

Edge AI에서 빅 데이터를 전송하기 위한 UART 통신 분석

전제홍¹, 조정훈²

¹경북대학교 전자공학과 석사과정

²경북대학교 전자공학과 교수

jehongjeon@knu.ac.kr, jcho@knu.ac.kr

Analysis of UART Communication for Transmitting Big Data in Edge AI

Je-Hong Jeon¹, Jeong-Hun Cho²

¹Dept. of Electronic Engineering, Kyungpook National University

²Dept. of Electronic Engineering, Kyungpook National University

요 약

Edge AI 기술은 자동차, 가전, 스마트폰 등 우리 주변의 다양한 기기에 탑재되어 있다. Edge AI 를 구동하는 프로세서는 여러 종류로 나뉘는데, 대표적으로 저성능의 Microprocessor와 고성능 Microcomputer로 분류할 수 있다. 그중에서도 Microprocessor는 메모리와 저장 용량이 작아 Edge AI 를 구동하기 위한 빅 데이터를 메모리와 저장공간에 저장할 수 없기 때문에 통신을 사용하여 다른 기기로부터 데이터를 받아 연산을 수행해야 한다. 하지만 Microprocessor에서 통신은 빅 데이터와 같은 숫자로 이루어진 값을 전송하는 데에만 사용되는 것이 아니다. 디버깅이나 Processor의 정보 표시 등 문자열을 함께 사용하는 경우가 많은데, 문자열과 숫자 데이터를 함께 주고받으면 빅데이터와 같은 많은 데이터를 전송할때 시간이 오래 걸린다는 문제가 있다. 본 논문에서는 Edge AI에서의 빅 데이터를 빠르게 전송할 수 있는 방법을 제안한다.

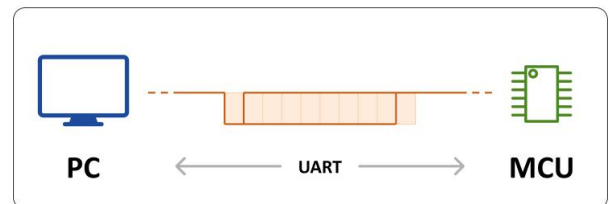
1. 서론

오늘날 Edge AI 기술은 단순한 가전제품부터 첨단 기술의 집약체인 스마트폰까지 다양한 기기에 탑재되어 있다[1]. Edge AI 를 구동하기 위해서는 Microprocessor 또는 Microcomputer가 탑재되어야 한다. Microprocessor는 Edge AI를 구동하는데 있어 메모리 용량과 저장공간이 작고, 처리 속도가 느리다는 한계점을 가지고 있음에도 불구하고 저전력, 저비용의 장점으로 인해 많이 사용되고 있다[2]. Microprocessor의 적은 메모리 용량을 극복하기 위한 방법으로 Edge AI에서 사용될 빅 데이터를 다른 기기로부터 통신을 통해 데이터를 받아 연산을 수행할 수 있다. Microprocessor에서 통신으로 정보를 주고받는 상황은 디버깅, 상태 메시지 표시, 변수 값 전달 등 여러가지가 있다. 문자열 전송은 정보의 길이 제한이 없고 다양한 종류의 데이터를 전송할 수 있기 때문에 Microprocessor에서 통신으로 정보를 주고받는 보편적인 방법으로 사용된다. 하지만 문자열을 통해 데이터를 전송하게 되면 바이트로 데이터를 전송한 것에 비해 데이터 전송 시간이 느리다는

문제가 있다. 본 논문에서는 위 현상이 나타나는 원인을 분석하고 빅 데이터를 효과적으로 전송하기 위한 방안을 제안한다.

2. Edge AI에서 빅 데이터를 전송하기 위한 UART 통신 분석

Microprocessor는 STMicroelectronics 사의 STM32F411 프로세서를 사용하였고, 빅 데이터를 전송하기 위한 장치로 X86 기반의 Windows 기반 PC를 사용하였다. 그림 1은 Microprocessor와 PC가 UART 통신을 통해 정보를 주고받는 모습을 나타낸다. UART 통신 설정은 표 1 과 같이 설정하였다.

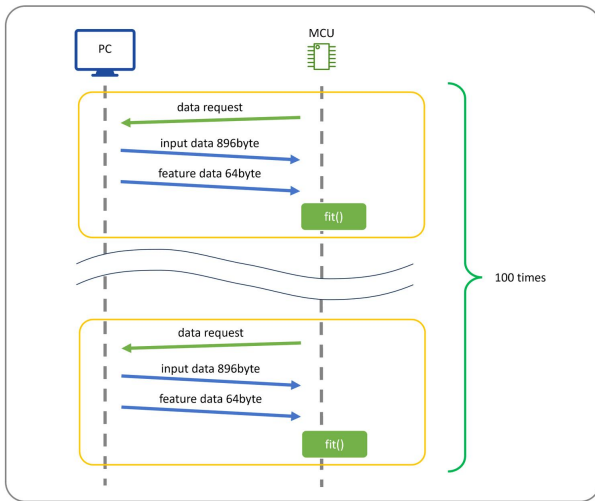


(그림 1) Microprocessor의 UART 통신 예

UART 통신 설정	
baud Rate	115200 bits/s
Word Length	8 bit
Parity	No Parity
Stop Bits	1 bit

<표 1> UART 통신 설정 값

그림 2는 본 연구에서 사용한 Edge AI의 데이터를 전송하기 위해 구성한 코드의 통신 흐름도를 나타낸다. Edge AI 데이터의 input 이 28 byte, feature 가 2 byte로 구성된 데이터가 3200세트 구성되어 전체 데이터의 크기는 총 96000byte 이고, 한 번의 전송에 32세트씩 전송하여 총 100번의 batch를 반복한다. PC 에서 9데이터는 csv 형식의 텍스트 파일로 저장되어 있고, 이를 Serial 통신을 이용하여 전송하면 Microprocessor 에서 scanf 를 사용하여 수신된 데이터를 읽도록 구성한 결과, 96000byte의 데이터를 모두 수신하는데 대략 25분이 소요되는 것을 확인할 수 있었다.



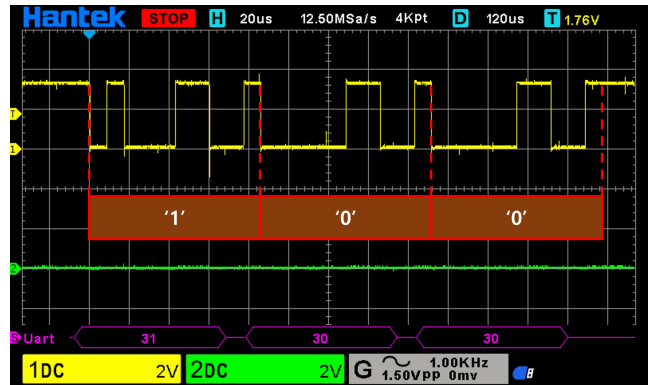
(그림 2) Edge AI의 빅 데이터 전송 흐름도

하지만 UART 통신 속도를 통해 96000byte 를 전송하는데 걸리는 시간을 계산해보면 위 결과가 지나치게 오래 걸렸다는 것을 알 수 있다. UART 통신에서 1byte 의 데이터를 보내기 위해서는 Start bit 와 Stop bit 를 포함하여 총 10bit를 전송해야 한다 [3]. 데이터 1byte를 보내는 데 필요한 시간은 수식 1과 같이 계산할 수 있다.

$$1s \div 115,200bits/s \times 10bit = 86.81us \quad (1)$$

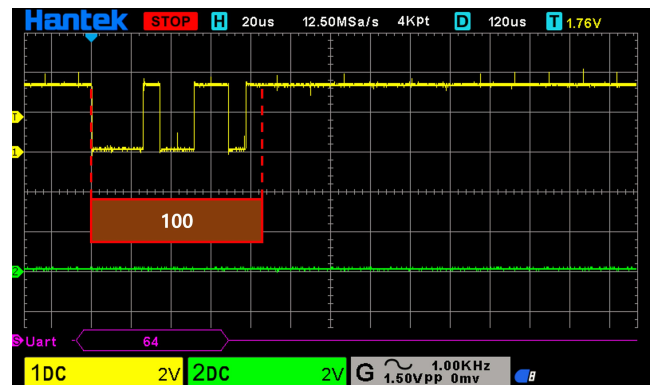
따라서 96000byte의 데이터를 전송하는데 걸리는 시간은 $86.81us * 96000byte = 8.33s$ 이다. 이러한 현상이 나타난 원인을 분석한 결과 scanf 를 사용한 것이 가장 큰 원인으로 분석되었다.

scanf 는 문자열을 읽어 숫자로 변환해주는 역할을 한다. 이는 PC로부터 데이터가 문자열로 전송이 되었다는 것을 뜻하며, 그림 3과 같이 숫자 100을 전송하기 위해 '1' '0' '0' 과 같이 3byte로 나누어 보낸다. 또한 scanf 는 개행 문자가 수신될 때까지 대기하고, 문자열에서 숫자 데이터를 추출하는 작업을 추가로 진행하므로 추가적인 시간이 더 소요된다.



(그림 3) UART 통신으로 100을 문자열로 전송하는 모습

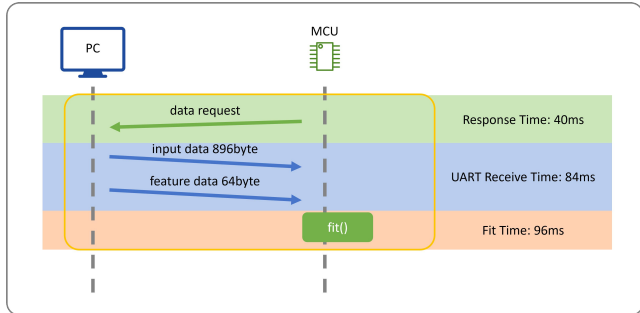
이 문제를 해결하고자 그림 4와 같이 데이터를 byte 단위로 전송하기 위해 기존의 scanf를 HAL_UART_Receive로 변경한 결과, 96000byte의 데이터를 모두 수신하는데 걸린 시간은 약 22초 가량이 소요되는 것으로 확인되었다.



(그림 4) UART 통신으로 100을 바이트로 전송하는 모습

그림 5는 문제 개선 후 1batch에 걸리는 시간을 분석한 것이다. 1batch 에서 UART 수신에만 소요

된 시간은 84ms 이고, 이를 100번 반복했을 때 UART 수신에 소요되는 시간은 총 8.4s 로 계산된다. 이는 앞서 수식 (1)을 통해 계산된 96000byte의 UART 전송 시간인 8.33s 과 거의 일치하는 것을 확인할 수 있다.



(그림 5) 개선 후 동작 시간 분석

결론적으로 문자열 전송 방법을 byte로 전송으로 변경하여 98.54% 속도 개선을 이뤄낼 수 있었다. 빅 데이터와 같은 대용량의 데이터를 주고받는 상황에서는 문자열 데이터를 주고받는 것보다 바이트 단위의 데이터를 사용하는 것이 전송 속도 면에서 훨씬 더 유리하다는 것을 알 수 있다.

ACKNOWLEDGEMENT

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 1711160343. 차량 ECU 응용소프트웨어 개발 및 검증자동화를 위한 가상 ECU 기반 차량레벨 통합 시뮬레이션 기술개발).

참고문헌

- [1] S. U. Amin and M. S. Hossain, "Edge Intelligence and Internet of Things in Healthcare: A Survey," in *IEEE Access*, vol. 9, pp. 45-59, 2021, doi: 10.1109/ACCESS.2020.3045115.
- [2] B. Sudharsan, J. G. Breslin and M. I. Ali, "ML-MCU: A Framework to Train ML Classifiers on MCU-Based IoT Edge Devices," in *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15007-15017, 15 Aug.15, 2022, doi: 10.1109/JIOT.2021.3098166.
- [3] R. K. Agrawal and V. R. Mishra, "The design of high speed UART," 2013 IEEE Conference on Information & Communication Technologies, Thuckalay, India, 2013, pp. 388-390, doi: 10.1109/CICT.2013.6558126.