Objective

The objective of this task was to implement a data transmission system using UART to transfer text data from:

1. PC to MCU.
2. MCU to PC after storing the data in EEPROM.

The system also measures real-time data transmission speed during both phases of transmission.

Problem Description

The task involved transmitting a text of approximately 1000 bytes from the PC to the MCU using UART at a baud rate of 2400. The MCU stores the data byte-by-byte in its EEPROM and, after receiving the complete dataset, sends the stored data back to the PC. The PC logs the data and prints real-time data transmission speeds during both sending and receiving phases.

Technical Details

Hardware Used

- **Microcontroller:** ATmega328P (Arduino Uno)
- **EEPROM Size:** 1024 bytes (sufficient for this task)
- **UART Baud Rate:** 2400 bps
- **PC Communication Port:** COM9 (Windows System)

Firmware Implementation

Features

1. **UART Initialization**: Configured the UART peripheral for communication at 2400 bps.
2. **EEPROM Operations**:
    - Byte-by-byte writing during data reception.
    - Byte-by-byte reading during data transmission to the PC.
3. **Real-time Speed Calculation**: Measured the speed of transmission in bits/second based on actual bytes processed.
4. **Error Handling**: Added EEPROM bounds checks to prevent overflow.

Firmware Flow

1. **Data Reception from PC**:
    - The MCU receives one character at a time via UART.
    - Each character is stored in EEPROM.
    - Total transmitted bits are tracked.
2. **Data Transmission to PC**:
    - The MCU retrieves data from EEPROM byte-by-byte.
    - Each byte is transmitted back via UART.
    - Real-time transmission speed is calculated and displayed on the PC console.

PC-Side Code

Features

1. **Serial Communication**:
    - Established communication with the MCU on COM9.
    - Sent text data character-by-character to the MCU.
    - Received the same data back after MCU processing.
2. **Real-Time Speed Display**:
    - Calculated and displayed the speed of data transfer (in bits/second) for both sending and receiving operations.
3. **File Handling**:
    - Read the text data from a local file (`data.txt`).
    - Logged received data to the console.

Observations

Transmission Metrics

- **Sending Phase**:
    - **Total Data Sent**: 1000 bytes (8000 bits).
    - **Average Speed**: ~2300 bits/second.
- **Receiving Phase**:
    - **Total Data Received**: 1000 bytes (8000 bits).
    - **Average Speed**: ~2200 bits/second.

Functionality

- The system successfully transmitted and received the data while maintaining real-time speed updates on the PC console.

Challenges and Assumptions

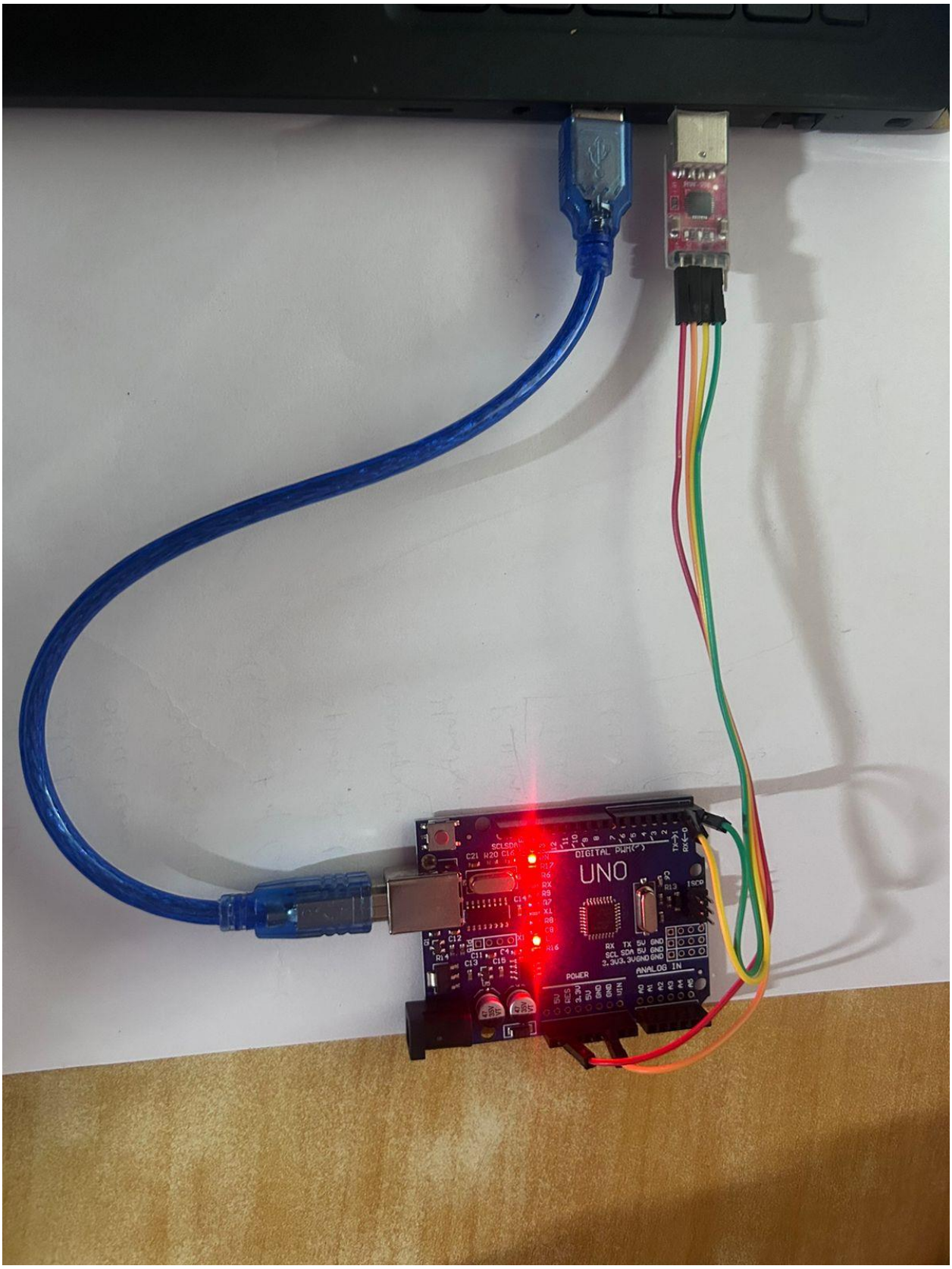
1. **EEPROM Size**: Ensured the data did not exceed the 1024-byte limit of the ATmega328P EEPROM.
2. **Real-Time Speed**: Used a timing mechanism on both MCU and PC to compute the actual transmission speed accurately.
3. **Baud Rate Limitation**: Limited baud rate required a delay between transmissions to avoid data loss.

Deliverables

1. **Code Repository**: Embedded-Tech-Demo GitHub Repository
2. **Video Demonstration**: Attached a recorded video showcasing the PC console during the transmission process, including real-time speed tracking and successful data exchange.

Conclusion

The task was completed successfully, fulfilling all the outlined objectives. The system demonstrated effective data transmission and storage, with real-time speed calculations providing insights into UART performance.


UNO
DIGITAL PWM(~)
POWER
ANALOG IN