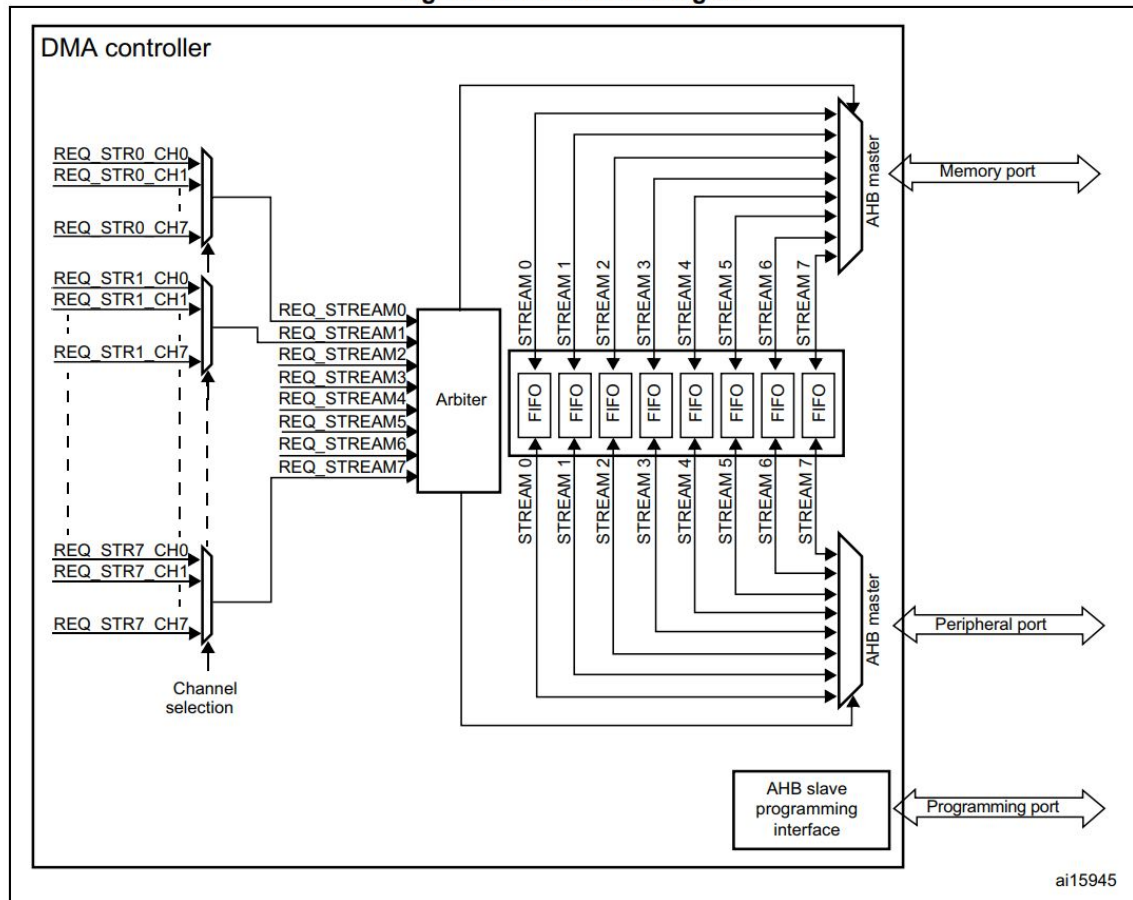


Lab 7 - DMA

In this lab, you will be using DMA (Direct-memory-access) to transfer data. DMA allows your application to continue working while data can be copied from one location to another.

STM32 has two DMA controllers. Each DMA has the following block diagram.

Figure 32. DMA block diagram



Each controller is connected to different bus. Please see additional information in the reference manual. Each controller can work in this following operations:

- Memory to Memory
- Peripheral to Memory
- Memory to Peripheral

When the controller is working in peripheral to memory or memory to peripheral, the controller will transfer data upon “request” input. The request table can be found in Table 42. of STM32F4 Reference Manual.

Your Tasks:

1. The first task is to use memory to memory feature of the DMA. Your program will do the following

- Create two arrays of the 255 bytes, `arr0` and `arr1`.
- Fill the first array with
`arr0[i] = ((i + 55) * 37) % 57;`
- Fill the second array with zero
- Upon use hitting the USER button, it will copy data from `arr0` to `arr1`

Because of the bus structure of STM32, you can only do memory to memory transfer using DMA2 controller.

2. In the second task, your program must do the following:
 - Accept data through UART. However, it will return data only after 256 characters are received.
 - During the time STM32 is sending the data back to the computer, it must still accept more input, and would send the data back after 256 characters.
 - An LED must always blink 200ms period.
 3. The last task is similar to the second one, however, it will send data back at the rate of 10 characters per second.
- Remark
 1. You must implement all of these memory/peripheral operations using DMA in order to complete the task.
 2. For Task 3, since the rate of transfer is much slower than the rate of UART, you can use TIMER to trigger interrupt request. Note that you will be using a different DMA Stream/Channel than the typical UART one.

