Atmel AT02509: In House Unit with Bluetooth Low Energy Module Hardware User Guide

8-bit Atmel Microcontroller

Features

- Low power consumption
- Interface with BLE with UART
- Bi-direction wake up lines between BLE and Atmel® AVR® XMEGA® microcontroller
- 2Mbit DataFlash storage
- Customized segment LCD screen
- Buttons with multiple functions
- Buzzer and LED indicators
- Temperature sensor

Description

This document describes the design details of the In House Unit (IHU) project. It will guide the user to know and start the design based on this IHU board. It gives a clear concept of the entire design.

The IHU design with CSR Bluetooth® Low Energy (BLE) module is a platform to evaluate the integration of the BLE module and the Atmel AVR XMEGA microcontroller.

The kit offers a fixed interface with BLE and LCD and demos the basic communication between them. It is quite easy for the user to get started using AVR XMEGA peripherals.

For this reference design, the hardware design files (schematic, BoM and PCB gerber) can be downloaded from Atmel website. The provided hardware documentation can be used with no limitations to manufacture the reference hardware solution for the design.
# Table of Contents

1. General Description ......................................................................................... 3
   1.1 System overview ......................................................................................... 3
   1.2 Block diagram ............................................................................................. 3
   1.3 Function overview ..................................................................................... 4

2. Block Description .............................................................................................. 4
   2.1 UART communication with BLE ................................................................. 4
   2.2 LCD screen ................................................................................................ 4
   2.3 System clock consideration ......................................................................... 6
   2.4 Buttons ....................................................................................................... 7
   2.5 Ambient temperature sense ......................................................................... 9
   2.6 Battery voltage monitor ............................................................................ 11
   2.7 Power consumption ................................................................................... 11

3. Revision History ............................................................................................... 13
1. General Description

1.1 System overview

The system includes the main controller (Atmel ATxmega128B1), Bluetooth Low Energy (BLE) module, customized segment LCD, and DataFlash (Atmel AT45DB021) which can be used to store data from wireless nodes through the CSR BLE module or from internal source. Three LEDs, five buttons and one buzzer are designed as human machine interface. The system is powered by 2*AAA batteries.

Figure 1-1. IHU board.

1.2 Block diagram

Figure 1-2. Block diagram of the system.
1.3 Function overview

The main functions are:
1. The MCU reads and logs data from the BLE via the UART.
2. Display the necessary information on the LCD.
3. Data is stored in the DataFlash.
4. LEDs and buzzer are used as a status indication.
5. Buttons are open for the user to input something.

2. Block Description

2.1 UART communication with BLE

The communication between the MCU and BLE depends on the UART.
There are totally four lines between the MCU and BLE. Figure 2-1 shows the hardware connection.

Figure 2-1. Hardware connection between the MCU and BLE.

- Rx line defined as the data from BLE (PIO0) to MCU (PC2)
- Tx line defined as the data from MCU (PC3) to BLE (PIO1)
- BLE (PIO9) connects to MCU (PB2), defined as the BLE to wake up MCU
- MCU (PB1) to BLE (PIO10), defined as the MCU wake up the BLE

Input pin PB2 is asynchronous pin which can wake the device from all sleep modes. While the MCU is working on sleep mode, logic changes in the pin can trigger the interrupt to exit the sleep mode.

2.2 LCD screen

Since the MCU contains the LCD driver, the LCD is connected to the MCU directly. For LCD seg lines’s IO map, refer to Figure 2-3.

The LCD connection in this way will be convenient for the firmware engineer to code the display functions.
Figure 2-2. Hardware connection of LCD.

![Hardware connection of LCD](image)

Figure 2-3. Segment definition of LCD.

![Segment definition of LCD](image)
2.3 System clock consideration

The XMEGA device will start up with internal 2MHz by default. The clock accuracy will be 1.5% after factory calibration.

For this kit, we need to use the XMEGA to run the calendar, so that an accurate timer is necessary. An external 32.768kHz crystal is a reliable clock source for the user to run the calendar via the internal RTC module.

The 32.768kHz crystal is connected to pin 25 and 26 of the MCU. Figure 2-4 shows the details.

Figure 2-4. Hardware connection of 32.768kHz crystal.

The XMEGA has two selectable groups of pins to connect to an external crystal. The 32.768kHz crystal is connected to the “ALTERNATE” group in this design. To use this module, user must change the default value of the fuse settings.

The fuse settings can be found in Atmel Studio 6 → Tools → Device Programming. Figure 2-5 shows the details.

Figure 2-5. Fuse setting under Atmel Studio 6 with JTAG mkII.
2.4 Buttons

There are totally five buttons in this kit. All buttons are mapped to the IO pins directly. PG2 is the common pin for the async input.

Figure 2-6. Hardware connection of buttons.

Power-save mode is the deepest sleep mode that LCD can work. It is recommended to use this mode as the main sleep mode. Idle mode can be used while low speed communication is ongoing. PG2 is designed as the asynchronize wakeup source when a button pressed. See Figure 2-7 for the relationship between the XMEGA sleep modes and wake up source.

Figure 2-7. AVR XMEGA sleep mode and wake-up source.

Table 2-1. Active clock domains and wake-up sources in the different sleep modes.

<table>
<thead>
<tr>
<th>Sleep Modes</th>
<th>CPU Clock</th>
<th>Peripheral and USB Clock</th>
<th>RTC and LCD Clock</th>
<th>System Clock Source</th>
<th>RTC Clock Source</th>
<th>USB Resume</th>
<th>Asynchronous Port Interrupts</th>
<th>TWI Address Match Interrupts</th>
<th>RTI and LCD Clock Interrupts</th>
<th>All Interrupts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Power down</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Power save</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Standby</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Extended standby</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

A special method should be used to wake up the XMEGA from the power-save mode. Figure 2-8 gives the steps to wake up the XMEGA by pressing the button.
Before entering the power-save mode, set PG2 as input with internal pullup enabled, PG0, PG1, PG3, PG4, and PG5 set as output logic 0. If one or more buttons are pressed while XMAGA is sleeping, the PG2 will detect a falling edge, an async interrupt will be executed and wake up the MCU. Then, the PG2 set as output logic 0, and PG0, PG1, PG3, PG4, PG5 set as input with internal pull up. The user can easily know which button(s) is pressed via reading the IO data on these pins.

The XMEGA wake-up time is short enough for the operation above. Figure 2-9 is the wake-up time, which is extracted from the XMEGA datasheet.

**Figure 2-8.** AVR XMEGA button detection in Power-save mode.

**Figure 2-9.** AVR XMEGA wake-up time.
2.5 Ambient temperature sense

The temperature sensor circuit consists of a serial connection of a normal resistor and a NTC resistor. The part number of the NTC sensor is 103AT-4 from Semitec (http://www.semitec.co.jp). Figure 2-10 shows the relationship between the NTC resistance and the temperature, which is extracted from the 103AT-4 Datasheet. The data in red frame shows the temperature range between -20 to +60°C.

**Figure 2-10. Temperature and resistance table from the datasheet of 103AT-4.**

<table>
<thead>
<tr>
<th>Temperature</th>
<th>162AT</th>
<th>202AT</th>
<th>502AT</th>
<th>103AT</th>
<th>203AT</th>
<th>503AT</th>
<th>104AT</th>
<th>105AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>°C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>24.46</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>24.71</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>24.96</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>25.21</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>25.46</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>25.71</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>25.96</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>26.21</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>26.46</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>26.71</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>26.96</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>27.21</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>27.46</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>27.71</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>27.96</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>28.21</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>28.46</td>
<td>55.68</td>
<td>154.6</td>
<td>1263</td>
<td>3168</td>
<td>1147</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2-11 is the hardware connection of the NTC.**

**Figure 2-11. Hardware connection of the temperature sense.**

ADCA2 (PA2) is used to convert this divided voltage to digital. A formula can be applied to calculate the ADC result of voltage on R\_NTC:

\[
\text{ADC2value (12bit) = 4096*(V\_PA0*(R\_NTC/(R\_NTC+R3))/V\_REF)}
\]

Note that in this formula, V\_PA0 will be changed from time to time due to the discharge of the battery. This voltage can be sampled by ADCA1 (PA1) channel.

\[
\text{V\_PA0 = (ADC1value (12bit)*V\_REF*(R1+R2))/(4096*R2)}
\]
Thus, we can get a new formula containing both ADCA1 and ADCA2 result can be generated to calculate the resistance of the NTC:

$$R_{NTC} = \frac{1}{(ADCA1/ADCA2) \times (R1+R2)/R2\times R3 - (1/R3)}$$

$R1=300\,\Omega$, $R2=110\,\Omega$, $R3=240\,\Omega$, Table 2-1 shows the relationship between the ADCA1/ADCA2 and the temperature.

Table 2-1. Relationship between the ADCA1/ADCA2 and temperature.

<table>
<thead>
<tr>
<th>ADCA1/ADCA2</th>
<th>$R_{NTC}(\Omega)$</th>
<th>Temp(°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.218</td>
<td>67.77</td>
<td>-20</td>
</tr>
<tr>
<td>1.474</td>
<td>53.41</td>
<td>-15</td>
</tr>
<tr>
<td>1.784</td>
<td>42.47</td>
<td>-10</td>
</tr>
<tr>
<td>2.168</td>
<td>33.90</td>
<td>-5</td>
</tr>
<tr>
<td>2.629</td>
<td>27.28</td>
<td>0</td>
</tr>
<tr>
<td>3.188</td>
<td>22.05</td>
<td>5</td>
</tr>
<tr>
<td>3.853</td>
<td>17.96</td>
<td>10</td>
</tr>
<tr>
<td>4.652</td>
<td>14.69</td>
<td>15</td>
</tr>
<tr>
<td>5.594</td>
<td>12.09</td>
<td>20</td>
</tr>
<tr>
<td>6.707</td>
<td>10.00</td>
<td>25</td>
</tr>
<tr>
<td>8.014</td>
<td>8.313</td>
<td>30</td>
</tr>
<tr>
<td>9.546</td>
<td>6.940</td>
<td>35</td>
</tr>
<tr>
<td>11.319</td>
<td>5.827</td>
<td>40</td>
</tr>
<tr>
<td>13.380</td>
<td>4.911</td>
<td>45</td>
</tr>
<tr>
<td>15.747</td>
<td>4.160</td>
<td>50</td>
</tr>
<tr>
<td>18.478</td>
<td>3.536</td>
<td>55</td>
</tr>
<tr>
<td>21.590</td>
<td>3.020</td>
<td>60</td>
</tr>
</tbody>
</table>

Although the reference voltage of the ADC seems to be irrelevant in this formula, it is important to select a proper reference voltage to get higher accuracy (a balance between dynamic range and resolution). Considering the voltage level on PA0, internal 1V reference is recommended in this case.

Figure 2-11 shows the circuit of temperature sensing. There are several µA consumed by this circuit, which is comparable to the MCU power consumption in the Power Save mode, and obviously too high. PA0 is therefore designed to switch the temperature sensing circuit off during sleep state.

The input impedance will affect the maximum sample rate of the XMEGA ADC. The different input impedance on ADCA1 and ADCA2 leads to two maximum sample rates; 131ksps for ADCA1 and 191ksps for ADCA2. This should be enough for temperature sensing purpose.

Each time PA0 switches from low to high (enabling the temperature sensing circuit), it is necessary to wait until C2 and C3 are fully charged. Theoretically, the wait time is the longer the better, so in the actual application 8RC is used. Thus, 22ms is necessary for ADCA1 channel and 14ms for ADCA2. Considering the temperature calculation needs both data in the same time, it is recommended to use 22ms as the overall wait time.

Figure 2-12 shows the time flow for a sample process.
2.6 Battery voltage monitor

For the battery voltage monitor, it is quite similar with the temperature application above. ADCA4 is designed as the battery voltage monitor input.

Figure 2-13. Hardware connection of the battery voltage monitor.

The maximum sample rate in ADCA4 is 13.7ksps and the wait time for the ADCA4 is 220ms.

2.7 Power consumption

There are four main blocks of circuits consume power:

- MCU
- DataFlash
- BLE
- Others

Figure 2-14. The four blocks of power consumption.
As described in the XMEGA datasheet, the maximum current of the MCU is 15mA, but in this application, the current will be maximum 1.6mA (@2MHz) in active mode. Various sleep modes can further reduce the power consumption. For example, MCU can stay in idle mode while in low speed communication state and stay in power-save mode when waiting input from button-press or data reception from the CSR BLE module.

Dataflash needs up to 16mA when in operation, the BLE module consumes maximum 16mA. Others require 5µA to 10µA depending on the operation power supply.

Note: The maximum speed of XMEGA should be limited to 12MHz while the power supply is 1.8V. Considering the low speed communication, ADC sampling and Data flash writing, it is necessary to choose a proper MCU frequency to achieve the maximum energy saving.

For example, if there is an application as follows,

- Every 30s receive a packet
- Each packet last 50ms for data transmission
- 10ms for data writing to Data flash
- LEDs and buzzer active 100 times each day with 100ms each time

Then the average current should be approx.:

\[
\frac{(50\text{ms}(16\text{mA}+2\text{mA}) + 10\text{ms}(16\text{mA}+2\text{mA}))/30000\text{ms} + 5\text{mA}*100*0.1\text{s}}{86400\text{s}} + 10\mu\text{A} = 0.0466\text{mA}
\]

If the system is powered by 2*AAA batteries with 1000mAh whole life cycle capacity, it can last more than 21459 hours, which is approx. 2.4 years.
## 3. Revision History

<table>
<thead>
<tr>
<th>Doc. Rev.</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>42066B</td>
<td>09/2013</td>
<td>The last section (“The kit offers…”) on the front page is added. Former Chapter 3 (Firmware) has been removed</td>
</tr>
<tr>
<td>42066A</td>
<td>04/2013</td>
<td>Initial document release</td>
</tr>
</tbody>
</table>