

## CT Übungsaufgaben

### GPIO Konfiguration und Programmierung

Auf dem STM32F429 Discovery Board soll der Wert des User Buttons (B1) eingelesen werden und der invertierte Wert an der grünen User Led (LD3) oder der roten User Led (LD4) oder in einem eigenen Muster ausgegeben werden. Die Dokumentation finden Sie unten (Seite 3) und verwenden Sie das in OLAT abgelegte Referenzmanual.

#### Aufgaben:

- a) Welche GPIO-Ports müssen für die Ein- bzw. Ausgabe angesprochen werden?  
Geben Sie den Namen und die Basisadressen an.

GPIO Port A -> Reference Manual nachschauen bei GPIOA -> 0x4002'0000  
GPIO Port G -> Reference Manuel nachschauen bei GPIOG -> 0x4002'1800

- b) Welche Register müssen für die Ausgabe an der LED konfiguriert werden?

Vorgehen: Geben Sie die Namen und die Adressen an.

1. Alle Types aufschreiben (MODERN, TYPER...)
2. Nachschauen unter welchem GPIO Port das LED angeschlossen ist
3. Nachschauen wie der Offset ist
4. Basis-Adresse von dem angeschlossenen GPIO Port + Offset berechnen = Adresse für Type

MODER 0x4002'1800  
TYPER 0x4002'1804  
SPEEDR 0x4002'1808  
PUPDR 0x4002'180C

Vorgehen: c) Konfiguration (Bit-Werte), welche Bits müssen vom Reset her verändert werden?

1. Überprüfen an welchem Pin die LED angeschlossen -> GPIO G PIN13
2. Prüfen welchen Mode wir für das jeweilige Register setzen wollen
3. Schema am Datenblatt ablesen wie gerechnet wird
4. Berechnung durchführen

MODER 0x1 << (2\*13)  
TYPER 0x0 << 13  
SPEEDR 0x1 << (2\*13)  
PUPDR 0x0 << (2\*13)

- d) Welche Register müssen für die Eingabe konfiguriert werden?

Geben Sie die Namen und die Adressen an.

-> Eingabe erfolgt über Button -> GPIO Port A relevant -> Startadresse 0x4002'000

MODER 0x4002'0000  
TYPER 0x4002'0004  
SPEEDR 0x4002'0008  
PUPDR 0x4002'000C

- e) Konfiguration (Bit-Werte), welche Bits müssen vom Reset her verändert werden?

MODER 0x0 << (2\*0)  
TYPER -> Wird nicht verwendet  
SPEEDR -> Wird nicht verwendet  
PUPDR 0x0 << (2\*0)

- f) Wie werden die beiden LED gelöscht? Geben Sie Port, Registername und -Adresse sowie die dazugehörigen Bitwerte an

- g) Schreiben Sie eine Eingabemethode, die den aktuellen Zustand des User Buttons B1 einliest und zurückgibt, damit wissen Sie, welche Werte B1 im Ruhezustand und im gedrückten Zustand liefert.

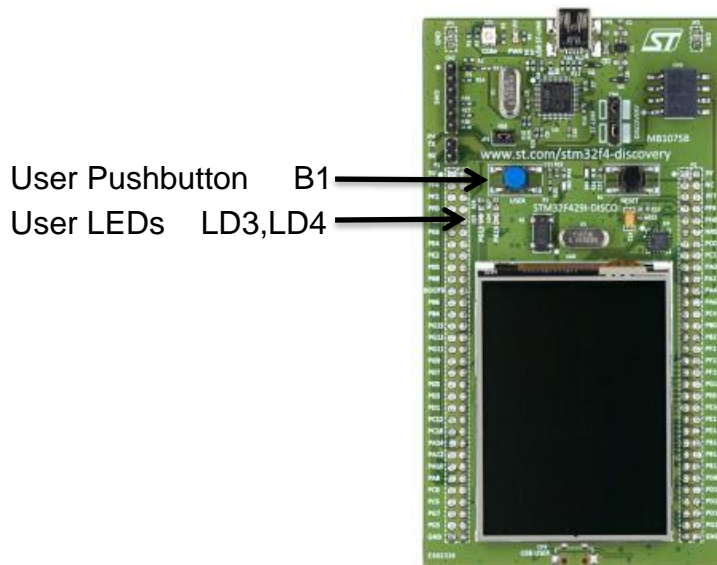
**Hinweis:**

Sie können die GPIO *mit dem gegebenen* Programmrahmen auch auf einem STM32F429 ohne CT-Board verwenden. Auf dem CT-Board muss dazu der Drehschalter P10 in Stellung 0 gesetzt werden. Zudem muss im Projekt-Teil „Options for target 1“ im Reiter C/C++ im Abschnitt „Preprocessor Symbols“ im Feld „Define:“ **NO\_FMC** eingetragen werden.

**Ideen für die Programmierung (freiwillig):**

- h) Anschliessend erweitern Sie die main-Schleife so, dass das Drücken des Buttons an der oder den Led angezeigt wird.

## Ausschnitt aus der Dokumentation des STM32F429 Boards



### 4.4 LEDs

- User LD3:  
Die grüne LED ist eine User LED, die über **I/O Port G Pin 13** des STM32F429ZIT6 Boards angeschlossen ist.
- User LD4:  
Die rote LED ist eine User LED, die über **I/O Port G Pin 14** des STM32F429ZIT6 Boards angeschlossen ist.

### 4.5 Taster

- B1 USER:  
Der blaue User-Button ist über **I/O Port A Pin 0** des STM32F429ZIT6 Boards angeschlossen.

Die komplette Dokumentation des STM32F429ZIT6 Boards ist auf OLAT zu finden.