

# Ports: Eingabe und Ausgabe

## 1. Ausgabe

- a. Beliebige Zahlenwerte können einfach auf das Output-Data-Register (ODR) eines Ports (GPIOx) ausgegeben werden

Beispiele:

```
GPIOC->ODR=0b10101010; //schaltet PC1, PC3, PC5, PC7 auf 1, die angeschlossenen LEDs sind eingeschaltet. PC0, PC2, PC4 und PC6, PC8..PC15 werden gleichzeitig auf 0 geschaltet, angeschlossenen LEDs sind aus.
```

- b. Einzelne Bits auf 1 oder 0 setzen.

```
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_5, 1) //setzt PC5 auf 1
```

```
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_5, 0) //setzt PC5 auf 0
```

```
HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_5) // negiert PC5 (0->1 bzw. 1->0)
```

## 2. Eingabe

- a. Beliebige Zahlenwerte können vom Input-Data-Register eingelesen und weiterverarbeitet werden.

Beispiel: (x ist eine Integer-Variable)

```
x=GPIOB->IDR;
```

```
if (x==0x55) {...}
```

- b. Nur die unteren 8 Bits des Input-Data-Registers enthalten einen gültigen Wert. Dann müssen diese 8 Bits maskiert werden.

Beispiel:

```
x=GPIOB->IDR&0xFF;
```

```
x=GPIOB->IDR&0b11111111;
```

Bei beiden Instruktionen bleiben die Bits 0 .. 7 unverändert, die Bits 8 .. 15 werden als 0 in x eingetragen:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	1	0	0	1	0	0	1	1	0	1	1	1	0	1	0
&	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
=	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0

- c. Einzelne Bits prüfen.

```
x=HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_5); //x wird 0 oder 1 je nach Wert von PB5
```

```
if (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_5) == 1) { Anweisungen für PB5=0 }
```

## 3. Weitere Codebeispiele

- a. `GPIOC->ODR=GPIOC->ODR+1;` //GPIOC->ODR wird um 1 erhöht
- b. `GPIOC->ODR++;` //GPIOC->ODR wird um 1 erhöht
- c. `GPIOC->ODR=GPIOC->ODR<<1;` //GPIOC->ODR wird um eine Stelle nach links verschoben
- d. `GPIOC->ODR=GPIOC->ODR>>3;` //GPIOC->ODR wird um 3 Stellen nach rechts verschoben
- e. `if ((GPIOC->ODR&GPIO_PIN_7)>0) HAL_GPIO_Write_Pin(GPIOA,GPIO_PIN_3);`  
//wenn Bit 7 von GPIOC\_ODR=1 dann wird Bit 3 von GPIOA->ODR auf 1 gesetzt.

4. Aufgaben: Schreiben Sie die Codezeilen in der Programmiersprache C
- a. Wenn PA1=1 dann an GPIOC die Zahl 35 ausgeben
  - b. Der Ausgabewert an GPIOC wird um 5 erhöht, wenn 25 überschritten wird, wird wieder bei 0 angefangen.
  - c. Nur die unteren 4 Bit von den Eingängen von GBIOB sollen auf GPIOC ausgegeben werden
  - d. Wenn PB3=0 ist soll an Bit 0 von GPIOC eine 1 ausgegeben werden
  - e. Setzen Sie die Bits 0,2,4,6 von GPIOC->ODR auf 1 und gleichzeitig die Bits 1,3,5,7 auf 0
  - f. Negieren Sie GPIOC->ODR bitweise
  - g. Wenn der Ausgang PC3=1 ist sollen die Ausgänge von GPIOA den Wert 0 annehmen.
  - h. Wenn PA2 und PA3 gleichzeitig 1 sind sollen die alle 16 Ausgänge von GPIOC auf 1 gesetzt werden.
  - i. Wenn PA1 = 1 wird dann GPIOC->ODR um 1 hochzählen. Danach warten bis PA1 wieder 0 wird
  - j. Warten auf PA10 =1, während gewartet wird soll der Mikrocontroller nichts tun
  - k. Warte auf Taste PA10=1, danach
    - Warte auf Taste PA6=1, danach
      - 1. Zähle GPIOC->ODR um 5 hoch