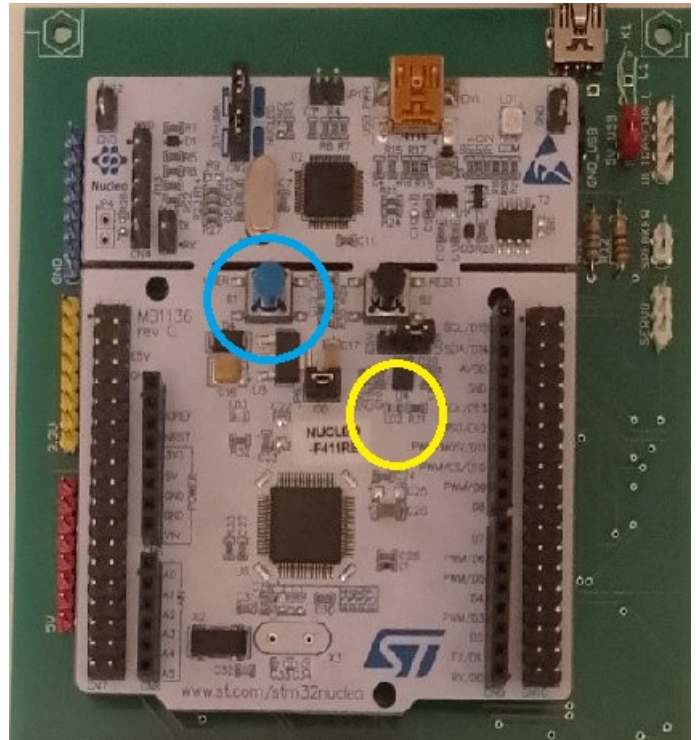


Aufgabe 1 Bits Schieben

Für viele Aufgaben ist es nötig zusätzliche Geräte an die Ports des Microcontrollers anzuschließen. Allerdings gibt es auch die Möglichkeit über eine LED und ein Button auf der Platine des Microcontrollers ein Signal einzulesen und auszugeben. Die LED ist an Port A an Pin 5 angeschlossen. Der Button an Port C Pin 13. Da Port C auch mit den Steckplätzen der 10 poligen Steckern angeschlossen ist, muss sichergestellt sein, dass am Steckplatz für das Highbyte von Port C nichts verbunden ist.



- Lege ein neues Projekt mit dem Namen 'Aufgabe_LED_durch_Button' an und kopiere die Datei Aufgabe_LED_durch_Button.s in den src-Ordner
- Betrachte die ausgeteilte Datei Aufgabe_LED_durch_Button.s diese wird Code für das Projekt bereitstellen, allerdings sind an mehreren Stellen kleine TODOs eingebaut. Meist wurde ein ?-Symbol platziert um ein Wert/Befehl zu bestimmen, damit das Programm funktioniert. Die TODOs habe folgende Struktur:

Listing 1: Ausschnitt einer Aufgabe

```
1 // GPIOA ----- LED an Port A auf dem Board als Output
2 ldr r1,=GPIOA // R1 = Basisadresse von Port A
3 // TODO: Ersetze ? durch die Hexadezimal des Bitmusters mit einem Ausgang an der Stelle des LED-Pins
4 ldr r2,=? // MODER_von_A = 00_00_00_00_00_00_00_00_01_00_00_00_00
5 strh r2,[R1,#MODER] // Port A Modus
```

Die Lösung des TODOs wäre in diesem Fall das Bitmuster, ausgedrückt als Hexadezimalzahl. Wenn nur der LED Pin an Port A als Ausgang verwendet werden soll, so muss das Muster 01 für Output an die Stelle des Port Pins 5 platziert werden. Die Stelle könnte man errechnen (Pin 5 ist an Bits $5*2$ und $5*2+1$) oder aus der Formelsammlung nachlesen (Abschnitt GPIO Hardwareaufbau und Register auf Seite 6). Dementsprechend wird ein Bitmuster mit nur Nullen außer Position 10 gesucht. Das entspricht 0x400 bzw. 0x00000400 wenn man alle 32 Bit zur Verdeutlichung ausschreiben will. Das kann dann an die Stelle des ?-Symbols eingesetzt werden um die folgende Lösung zu erhalten:

Listing 2: Lösung einer Aufgabe

```
1 // GPIOA ----- LED an Port A auf dem Board als Output
2 ldr r1,=GPIOA // R1 = Basisadresse von Port A
3 // TODO: Ersetze ? durch die Hexadezimal des Bitmusters mit nur einer 1 an der Stelle des LED-Pins
4 ldr r2,=0x00000400 // MODER_von_A = 00_00_00_00_00_00_00_00_01_00_00_00_00
5 strh r2,[R1,#MODER] // Port A Modus
```

- c) Löse nun die TODOs im Code. Wenn alle Stellen richtig gelöst wurden, so kann das Programm auf den Microcontroller geladen werden. Die LED reagiert nun auf das Drücken des blauen Buttons.
- d) Zusatz: Erstelle ein neues Projekt `Port_C_high_auf_Port_B_low` und entwerfe ein Programm, dass das Highbyte von Port C ausliest und auf dem Lowbyte von Port B ausgibt. Als Starthilfe kannst du den Code aus dem ersten Übungsblatt zum Microcontroller nutzen.