AVR602: Using the ATtinyX3U Top Module

Features

- Complete Starter Kit for ATtiny43U
  - Top Module Easily Connects to the STK®600
- Fully Configured Boost Circuitry – No Additional Components Required
  - Variable Voltage Source of STK600 Performs as Virtual Battery
- Supports Both SOIC and MLF Packages
- Serial and Parallel Programming Supported
- Zero Insertion Force Socket for Programming Discrete SOIC Devices

1 What is included in the ATSTK600-TinyX3U

The following item is included in the package:

- ATtinyX3U card, with ATtiny43U SOIC or MLF device mounted

Figure 1-1. ATtinyX3U Top Module (Revision C) for STK600
2 Preparing the Equipment

Use the figures below as a reference and follow instructions in sections 2.1 and 2.2.

Figure 2-1. STK600 with ATtinyX3U Assembled
2.1 Hardware

See Figure 2-1 and Figure 2-2 for guidance. Make the following preparations before proceeding:

- Release the plastic screws on the STK600 and mount the ATtinyX3U Top Module. Using the plastic screws, fasten and secure the top module to the board.
- On the ATtinyX3U, mount jumper J3. This will configure the device to work without the integrated boost regulator.
- On STK600, set the clock switch to position ‘INT’. Then short jumpers RESET, AREF0 and AREF1.

2.2 Firmware

Make the following preparations before proceeding:

- Make sure the required version of AVR Studio is installed on your computer.
- Open AVR Studio from the Start menu of Windows®. Click ‘Cancel’ to skip the welcome window. In the ‘Help’ menu click ‘AVR Tools User Guide’ to open the online help files. Browse to section ‘STK600’ for detailed instructions on the hardware.
3 Setting Up the Power Source

The ATtiny43U can be powered either from the voltage source of the STK600 or from the on-chip boost regulator.

3.1 Powering the Device Directly from the Voltage Source of STK600

How to set up for use with STK600 as voltage source:

- Make sure the STK600 power switch is in ‘OFF’ -position
- On STK600, short jumper VTARGET
- On the ATtinyX3U, open jumper J8 and short jumper J3
- Using the USB cable connect the STK600 to the computer, then turn the STK600 power switch to ‘ON’ -position. If asked, proceed to install a USB driver on the computer. Choose the recommended option and let Windows find a driver automatically
- In AVR Studio, open the 'Tools' menu, select 'Program AVR' and click on 'Connect'. Choose STK600 as 'Platform' and USB as 'Port' option. Then click ‘Connect’
- When connection has been made, go to the 'HW Settings' tab in the programming window. In the ‘Voltages’ section, use the scroll bar to set ‘VTarget’ to a value between 1.8V and 5.5V and then click the 'Write' button. See Figure 2-3

The ATtiny43U chip on the ATtinyX3U Top Module should now be powered and start to run.

3.2 Using the STK600 as Supply Voltage for the On-Chip Boost Regulator

How to set up for using the on-chip boost regulator with STK600 as voltage source:

- Using the procedure outlined in section 3.1 change VTarget to 0V
- Switch off the STK600, then open jumper VTARGET
On the ATtinyX3U, open jumper J3 and close jumper J8. Then use a wire to strap pin 1 of jumper J3 and pin 1 of VTARGET. Note polarity! See Figure 2-2 and Figure 2-4 for guidance.

Figure 2-4. Strapping pin 1 (left) of J3 to pin 1 (right) of VTARGET

• Turn on the STK600
• In AVR Studio, open the 'Tools' menu, select 'Program AVR' and click on 'Connect'. Choose STK600 as 'Platform' and USB as 'Port' option. Click 'Connect'. Then browse to the 'HW Settings' tab, set VTarget to 1.5 and click the 'Write' button

NOTE WELL
Make sure strap between J3 and VTARGET is properly connected. Wrong polarity may cause permanent damage! See Figure 2-4 for guidance.

The application software on the ATtiny43U should now start to run the programmed code. In the ‘HW Settings’ tab of AVR Studio, experiment with VTarget values from 1.8V and 0.9V. STK600 can supply VTarget voltages of 0.9V and above.

Notes
• Pin 1 is the positive terminal and pin 2 is the negative terminal of jumper J3
• Refer to the ATtiny43U data sheets for voltages where the device will start up and shut down
• In this configuration, pushing the 'Read' button of section 'Voltages' in the ‘HW Settings’ returns the voltage at pin 2 of VTARGET in the field ‘VTarget’. In this case the reading corresponds to the boost regulated voltage on pin VCC of ATtiny43U, not the input voltage to the boost regulator
3.3 Using an External Source to Power the Chip from the On-Chip Boost Regulator

The hardware can be configured for ATtiny43U boost regulator supply voltage to be provided from an external source, such as a single-cell battery. To do so, make the following adjustments:

- On STK600, open jumper VTARGET or remove strap if present
- On ATtinyX3U, open jumper J3 and close jumper J8
- Power on the STK600
- Connect external voltage source to jumper J3. Note polarity! Pin 1 is BATS and pin 2 is GND

**Note**
The STK600 must always be powered on when using an external voltage source.

4 Programming ATtiny43U

For information on device programming options, see the device data sheet.

4.1 Programming the Soldered Device

The soldered device can be programmed in both serial and parallel mode, as described in the following sections.

4.1.1 Serial Mode

How to set up serial programming of the soldered device:

- Choose the power supply option for the ATtiny43U and configure it according to one of the schemes described in “Setting Up the Power Source” on page 4
- On the STK600, use a 6-pin ribbon cable to wire the connector marked ISP/PDI to the 6-pin connector next to it. See Figure 4-1 for guidance. Note polarity

**Figure 4-1.** Serial programming mode, soldered device
• In AVR Studio, open the programming window (Tools – Program AVR – Connect). On the ‘Main’ tab set ‘Programming Mode and Target Settings’ to ‘ISP mode’. Then click ‘Settings’ in ‘Programming Mode and Target Settings’ and make sure ‘ISP Clock’ is below ¼ of the device clock frequency.

Check connections by clicking ‘Read Signature’ in section ‘Device and Signature Bytes’ on tab ‘Main’. The correct signature for ATtiny43U is 0x1E 0x92 0x0C.

4.1.2 Parallel Mode

How to set up the hardware for parallel programming of soldered device:

• Choose the power supply option for the ATtiny43U and configure it according to one of the schemes described in “Setting Up the Power Source” on page 4.

• On the STK600, use one 10-pin ribbon cable to wire the connector marked PROGDATA to the header next to it. Then use another 10-pin ribbon cable to wire connector PROGCTRL to the header next to it. See Figure 4-2 for guidance. Note polarity.

Figure 4-2. Parallel programming mode, soldered device.

4.2 Programming a Device in the ZIF Socket

The ATtinyX3U Top Module includes a zero insertion force (ZIF) socket for programming purposes. The socket is wired to the four connectors J9 (PROG DATA),
J10 (PROG CTRL), J11 (ISP) and J12 (AUX). No other parts of the board are connected to the socket.

The device in the socket can be programmed in both serial and parallel mode, as described in the following sections.

### 4.2.1 Serial Mode

How to set up the hardware for serial programming of a socketed device:

- Choose the power supply option for the ATtiny43U and configure it according to one of the schemes described in “Setting Up the Power Source” on page 4
- On the STK600, use a 6-pin ribbon cable to wire the connector marked ISP/PDI to the 6-pin ISP connector on the ATtinyX3U Top Module. See Figure 4-3 for guidance. Note polarity

![Figure 4-3. Serial programming mode, socketed device](image)

- In AVR Studio, open the programming window (Tools – Program AVR – Connect). On the ‘Main’ tab set ‘Programming Mode and Target Settings’ to ‘ISP mode’. Then click ‘Settings’ in ‘Programming Mode and Target Settings’ and make sure ‘ISP Clock’ is below ¼ of the device clock frequency
- Check connections by clicking ‘Read Signature’ in section ‘Device and Signature Bytes’ on tab ‘Main’. The correct signature for ATtiny43U is 0x1E 0x92 0x0C.

### 4.2.2 Parallel Mode

How to set up the hardware for parallel programming of socketed device:

- Choose the power supply option for the ATtiny43U and configure it according to one of the schemes described in “Setting Up the Power Source” on page 4
- On the STK600, use one 10-pin ribbon cable to wire the connector marked PROGCTRL to the connector marked PROG CTRL on the ATtinyX3U Top
Module. Use another 10-pin ribbon cable to connect PROGDATA on STK600 to PROG DATA on ATtinyX3U. See Figure 4-4 for guidance. Note polarity

Figure 4-4. Parallel programming mode, socketed device

- In AVR Studio, open the programming window (Tools – Program AVR – Connect). On the ‘Main’ tab set ‘Programming Mode and Target Settings’ to ‘PP/HVSP mode’.

Check connections by clicking ‘Read Signature’ in section ‘Device and Signature Bytes’ on tab ‘Main’. The correct signature for ATtiny43U is 0x1E 0x92 0x0C.

5 Known Issues

This is a list of known issues. Some or all issues may be fixed in future revisions of hardware or software. To submit an issue that is not listed here, please contact technical support at avr@atmel.com.

Currently, there are no known issues.
6 Schematic
Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATMEL’S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL’S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel’s products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2009 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, AVR®, AVR Studio®, STK® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® is a registered trademark of Microsoft Corporation in US and or other countries. Other terms and product names may be trademarks of others.