Solution: UDP Sensor Client-Server (Windows Version)

Server Code

```
#include <iostream>
#include <winsock2.h>
#include <ws2tcpip.h>

#pragma comment(lib, "ws2_32.lib")

int main() {
    WSADATA wsaData;
    WSAStartup(MAKEWORD(2, 2), &wsaData);

    SOCKET sockfd;
    struct sockaddr_in server_addr, client_addr;
    int addr_len = sizeof(client_addr);
    char buffer[1024];

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (sockfd == INVALID_SOCKET) {
        std::cerr << "Socket creation failed" << std::endl;
        WSACleanup();
        return 1;
    }

    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = INADDR_ANY;
    server_addr.sin_port = htons(5000);

    if (bind(sockfd, (struct sockaddr*)&server_addr,
sizeof(server_addr)) == SOCKET_ERROR) {
        std::cerr << "Bind failed" << std::endl;
        closesocket(sockfd);
        WSACleanup();
        return 1;
    }

    std::cout << "Server listening on port 5000..." << std::endl;

    while (true) {
        memset(buffer, 0, sizeof(buffer));
        int n = recvfrom(sockfd, buffer, sizeof(buffer), 0,
                         (struct sockaddr*)&client_addr,
&addr_len);

        if (n > 0) {
            char client_ip[INET_ADDRSTRLEN];
            inet_ntop(AF_INET, &client_addr.sin_addr, client_ip,
INET_ADDRSTRLEN);
            std::cout << "Received from " << client_ip << ": " <<
buffer << std::endl;

            // Optional acknowledgment
            // sendto(sockfd, "ACK", 3, 0, (struct
sockaddr*)&client_addr, addr_len);
        }
    }

    closesocket(sockfd);
    WSACleanup();
    return 0;
}
```

Client Code

```
#include <iostream>
#include <winsock2.h>
#include <ws2tcpip.h>
#include <thread>
#include <chrono>

#pragma comment(lib, "ws2_32.lib")

int main() {
    WSADATA wsaData;
    WSAStartup(MAKEWORD(2, 2), &wsaData);

    SOCKET sockfd;
    struct sockaddr_in server_addr;

    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (sockfd == INVALID_SOCKET) {
        std::cerr << "Socket creation failed" << std::endl;
        WSACleanup();
        return 1;
    }

    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(5000);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    while (true) {
        const char* message = "Temperature: 23°C";

        sendto(sockfd, message, strlen(message), 0,
               (struct sockaddr*)&server_addr,
sizeof(server_addr));

        std::cout << "Sending: " << message << std::endl;

        std::this_thread::sleep_for(std::chrono::seconds(1));
    }

    closesocket(sockfd);
    WSACleanup();
    return 0;
}
```

Explanation

Server

- Initializes Winsock with `WSAStartup()`.
- Creates a UDP socket.
- Binds to port `5000`.
- Waits for incoming datagrams with `recvfrom()`.
- Prints received message and client's IP.

Client

- Initializes Winsock.
- Creates a UDP socket.
- Sends a message every second to the server using `sendto()`.
- Uses `std::this_thread::sleep_for()` for delay.

Summary

This solution demonstrates how to implement a UDP client-server program on Windows using Winsock2.