

Ethernet mit Mikrocontrollern

Wie funktioniert TCP/IP mit 2kB RAM?

28. Dezember 2006

<http://lochraster.org>

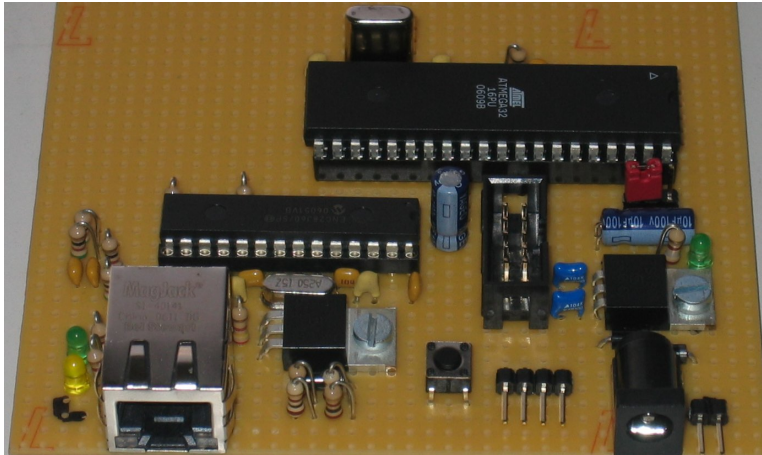
- Alexander Neumann (fd0)
 - Studiert Informatik an der RWTH Aachen
 - Idee zum Etherrape
- Michael Schwab (shorty)
 - Selbstständig/eigene Firma de-SOLUTION
 - Konzeption der Anwendungen
 - Finanzierung

- Messen (Blumenwasserstand, Temperatur)
- Steuern (Steckdosen, Licht)
- Regeln (Heizungen)

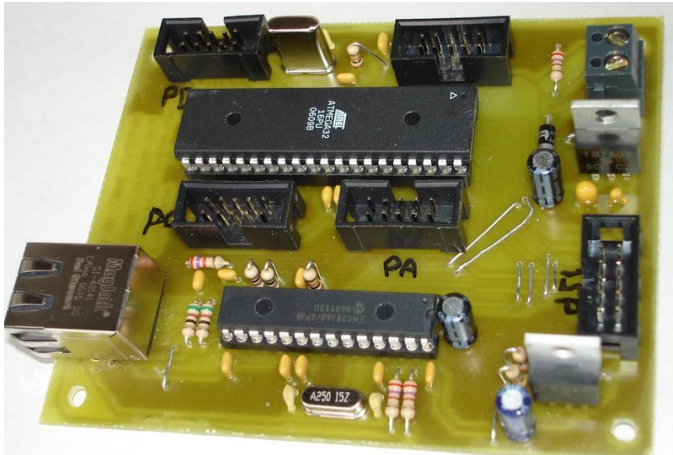
Warum nicht selber bauen?

- 10Mbit
- Atmel ATmega644
- Deshalb: Freie Toolchain (avr-gcc)
- Freie Firmware (GPL, TCP/IP Stack: BSD)
- Keine SMD-Teile (fast)

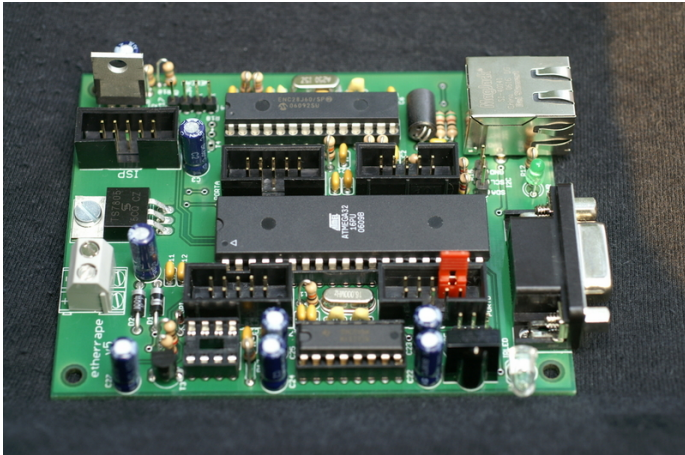
"etherrape"-Prototyp (v1):



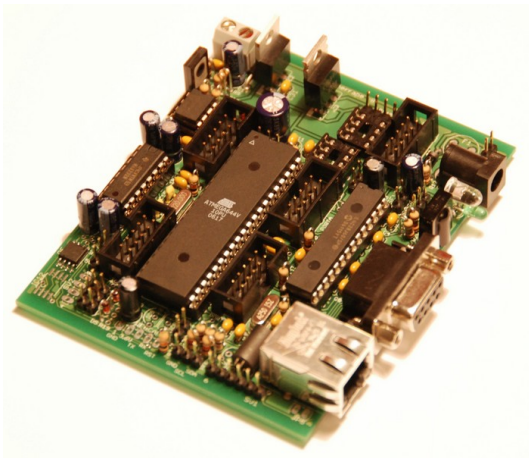
"etherrape"-Prototyp (v2):



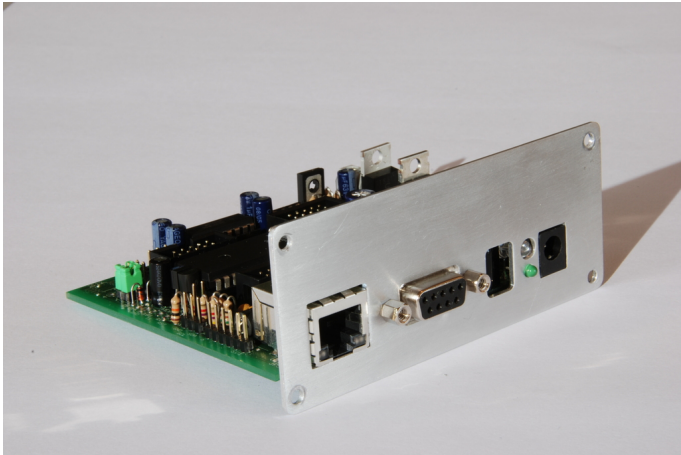
"etherrape"-Prototyp (v5):



"etherrape"-Prototyp (v8):



Aktuell (v9):



- 64KB FlashROM
- 4KB RAM
- 2KB EEPROM

- Ethernet
- RS232/RS485/RS422
- Infrarot (RC5 senden/empfangen)
- Webcam
- Dallas 1-Wire (Temperatursensoren)
- 2MB Dataflash

- Daten erfassen
 - Temperatur / Luftfeuchtigkeit (1-Wire)
 - Blumenwasserstand (Yeah!)
- Dinge steuern
 - Fernseher etc (IR)
 - fNordlichter (RS485, → Lounge)
 - Server (RS232)
 - Steckdosen (FS20, per Funk oder Kabel)
 - anderer FS20-Kram

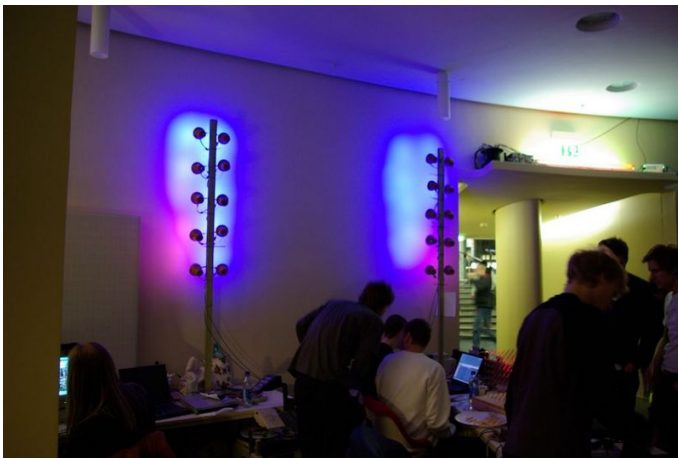
- TCP Retransmits
- IP Packet Reassembly
- Out-Of-Order Reception

Zunächst Implementation eines eigenen TCP/IP Stacks, später Portierung von uIP.

- Extrem klein
- Unterstützung von UDP und TCP
- Vorbereitet: IPv6
- Auf wenig RAM ausgelegt (minimal: 400 Byte)

- Kodiere Verbindungsstatus
- Reproduziere Daten auf Anfrage
- Kein Unterschied zwischen send und retransmit

```
void application(void) {
    if(uiplib_aborted()) {
        aborted();
    }
    if(uiplib_closed()) {
        closed();
    }
    if(uiplib_connected()) {
        connected();
    }
    if(uiplib_acked()) {
        acked();
    }
    if(uiplib_newdata()) {
        newdata();
    }
    if(uiplib_rexmit() ||
        uip_newdata() ||
        uip_acked() ||
        uip_connected() ||
        uip_poll()) {
        senddata();
    }
}
```



```
echo -n -e '\x05\x02\x00\xff\x00' \  
| socat -t 0 - UDP4:etherrape.lochraster.org:2323
```

```
echo -n -e '\x00\x03\x01\xff\x00\x00\xff' \  
| socat -t 0 - UDP4:etherrape.lochraster.org:2323
```

- Adresse (0 Broadcast, 1-20 Lampen)
- Opcode (0x02 Farbe setzen, 0x03 Faden)
- (bei Fade: 2 Byte Geschwindigkeit)
- RGB-Farbcode

- Man kann nette Hardware noch selber bauen
- Mikrocontroller sind cool
- TCP/IP mit 2kB RAM ist möglich!

Fragen?

Vielen Danke fürs Zuhören!

Infos und Webshop:

<http://www.lochraster.org/etherrape>