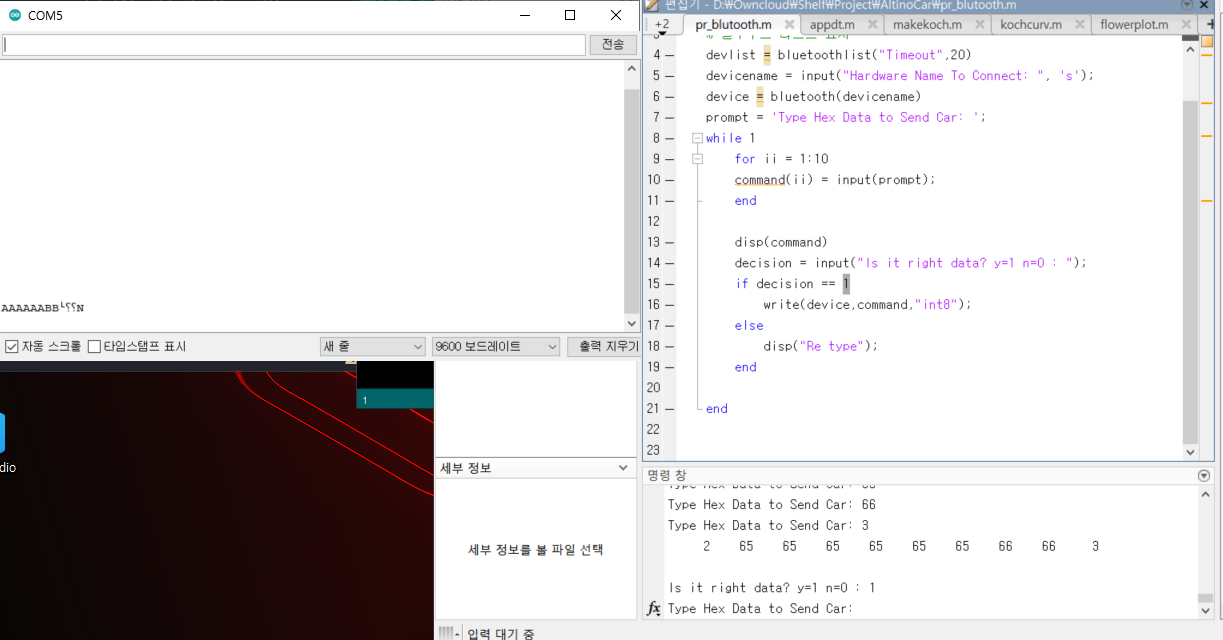
2021-04-28 시리얼 통신 과정 에서 의 문제1



매트랩 에서의 전송

0x02 0x65 0x65 0x65 0x65 0x65 0x65 0x66 0x66 0x03

STX A A A A A A A B B ETX

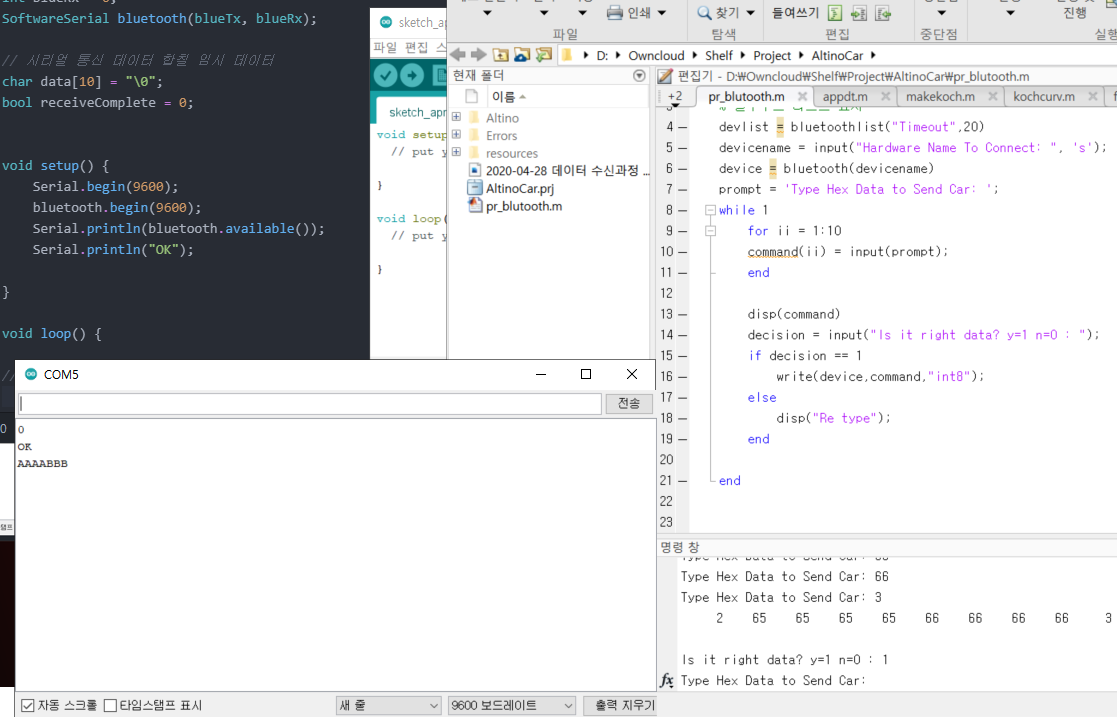
하지만 수신받은 데이터는

AAAAAABB⸮⸮N

내가 생각한 원인

1. 아두이노 데이터 수신 코드에서 Bluetooth.read() 에서 for 문 반복 횟수가 데이터 보다 많기 때문에 발생
2. 아두이노에서 데이터를 수신 받고 이를 컴퓨터 시리얼 모니터로 전송하는데 코드 위치가 아두이노의 loop{} 문 안에 있어 블루투스로 들어온 데이터가 없어도 계속적으로 시리얼 모니터로 가비지 값이 전송되는 문제

해결!



<소스 코드>

void loop() {

// 블루투스 버퍼크기 10까지 대기

if (bluetooth.available() >= 10) {

// 수신 패킷 STX 확인

if (bluetooth.read() == 0x02) {

// 수신 받은 데이터 합치기

for (int i = 0; i<7; i++) {

data[i] = bluetooth.read();

}

// 데이터 수신완료후 컴퓨터 시리얼 모니터로 전송

receiveComplete = 1;

Serial.write(data);

Serial.print("\n");

}

// 시작 바이트가 STX가 아닌경우 데이터 폐기

else {

while(bluetooth.available()) {

bluetooth.read();

}

}

}

}

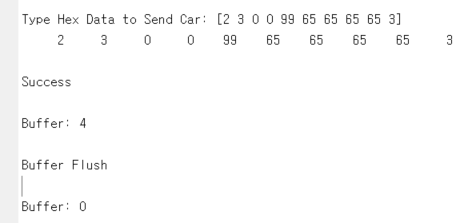
원인은 내가 예상한 1,2 번 모두 인 것 같다.

먼저 for문 반복 횟수를 기존 9회 에서 7회로 줄였고

결과값 출력을 기존 loop문 맨 마지막에서 실행 되는 것이 아니라 블루투스 데이터 수신 과정의 for문 실행이 끝나자 마자 PC로 시리얼 모니터를 보내는 것으로 작성 하였다.

잘못 전송된 데이터가 버퍼에 남아 있으면 향후 통신에 장애를 가져올 수 있기 때문에 잘못된 데이터를 폐기하는 코드도 삽입 하였다.

2021-04-29 버퍼에 가비지 데이터가 있는 오류





문제는 알티노에서 데이터를 다 읽었는데도 버퍼에 데이터가 남았다는 것이다.

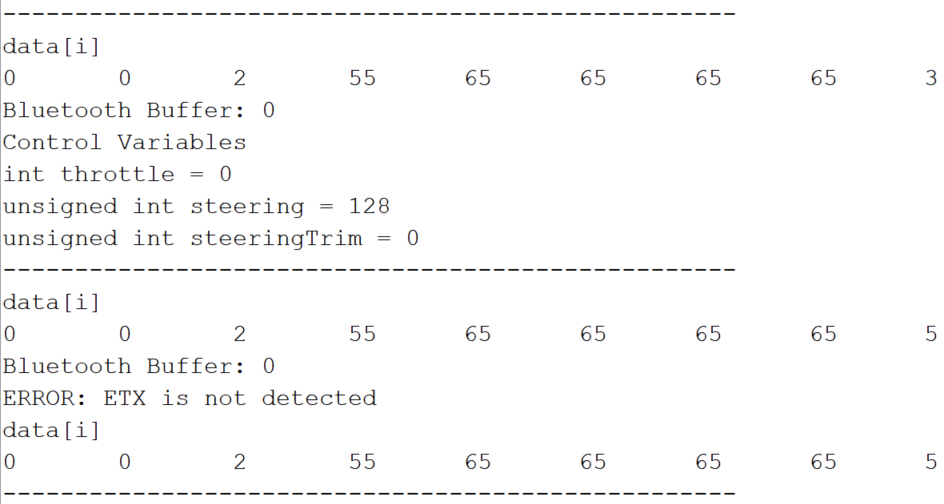
일단 발견한 오류는 data[i] for 문에서 데이터를 모두 읽어 들이지 않아 2개의 데이터가 남아서 그런건 지 모르겠다 일단 수정.

중간에 개발 하면서 패킷 검증 알고리즘 추가 했는데 버그가 생겼다.

버그는 정상인 패킷도 모두 비정상으로 판별하는 것 이었는데 한 1시간 씨름하다가 버그를 수정 했다. 원인은 단순한 오타와 형변환을 안해서 생긴 문제 였다.

이 사건을 계기로 디버그 모드를 만들었다. 전처리기 를 사용해 코드상에 DEBUG\_MODE 가 정의되어 있으면 디버그 모드로 작동되고 아니면 릴리즈 모드 즉 실제 작동 되는 모드로 구동 되게 하였다.

또한 디버그 모드에서는 자동차는 작동 되지 않게 했고, 시리얼 모니터를 통해 알티노 내부의 변수 값 등으로 볼 수 있게 하였다.



2021-04-30 알티노 전진 제어 변수 throttle 이상 현상

디버그 모드를 활용하며 개발 하던 도중, 중요한 것을 발견 했다.

분명 나는 속도 300 을 의도로 데이터를 전송 했는데 알티노 에서는 전진값이 700 가까이 나와있는 것이다.

디버그 모드가 빛을 보인 순간이다. 디버그 모드 없었 을 때는 작동 하는지 안하는지 로만 오류를 판단 했는데, 이 경우도 작동은 됬었다. 하지만 실제로는 나의 의도와 다른 제어값이 된게 큰 문제 였다.

원인을 파악해 본 결과 비트 시프트 연산자가 문제였다.

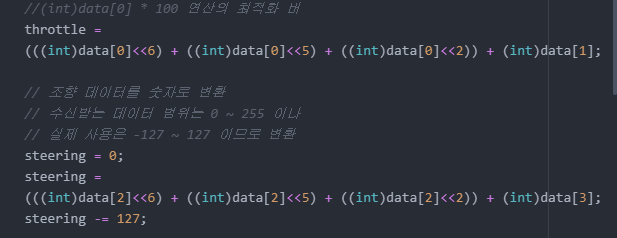
내가 speed\_H 바이트를 << 8 비트 시프트 연산을 수행했는데, 이는 X100 이 아닌 X256 이었다.

그걸 착각하고 계속 사용해 왔던 나였다.

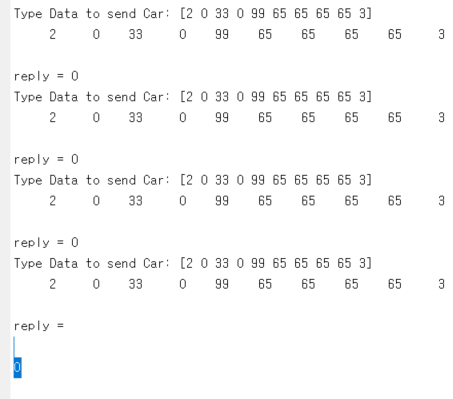
이를 알아내고 바로 수정을 했다. 그냥 (int)data[0] \* 100 을 하려고 했으나 해당 연산은 간단하지만 실시간으로 연산하는 만큼 최대한 연산 시간을 줄이려고 비트 시프트를 이용해 곱셈 연산을 이용 했다.

X 100 은 (64 + 32 + 4) 로 인수분해가 가능하므로 이는 2^6, 2^5, 2^2 이다

즉 <<6 <<5 <<2 의 비트 시프트 연산과 같고 이를 활용했다.



2021-05-01 알티노 수신응답 관련 이슈



버그는 매트랩에서 데이터를 보냈을 때 정상 데이터면 알티노에서 0을 리턴 하는데

문제는 가끔씩 리턴 값에 다른 데이터 ( 예를 들어 줄바꿈 문자 ) 등이 전송 되는 현상이 발견됬다.

이를 해결하기 위해 알티노의 수신응답 부분을 확인했으나 오류를 찾지 못하였다.

그래서 생각한 방법은

1. 어차피 다른 데이터가 섞여 들어가도 0 이라는 알티노가 보낸 성공 응답은 존재하므로 다른 문자를 무시하고 숫자만 인식하는 코드를 짜기.

해당 문제를 해결했다.

for ii = 1:10

reply = read(device,1,"string");

switch reply

case '0'

fprintf("reply = %s\n", reply);

break;

case 'e'

fprintf("reply = %s\n", reply);

break;

case 's'

fprintf("reply = %s\n", reply);

break;

end

end

먼저 for 문을 통해 10번 정도 수신을 받을수 있게 했고, 이후 switch 문을 통해 응답값에 따른 실행 명령을 두었다.

따라 잘못된 데이터가 들어오면 10번 재수신이 이루어 질 수 있도록 하였다.

2021-05-01 블루투스 통신 지연 시간 실험



대상과의 거리 1m

실험 장소 : 내방

실험 시간 2021-05-01 11:40

모듈모델명: HC-06

BaudRate : 9600bps

평균지연시간 : 71.6221 ms

최대지연시간 : 184.1763 ms

최소지연시간 : 39.6085 ms

지연 시간을 줄이기 위한 노력

1. 보드레이트를 높인다 9600bps -> 115200bps

* 알아본 결과 소프트웨어 시리얼은 9600bps 이상부터 정상 작동 이 안되기 때문에 115200bps 속도로 블루투스 시리얼 통신을 하려면 UART 에 직접 연결 해야 하는데 알티노는 아두이노와 알티노 본체와의 통신을 UART 로 통신 하므로 연결이 불가능 하다.

1. 패킷을 최적화 한다

* 여기서 더 최적화 할 수 있는지 알아 봐야 겠다

< 알티노로부터 센서 데이터를 받아올 때 지연시간 >

. 

평균지연시간 : 109.4723 ms

최대지연시간 : 172.1688 ms

최소지연시간 : 88.8932 ms

< 알티노 센서값 전송 알고리즘 개선후 >



------------- 실험 결과 ---------------

평균 레이턴시 : 44.55151ms

최대 레이턴시: 323.5748ms

최소 레이턴시: 2.786100ms

Success: 100 Fail: 0>>

2021-05-21

알티노 개발 마무리 및 feature 개발

Feature 으로 알티노의 부가 장비를 제어 기능 추가

전조등,. 클락션 등

이때 1 바이트안에 비트 연산으로 모든 기능 제어

개발 성공 모든 기능 정상 작동

클락션기능 및 Led 제어

소스코드 모듈화 완료

함수를 기능별로 분리

<Open space - 1m 거리 에서 통신 성능 >



------------- 실험 결과 ---------------

평균 레이턴시 : 142.7779 ms

최대 레이턴시: 199.0507 ms

최소 레이턴시: 127.4524 ms

Success: 200 Fail: 0

Dataloss: 0 MissingSTX: 0 MissingETX:0

손실율: 0 %

<Open space - 2m 거리 에서 통신 성능 >



------------- 실험 결과 ---------------

평균 레이턴시 : 138.4661 ms

최대 레이턴시: 414.6047 ms

최소 레이턴시: 92.42500 ms

Success: 200 Fail: 0

Dataloss: 9 MissingSTX: 1 MissingETX:0

손실율: 5 %

<Open space - 3m 거리 에서 통신 성능 >



------------- 실험 결과 ---------------

평균 레이턴시 : 139.7327 ms

최대 레이턴시: 214.7232 ms

최소 레이턴시: 97.36470 ms

Success: 200 Fail: 0

Dataloss: 2 MissingSTX: 0 MissingETX:0

손실율: 1 %

<Close space - 5m 거리 에서 통신 성능 >

* 중간에 벽 존재



------------- 실험 결과 ---------------

평균 레이턴시 : 1.381045e+02

최대 레이턴시: 4.436091e+02

최소 레이턴시: 42.46160e+01

Success: 200 Fail: 0

Dataloss: 6 MissingSTX: 0 MissingETX:0

손실율: 3 %>>