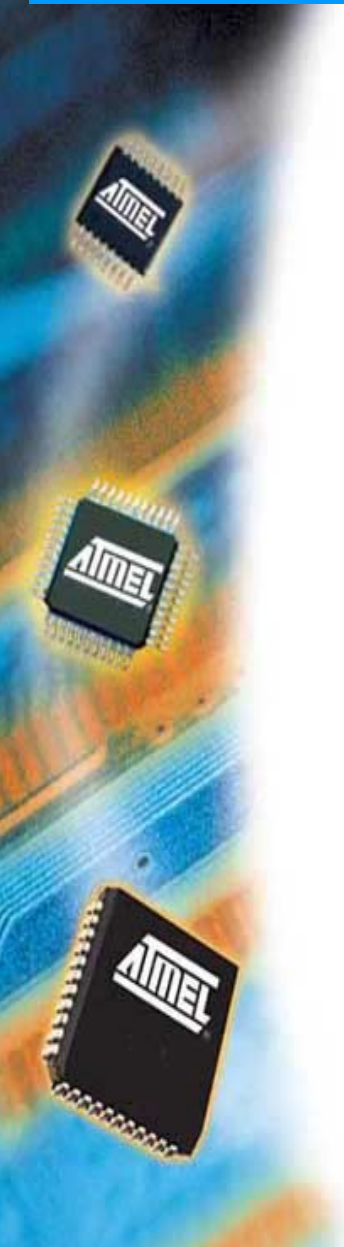# Introduction to the Atmel AVR Butterfly
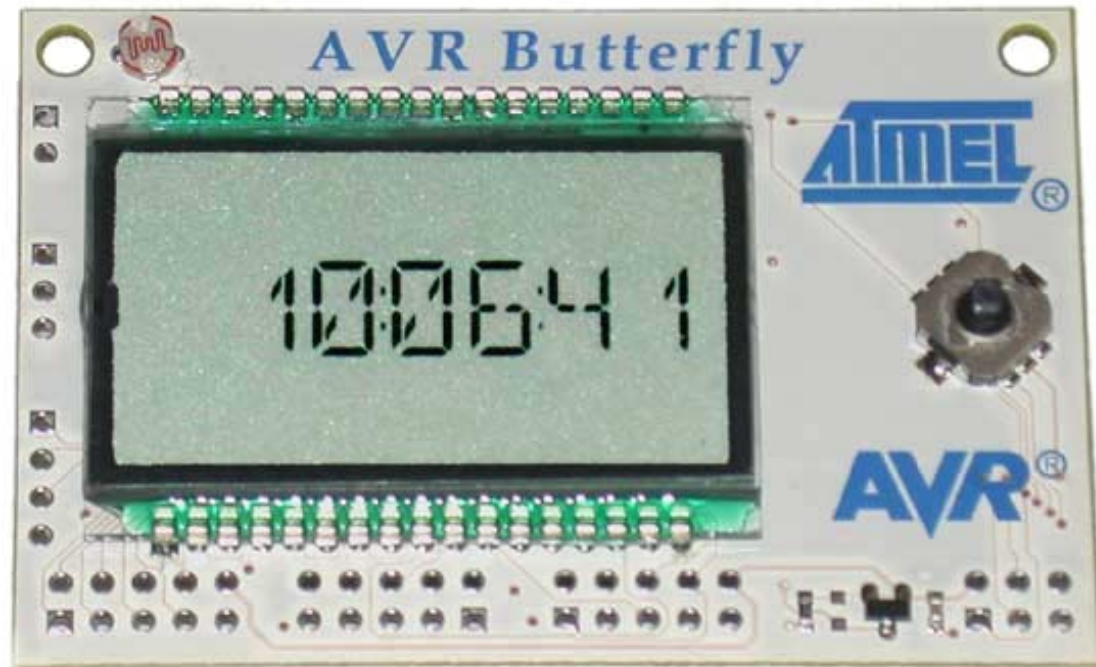
The AVR Butterfly is designed to demonstrate the benefits and key features of the AVR microcontrollers

- **The AVR architecture in general and the ATmega169 in particular**
- **Low Power Design**
- **The MLF package type**
- **Using peripherals**
  - ➤ **LCD controller**
  - ➤ **Memories**
    - **Flash, eeprom, sram, external Data Flash**
  - ➤ **Communication interfaces**
    - **UART, SPI, USI**
  - ➤ **Programming methods**
    - **Selfprogamming/ Bootloader, SPI, Parallel, JTAG**
  - ➤ **Analog to Digital Converter (ADC)**
  - ➤ **Timers/Counters**
    - **Real Time Counter (RTC)**
    - **Pulse Width Modulation (PWM)**
  - ➤ **etc….**
- **It also serves as a development kit for the ATmega169, and can be bought and used as a standard module in the customers own products**
- **Sales price: 19$**

Graphical overview

AVR Butterfly

ATmega169

32 kHz
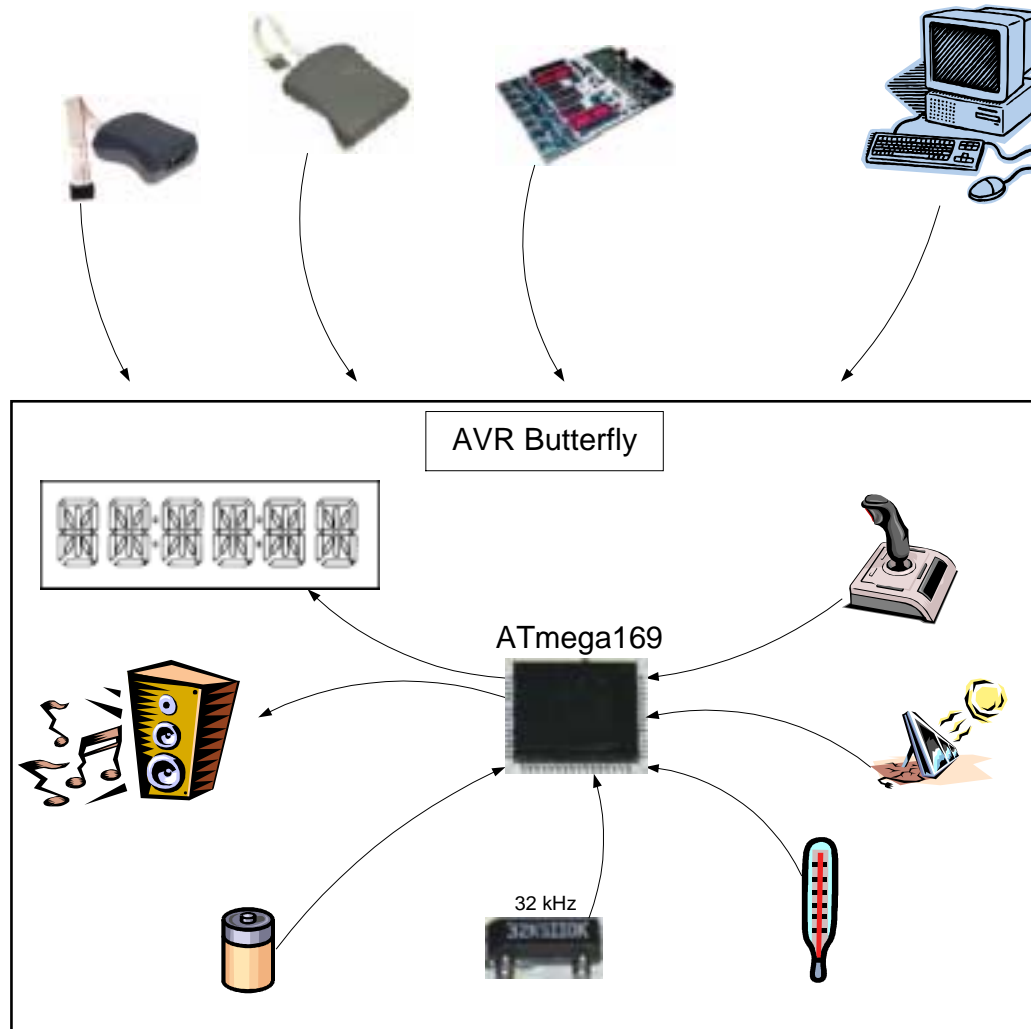
ATmega169

- **16KB Flash**
- **512B EEPROM**
- **1KB Internal SRAM**
- **JTAG Interface**
- **4 x 25 Segment LCD Driver**
- **Two 8-bit Timer/Counters**
- **One 16-bit Timer/Counter**
- **Real Time Counter**
- **Four PWM Channels**
- **8-channel, 10-bit ADC**
- **USART**
- **SPI**
- **Universal Serial Interface**
- **Watchdog Timer**
- **Analog Comparator**
- **Power-on Reset and Brown-out Detection**
- **Internal Calibrated Oscillator**
- **Five Sleep Modes:**
  - **Idle, ADC Noise Reduction, Power-save, Power-down, and Standby**
- **53 Programmable I/O Lines and 1 Input Line**
- **64-lead TQFP and 64-pad MLF**
- **Operating Voltage:**
  - **1.8 - 3.6V for ATmega169V**
  - **2.7 - 3.6V for ATmega169L**
- **Temperature Range:**
  - **0°C to 70°C**
- **Speed Grade:**
  - **0 - 1 MHz for ATmega169V**
  - **0 - 4 MHz for ATmega169L**
- **Ultra-Low Power Consumption**
  - **Active Mode:**
    - **1 MHz, 1.8V: 300µA**
    - **32 kHz, 1.8V: 20µA (including Oscillator)**
    - **32 kHz, 1.8V: TBD (including Oscillator and LCD)**
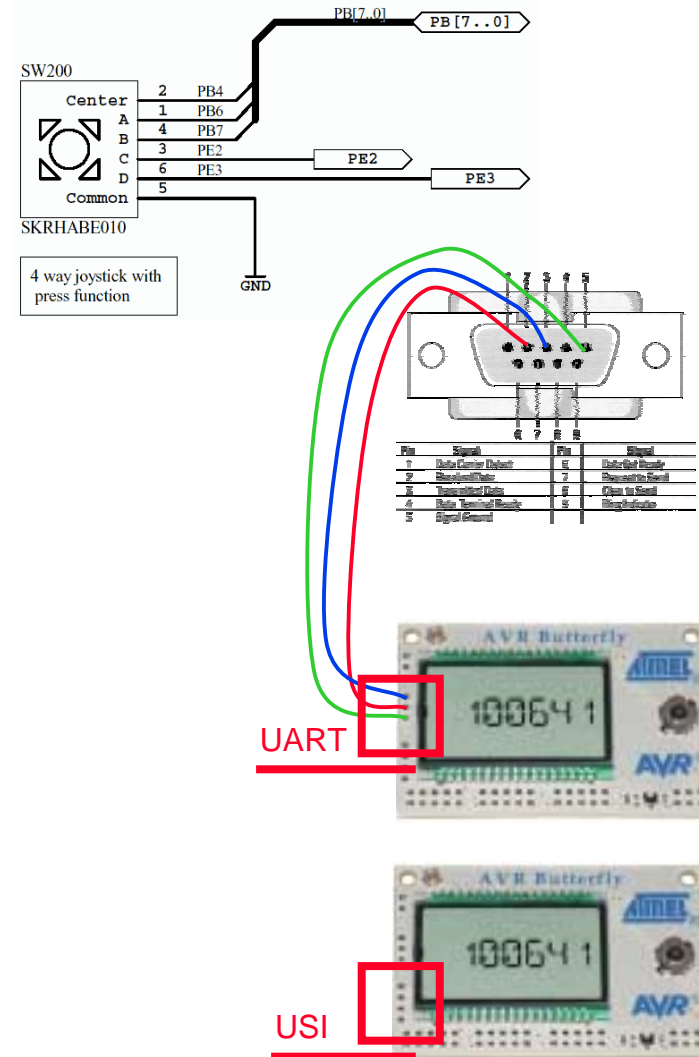 - **Power-down Mode:**
    - **0.5µA at 1.8V**
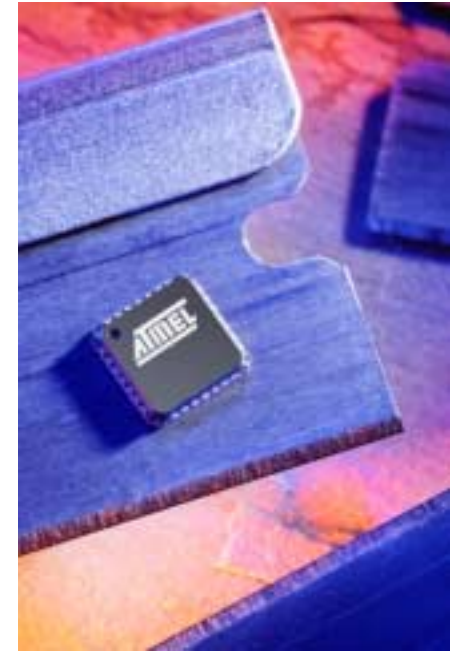
- **Joystick**
  - ➤ **4 directions**
  - ➤ **1 center push**

- **UART**
  - ➤ **Available on pin header J406**
  - ➤ **With level converters**
    - **Just connect TxD, RxD and GND**
   - **Vcc min 2.0V**

- **USI**
  - ➤ **Available on pin header J405**
  - ➤ **Uses 3 pins: PE4,PE5,PE6**
  - ➤ **If USI is not needed then the pins can be used as normal IO**

- **Reset**
  - ➤ **Short cut pin 5 & 6 on the ISP header (J403)**

- **Temperature**
  - ➢ **Surface mounted NTC (Negative Temperature Coefficient) thermistor**
  - ➢ **Tested -10 → +60 deg**
  - ➢ **Connected to ADC0 (PF0)**

- **Light**
  - ➢ **Trough hole mounted LDR (Light Dependent Resistor)**
  - ➢ **NSL19M51 CdS photoconductive cell**
  - ➢ **6K@10lux, 20M@darkness**
  - ➢ **Connected to ADC2 (PF2)**

- **Voltage**
  - ➢ **2 pin header**
  - ➢ **Voltage readings from 0-5V**
 - ➢ **Connected to ADC1 (PF1)**

- **LCD (Liquid Crystal Display)**
  - ➢ **Features six 14-segments digits, and some additional segments**
  - ➢ **All in all the display supports 120 segments**
  - ➢ **ATmega169 supports 100 segments**
  - ➢ **Same LCD and selected segments as on STK502**

- **Piezo-element**
  - ➢ **For alarms and "music"**
  - ➢ **Connected to Timer1 PWM Output A (PB5)**
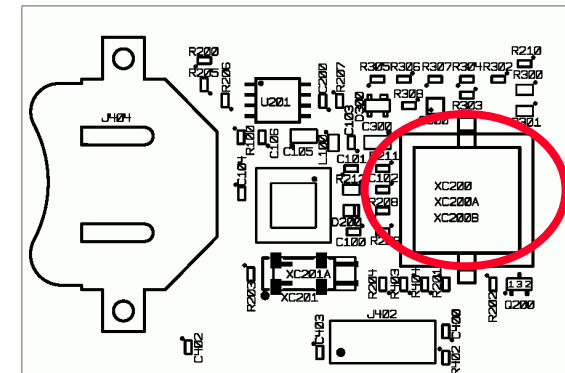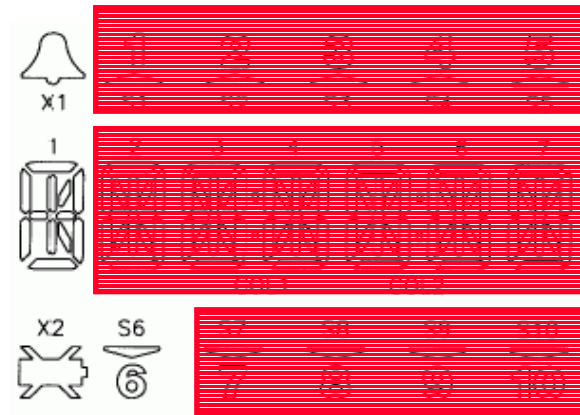
- **ATmega169**
  - ➢ **16Kbyte FLASH**
  - ➢ **512byte EEPROM**
  - ➢ **1Kbyte SRAM**

- **AT45DB041B**
  - ➢ **4Mbit data flash**
  - ➢ **SPI interface**
  - ➢ **Vcc 2.5-3.6V**
  - ➢ **Low level drivers included**
  - ➢ **Connected to SPI bus for external programming**
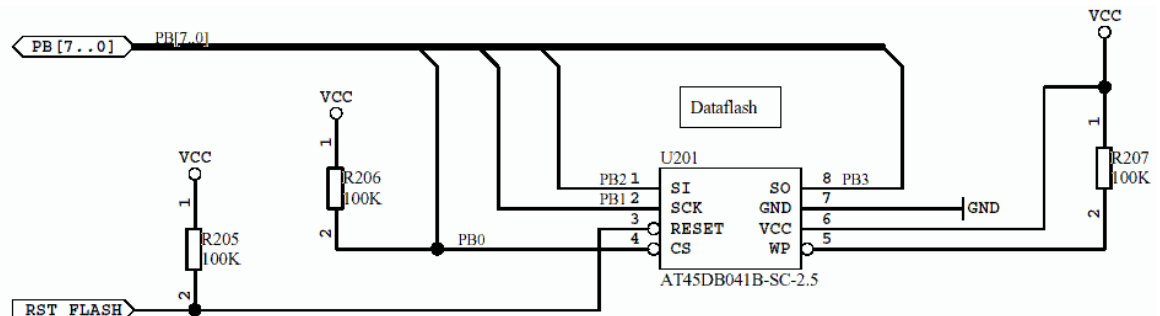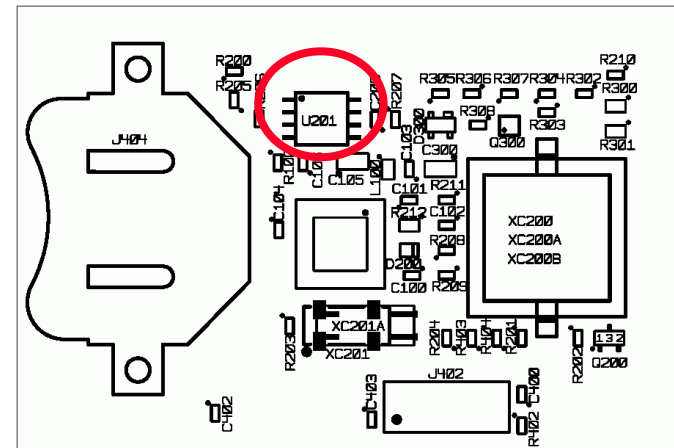
- **Real Time Counter (RTC)**
  - ➢ **On-board 32kHz Xtal for tracking of clock and date**

- **Internal Calibrated RC oscillator on ATmega169**
  - ➢ **Use prescaler to get 31k-8MHz system clock**
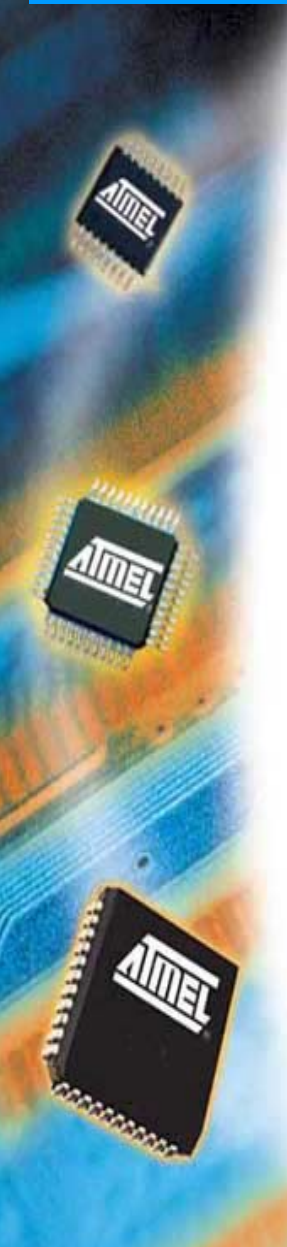
- **Button cell battery**
  - **CR-2450**
  - **3V**
  - **600mAh**

- **Using external power supply**
  - **GND and Vcc are available on several pin headers**
  - **J400 & J401pin 9 & 10**
    - **Parallel programming header**
  - **J402 pin 2 & 4**
    - **JTAG header**
  - **J403 pin 6 & 2**
    - **SPI header**
  - **Use of external power will not charge the battery**
  - **Remove the battery, or use a voltage level higher then the battery**

Vcc/GND

- **Bootloader**
  - ➤ **Enables upgrade of the application code from a PC without any external hardware**
  - ➤ **Frontend software: AVRprog (included with AVR Studio)**
  - ➤ **Uses UART (J406)**
  - ➤ **While in the bootloader: Hold the ENTER button while starting AVRprog**

UART
JTAG
Parallel
ISP

- **JTAG**
  - ➤ **Programming and On-Chip Debugging**
  - ➤ **JTAG header (J402) is located on the "back side" of the PCB**
  - ➤ **Use an external power source when using JTAGICE due to increased power consumption**
  - ➤ **Always press "stop debugging" in AVR Studio to automatically disable the OCD fuse**

- **In System Programming (ISP)**
  - ➤ **STK500 Connect ISP6PIN to SPI bus (J403)**
  - ➤ **Note that ATmega169 uses PB0 to control the external data flash**

- **Parallel programming (High voltage)**
  - ➤ **STK500**

    On the Butterfly, move the 0ohm resistors: R404 to R403 and R203 to R204
    Make these connections between the STK500 and the Butterfly:

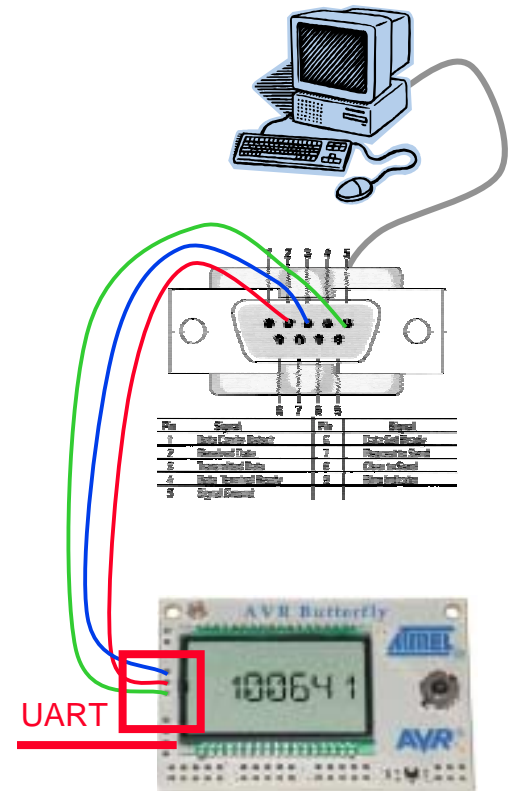| STK500	AVR Butterfly
PROGCTRL	J401 (PORTD)
PROGDATA	J400 (PORTB)
BSEL2.1	J402.8 (JTAG)
XTAL1.1	J402.10 (JTAG)
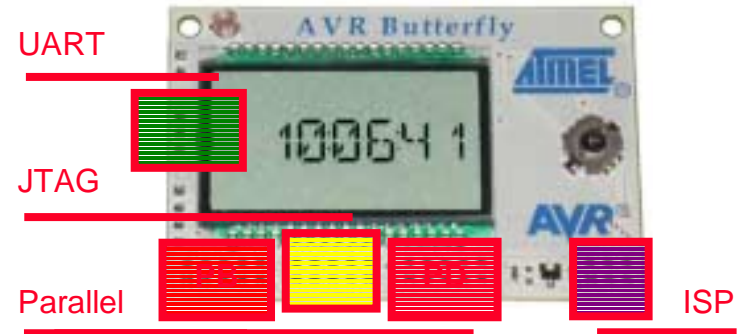
- **Written in IAR EWAVR**
- **All source code included**
- **Bootloader code**
- **Application code**
  - ➤ **State machine**
  - ➤ **Features included**
    - **Name-tag**
    - **Clock (date)**
    - **Temperature measurements**
    - **Light measurements**
    - **Voltage readings**
    - **Play tunes/melodies**
    - **Auto power save**
    - **Adjust LCD contrast**
  - ➤ **More functions that can be added later on....**
    - **Calculator**
    - **Reminder function**
    - **Alarm (daily alarms, kitchen-timers, etc...)**
    - **Play tunes/melodies (Karaoke-function)**
    - **With the 4Mbit dataflash one can store large amount of data. Some examples:**
      - **AVR Info bank (Basic info on all AVR-parts)**
      - **Your local bus-table**
      - **Melodies**
      - **+++**

- **Enables upgrade of the application code from a PC without any external hardware**
- **Based on the appnote AVR109: Self Programming AVR**
  - ➤ **But uses the new buffer mode for more efficient data downloading**
- **Uses AVRprog as PC frontend**
  - ➤ **Included with AVR Studio**
- **Uses UART**
- **Uses the 1024Byte boot block size**

- **Operation**
  - ➤ **AVR fused to start in bootloader after Reset**
  - ➤ **Short cut J403 pin 5 & 6 to make a hardware reset (or jump to bootloader from your application code)**
  - ➤ **Bootloader goes straight into sleep**
    - **Power Save Mode**
  - ➤ **Hold ENTER button while starting AVRprog, to enable downloading of new application code over the UART.**
  - ➤ **Press UP on joystick to wake up from sleep and enter Application section**
  - ➤ **Lockbits are used to avoid self deletion**
   - **SPM is not allowed to write to the Boot Loader section**

UART

Use the joystick to navigate in the menu

- **UP/DOWN**
  Jump between the menu items
- **RIGHT**
  Enter a sub menu item
- **LEFT**
  Jump back one level
- **Center button is ENTER**
  Edit/play a menu item

AVR Butterfly			
Time	Clock	"12:35:45"	Adjust clock
	Date	"03:04:25"	Adjust date
Music	"Fur Elise"		
	"Mozart"		
	"Minuet"		
	"Sirene1"		
	"Sirene2"		
Name	"Your Name"	Enter name	
		Download name	
Temperature	"+26°C"		
Voltage	"2V3"		
Light	"ADC28A"		
Options	Display	Adjust contrast	
	Bootloader	Jump to Bootloader	
	Power Save Mode	Press Enter to sleep	
	Auto Power Save	Min 90	
		Min (...)	
		Min 05	
		OFF	

- **Connect RS-232 on PC to UART on Butterfly**
- **Use joystick to enter sub menu "Download name" and press ENTER to enable UART**
- **Start any PC terminal software**
  - ➢ **Use connection settings:**
    - **Baudrate: 19200**
    - **Databits: 8**
    - **Parity: None**
    - **Stop bits: 1**
  - ➢ **In terminal window:**
    - **<Type text>**
    - **Press ENTER (↵)**
- **The text is stored in eeprom and scrolls over the display**
- **Stores up to 25 characters**
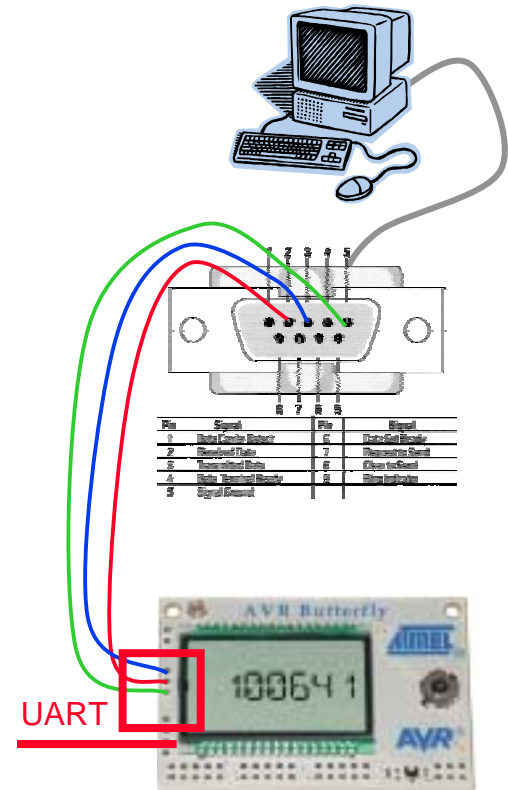
UART

Sleep modes are used to minimize power consumption

- **Normally Power Save Mode is used.
  But Idle Mode is used for:**
  - ➤ **Piezo-element (playing tunes)**
  - ➤ **UART communication**
- **In Power Save Mode**
  - ➤ **LCD controller is running (if enabled)**
  - ➤ **Asynchronous timer is running**

    Used for wake up for
    - • **LCD update**
    - • **RTC update**
- **"Auto Power Save"**
  - ➤ **Turns off the display controller before entering the sleep mode:
    Power Save Mode**

Power consumption and estimated lifetime

- **Playing tunes (incl. text)**
  - ➤ **Power consumption: approx 400uA**
  - ➤ **Lifetime (playing 24 hours a day): approx 2 month**
- **Presenting text in the LCD**
  - ➤ **Power consumption: approx 35-40uA
    (depends on whether the text is scrolling or not)**
  - ➤ **Lifetime (24 hours a day): over 1,5 year**
- **In power save mode (only the RTC ticking)**
  - ➤ **Power consumption: approx 9uA**
  - ➤ **Lifetime: approx 7 years**

Atmel AVR Butterfly