AN-1014
Application Note

Using the Atmel® picoPower™ AVR® Microcontroller with Cymbet™ EnerChips™

A Demonstration Vehicle for Cymbet Solid State Thin Film Rechargeable Batteries

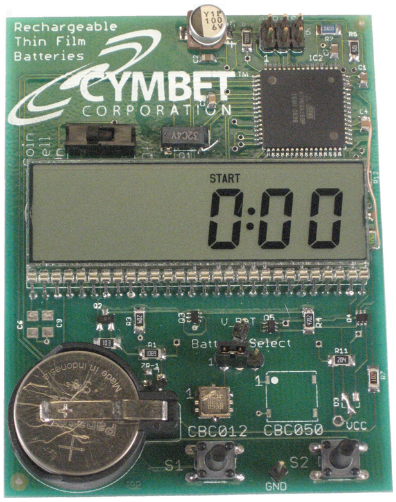


Figure 1. Cymbet Atmel picoPower AVR Demo Board.

Introduction

The Cymbet EnerChip and Atmel picoPower AVR are combined in a demonstration board designed to show the advantages of Cymbet EnerChip rechargeable batteries for embedded processor applications. Cymbet EnerChips provide virtually unlimited system life. They are smaller, lighter and environmentally friendlier than primary lithium coin cells for backup power. The demo board consists of an Atmel Atmega169P picoPower AVR microcontroller with integrated LCD controller, Cymbet EnerChip cells, and a Cymbet charge control circuit.

System Considerations

Cymbet EnerChips are ideal for applications requiring a high number of charge/discharge cycles, such as backing up microcontrollers during intermittent power outages. Such systems will need a charge controller to keep the EnerChip at a full state of charge without overcharging the battery and thereby reducing its useful life. The charge circuit will need a source of voltage higher than the EnerChip charging voltage of 4.1V. This may require a charge pump. The charge circuit must regulate the charge voltage and have a means of isolating itself from the EnerChip (current blocking) to prevent the EnerChip from discharging through the charge circuit when system power is off. Isolation from the main power supply and voltage reduction from the battery output to the load may also be necessary.

Consideration needs to be given to the minimum voltage needed by the system to run off backup power, on how to switch between main power and backup power, and on how to signal the microcontroller that it will be running off backup power. Special algorithms may be used so that the microcontroller sheds loads and puts itself into a low-power sleep mode during backup.

The power-fail sensing circuit must not feed power back into the main power bus when the system switches over to the EnerChip for backup power. In the Demo Board, a 4.7 MOhm resistor is used to limit current on the V_{REF} line to ≈500 nA. A diode would have eliminated all current leakage and have been an even better choice.

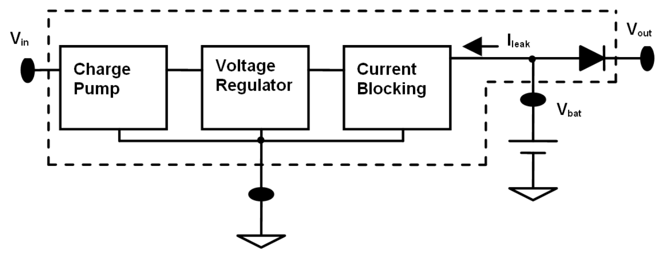


Figure 2. Charge Controller Block Diagram.

Charge Circuit Description

The charge circuit in *Figure 4* meets all key system requirements. It is powered by two lithium coin cells, so that no boost charge pump is needed. Charge voltage regulation utilizes a Zetex ZR40401F41TA shunt voltage regulator, but could also have used a low I_Q LDO, such as a TI TPS715XX regulator.

Current blocking to prevent the EnerChip from being discharged when main power is off is implemented with transistors Q2, Q3, and Q5. Transistors Q3 and Q5 are dual packaged devices, with the two transistors wired in series so that the off-state leakage current is negligible for a µAh-rated EnerChip.

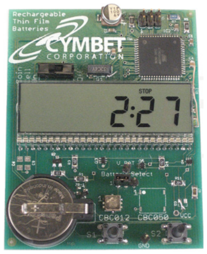
PNP transistor Q2 is used as a comparator to shut off the charge circuit when the input voltage falls below ≈ 4.8V (4.1V plus the V_{BE} of the transistor). When Q2 turns off, N-channel FET Q3 is turned off, which then turns off P-channel FET Q5.

Demonstration Description

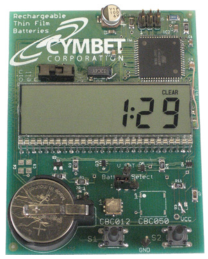
Two on-board coin cells are used as the primary power source in lieu of an external power supply. Place the two coin cells in series in the coin cell holder, with the negative terminal toward the board for both cells. Position the battery-select jumper to select the correct EnerChip. Close power switch SW1 for charging. Also press pushbutton S1 (at bottom left of board) to turn off the LCD display and minimize current drain. The EnerChip will be 80% charged within 30 minutes and 100% charged after 45 minutes. See AN-1003 for more details on battery charging times.

After the battery is fully charged, slide the power switch to Off and back to On. This will turn on the LCD display. The indicator caption will be START. The board will wait for an operator input.

Display ready for timing.



Display while timing after pressing S2.



Display stopped after pressing S2.

Press pushbutton S2 (at bottom right of board) to start the timer function. The LCD will show elapsed minutes and seconds. The indicator caption will be STOP. Press S2 again to stop timing. The indicator caption will be CLEAR. Press S2 once more to clear the display and reset the timer.

Turn off power switch SW1 or remove the coin cells to demonstrate Demo Board operation from the EnerChip alone.

Press pushbutton S1 (at bottom left of board) to toggle the LCD display on and off. With the LCD off, current draw on the EnerChip will be ≈ 2 µA. A 50 µAh CBC050 EnerChip will then provide backup for 25 hours and a CBC012 EnerChip for 6 hours. With the LCD on, current draw will be ≈ 14 µA, and the above times will be reduced to 3.6 hours and 51 minutes, respectively. To measure current draw, insert a micro-Ammeter in place of the battery-select jumper.

The 3-1/2 digit LCD display will first show minutes and seconds in MM:SS format. After 19:59 (19 min: 59 sec), it switches to HH:MM format. After 19:59 (19 hrs: 59 min), the unit stops counting.

When power switch SW1 is in the Off position, the display is automatically blanked and the EnerChip is being discharged. When finished with the evaluation, remove the battery select jumper to prevent the EnerChip from becoming fully discharged, possibly reducing its life.

Software Description

The software uses an interrupt driven control loop. The processor is normally in the "powersave" mode to conserve power. Timer2 is clocked off the external 32.768 KHz crystal and generates an interrupt at 1 sec intervals. This wakes the processor, which will then update the display as needed, then go back to sleep.

Timer Initialization Routine

```
void timer2_init(void)
{
#asm("cli")          // clear global interrupt flag
TIMSK2 = 0;          // disable OCIE2A and TOIE2
ASSR=0x08;           // use external 32 crystal
TCNT2=0x00;          // clear counter
TCCR2A = 0x05;       // divide by 8
while ((ASSR & 1) | (ASSR & 4));
                     // wait for tcn2ub and tcr2ub to clear
TIFR2 = 0x7F;        // set overflow limit
TIMSK2 = 0x01;       // set overflow interrupt enable bit
#asm("sei")          // reset global interrupt flag
}
```

LCD Setup Registers

```
LCDCRA=0x80;         // normal waveform; buffers on
LCDCRB=0xD7;         // now set for external clock 32.768
LCDCCR= 0x0F;        // contrast control
LCDFRR=0x02;         // clock divider
```

Other LCDs might require different settings. Further power reductions can be obtained by turning off the buffers and using the ATmega169 "Low Power" LCD waveform; however, this may reduce the contrast and sharpness of the display. Consult Atmel, www.atmel.com, for further details and latest information on the ATmega169P AVR microcontroller.

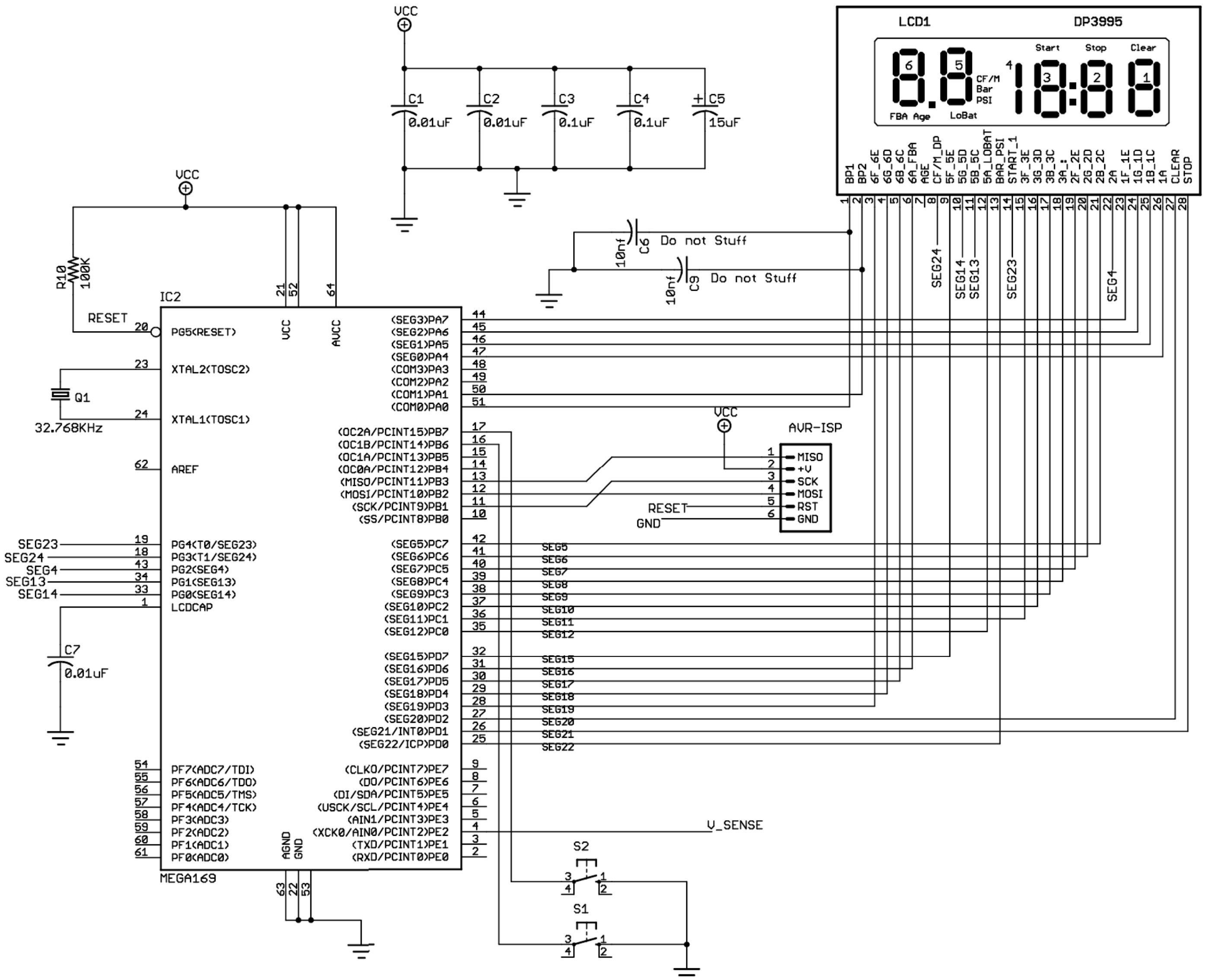


Figure 3. Schematic Diagram, Atmel ATmega169P Microcontroller.

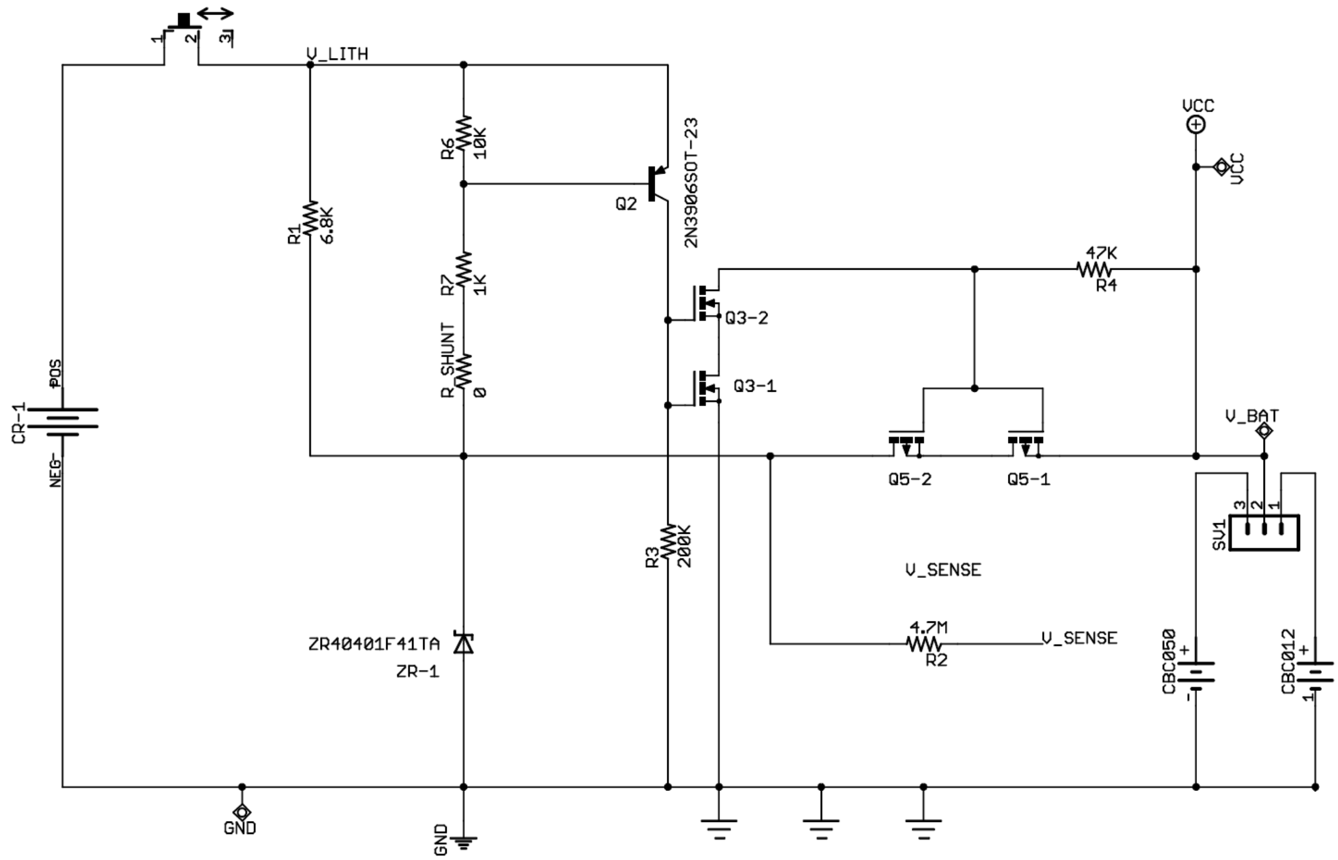


Figure 4. Charge Control Circuit.