LFX Coding Challenge: DMA-based UART Communication

Challenge Overview:

Design and implement a program for a Tiva-C microcontroller (TM4C123GH6PM) to demonstrate **Direct Memory Access (DMA)** for efficient data transfer between UART peripheral and transmitter and receiver buffers defined in data memory (RAM). The goal is to bypass the CPU for data handling while ensuring robust UART communication.

- Participants are restricted from using **Tivaware libraries** and must directly manipulate hardware registers.
- You can use the TM4C123.h header file.

Challenge Requirements:

1. **Direct DMA Implementation**:
   Configure the DMA controller to:
   - Transmit a predefined data buffered in RAM via UART.
   - Simultaneously receive data from the UART and store it in a separate buffer.
2. **Custom Register-Level Programming**:

   Write all configurations (UART, DMA, clock, GPIO) directly using Tiva hardware registers. Tivaware library usage is prohibited.
3. **Visualization in Keil µVision Simulator**:

   Test the code on the Keil µVision simulator. (Free version is available). Step wise run the code in the debugger and see the register values being updated. Check the DMASTATUS register for the state of the transfer.
4. **Challenge Objectives**:
   - Properly configure UART and DMA without using Tivaware functions.
   - Ensure efficient data handling with minimal CPU involvement.
   - Demonstrate understanding of Tiva peripherals and DMA operation.

Input and Output:

- **Input**: A predefined string to be transmitted through UART.
- **Output**: The same string received back through UART and stored in a buffer. (For simulation, you can use loopback mode if needed.)

Deliverables:

1. **Source Code**: A complete program written in C.
2. **Video Submission**:
   - Explain the code's functionality and implementation.
   - Demonstrate the program's working using the Keil µVision simulator.
3. **Optional Documentation**: A brief text or PDF document summarizing the key points of the implementation.