Using the STM8S-DISCOVERY GPIOs and interrupt controller to drive LEDs

Application overview

This application is based on the STM8S-DISCOVERY. It demonstrates how to use the STM8S GPIOs and interrupt controller to drive a set of LEDs.

Once the STM8S105C6T6 is powered up through an USB cable connected to the host PC, LEDS LD2 and LD5 start blinking meaning that the programming operation has completed successfully.

Each time the pushbutton is pressed, the interrupt controller asserts an interrupt that is used to control the I/Os, and change the LED behavior.

Reference documents

■ STM8S-DISCOVERY evaluation board user manual (UM0817).

■ Developing and debugging your STM8S-DISCOVERY application code user manual (UM0834).

Contents

List of tables

List of figures

1 Application description

1.1 Hardware requirements

No on-board resources are required.

Table 1 gives the list of external components required by the application.

Table 1. List of external components

| External components	Value	Comments
LD2, LD3, LD4, LD5	-	Standards LEDs
R2, R3, R4, R5	510 Ω	Protective resistors
R1	4.7 kΩ	Pull-up resistor
R	100 Ω	Debounce filter
C	100 nF	
Pushbutton	-	Standard pushbutton

1.2 Application schematics

Figure 1 shows how to interface the LEDs and the pushbutton with the STM8S-DISCOVERY.

For details on STM8S-DISCOVERY implementation, refer to the board schematics provided in the STM8S-DISCOVERY user manual (UM0817).

Protective resistors, R2, R3, R4, and R5, are mandatory to limit the current to a value that does not harm the LEDs.

The pushbutton requires a debounce filter (RC) and a pull-up resistor (R1) to avoid triggering several interrupts due to the mechanical bouncing of the button.

Figure 1.   Application schematics
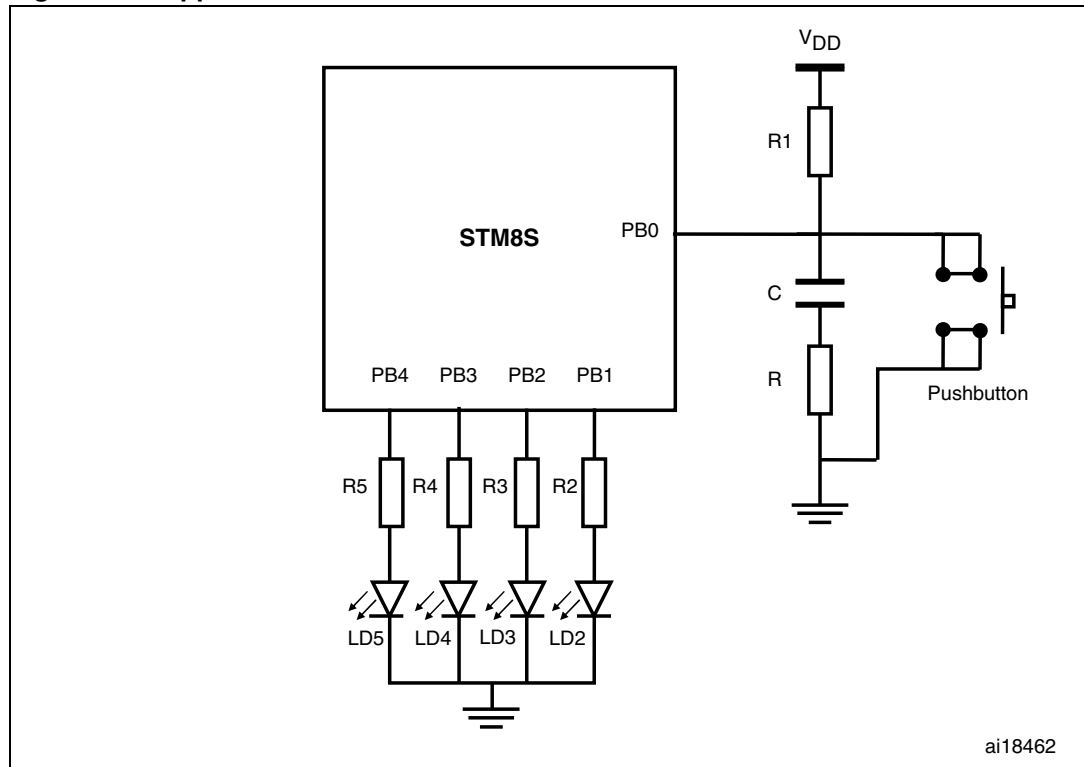
1.3   Application principle

At startup, LD2 and LD5 start blinking meaning that the STM8S105C6T6 Flash memory has been successfully programmed.

Pressing the pushbutton generates an interrupt which is handled by the application software to drive the LEDs.

Only one of the two pairs of LEDs, LD2/LD5 and LD3/LD4 blink at a time. A button event triggers the blinking of the other pair while switching off the first one.

The LEDs blinking conditions are described in *Table 2*.

Table 2.   LEDs configuration

Application	LED state
At startup	LD2 and LD5 blink
On button event	The blinking LED pairs are swapped

2 Software description

The application software uses STM8S standard firmware library to control the general purpose features described in *Section 2.1*.

2.1 STM8S peripheral configuration

2.1.1 GPIOs

The application drives the MCU I/Os to interface the microcontroller with external hardware components. The GPIO_Init() function configures PB0 as floating input with interrupt to detect pushbutton events, and PB1/PB2/PB3/PB4 as output push-pull to control the LEDs.

2.1.2 EXTI

The external interrupt controller is configured through the EXTI_SetExtIntSensitivity() function to handle the external interrupts on PB0.

The external interrupt sensitivity is configured to trigger an interrupt each time a falling edge, and only a falling edge, is detected on PB0.

2.2 Standard STM8S standard firmware library configuration

The *stm8s_conf.h* file of the STM8S standard firmware library is used to configure the library by enabling the peripheral functions used by the application.

The following define statements must be present:

```
#define _GPIO 1 /* enables the GPIOs */
#define _EXTI 1 /* enables the EXTI */
```

2.3 Application software flowchart

2.3.1 Main loop flowchart

The code main loop implements the algorithm that controls the LEDs according to pushbutton events.

The blinking LED pair is selected by setting the *ButtonState* flag.

Each time the pushbutton is pressed, an interrupt is triggered and *ButtonState* is complemented (see *Section 2.3.2: Interrupt function flowchart*). The main loop code tests *ButtonState* and selects the blinking LED pair according to its value (see *Table 2*).

The Delay() function generates a delay between the LED ON and OFF states so that we can see them blink.

Figure 2 shows the flowchart of the application software main loop.
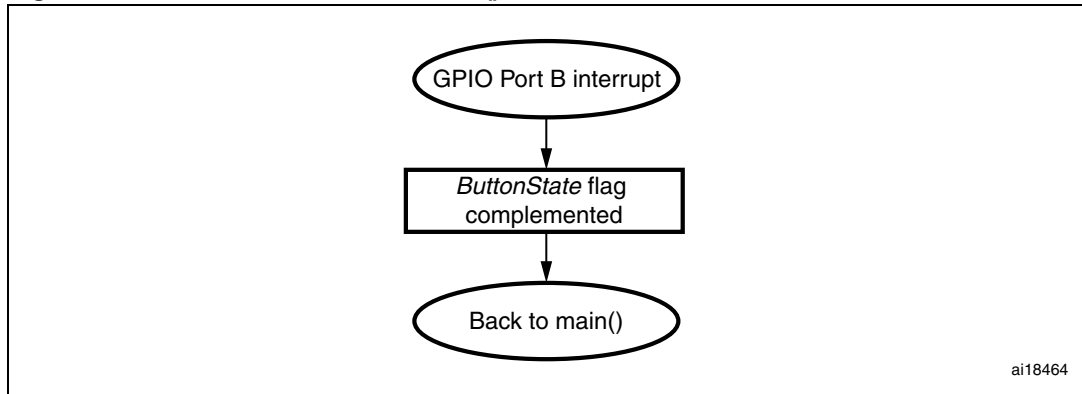
Figure 2. Main loop flowchart
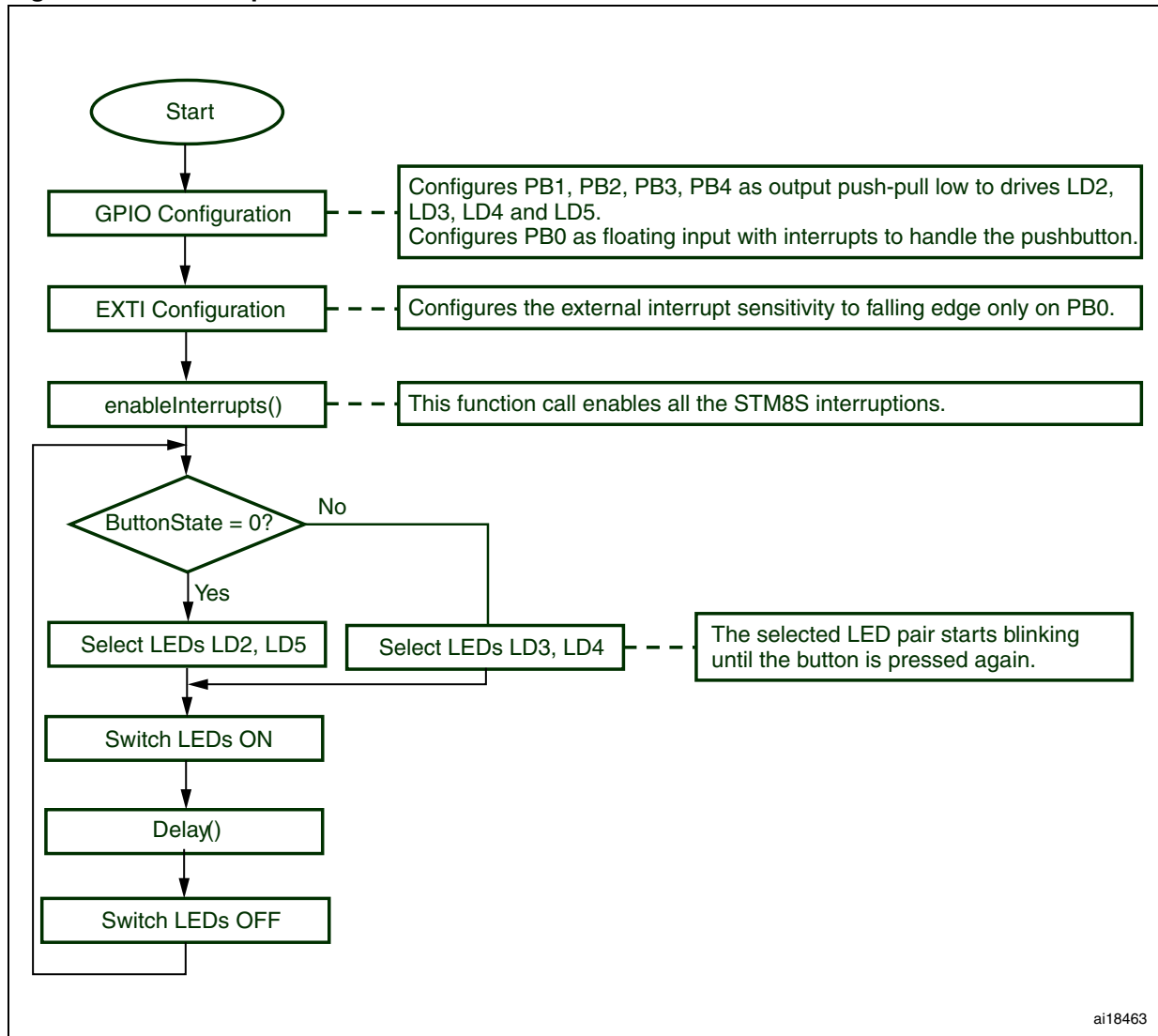
2.3.2 Interrupt function flowchart

Each time an interrupt is asserted, the EXTI_PORTB_IRQhandler() function complements the *ButtonState* flag and the main loop behaves accordingly (see *Table 2*).

Figure 3 shows the flowchart of the EXTI_PORTB_IRQhandler() interrupt function.

Figure 3.   EXTI_PORTB_IRQhandler() function flowchart

3 Revision history

Table 3.   Document revision history

Date	Revision	Changes
12-Nov-2010	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.