Page STK600 User Guide 1 of 35 STK600 User Guide

1. Introduction

Overview Features What's New Known Issues Device Support 2. Getting Started

Kit Contents System Requirements Quick Start Connecting the Hardware Example Application: Using LEDs and Switches 3. Target Socket System

Socket System Socket Card and Routing Card Selecting the correct routing and socket cards Mounting the cards Signal integrity 4. Hardware Description

STK600 Block Diagram Target Voltage VTG Analog Reference Voltages RESET Control Port Connectors AVR32 Pin Mapping LEDs and Switches Clock Settings User RS232 Interface DataFlash Non-Volatile Memory Expansion Connectors User USB connector CAN transceiver LIN transceiver Miscellaneous 5. Programming

ISP Programming Parallel High Voltage Programming Serial High Voltage Programming JTAG Programming PDI Programming In-System Programming of an External Target System 6. Troubleshooting and Support

Troubleshooting Guide Technical Support Manual Firmware Upgrade

**Chapter 1. Introduction**

**Overview**

Congratulation with your STK600 AVR® Flash MCU Starter Kit. The STK600 is a complete starter kit and development system for the AVR and AVR®32 flash microcontrollers from ATMEL® Corporation. It is designed to give designers a quick start to develop code on the AVR, combined with advanced features for using the starter kit to prototype and test new designs.

New firmware releases for STK600 are embedded with the releases of AVR Studio®. The upgrade process will start when you connect to the STK600 board (you will be asked to perform the procedure). If for some reason the automatic upgrade fails, please try the manual upgrade procedure.

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

**Features**

●

AVR Studio 4/AVR32 Studio compatible

●

USB Interface to PC for programming and control

●

Powered from USB bus or from an external 10-15V DC power supply

●

Serial In-System Programming (ISP) of AVR devices

●

JTAG programming of AVR and AVR32 devices

●

ISP and JTAG programming of AVR devices in external target systems

●

Flexible routing and socket card system for easy mounting of all supported devices

●

8 push-buttons for general use

●

8 LEDs for general use

●

All AVR I/O ports easily accessible through pin header connectors

●

Expansion connectors for plug-in modules and prototyping area

●

On-board 2Mbit Dataflash for non-volatile data

●

USB mini-AB (On-The-Go) connector for USB devices

●

PHY and DSUB-9 connetor for RS232 interface

●

PHY and DSUB-9 connector for CAN bus

●

PHY and header for LIN bus

●

Device board with an ATmega2560 AVR microcontroller is included.

**Note**

Socketcards and routingcards must be bought separately.

**What's New**

**April 8th, 2008**

STK600 firmware versions 02.05, 02.00, 02.00

**March 27th, 2008**

STK600 firmware versions 02.04, 02.00, 02.00

**Known Issues**

There are at the moment no known issues with the STK600.

**Device Support**

AVR Studio and AVR32 Studio has support for a range of devices in all speed grades. Support for new AVR devices may be added in new versions

Page 2 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

Page STK600 User Guide 3 of 35 of the software. Latest versions of AVR Studio and AVR32 Studio are always available from www.atmel.com.

Table 1.1. AVR Studio device support

Device Routing card Socket card Programming modes Comment ATtiny11 STK600-RC008T-2 STK600-DIP HVSP ATtiny12 STK600-RC008T-2 STK600-DIP HVSP, ISP ATtiny13 STK600-RC008T-2 STK600-DIP HVSP, ISP ATtiny15 STK600-RC008T-7 STK600-DIP HVSP, ISP ATtiny2313 STK600-RC020T-1 STK600-DIP HVPP, ISP ATtiny43U STK600-TinyX3U HVPP, ISP ATtiny24 STK600-RC014T-12 STK600-DIP HVSP, ISP ATtiny44 STK600-RC014T-12 STK600-DIP HVSP, ISP ATtiny84 STK600-RC014T-12 STK600-DIP HVSP, ISP ATtiny25 STK600-RC008T-2 STK600-DIP HVSP, ISP ATtiny45 STK600-RC008T-2 STK600-DIP HVSP, ISP ATtiny85 STK600-RC008T-2 STK600-DIP HVSP, ISP ATtiny26 STK600-RC020T-8 STK600-DIP HVPP, ISP ATtiny261 STK600-RC020T-8 STK600-DIP HVPP, ISP ATtiny461 STK600-RC020T-8 STK600-DIP HVPP, ISP ATtiny861 STK600-RC020T-8 STK600-DIP HVPP, ISP ATtiny167 STK600-RC020T-23 STK600-SOIC HVPP, ISP ATtiny28 STK600-RC028T-3 STK600-DIP HVPP ATtiny48 STK600-RC028M-6 STK600-DIP HVPP, ISP ATtiny88 STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega8515 STK600-RC040M-4 STK600-DIP HVPP, ISP ATmega8535 STK600-RC040M-5 STK600-DIP HVPP, ISP ATmega162 STK600-RC040M-4 STK600-DIP HVPP, ISP, JTAG ATmega16 STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega32 STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega164P STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega324P STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega644 STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega644P STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega1284P STK600-RC040M-5 STK600-DIP HVPP, ISP, JTAG ATmega8 STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega48 STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega88 STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega168 STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega48P STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega88P STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega168P STK600-RC028M-6 STK600-DIP HVPP, ISP ATmega328P STK600-RC028M-6 STK600-DIP HVPP, ISP, JTAG ATmega64 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG ATmega128 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG ATmega1281 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG ATmega2561 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG ATmega165 STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega165P STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega169 STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega169P STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega325 STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega325P STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega329 STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega329P STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega645 STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega649 STK600-RC064M-10 STK600-TQFP64 HVPP, ISP, JTAG ATmega640 STK600-RC100M-11 STK600-TQFP100 HVPP, ISP, JTAG ATmega1280 STK600-RC100M-11 STK600-TQFP100 HVPP, ISP, JTAG ATmega2560 STK600-RC100M-11 STK600-TQFP100 HVPP, ISP, JTAG ATmega2560 STK600-ATMEGA2560 HVPP, ISP, JTAG

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Page STK600 User Guide 4 of 35 ATmega3250 STK600-RC100M-18 STK600-TQFP100 HVPP, ISP, JTAG ATmega3250P STK600-RC100M-18 STK600-TQFP100 HVPP, ISP, JTAG ATmega3290 STK600-RC100M-18 STK600-TQFP100 HVPP, ISP, JTAG ATmega3290P STK600-RC100M-18 STK600-TQFP100 HVPP, ISP, JTAG ATmega6450 STK600-RC100M-18 STK600-TQFP100 HVPP, ISP, JTAG ATmega6490 STK600-RC100M-18 STK600-TQFP100 HVPP, ISP, JTAG AT90USB82 STK600-RC32U-20 STK600-TQFP32 HVPP, ISP AT90USB162 STK600-RC32U-20 STK600-TQFP32 HVPP, ISP ATmega32U4 STK600-RC044U-25 STK600-TQFP44 HVPP, ISP, JTAG AT90USB646 STK600-RC064U-17 STK600-TQFP64 HVPP, ISP, JTAG AT90USB1286 STK600-RC064U-17 STK600-TQFP64 HVPP, ISP, JTAG AT90USB647 STK600-RC064U-17 STK600-TQFP64 HVPP, ISP, JTAG AT90USB1287 STK600-RC064U-17 STK600-TQFP64 HVPP, ISP, JTAG AT90CAN32 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG AT90CAN64 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG AT90CAN128 STK600-RC064M-9 STK600-TQFP64 HVPP, ISP, JTAG ATmega32C1 STK600-RCPWM-22 STK600-TQFP32 HVPP, ISP AT90PWM2 STK600-RCPWM-19 STK600-SOIC HVPP, ISP AT90PWM3 STK600-RCPWM-19 STK600-SOIC HVPP, ISP AT90PWM2B STK600-RCPWM-19 STK600-SOIC HVPP, ISP AT90PWM3B STK600-RCPWM-19 STK600-SOIC HVPP, ISP AT90PWM216 STK600-RCPWM-19 STK600-SOIC HVPP, ISP AT90PWM316 STK600-RCPWM-19 STK600-SOIC HVPP, ISP ATmega32M1 STK600-RCPWM-22 STK600-TQFP32 HVPP, ISP ATmega32HVB STK600-RC044M-24 STK600-TSSOP44 HVPP, ISP ATxmega128A1 STK600-RC100X-13 STK600-TQFP100 JTAG, PDI ATxmega128A1\_revD STK600-RC100X-13 STK600-TQFP100 JTAG, PDI ATxmega64A1 STK600-RC100X-13 STK600-TQFP100 JTAG, PDI

Table 1.2. AVR32 Studio device support

Device Routing card Socket card Programming modes Comment AT32UC3A0512 STK600-uC3-144 JTAG AT32UC3A0256 STK600-uC3-144 JTAG AT32UC3A0128 STK600-uC3-144 JTAG AT32UC3A1512 STK600-RCuC3A100-28 STK600-TQFP100 JTAG AT32UC3A1256 STK600-RCuC3A100-28 STK600-TQFP100 JTAG AT32UC3A1128 STK600-RCuC3A100-28 STK600-TQFP100 JTAG AT32UC3B0256 STK600-RCuC3B0-21 STK600-TQFP64-2 JTAG AT32UC3B0128 STK600-RCuC3B0-21 STK600-TQFP64-2 JTAG AT32UC3B064 STK600-RCuC3B0-21 STK600-TQFP64-2 JTAG AT32UC3B1256 STK600-RCuC3B48-27 STK600-TQFP48 JTAG AT32UC3B1128 STK600-RCuC3B48-27 STK600-TQFP48 JTAG AT32UC3B164 STK600-RCuC3B48-27 STK600-TQFP48 JTAG

**Chapter 2. Getting Started**

**Kit Contents**

The box contains:

●

STK600 starter kit evaluation board

●

Cables for STK600:

○

Two 10-wire cables for I/O ports and Parallel mode programming

○

One 6-wire cable for In-System Programming

○

Four 2-wire cable for UART and dataflash connections

●

USB cable

●

DC power cable

●

Atmel CD-ROM with datasheets and software

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Device board with an ATmega2560 AVR device

●

Two sets of screws and nuts, and one set of clips

**System Requirements**

Operating system/software requirements:

●

Windows 98/NT/2000/XP/XP x64/VISTA

●

Internet Explorer 6.0 or later (Latest version is recommended)

Minimum Recommended hardware:

●

Intel Pentium 200MHz processor or equivalent

●

800x600 screen (1024x768 recommended)

●

256 MB memory

●

100 MB free hard disk space

●

A free USB port on the host PC or on a self-powered USB hub

Optional: 10-15V DC/50mA power supply for powering STK600.

**Quick Start**

The STK600 starter kit is shipped with a device board with a ATmega2560 AVR microcontroller.

The STK600 can source power to the microcontroller trough the USB cable. Remember that the power available trough the USB cable is limited. If your application attaches several peripherals to the STK600, you should use an external power source connected to the DC input socket on STK600. The external power supply should be 9-15V DC with positive center connector.

The power switch turns the STK600 main power on and off. The red LED is lit when power is on, and the status LED will turn green. The green LED beside the VTG jumper indicates that the target voltage is present.

An example application is described in the Example Application section. To evaluate the example, copy the code into a new project in AVR Studio.

Build the project (menu: Build) and program the resulting hex file to the target (menu: Tools/Program AVR)

To program the code using ISP:

●

Connect a 6 lead flat cable between the two headers marked "ISP"

●

Adjust target voltage from the HW settings tab. The voltage must be set in the range 1.8-5-5V.

●

[1]

●

Select ISP on the main tab and set ISP frequency to 200Khz or lower

Program the application hex file from the program tab

To run the demo:

●

Connect PORTB to LEDS using a 10 lead flat cable

●

Connect PORTD to SWITCHES using a 10 lead flat cable

●

Press one of the switches. The leds will display a blinking pattern, dependant of which switch is pressed

Page STK600 User Guide

5 of 35 ●

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Page STK600 User Guide 6 of 35 [1] The ATmega2560 has a default fuse setting that makes it run on a 1MHz internal clock. The maximum ISP frequency is 1/4 of the target clock frequency.

**Connecting the Hardware**

The STK600 must be connected to a host PC with a USB cable. Connect the cable to a free USB port on the PC or on a USB hub. The USB port must be capable of supplying 500mA. If using a USB hub, make sure it has an external power supply.

Connect the other end of the USB cable to the USB connector on STK600 sitting next to the DC jack.

Optionally, if STK600 is to be connected to external hardware that consumes more than 300mA, an external DC power supply can be connected to the DC jack on STK600. The cable supplied with the kit can be used. Connect the center pin to the positive voltage and the cap to ground.

See Chapter 3, Target Socket System on how to set up the routing card and socket card.

**Example Application: Using LEDs and Switches**

Copy the code from this document into AVR Studio.

Tip: You need to adjust the include file specified at the 2nd line if you want to run the demo application on a device different from the ATmega2560.

;\*\*\*\*\* STK600 LEDS and SWITCH demonstration .include "m2560def.inc" .def Temp =r16 ; Temporary register .def Delay =r17 ; Delay variable 1 .def Delay2 =r18 ; Delay variable 2 ;\*\*\*\*\* Initialization RESET:

ser Temp out DDRB,Temp ; Set PORTB to output ;\*\*\*\* Test input/output LOOP:

out PORTB,temp ; Update LEDS sbis PIND,0x00 ; If (Port D, pin0 == 0) inc Temp ; then count LEDS one down sbis PIND,0x01 ; If (Port D, pin1 == 0) dec Temp ; then count LEDS one up sbis PIND,0x02 ; If (Port D, pin2 == 0) ror Temp ; then rotate LEDS one right sbis PIND,0x03 ; If (Port D, pin3 == 0) rol Temp ; then rotate LEDS one left sbis PIND,0x04 ; If (Port D, pin4 == 0) com Temp ; then invert all LEDS sbis PIND,0x05 ; If (Port D, pin5 == 0) neg Temp ; then invert all LEDS and add 1 sbis PIND,0x06 ; If (Port D, pin6 == 0) swap Temp ; then swap nibbles of LEDS ;\*\*\*\* Now wait a while to make LED changes visible. DLY: dec Delay brne DLY dec Delay2 brne DLY rjmp LOOP ; Repeat loop forever

**Chapter 3. Target Socket System**

**Socket System**

STK600 is designed to support all AVR devices (both AVR and AVR32) with internal flash memory. A system based on socket and routing cards is used to support different package types and pinouts on the STK600 board.

The picture below shows an STK600 with a mounted routing card and socket card.

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

**Socket Card and Routing Card**

A socket card is a general card that does not have any device specific hardware. E.g. a TQFP-64 socket card can be used for all devices that comes in a TQFP-64 package, regardless of pinout.

A routing card is a device specific card. It routes signals between the STK600 motherboard and the socket card. Note that several devices may use the same routing card if they share the same pinout.

A set of spring loaded connectors make the connection between the motherboard, routing card and socket card. Clips or screws hold the stack of cards together.

In addition to the socket and routing cards included in the kit, there are several add-on packs available to expand the part support for the STK600. See the Device Support page to get an overview over the different socket and routing cards.

**Selecting the correct routing and socket cards**

Page 7 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

Selecting the correct routing and socket card can be done by looking at the Device Support table.

Correct routing and socket card can also be found by just selecting the correct device in the STK600 programming dialog in AVR Studio. A notification will display the correct routing and socket card to use, unless the STK600 already has the correct cards mounted. More information on the programming dialog can be found in the Programming Dialog pages in the AVR Studio help.

Note that some of the devices has a part specific socket card, i.e. a routing card that has a socket. In this case there is only one card to install onto the motherboard.

**Mounting the cards**

Mounting the routing and socket cards can either be done by plastic clips or plastic screw/nuts. Both sets are included in the STK600 package. Install either the clips or the nuts to the motherboard depending on what solution you want to use.

**Using clips**

**Motherboard**

The clips should be installed from the bottom side of the STK600 motherboard. When properly installed, two plastic locking springs holds the clip in place.

**Routing card**

Align the clips with the white lines on the motherboard. The routing card could now be placed above the four clips. Make sure that the routing card has the correct orientation, i.e. the text should face upwards, and the white dot in the corner should match the one on the STK600. Press down the routing card (i.e. compress the spring loaded connector on the STK600) and turn the clip 45 degrees in the clockwise direction so that it aligns with the white line on the routing card. It is easiest to do two opposite clips before locking the two last.

Page 8 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

**Socket card**

Connecting the socket card is done in the same way as the routing card. Make sure that the clips align with the white line outside the clip holes on the routing card, then mount the socket card. The white spot on the socket card should align with the one on the routing card. Press down the socket card (i.e. compress the spring loaded connector on the socket card) and turn the clip 45 degrees in the clockwise direction until it aligns with the white line outside the clip hole. It is easiest to do two opposite clips before locking the two last.

**Note**

Always rotate the clips within the 45 degrees window. Do not turn them around, that could cause the routing card below to un-hock from the clip.

**Using screws and nuts**

**Motherboard**

Insert the nuts into the STK600 motherboard from the bottom side. When properly installed the two locking springs should hold the nut in place.

**Routing and socket card**

Place the routing card above the mother board, make sure that the whit spot in the corner matches the white spot on the motherboard. The small plastic taps on the ends of the spring loaded connectors should mate with the holes in the routing card. When the routing card is in the correct position, place the socket card onto the routing card with the white spot matching the one on the routing card. Make sure that the plastic taps on the connectors on the socket card mates with the routing card as well.

Insert the four screws, and tighten them firmly into the nuts.

Page 9 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

**Signal integrity**

STK600 is designed to support a wide range of devices with different packages and pinouts. Many compromises has been taken to make this possible with one motherboard. The signal integrity is not optimised due to this. STK600 is not a reference design in any way, but a kit that serves as socket programmer with some additional peripheral hardware to get started with the AVR. Serial communication at the highest frequencies may not work.

**Chapter 4. Hardware Description**

**STK600 Block Diagram**

**Target Voltage VTG**

Page 10 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

The VTG voltage is the supply voltage to the target AVR microcontroller. It is connected to the AVR's VCC pin. VTG can either be generated by STK600, or it can be supplied from an external source.

**On-board VTG source**

The on-board VTG source is set from AVR Studio. To use this source, the VTARGET jumper must be mounted. The on-board supply can be adjusted from 0.9 to 5.5V in steps of 0.1V.

VTG can also be set to 0V, but due to hardware limitations, onboard generated VTG can not be set between 0 and 0.9V.

Note: Always check the AVR datasheet for operating voltage range before adjusting VTG.

**Powering STK600 from USB**

When STK600 is powered through the USB cable approximately 300 mA can be delivered to the target section.

**Powering STK600 from an external DC source**

If an external DC jack input is used approximately 1A can be delivered to the target section.

**External VTG**

If the VTARGET jumper is removed, VTG must be supplied from an external source. Connect the external source to one of the VTG pins on any of the PORT headers. Always connect common ground (GND) when using an external VTG voltage.

When using an external source, the user must ensure that VTG is larger than any of the AREF voltages.

Note: The kit must always be powered when using an external VTARGET supply. If VTG voltage is supplied from an external source while the main power switch is in the off position the kit may become damaged.

**Status LEDs**

**VTARGET LED**

A green LED next to the VTARGET jumper will be lit when there is a voltage of 0.9V or higher available on the VTG net.

**STK600 Status LED**

If a short circuit is detected when using the on-board VTG supply, the STK600 status LED will blink red.

**Analog Reference Voltages**

The AVR's A/D converter needs a reference voltage to set its converting range. STK600 can supply two of these voltages, AREF0 and AREF1.

For all tinyAVR and megaAVR devices, only the AREF0 is connected trough the routing card, whilst for xmegaAVR devices, AREF0 is connected to the positive reference input and AREF1 is connected to the negative reference input on reference 0.

If the AREF0/AREF1 jumpers are mounted, the on-board Analog Reference Voltage sources are connected to the target AVR’s AREF pins. The On-board Analog Reference Voltages can be adjusted from the PC software in the range 0 to 5.5V, but not above VTARGET. The resolution and accuracy is 10mV.

The AREF0 and AREF1 generated voltages can also be connected to the analog comparator.

The target AVR’s AREF signals are accessible on the AUX header. The figure below shows the connection of the AREF signals, the target section and the AREF sources.

Page 11 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

**Page STK600 User Guide 12 of 35 Using external voltage reference**

When the AREF0/AREF1 jumper is disconnected, ADC reference voltage can be supplied from an external source, by connecting to the AREF0/AREF pins on the AUX header.

When using external an source for AREF, the user must control that VTARGET is at a higher voltage level than AREF. This can be controlled easily by reading the VTG value from AVR Studio before setting AREF.

NOTE: The AREF0 and AREF1 voltage which are visible in the PC software are the STK600 generated voltages. Externally applied AREF voltages cannot be read from AVRStudio.

**Using the internal voltage reference**

If the AVR's internal voltage reference is used, the AREF0/AREF1 jumper must be removed.

**Using Aref as analog input**

AREF0 and AREF1 can also be used as analog inputs to any of the ADC channels.

For tiny and mega devices, where only AREF0 is connected to the device via the routing card, AREF1 can be connected to a ADC channel by connecting a cable from AREF1 on the AUX header and to the port pin header corresponding to the adc channel. (check the device datasheet for which port pin header to connect to).

If the AVR's internal voltage reference, or an external voltage reference is used, the AREF0/AREF1 jumper must be removed. If this is the case, AREF0 or AREF1 can be used as analog inputs by connecting a cable from pin 1 on the AREF0/AREF1 header to the port pin header corresponding to the adc channel.

**AREF decoupling capacitor**

The routing card has a decoupling capacitor on AREF. This is marked with silkprint on the PCB. For some AVR devices the AREF pin is on a pin which also is part of the high-voltage programming interface. On these routing cards the aref capacitor is not mounted, as it would make it impossible to use the high-voltage programming interface. A capacitor can be soldered to achieve better noise performance. A typical value is 10nF. The footprint for the capacitor is SMD size 0603.

**Short circuit protection**

The internal AREF voltage generators have a short circuit protection. If the STK600 measures the AREF0/AREF1 to be 0.3V or more below the setpoint, AREF will be shut off. When this happens, the status LED will blink red. The AREF0 and AREF1 will also be shut down by the Master MCU if a short circuit is detected on VTarget (in addition to shutting down VTarget). In this case, the status LED will blink red.

**RESET Control**

The STK600 controls the RESET signal to the target AVR. Under normal operation, the RESET line is held in an inactive, high state (pull-up to VTG).

**The RESET jumper**

The RESET jumper connects the RESET pin on the target AVR to the STK600. When the RESET jumper is mounted, STK600 controls the RESET signal. When the RESET jumper is not mounted, the RESET signal is disconnected. This latter is useful for prototyping applications with an external reset system.

The RESET jumper must always be mounted when High-Voltage programming an AVR device. If using an external reset system, it must allow the reset line to be controlled by the STK600 during programming.

**The RESET button**

STK600 has a reset button that resets the target AVR when being pushed. The button has no function if the RESET jumper is not mounted.

**RESET signal on AUX header**

The target AVR's RESET signal is accessible on the AUX header. This pin can be used to apply an external RESET signal. When applying an external reset signal, the reset jumper must be removed.

**12V Programming Voltage**

During High-Voltage Programming, STK600 applies 12V to the AVR’s RESET line. Thus, an external reset circuit not capable of handling this must be disconnected before High-Voltage programming the AVR.

**External RESET de-coupling**

When connected to an external system, there is often an external pull-up resistor and a capacitor connected to the reset line. A typical reset connection is shown below.

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

If the external pull-up resistor is too strong (i.e << 4.7kOhm), STK600 may not be able to pull the RESET line low.

See also: Reset Line when programming an external target.

**Port Connectors**

All I/O port pins on the target AVR mounted on the STK600 are available on port pin connectors. These are labeled PORTA, PORTB, PORTC etc.

Depending to the AVR in use, different ports will be available.

The picture below shows PORTA. Here, all port pins PA0 to PA7 are available, in addition to the target voltage VTG and GND.

The other port connectors are identical, only with different signal names.

Cables can be mounted between the port connectors and the other peripherals on the board, or to external hardware.

**AUX port connector**

The AUX connector is located in the port connector area. The figure shows the pinout for the connector.

The following signals are available:

**AREF0**

Analog Reference voltage. This pin is connected to the AREF pin on devices having a single separate analog reference pin. For devices with two AREF pins, this pin in connected to the AREF+ pin. The AREF0 voltage is controlled from the PC software if the AREF0 jumper is mounted.

**AREF1**

Analog Reference voltage. This pin is connected to the AREF- pin on devices having two separate analog reference pin. The AREF1 voltage is controlled from the PC software if the AREF1 jumper is mounted.

**XTAL1**

The XTAL1 signal on the AVR is routed to this pin. If the CLOCK switch is set to the INT position, this pin can be used to apply an external clock signal.

**XTAL2**

The AVR's XTAL2 pin. If the CLOCK switch is set to the INT position, this pin can be used for external crystal with the XTAL1 pin.

**RESET**

The RESET pin on the AVR is available on this pin.

**32KHz**

A 32kHz clock signal is available on this pin. It can be connected to a device's TOSC1 pin in order to implement a real-time clock. Place a jumper between the 32KHz pin and the neighboring TOSC1 pin.

**TOSC2 and TOSC1**

For AVRs with a timer that can be clocked from an external low-frequency crystal, these two pins are available. The TOSC1 pin can easily be connected to the AUX port's 32KHz pin (32kHz clock signal) by a jumper.

Page 13 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

Page STK600 User Guide 14 of 35 AVR32 Pin Mapping

The AVR32 pin mapping is described in the tables below.

**STK600-RCuC3A100-28 (100-pin AT32UC3A devices)**

Device Pin Name STK600 Signal Name PA00 PA0 PA01 PA1 PA02 PA2 PA03 PA3 PA04 PA4 PA05 PA5 PA06 PA6 PA07 PA7 PA08 PB0 PA09 PB1 PA10 PB2 PA11 PB3, MISO PA12 PB4, MOSI PA13 PB5, SCK PA14 PB6 PA15 PB7 PA16 PC0 PA17 PC1 PA18 PC2 PA19 PC3 PA20 PC4 PA21 PC5 PA22 PC6 PA23 PC7 PA24 PD0 PA25 PD1 PA26 PD2 PA27 PD3 PA28 PD4 PA29 PD5 PA31 PD7 PB00 PE0 PB01 PE1 PB02 PE2 PB03 PE3 PB04 PE4 PB05 PE5 PB06 PE6 PB07 PE7 PB08 PF0 PB09 PF1 PB10 PF2 PB11 PF3 PB12 PF4 PB13 PF5 PB14 PF6 PB15 PF7 PB16 PG0, USB\_ID PB17 PG1, UVCON PB18 PG2 PB19 PG3 PB20 PG4 PB21 PG5 PB22 PG6

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Page STK600 User Guide 15 of 35 PB23 PG7 PB24 PH0 PB25 PH1 PB26 PH2 PB27 PH3 PB28 PH4 PB29 PH5 PB30 PH6 PB31 PH7 PC00 TOSC1 PC01 TOSC2 PC02 XTAL1 PC03 XTAL2 TDI TDI TDO TDO TMS TMS TCK TCK DP DP DN DN VBUS VBUS VDDIN VTG AREF0 AREF0 VDDANA VTG

**STK600-RCuC3B0-21 (64-pin AT32UC3B devices)**

Device Pin Name STK600 Signal Name PA3 PA3 PA4 PA4 PA5 PA5 PA6 PA6 PA7 PA7 PA8 PB0 PA9 PB1 PA10 PB2 PA11 PB3, TOSC1 PA12 PB4, TOSC2 PA13 PB5 PA14 PB6, MOSI PA15 PB7, SCK PA16 PC0 PA17 PC1 PA18 PC2, XTAL1 PA19 PC3, XTAL1 PA20 PC4 PA21 PC5 PA22 PC6 PA23 PC7 PA24 PD0 PA25 PD1, MISO PA26 PD2, USB\_ID PA27 PD3, UVCON PA28 PD4 PA29 PD5 PA30 PD6 PA31 PD7 PB0 PE0 PB1 PE1 PB2 PE2 PB3 PE3

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Page STK600 User Guide 16 of 35 PB4 PE4 PB5 PE5 PB6 PE6 PB7 PE7 PB8 PF0 PB9 PF1 PB10 PF2 PB11 PF3 RESET RESET TDI TDI TDO TDO TMS TMS TCK TCK DP DP DN DN VBUS VBUS VDDIN VTG AREF0 AREF0 VDDANA VTG

**STK600-RCuC3B48-27 (48-pin AT32UC3B devices)**

Device Pin Name STK600 Signal Name PA3 PA3 PA4 PA4 PA5 PA5 PA6 PA6 PA7 PA7 PA8 PB0 PA9 PB1 PA10 PB2 PA11 PB3, TOSC1 PA12 PB4, TOSC2 PA13 PB5 PA14 PB6, MOSI PA15 PB7, SCK PA16 PC0 PA17 PC1 PA18 PC2, XTAL1 PA19 PC3, XTAL1 PA20 PC4 PA21 PC5 PA22 PC6 PA23 PC7 PA24 PD0 PA25 PD1, MISO PA26 PD2, USB\_ID PA27 PD3, UVCON RESET RESET TDI TDI TDO TDO TMS TMS TCK TCK DP DP DN DN VBUS VBUS VDDIN VTG AREF0 AREF0 VDDANA VTG

**STK600-uC3-144 (144-pin AT32UC3A devices)**

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Page STK600 User Guide

17 of 35 Device Pin Name STK600 Signal Name PA0 PA0 PA1 PA1 PA2 PA2 PA3 PA3 PA4 PA4 PA5 PA5 PA6 PA6 PA7 PA7 PA8 PB0 PA9 PB1 PA10 PB2 PA11 PB3, MISO PA12 PB4, MOSI PA13 PB5, SCK PA14 PB6 PA15 PB7 PA16 PC0 PA17 PC1 PA18 PC2 PA19 PC3 PA20 PC4 PA21 PC5 PA22 PC6 PA23 PC7 PA24 PD0 PA25 PD1, PP0 PA26 PD2, PL4 PA27 PD3, PL5 PA28 PD4, PL6 PA29 PD5, PP2 PA30 PD6 PB0 PE0 PB1 PE1 PB2 PE2 PB3 PE3 PB4 PE4, PP3 PB5 PE5 PB6 PE6 PB7 PE7 PB8 PF0 PB9 PF1 PB10 PF2 PB11 PF3 PB12 PF4 PB13 PF5 PB14 PF6 PB15 PF7 PB16 PG0, USB\_ID PB17 PG1, PL7, UVCON PB18 PG2 PB19 PG3 PB20 PG4 PB21 PG5 PB22 PG6 PB23 PG7 PB24 PH0 PB25 PH1 PB26 PH2 PB27 PH3

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

Page STK600 User Guide 18 of 35 PB28 PH4 PB29 PH5 PB30 PH6 PB31 PH7, PP6 PX0 PJ0 PX1 PJ1 PX2 PJ2 PX3 PJ3 PX4 PJ4 PX5 PJ5 PX6 PJ6 PX7 PJ7 PX8 PK0 PX9 PK1 PX10 PK2 PX11 PK3 PX12 PK4 PX13 PK5 PX14 PK6 PX15 PK7 PX16 PL0 PX17 PL1 PX18 PL2 PX19 PL3 PX20 PM0 PX21 PM1 PX22 PM2 PX23 PM3 PX24 PM4 PX25 PM5 PX26 PM6 PX27 PM7 PX28 PN0 PX29 PN1 PX30 PN2 PX31 PN3 PX32 PN4 PX33 PN5 PX34 PN6 PX35 PN7 PX36 PP1 PX37 PP4 PX38 PP5 PX39 PP7 XIN0 XTAL1 XOUT0 XTAL2 XIN1 crystal socket XOUT1 crystal socket TOSC1 TOSC1 TOSC2 TOSC2 RESET RESET AREF0 AREF0 TDI TDI TDO TDO TMS TMS TCK TCK DP DP DN DN VBUS VBUS VDDANA VTG

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

VCCIN, VBOOST, VCC0, VCC2, VCC3, VCC5, VCC6, VCC7 VTG

**LEDs and Switches**

STK600 has eight LEDs and eight switches that can be connected to I/O pins on the AVR. The LEDS and SWITCHES connectors are found in the port connector area.

**LEDs**

The LEDs are labeled LED0 to LED7. The corresponding pins on the LEDS header have the same labels.

The LED hardware is shown in the figure below. The transistor circuit ensures the LED brightness is independent of target voltage.

To light one of the LEDs, the corresponding pin found on the LEDs header must be pulled to GND.

To control the LEDs from the AVR, connect a cable between the LEDS header and one of the PORT headers. Use a 10-wire cable to connect to all eight LEDs or a two-wire cable to control one or two LEDs.

Just like the PORT headers, the LEDS header has GND and VTG on pin 9 and 10. When using a 10-wire cable, make sure the pin one indication on the cable (red wire, triangular arraow pointing on pin one) aligns to pin 1 on both the LEDs header and PORT pin header.

The I/O port connected to LEDn will not source any significant current when LEDn is driven high, but it will sink a current of approximately 18mA when LEDn is pulled to GND.

**Switches**

The switches are labeled SW0 to SW7, and are available on the SWITCHES header. The switch hardware is shown below:

When pressing one of the switches, the corresponding SW pin on the SWITCHES header will be pulled low. When the switch is released, the switch's 10k pull-up will pull the line to VTG. The 150 Ohm resistor prevents a large current to flow to ground in case of wrong wiring.

Connect a cable between the SWITCHES header and one of the PORT headers. Use a 10-wire cable to connect to all eight switches or a two-wire cable to connect to one or two switches.

Note: On most AVR pins configured as input, you can enable an internal pull-up, removing the need for an external pull-up on the push button. In the STK600 design, an external 10K pull-up is present to give all users a logical '1' on SWn when the push button is not pressed, even if the internal pull-up is not enabled.

**Clock Settings**

Page 19 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

STK600 includes several clock options for the target AVR.

A switch selects between the following three options:

●

Programmable clock generator

●

Crystal oscillator (with socket for a crystal)

●

XTAL1 Pin tri-stated (to be used with the AVR's internal RC oscillator)

**Programmable Clock Generator**

The programmable clock generator is set from the PC software. The frequency can be set from 1.1kHz to 32MHz with 0.5% accuracy.

To use the programmable clock generator as clock source, set the CLOCK switch to EXT position.

**Crystal Oscillator**

The on-board crystal oscillator will work with ceramic resonators or crystals between 4 and 24MHz (AT-cut, fundamental, and parallel resonant crystals). Place a crystal in the crystal socket (located next to the PROGRAM button).

To use the crystal oscillator as a clock source, set the CLOCK switch to the XTAL position.

**XTAL1 Pin Tri-stated**

If the target AVR run on the internal oscillator, the XTAL1 pin can be disconnected from the clock sources on STK600.

To disconnect the XTAL1 pin, set the CLOCK switch to the INT position.

**Real Time Clock**

The STK600 also features a 32768 Hz oscillator, which can be used to make a real time clock. The output from the oscillator is available on the 32KHz pin on the AUX header. This clock can be routed to the TOSC1 pin on the target AVR by placing a jumper between the 32KHz and TOSC1 pin on the AUX header.

Page 20 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

See also Port Connectors for more information about the AUX header.

**Other Considerations**

**High-Voltage Programming**

When programming the target AVR in High-Voltage programming mode, the clock settings is overridden and the device is clocked directly from the STK600 controller. The clock selection switch can be set to any position.

**On-chip Crystal Oscillator**

In a real-life application where the crystal can be placed close to the AVR's XTAL1 and XTAL2 pins, there is no need for an external oscillator circuit. The long clock signal lines and socket system connectors on STK600 makes it difficult to drive a crystal with the on-chip oscillators. This is resolved by having a crystal oscillator on STK600. The oscillator is designed to operate over the full target voltage range.

**Shared XTAL1/Port Pin**

Some AVR devices have a XTAL1 pin which also can be used as a regular I/O port pin. The routing card for these devices will connect the device pin to both the XTAL1 net and a port pin header on the STK600. Hence, to use the pin as a I/O port the clock selection switch must be set to position INT to disconnect the clock drivers on STK600 from the pin.

**User RS232 Interface**

The STK600 includes a RS232 hardware that can be used for communication between the target AVR microcontroller in the socket and a PC serial port. STK600 has a 9-pin DSUB connector that can be connected to a PC with a straight serial cable (not a null modem cable).

To use the RS232 interface, the AVR's UART pins must be connected to the appropriate pins on the "RS232 SPARE" pin header. Use a 2-wire cable to connect the AVR's RXD and TXD pins to the pin header. The "RS232 SPARE" pin header is found in the target header section, while the DSUB marked "RS232" is located on the other end of the card.

Optionally one can connect the RTS (Request To Send) and CTS (Clear To Send) signals to two free I/O ports. The RTS and CTS signals are used for flow control. The connection is shown below.

Page 21 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

If the RTS and CTS lines are not controlled by the AVR, a jumper shorting the RTS and CTS pins on the "RS232 SPARE" header can resolve communication problems, if the PC side expects these handshake lines to be active.

**Flow control and RTS/CTS signaling**

Flow control is used to avoid data loss in transmission when one party is unavailable to receive data. When a DTE (such as a PC) wants to stop the data flow into it, it negates RTS. (read a negated "Request To Send" as "request NOT to send to me" (stop sending)). When the PC is ready for more bytes it asserts RTS and the flow of bytes to it can resume. Flow control signals are always sent in a direction opposite to the flow of bytes that is being controlled. DCE equipment (I.E. AVR) works the same way but sends the stop signal out the CTS pin (negated CTS: "(you are) NOT Cleared To Send").

**DataFlash Non-Volatile Memory**

An AT45DB041B 4Mbit DataFlash is included on the STK600 for non-volatile data storage. This is a high-density flash memory chip with SPI serial interface. Detailed datasheet of the DataFlash can be obtained from the Atmel CD-ROM or from the Atmel web site.

The dataflash can be connected to the I/O pins of the microcontroller sockets. The 4-pin header marked "DATAFLASH" can be used for connecting the SPI interface of the dataflash to the I/O pins on the target AVR microcontroller in the socket. 2-wire cables are included with STK600 for connecting the dataflash to the I/O pins. The connection of the I/O pins is shown on the picture below.

The block schematic of the dataflash connection is shown below, for connection of the dataflash to the AVR hardware SPI interface.

More information about how to use the DataFlash can be found on www.atmel.com.

**Expansion Connectors**

STK600 has two expansion connectors. All AVR I/O ports, programming signals, and control signals are routed to the expansion connectors. The expansion connectors allow easy prototyping of applications with STK600.

Page 22 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

The connectors to be used on an expansion board is manufactured by FCI and has P/N: 61082-101402LF. See also www.fciconnect.com for more information.

The connectors must be placed with exactly 119mm between center to center. The expansion board must have a maximum width of 55mm to avoid collision with components on the mainboard.

The pinout of the expansion connectors is shown in the table below.

Page STK600 User Guide 23 of 35 Table 4.5. EXPAND connector pinout

EXPAND0 EXPAND1 GND 1 2 VTG GND 1 2 VTG PA0 3 4 PA1 PJ0 3 4 PJ1 PA2 5 6 PA3 PJ2 5 6 PJ3 PA4 7 8 PA5 PJ4 7 8 PJ5 PA6 9 10 PA7 PJ6 9 10 PJ7 GND 1112 VTG GND 1112 VTG PB0 1314 PB1 PK0 1314 PK1 PB2 1516 PB3 PK2 1516 PK3 PB4 1718 PB5 PK4 1718 PK5 PB6 1920 PB7 PK6 1920 PK7 GND 2122 VTG GND 2122 VTG PC0 2324 PC1 PL0 2324 PL1 PC2 2526 PC3 PL2 2526 PL3 PC4 2728 PC5 PL4 2728 PL5 PC6 2930 PC7 PL6 2930 PL7 GND 3132 VTG GND 3132 VTG PD0 3334 PD1 PM0 3334 PM1 PD2 3536 PD3 PM2 3536 PM3 PD4 3738 PD5 PM4 3738 PM5 PD6 3940 PD7 PM6 3940 PM7 GND 4142 VTG GND 4142 VTG PE0 4344 PE1 PN0 4344 PN1 PE2 4546 PE3 PN2 4546 PN3 PE4 4748 PE5 PN4 4748 PN5 PE6 4950 PE7 PN6 4950 PN7 GND 5152 VTG GND 5152 VTG PF0 5354 PF1 PP0 5354 PP1 PF2 5556 PF3 PP2 5556 PP3 PF4 5758 PF5 PP4 5758 PP5 PF6 5960 PF7 PP6 5960 PP7 GND 6162 VTG GND 6162 VTG PG0 6364 PG1 PQ0 6364 PQ1 PG2 6566 PG3 PQ2 6566 PQ3 PG4 6768 PG5 PQ4 6768 PQ5 PG6 6970 PG7 PQ6 6970 PQ7

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20

GND 71 72 VTG GND 71 72 VEXT PH0 73 74 PH1 GND 73 74 VEXT PH2 7576 PH3 VCC 7576 GND PH4 7778 PH5 VCC 7778 GND PH6 79 80 PH7 PDATA0 79 80 PDATA1 GND 81 82 VTG PDATA2 81 82 PDATA3 XTAL1 83 84 AREF0 PDATA4 83 84 PDATA5 XTAL2 85 86 AREF1 PDATA6 85 86 PDATA7 GND 87 88 MOSI PCTRL0 87 88 PCTRL1 TOSC1 89 90 MISO PCTRL2 89 90 PCTRL3 TOSC2 91 92 SCK PCTRL4 91 92 PCTRL5 TGT\_RST 93 94 TDI PCTRL6 93 94 PCTRL7 VCC6 95 96 TDO GND 95 96 VCC3 GND 97 98 TMS B\_ID0 97 98 B\_ID1 VCC6 99 100 TCK B\_ID6 99 100 B\_ID7

**User USB connector**

STK600 has a USB connector that target AVR devices with USB interface can utilize. The connector is a mini-AB connector that supports On-the- go functionality. The routing card for the device connects the USB connector to the appropriate pins on the AVR.

**VBUS Generation**

When the target AVR is acting as an On-the-go master it must supply VBUS voltage to the USB device it is controlling. To do so, place a jumper on the pin header (J901) next to the USB connector. The FET is controlled by the UVCON signal, also routed to the target AVR.

When not using the VBUS generation feature, the jumper must be removed.

Note: VCC is 5.3V

Page 24 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

**CAN transceiver**

Controller Area Network (CAN) is a broadcast, differential serial bus standard typically used in the automotive industry. CAN features high immunity to electromechanical noise and arbitration free fixed priority.

STK600 features the ATA6660 CAN transceiver. A male DB9 connector and a 10 pin header is provided for bus connection

If a network termination is needed (CAN baudrate higher than 100 Kb/s), the 'TERM' jumper can be mounted to insert a 120 ohms resistor between CAN-H and CAN-L.

'SLOPE CTRL' is provided to adjust the CAN signal slopes and prevent unsymmetrical transients on the bus lines. The center pin on 'SLOPE CTRL' is connected to the ATA6660 RS pin. This must be held below 0.87·VTG which is the standby threshold voltage for AT6660.

Mount a jumper to either side of 'SLOPE CTRL' to prevent AT6660 from going to standby.

The CAN transceiver is connected to the MCU trough the two-pin (rx and tx) 'CAN' header near the switches on STK600. The target MCU can be any AVR (bit banging or USART), but is more typically one of the AT90CAN series which support the CAN protocol in hardware.

**LIN transceiver**

Local Interconnect Network (LIN) is a broadcast serial network comprising one master and many (up to 16) slaves. The LIN bus is typically used in the automotive industry as smaller and less expensive sub-network of a CAN bus to integrate intelligent sensor devices or actuators.

STK600 features the ATA6661 LIN transceiver. a three-pin header serves to connect to the bus. With the ATA6661 an AVR on the STK600 can implement a LIN master or a LIN slave.

Page 25 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

The 'M LIN' jumper provides the master node pull-up, required if the application running on STK600 is the LIN bus master.

The 3-pin LIN connector must provide V-battery ('BAT') 12V>BAT>5V, and GND. 'BAT' must be supplied from an external source. For further reference, see the ATA6661 datasheet.

The LIN transceiver is connected to the MCU trough the six-pin 'LIN' header near the switches on STK600. The target MCU will usually implement the LIN protocol in software trough a USART interface. The 'NSLP' pin must be actively driven high to keep the ATA6661 from sleep mode.

Note: Due to ATA6661 design, it is mandatory to enable the internal pull-up on PD2 (RxLIN) when LIN is used (c.f. AT90CAN128 Datasheet, section “I/O Ports”).

**Miscellaneous**

STK600 has two push buttons and three LEDS for special functions and status indication. The following section explains these features. The figure above shows the placement of these functions.

**PROGRAM Push Button**

Future versions of AVR Studio may upgrade the master MCU on STK600. AVR Studio will then detect old software versions of STK600 and update the Flash program memory of the master MCU. To do this the user is required to push the PROGRAM button when powering on STK600. AVR studio issues instructions on how to perform the upgrade during the upgrade process.

**Main Power LED**

The red power LED is directly connected to the STK600 main power supply. The power LED is always lit when power is applied to STK600.

**Target Power LED**

Page 26 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

The target power LED is lit when voltage applied to the target AVR device is 0.9V or higher.

**Status LED**

●

ORANGE: Busy programming

●

ORANGE/RED Blinking: Upgrade mode

●

RED: No board detected.

●

GREEN: READY

●

ORANGE Blinking: Wrong combination of routing and socket card.

●

RED Blinking: VTarget or Aref short circuited.

●

RED Blinking high frequency: To much current drawn from supply. If powered from USB, try to connect an external supply to DC jack.

During programming the LED has orange color. When the target AVR device is successfully programmed the LED will turn green

**Chapter 5. Programming**

**ISP Programming**

In-System Programming uses the AVR internal SPI (Serial Peripheral Interface) to download code into the flash and EEPROM memory of the AVR. ISP programming requires only VCC , GND, RESET and 3 signal lines for programming. No high voltage signals are required. The ISP programmer can program both the internal flash and EEPROM, fuses, lockbits and calibration bytes.

Note that the ISP frequency (SCK) must be less than 1/4 of the target clock. The ISP frequency is set by the STK600 programming dialog in AVR Studio.

Note that if ISP programming will NOT work if one or more of the following cases are true:

●

SPIEN, SPI Enable fuse is un-programming

●

RSTDISBL, Reset Disable fuse is programmed (for devices with this fuse)

●

DWEN, DebugWIRE Enable fuse is programmed (for devices with this fuse)

Refer to the AVR datasheet for information about the fuses.

Use High-Voltage programming to re-enable the ISP interface from the situation listed above. Either HVPP or HVSP depending on what is supported by the AVR.

**Hardware setup for On-board programming**

1. Mount the routing and socket card and the target device. See the Socket System section on how to do this.

2. Connect a 6-wire cable between the two 6-pins ISP headers on the STK600. See picture below.

3. Ensure that the VTARGET jumper is mounted, and that the voltage is the within the operating range for the target device.

Page 27 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

See the Programming Dialog pages in the AVR Studio help file for information on the STK600 programming dialog.

The pinout of the 6 and 10-pins ISP header is shown below:

It is not necessary to remove the ISP cable while running a program in the AVR. The port pins used for ISP programming can be used for other purposes in your program.

See also: In-System Programming of an External Target System

**AREF**

The AREF0 jumper must be removed before programming of devices that have AREF on a pin used by the serial programming interface.

Devices that are affected by this uses these routing cards:

●

STK600-RC008T-2

**Parallel High Voltage Programming**

**Hardware setup for On-board programming**

Follow the steps below to do Parallel High Voltage Programming. Note that this interface is only intended for use on-board STK600.

1. Mount the routing and socket card and the target device. See the Socket System section on how to do this.

2. Use the two 10-wire cables supplied with the STK600 to connect the PROG DATA and the PROG CTRL to the target device, as shown in the

picture below.

3. Mount both the VTARGET jumper and the RESET jumper

4. Ensure that VTarget is between 4.5V and 5.5V

Page 28 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

See the Programming Dialog pages in the AVR Studio help file for information on the STK600 programming dialog.

Note that the AREF jumper must be removed before programming of devices that have AREF on a pin used by the High Voltage programming interface.

Devices that are affected by this uses these routing cards:

●

STK600-RC008T-7

●

STK600-RC020T-8

●

STK600-RC014T-12

●

STK600-RC020T-23

**Serial High Voltage Programming**

The AVRs with low pin count have too few pins to use parallel communication during High-Voltage programming. They use serial communication instead. This means that less signals have to be routed. Note that this interface is only intended for use on-board STK600.

**Hardware setup for On-board programming**

1. Mount the routing and socket card and the target device. See the Socket System section on how to do this.

2. Use the a 10-wire cable supplied with the STK600 to connect the PROG DATA to the target device, shown in the picture below.

3. Note for ATtiny24/44/84 a cable on PROG CTRL is required as well, as for the Parallel High Voltage Programming.

4. Mount both the VTARGET jumper and the RESET jumper.

5. Ensure that VTarget is between 4.5V and 5.5V before programming.

See the Programming Dialog pages in the AVR Studio help file for information on the STK600 programming dialog.

Page 29 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

**JTAG Programming**

AVR devices with a JTAG port can be programmed through this interface.

**Hardware setup for On-board programming**

1. Mount the routing and socket card and the target device. See the Socket System section on how to do this.

2. Connect a 10-wire cable between the two 10-pins JTAG headers on the STK600. See picture below.

3. Ensure that the VTARGET jumper is mounted, and that the voltage is the within the operating range for the target device.

See the Programming Dialog pages in the AVR Studio help file or the AVR32 Studio help for information on how to program the device using JTAG.

The pinout of the JTAG header is shown below:

Page 30 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

See also: In-System Programming of an External Target System

**PDI Programming**

All ATxmega devices has the new PDI programming and debugging interface. It can, in-system, download code into the flash application and boot memories, EEPROM memory, fuses, lockbits and signature information.

The PDI interface requires two of the device's pins, PDI\_DATA and PDI\_CLOCK. On STK600, they are found on the ISP/PDI connector.

**Hardware setup for On-board programming**

1. Mount the routing and socket card and the target device. See the Socket System section on how to do this.

2. Connect a 6-wire cable between the two 6-pins ISP/PDI headers on the STK600. See picture below.

3. Ensure that the VTARGET jumper is mounted, and that the voltage is the within the operating range for the target device.

The pinout of the 6-pins ISP/PDI header when in PDI mode is shown below:

**In-System Programming of an External Target System**

Page 31 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

The STK600 can be used as a programmer to program AVR devices in other applications. There are two different ISP connector pin-outs available, a 6-pin and a 10-pin version. Both are supported by STK600. The 6-pin header is a combined ISP and PDI connector. In addition STK600 can be used as a JTAG programmer for AVR devices with a JTAG interface.

ISP connector pinouts

JTAG connector pinout

PDI connector pinout

Select the device to be programmed in the same way as programming a device on STK600. The VCC of the target application is detected by STK600 and signals are converted into voltage levels suitable for the target system.

Note: If the other application has its own power supply, the VTARGET jumper must be removed before connecting STK600 to the other application. STK600 may be damaged if the VTARGET jumper is not removed.

ISP or PDI programming an external target using the 6-pin connector

ISP programming an external target using the 10-pin connector

Page 32 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

JTAG programming an external target

See the Programming Dialog pages in the AVR Studio help file for information on the STK600 programming dialog.

See also the Target Voltage section.

**Reset Line**

The Reset line on any target board connected to STK600 should have a pull-up resistor. This pull-up should not be stronger than 2.2k ohm (i.e. it should not be below 2.2kohm). If the pull-up resistor on the reset line is to strong, the short circuit protection will trigger when the reset is forced low by the STK600. Any de-coupling capacitor should not be larger than 10uF.

**Chapter 6. Troubleshooting and Support**

**Troubleshooting Guide**

Page STK600 User Guide

33 of 35 Table 6.1.

**Problem Reason Solution**

The main power LED is dark

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20 The power switch is OFF Turn on the power switch No power source is connected to STK600

Do one of the following:

●

Connect a USB cable between STK600 and a PC. Make sure the PC is turned on.

●

Connect a DC power cable to STK600. Note: The DC-jack must have a center pin with positive polarity.

The preprogrammed example code does not toggle the LEDS

There is no AVR device in the socket

Plug the AVR device into the right socket (see ISP Programming and Parallel High Voltage Programming). The LEDS are not connected to the I/O ports

Connect the LEDS header to the PORTD header, and SWITCHES header to PORTB header (see LEDs and Switches).

No target voltage

Ensure that the VTG jumper is mounted and that VTG is set above 1.8V from AVRStudio The flash memory is erased Reprogram the AVR device

The AVR device can not be programmed using SPI.

The wrong AVR orientation

device is inserted with

Check that the notch on the AVR socket matches the notch on the AVR device.

The ISP/PDI headers are not connected

Connect the 6-pin flexible cable between the two 6-pin ISP/PDI headers.

The VTARGET voltage is too low

Check the AVR datasheet for minimum operating voltage

The memory lock bits are programmed

Erase the memory before programming

The SPI enable fuse is un- programmed

Program the SPIEN fuse using Parallel High Voltage Programming or Serial High Voltage Programming Reset disable fuse is set. Check reset disable fuse.

SPI frequency is too high.

Check STK600 SPI frequency and make sure it is lower than target clock divided by 4. CKDIV fuse is set Reduce ISP programming speed External pullup resistor on reset line too low.

Ensure that external pullup resistor is ≥4.7kOhm.

AREF0 jumper mounted

For some devices the AREF0 is connected to a pin used for the ISP interface. The AREF0 jumper must therefore be removed to do serial programming of these devices. See the ISP Programming section for which routingcards this applies.

The VTARGET voltage is too

Check the AVR datasheet for minimum operating voltage

Page STK600 User Guide

34 of 35 low The JTAG headers are not

Connect the 10-pin flexible cable between the two 10-pin JTAG headers. See

The AVR device can not be programmed using JTAG.

connected

JTAG Programming. The JTAG enable fuse is un-

Program the JTAGEN fuse using Parallel High Voltage Programming or Serial programmed

High Voltage Programming The memory lock bits are programmed

Erase the memory before programming

The AVR device can not be programmed using high- voltage programming.

file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm

2008-06-20 The low VTARGET voltage is too

Ensure that the voltage is at least 4.5V.

The high-voltage programming headers are not connected

Mount cables between the programming headers. See Parallel High Voltage Programming or Serial High Voltage Programming The reset jumper is not mounted Mount the reset jumper The I/O ports are connected to peripheral circuitry (leds, switches, etc.)

Remove all peripheral connections from the I/O ports belonging to the high- voltage programming interface.

The memory lock bits are programmed

Erase the memory before programming

AREF0 jumper mounted

For some devices the AREF0 is connected to a pin used for the parallel programming interface. The AREF0 jumper must therefore be removed to do parallel programming of these devices. See the Parallel High Voltage Programming section for which routingcards this applies.

AVR studio can not connect to STK600

USB cable is not connected, or power is off

Connect USB cable

Firmware is in an hang-up state Toggle power on STK600.

The status LED is blinking orange

Wrong combination of routing and socketcard, or card removed when the kits is powered.

Check the device support file for routing and socketcard combination. Allways turn of kit power before removing or mounting routing and socketcards.

The status LED is blinking red

There is a short circuit on VTarget or AREF

Resolve the short circuit.

The status LED is blinking red

To supply

much current drawn from

If the kit is powered from USB, try connecting a external power to the DC jack.

The LEDs don't work (running from external VTarget.)

STK600 must be powered for LEDs to work.

Supply power to STK600 and turn it on.

**Technical Support**

For technical support please contact avr@atmel.com. When requesting technical support for STK600 please include the following information:

●

Version number of AVR Studio. This can be found in AVR studio menu "Help/About"

●

PC processor type and speed

●

PC operating system and version

●

What target AVR device is used (Complete part number)

●

Programming voltage

●

Jumper settings

●

A detailed description of the problem

**Manual Firmware Upgrade**

If an automatic firmware upgrade fails, or for some other reason connection to STK600 cannot be established, a manual firmware upgrade may resolve the problem.

Before starting this procedure, make sure the latest AVR Studio release is installed on the computer.

1. Turn off STK600 and connect it to the PC using the USB cable

2. Press and hold the PROGRAM button when turning on the STK600 power switch. The status LED will flash red and orange, indicating

upgrade mode.

3. In AVR Studio, go to the Tools menu and select STK600 Upgrade

4. The Atmel STK600 Upgrade program will start. If connection is established, the Status will show "STK600 present"

5. Press the Start Upgrade button. The upgrade program will upgrade STK600's firmware.

6. When complete, a message box will show if the upgrade was successful or not. Cycle power on STK600.

7. If the upgrade was successful, the status led will now be green. Try to connect to the starter kit with the programming tool in AVR Studio.

Page 35 of 35 STK600 User Guide

2008-06-20 file://C:\Documents and Settings\PAWEL\Ustawienia lokalne\Temp\~hhE369.htm