Building Interactive Devices and Objects

Prof. Dr. Michael Rohs, Dipl.-Inform. Sven Kratz
michael.rohs@ifi.lmu.de
MHCI Lab, LMU München
## Schedule

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Topic</th>
<th>Group Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.4.2012</td>
<td>Session 1: Introduction</td>
<td>Team building</td>
</tr>
<tr>
<td>2</td>
<td>26.4.2012</td>
<td>Session 2: Microcontrollers &amp; Electronics</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.5.2012</td>
<td>Session 3: Sensors</td>
<td>Concept development</td>
</tr>
<tr>
<td>4</td>
<td>10.5.2012</td>
<td>CHI</td>
<td>Concept development</td>
</tr>
<tr>
<td>5</td>
<td>17.5.2012</td>
<td>Christi Himmelfahrt</td>
<td>Concept development</td>
</tr>
<tr>
<td>6</td>
<td>24.5.2012</td>
<td>Session 4: Actuators</td>
<td>Concept presentation, Hardware requ.</td>
</tr>
<tr>
<td>7</td>
<td>31.5.2012</td>
<td>Session 5: Physical Objects (Sven)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7.6.2012</td>
<td>Fronleichnam</td>
<td>Project</td>
</tr>
<tr>
<td>9</td>
<td>14.6.2012</td>
<td></td>
<td>Project</td>
</tr>
<tr>
<td>10</td>
<td>21.6.2012</td>
<td></td>
<td>Project</td>
</tr>
<tr>
<td>11</td>
<td>28.6.2012</td>
<td></td>
<td>Project</td>
</tr>
<tr>
<td>12</td>
<td>5.7.2012</td>
<td></td>
<td>Project</td>
</tr>
<tr>
<td>13</td>
<td>12.7.2012</td>
<td></td>
<td>Evaluation</td>
</tr>
<tr>
<td>14</td>
<td>19.7.2012</td>
<td></td>
<td>Evaluation, Presentation</td>
</tr>
</tbody>
</table>
Trash Can

http://www.youtube.com/watch?v=cbEKAwCoCKw
Scratch Mat

http://www.youtube.com/watch?v=NfFzmRQriss
Stairs

http://www.youtube.com/watch?v=LNC5_17H-1A
POWER MANAGEMENT
AND SLEEP MODES
Power Management and Sleep Modes

• Critical for battery-operated devices
• Power consumption mainly depends on supply voltage, operating frequency, active modules
  – Save power by shutting down unused modules
  – Active supply current: 11 mA @ 8 MHz, 20 mA @ 16 MHz
• Sleep modes (currents for ATmega8 at 5V)
  – Idle 5 mA @ 8 MHz, 9 mA @ 16 MHz
  – ADC noise reduction
  – Power-down 1 µA
  – Power-save 15 µA
  – Standby 45 µA @ 1 MHz, 155 µA @ 6 MHz,
• Wake up by internal or external interrupts
Sleep Modes

• Idle
  – Stops the CPU, but interrupt system, USART, Analog Comparator, ADC, Timer/Counters, etc., continue operating

• ADC noise reduction
  – Stops the CPU, but ADC, Timer/Counter2, etc., continue
  – Improves the noise environment for the ADC

• Power-down
  – External oscillator stopped, external interrupts continue operating
  – Lowest power consumption, but longest wake-up delay
  – Wakes up from: external interrupt on INT0 or INT1, etc.

• Standby
  – Like power-down, but oscillator keeps running
  – Fast wakeup
## ATmega8 Wakeup Sources

<table>
<thead>
<tr>
<th>Sleep Mode</th>
<th>Active Clock Domains</th>
<th>Oscillators</th>
<th>Wake-up Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>clk\textsubscript{CPU}</td>
<td>clk\textsubscript{FLASH}</td>
<td>clk\textsubscript{IO}</td>
</tr>
<tr>
<td>Idle</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ADC Noise Reduction</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>
</tr>
<tr>
<td>Power Save</td>
<td>X(2)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Standby\textsuperscript{(1)}</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Notes:
1. External Crystal or resonator selected as clock source
2. If AS2 bit in ASSR is set
3. Only level interrupt INT1 and INT0

Source: Atmel Datasheet
Minimizing Power Consumption

- Port pins should not drive resistive loads (see p. 36, 55f)
- Switch off analog comparator (p. 187)
- Switch off ADC (p. 200)
- Do not use brown-out detector
- Do not use watchdog timer

- Enable INT0 or INT1 to wake up from power-down
- Enter sleep mode with “sleep” instruction
- Continue after “seep” instruction after wake up
Example Code for Power-Down (ATmega8)

• Set interrupt service routine for INT0
  ISR(INT0_vect) {
    cli(); // disable further interrupts
  }

• Sleep routine (enable INT0 to wake up from power-down)
  void my_power_down () {
    cli(); // disable interrupts
    ACSR = (1 << ACD); // switch off analog comparator
    ADCSRA = (0 << ADEN); // switch off ADC
    DDRD &= ~0b100; // PD2 (INT0) as input
    PORTD |= 0b100; // enable pullup resistor on PD2 (INT0)
    ...
Example Code for Power-Down (ATmega8)

... MCUCR = (0 << ISC00); // low level on PD2 (INT0) generates interrupt GICR = (0 << INT1) | (1 << INT0); // enable external interrupt INT0 sei(); // enable interrupts MCUCR |= (1 << SE) | (2 << SM0); // sleep enable, power-down mode asm volatile ("sleep"); // assembler instruction for entering sleep mode cli(); // woken up, disable interrupts GICR = (0 << INT1) | (0 << INT0); // disable external interrupts sei(); // enable interrupts

}
WLAN MODULE
Roving RN-XV WLAN Modules

- Simple communication via WLAN
  - Roving RN-171 WiFi chip
  - UDP, TCP, HTTP, FTP
  - rovingnetworks.com/products/RN_XV

- Serial I/O to WLAN module

- Requires 3.3V power supply

- Connections
  - Pin 1: 3.3V power supply (use 3.3V voltage regulator)
  - Pin 2: TX (connect to RX of ATmega8, direct)
  - Pin 3: RX (connect to TX of ATmega8, via voltage divider!)
  - Pin 10: GND (connect to common ground)
Roving RN-XV WLAN Modules

• Pin 3 of RN-XV (RX): connect to TX of ATmega8
  – via voltage divider!
  – TX of ATmega8 uses +5V
  – RX of RN-XV expects +3.3V

• Example: $R_1 = 3000 \, \Omega$, $R_2 = 1500 \, \Omega$
  – $U_1 = 5V \times \frac{R_1}{R_1 + R_2} = 3.33V$
  – can also use $R_1 = 4400 \, \Omega$, $R_2 = 2200 \, \Omega$
USB-to-Serial Converter

• Serial side (RS232)
  – RTS (request-to-send, green)
  – RX (receive, yellow)
  – TX (transmit, orange)
  – 5V (red)
  – CTS (clear-to-send, brown)
  – GND (black)

• Virtual COM Port Drivers
  – www.ftdichip.com/Drivers/VCP.htm

contains FDTI FT232RQ chip
FTDI FT232RQ Virtual COM Port Drivers

- Virtual COM port drivers
  - http://www.ftdichip.com/Drivers/VCP.htm

- USB device appears as virtual COM port
  cd /dev
  ls -l | grep usb
cu.usbserial-A100OXPZ
tty.usbserial-A100OXPZ

- Shows up in System Profiler
Roving-Modul (3.3V!)
Connect Roving Module via USB
Roving RN-XV Commands

• Start terminal: screen /dev/tty.usbserial-A100XZ 9600
• Enter command mode: $$$
• Initialize
  factory RESET
  reboot
  set wlan ssid MYSSID
  set wlan pass 12345678
  set wlan join 1
  save
  reboot

• Get information
  ver, get ip, get adhoc, get com, get dns, etc.

Details:
rovingnetworks.com/products/RN_XV