

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 → Eingabe 0x40020000
GPIO Port G → Ausgabe 0x40021800

- b) Welche Register müssen für die Ausgabe an der LED konfiguriert werden?
Geben Sie die Namen und die Adressen an.

MODER 0x40021800
TYPER 0x40021804
SPEEDR 0x40021808
PUPDR 0x4002180C

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

MODER 0x1 << (2*13) // Mode Out, Pin 13
TYPER 0x0 << 13 // Push Pull, Pin 13
SPEEDR 0x1 << (2*13) // Speed med., Pin 13
PUPDR 0x0 << (2*13) // No PullUp, Pin 13

- d) Welche Register müssen für die Eingabe konfiguriert werden?
Geben Sie die Namen und die Adressen an.

MODER 0x40020000
TYPER 0x40020004 // für Eingabe nicht relevant
SPEEDR 0x40020008 // für Eingabe nicht relevant
PUPDR 0x4002000C

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

MODER 0x0 << (2*0) // Mode In, Pin 0
TYPER // Wird nicht verwendet, ignoriert
SPEEDR // Wird nicht verwendet, ignoriert
PUPDR 0x0 << (2*0) // No PullUp, Pin 0

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

Port G + BSSR_offset 0x1 << (16+13) // Clr 0..15 → bit 16..31
0x40021C00 + 0x18
Port G + BSSR_offset 0x1 << (16+14) // Clr 0..15 → bit 16..31
0x40021C00 + 0x18

- 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.

Port A + IDR_offset
0x40020000 + 0x10

Gelesenen Wert mit 0x1 maskieren, dann
auf 1 (gedrückt) testen.

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.

Port G + BSSR_offset
0x40021C00 + 0x18

0x1 << (13)

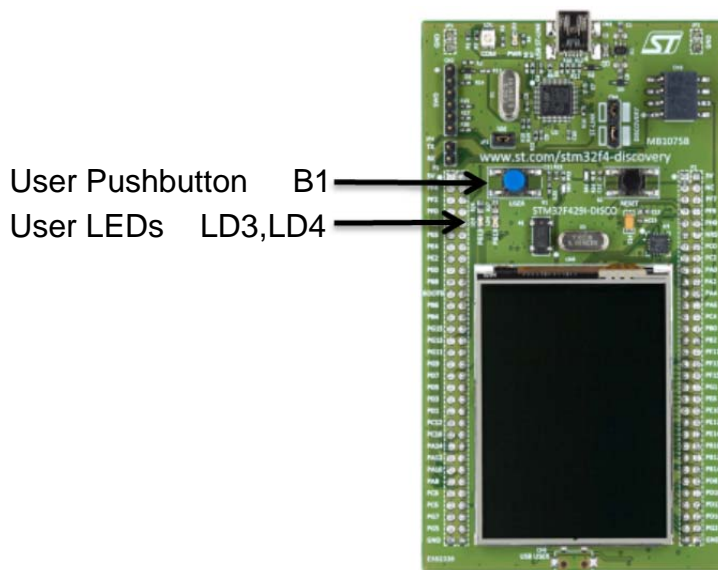
Port G + BSSR_offset
0x40021C00 + 0x18

0x1 << (13) und 0x1 << (16+14)

Port G + BSSR_offset
0x40021C00 + 0x18

0x1 << (14) und 0x1 << (16+13)

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.