1,2	USART 1,2 and UART 3,5
USB_HDxx	2 channel USB-Host
USB_DDP, DDM	USB-Device
I2C_DATA, I2C_CLK	two wire interface
CAN_TXD, CAN_RXD, CAN_VCC	CAN (5V-TTL-Level) with 5V-supply for external CAN-Driver
VBAT	backup battery input (3,3V) for RTC
ETH0_TX-, TX+, RX-, RX+; +2V5_EPHY	Ethernet-interface with reference voltage
AD0, 1, 2; ADTRG, ADVREF	analog digital converter inputs and reference-voltage input (0..3,3V)
SD_MCxxx	2 channel SD-card / MM-Card interface
PCK1	programmable clock output
TIOxx	Timer Input/Output
RK0, TK0, RF0, TF0,RD0, TD0,	synchronous serial interface
FPGA_IOxx	general purpose FPGA-IO's (can used as AD-Bus, SPI, IO's, Timer, Counter or any other digital function)
+3V3	3,3V-supply (about 1A)
GND	Signal ground

Table 2: Signal description connector X500

Connector X501 (not mounted in series)

Signal	Pin	Pin	Signal
+3V3	1	2	/RESET
/JTAGSEL	3	4	GND
ARM_TDI	5	6	ARM_TDO
ARM_TCK	7	8	ARM_TMS
GND	9	10	ARM_/NRST
ARM_RTCK	11	12	DRXD
DTXD	13	14	GND
FPGA_TDI	15	16	FPGA_TDO
FPGA_TCK	17	18	FPGA_TMS
GND	19	20	PWR_TDI
PWR_TDI	21	22	PWR_TCK
PWR_TCK	23	24	GND

Table 3: Pinout connector X501

Pin function:
All JTAG pins for debugging and programming the CPU, FPGA and Power sequencer.

4.2 Jumper and Switch Configuration

The jumper selects the reference voltage for the internal ADC. By default, the jumper is configured for a functionality of the ADC without external components. Change positions only if you really need a higher accuracy or other reference level in your application! The following figure shows the positions of jumper and switch S1 which are placed on the top-side of the module.

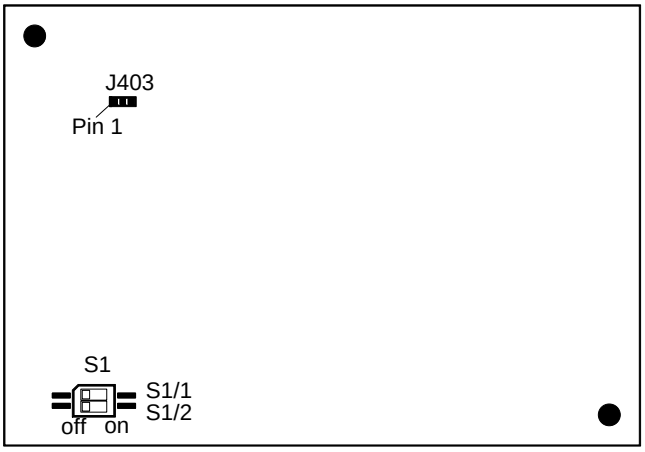


Figure 4: Jumper positions PCB 4258.0 (PLD)

The function of S1 depends on the application. Please refer to the software manual for further information.
The following table lists each solder jumper and its function on the ECUcore-9G20.

Jumper Switch	Jumper Pad Setting	Signal	Function
J403	1-2 (default)	ADVREF	ADVREF is connected to +3,3V supply
	2-3		ADVREF is connected to socked stripe X500/C8
S1/1	on	/BOOT	application defined default: software starts only till bootloader uboot
	off		software starts till linux
S1/2	on	/BMS	cpu starts from internal rom (atmel bootloader)
	off		cpu starts from external flash (uboot, Linux)

Table 4: Overview of the Solder Jumper and Switch

4.3 Power Supply

The ECUcore-9G20 must be supplied with an input voltage of +3.3VDC. The typical current consumption is approximately 400mA.

The 3,3V directly supplies:

- AT91SAM9G20 IO voltage
- FPGA IO voltage
- Flash, RAM
- RTC, Temperature sensor
- FPGA configuration device
- Power sequencer

So be careful and provide a good voltage with low tolerance and low ripple. See "Technical Data" for detailed information.

The onboard switching regulators generate all other needed voltage which is:

- 1V for SAM9G20 core voltage
- 1,2V for FPGA core supply
- 5V for CAN controller

The power up ramp for FPGA should be faster than 10ms.

4.4 Power Sequencer

Functions of the Power sequencer:

- watching all voltages and reset the module if a voltage is too low
- if the FPGA is equipped with decryption unit, the FPGA IO and core voltage must be switched according to the power up sequence required by Lattice
- watchdog to be used by the application
- analog power fail input and interrupt output
- manual reset input and stretched reset output

a) Watching voltages

Voltage	Min Level
3,3V	3,1V
1V	0,95V
1,2V	1,1V
5V	4,73V

b) Power up sequence for FPGA

Lattice recommends that the core supply of FPGA should reach its minimum earlier than the IO voltage at power up sequence. So the power sequencer first switches the core voltage and then the IO voltage.

c) Watchdog

The implemented watchdog has a timeout limit of 1,6s. It starts with the first triggering and the watchdog timer is reset by the rising edge of the trigger input. Then it should be retriggered in a faster cycle time than 1,6s. The only way to stop the watchdog is the manual reset or re-power-up of the system.

The Trigger-Pin 'WDI' is connected to the socket stripe X500/B5 with 10k pulldown to GND. Any ECUcore pin can be connected externally to WDI. On the baseboard, the ECUcore pin is connected to the FPGA pin FPGA_IO3 at X500/C26.

To control the FPGA pins, the default configuration maps all FPGA pins to the FPGA-register.

d) Power fail

The power fail input of the Power sequencer is connected to the socket stripe pin X500/B04.

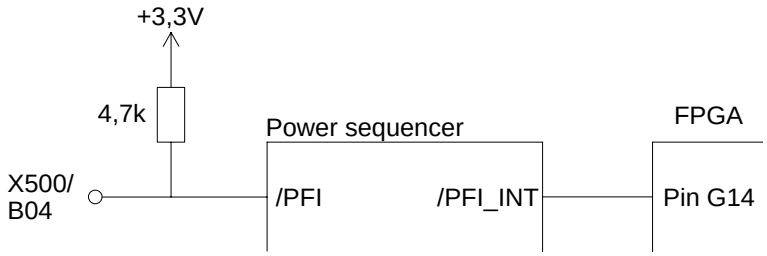


Figure 5: /PFI circuit

If the voltage at /PFI falls below 3,095V, the /PFI_INT switches to low.

With the 4,7k Resistor, an external digital signal can be used. This can be the power good output (e.g. open collector output) of a power supply.

The /PFI voltage can be analog (range 0..3,3V) too. But in level-calculation note the 4,7k to 3,3V!

e) Manual Reset

The manual reset (/MR) is connected to X500/B02 with 10k pullup to 3,3V. A reset occurs if the manual reset is switched to GND.

If a reset occurs, the core voltage supply is switched off for 50ms. This implies that the AT91SAM9G20 sets its reset out pin (/Reset) to low and also resets all other onboard components (FPGA, Ethernet PHY, Flash, CAN controller). The reset out pin is connected to X500/B03.

4.5 Chip Configuration after Reset

The start vector depends on the switch S1/2. If the switch is off (default), the CPU starts from the external Flash with the uboot-bootloader.
AT91SAM9G20 starts with a startup configuration with fixed buswidth, busfrequency, timing and control signals. Changes are not possible via configuration pins.

If the switch S1/2 is on, the CPU starts from the internal ROM with the Atmel-bootloader. Then it checks the peripheral devices (serial Flash, SDcard) for executable code. If no valid application is found, SAM-BA Boot is executed afterwards. It waits for transactions either on the USB device, or on the DBGU serial port.

Both, main and slow clock crystals, are mounted. So a precise slow clock is available, needed for download via USB with SAM-BA. The slow clock mode is also possible for power saving applications.

4.6 SDRAM

The AT91SAM9G20 has one external bus interface 32bit demultiplexed. This interface is shared by SDRAM, Flash and FPGA. It can be externally accessed at the socket stripe connector throughout the FPGA.

Two 16bit-Synchronous DRAMs are mounted and connected as one RAM with 32bit-buswith.
The RAM density by default is 2x16MiB, but layout provides RAM densities up to 2x64MiB.

By default, a RAM with 7,5ns cycle time for 133MHz busfrequency is mounted. It supports the PC133 bus mode of CPU.

4.7 NOR-Flash

The board is equipped with NOR-Flash because it povides a higher security in terms of data retention compared to NAND-Flash. NAND-Flash works with bad sectors and this can be problematic when such a sector influences the boot-up routine of the module in early time. The module is designed for industrial applications with high requirements for safe operation.

The Flash (default: Spansion S29GL128P) is connected via a 16bit bus. It works with 25ns access time. The density by default is 16MiB. It is possible to use up to 64MiB; it is limited by the address range of the CPU.

4.8 FPGA

The FPGA is used to provide more functionality. There are more IO pins and gates available in the FPGA for more flexibilty. This makes possible e.g. to realize 32bit counter inputs and pulse outputs. Also the FPGA multiplexes the A/D bus to operate with the SJA1000 CAN controller and channels the SPI bus and the A/D bus to the connector for more capabilities at control signals. This enables a high flexibility in using connector pins.

The complete external bus interface is connected to the FPGA, except for SDRAM-specific signals.

The scope of delivery of the ECUcore-9G20 includes the following FPGA features:

- general purpose IO expander
- all inputs are interrupt capable
- so a lot of onboard interrupts will be handled by FPGA, such as RTC, PFI, PHY, CAN
- bus multiplexing for CAN controller
- realizing the external bus interface at the socket stripe with separate CS-managing and eg. CS/Read-combination at single pin
- realizing the external SPI bus with separate /CS
- some pulse outputs and counter inputs with high speed and high resolution capabilities

More information about FPGA functionality is described in Manual "L-1244e_0x FPGA Manual ECUcore-9G20".

FPGA-JTAG-signals onboard are connected to AT91SAM9G20. So modules with Linux support programming the CPLD from the CPU without needing external programming devices for FPGA.

JTAG is not supported in series modules. X501 is not mounted.
If JTAG is needed for programming or debugging the FPGA, X501 must be mounted. The FPGA_x –signals are relevant for FPGA JTAG. No pullups or pulldowns are required, all mounted onboard.

Resistor	Signal
pullup 10k	PLD_TDI, PLD_TDO, PLD_TMS
pulldown 10k	PLD_TCK

The FPGA Lattice ECP2-6 "LFE2-6SE-6FN256I" with configuration device numonyx "M25P80-VMN6P" is used.

4.9 Ethernet Controller

The AT91SAM9G20 supports one 10/100 Ethernet channel by an internal MAC with MII/RMII interface.
An onboard PHY chip KSZ8721BLI from Micrel realizes a 10/100 physical interface. The PHY is connected with the RMII interface of the CPU.

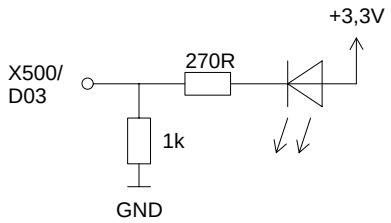
Board connector signals are:

Signal	Description
Eth0_Tx+	Tx+ from PHY
Eth0_Tx-	Tx- from PHY
Eth0_Rx+	Rx+ from PHY
Eth0_Rx-	Rx- from PHY
2V5_EPHY	2,5V PHY-Supply as reference for transformer
Link/Act	output of PHY-LED0 lowactiv
Speed	output of PHY-LED1 lowactiv

Table 5: Ethernet signals

Connect Link/Act and Speed LED with a 270R series resistor, not directly!

The Speed LED is also a strapping pin means that this pin is read at reset. When the pin is open (or high with LED), the PHY is configured for 100MBbps Speed. If you need 10MBbps, place a 1k pulldown resistor to GND at this pin.



Tx and Rx Signals are pulled up with 49,9R to 2,5V-EPHY onboard. So you need only the transformer and connector as external components for communication.

SYS TEC electronic has aquired a pool of these MAC addresses. The MAC address for the first Ethernet interface Eth0 is barcode-labelled and attached on the ECUcore-9G20.

4.10 I2C Module

The ECUcore-9G20 features one I2C interface. This is a 2-wire serial bus used for communication with I2C devices. The bus is brought out via the board connector. The ECUcore-9G20 comes with two onboard I2C devices. Please refer to the table below.

I2C device	Address
Real-Time-Clock Epson 8564JE (U201)	0xA2
Temperature Sensor TI TMP101 (U202)	0x90 (default) 0x92 upon request

Table 6: I2C Components

The I2C module defaults to GPIO after reset. The GPIO module must be configured to enable the peripheral function of the appropriate pins prior to configuring the I2C-Module.

4.10.1 Real Time Clock

The ECUcore-9G20 is equipped with a Real time clock to manage real-time applications. The device offers functions such as calendar clock, alarm and timer. It also outputs pre-defined frequencies (32.768kHz, 1024Hz, 32Hz, 1Hz) via the CLKOUT pin.

RTC characteristics:

- Built-in crystal running at 32768Hz
- Programmable alarm, timer and interrupt functions
- Low power consumption:
  - Bus active: ≤ 1mA
  - Bus inactive, CLKOUT inactive: ≤ 1µA

The Real time clock is supplied with 3.3V DC. If the system voltage is off, a backup battery (connected at X500/A48) supplies the RTC.

Device address:
- 0xA2 when write mode
- 0xA3 when read mode

The pin CLKOE is connected to 3.3V DC. So the clock output can be enabled by setting the bit 'FE' in register 'Clock Out frequency' to 1.

CLKOUT and RTC_INT are connected to FPGA and mapped into FPGA registers to be used by the application.

Signal	Pin at FPGA
CLKOUT	FPGA Pin F15
/RTC_INT	FPGA Pin F14

4.10.2 Temperature Sensor

The ECUcore-9G20 features an optional Temperature sensor TMP101 to record ambient temperatures to, e.g, enable protection from overheating. The ECUcore-9G20 just provides the physical connection between the CPU and the sensor. The communication or any protective measures are software functions to be provided by the user application.

The address is adjustable by a resistor. The following table shows the various assembly options.

Resistor	ADD0 signal	Address
equipped (default)	0 (GND)	1001000x = 0x90
not equipped (upon request)	Float	1001001x = 0x92

Table 7: Temperature Sensor Address

Temperature sensor characteristics:

- Temperature resolution of 0.0625°C
- Temperature range of -55°C to +125°C
- Alert pin as interrupt source if temperature exceeds defined limits

The Alert-Pin is connected to FPGA Pin 11 and mapped to FPGA register to be used by the application.

4.11 SPI Interface

The ECUcore-9G20 allows high-speed serial communication with SPI devices such as EEPROM. The SPI bus signals are not directly brought out via the board connector. The FPGA is used to channel the SPI data and clock signal to the connector. So it is possible to generate more CS signals (CPU-CS-Pins are shared with Address-Pins). It is also possible to realize the full SPI functionality in the FPGA and release the CPU from SPI communication.

The SPI is not used onboard, but there is an EEPROM at the Development Board to test the function.

The following table shows the available SPI signals.

X500-Pin	Connector signal	SPI signal	Description
A26, B27, A27	FPGA_IO44 FPGA_IO45 FPGA_IO 46	/SPI_CS0 /SPI_CS1 /SPI_CS2	SPI-ChipSelect
B28	FPGA_IO47	SPI_MTSR (SOUT)	Master Transmit Slave Receive
A28	FPGA_IO48	SPI_MRST (SIN)	Master Receive Slave Transmit
B29	FPGA_IO49	SPI_CLK (SCK)	Clock

Table 8: SPI Signals

4.12 CAN Interface

The ECUcore-9G20 includes one CAN interface. It is realized by the onboard CAN controller SJA1000. The CAN controller has a multiplexed Address/Data bus. The FPGA is used to adapt it to the demultiplexed CPU bus interface. The CAN bus is brought out via the board connector as 5V-TTL interface.

The CPU provides different pins for CAN. The default connection is:

Externally a 5V CAN transceiver can be directly connected to CAN pins (by using the CAN_VCC Pin X500/B25). Alternatively, a galvanic decoupled CAN interface can be build to save the module for EMC. Drive capability of CAN output pins is 8mA. CAN_VCC is dimensioned for maximum 100mA.

4.13 Serial Interface

The AT91SAM99G20 supports up to 4 independent USARTs and two 2-wire UARTs.

Available on the board connector is:

- USART0 with DTR,DSR,DCD,RTS,CTS and SCK
- USART1 with RTS,CTS and SCK
- USART2 with SCK
- UART3, UART4 (Debug UART) and UART5

All signals are brought out with LVTTL-Level. They are used to interface serial communication via RS232 or RS485 by external transceivers.

4.14SDCard/MMC Interface

The AT91SAM9G20 has a two-slot multimedia-card interface which means the two interfaces use one clock. Both interfaces (A and B) are brought out via connector. Each interface has 4 data pins for the usage as SDCard or MMC interface.

Card Detect and Card Protect are not supported by default. Use FPGA_IO Pins to watch these signals.

MicroSD sockets only provide SLOT-contact to indicate a closed socket. At the Development Board this signal is connected to pin AD2 (X500/D08) via jumper. The SLOT-contact can be watched by the software.

4.15USB Interfaces

The AT91SAM9G20 provides one USB2.0 (12Mbit/s) device and two USB2.0 (12Mbit/s) host interfaces. Each interface is brought out at the connector with its P and M signal.
No security components (such as TVS-Diodes) are provided at the module, they must be mounted near the USB connector on the baseboard.
Overcurrent protection is not supported by default. If necessary, use an external current limiting IC with overcurrent-flag and connect it to the FPGA_IO.

4.16SSC Interface

The Synchronous Serial Controller (SSC) provides a synchronous communication link with external devices. It supports many serial synchronous communication protocols generally used in audio and telecom applications such as I2S, Short Frame Sync, Long Frame Sync.
The six-signal lines are directly brought out at connector X500.

4.17EmbeddedICE Port

The EmbeddedICE-Port is not provided by the series module. It requires the socket stripe X501. This connector is not mounted.

ECUcore-9G20 is supposed for programming in Linux. There is no hardware debugging needed at all. To Debug at this interface a separate Debugger is necessary too.

If mounting and using the connector, the ARM_x signals are relevant for ICE and must be connected to Debugger. No pullups ore pulldowns are required. Pullups 4k7 are mounted onboard at /NRST, TDI, TMS and TCK.

The JTAGSEL pin is for selecting the ICE or JTAG mode. The JTAG mode is for the usage with BoundaryScan hardware and not needed for debugging. So leave open the pin when using ICE! For using JTAG, put it to GND.

5 Technical Data

The physical dimensions of the ECUcore-9G20 is shown in the figure below.

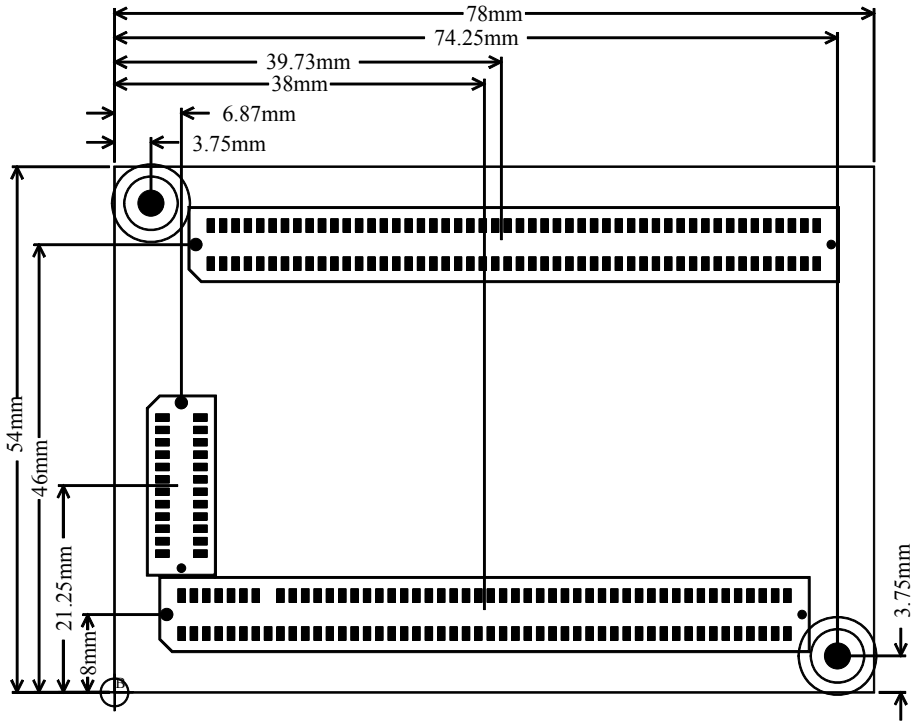


Figure 6: Physical Dimensions

The height including the board connector and components is about 9mm. The thickness of the PCB is about 1.6mm. The maximum component height on top is about 3mm.

dimensions	78mm x 54mm x 8mm
weight	approximately 21,5g
operating temperature	-40°C to +85°C
storage temperature	-40°C to +85°C
operating voltage	3.3V DC ± 5%
current consumption	typ. 400mA
I/O-Level (excl. CAN)	3.3V DC ± 5%
CAN-Level	5V DC ± 5%

Table 9: Technical Data

Document:	Hardware Manual ECUcore-9G20
Document number:	L-1255e_01, Edition March 2010

How would you improve this manual?

Did you find any mistakes in this manual? page

Submitted by:

Customer number: _____

Name: _____

Company: _____

Address: _____

Return to:

SYS TEC electronic GmbH
August-Bebel-Str. 29
D-07973 Greiz
GERMANY
Fax : +49 (0) 36 61 / 62 79 99