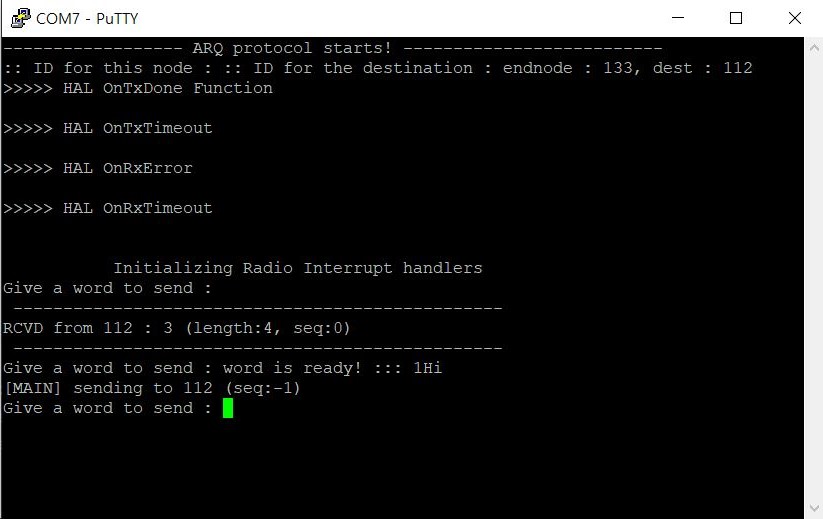
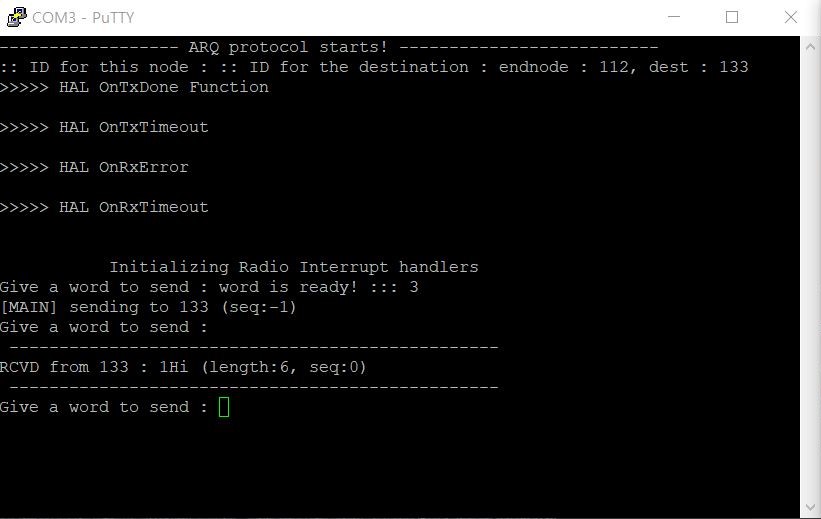
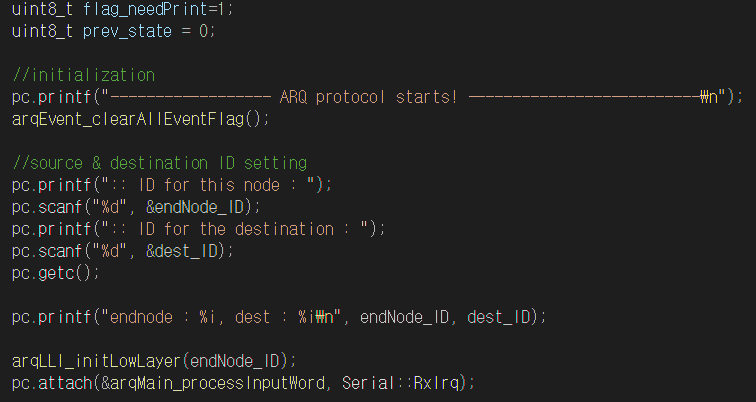
1. 결과 캡쳐

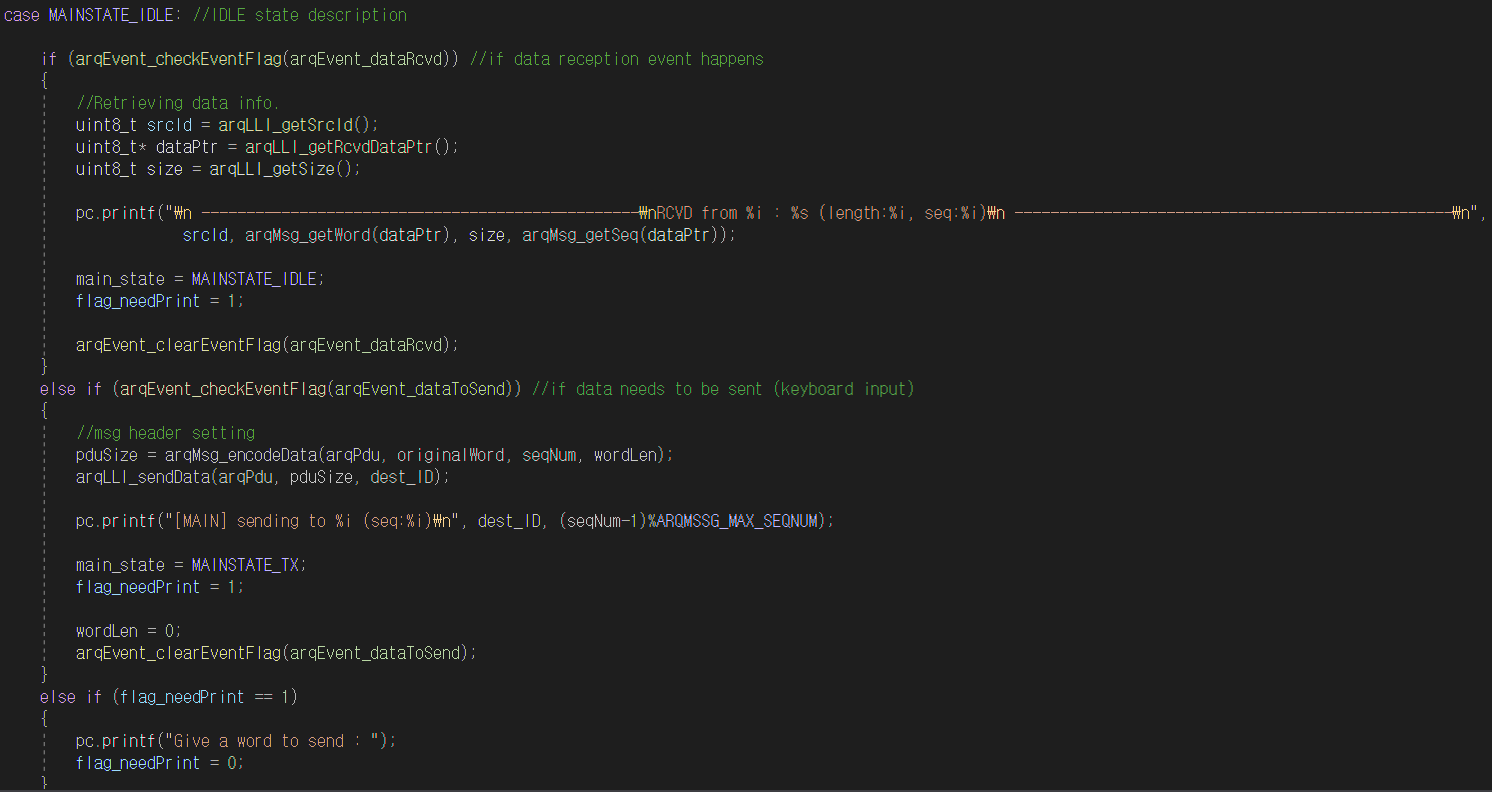


1. 동작과정 설명

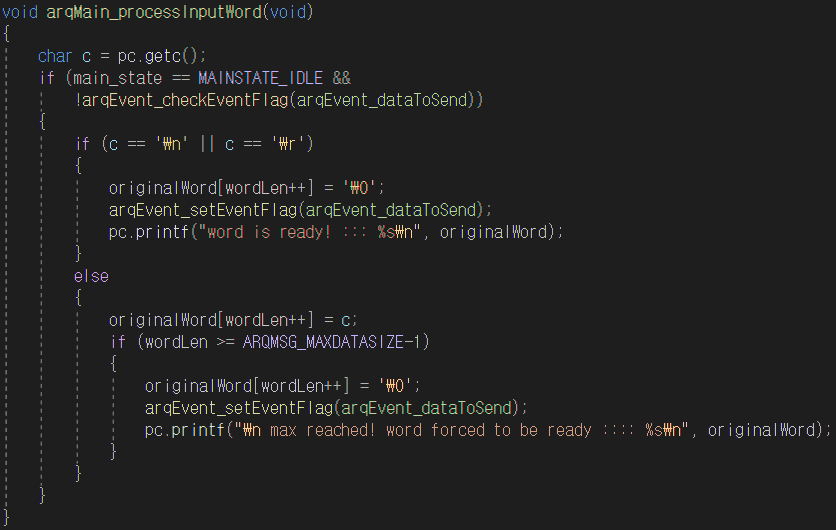
**src 노드의 ID 가 1이고, dest 노드의 ID가 2일때, src에서 dest노드로 "hello" 라는 메시지를 보내는 경우의 모든 코드 동작을 설명하시오. (노드의 ID 세팅은 완료되어 IDLE 상태로 정상동작중인 상태로 가정)**



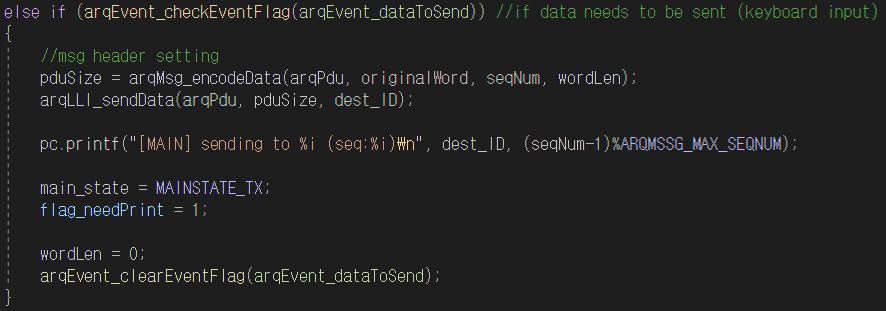
main.cpp의 코드의 main부분에서 처음 초기화와 세팅하는 위 부분에서 src 노드의 경우 endNode\_ID가 1, dest 노드의 경우 2로 설정 되어있고, dest\_ID의 값은 서로의 ID로 설정 되어있다. 또한 각 노드의 상태에 관한 변수인 main\_state는 미리 정의된 MAINTSTATE\_IDLE, 0으로 초기화 되어있다. arqEvent\_clearAllEventFlag(); 함수를 통해 각 노드의 모든 event flag값을 clear한다. arqLLI\_initLowLayer(endNode\_ID); 함수를 통해 각 노드의 PHYMac 주소를 초기화 한다. 그리고 각 노드의 puTTy 화면에 자신의 ID와 destination 노드의 ID가 출력된다. pc.attach(&arqMain\_processInputWord, Serial::Rxlrq); 함수를 통해 puTTy에서 arqMain\_processInputWord(void) 함수를 통해 계속 입력을 받는다.

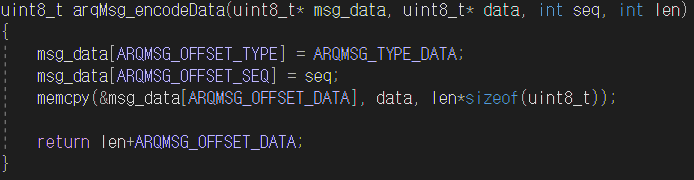


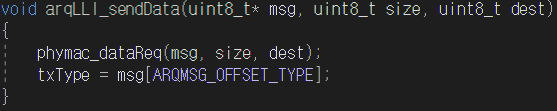
while(1)문에서 각 노드는 prev\_state의 값과 main\_state의 값이 0으로 동일 하므로, 첫번째 if(prev\_state != main\_state)문에 걸리지 않는다. 다음 switch(main\_state)문에서 case MAINSTATE\_IDLE:에 걸리게 되고, 처음에 모든 event flag를 clear했기 때문에 else if(flage\_needPrint == 1)문에 걸리게 되어, src와 dest 노드 둘 다 puTTY 화면에 “Give a word to send : ”가 출력된다.

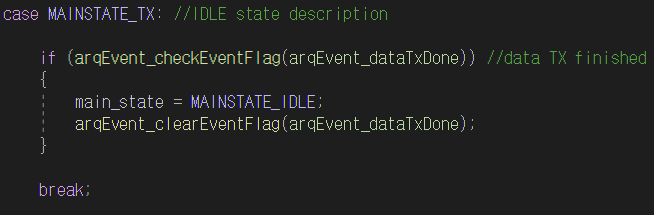


먼저 src 노드(Tx)에서 “hello”라는 메시지가 입력이 되면, arqMain\_processInputWord(void)함수를 통해 hello를 입력받고 arqEvent\_dataToSend의 event flag를 arqEvent\_setEventFlag(arqEvent\_dataTosend)함수를 통해 setting하고 “word is ready! ::: hello”가 화면에 출력된다. 이때 변수 unsigned char 배열 originalWord에는 ‘hello\0’이 저장되고, 변수 wordLen에는 6이 저장이 된다.

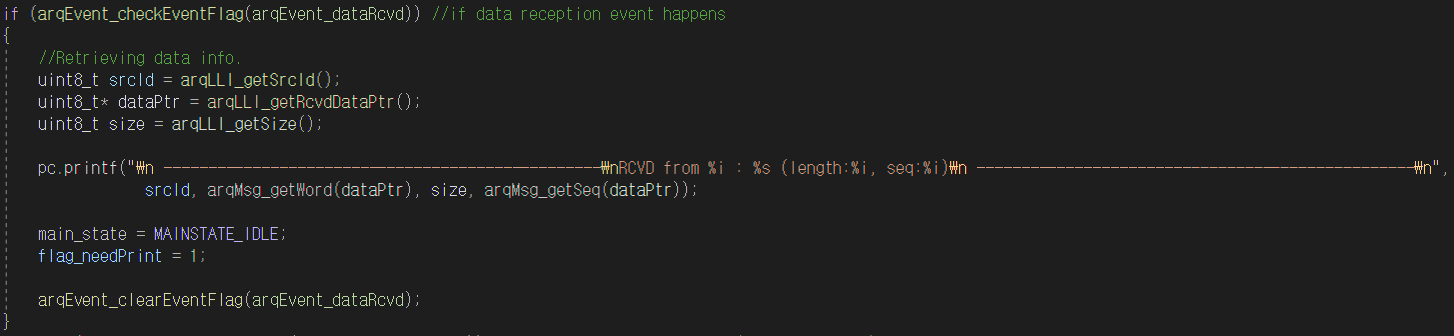








src 노드에서 메시지를 입력 받아 arqEvent\_dataToSend flag가 setting됐으므로, case MAINSTATE\_IDLE:에서 else if (arqEvent\_checkEventFlag(arqEvent\_dataToSend))문에 걸린다. 이때 arqMsg\_encodeData(arqPdu, originalWord, seqNum, wordLen) 함수를 통해 arqPdu[ARQMSG\_OFFSET\_TYPE]에 0을 저장하고, arqPdu[ARQMSG\_OFFSET\_SEQ]에 0을 저장한다. arqPdu[ARQMSG\_OFFSET\_DATA]부터 originalWord를 memcpy하고, wordLen에 ARQMSG\_OFFSET\_DATA를 더한 값 8을 변수 pduSize에 저장한다. arqLLI\_sendData 함수를 통해 phymac\_dataReq 함수가 호출되고, arqPdu, pduSize를 dest\_ID가 2인 dest노드로 전송한다. 변수 txType에 arqPdu[ARQMSG\_OFFSET\_TYPE]의 값인 0이 저장된다. 그리고 src 노드의 puTTY 화면에 “[MAIN] sending to 2 (seq:-1)”이 출력된다(값이 0인 seqNum에서 1을 빼고 1024인 ARQMSSG\_MAX\_SEQNUM으로 모듈러 연산을 해주기 때문에 값이 -1). 그리고 변수 main\_state에 MAINSTATE\_TX인 1을 저장하여 src 노드의 state를 TX로 바꿔준다. 또한 다음 단어를 받을 수 있도록, flag\_needPrint 변수를 1로 변경한다. 변수 wordLen의 값을 0으로 변경한다. 마지막으로 arqEvent\_clearEventFlag 함수를 통해 arqEvent\_dataToSend flag를 clear한다. flag\_needPrint 변수의 값이 1이므로 else if(flage\_needPrint == 1)문에 걸리게 되어, puTTY 화면에 “Give a word to send : ”가 출력되어 입력을 기다린다. src 노드의 동작은 계속 while(1)문, 즉 무한루프 안에서 동작하는데, 현재 변수 prev\_state에 저장되어 있는 값은 0이고, 변수 main\_state에 저장 되어있는 값은 1로 값이 서로 다르다. 따라서 if(prev\_state != main\_state)문에 걸리게 되고, debug\_if(DBGMSG\_ARQ, "[ARQ] State transition from %i to %i\n", prev\_state, main\_state);의 경우 첫번째 parameter인 DBGMSG\_ARQ가 참, 즉 1인 경우에만 두번째 parameter가 출력된다. 하지만 DBGMSG\_ARQ가 0으로 정의 되어있으므로, 출력은 이루어지지 않는다. 따라서 다음 명령인 prev\_state에 현재 변수 main\_state에 저장되어있는 값인 1을 저장한다. 동작이 정상적으로 이루어짐을 가정했으므로, 메시지 전송이 완료되면 arqEvent\_dataTxDone flag가 setting 된다. 현재 src 노드의 변수 main\_state의 값은 MAINSTATE\_TX이므로 switch문에서 case MAINSTATE\_TX: 에 걸리게 되고, arqEvent\_dataTxDone가 setting 되었으므로 if (arqEvent\_checkEventFlag(arqEvent\_dataTxDone)) 조건문에 걸리게 되어 변수 main\_state의 값을 0인 MAINSTATE\_IDLE로 바꾸어 src 노드의 state를 IDLE 상태로 바꾸어준다. 또한 여기서도 무한 루프를 돌면서 if(prev\_state != main\_state)문에 걸리게 되고 prev\_state의 값을 main\_state의 값인 0으로 갱신한다.



메시지가 src 노드에서 dest 노드로 전송되었으므로, dest 노드의 arqEvent\_dataRcvd flag의 값이 setting 되고 현재 dest 노드의 state는 IDLE 상태이므로 switch문에서 case MAINSTATE\_IDLE: 에 걸리게 된다. 또한 arqEvent\_dataRcvd flag의 값이 setting 되었기 때문에 if (arqEvent\_checkEventFlag(arqEvent\_dataRcvd)) 조건문에 걸리게 된다. 이때 src 노드의 ID를 반환해주는 arqLLI\_getSrcId(); 함수를 통해 변수 srcId에 src 노드의 ID인 1이 저장되고, 도착한 uint8\_t 배열 형태의 rcvdData(src 노드에서의 arqPdu)의 포인터 주소 값을 반환하는 arqLLI\_getRcvdDataPtr(); 함수를 통해 포인터 변수 dataPtr에 저장한다. 그리고 rcvdData 배열의 크기를 반환해주는 arqLLI\_getSize(); 함수를 통해 변수 size에 arqPdu의 크기인 8을 저장한다. 따라서 dest 노드의 puTTY 화면에

" -------------------------------------------------

RCVD from 1 : hello (length:8, seq:0)

-------------------------------------------------" 가 출력된다. 이때 seq에 해당하는 값은 도착한pdu(dataPtr) 배열에서 ARQMSG\_OFFSET\_SEQ를 이용해 sequence number를 반환해주는 함수 arqMsg\_getSeq(dataPtr)를 통해 0이 출력되고, 메시지는 도착한 pdu(dataPtr)에서 ARQMSG\_OFFSET\_DATA를 이용해 메시지에 해당하는 부분의 포인터를 반환해주는 함수 arqMsg\_getWord(dataPtr)를 통해 “hello”가 출력된다. 그리고 dest 노드의 변수 main\_state 값을 0인 MAINSTATE\_IDLE로 변경하여 state를 IDLE상태로 바꾸고, 문자를 입력 받기 위해 변수 flag\_needPrint의 값을 1로 변경한다. 마지막으로 arqEvent\_clearEventFlag(arqEvent\_dataRcvd); 함수를 통해 arqEvent\_dataRcvd flag의 값을 clear한다. 마지막으로 dest 노드의 flag\_needPrint의 값이 1이므로 else if(flage\_needPrint == 1)문에 걸리게 되어, puTTY 화면에 “Give a word to send : ”가 출력되어 입력을 기다린다.

위와 같이 ID가 1인 src 노드에서 ID가 2인 dest 노드로 “hello” 메시지를 전송했을 때, 코드의 동작은 위와 같다.