

Frågor om microcontroller

Nu kan det vara dags att stanna upp och läsa lite i manualerna, samt lösa några uppgifter, innan vi går vidare. Svara på frågorna på svarsbladet längst bak i häftet!

- 1) Läs i databladet för STM32L476xx kapitel 2 (den controller vi använder i NUCLEOkortet är STM32L476RG¹). Studera speciellt Table 2 och Figure 1 så får du en översikt över alla periferienheter som finns förutom digitala in- och utgångar.
- 2) Läs översiktligt igenom UM1860: Getting started with STM32CubeL4... Vi kommer att använda HAL-biblioteket när vi programmerar NUCLEO-kortet. När man som du är ny med en mikrocontroller kan det bli mycket information att ta in, men läs igenom översiktligt så att du får en inblick i arkitekturen för mjukvaran.
- 3) Läs sedan UM1884 Description of STM32L4/L4+ HAL and low-layer drivers, kapitel 2. Denna manual är vår uppslagsbok när det gäller API² för HAL³. I kapitel 2 finns en översikt över HAL-biblioteket. Läs kapitlet översiktligt och besvara följande frågor:
 - a) Det finns tre API programmeringsmodeller: polling, interrupt och DMA. Förklara vad som menas!

Svar: When operating in Polling mode, the HAL functions provide the process status upon the successful completion of data processing in a blocking mode. An operation is considered finished when the HAL_OK status is returned by the function. If not, an error status is delivered. Further insights can be obtained using the HAL_PPP_GetState() function. Data processing is internally managed within a loop, and a timeout, specified in milliseconds, is employed to safeguard against process hanging.

Operating in Interrupt mode, the HAL function provides the process status after initiating data processing and enabling the relevant interruption. The operation's conclusion is signified by a callback function declared as a weak function. Users have the flexibility to customize this function to receive real-time notifications about the process's completion. Additionally, the process status can be retrieved through the HAL_PPP_GetState() function.

When operating in DMA mode, the HAL function furnishes the process status upon commencing data processing through the DMA and enabling the relevant DMA interruption. The conclusion of the operation is denoted by a callback function, declared as a weak function, which users can tailor to receive real-time updates regarding the process's completion. Furthermore, users can obtain the process status via the HAL_PPP_GetState() function.

- b) Vad menas med att koden är reentrant?

¹ L4 är produktserie, 76 är produktlinje i serien, R står för 64 pinnars kapsel, G betyder att kretsen har 1 MByte flashminne

² API Application Programming Interface

³ HAL Hardware Application Layer

Svar: Reentrant code operates without any reliance on static or global non-constant data. Reentrant functions can interact with global data, as illustrated by a reentrant interrupt service routine that accesses volatile global hardware status, such as a serial port read buffer. Nevertheless, it is important to exercise caution when working with static variables and global data. The recommended approach is to use atomic read-modify-write instructions exclusively on these variables. This precaution is taken to ensure that an interrupt or signal does not interrupt the execution of such an instruction, as reentrant code refrains from altering its own code.

- c) Vad menas med att implementationer av HAL APIs kan anropa user-callback functions, dvs. vad innebär user-callback?

Svar: The implementation of HAL APIs that utilize user-callback functions allows for user-specific functionality in the software. User-callback functions are functions that the user (developer) can define and specify to be called by the HAL (Hardware Abstraction Layer) when certain events occur. These events can include: peripheral system level Initialization/De-Initialization (clock, GPIOs, interrupt, DMA), Peripherals interrupt events and Error events.

- 4) Använd datablad för **microcontroller** och manual för NUCLEO-kortet för att ta reda på hur microcontrollern klockas. Vilka alternativ finns det för att klocka microcontrollern?

Svar: Below are three clock sources:

LSE - This is the 32.768 kHz crystal utilized by the STM32 embedded RTC. There are three ways to configure the pins corresponding to the low-speed clock.

MCO - It represents the 8 MHz clock sourced from the ST-LINK MCU and is intended for the STM32 microcontroller.

HSE - An 8 MHz oscillator designated for the STM32 microcontroller. It's important to note that this clock source is not available on the STM32 Nucleo-64 board. There are four ways to configure the pins corresponding to the external high-speed clock.

- 5) Nu skall du studera GPIO i microcontrollern, dvs. när en pinne används som ingång eller utgång. Du ska söka information i datablad för STM32L476xx när det gäller elektriska specifikationer och i referensmanualen RM0351 när det gäller logisk uppbyggnad av hårdvaran i periferienheten.

- a) Hur bestäms det hur en pinne konfigureras som ingång eller utgång? Vilka register skall påverkas och på vilken eller vilka adresser ligger de om GPIO-port D skall påverkas.

Svar: Software configuration allows each GPIO pin to function as either an output (in a push-pull or open-drain mode), an input (with or without pull-up or pull-

down capabilities), or as a peripheral alternate function. Numerous GPIO pins have the additional capability to be used for digital or analog alternate functions. High-speed I/O toggling is made possible through their connection to the AHB2 bus. For added security against inadvertent write operations to the I/Os registers, there exists a mechanism for locking the configuration of the I/Os alternate function, which can be activated by following a specific predefined sequence.



- b) Markera i figurerna 23, 24 och 26 nedan (från referensmanualen) vilka transistorer som är påverkade on eller off för följande fall
- i) Ingång med pullup-motstånd (Yellow color)
 - ii) Flytande ingång (floating) (Red color)
 - iii) Utgång push-pull (green)
 - iv) Utgång open drain (blue)
- c) Om en pinne konfigureras som utgång, hur mycket ström kan den leverera?
Max 20 mA.
- d) Om en pinne konfigureras som ingång, hur hög spänning får man maximalt lägga på pinnen utifrån? Det kan vara olika för olika pinnar, specificera!
TT_xx I/O: 3.9V
BOOT0: 9V
all other: 7.2V or 9.1V.
- e) Inom vilket område ska matningsspänningen till microcontrollern ligga vid normal drift?
1.1V to 3.6V operating supply voltage (VDD).

Figure 23. Basic structure of an I/O port bit

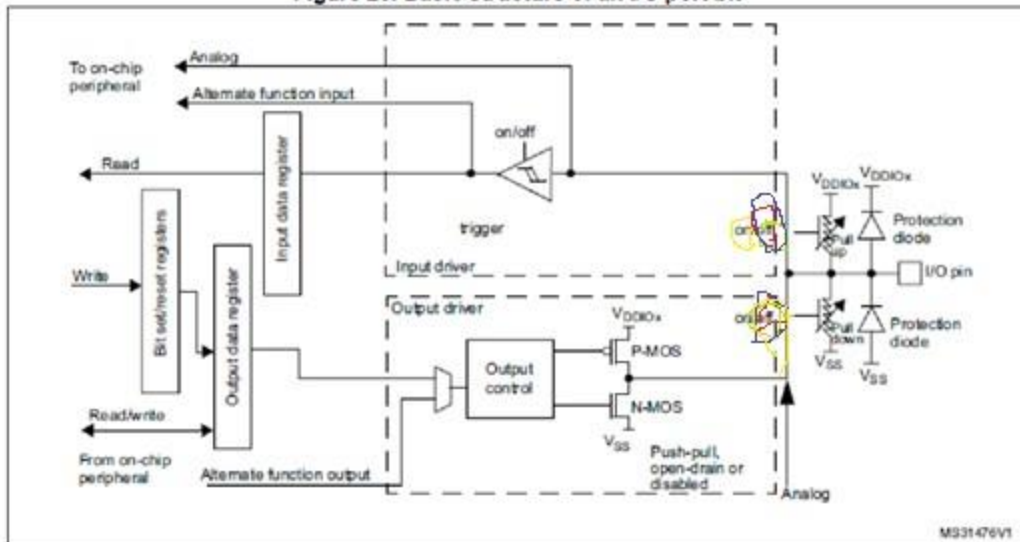
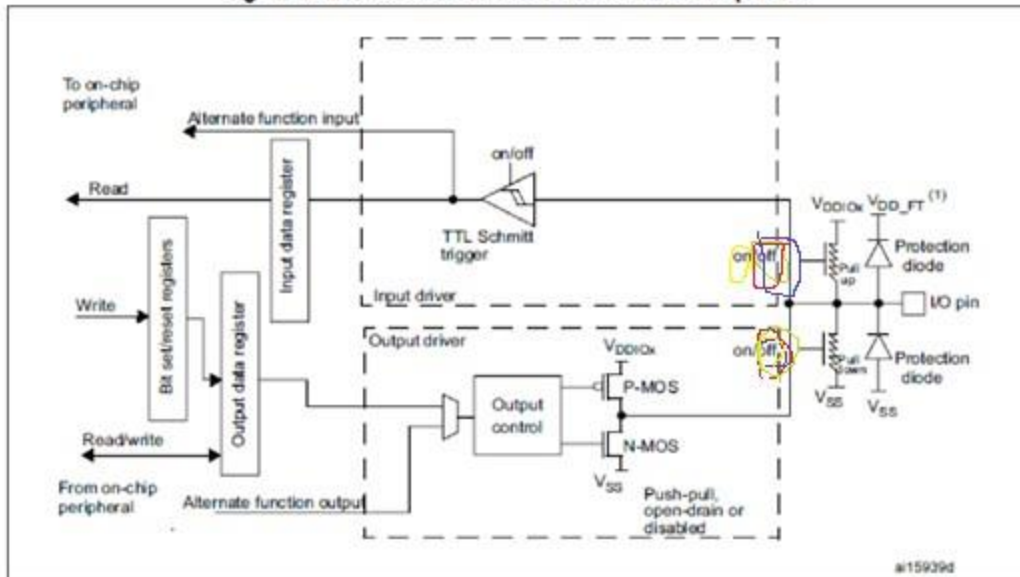


Figure 24. Basic structure of a 5-Volt tolerant I/O port bit



1. V_{DD_5T} is a potential specific to five-volt tolerant I/Os and different from V_{DD} .

Figure 26. Output configuration

