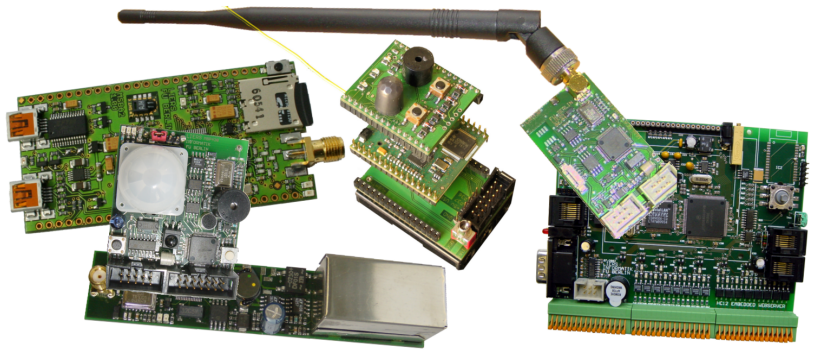


19579 - PR: Programming Wireless Sensor Networks

Introduction

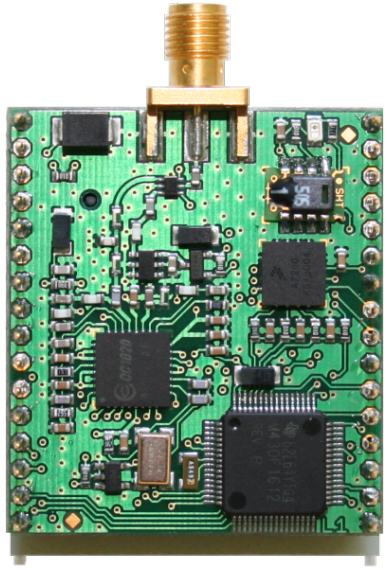
Oliver Hahm, Kaspar Schleiser

Department of Mathematics and Computer Science
Institute of Computer Science
October 20th, 2010

Outline

1. Embedded Systems
2. Hardware
3. Software
4. Toolchain
5. Examples
6. Further Reading

- Slow CPU
- Little RAM and (Flash) ROM Memory
- Mostly no MMU
- Limited energy resources
- FPU might be missing
- Specialized and very basic OS
- Difficult to debug

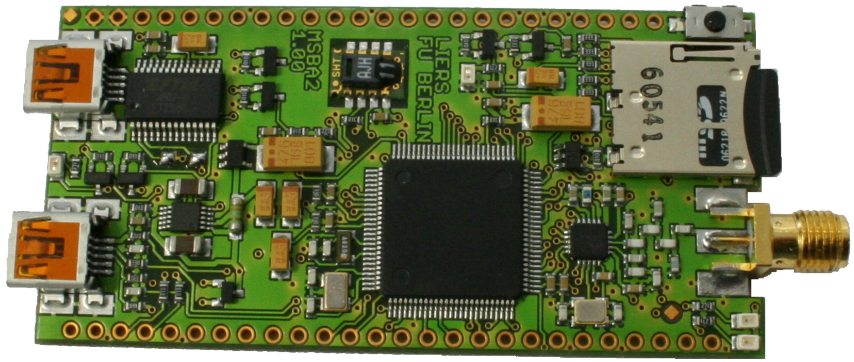


Applications of Wireless Sensor Networks

- home and building automation
- industrial automation
- asset management and logistics
- personal health and fitness
- improved energy efficiency and smart metering
- real-time environmental monitoring
- vehicular automation
- disaster zones, unaccessible or hazardous areas

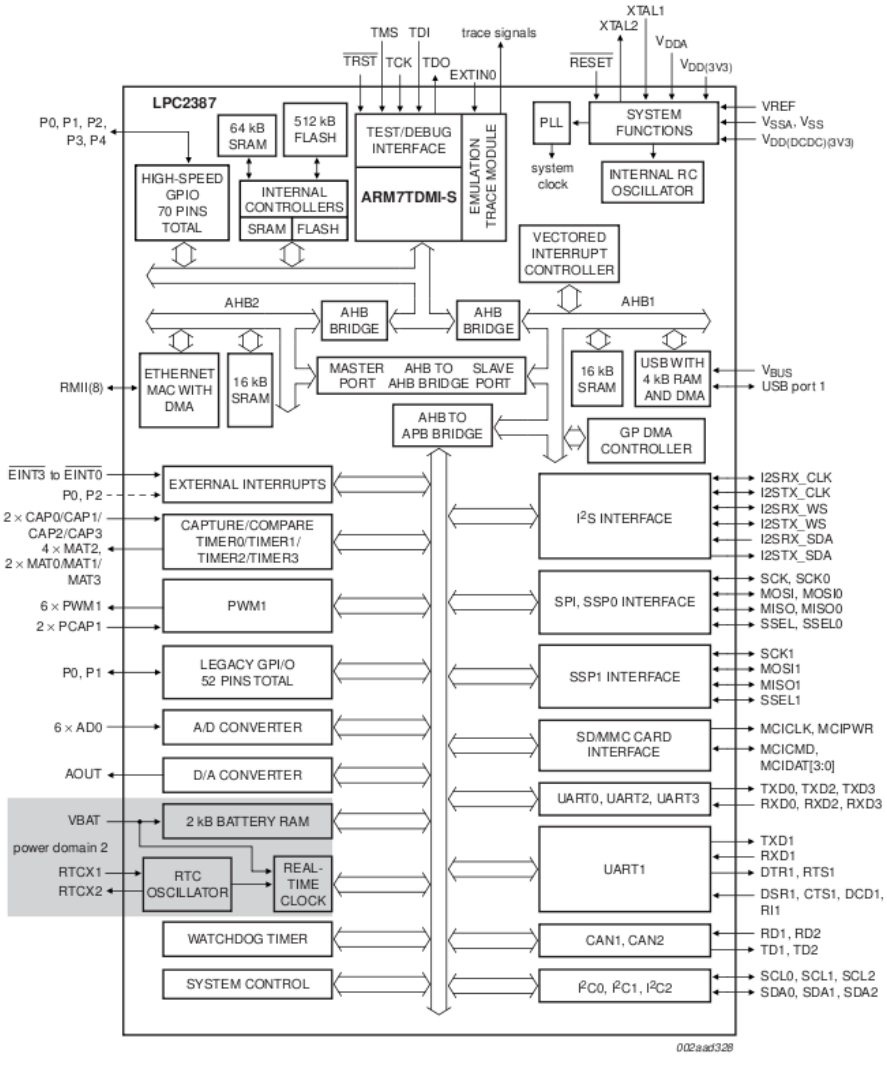
MSB-A2

- ARM 7 microcontroller
- CC1100 radio transceiver @ 868 MHz
- SHT11 temperature and humidity sensor
- Coulomb counter
- Micro SD card holder
- one button
- three LEDs
- two micro USB slave ports
- a lot of free IO pins



LPC2387 - ARM7TDMI-S

- up to 72 MHz
- 32 bit architecture
- 96 kB RAM
- 512 kB ROM
- Real Time Clock
- USB interface
- Ethernet MAC
- 4 UARTS
- SPI interface
- three I^2C interfaces



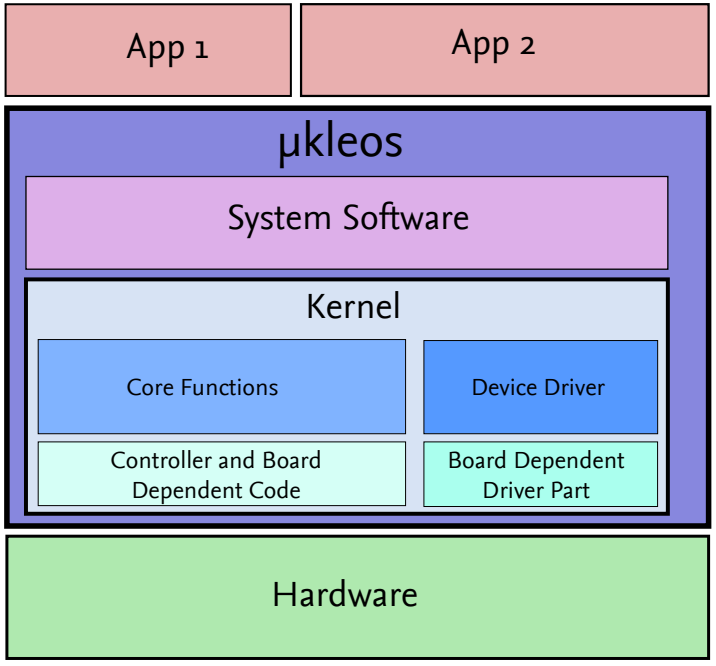
µkleos - Operating System

- Small operation system for microcontrollers
- Microkernel system
- Nonpreemptive scheduler
- Modular design, easy portable
- Existing ports for:
  - MSB-A2
  - MSB-430H
  - eZ430-Chronos
- Documentation needs work

http://www.ukleos.org

Software Architecture

- Application code in /projects/$APP
- System functions in /sys/
- Board independent part of driver code in /driver/$DEVICE
- Board dependent and board dependent driver code in /board/$BOARD
- Controller dependent code in /cpu/$CPU
- Hardware independent core functions (scheduler, messaging, mutexes . . .) in /core/

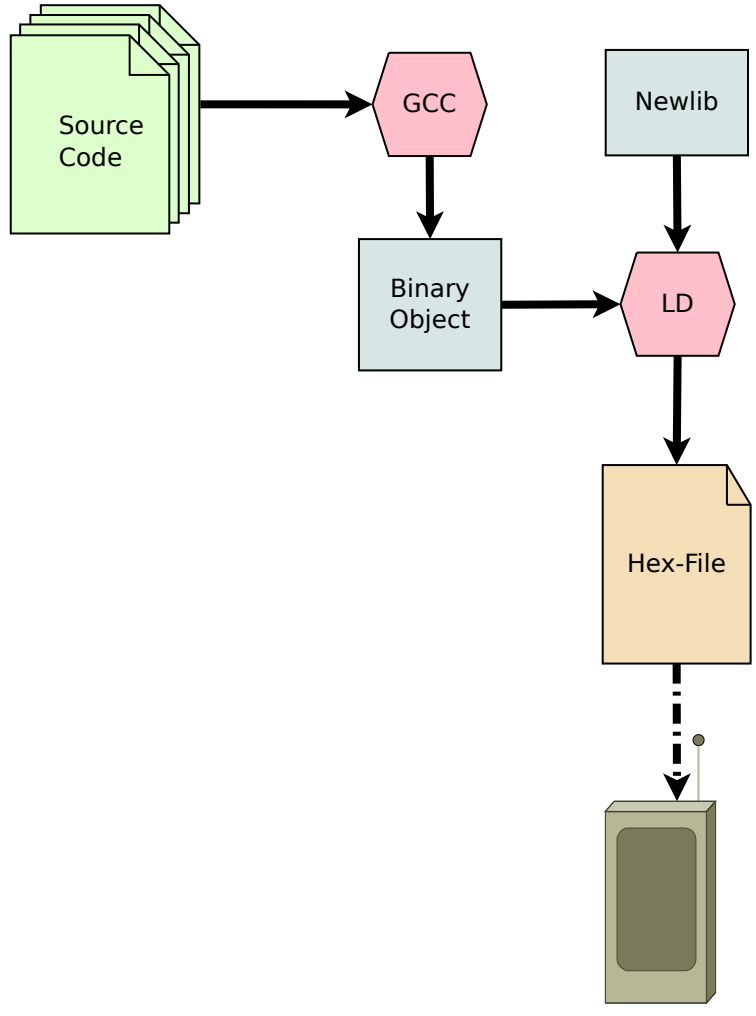


Toolchain

GNU ARM/Newlib toolchain

- GNU binutils
- GNU compiler set (GCC)
- GDB and Insight debugger
- Newlib c library

So generated hex files are programmed onto the microcontroller using serial line and ARM bootloader.



Example 1 - LED blinker


```
#include <stdio.h>
#include <board.h>
#include <hwtimer.h>

int main(void) {
    LED_RED_ON;
    hwtimer_sleep(1000 * 1000);
    printf("Hello World!\n");
    LED_RED_OFF;
    LED_GREEN_ON;

    return 0;
}
```

Example 2 - Timer


```
#define MSEC  (1000)
#define SEC   (1000 * MSEC)

void blinker(void *flag);

swtimer_t t1;

int main(void) {
    hwtimer_init();
    swtimer_init();
    swtimer_usleep(3 * SEC);
    LED_RED_ON;
    swtimer_set_cb(&t1, (3 * SEC), blinker, NULL);
    return 0;
}

void blinker(void *flag) {
    LED_GREEN_TOGGLE;
    LED_RED_TOGGLE;
    swtimer_set_cb(&t1, SEC, blinker, NULL);
}
```

Example 3 - Messaging


```
int main(void) {
    msg m;
    int pid = thread_create(KERNEL_CONF_STACKSIZE_MAIN, PRIORITY_MAIN-1,
                CREATE_WOUT_YIELD | CREATE_STACKTEST, second_thread, "pong");
    m.content.value = 1;
    while(1) {
                msg_send(&m, pid, true);
                msg_receive(&m);
                printf("Got msg with content %i\n", (int) m.content.value);
    }
}

void second_thread(void) {
    msg m;
    while(1) {
        msg_receive(&m);
        printf("2nd: got msg from %i\n", m.sender_pid);
        m.content.value++;
        msg_send(&m, m.sender_pid, true);
    }
}
```

Example 4 - Sensor reading


```
int main(void)
{
    sht11_val_t sht11_val;
    uint8_t success = 0;

    hwtimer_init();
    swtimer_init();
    sht11_init();
    while (1) {
        success = sht11_read_sensor(&sht11_val, HUMIDITY | TEMPERATURE);
        if (success) {
            printf("%-6.2f°C %5.2f%% (%5.2f%%)\n",
                    sht11_val.temperature,
                    sht11_val.relhum,
                    sht11_val.relhum_temp);
        }
        LED_RED_TOGGLE;
        swtimer_usleep(1000 * 1000);
    }
}
```

Further Reading

- **Embedded Programming**
  - J. Wiegelmann, Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, Hüthig, 2009
  - A. Sloss, D. Symes, C. Wright, ARM System Developer's Guide, Elsevier, 2004
- **General**
  - I. Akyildiz, Wireless Sensor Networks, Wiley, 2010
  - A. Tanenbaum, Modern Operating Systems, Pearson Education, 2007
  - B. Kernighan, D. Ritchie, C Programming Language, Prentice Hall International, 1988

End of the First Steps