## Table of Contents

### Section 1
Introduction ........................................................................................................... 1-1  
1.1 Overview ......................................................................................................... 1-1  

### Section 2
Using the ATAVRAUTO200........................................................................ 2-3  
2.1 Overview ......................................................................................................... 2-3  
2.2 Power Supply .................................................................................................. 2-4  
2.3 Oscillator Sources ........................................................................................... 2-4  
2.4 On-board Resources ......................................................................................... 2-4  
2.5 In-System Programming ................................................................................ 2-8  
2.6 Debugging ........................................................................................................ 2-11  

### Section 3
Technical Specifications ...................................................................................... 3-13  

### Section 4
Technical Support .................................................................................................. 4-14  

### Section 5
Complete Schematics ............................................................................................. 5-17  

Section 1

Introduction

Congratulations on your purchase of the ATAVRAUTO200 board. This board includes all elements necessary for designers to quickly develop code related to LIN communication node implementing the ATmega88 and for prototyping and testing of new designs.

1.1 Overview

This document describes the ATAVRAUTO200 dedicated to the ATmega88 AVR micro controllers. This board is designed to allow an easy evaluation of the product using demonstration firmware.

To increase its demonstrative capabilities, this stand alone board has numerous on-board resources (motor relay, motor FET, hall sensor inputs, current measurements, power supply measurement, LIN, push buttons).

Figure 1-1. ATAVRAUTO200
The ATAVRAUTO200 provides the following features:

- **ATmega88 QFN32**
- **AVR Studio® software interface**,\(^{(1)}\)
- **Power supply**
  - Regulated 5V
  - From LIN connector (LIN network power supply)
- **JTAG connector**:
  - for on-chip In Situ Programming (ISP)
  - for on-chip debugging using JTAG ICE
- **DC Motor connector**
  - DC Motor power supply output
  - Hall effect sensor(s) power supply and input(s)
- **Serial interface**:
  - 1 LIN interface 1.3 and 2.0 compliant (firmware library available on the ATMEL website for LIN 1.3).
- **On-board resources**:
  - 1 LIN transceiver with internal regulator
  - Relay for DC motor control
  - Shunt Resistor for motor current measurement
  - Speed/Position measurement Inputs
  - Power supply measurement
- **System clock**:
  - Internal RC oscillator
- **Dimension**: 45 mm x 45 mm

\(^{(1)}\) The ATmega88 is supported by AVR Studio, version 4.12 or higher. For up-to-date information on this and other AVR tool products, please consult our web site. The newest version of AVR Studio, AVR tools and this user guide can be found in the AVR section of the Atmel web site, http://www.atmel.com.
Section 2
Using the ATAVRAUTO200

2.1 Overview

Figure 2-1. Board Overview

Figure 2-2. Block Diagram
2.2 Power Supply

The on-board power supply circuitry is supplied through the LIN connector.

2.2.1 LIN powered

The LIN connector power line is used to provide VBAT to the ATAVRAUTO200 LIN transceiver.

A LIN network has to be connected to have your LIN interface function (Input supply from 8 up to 18V DC, see Figure 2-3 on page 5).

2.3 Oscillator Sources

The ATAVRAUTO200 board allows only one oscillator source:

- Internal RC oscillator (Default configuration).

**Note:** The “Divide by 8” Fuse is configured by default. The first step in the demonstration application is to clear the prescaler to have the internal RC oscillator running at 8MHz:

```
CLKPR = (1<<CLKPCE);  // Clear Prescaler
CLKPR = 0;
```

2.3.1 Internal RC oscillator

A LIN Slave node with a run-time oscillator calibration can be used with the internal RC oscillator.

At ambient temperature and normal Vcc, the internal oscillator is precise enough to be compliant with LIN 1.3 and 2.0 specifications. For wider temperature and/or power ranges, a run-time calibration of the internal RC oscillator can be used as explained in the application note AVR140: “ATMega48/88/168 family run-time calibration of the internal RC oscillator” available on the Atmel website.

2.4 On-board Resources

2.4.1 LIN & Power supply

The LIN screwed connector allows the user to select his own connector.

**Note:** The LIN power supply input is reverse voltage protected.

LIN transceiver control is realized by the microcontroller. All modes depend on microcontroller’s ports configuration.
Table 2-1. LIN resources

<table>
<thead>
<tr>
<th>Function</th>
<th>Port</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIN_NSLP</td>
<td>PD2</td>
<td>Low</td>
<td>LIN transceiver in Sleep mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>LIN transceiver in normal mode</td>
</tr>
<tr>
<td>NRES_LIN</td>
<td>PC6</td>
<td>Low</td>
<td>Perform MCU reset when NISP Jumper is inserted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>No Action</td>
</tr>
</tbody>
</table>

Figure 2-3. LIN transceiver and power supply

Note: The LIN transceiver undervoltage protection can be disabled by removing the NISP jumper.

Note: The NISP jumper has to be removed when programming.

2.4.2 Power supply measurement

The voltage measurement is realized with a bridge of resistors. The read value is 0.281 of the LIN power supply (47 KΩ / (47 KΩ + 120 KΩ)). Input voltage on channel 1 of the ADC is limited to 5.1V by a zener diode. This will give a voltage reading range from 0 to 18.1V with Vcc as reference.

The power supply measurement can be performed using the A/D converter. See the ATmega48/88/168 datasheet for how to use the ADC. The input voltage value \( V_{IN} \) is calculated with the following expression:

\[
V_{IN} = 3.55 \times V_{ADC1}
\]

- Where: \( V_{IN} \) = Input voltage value (V)
- \( V_{ADC1} \) = Voltage value on ADC-1 input (V)
2.4.3 Motor relay

DC Motor can be operated through a relay. It is supplied with Vbat, -Vbat or 0V.

- The relay allows the motor to be operated in two rotating directions, or to be stopped.

<table>
<thead>
<tr>
<th>Function</th>
<th>Port</th>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mot_A</td>
<td>PB1</td>
<td>Low</td>
<td>Relay coil1 OFF (Normaly closed switch activated)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>Relay coil1 ON (Normaly opened switch activated)</td>
</tr>
<tr>
<td>Mot_B</td>
<td>PB2</td>
<td>Low</td>
<td>Relay coil2 OFF (Normaly closed switch activated)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High</td>
<td>Relay coil2 ON (Normaly opened switch activated)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mot_A</th>
<th>Mot_B</th>
<th>Motor Supply</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>0V</td>
<td>Motor stopped</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>-Vbat</td>
<td>Motor running (Direction B)</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>+Vbat</td>
<td>Motor running (DirectionA)</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>0V</td>
<td>Motor stopped</td>
</tr>
</tbody>
</table>

Figure 2-4. Power supply measurement through ADC1

Figure 2-5. Motor on board command schematics
2.4.4 Current measurement

Motor current is measured using a shunt resistor. External differential amplifier (on board) is connected to ADC to measure shunt resistor voltage.

Amplifier output (current image voltage) is connected to
- ADC0 pin for current acquisition
- AIN1 pin to detect max current peak (compared to AIN0 through internal analog comparator)

The current measurement (I) can be performed using the A/D converter. See the ATmega48/88/168 datasheet for how to use the ADC. The input voltage value ($V_{ADC-0}$) is calculated with the following expression:

\[
\begin{align*}
    (V_{ADC-0} &= Gain \times V_{shunt} = Gain \times R_{shunt} \times I = 30,16 \times 0.005 \times I) \\
    V_{ADC-0} &= 0.151 \times I
    \end{align*}
\]

Analog comparator allows peak current detection. It provides interrupts on analog comparator output change. See the ATmega48/88/168 datasheet for how to use the Analog comparator. Comparison voltage is determined for a 12A peak which leads to:
- $V_{AIN0} = 1.5V$

*Figure 2-6. Current Acquisition chain and current Peak detection*

2.4.5 Speed/Position measurement inputs

A screw connector with 4 inputs can be used to plug two hall effect sensors. The two hall effect sensors inputs are connected to the two external interrupt pins (INT0 and INT1) of the microcontroller.

*Figure 2-7. Hall sensor effect interface*
2.4.6 LED

The ATAVRAUTO200 includes one green LED implemented on one I/O pin. It is connected to the “PortD Pin3” of the ATmega48/88/168. To light on the LED, the corresponding port pin must drive a low level. To light off the LED, the corresponding port pin must drive a high level.

![LED schematic](image)

2.4.7 BOOT

An additional jumper (BOOT) has been added. This jumper is available for custom use. For example: the BOOT jumper can be used to switch from the application to the bootloader by firmware (Not implemented in the example) by reading the pin7 of PortB.

![BOOT Jumper](image)

2.5 In-System Programming

The ATmega88 can be programmed using specific SPI serial links. This sub section will explain how to connect the programmer.

The Flash, EEPROM memory (and all Fuse and Lock Bit options ISP-programmable) can be programmed individually or with the sequential automatic programming option.

*Note:* If debugWire fuse is enabled, AVR ISP can’t be used. If debugWire fuse is disabled, JTAGICE mkII have to be used in ISP mode to enable debugWire fuse.

*Note:* When programming, the NISP jumper has to be removed.
2.5.1 Using the ATAVRAUTO900 Adaptator

An additional adapter has to be used to program the board using IPS or JTAG mode. The 10 pins connector is used for the JTAGICE mkII device and the 6 pins connector is used for the AVRISP device. To plug the ATAVRAUTO900 connector to the board, the arrow (on the adapter) has to be in front of the point (on the board).

![ATAVRAUTO900 Connection](image)

**Table 2-4. ICE Connector**

<table>
<thead>
<tr>
<th>PIN</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TCK</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>TDO</td>
</tr>
<tr>
<td>4</td>
<td>VCC</td>
</tr>
<tr>
<td>5</td>
<td>TMS</td>
</tr>
<tr>
<td>6</td>
<td>NRES</td>
</tr>
<tr>
<td>7</td>
<td>VCC</td>
</tr>
<tr>
<td>8</td>
<td>NC</td>
</tr>
<tr>
<td>9</td>
<td>TDI</td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
</tr>
</tbody>
</table>

**Table 2-5. ISP Connector**

<table>
<thead>
<tr>
<th>PIN</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MISO</td>
</tr>
<tr>
<td>2</td>
<td>VCC</td>
</tr>
<tr>
<td>3</td>
<td>SCK</td>
</tr>
<tr>
<td>4</td>
<td>MOSI</td>
</tr>
<tr>
<td>5</td>
<td>NRES</td>
</tr>
<tr>
<td>6</td>
<td>GND</td>
</tr>
</tbody>
</table>
Using the ATAVRAUTO200

2.5.2 Programming with AVR ISP Programmer

The AVR ISP programmer is a compact and easy-to-use In-System Programming tool for developing applications with ATmega88. Due to its small size, it is also an excellent tool for field upgrades of existing applications. It is powered by the ATAVRAUTO200 and an additional power supply is thus not required.

The AVR ISP programming interface is integrated in AVR Studio.

To program the device using AVR ISP programmer, connect the AVR ISP to the adaptor (ATAVRAUTO900) and connect the adaptor to the connector of the ATAVRAUTO200.

Figure 2-11. Programming from AVR ISP programmer using ATAVRAUTO900.

Note: See AVR Studio® on-line Help for information.

2.5.3 Programming with AVR JTAGICEmkII

The ATmega48/88/168 can be programmed using specific JTAG link: 3-wire debug-WIRE interface. To use the AVR JTAGICEmkII with an ATAVRAUTO200 the ATAVRAUTO900 adaptor has to be used. Then the JTAG probe can be connected to the ATAVRAUTO200 as shown in the following Figure 2-12.

To use the JTAGICEmkII in ISP mode the 3 jumpers “SCK”, “MISO” and “MOSI” of the adaptor (ATAVRAUTO900) should be connected.

Figure 2-12. JTAGICE mkII probe connecting through debugWIRE interface

Note: When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the debugWIRE system within the target device is activated. \textbf{RESET pin is configured as communication gateway between ATmega48/88/168 and JTAG. JTAGICE mkII must have control over it.}

Note: See AVR Studio® on-line Help for information.
2.6 Debugging

2.6.1 Debugging with AVR JTAG ICE mkII

The ATAVRAUTO200 can be used for debugging with JTAG ICE MK II.

Connect the JTAG ICE mkII as shown in Figure 2-12 for debugging, please refer to AVR Studio Help information.
Section 3
Technical Specifications

■ System Unit
  – Physical Dimensions...........................................L=45 x W=45 x H=8 mm
  – Weight ..............................................................................25 g

■ Operating Conditions
  – Internal Voltage Supply .................................................. 5.0V
  – External Voltage Supply ...........................................7V -18V
Section 4
Technical Support

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

- Which target AVR device is used (complete part number)
- Target voltage and speed
- Clock source and fuse setting of the AVR
- Programming method (ISP, JTAG or specific Boot-Loader)
- Hardware revisions of the AVR tools, found on the PCB
- Version number of AVR Studio. This can be found in the AVR Studio help menu.
- PC operating system and version/build
- PC processor type and speed
- A detailed description of the problem
Section 5

Complete Schematics

On the next pages, the following documents of ATAVRAUTO200 are shown:

- Complete schematics,
- Bill of materials,
- Assembly drawing.
Figure 5-1. ATAVRAUTO200 schematic
## Bill of Materials

**Source Data From:**

**Project:**

**Variant:**

**Date:** 11/14/07

### Table

<table>
<thead>
<tr>
<th>Designator</th>
<th>Description</th>
<th>Reference</th>
<th>Fabricant</th>
<th>Designation</th>
<th>Code Command</th>
<th>Unit de Vente</th>
<th>Quantity</th>
<th>Price UOV</th>
<th>Universel</th>
<th>Footprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>RELAY</td>
<td>ICONIC</td>
<td>PHOENIX</td>
<td>24</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.00</td>
<td>0.62</td>
</tr>
<tr>
<td>S1</td>
<td>RELAY</td>
<td>ICONIC</td>
<td>PHOENIX</td>
<td>24</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.00</td>
<td>0.62</td>
</tr>
<tr>
<td>L2</td>
<td>RELAY</td>
<td>ICONIC</td>
<td>PHOENIX</td>
<td>24</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.00</td>
<td>0.62</td>
</tr>
<tr>
<td>S2</td>
<td>RELAY</td>
<td>ICONIC</td>
<td>PHOENIX</td>
<td>24</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.00</td>
<td>0.62</td>
</tr>
<tr>
<td>L3</td>
<td>RELAY</td>
<td>ICONIC</td>
<td>PHOENIX</td>
<td>24</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.00</td>
<td>0.62</td>
</tr>
<tr>
<td>S3</td>
<td>RELAY</td>
<td>ICONIC</td>
<td>PHOENIX</td>
<td>24</td>
<td>10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.62</td>
<td>0.00</td>
<td>0.62</td>
</tr>
</tbody>
</table>

**Figure 5-2:** ATAVRAUTO200 Bill of Materials
Figure 5-3. ATAVRAUTO200 assembly drawing
Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel’s Terms and Conditions located on the Company’s web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© 2007 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are®, AVR®, AVR Studio® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® and others are registered trademarks or trademarks of Microsoft Corporation in the US and/or other countries. Other terms and product names may be trademarks of others.