

## FAQ0124

## Frequently Asked Questions

## How to achieve GPIO fast toggle in Keil?

**Questions:**

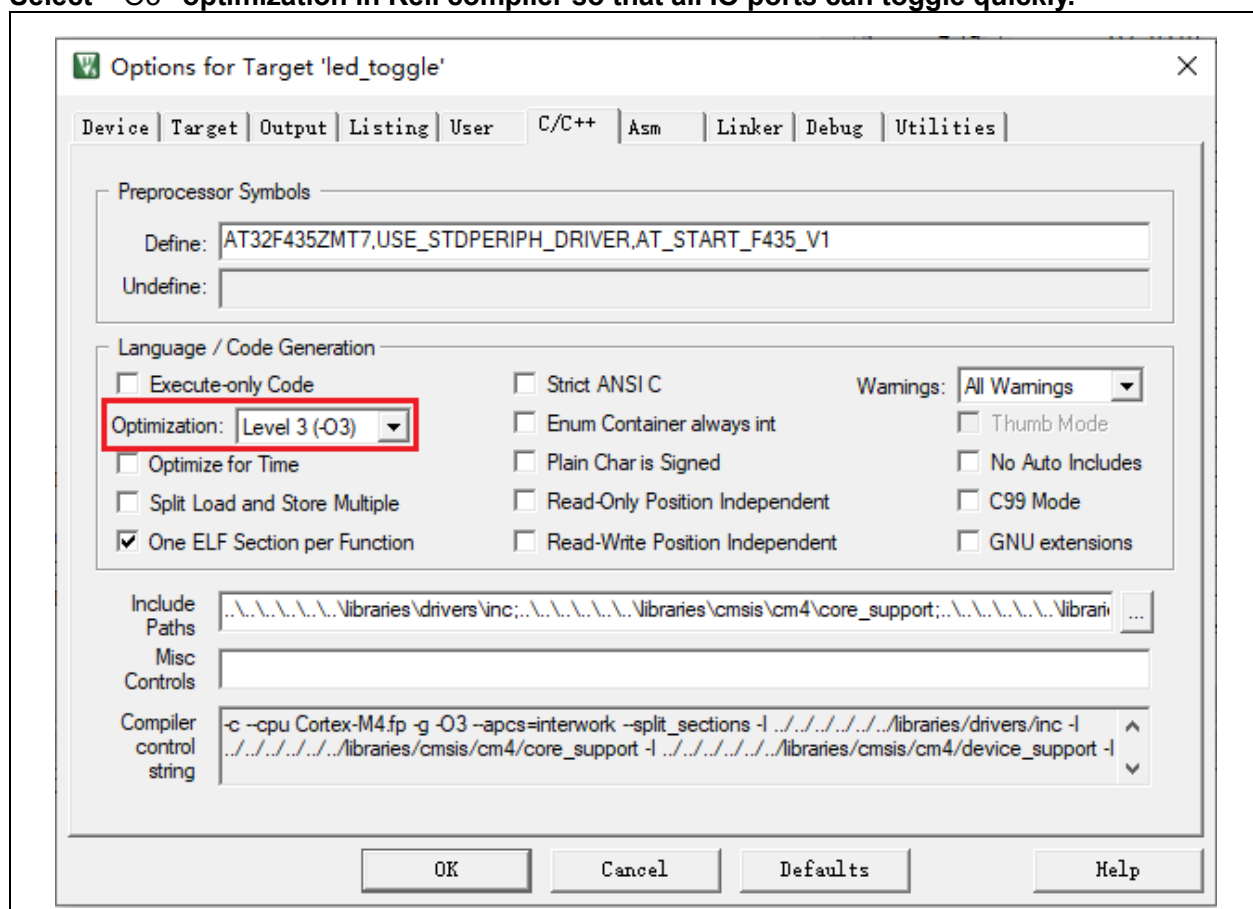
In Keil environment, when an IO port is configured in push-pull output mode, and it must be toggled quickly. But after running code based on the compiler's optimization level "-O0", it is found that only PA port has a fast toggle rate, other ports not.

**Answer:****Reason:**

In Keil environment, if the optimization level "-O0" is selected, it only can optimize the code of PA port without taking into account of other ports. This issue has nothing to do with microcontrollers (they work normally). It is related to Keil C language.

**Solution 1:**

Select "-O3" optimization in Keil compiler so that all IO ports can toggle quickly.



**Solution 2:**

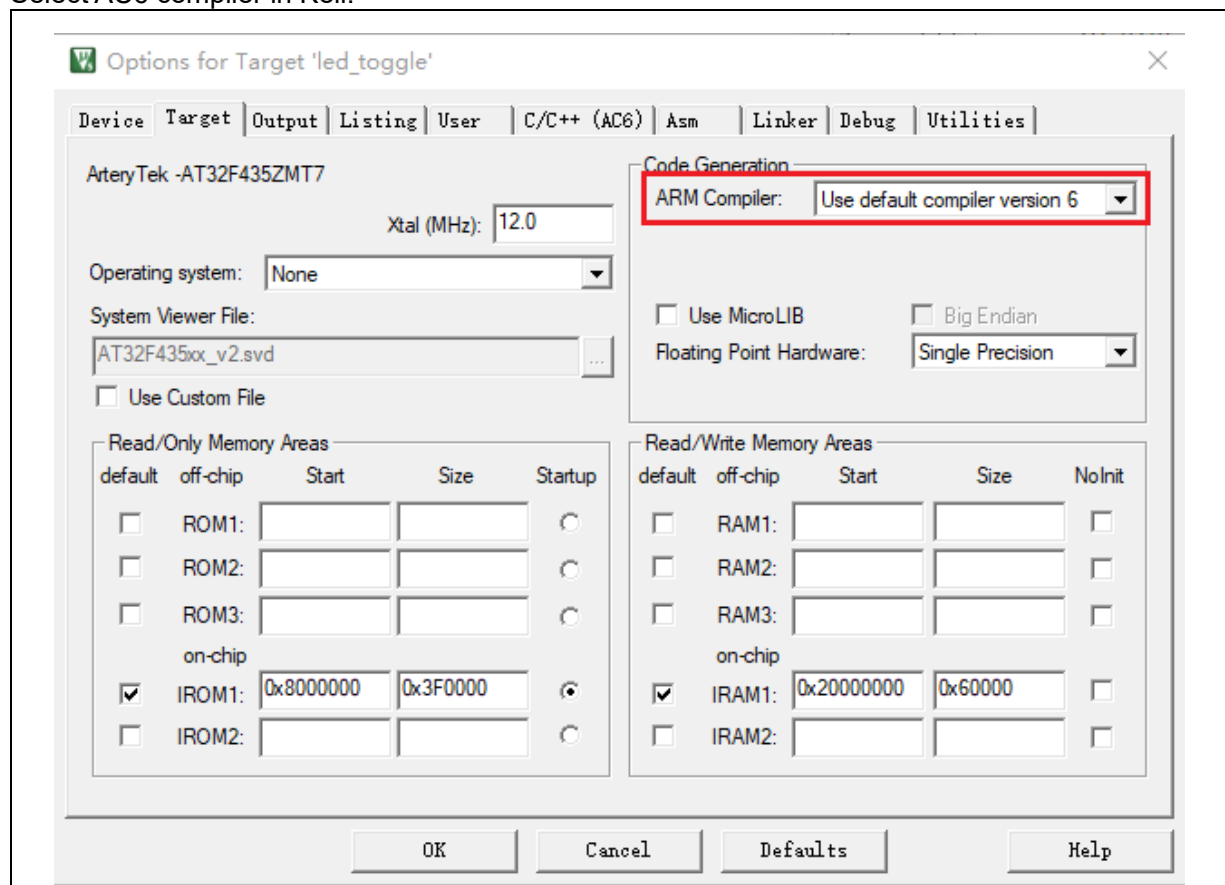
After using “-O3” optimization level, it may be not so convenient for debugging due to the fact that there are not so many debug signals. If so, users can use “#pragma” compiling instruction to specify which codes need “-O3” optimization. This method would not affect other codes.

Example: Pack IO toggle into a function, and then assign “-O3” optimization level to this function.

```
#pragma push
#pragma O3
void Toggle_IO(gpio_type * GPIO_x, uint16_t uGPIO_pins)
{
    GPIO_x->clr = uGPIO_pins;
    GPIO_x->scr = uGPIO_pins;
    GPIO_x->clr = uGPIO_pins;
    GPIO_x->scr = uGPIO_pins;
    GPIO_x->clr = uGPIO_pins;
    GPIO_x->scr = uGPIO_pins;
    GPIO_x->clr = uGPIO_pins;
    GPIO_x->scr = uGPIO_pins;
}
#pragma pop
```

**Solution 3**

Select AC6 compiler in Keil.

**Solution 4:**

Switch to IAR for compiling.

**Type:** MCU application

**Applicable products:** AT32F421, AT32F435, AT32F437, AT32F425, AT32L021

**Main function:** None

**Other function:** None

## Document revision history

Date	Revision	Changes
2022.3.4	2.0.0	Initial release

## IMPORTANT NOTICE – PLEASE READ CAREFULLY

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

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license granted by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement on any patent, copyright or other intellectual property right.

Purchasers hereby agree that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any aircraft application; (C) any aerospace application or environment; (D) any weapon application, and/or (E) or other uses where the failure of the device or product could result in personal injury, death, property damage. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and Purchasers are solely responsible for meeting all legal and regulatory requirements in such use.

Resale of ARTERY products with provisions different from the statements and/or technical characteristics stated in this document shall immediately void any warranty grant by ARTERY for ARTERY's products or services described herein and shall not create or expand any liability of ARTERY in any manner whatsoever.

© 2022 Artery Technology -All rights reserved