

AVR-T32U4 development breadboard **USER'S MANUAL**

Revision B, June 2012
Designed by OLIMEX Ltd, 2012



All boards produced by Olimex LTD are ROHS compliant

Disclaimer:

© 2012 Olimex Ltd. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other terms and product names may be trademarks of others.

The information in this document is provided in connection with Olimex products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Olimex products.

Neither the whole nor any part of the information contained in or the product described in this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder, unless it is clearly stated by Olimex that the product is licensed under the Open Source Hardware (OSHW) Definition 1.0.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by OLIMEX in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded. This document is intended only to assist the reader in the use of the product. OLIMEX Ltd. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

**Thank you for purchasing AVR-T32U4 development board assembled by
OLIMEX LTD**

Table of Contents

CHAPTER 1.....	
OVERVIEW.....	
1. Introduction to the chapter.....	5
1.1 Features.....	5
1.2 Similar boards.....	5
1.3 Organization.....	6
Chapter 2.....	
Introduction to arduino/pinguino/maple.....	
2. What is Arduino?.....	7
2.1 First steps with Arduino IDE.....	8
2.1.1 For purchases after the official release of Leonardo.....	8
2.1.2 For purchases before the official release of Leonardo.....	8
CHAPTER 3	
SETTING UP THE AVR-T32U4 BOARD.....	
3. Introduction to the chapter.....	10
3.1 Electrostatic warning.....	10
3.2 Requirements.....	10
3.3 Powering the board.....	11
3.4 Prebuilt software.....	12
CHAPTER 4.....	
AVR-T32U4 BOARD DESCRIPTION.....	
4. Introduction to the chapter.....	13
4.1 Layout (top view).....	13
4.2 Layout (bottom view).....	14
CHAPTER 5.....	
THE ATMEGA32U4 MICROCONTROLLER.....	
5. Introduction to the chapter.....	15
5.1 The microcontroller's features.....	15
CONTROL CIRCUITY.....	
6. Introduction to the chapter.....	18
6.1 Reset.....	18
6.2 Clock.....	18
CHAPTER 7.....	
HARDWARE.....	
7. Introduction to the chapter.....	19

7.1 ICSP pin holes.....	19
7.2 UEXT.....	20
7.4 Extension pins.....	21
7.5 USB mini connector.....	22
7.7 Jumper description.....	22
7.9 Additional hardware components.....	23
<u>CHAPTER 8.....</u>	
<u>MEMORY AND BLOCK DIAGRAM.....</u>	
8. Introduction to the chapter.....	24
8.1 Memory organization.....	25
<u>CHAPTER 9.....</u>	
<u>SCHEMATICS.....</u>	
9. Introduction to the chapter.....	26
9.1 Eagle schematic.....	26
9.2 Physical dimensions.....	28
<u>CHAPTER 10.....</u>	
<u>REVISION HISTORY.....</u>	
10. Introduction to the chapter.....	29
10.1 Document revision.....	29
10.2 Web page of your device.....	30

CHAPTER 1

OVERVIEW

1. Introduction to the chapter

Thank you for choosing the AVR-T32U4 development board from Olimex! This document provides a User's Guide for the Olimex AVR-T32U4 development board. As an overview, this chapter gives the scope of this document and lists the board's features. The document's organization is then detailed.

The AVR-T32U4 development board enables code development of applications running on the ATmega32U4 microcontroller, manufactured by Microchip technology.

1.1 Features

- ATMEGA32U4 microcontroller, all in one USB communication no need for external FTDI chip.
 - Input power supply taken from mini USB
 - Feeds power to the breadboard bus
 - Works both on 3.3V and 5V
 - UEXT connector which allow many existing modules like RF, ZIGBEE, GSM, GPS to be connected
 - The LEDs and the BUTTONs are on the edge of the board so there is easy access even if the boards have shields on them
 - Mini USB connector is used which is common and used in most cell phones, so you do not have to buy other cables
-

1.2 Similar boards

AVR-T32U4 is ARDUINO-LEONARDO-like breadboard with ATmega32U4.

It is a powerful board considering its low price but yet a starter board for the Arduino family.

The more powerful Arduino boards we have in the range are OLIMEXINO-32U4 again based on Leonardo and OLIMEXINO-328 – based on Arduino Duenomilanove. Both differ from AVR-T32U4 in having the standard Arduino shield connectors and additional features. OLIMEXINO-328

has a more sophisticated supply circuit, USB OTG connector and there are more Arduino projects available for it.

1.3 Organization

Each section in this document covers a separate topic, organized as follow:

- Chapter 1 is an overview of the board usage and features
- Chapter 2 has a short introduction to Arduino platform and its derivatives Pinguino and Maple
- Chapter 3 provides a guide for quickly setting up the board
- Chapter 4 contains the general board diagram and layout
- Chapter 5 describes the component that is the heart of the board: the ATmega32U4 microcontroller
- Chapter 6 is an explanation of the control circuitry associated with the microcontroller to reset. Also shows the clocks on the board
- Chapter 7 covers the connector pinout, peripherals and jumper description
- Chapter 8 shows the processor diagram and memory map
- Chapter 9 provides the schematics
- Chapter 10 contains the revision history

CHAPTER 2

INTRODUCTION TO ARDUINO/PINGUINO/MAPLE

2. What is Arduino?

Arduino is an open-source electronics prototyping platform, designed to make the process of using electronics in multidisciplinary projects easily accessible. The hardware consists of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board I/O support. The software consists of a standard programming language and the boot loader that runs on the board.

Arduino hardware is programmed using a Wiring-based language (syntax + libraries), similar to C++ with some simplifications and modifications, and a Processing-based Integrated Development Environment (IDE).

The project began in Ivrea, Italy in 2005 aiming to make a device for controlling student-built interaction design projects less expensively than other prototyping systems available at the time. As of February 2010 more than 120,000 Arduino boards had been shipped. Founders Massimo Banzi and David Cuartielles named the project after a local bar named Arduino. The name is an Italian masculine first name, meaning "strong friend". The English pronunciation is "Hardwin", a namesake of Arduino of Ivrea.

More information could be found at the creators web page <http://arduino.cc/> and in the Arduino Wiki <http://en.wikipedia.org/wiki/Arduino>

To make the story short - Arduino is easy for beginners who lack Electronics knowledge, but also does not restrict professionals as they can program it in C++ or mix of Arduino/C++ language. There are thousands of projects which makes it easy to startup as there is barely no field where Arduino enthusiasts to have not been already.

Arduino has inspired two other major derivatives - MAPLE and PINGUINO. Based on 8-bit AVR technology the computational power of Arduino boards is modest, this is why a team from MIT developed the MAPLE project which is based on ARM7 STM32F103RBT6 microcontroller. The board has same friendly IDE as Arduino and offers the same capabilities as hardware and software but runs the Arduino code much faster. The Maple project can be found at <http://leaflabs.com>

In parallel with Arduino another project was started called PINGUINO. This project chose its first implementation to be with PIC microcontrollers, as AVRs were hard to find in some parts of the world like South America so it is likely to see lot of PINGUINO developers are from that part of the

world. PINGUINO project founders decided to go with Python instead Java for processing language. For the moment PINGUINO is much more flexible than Arduino as it is not limited to 8bit microcontrollers. Currently the IDE, which has GCC in background, can support 8-bit PIC microcontrollers, 32bit PIC32 (MIPS) microcontrollers and ARM7/CORTEXM3 microcontrollers which makes PINGUINO very flexible because once you make your project you can migrate easily through different hardware platforms and not being bound to a single microcontroller manufacturer. The PINGUINO project can be found at: <http://www.pinguino.cc>.

2.1 First steps with Arduino IDE

Depending when you purchased the board there are two scenarios since the PID (product ID) which is stored in the firmware and expected by the IDE is different between versions 1.0.0 and 1.0.1 (1.0.1 is the first release of the Arduino IDE with official support for Leonardo).

2.1.1 For purchases after the official release of Leonardo

Boards produced after the official release of Arduino Leonardo have the latest bootloader available with the adjusted bootloader PID. Download the latest Arduino IDE and extract the package. You can find the needed driver for the bootloader located in folder /drivers. You click update on the device in device manager and point the installer to the /drivers folder.

Then launch Arduino IDE and set the correct board and COM port in TOOLS menu.

From File->Open navigate to the examples folder which is found in the Arduino IDE installation folder. Choose a simple example to begin with like BlinkWithoutDelay.ino and open it. Click verify and after it compiles click "Upload". Voila, you have uploaded code to your Arduino device.

Now start exploring the world of Arduino! More info and help on the software can be found and the official Arduino web site: <http://www.arduino.cc/>.

2.1.2 For purchases before the official release of Leonardo

If you have purchased the T-32U4 before the official release of the Arduino Leonardo board there are two scenarios for using Arduino IDE:

- 1) Download Arduino IDE 1.0 RC2 or Arduino IDE 1.0.0. When you extract the package check if the Arduino Leonardo section is the \hardware\arduino\boards is uncommented. If the section has number signs (#) in front remove the number signs (#).

2) If you want to use the latest version (which by the time of writing is 1.0.1) you would need to change to bootloader firmware. This will require an AVR programmer tool.

You can get the bootloader in .elf format from our web site. You can also use the original bootloader hex if you set the fuses and lock bits for the processor as follows:

```
low_fuses=0xff  
high_fuses=0xd8  
extended_fuses=0xcb
```

lock_bits (choose “SPM Prohibited in Boot Section”) from the drop down menu which will set the bits either to EF or 2F.

After you have downloaded and extracted the desired IDE you need to install the driver for your operating system. You can find the needed driver for the bootloader located in folder /drivers. You click update on the device in device manager and point the installer to the /drivers folder.

Then launch Arduino IDE and set the correct board and COM port in TOOLS menu.

From File->Open navigate to the examples folder which is found in the Arduino IDE installation folder. Choose a simple example to begin with like BlinkWithoutDelay.ino and open it. Click verify and after it compiles click “Upload”. Voila, you have uploaded code to your Arduino device.

Now start exploring the world of Arduino! More info and help on the software can be found and the official Arduino web site: <http://www.arduino.cc/>.

CHAPTER 3

SETTING UP THE AVR-T32U4 BOARD

3. Introduction to the chapter

This section helps you set up the AVR-T32U4 development board for the first time.

Please consider first the electrostatic warning to avoid damaging the board, then discover the hardware and software required to operate the board.

The procedure to power up the board is given, and a description of the default board behavior is detailed.

3.1 Electrostatic warning

AVR-T32U4 is shipped in a protective anti-static package. The board must not be exposed to high electrostatic potentials. A grounding strap or similar protective device should be worn when handling the board. Avoid touching the component pins or any other metallic element.

3.2 Requirements

In order to set up the AVR-T32U4, the following items are required:

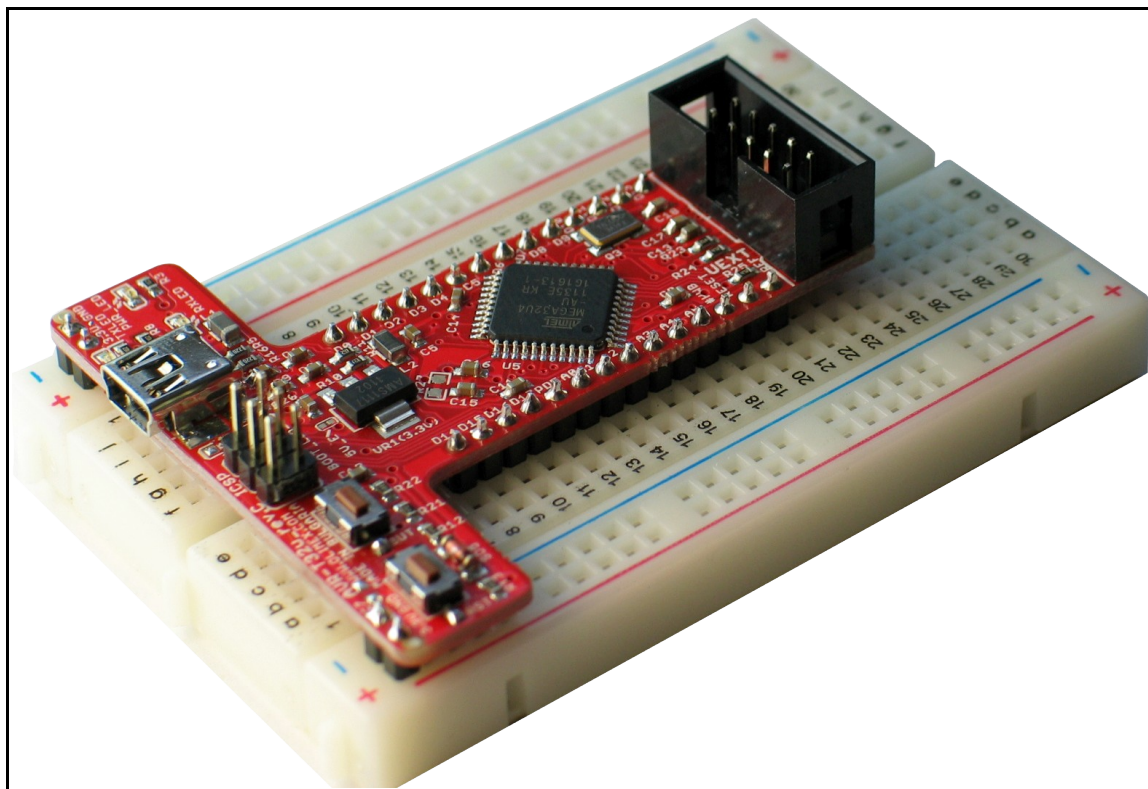
If using the free ARDUINO IDE the ONLY requirement is a miniUSB to USB-A cable to connect to a computer.

Since this is a board mainly focused on breadboarding having a breadboard is highly advisable. This way you can have access to the free pins using jumper wires and no need of pesky soldering and unsoldering! Good and cheap choices are the ones we offer:

BREADBOARD-1 - 82x52x10 mm solderless breadboard for experimenting

BREADBOARD-MINI - 45x35x8.5 mm solderless breadboard for experimenting

On the next page you can find a picture of AVR-T32U4 mounted on a breadboard.



If you want to upgrade the bootloader, or program the board without the ARDUINO-IDE there are pinouts for ICSP programmer (note that there IS a 6-pin connector mounted). You can use our AVR-ISP-ISO, or AVR-ISP-TINY programmer for custom programming without ARDUINO IDE.

Having an AVR programmer is highly advisable since the bootloader software is still not perfect and it is possible to destroy it only by software means (for instance disabling all interrupts and then entering infinite loop would hang the bootloader) which would require reinstalling the bootloader.

3.3 Powering the board

The AVR-T32U4 board is self-powered by the miniUSB if it is connected to a USB hub of a computer.

Alternatively it can be powered via the POWER JACK by 7Vdc to 12Vdc or via the 3.7V-LI_BAT connector by 3.7V battery.

The bootloader starts automatically.

On powering the board PWR_LED should turn on. RXLED and TXLED should turn on and monitor the send receive on the virtual COM port.

3.4 Prebuilt software

The board comes with prebuilt ARDUINO bootloader. The different part between the other Arduinos and Leonardo is that Leonardo doesn't use FTDI chip nor external chip for the virtual COM port but it is handled by the ATmega32U4.

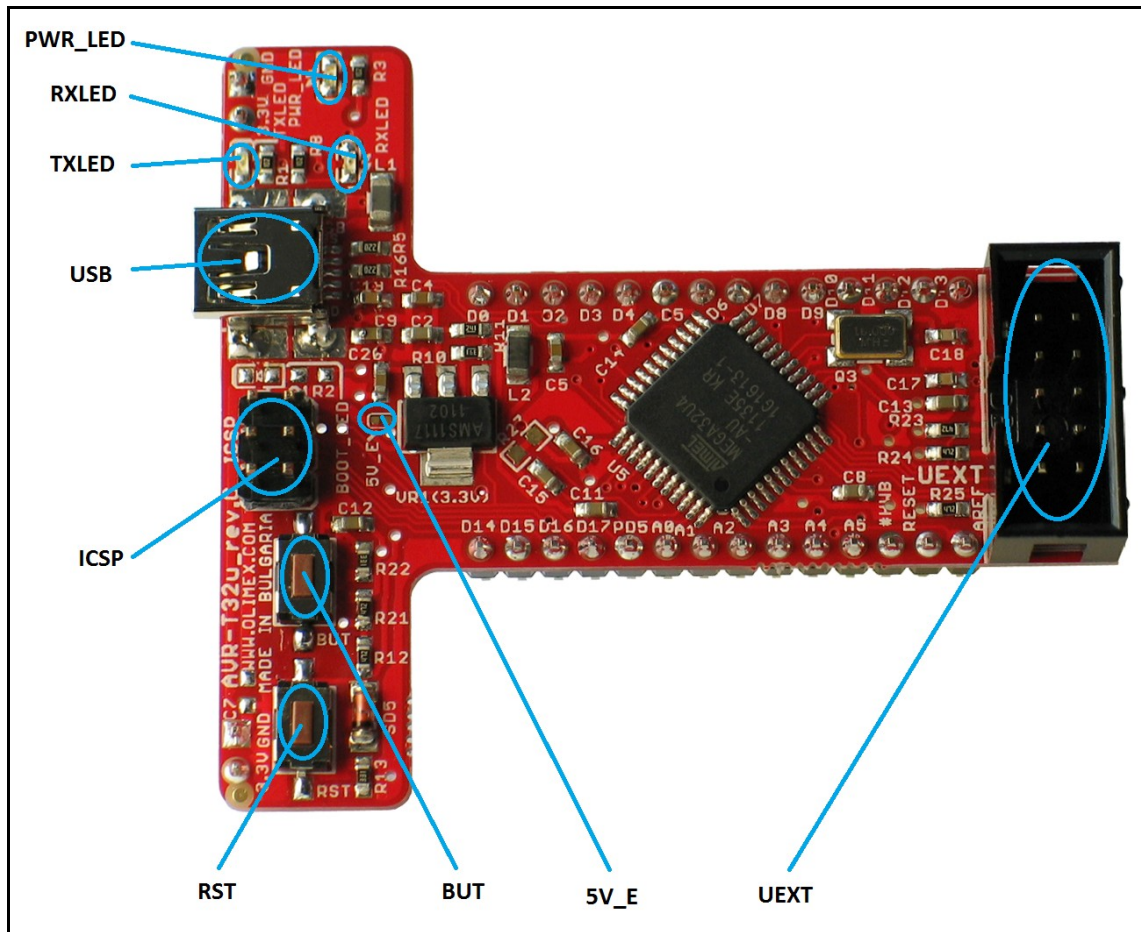
CHAPTER 4

AVR-T32U4 BOARD DESCRIPTION

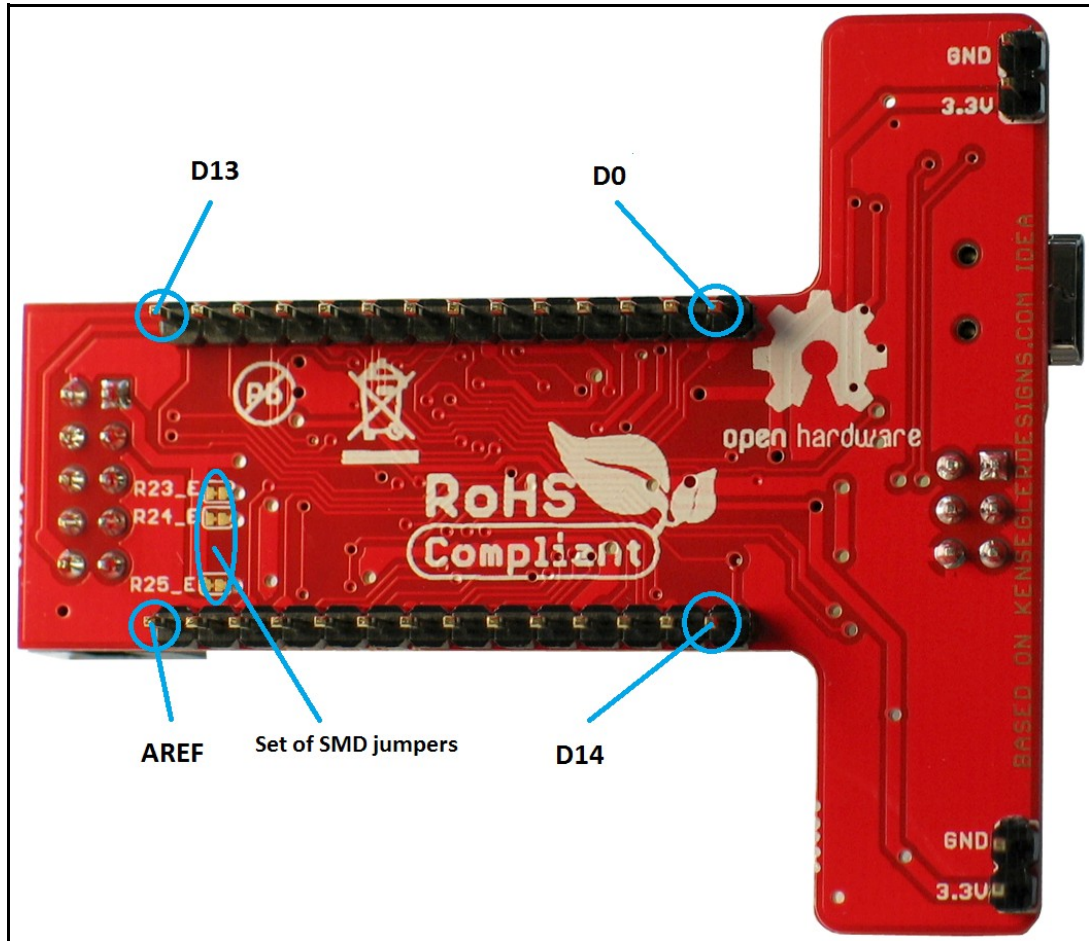
4. Introduction to the chapter

Here you get acquainted with the main parts of the board. Note the names used on the board differ from the names used to describe them. For the actual names check the AVR-T32U4 board itself.

4.1 Layout (top view)



4.2 Layout (bottom view)



CHAPTER 5

THE ATMEGA32U4 MICROCONTROLLER

5. Introduction to the chapter

In this chapter is located the information about the heart of AVR-T32U4 – its microcontroller. The information is a modified version of the datasheet provided by its manufacturers.

5.1 The microcontroller's features

- High Performance, Low Power AVR® 8-Bit Microcontroller
 - Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-Chip 2-cycle Multiplier
 - Non-volatile Program and Data Memories
 - 32K Bytes of In-System Self-Programmable Flash
 - 2.5K Bytes Internal SRAM
 - 1K Bytes Internal EEPROM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C(1)
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - All supplied parts are preprogrammed with a default USB bootloader
 - Programming Lock for Software Security
 - JTAG (IEEE std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
 - USB 2.0 Full-speed/Low Speed Device Module with Interrupt on Transfer Completion
 - Complies fully with Universal Serial Bus Specification Rev 2.0
 - Supports data transfer rates up to 12 Mbit/s and 1.5 Mbit/s
-

- Endpoint 0 for Control Transfers: up to 64-bytes
- 6 Programmable Endpoints with IN or Out Directions and with Bulk, Interrupt or Isochronous Transfers
- Configurable Endpoints size up to 256 bytes in double bank mode
- Fully independent 832 bytes USB DPRAM for endpoint memory allocation
- Suspend/Resume Interrupts
- CPU Reset possible on USB Bus Reset detection
- 48 MHz from PLL for Full-speed Bus Operation
- USB Bus Connection/Disconnection on Microcontroller Request
- Crystal-less operation for Low Speed mode
- Peripheral Features
 - On-chip PLL for USB and High Speed Timer: 32 up to 96 MHz operation
 - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
 - Two 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - One 10-bit High-Speed Timer/Counter with PLL (64 MHz) and Compare Mode
 - Four 8-bit PWM Channels
 - Four PWM Channels with Programmable Resolution from 2 to 16 Bits
 - Six PWM Channels for High Speed Operation, with Programmable Resolution from 2 to 11 Bits
 - Output Compare Modulator
 - 12-channels, 10-bit ADC (features Differential Channels with Programmable Gain)
 - Programmable Serial USART with Hardware Flow Control
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
 - On-chip Temperature Sensor
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal 8 MHz Calibrated Oscillator
 - Internal clock prescaler & On-the-fly Clock Switching (Int RC / Ext Osc)
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Package
 - All I/O combine CMOS outputs and LVTTL inputs
 - 26 Programmable I/O Lines

- 44-lead QFN Package, 7x7mm
- Operating Voltages
 - 2.7 - 5.5V
- Operating temperature
 - Industrial (-40°C to +85°C)
- Maximum Frequency
 - 8 MHz at 2.7V - Industrial range
 - 16 MHz at 4.5V - Industrial range

For comprehensive information on the microcontroller visit the Atmel web page for a datasheet.

At the moment of writing the microcontroller datasheet can be found at the following link:

<http://www.atmel.com/Images/doc7766.pdf>

CHAPTER 6

CONTROL CIRCUITY

6. Introduction to the chapter

Here you can find information about reset circuit, power circuit and quartz crystal locations.

6.1 Reset

AVR-T32U4 reset circuit includes R12 (4.7k Ω), R13 (330 Ω), SD5 (Shottky diode), ATMega32U4 pin 13 (#RESET) and a RESET button. Note that it also can be found at the ICSP header pin 5 and test pad RESET.

6.2 Clock

16 MHz quartz crystal Q1 is found at pins 16 and 17 of the processor.

CHAPTER 7

HARDWARE

7. Introduction to the chapter

In this chapter are presented the connectors that can be found on the board all together with their pinout. Proto area is shown. Jumpers functions are described. Notes and info on specific peripherals are presented. Notes regarding the interfaces are given.

7.1 ICSP pin holes

The 6 pin ICSP header provides interface for custom programming/debugging outside of the PINGUINO IDE. Note the pin holes have a connector mounted over. The pinout can be found in the table below. It is important to check the jumper configuration for proper use of the ICSP pin hole .

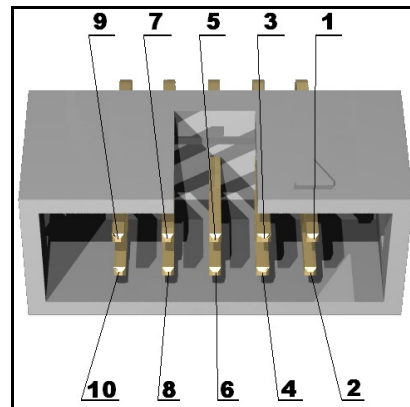
ICSP pin holes			
Pin #	Signal Name	Pin #	Signal Name
1	MISO (PB3)	4	MOSI (PB2)
2	Vcc	5	RESET
3	SCK (PB1)	6	GND

7.2 UEXT

AVR-T32U4 board has UEXT connector and can interface Olimex's UEXT modules. For more information on our UEXT modules please visit:

<http://www.olimex.com/dev/OTHER/UEXT.pdf>

Pin #	Signal Name
1	+3.3V
2	GND
3	TXD
4	RXD
5	SCL
6	SDA
7	MISO
8	MOSI
9	SCK
10	UEXT_#CS



7.4 Extension pins

For your convenience the pads are named individually near each of them. Please take extra care about the numbering but consider that there might be offset.

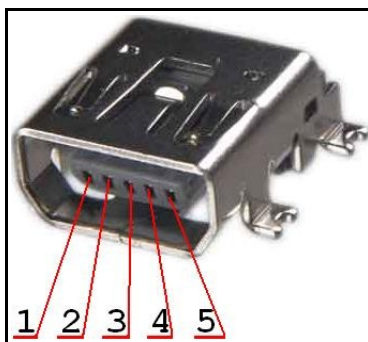
Also note that the board comes with no connectors soldered on the pin holes. Before being able to connect with a shield you have to solder connectors yourself.

Pad Name	Signal	Pad Name	Signal
AREF	AREF	D13(UEXT_#CS)	PF7/ICP3/CLK0/OC4A
RESET	RST	D12	PD6/T1/#OC4D/ADC9
#HWB (BUT)	PE2/#HWB	D11	PB7/PCINT7/OC0A/OC1C/ #RTS
A5	PF0/ADC0	D10	PB6/PCINT6/OC1B/OC4B/ ADC13
A4	PF1/ADC1	D9	PB5/PCINT5/OC1A/#OC4B/ ADC12
A3	PF4/ADC4/TCK	D8	PB4/PCINT4/ADC11
A2	PF5/ADC5/TMS	D7 (BOOT_LED_	INT6/AIN0/PE6
A1	PD6/ADC6/TDO	D6	PD7/T0/OC4D/ADC10
A0	PF7/ADC7/TDI	D5	PC6/OC3A/#OC4A
PD5(TXLED)	PD5/XCK/CTS	D4	PD4/ICP1/ADC8
D17(MISO)	PB3/PDO/PCINT3/ MISO	D3 (SCL)	PD0/OC0B/SCL/INT0
D16(MOSI)	PB2/PDI/PCINT2/ MOSI	D2 (SDA)	PD1/SDA/INT1
D15(SCK)	PB1/PCINT1/ SCLK	D1 (TXD)	PD3/TXD1/INT3
D14	PB0/SS/PCINT0	D0 (RXD)	PD2/RXD1/INT2

7.5 USB mini connector

Note that the USB is configured only as device.

Pin #	Signal Name
1	+5V
2	D-
3	D+
4	Not connected
5	GND



7.7 Jumper description



5V_E

When closed connects D8 from Arduino shield connector to FET3 so you can control the power on the UEXT by software means.

When open D8 becomes GPIO. SMD jumper.

Default state is closed.



R23_E

This jumper controls whether the board is powered by 3.3Vdc source or 5Vdc.

Default position is 2-3 - 5Vdc.

**R24_E**

When closed enables analog GND. SMD jumper.

Default state is closed.

**R25_E**

When closed connects D8 from Arduino shield connector to FET3 so you can control the power on the UEXT by software means.

When open D8 becomes GPIO. SMD jumper.

Default state is closed.

7.9 Additional hardware components

The components below are mounted on the AVR-T32U4 board but are not discussed above. They are listed here for completeness:

General purpose button

2 LEDs showing virtual port activity

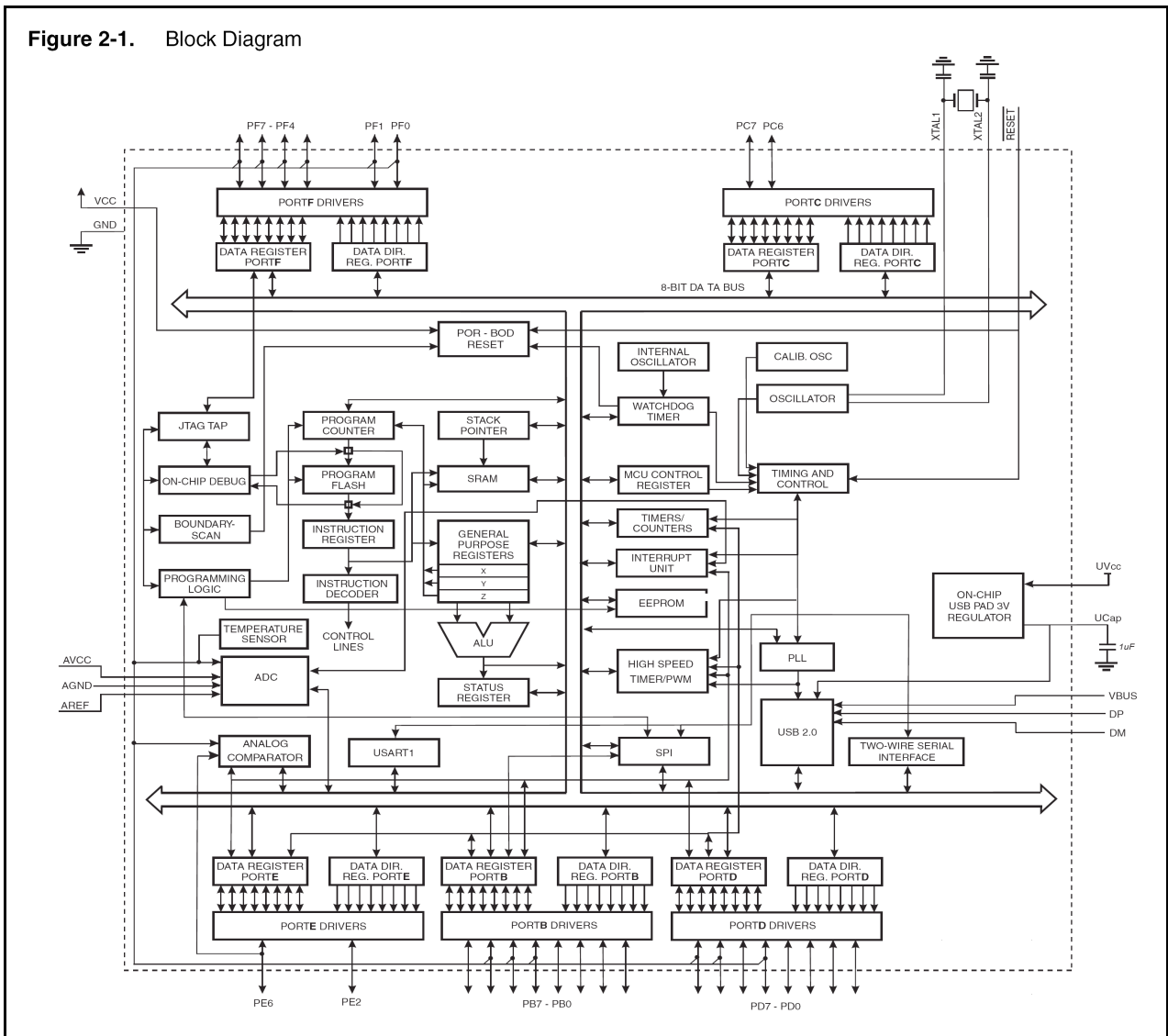
CHAPTER 8

MEMORY AND BLOCK DIAGRAM

8. Introduction to the chapter

Below is located the block diagram of the processor and on the next page you can find a memory map for this family of processors. It is strongly recommended to refer to the original datasheet released by Atmel for ones of higher quality.

Figure 2-1. Block Diagram



8.1 Memory organization

Memory		Mnemonic	ATmega32U4
Flash	Size	Flash size	32K bytes
	Start Address	- 0x0000	
	End Address	Flash end	0x7FFF ⁽¹⁾ 0x3FFF ⁽²⁾
32 Registers	Size	-	32 bytes
	Start Address	-	0x0000
	End Address	-	0x001F
I/O Registers	Size	-	64 bytes
	Start Address	-	0x0020
	End Address	-	0x005F
Ext I/O Registers	Size	-	160 bytes
	Start Address	-	0x0060
	End Address	-	0x00FF
Internal SRAM	Size	ISRAM size	2,5K bytes
	Start Address	ISRAM start	0x100
	End Address	ISRAM end	0x0AFF
External Memory	Not Present.		
EEPROM	Size	E2 size	1K bytes
	End Address	E2 end	0x03FF

Notes: 1. Byte address.
2. Word (16-bit) address.

CHAPTER 9

SCHEMATICS

9. Introduction to the chapter

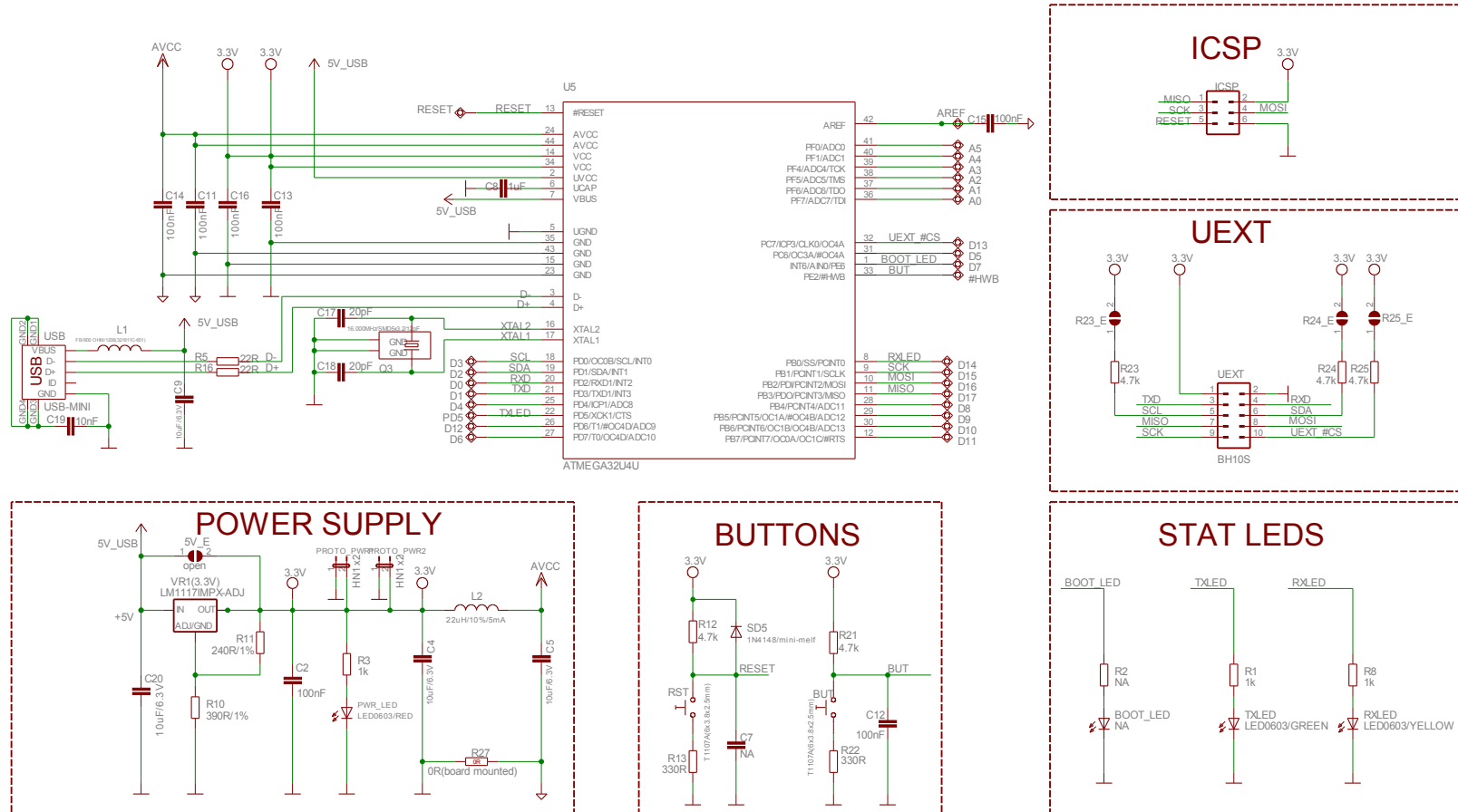
In this chapter are located the schematics describing logically and physically AVR-T32U4.

9.1 Eagle schematic

AVR-T32U4 schematic is visible for reference here. You should zoom the pdf for a better view and also string search is available. You can also find them on the web page for AVR-T32U4 at our site: <http://www.olimex.com/dev/AVR-T32U4.html>. They are located in HARDWARE section.

Note that AVR-T32U4 schematic is released under [Creative Commons Attribution-Share Alike 3.0 United States License](#) and the whole project is released under

The EAGLE schematic is situated on the next page for quicker reference.



AVR-T32u Rev.C
 Designed by: WWW.OLIMEX.COM/DEV
 BASED ON WWW.KENSEGLERDESIGNS.COM IDEA
 Released under Creative Commons Attribution-Share Alike 3.0 United States License

CHAPTER 10

REVISION HISTORY

10. Introduction to the chapter

In this chapter you will find the current and the previous version of the document you are reading. Also the web-page for your device is listed. Be sure to check it after a purchase for the latest available updates and examples.

10.1 Document revision

Revision	Changes	Modified Pages
Initial	Initial Creation	All
B, 25.06.12	Added links to chapters	3, 4
	Added info on different bootloader versions	8, 9

10.2 Web page of your device

The web page you can visit for more info on your device is <http://www.olimex.com/dev/AVR-T32U4.html>. There you can find more info and some examples.

Order codes:

AVR-T32U4 - completely assembled and tested

USB-MINI-CABLE - USBmini to USB-A cable

BREADBOARD-1 - 82x52x10 mm solderless breadboard for experimenting

BREADBOARD-MINI - 45x35x8.5 mm solderless breadboard for experimenting

How to order?

You can order to us directly or by any of our distributors.

Check our webpage <http://www.olimex.com/> for more info.