

EDEN

Entropias Daten- und Elektrizitäts-Netz

florolf

Entropia

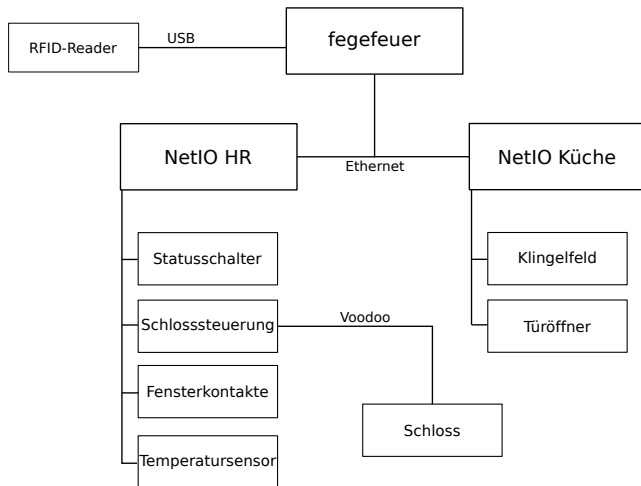
18. November 2012

1 Die Vergangenheit

2 EDEN

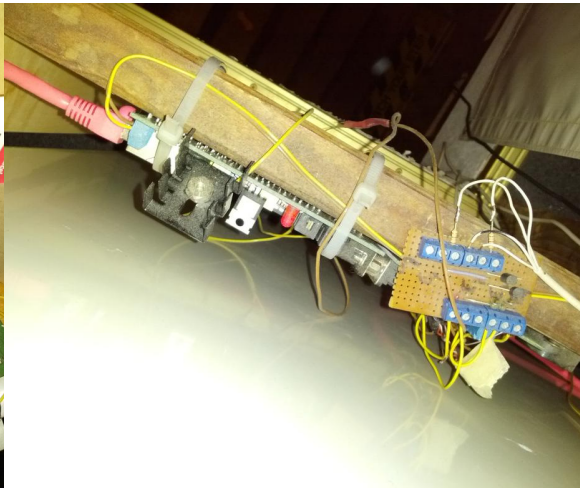
- Hardware
- CAN-Bus
- Software
- Die Gegenwart
- Die Zukunft

- Clubstatus
- Türschloss
- Türöffner unten
- Fensterüberwachung



- 10-MBit-Ethernet
- 8x Digital-Out
- 4x Digital-In
- 4x ADC

- 10-MBit-Ethernet
- 8x Digital-Out
- 4x Digital-In
- 4x ADC
- Default-Firmware: Pollen, I/O fix
- Ethersex: Instabil



- Daten (CAN)
- Elektrizität (permanent: 5V, geschaltet: 5V/12V)

ADAM Analog-/Digital-Adapter-Modul (heute: iomodul)

EVA Einspeise- und Verwaltungs-Anlage (kontrollsucht + busmaster + Industrienetzteil)

APFEL AbschlussPunkt Für Eden-Leitungen

BAUM BusAusweiter Und -Multiplikator (s.U.)

SERPENS SERPENS ist Entropias Rundum Praktischer EDEN-NetzStecker

5V permanent Für Basisinfrastruktur (z.B. Statusschalter, Türschloss)

5V permanent Für Basisinfrastruktur (z.B. Statusschalter, Türschloss)

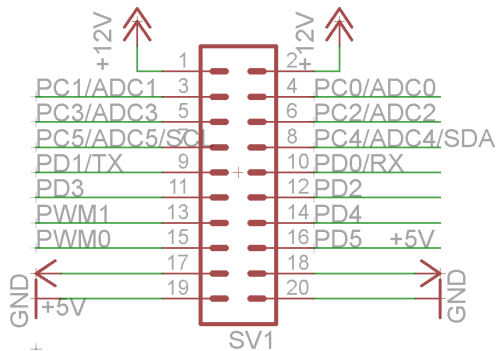
5V geschaltet Optionale Kleinelektronik

12V geschaltet Hochspannung!

- Dicke Verlegekabel im Kriechgang
- Dünnerer Strang am Balken
- Maschen über Querbalken (\Rightarrow bessere Spannungsverteilung)
- Netzspannung überprüft durch iomodule



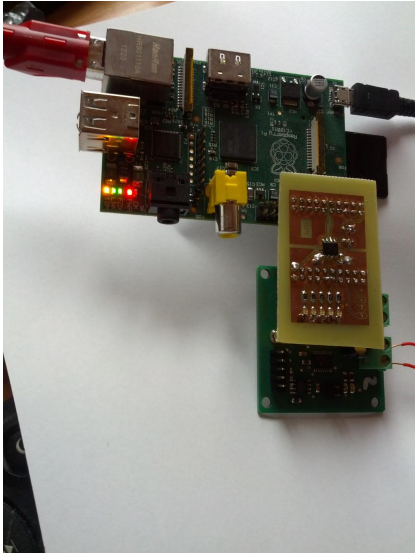
- ATmega88a (8k Flash, 1k RAM, 512 Bytes EEPROM), 20 MHz
- MCP2515
- (MCP2551)



- 13.5 GPIOs
- USART
- I2C
- 6x ADC
- 2x PWM
- MCP2515: 3x In, 2x Out (iomodule: 2 Jumper, 2 LEDs)

Busmaster

aka EVA

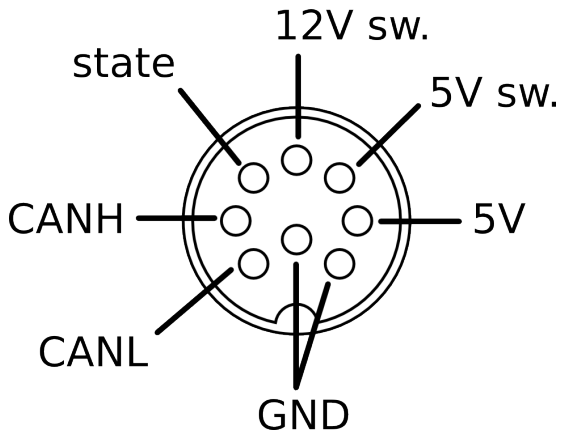


- iomodul + Aufsatz = Busmaster
- SLCAN (s.U.)
- Stromversorgung schalten

- 1983 von Bosch entwickelt. Heute hauptsächlich Automobilindustrie.
- Multi-master, Fehlerkorrektur, Priorisierung
- Adressen 11/29 bits
- Bis zu 8 Byte Payload

- Bus → Terminierung (APFEL)
- Stichleitungen: ungern (BAUM)

- Bus → Terminierung (APFEL)
- Stichleitungen: ungern (BAUM)
- Topologie bei uns: Bewegter Vortrag!
- Problem: Verkabelung/Platz
 - Aufputzdosen?
 - Hutschienegehäuse?



- Spannung permanent
- Spannung geschaltet
- CAN
- Clubstatus
- weiblich

- Nur EIDs
- 125 kBaud
- Gerichtete Kommunikation + Events
- 12bit Node-IDs
- Port 0: Management

0 type: direct	111111111111 destination (12bit)	111111111111 source (12bit)	1111 port (4bit)
1 type: event	0 globally assigned	11111111111111111111111111111111 event ID (27 bit)	
1 type: event	1 locally assigned	111111111111 by node	1111111111111111 event ID

```
#include "lib/can/can.h"

void app_configure(void) {}
void app_init(void) {}
void app_loop(void) {}
uint8_t app_handle(struct can_message *m) { return 0;}
```

- ADAM
- DALI (ERLEUCHTUNG)
- IR
- RS232-Adapter

- Remote-Flashen der Firmware
- Kleiner (440 Byte) Bootloader¹
- Vorgehen:
 - 1 make deploy
 - 2 Jumper 1 setzen
 - 3 `mgmt slcan0 reset NODEID`
 - 4 `\o/`

¹<https://github.com/florolf/canboot>

- Linux-Framework für CAN
- Unterstützt div. CAN-Interfaces
- Bei uns: SLCAN (CAN über TTYs)
- Multiplexen zwischen mehreren Anwendungen
- Low-Level-Bindings für Python, Java²
- C-Wrapper: libeden

```
root@kontrollsucht:~$ ip l
[...]  
3: slcan0: <NOARP,UP,LOWER_UP> mtu 16 qdisc pfifo_fast state U  
NKNOWN mode DEFAULT qlen 10  
    link/can
```

²<https://github.com/entropia/libsocket-can-java>

SocketCAN `dpipe ssh can@kontrollsucht.intranet.entropia.de`
`= canpipe slcan0`

socketcand³ Userspace-only mittels TCP

API z.B. `http://kontrollsucht.intranet.entropia.de/`
`api/verstaerker/`

³<https://github.com/dschanoeh/socketcand/blob/master/doc/protocol.md>

- Kleinbus: Verstärkersteuerung

- Kleinbus: Verstärkersteuerung



- Nächster Schritt: Verkabelung hinbekommen

- NetIOs ersetzen
- DALI
- Heizungssteuerung
- Schaltbare Steckdosen
- Rauchmelder ankoppeln
- Mädefüllstandsanzeige
- Spülmaschinenstatus
- Kelleraufzug
- Collect ALL the data
 - Temperatur
 - Luftfeuchtigkeit
 - Luftqualität (CO, NO_x, Drogen)
 - Stromverbrauch
- \$ELEKTRONIKPROJEKT?
- ...?

Fragen?