시리얼통신 데이터값 확인

2조

Contents

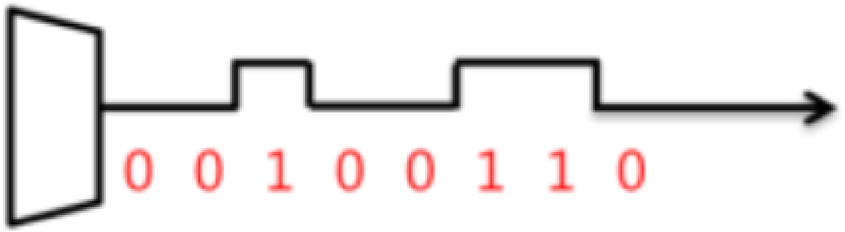
01 개요

목적

- 시리얼 포트를 통해 데이터를 수신하고, 수신된 데이터의 ASCII 값을 읽어 상위 비트 제거 및 보수 처리를 포함한 체크섬 계산을 통해 시리얼 통신의 기본적인 이해를 돕고, 데이터 무결성을 검증하는 방법을 학습하기 위함.

주요기능

- 데이터 수신, 체크섬 계산, 데이터 저장 및 조회

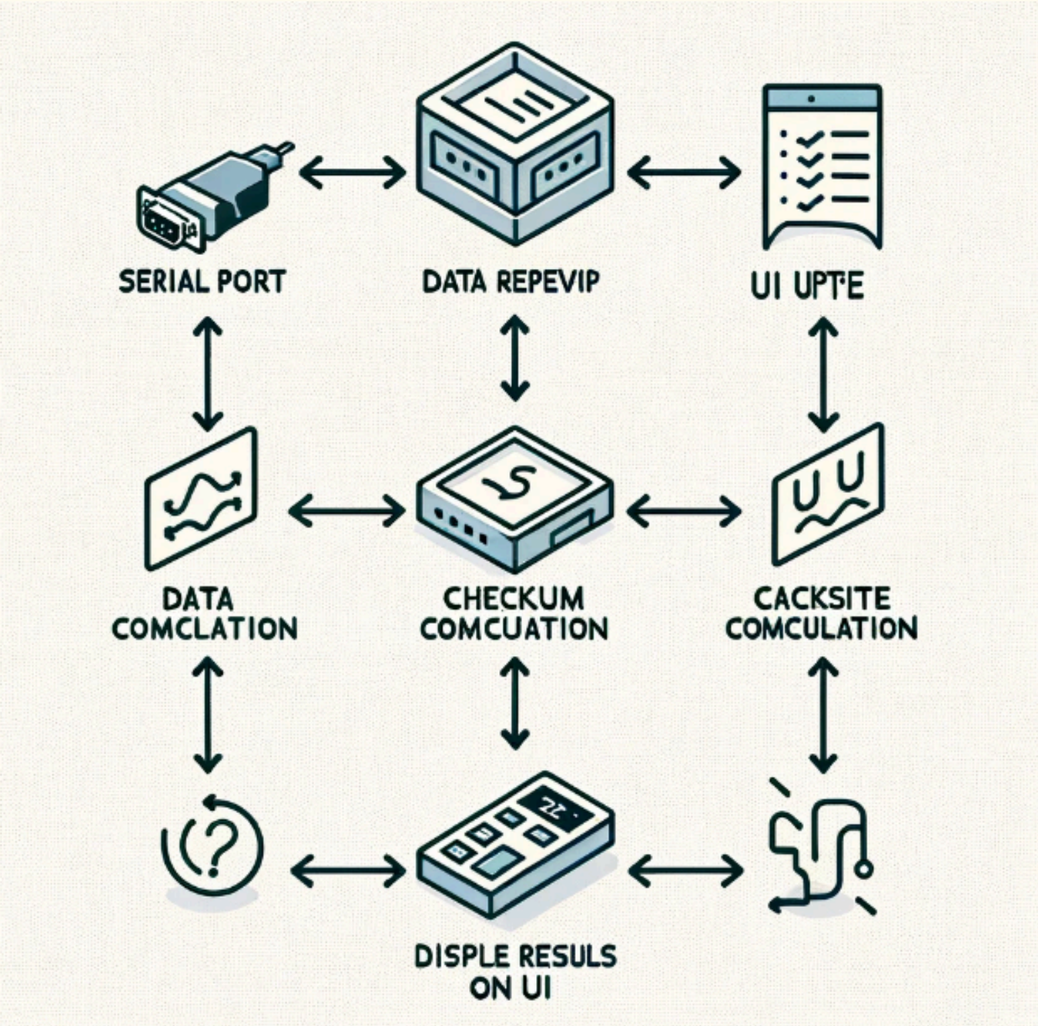


직렬 통신

02 시스템 구조

- 데이터 흐름:
  - 시리얼 포트 : 데이터 입력
  - 데이터 수신 : ASCII 값 수신
  - 체크섬 계산 : 수신된 값으로 체크섬 계산
  - UI 업데이트 : 결과를 UI에 표시

- 주요 컴포넌트:
  - SerialPort: 시리얼 포트 통신을 담당.
  - DataReceivedHandler: 데이터 수신 핸들러
  - UI 요소: 텍스트 박스, 버튼 등 UI 구성 요소



03 데이터 수신 및 처리

```
private void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
{
    SerialPort sp = (SerialPort)sender;
    while (sp.BytesToRead > 0)
    {
        int receivedAscii = sp.ReadByte();

        // 줄 바꿈 문자를 체크 (CR + LF)
        if (receivedAscii == 10 || receivedAscii == 13)
        {
            // CR 또는 LF를 수신하면 체크섬 출력
            this.Invoke(new MethodInvoker(delegate
            {
                textBoxOutput.AppendText($"체크섬: {checksum}{Environment.NewLine}");
                Console.WriteLine($"체크섬: {checksum}");
            }));
            checksum = 0; // 체크섬 초기화
            receivedAsciiList.Clear(); // 리스트 초기화
```

시리얼 포트에서 데이터가 수신될 때 호출됩니다. 수신된 ASCII 값을 읽고, 줄 바꿈 문자를 만나면 체크섬을 계산하고 출력합니다. 줄 바꿈 문자가 아닌 경우 ASCII 값을 체크섬에 추가하고 리스트에 저장합니다.

03 데이터 수신 및 처리

```
        // 만약 CR(13)을 수신했다면 LF(10)도 같이 수신될 수 있으므로 추가로 처리
        if (receivedAscii == 13 && sp.BytesToRead > 0 && sp.ReadByte() == 10)
        {
            // LF(10) 처리, 별도의 작업 없음
        }
    }
    else
    {
        checksum += receivedAscii;
        receivedAsciiList.Add(receivedAscii); // 리스트에 ASCII 값 저장

        char receivedChar = (char)receivedAscii; // 현재 수신된 문자 저장
        this.Invoke(new MethodInvoker(delegate
        {
            textBoxOutput.AppendText($"문자: {receivedChar}, ASCII: {receivedAscii}, 체크섬: {checksum}
{Environment.NewLine}");
            Console.WriteLine($"문자: {receivedChar}, ASCII: {receivedAscii}, 체크섬: {checksum}");
        }));
    }
  }
}
```

04 체크섬 계산

- 체크섬 계산 방식

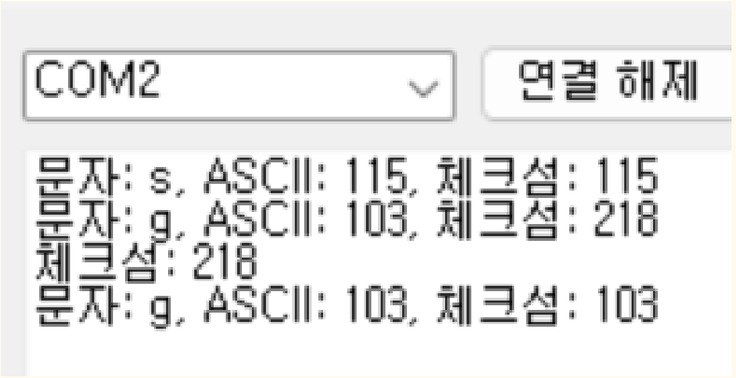
  체크섬은 수신된 모든 ASCII 값의 합계로 계산됩니다.
  이 합계는 8비트로 제한되며, 줄 바꿈 문자를 수신하면
  계산된 체크섬을 출력하고 초기화합니다.

```
// 줄 바꿈 문자를 체크 (CR + LF)
if (receivedAscii == 10 || receivedAscii == 13)
{
    // 체크섬 계산: 상위 비트 제거 후 보수 처리
    int checksumTotal = checksum & 0xFF; // 상위 비트 제거
    int checksumValue = ~checksumTotal + 1; // 보수 처리

    // CR 또는 LF를 수신하면 체크섬 출력
    this.Invoke(new MethodInvoker(delegate
    {
        textBoxOutput.AppendText($"체크섬: {checksumValue}
{Environment.NewLine}");
        Console.WriteLine($"체크섬: {checksumValue}");
    }));

    checksum = 0; // 체크섬 초기화
    receivedAsciiList.Clear(); // 리스트 초기화
}
```

05 UI 업데이트

- 수신된 데이터와 체크섬을 텍스트 박스에 출력

```
this.Invoke(new MethodInvoker(delegate
{
    textBoxOutput.AppendText($"문자: {수신된문자}, ASCII: {수신된Ascii}, 체크섬: {체크섬}{Environment.NewLine}");
    Console.WriteLine($"문자: {수신된문자}, ASCII: {수신된Ascii}, 체크섬: {체크섬}");
}));
```



수신된 데이터와 체크섬 값을 실시간으로 텍스트 박스와 콘솔에 출력하여 사용자가 확인할 수 있도록 합니다. Invoke를 사용하여 UI 스레드에서 UI를 업데이트합니다.

06 배운 것

이번 프로젝트를 통해 시리얼 통신, 데이터 무결성 검증, ASCII 코드 처리, 이벤트 기반 프로그래밍, UI 업데이트, 디버깅 및 문제 해결 등의 중요한 기술을 학습하였습니다.
임베디드 시스템, 네트워크 프로그래밍, 데이터 로깅 시스템 등 다양한 분야에서 유용하게 활용될 수 있습니다.

감사합니다