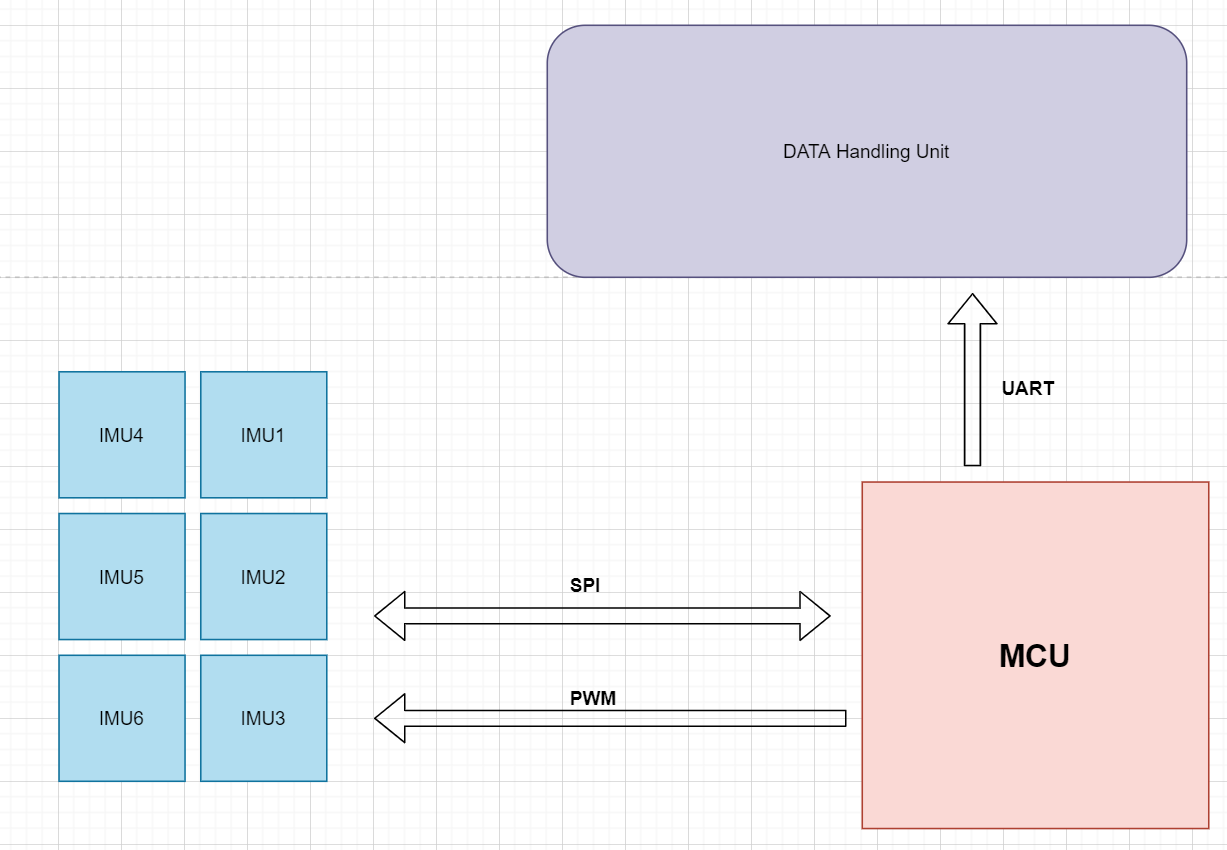
**STM32 Embedded C Code for IMU Array**

**Introduction:**

This code’s main functionalities are to retrieve FIFO data from an array of six IMUs by SPI, doing numerical data processing and sending the processed data to a data handling unit through UART. The data of all IMUs are synchronized by a PWM signal from the MCU, change of the PWM frequency will result in the change the IMU FIFO output data rate. The FIFO rate of each IMU is required to run at least 800Hz. In order to boost the chip’s productivity, using DMA to create SPI burst read to read the FIFO is able to support 1200Hz reading of six IMUs with the current code.

**System Block Diagram:**

****

*Figure 1: Overall system block diagram*

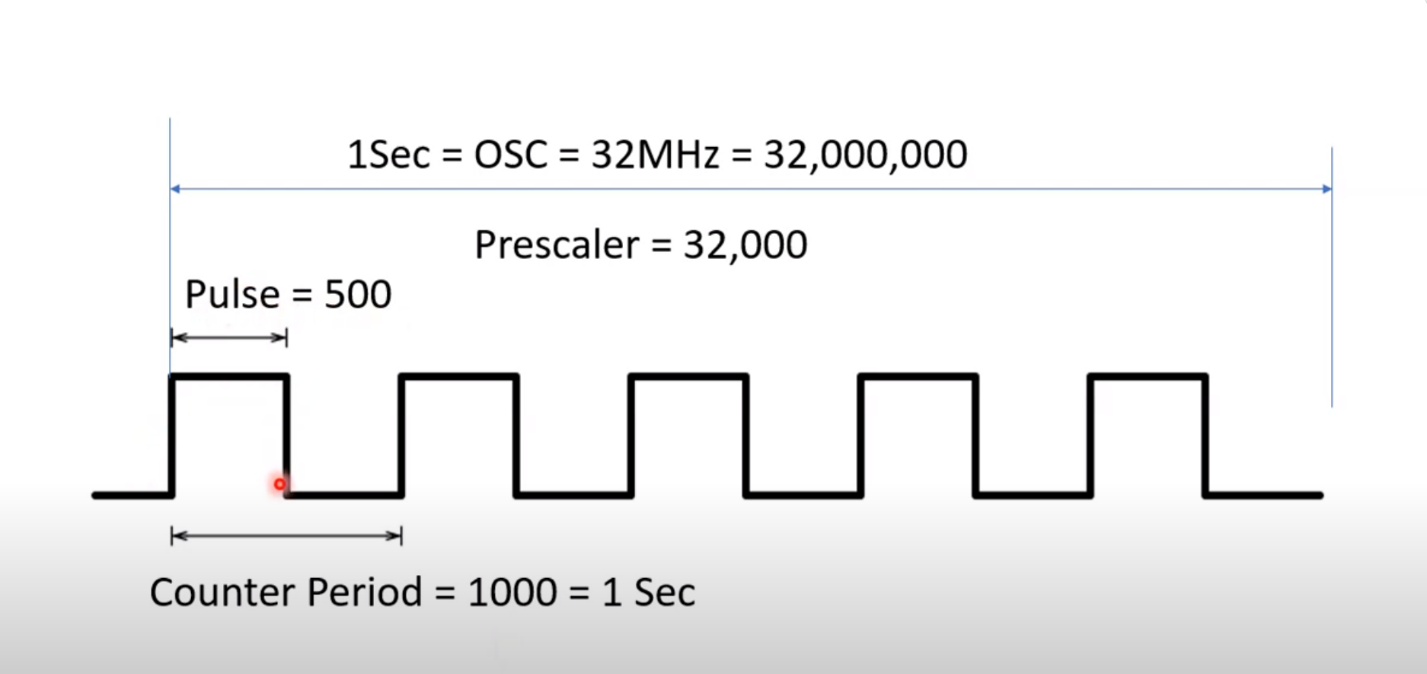
**Software used:**

**IDE:** STM32CUBEIDE

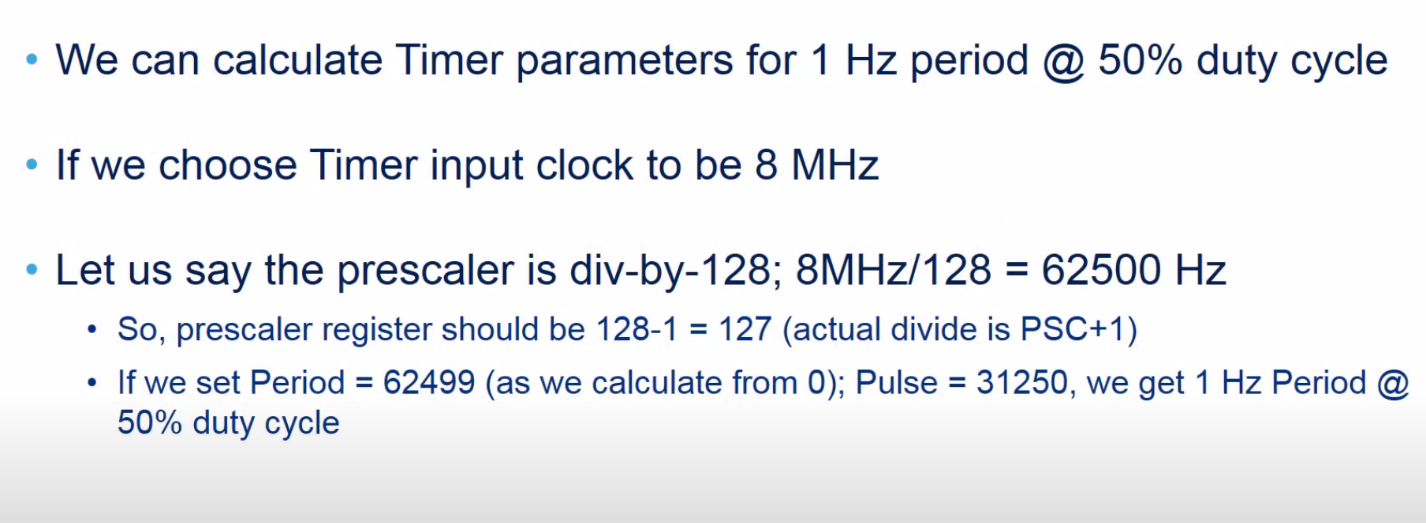
**Init/Config generator:** STM32CUBEMX

**Diligent Discovery**: Waveform

**PWM Frequency calculation:**

**

*Figure 2: PWM calculation 1*

**

*Figure 3: PWM calculation 2*

**GPIO:**

**SPI1:**

* SPI1\_CLK - > PA5 (SCL in LSM6DSL)
* SPI1\_MISO->PA6 (SDO/SA0 in LSM6DSL)
* SPI\_MOSI-> PA7 (SDA in LSM6DSL)
* CS: B15, B14, B13, B3, B4, B5 (CS: B15 for one IMU)

**UART1**

* UART1\_TX ->PA9
* UART1\_RX -> PA10

**UART2**

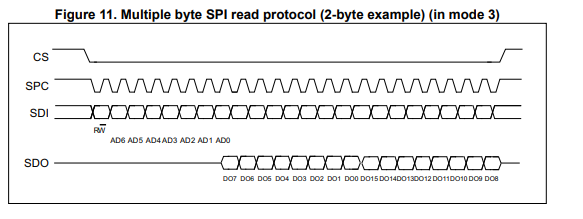
* UART2\_TX ->PA2
* UART2\_RX->PA3

**PWM**

* Timer\_3 ->PC6
* Timer\_4->PB6 (INT2 in LSM6DSL)

**SPI burst read:**

A SPI burst read means it can read successive memory address without issuing the new command. In this mode, the address is automatically updated in the slave. According to the lsm6dsl application note 4.7.1, FIFO address will go from the first register to the last register and then go back to the first one. Using burst mode can minimized the SPI read time by sending less command to the slaves. The more bytes are read from the slaves at one transmission, the more process time it will be saved.



*Figure 4: Multiple byte SPI read from LSM6DSL datasheet*

Figure 4 shows an example of a burst read; MISO can read two bytes even when MOSI is issuing one a read command. In order to create SPI burst read for LSM6DSL, we need to make the SPI waveform as similar to this diagram as possible. Notices that the clock cycle are 24 in this diagram and it is hard to find a build in SPI function that has 24 clock cycle per transmission. However, SPI DMA function provides the ability to transmit data without leaving gap between each byte, this feature makes 24 clock cycle become feasible if we sets 8 clock cycles per transmission and transmit 3 bytes each time. Now that we can have 8\*3 clock cycle, to create figure 3, we only need to send MOSI with a read command to first FIFO address and leave the rest of 2 bytes 0, then MISO will do its work.

This description is also true for more bytes reading if we increase the byte that is sent.

The return array structure is slightly strange, and my best guess is because we have 8 bytes per transmission.

By doing some tests, we know that the return array pattern is

For reading one byte, the return array pattern is 0 0 D

For reading two byte, the return array pattern is 0 0 D1 0 D2

Based on Figure 4, we see that the data is read in MSB, the first data that is being read is the first bytes. In the case of FIFO, the read byte pattern is FIFO\_DATA\_OUT\_L first, FIFO\_DATA\_OUT\_H next and then goes back to FIFO\_DATA\_OUT\_L.