

# Das Carrierpigeon-Projekt

Julius Adorf, Marek Kubica, Hong-Khoan Quach

Technische Universität München

13.11.2009

-

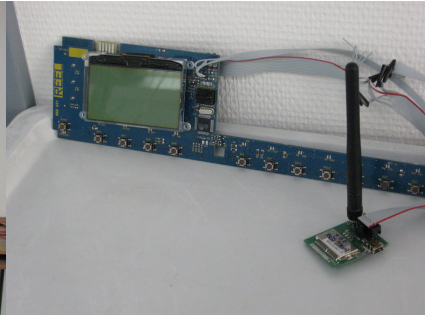
ETI-Projekt GP 8  
Sommersemester 2009

- Idee
- Komponenten
- Herausforderungen
- Ausblick

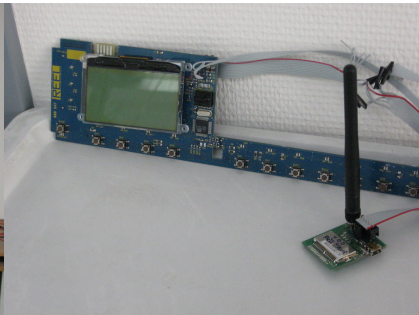
# Idee



# Idee

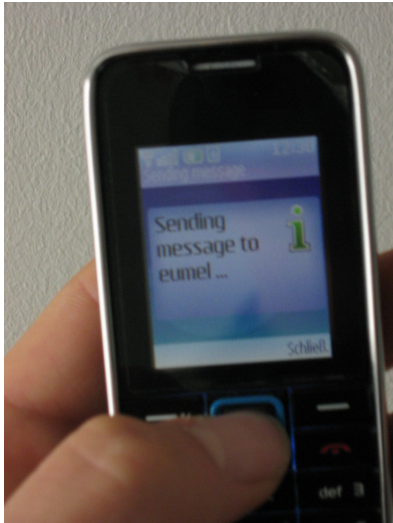


# Idee

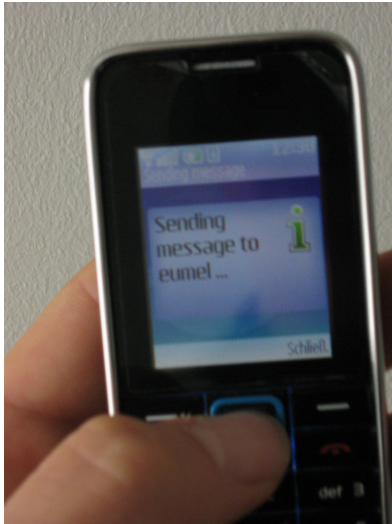




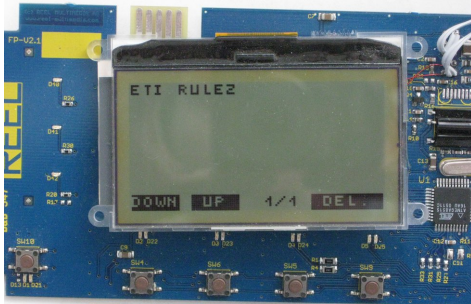










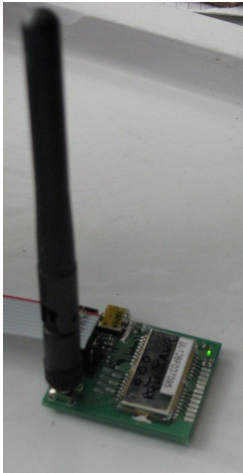


# Komponenten - Board



## Mainboard

- Mikrocontroller
- LEDs
- LCD
- Buttons



## Antenne & Controller

- über UART ansprechbar
- kennt AT/Modem-Befehle
- kümmert sich selbst um die Verbindung

- Bluetooth-Clients dürfen keinen Schaden anrichten
- Programm darf nicht abstürzen
- Neustart soll immer möglich sein

- der Server hat immer Recht
- Timeouts sorgfältig programmiert
- Globale Variablen vermieden
- While-Schleifen vermieden
- While-Schleifen überprüft

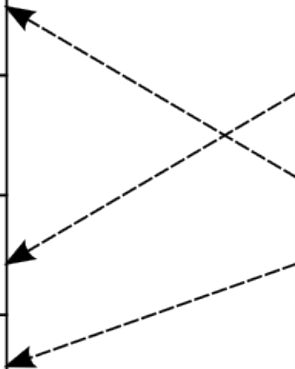
# Herausforderung - Speicherverwaltung

## EEPROM

0	Nachricht 1
120	Nachricht 2
240	Nachricht 3
360	Nachricht 4
480	Superblock

## Superblock

480	Nachrichtenzähler
481	Blockzeiger 1
482	Blockzeiger 2
483	Blockzeiger 3
484	Blockzeiger 4





- Nachrichtenzähler überprüfen
- Blockzeiger überprüfen
- Blockzeiger auf Duplikate überprüfen

Damit der Techniker nicht zum Hausbesuch kommen muss.

## Die Hardware

Flash	8 KB
EEPROM	512 Byte
SRAM	512 Byte

- ATmega8515, 8-Bit AVR-MCU
- 512 Byte = 512 Zeichen!
- Harvard-Architektur

## Bugs

- Stack und Heap kollidieren
- Unerklärliches Verhalten
- Heisenbugs

## Diagnose

- `avr-size` gibt Gesamtspeicher aus
- `avr-nm` gibt Speicherverbrauch granular aus
- Distanz von Stack (SP) und Heap (`&_heap_start`)  
ermittelbar

# Herausforderung - Speicher sparen

## Dynamische Speicherverwaltung

- Nur Strings allokieren die wir brauchen
- Funktioniert nicht, Heisenbugparade

## Buffer sharing

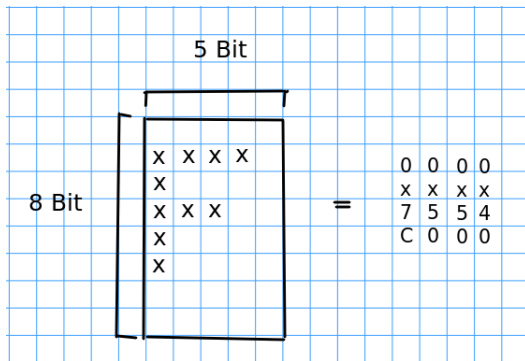
- Vorher: jeder Programmteil eigene Buffer
- Nachher: alles über globalen Buffer

## Harvard ausnutzen

- Zeichenmapping verbraucht viel RAM (235 Bytes)
- PROGMEM-Erweiterung des AVR-GCC

# Herausforderung - LC-Display und Textausgabe

- Punkt ausgeben
- Buchstaben (8x5 Matrix)



- Text verkehrt rum

⇒ Handbücher ganz genau lesen



- Protokollerweiterung
- Mehr Speicher einbauen
- Korrekte Zeilenumbrüche
- Automatische Tests mit GNU Debugger und AVR-Simulator

## Alternative Entwicklung:



An den Erfolg des digitalen Türschilds glaubt Gabi Dreo-Rodosek, Professorin der Bundeswehr-Uni in Neubiberg. Foto: Claus Schunk

## Infos statt Rätsel

### Bundeswehr-Uni arbeitet an „digitalem Türschild“

**Neubiberg** ■ Jürgen Sauerland steht nicht gern vor verschlossenen Türen. Besonders dann, wenn er nicht weiß, wann derjenige, den er aufsuchen will, wieder an seinen Schreibtisch zurückkehrt. Als Informatiker, der an der Bundeswehr-Universität in

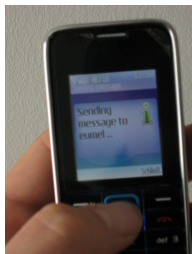
An den Türen der Informatiker an der Bundeswehr-Hochschule sind die Boards der Zukunft schon installiert. Der Besucher erfährt, dass der Professor gerade in der Vorlesung ist, aber in zehn Minuten zurück sein will. Über biometrische Daten mit Hilfe eines Ein-

(Süddeutsche Zeitung, 12.11.2009)



- Anforderungen erfüllt!
- Einsatz verschiedener Techniken ...
- ... was am Ende auch funktioniert hat!





Wer es ausprobieren mag:

- <http://bitbucket.org/jeadorf/carrierpigeon/downloads>
- Mobile-Client (JAR und evtl. JAD) herunterladen und auf das Handy spielen

Hardware	ATmega 8515, BTM-222, ST7565 LCD, AVRISP2
Languages	C, Java, Python
Framework	Java Micro Edition, Peter Fleury's UART library
Compilation	avr-gcc, avr-objcopy, avr-strip, avrdude, avr-nm, avr-size, splint, python, ant
Helpers	hcitool, rfcomm, gtkterm, jpneulator