RN-Control und PAN3101

by Profanter Stefan

Dieser Artikel soll eine Anleitung dazu sein, wie man einen Maus-Sensor PAN3101 ans RN-Control Board anschließt und anschließend die Daten des Sensors auswerten kann.

Benötigte Hardware:

- RN-Control Board
- Maus-Sensor PAN3101 (oder ähnliche) mit LED und Linse
  Diese Teile kann man alle direkt aus einer Maus ausbauen: z.B. Die Revoltec LightMouse Portable (ca. 10€), welche bereits den PAN3101 Sensor beinhaltet
- Lötgerät zum anlöten der Kabel

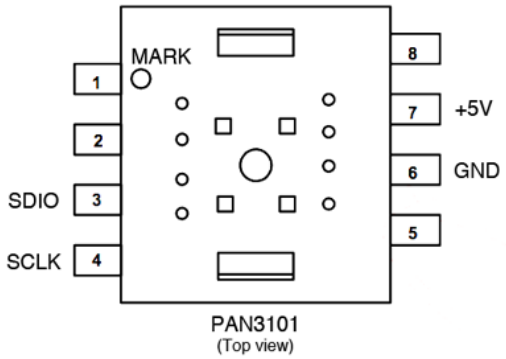
Vorbereitungen:

Zuerst sollte man die Maus aufschrauben. Dann am USB Anschließen und messen, mit welcher Spannung die LED betrieben wird, falls man diese später selbst ansteuern möchte und die Maus bereits zerlegt ist.

In unserem Fall beträgt die Spannung bei vollem Leuchten: 1,4V und bei mittlerem Leuchten: 0,866V

Kabel anlöten:

Zuerst werfen wir einen Blick auf die Anschlüsse des PAN3101

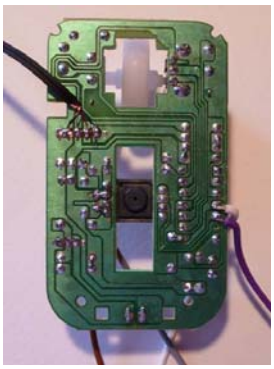
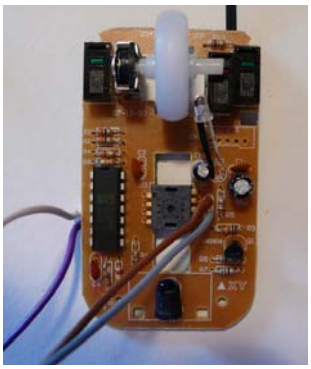


Am Pin 7 wird der Pluspol von 5V angeschlossen, am Pin 6 der Minuspol von 5V.

Der Pin 3 und 4 dienen zum Auslesen der Position.

Eigentlich genügen diese 4 Pins für uns. Am Port 5 kann zusätzlich die LED gesteuert werden. Denn wird die Maus bewegt, dann wird an Pin 5 eine Spannung von ca. 150mV angelegt. Man müsste dann diese Spannung über einen Transistor „verstärken“.

Also an den Pins 3, 4, 6, 7 jeweils einen Kabel anlöten. Dazu kann auch der USB-Kabel der Maus verwendet werden. Die Kabel müssen nicht direkt am PAN3101 angelötet werden, sondern können auch an einem anderen Ort auf der Platine angelötet werden, wobei aber hier eine Verbindung zum PAN3101 bestehen muss, wie hier im Bild:

Beim 2. Bild wurde einfach der Kondensator, welcher an der Position des grauen und braunen Kabels war, entfernt und diese Kabel hinein gelötet.

Braun = +5V
Grau = GND
Violett = SDIO
Weiß = SCLK

Anschließen am RN-Control Board

In unserem Fall wird der braune und graue Kabel beim 5V Ausgang des RN-Control Boards angeschlossen, wobei die Polung zu beachten ist: braun = +5V und grau = GND.

Violett wird am Port A angeschlossen, Weiß am Port C. Für die Pins habe ich beides mal den Pin 5 verwendet, also PA5 und PC5.

Programmieren des RN-Control Boards

Als Codegrundlage habe ich jenen vom Artikel „Maussensor“ aus dem Portal RN-Wissen verwendet: http://www.roboternetz.de/wissen/index.php/Maussensor

Hier der Code:
Wurde der Sensor an einem Anderen Port / Pin des RN-Control angeschlossen, müssen auch die #define zu Beginn des Codes verändert werden.
Die Datei „rncontrol.h“ kann im Roboternetz gefunden werden: www.roboternetz.de

```
#include <stdlib.h>
#include <avr/io.h>

#include "rncontrol.h"

#define DDR_SCK  DDRC                   /*!< DDR fuer Maus-SCLK */
#define DDR_SDA  DDRA                   /*!< DDR fuer Maus-SDA  */

#define PORT_SCK PORTC                  /*!< PORT fuer Maus-SCK  */
#define PORT_SDA PORTA                  /*!< PORT fuer Maus-SDA  */
#define PIN_SDA          PINA                   /*!< PIN fuer Maus-SDA  */

#define SCK_PIN          (1<<PC5)       /*!< PIN nummer fuer Maus-SCK  */
#define SDA_PIN          (1<<PA5)       /*!< PIN nummer fuer Maus-SDA  */


/*!
 * Uebertraegt ein Byte an den Sensor
 * @param data das Byte
 */
void pan_writeByte(unsigned char data){
        signed char i;

        DDR_SDA|= SDA_PIN;                      // SDA auf Output
```

```
        for (i=7; i>=0; i--){

                PORT_SCK &= ~SCK_PIN;            //SCK auf Low, Daten vorbereiten

                if(data&(1<<i)){         //Bit rausschieben
                        PORT_SDA|=SDA_PIN;
                }else{
                        PORT_SDA&=~SDA_PIN;
                }

                PORT_SCK |= SCK_PIN;             // SCK =1 Sensor uebernimmt auf steigender Flanke

                _delay_us(1);                    //Sensor Zeit lassen um Bit zu holen
        }

        DDR_SDA &=~ SDA_PIN;                     //HI-Z state
        PORT_SDA &=~ SDA_PIN;

}

/*!
 * Liest ein Byte vom Sensor
 * @return das Byte
 */
unsigned char pan_readByte(void){
signed char i;
unsigned char data=0;

        _delay_us(3);                            //Sensor Zeit lassen um die Daten aus dem Register zu holen

        for (i=7; i>-1; i--){
                PORT_SCK &= ~SCK_PIN;            // SCK =0 Sensor bereitet Daten auf fallender Flanke vor !

                _delay_us(1);                    //Sensor kurz zeit lassen


                PORT_SCK |= SCK_PIN;             // SCK =1 Daten lesen  auf steigender Flanke

                if(PIN_SDA&SDA_PIN){             //BIT einlesen
                        data |= (1<<i);
                }else{
                        data &=~ (1<<i);
                }

        }
        return data;
}

/*!
 * Uebertraegt ein write-Kommando an den Sensor
 * @param adr Adresse
 * @param data zu schreibendes byte
 */
void pan_write(unsigned char adr, unsigned char data){
        adr|=(1<<7);
        pan_writeByte(adr);  //rl MSB muss 1 sein für Write Operation
        pan_writeByte(data);
}



/*!
 * Schickt ein Lesekommando an den Sensor
 * und liest ein Byte zurueck
 * @param adr die Adresse
 * @return der registerwert
 */
unsigned char pan_read(unsigned char adr){

        pan_writeByte(adr);
        return pan_readByte();
}


/*!
 * Initialisiere PAN3101

 !! Muss unbedingt ganz am ANFANG von main stehen, sonst gibts FEHLER !!
 (wenn der PAN3101 sich initialisiert hat, bevor der Controler SCK und
 SDA auf Output gestellt hat)
 Deshalb kann es auch sinnvoll sein die Powerup Zeit in den Config Bits
 auf 4ms zu stellen oder noch besser mit Boden zu arbeiten.

 */
void pan_init(void){

        DDR_SCK  |= SCK_PIN;     // SCK auf Output
        DDR_SDA |= SDA_PIN;              //SDA auf Output

        PORT_SCK |= SCK_PIN;     // SCK auf high
        PORT_SDA|= SDA_PIN;              //SDA auf high

// hier müssen bei Umstellung auf PAN101 die entsprechenden Register gesetzt werden
//Reset PAN3101
```

```
        pan_write(0x00,0x80);
// kein Sleep modus
        pan_write(0x00,0x01);
}


int main(void){

unsigned char ino;
signed char x,y;
signed short posx=0,posy=0;

//ganz an den Anfang damit der Controller schneller asl der PAN ist um Fehler zu vermeiden
pan_init();
//Individuelle Port Configuration und Initialisierung
init_USART();    //USART konfigurieren

sendUSART("\r\n\n\n"); //Sendet einen kleinen Begrüßungstext. "\r" setzt den Cursor wieder auf Zeilenanfag, "\n"
beginnt dann die nächste Zeile
        sendUSART("**** Hallo! Dies ist ein PAN3101 Test *****\r\n");
        sendUSART("\r\n");



while(1){

        //Endlosschleife

        ino=pan_read(0x16);

        //wenn 7tes bit vom Register 0x16 gesetzt ist wurde die Maus bewegt => Bewegungsdaten abfragen
        if(ino&(1<<7)){
                //Deltax Register auslesen
                x=pan_read(0x17);
               //und zu der Positionvariable addieren
                posx=posx+x;

               /* Nachschaun ob das Überlauf-Bit im Register 0x16 gesetzt ist
                  wenn das der Fall ist muss je nach Vorzeichen der Deltax Variable x
                  noch 128 (überlauf nach oben) dazugezählt oder eben 128 abgezogen werden
               */
                if(ino&(1<<3)){
                        if(x<0){
                                posx-=128;
                        }else{
                                posx+=128;
                        }
                }

               //ab hier nochmal das Gleiche für die yRichtung

                y=pan_read(0x18);
                posy=posy+y;

                if(ino&(1<<4)){
                        if(y<0){
                                posy-=128;
                        }else{
                                posy+=128;
                        }
                }

        }

               //hier kann jeder seine Ausgabevariante selber wählen ;)

                sendUSART("Koord: ");
                sendUSARTInt(posx);
                sendUSART("|");
                sendUSARTInt(posy);
                sendUSART("\r\n");
                //lcd_ausgabe_int(0,3,posx);
                //lcd_ausgabe_int(7,3,posy);

}

return 0;
}
```

In Betrieb setzen:

Damit der Sensor auch richtig funktioniert, muss die Linse, welche bei der Maus eingebaut war, verwendet werden, ohne Linse funktioniert der Sensor nicht. So auch die rote LED!!!

Einfach das RN-Control Board mit dem oberen Code programmieren, unter Spannung setzen und dann die Daten verarbeiten.

Das ist bereits alles gewesen. Jetzt könnte man noch die Platine zurechtschneiden und ein eigenes Gehäuse bauen. Eurer Kreativität sind keine Grenzen gesetzt.

Vielen Dank der RN-Community für deren Unterstützung!!!