

# C

- [Anwendungen](#)
  - [COM-Port unter Linux ansteuern](#)

# Anwendungen

# COM-Port unter Linux ansteuern

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>

int com_port;
//char get_buffer[9];
const char* get_buffer;

int init_com() {
    struct termios options;
    //COM1
    com_port = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
    if (com_port == -1) {
        printf("send_to_com: Unable to open /dev/ttyS0!\n");
        return -1;
    } else {
        fcntl(com_port, F_SETFL, 0);
    }

    //get the current options for the port
    tcgetattr(com_port, &options);

    //set the baud rates to 4800
    cfsetispeed(&options, B4800);

    cfsetospeed(&options, B4800);

    //enable the receiver and set local mode
    options.c_cflag |= (CLOCAL | CREAD);

    //set the new options for the port
    tcsetattr(com_port, TCSANOW, &options);

#ifdef DEBUG
```

```
    printf("/dev/ttyS0 was opened successfully!\n");
#endif
    return 0;
}

void send_to_com(char buffer[9]) {
    write(com_port, buffer, 9);
#ifdef DEBUG
    printf("wrote %s to /dev/ttyS0\n", buffer);
#endif
}

void get_from_com() {
    read(com_port, get_buffer, sizeof(get_buffer));
#ifdef DEBUG
    printf("read %s to /dev/ttyS0\n", get_buffer);
#endif
}

int main (void) {
    init_com();
    send_to_com("00??00??\r");
    get_from_com();
    printf("%s", get_buffer);
    return 0;
}
```