

Tasterabfrage mit PINC

Wenn ein Port mit Hilfe des zugehörigen Data Direction Registers (DDRx) als Eingang festgelegt wurde, dann kann der Zustand des Ports eingelesen werden und sein Wert als Variable gespeichert werden. Für das Einlesen eines Ports als digitalen Eingang werden die folgenden „Befehle“ genutzt:

`digitalRead(n);` oder

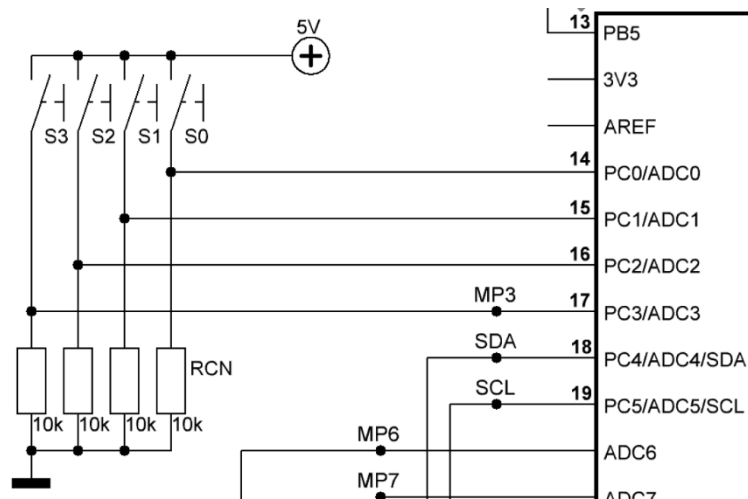
`PINA` für Port A;

`PINB` für Port B;

`PINC` für Port C;

`PIND` für Port D

Die vier Taster sind an den „unteren“ vier Bit des Ports C am Mikrocontroller angeschlossen.



Port C kann als digitaler Eingang genutzt werden. So kann der Wert von Port C in einer Variable vom Typ `int` oder `unsigned int` gespeichert werden.

Für das Einlesen des Ports C als digitalen Port wird das Register `PINC` genutzt.

Der nachfolgend dargestellte Quellcode ist für die Abfrage der vier Taster geeignet:

```
int taster;                // Deklaration des integers taster
taster = PINC & 0x0f;      // Einlesen Port C und Bit-Maskierung
// taster = PINC & 0b00001111; oder taster = PINC & 15; geht auch
```

Das bitweise Verknüpfen des binären Werts der Variable taster mit dem hex-Wert 0f:

Beispiel: 0 1 0 1 1 0 0 1 (Port C)
 Of_{hex} 0 0 0 1 1 1 1 // BITMASKIERUNG
 bitweise & 0 0 0 1 0 0 1 = 9_{dec} = 09_{hex}
 → S4 und S1 gedrückt, da an PC0 und PC3 5V anliegen.

Durch die vier „0“ des „higher nibble“ werden die zwei 1en an PC4 und PC6 unterdrückt. Das ist wichtig, da an PC4 bis PC6 Potentiometer angeschlossen sein können!!!

Wichtig! Merke:

Wenn die Tasterabfrage als Funktion realisiert ist, dann kann diese Funktion beliebig häufig von jeder Stelle des Programms aufgerufen werden. Das ist ein großer Vorteil!

Taster S1 bis S4 an Port C

Die Taster S0 bis S1 sind am nanoBoard an den Pins PC0 bis PC3 angeschlossen. Da die Tasterzustände eingelesen werden sollen, sind die entsprechenden Pins für den µC dann Eingänge. Das DDRC muss dann entsprechend beschrieben werden (1 bedeutet Ausgang / 0 bedeutet Eingang).

```
DDRC = 0b11110000;
DDRC = 0xf0;
DDRC = 240;                    oder    DDRC = 0;    (acht Eingangspins)
```

Die „Befehle“ zum Einlesen eines Ports sind:

```
int taster;
taster = PINC & 0x0f; // Auslesen & Bitmaskierung
```

Die Variable taster hat dann in Abhängigkeit von den gedrückten Tastern einen dezimalen Wert.

z.B.:	taster == 1	bedeutet	S ₀ gedrückt
	taster == 2	bedeutet	S ₁ gedrückt
	taster == 3	bedeutet	S ₀ und S ₁ gedrückt
	taster == 4	bedeutet	S ₂ gedrückt

Aufgaben

Aufgabe Taster.1:

Wenn ein Taster gedrückt wird, dann soll an Port B eine LED angehen.

Aufgabe Taster.2:

Wenn mehr als ein Taster gedrückt wird, dann sollen auch bis zu vier LEDs angehen. Jedem Taster ist eine bestimmte LED zugeordnet.

Aufgabe Taster.3:

Wenn S0 gedrückt wird, dann werden alle LEDs an Port D eingeschaltet. Wenn S3 gedrückt wird, dann werden alle LEDs an Port D ausgeschaltet. Wenn S2 gedrückt wird, dann wird von rechts beginnend immer eine LED mehr eingeschaltet. Wenn S1 gedrückt wird, wird jeweils eine LED wieder ausgeschaltet.

Aufgabe Taster.4:

Mit den vier Tastern soll ein Punkt auf dem OLED-Display gesteuert werden können (Vorarbeit für snake / OLED-Display ist Voraussetzung).

Aufgabe Taster.5:

Eine einzelne LED soll sich am Port D seitwärts bewegen. S1 und S2 steuern die Bewegungsrichtung. Mit S0 und S3 lässt sich die Bewegungsgeschwindigkeit beeinflussen.