

컴퓨터네트워크

실습 1 결과 보고서

2022.05.02

컴퓨터과학과 201810912 김연출

코드 동작 설명

```
COM7 - PuTTY
----- ARQ protocol starts! -----
:: ID for this node : :: ID for the destination : endnode : 7, dest : 6
>>>> HAL OnTxDone Function

>>>> HAL OnTxTimeout

>>>> HAL OnRxError

>>>> HAL OnRxTimeout

      Initializing Radio Interrupt handlers
Give a word to send : word is ready! ::: hello
[MAIN] sending to 6 (seq:-1)
Give a word to send :
-----
RCVD from 6 : world (length:8, seq:0)
-----
Give a word to send : 
```

```
COM6 - PuTTY
----- ARQ protocol starts! -----
:: ID for this node : :: ID for the destination : endnode : 6, dest : 7
>>>> HAL OnTxDone Function

>>>> HAL OnTxTimeout

>>>> HAL OnRxError

>>>> HAL OnRxTimeout

      Initializing Radio Interrupt handlers
Give a word to send :
-----
RCVD from 7 : hello (length:8, seq:0)
-----
Give a word to send : word is ready! ::: world
[MAIN] sending to 7 (seq:-1)
Give a word to send : 
```

1. Tx와 Rx 측 모두 Main 함수가 실행되면 endNode_id 와 dest_id를 사용자로부터 입력받아서 초기화하고 화면에 출력합니다.
2. arqLLI_initLowLayer 함수를 통해서 하위 레이어들을 초기화합니다. arqLLI_dataCnffunc 함

수를 통해서 DATA_CNF case일 때 arqEvent_setEventFlag 함수로 eventflag 값을 설정하고, arqLLI_dataIndFunc 함수를 통해서 DATA_IND case일 때 eventflag를 set 합니다.

3. pc.attach(&arqMain_processInputWord, Serial::RxIrq); 코드를 통해서

serial 클래스로 argMain_processInputWord 함수를 콜백 지정한다. 사용자가 송신할 메시지를 입력하는 것을 기다리는 상태이다.

사용자가 문자를 입력하면 originalWord에 문자열로 저장하고, eventFlag를 set 합니다. 오류 없이 준비가 되었다면 word is ready! 를 출력하고 그렇지 않으면 (길이가 상한을 넘어가면) max reached! Word forced to be ready를 originalWord와 함께 출력합니다.

4. While 문으로 들어오면서 debug 과정을 거치고 main_state에 따른 switch 문을 만나게 됩니다.

5-1. 송신 측

위의 arqMain_processInputWord 함수에서 eventFlag를 arqEvent_dataToSend로 set 하기때문에 idle 상태를 멈추고 조건 문에서 else if(arqEvent_checkEventFlag(arqEvent_dataToSend)로 분기합니다.

5-1-1. 메시지를 보내기 위해 encoding 하는 과정입니다. Pdu를 originalword, 시퀀스 넘버, 문자열 길이 등을 통해서 encoding 합니다.

5-1-2. arqLLI_sendData를 통해서 encoding된 pdu를 dataReq 동작합니다.

5-1-3. 목적지 id와 시퀀스 넘버를 포함한 메시지를 화면에 출력합니다.

5-1-4. main_state를 MAINSTATE_TX로 바꾸고 flag_needprint도 1로 초기화 합니다. wordLen도 0으로 clear , eventFlag도 arqEvent_clearEventFlag 함수를 통해서 clear 합니다.

5-1-5. rx측에서 arq를 받으면 arqLLI_dataCnfFunc에서 arqEvent_setEventFlag 함수를 통해서 eventFlag 값을 arqEvent_dataTxDone으로 바꿔줍니다.

5-1-6. eventFlag 값이 arqEvent_dataTxdone이므로 case MAINSTATE_TX의 조건문으로 분기하여 main_state를 MAINSTATE_IDLE로 바꾸고 eventFlag를 clear 합니다.

5-1-7. mainstate가 idle이고 flag_needprint 가 1이므로 "Give a word to send" 메시지를 출력합니다.

5-2. 수신 측

5-2-1. 데이터를 수신받으면 arqLLI_dataIndFunc 함수를 실행합니다. 함수에서는 memcpy를 통해서 수신받은 dataPtr을 복사하고 나머지 값들도 수신받은 정보로부터 얻어옵니다. 조건문에 맞게 분기하여 eventFlag 값을 set 합니다.

5-2-2. 원래 수신측은 idle 상태였지만 조건문에 맞게 eventFlag 가 변경되어 조건문으로 분기합니다. 함수를 토대로 수신받은 data의 소스와 내용과 크기에 대한 정보를 얻습니다.

5-2-3. arqMsg_getWord(), arqMsg_getSeq()함수를 이용하여 내용과 길이와 시퀀스 넘버를 포함한 메시지를 화면에 출력합니다.

5-2-4. 문자열 길이와 개행문자와 +ARQMSG_OFFSET_DATA의 합인 값이 출력이 되고, 시퀀스 넘버가 출력됩니다.

5-2-5. 수신이 완료가 되면 수신측은 더 이상 할일이 없기 때문에 idle 상태로 설정하고 eventFlag를 clear 합니다. 또한 fla_needPrint를 1로 set 해서 앞선 과정에서처럼 입력요청 메시지를 출력합니다.