APPLICATION NOTE

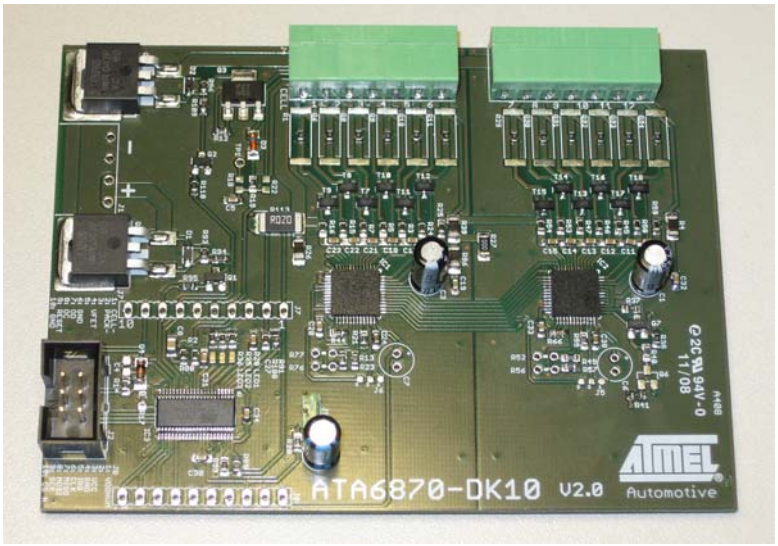
# User Guide for Atmel ATA6870 and Atmel ATmega32HVB Evaluation Kit Hardware

**ATA6870-DK10**

## Features

- Evaluation of Atmel® ATA6870
- Monitoring of 12 battery cells
  - Monitoring:
    - Overvoltage (every cell)
    - Undervoltage (every cell)
    - Overheating
    - Overcurrent
- Open clamp detection
- 12-bit battery cell measurement
- 12-bit temperature measurement
- Controlling of charge/discharge FETs
- Status LEDs for easy evaluation
- Charge balancing
- Coulomb counting for SOC determination

**Figure 1. Atmel ATA6870-DK10**



# 1. Introduction

The Atmel® ATA6870-DK10 is a demonstration board for the Atmel ATA6870, which offers an easy way to start evaluation of battery applications using the Atmel ATmega32HVB in combination with the Atmel ATA6870. The included software demonstrates implementation of a 12 Cell Battery Management System. The supplied code serves as an example of how to use the Atmel ATMega32HVB and Atmel ATA6870 together. The example is not a complete application intended for use with smart batteries, and it is best to use the devices in a slightly different way in a smart battery application.

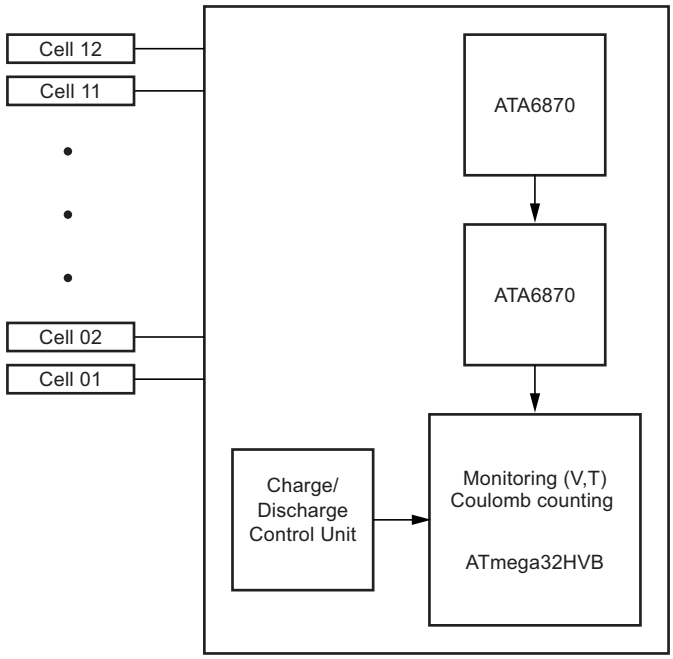
# 2. Safety Precautions When Using Li-ion Batteries

Please observe the safety guidelines supplied with the batteries. If improperly used or defective, li-ion and polymer batteries and packs may explode and cause a fire.

# 3. Demonstration Board

The Atmel ATA6870-DK10 was developed to allow easy evaluation of control software for a microcontroller which controls multiple Atmel ATA6870s. The sample code supplied demonstrates a simple permanent running measurement of voltages and temperatures.

**Figure 3-1. Board Concept**



## 3.1 System Start

Follow these steps to launch the system.

### 3.1.1 Installing the Hardware

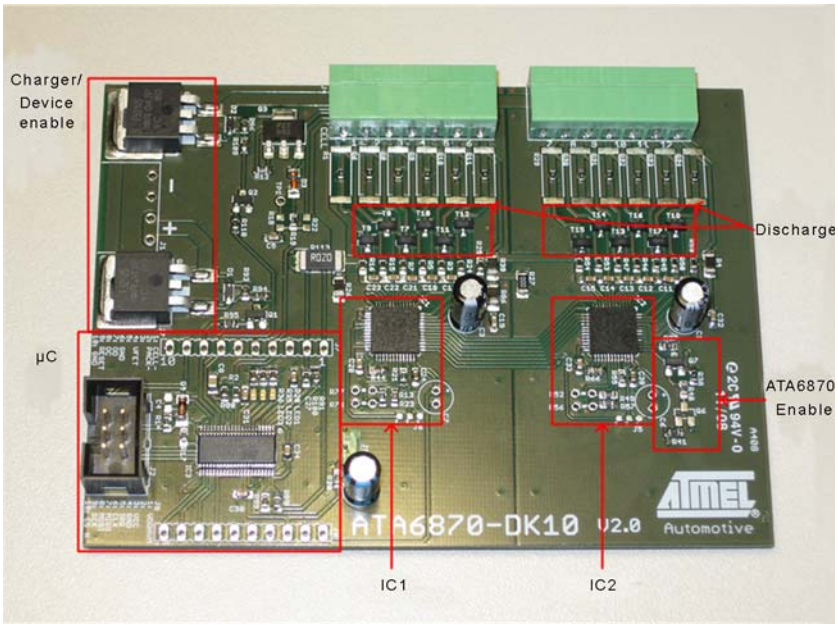
- Connect the load/charger to be powered between pack+ and pack- on J1
  - For demonstration purposes it is possible to use a resistor to simulate a load
- Connect the battery cell stack to the screw connectors on the demonstration board
  - Led 1 indicates the enabled status of the demonstration board (controlled by microcontroller SW)
- In case of emulating cells such as a voltage divider, apply sufficient voltage (see Section 3.3 “Powering the Board” on page 5)

### 3.1.2 Number of Cells

It is possible to run the board with a reduced number of cells. The minimum voltage for each IC is 6.9V. Cell 1 and cell 6 (MBAT) have to be connected. The missing cells should be connected to the upper cell potential of the module. For further information refer to the Atmel ATA6870 datasheet Section 7.3: Reduced Number of Battery Cells Configuration. For the voltage range see Section 3.3 “Powering the Board” on page 5. If fewer than 6 cells are used per IC, the config.h file should be adjusted (CELLSIC# under General Setting). See Section 4.1 “Supplied Code” on page 7 for further information on how to configure the supplied software correctly.

## 3.2 The Demonstration Board

**Figure 3-2. Evaluation Board with 2 Stacked Atmel ATA6870 and Atmel ATMega32HVB**



### 3.2.1 On-board Features

The demonstration board includes the following items:

- 2 × Atmel® ATA6870 QFN 7mm × 7mm
- Atmel ATMega32HVB
- 12 external N-channel MOSFETs for balancing of battery cells
- Connectors
  - ISP connector for programming/debugging the Atmel ATMega32HVB
  - Screw connectors for connecting up to 12 battery cells

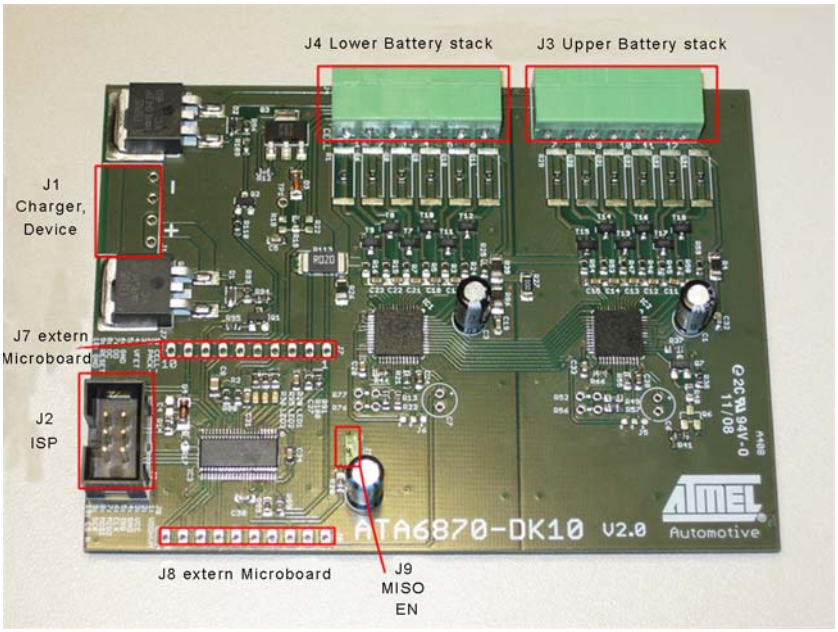
**Table 3-1. Connector Overview**

| J7 | Function | J8 | Function |
|---|---|---|---|
| 1 | CELL- | 1 | VDDHVM |
| 2 | PACK- | 2 | |
| 3 | | 3 | VCC |
| 4 | VFET | 4 | GND |
| 5 | | 5 | IRQ |
| 6 | GND | 6 | CLK |
| 7 | OD | 7 | MISO |
| 8 | OC | 8 | MOSI |
| 9 | RESET | 9 | SCK |
| 10 | GND | 10 | CS_N |

| | |
|---|---|
| **J1** | Connector for charger/device to be powered |
| **J2** | ISP connector |
| **J3** | Upper battery stack (cells 7-12) |
| **J4** | Bottom battery stack (cells 1-6) |
| **J9** | Jumper to enable/disable MISO line of Atmel ATA6870 |

J9 should never be set while the Atmel ATmega32HVB is being programmed or while it is entering debug mode. It can be mounted as soon as AVR Studio prompts for additional SPI lines to be connected in debug mode or after the device has been correctly programmed.

**Figure 3-3. Connectors**




## 3.3 Powering the Board

### 3.3.1 Power Supply

The board supports supply voltages from 13.8V (6.9V per Atmel ATA6870) to 60V. However, to run the board on voltages below 24V the ZDiode D3 needs to be replaced with a jumper to supply the Atmel® ATmega32HVB with sufficient voltage. If the jumper is mounted, the stack voltage should not exceed 48V! The Atmel ATmega32HVB supports operating voltage from 4V to 24V.

### 3.3.2 Emulating Cells

Battery cells can be emulated by connecting a voltage divider to the specified clamps. Section 3.1.1 “Installing the Hardware” on page 3 describes how to connect cells. The voltage limits for this setup are the same as for real batteries. Section 3.3.1 “Power Supply” on page 5 specifies these limits.

# 4. Software Description: Monitoring of Up to 12 Battery Cells

The supplied code is documented and easy to adjust for verifying the functions of the Atmel® ATA6870 and start BMS application development work.

After the board has been connected as described above the microcontroller automatically starts a cyclic measurement of voltages, temperature, and current. LED 1 indicates these cyclic measurements. It toggles in default operation. A continuously illuminated LED1 indicates an open clamp. See Section 4.2 “Open Cell Check” on page 7 for more information about open clamp detection. LED 2 indicates that for some reason the MOSFETS have been disabled. The default software disables the FETs in case of these events:

- Overvoltage (at least 1 cell exceeds the upper default threshold of 4.2V)
- Undervoltage (at least 1 cell exceeds the lower default threshold of 2.5V)
- Overcurrent (the current through the shunt exceeds the default threshold of 80mA)
- Overheating (the temperature exceeds the upper threshold, default value is 60°C)
- Low temperature threshold (the default threshold is -20°C)

LED 3 indicates whether the Atmel ATA6870s are turned on or not. An active LED indicates that the Atmel ATA6870s are enabled.

**Table 4-1. LED Functions**

| LED | Function |
|---|---|
| LED 1 | Indicates clamp is open when permanently illuminated<br>Indicates cyclic measurements when blinking |
| LED 2 | On indicates disabled MOSFETs for one of the reasons listed above |
| LED 3 | On indicates active Atmel ATA6870 |

The Atmel ATmega32HVB has no clock divider to provide an external slower clock than 1/2 CPU clock. Requirement of Atmel ATA6870 is $f_{CLK} > 2 \times f_{SPI}$. Hence, the clock frequency of 1MHz is mandatory to provide a 500kHz clock for the ADCs of the Atmel ATA6870 and 250kHz for SPI.

## 4.1 Supplied Code

### 4.1.1 config.h

This section refers to the config.h file provided in the zip archive with this Application Note. Only values in the User Setting paragraph should be changed!

```
------------ GENERAL SETTING-------------------------------
CELLSIC#                  Selecting which Cells are used Bits 0-5 -> Cells 1-6
------------ TEMPERATURE SETTING---------------------------
RES_REF#                  Value of the mounted reference resistor (default: 3300)
T_TLS                     Temperature belonging to the first Value in the lookup
                          table (index 0, default: -20)
T_TLE                     Temperature belonging to the last value in the lookup
                          table (default: 80)
T_TLSZ                    Temperature step size used in the lookup table (default:
1)
T_LOWERTHRESHOLD          Lower temperature threshold
T_UPPERTHRESHOLD          Upper temperature threshold
 ------------ COULOMBCOUNTER SETTING------------------------
SHUNT_RESISTANCE          Value of the shunt resistor in mOhm
RCC_CONVERSIONPERIOD      The cycle times for the Regular Current Check
                          0x00 - 256ms (default)
                          0x01 - 512ms
                          0x02 - 1s
                          0x11 - 2s
RCC_DIVIDEDSZ             0x01 to enable divided Voltage (Current) stepsize
RCC_CHARGETHRESHOLD       Threshold for charging current, exceeding the
                          threshold will turn off the Mosfets
RCC_DISCHARGETHRESHOLD    Threshold for discharging current, exceeding the
                          threshold will turn off the Mosfets
```

Other values should not be changed in the default HW setup!

## 4.2 Open Cell Check

The implemented function checks for open clamps by measuring the cell voltages two times. During the first check a normal measurement is completed and the values stored. During the second check the voltages are measured while the discharge function for all cells is active. If the two measurements for the same cell differ by more than 100mV it is very likely that one or more cells are not properly connected. The implemented method cannot be used to determine which cell is not properly connected. A continuously illuminated LED1 indicates an open clamp.

## 4.3 Voltage Measurements

The standard software loop measures the voltage ADC value and the offset ADC value for every cell and checks for overvoltage and undervoltage once per cycle. Further information about the acquiring of voltages can be found in the Atmel® ATA6870 datasheet Section 7.5.1. The formula for calculating the voltage:

$$\text{Voltage (Cell)} = 4\text{V} \times \left( \frac{V_{acq} - V_{offset}}{3031 - V_{offset}} \right)$$

## 4.4 Temperature Measurements

The default software only measures channel 1 of chip 1. The temperature sensors are based on a resistor divider using a standard resistor and an NTC resistor. This resistor divider is connected to the reference of the ADC for temperature measuring. Because the ADC is sharing the same reference value, the output of temperature measurement with ADC is ratio metric. Further information is found in the Atmel ATA6870 datasheet Section 7.5.3: Temperature Channel.

For this application Atmel recommends using Res_Ref1 = 3.3kΩ and RES_NTC1 R25 = 10kΩ, B = 3435. The software supplied for this board uses these values as default. The function uses a lookup table to determine the temperature. This table has to be edited if an NTC other than the recommended one is used. The values in the lookup table range from –20°C (index 0) to +80°C (index 100). These values can be edited via the config.h file in the User Settings section. More Information about this file can be found in Section 4.1 “Supplied Code” on page 7. The calculation of RES_NTC is carried out based on the formula provided in the Atmel ATA6870 datasheet Section 7.5.3:

$$\text{adc (out)} = 2048 \times \left(1 + \frac{\text{RES\_NTC(1)}}{(\text{RES\_NTC(1)} + \text{RES\_REF(1)})} \times \frac{8}{15} - \frac{8}{10}\right)$$

When using another NTC, the LookupADC.txt has to be edited to match the NTC used.

## 4.5 State of Charge Measurements

Highly precise SOC measurement is possible by combining the features of the Atmel ATmega32HVB and the Atmel ATA6870. The coulomb counting feature of the Atmel ATmega32HVB enables highly precise measurements of the change in the state of charge. Frequent reading of the current in a shunt is used to update the SOC frequently. The acquired cell voltages and temperatures can be used to determine the SOC without the Atmel ATmega32HVB. The easiest way is to compare the SOC measured by the added/extracted charge with the calculated SOC using the cell voltage, temperature, and the data provided by the manufacturer of the cells. Further information regarding the coulomb counting ADC as well as an implementation suitable for the Atmel ATmega16HVA is found in Application Note AVR352.

## 4.6 Overcurrent Protection

The current through the shunt is calculated by measured voltage drop. The limit can be set via the CADRDC/CADRCC register. The step size depends on the settings of the CADCSRC register and the shunt used. For further information about limiting current see the Atmel ATmega32HVB datasheet Section 19.4: Regular Current Detection Operation. The supplied software allows the feature to be tested by adjusting the values in the config.h file. More Information about this file can be found in Section 4.1 “Supplied Code” on page 7. Values/part of the code should only be changed if you are aware of possible consequences. The default implementation continuously measures the current and generates an interrupt if the entered thresholds are exceeded. The thresholds are defined in the config.h file. The thresholds are written to the registers in the function CCinit in the Atmel ATA6870_func.c file. Refer to the features of the Atmel ATmega32HVB in the coulomb counter section to learn more about the time the controller waits for the values to be written.

C Code Example

```
CADRCC = RCC_CADRCC;                    // Charge Threshold
while(CADCSRA & (1 << CADUB));          // Wait values to be written
CADRDC = RDC_CADRDC;                    // Discharge Threshold
while(CADCSRA & (1 << CADUB));          // Wait values to be written
```

# 5. Features of the Atmel ATmega32HVB

Since the Atmel® ATmega32HVB is a part of the Atmel AVR® family which is dedicated to battery management there are several special features such as coulomb counting and the control of the two charge/discharge MOSFETs.

## 5.1 Coulomb Counter

The coulomb counter ADC runs on a different clock than the CPU. This clock is slower and therefore several things have to be kept in mind before using it. Writing several registers in sequence takes a long time depending on the delays between each write cycle. A possible solution is given in the supplied software example:

C Code Example

```
void CCinit(){

        CADRCC = RCC_CADRCC;                    // Charge Threshold
        while(CADCSRA & (1 << CADUB));
        CADRDC = RDC_CADRDC;                    // Discharge Threshold
        while(CADCSRA & (1 << CADUB));
        SETBIT(CADCSRB,1<<CADRCIE);             // Interrupt Enable
        while(CADCSRA & (1 << CADUB));
                                                // Voltage Scaling
SETBIT(CADCSRC,RCC_DIVIDEDSZ<<CADVSE);
        while(CADCSRA & (1 << CADUB));
        SETBIT(CADCSRA,((1<<CADEN)|(1<<CADSE)|(RCC_CONVERSIONPERIOD<<1)));
                                                // ADC Enable, RCC Mode, Sampling
                                                // Interval
        while(CADCSRA & (1 << CADUB));
}
```

The Update Busy (CADUB) bit in CADSRA is cleared and written by hardware.

## 5.2 Charging/Discharging FETs

The two FETs are controlled by an N-channel FET driver. The pins (OC and OD) are designed for outputting a high voltage of approx. 13V. The status of the pins is controlled by software via the FCSR - FET control and status register.

C Code Example

```
void Configure_Fet(unsigned char Fet){

        if(Fet&0x01)
                SETBIT(FCSR, (1<<DFE));
        else
                CLEARBIT(FCSR,(1<<DFE));

        if(Fet&0x02)
                SETBIT(FCSR,(1<<CFE));
        else
                CLEARBIT(FCSR,(1<<CFE));
}
```

The example above implements an easy method to enable or disable the two FETs independently of each other. For more information, see the Atmel ATmega32HVB datasheet page 148ff.

# 6. Power Consumption

There are several ways to reduce the power consumption of the Atmel ATA6870 and the Atmel ATmega32HVB. Sleep modes are documented in the datasheet of the Atmel ATA6870 Section 7.1.1 and in the Atmel ATmega32HVB datasheet Section 10. This board allows the Atmel ATA6870 to be enabled/disabled using the Atmel ATmega32HVB software. The pin PB2 is used to control a transistor for activating/deactivating the Atmel ATA6870. Other options which are not implemented are the use of interrupts and a timer (sleep between cycles).

# 7. Schematic

**Figure 7-1. Schematic**

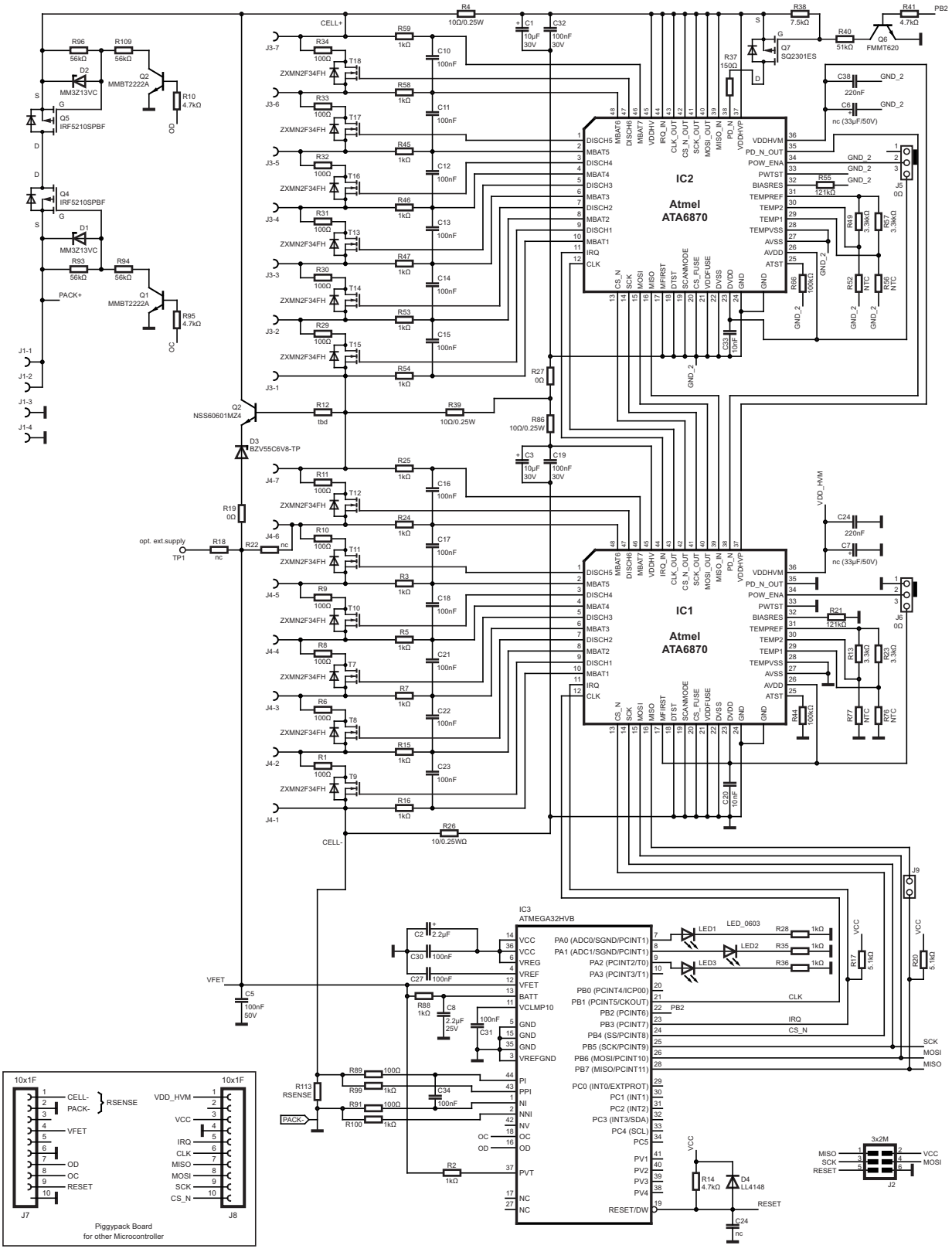


**Figure 7-2. PCB Top**

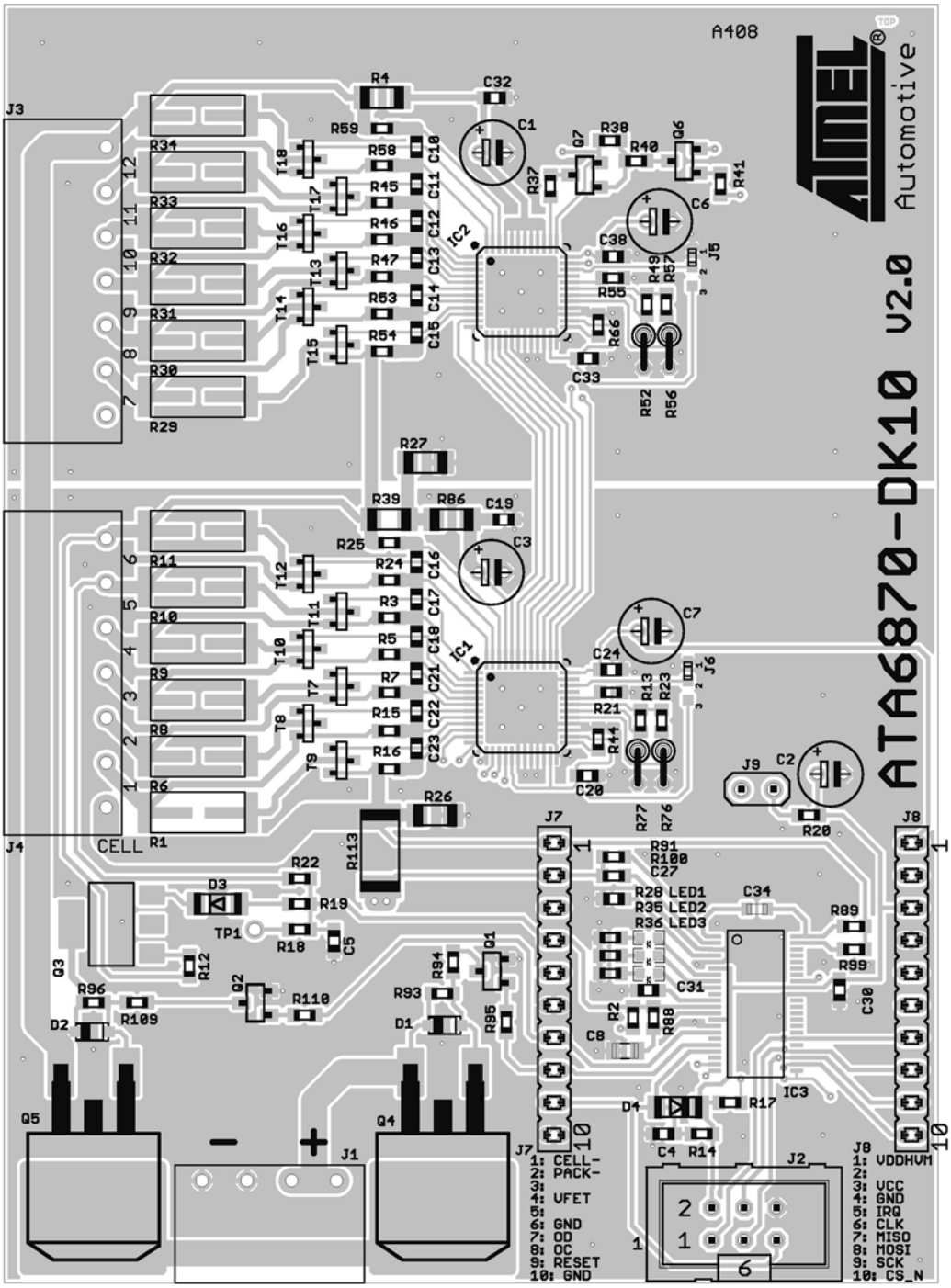
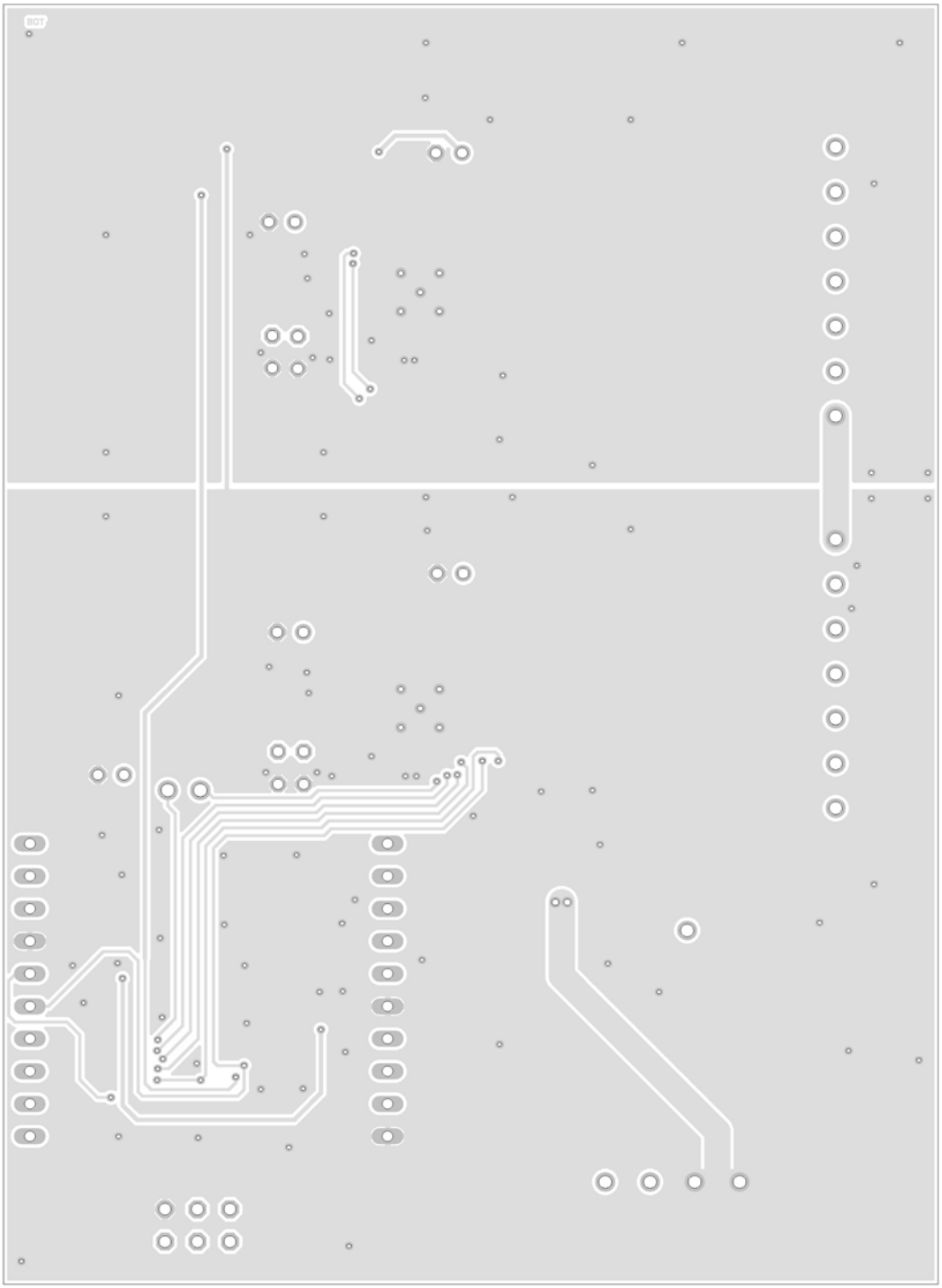


**Figure 7-3. PCB Bottom**



# 8. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

| Revision No. | History |
|---|---|
| 9228C-AUTO-02/15 | • Put document in the latest template |
| 9228B-AUTO-10/12 | • Section 4.4 “Temperature Measurements” on page 8 updated |


**Atmel Corporation** 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

# X-ON Electronics

Largest Supplier of Electrical and Electronic Components

***Click to view similar products for*** Power Management IC Development Tools ***category:***

***Click to view products by*** Microchip ***manufacturer:***

Other Similar products are found below :

EVB-EP5348UI MIC23451-AAAYFL EV MIC5281YMME EV 124352-HMC860LP3E DA9063-EVAL ADP122-3.3-EVALZ ADP130-0.8-EVALZ ADP130-1.8-EVALZ ADP1740-1.5-EVALZ ADP1870-0.3-EVALZ ADP1874-0.3-EVALZ ADP199CB-EVALZ ADP2102-1.25-EVALZ ADP2102-1.875EVALZ ADP2102-1.8-EVALZ ADP2102-2-EVALZ ADP2102-3-EVALZ ADP2102-4-EVALZ AS3606-DB BQ25010EVM BQ3055EVM ISLUSBI2CKIT1Z LP38512TS-1.8EV EVAL-ADM1186-1MBZ EVAL-ADM1186-2MBZ ADP122UJZ-REDYKIT ADP166Z-REDYKIT ADP170-1.8-EVALZ ADP171-EVALZ ADP1853-EVALZ ADP1873-0.3-EVALZ ADP198CP-EVALZ ADP2102-1.0-EVALZ ADP2102-1-EVALZ ADP2107-1.8-EVALZ ADP5020CP-EVALZ CC-ACC-DBMX-51 ATPL230A-EK MIC23250-S4YMT EV MIC26603YJL EV MIC33050-SYHL EV TPS60100EVM-131 TPS65010EVM-230 TPS71933-28EVM-213 TPS72728YFFEVM-407 TPS79318YEQEVM UCC28810EVM-002 XILINXPWR-083 LMR22007YMINI-EVM LP38501ATJ-EV