Ports: Eingabe und Ausgabe

1. Ausgabe
   1. 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.
   2. 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)

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

Beispiel: (x ist eine Integer-Variable)

x=GPIOB->IDR;

if (x==0x55) {…}

* 1. Nur die unteren 8 Bits des Input-Data-Registers enthalten einer 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 |

* 1. 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* }

1. Weitere Codebeispiele
   1. GPIOC->ODR=GPIOC->ODR+1; //GPIOC->ODR wird um 1 erhöht
   2. GPIOC->ODR++; //GPIOC->ODR wird um 1 erhöht
   3. GPIOC->ODR=GPIOC->ODR<<1; //GPIOC->ODR wird um eine Stelle nach links verschoben
   4. GPIOC->ODR=GPIOC->ODR>>3; //GPIOC->ODR wird um 3 Stellen nach rechts verschoben
   5. 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.
2. Aufgaben: Schreiben Sie die Codezeilen in der Programmiersprache C
   1. Wenn PA1=1 dann an GPIOC die Zahl 35 ausgeben
   2. Der Ausgabewert an GPIOC wird um 5 erhöht, wenn 25 überschritten wird, wird wieder bei 0 angefangen.
   3. Nur die unteren 4 Bit von den Eingängen von GBIOB sollen auf GPIOC ausgegeben werden
   4. Wenn PB3=0 ist soll an Bit 0 von GPIOC eine 1 ausgegeben werden
   5. Setzen Sie die Bits 0,2,4,6 von GPIOC->ODR auf 1 und gleichzeitig die Bits 1,3,5,7 auf 0
   6. Negieren Sie GPIOC->ODR bitweise
   7. Wenn der Ausgang PC3=1 ist sollen die Ausgänge von GPIOA den Wert 0 annehmen.
   8. Wenn PA2 und PA3 gleichzeitig 1 sind sollen die alle 16 Ausgänge von GPIOC auf 1 gesetzt werden.
   9. Wenn PA1 = 1 wird dann GPIOC->ODR um 1 hochzählen. Danach warten bis PA1 wieder 0 wird
   10. Warten auf PA10 =1, während gewartet wird soll der Mikrocontroller nichts tun
   11. Warte auf Taste PA10=1, danach
       * Warte auf Taste PA6=1, danach
         1. Zähle GPIOC->ODR um 5 hoch