1. **What does the AUTOEND bit in the CR2 register do? Why don't you want to use it when you'll be needing a restart condition?**

The AUTOEND bit in the I2C CR2 register is designed to automatically issue a STOP condition once the preset number of bytes, defined in the NBYTES field, has been transmitted or received. This feature streamlines simple I2C transactions by removing the need for software to manually terminate the communication. However, for operations that require a seamless transition between write and read sequences without releasing the I2C bus—such as when a repeated start condition is needed—the AUTOEND bit becomes counterintuitive. Activating it would inadvertently send a STOP condition, disrupting the intended flow of communication. Therefore, in scenarios where continuous control over the I2C bus is essential, and a repeated start condition must be initiated, it is crucial to clear the AUTOEND bit. This allows for the manual management of transaction endpoints, ensuring the bus remains under control for the entire sequence and facilitating an uninterrupted transition between operations.

2. **This lab used standard-mode 100 kHz I2C speed. What values would you write in the TIMINGR if we were using 400 kHz fast-mode?**

The values for a 400kHz would be:

```
// Setting I2C2 to 4kHz
I2C2->TIMINGR |= 0x9;
I2C2->TIMINGR |= (0x3 << 8);
I2C2->TIMINGR |= (0x1 << 16);
I2C2->TIMINGR |= (0x3 << 20);
I2C2->TIMINGR |= (0 << 28);
```

3. **This lab used blocking code. To implement it completely as non-blocking you would replace all of the wait loops with interrupts. Most flags in the I2C peripheral can trigger an interrupt if the proper enable bit is set. Find the interrupt enable bits that match the following flags: The Inter-Integrated Circuit (I2C) Interface 17 • TC • NACKF • TXIS (transmit interrupt) • ARLO**

The interrupt enabled bits for the ones we need to match are the following: TC (Transfer Complete Interrupt Enable): bit position 6. NACKF (Not Acknowledge Interrupt Enable): bit position 4. TXIS (Transmit Interrupt Enable): bit position 1. ARLO (Arbitration Lost Interrupt Enable): bit position 9. So the 32-bit encoding is 0000_0000_0000_0000_0000_0001_0010_1001.

4. **The gyro can operate in three full-scale/measurement ranges, measured in degrees-per-second (dps). What are these three ranges?**

The three full-scale measurements are 250 degrees per second (dps) 500 degrees per second (dps) 2000 degrees per second (dps).

5. **What is the I2C address of the gyro when the SDO pin is low? The lab has the pin set high, read the I2C section of the gyro datasheet.**

   When the SDO pin is low, the I2C address of the gyro is 0x6A. It is also written as 1101010 in binary.