

Gerätetreiber für UART auf ATmega128

Jan Helber. Florian Steinle. Lars Mildner

24. Oktober 2006

Inhaltsverzeichnis

- 1 Einleitung
 - Inbetriebnahme
- 2 Geräte Treiber (DeviceDriver)
 - Funktionen INIT
 - Funktionen RECV
 - Funktionen SEND
- 3 Ende

Inbetriebnahme

- ATmega richtig in Sockel stecken
- Programmierverbindung stecken

Inbetriebnahme

- ATmega richtig in Sockel stecken
- Programmierverbindung stecken
- UART Logic Level Converter nicht vergessen

Inbetriebnahme

- ATmega richtig in Sockel stecken
- Programmierverbindung stecken
- UART Logic Level Converter nicht vergessen
- FUSES (103C, Taktung)

Inbetriebnahme

- ATmega richtig in Sockel stecken
- Programmierverbindung stecken
- UART Logic Level Converter nicht vergessen
- FUSES (103C, Taktung)

Funktionen INIT

- `uart0_init(BAUD_RATE, databits, stopbits, parity, FREQ_CPU)`
parity: EVEN, ODD, DISABLED
- `io_init()`

Funktionen INIT

- `uart0_init(BAUD_RATE, databits, stopbits, parity, FREQ_CPU)`
parity: EVEN, ODD, DISABLED
- `io_init()`

Funktionen RECV

- `char uart0_get_char()`
- `int uart0_get_chararray(text)`

Funktionen RECV

- `char uart0_get_char()`
- `int uart0_get_chararray(text)`
- `uart0_waitack()`

Funktionen RECV

- `char uart0_get_char()`
- `int uart0_get_chararray(text)`
- `uart0_waitack()`
- `int uart0_get_chararray_command(text)`

Funktionen RECV

- `char uart0_get_char()`
- `int uart0_get_chararray(text)`
- `uart0_waitack()`
- `int uart0_get_chararray_command(text)`
- `int uart0_recv_complete()`

Funktionen RECV

- `char uart0_get_char()`
- `int uart0_get_chararray(text)`
- `uart0_waitack()`
- `int uart0_get_chararray_command(text)`
- `int uart0_recv_complete()`

Funktionen SEND

- `uart0_send(text, size)`
- `bool uart0_trans_complete()`

Funktionen SEND

- `uart0_send(text, size)`
- `bool uart0_trans_complete()`

Ende

weitere Infos

ausführliche Doku zur Inbetriebnahme

Vielen Dank

für Eure Aufmerksamkeit

Powered by

L^AT_EX