

Line Coding

2016707079 하상천

1. 모든 line coding 경우에 대해서 어떻게 0, 1을 mapping해서 보내는가?

NRZ-L : non return to zero-level, 예를 들면 0은 +V, 1은 -V처럼 2개의 voltage level 안에서 비트 값에 따라 voltage level을 결정한다.

NRZ-M : 0은 신호 변화 없이 전 상태 유지, 1은 신호 변화 발생(뒤집기)

NRZ-S : 0은 신호 변화 발생(뒤집기), 1은 신호 변화 없이 전 상태 유지

Unipolar RZ : 0은 pulse가 없는 0, 1은 절반 비트만 pulse 값을 갖는다. 즉 0은 0, 1은 절반 비트만 값을 갖고 나머지 절반 비트는 0이다.

Bipolar RZ : 0과 1이 반대 level pulse값으로 표현 된다. 즉 1은 절반 비트가 +V이고 나머지 절반 비트는 0이고, 0은 절반 비트가 -V이고 나머지 절반 비트가 0이다.

RZ-AMI : 0은 pulse가 없는 0, 1은 +V와 -V를 서로 교번하여 부호화
즉 0은 0, 1은 직전에 +V이면 그 다음에는 -V로 극성을 번갈아 대응시킴.

Bi- ϕ -L : Manchester code, 0은 negative에서 positive로 이동, 1은 positive에서 negative로 이동
즉, 0은 절반 비트가 -V 나머지 절반 비트는 +V 이고, 1은 절반 비트가 +V 나머지 절반 비트가 -V이다.

Bi- ϕ -M : 0과 1 둘 다 시작 전에 신호 변화 발생(뒤집기)이 일어난다. 0은 시작 전에 뒤집고 그 값을 계속 유지하고, 1은 시작 전에 뒤집고 절반 비트만

큼 그 값을 유지한 후 나머지 절반 비트는 다시 뒤집는다.

Bi- ϕ -S : 위의 방법과 반대로, 0은 시작 전에 뒤집고 절반 비트만큼 그 값을 유지한 후 나머지 절반 비트는 다시 뒤집는다. 1은 시작 전에 뒤집고 그 값을 계속 유지한다.

Delay modulation : 0이 연속적으로 나오지 않으면 이전 값을 계속 유지하고, 연속적으로 나올 시 뒤집는다. 1은 이전 값을 절반 비트만큼 유지하고 나머지 절반 비트는 뒤집는다.

Dicode NRZ : 0 다음 1이 나오거나, 1 다음 0이 나오면 극성이 바뀐다. 하지만 똑같은 값이 나오면 0을 보낸다.

Dicode RZ : 0 다음 1이 나오거나, 1 다음 0이 나오면 극성이 바뀌는데 절반 비트 동안은 값을 유지하고 나머지 절반 비트는 0을 보낸다. 똑같은 값이 나오면 0을 보낸다.

2. 이중에 맨체스터 코드는 누구인가?

-> Bi- ϕ -L 이다.

3. 이러한 line coding을 사용해서 보내면 얻을 수 있는 장점이 무엇인가?

잡음에 강인하며, 에러 검출이 뛰어난 장점을 보유하고 있다. 특히, 신호에 직류(DC) 성분을 포함하고 있지 않기 때문에 baseline wandering을 제거하는 회로가 필요하지 않다. 또한 연속적인 0과 1에 대해 synchronization을 할 수 있고, 자체 클럭킹을 통해 에러율이 낮고 안정적인 전송을 할 수 있다.