임베디드 5주차 과제 보고서

학번 : 12161532

이름 : 김난영

<코드분석>

TaskStartCreateTasks() 에서 semaphore와 event flag를 각각 초기화 한 다음, 4개의 Random Task와 1개의 Decision Task를 만든다. 그러면 RandomTask() 가 총 4번, DecisionTask()가 총 1번 호출된다.

랜덤 숫자를 생성할 때와 같이, send\_array[]에 랜덤 숫자를 넣을때도 세마포어를 획득한 뒤에 진행함으로써 syncronization을 수행한다.

OSFlagPost()에서는 s\_grp라는 event flag에 비트 신호를 주는데, 이때 비트신호는 0x01<<taskNumber이며 옵션이 OS\_FLAG\_SET 이므로 0을 1로 SET 시킨다. 이렇게 보낸 신호는 DecisionTask()의 OSFlagPend()에서 받는다.

여기선 플래그 즉, 0x0f이 옵션(OS\_FLAG\_WAIT\_SET\_ALL + OS\_FLAG\_CONSUME)에 의해 플래그의 모든 비트가 SET 되고, 조건이 만족될 때에 플래그의 비트들을 CLEAR 시키게 된다.

또한 DecisionTask() 에서는 4개의 태스크 중에서 가장 작은 숫자를 보내온 태스크를 선택하여 해당 배열 변수에 ‘W’ 문자를 저장하고, 나머지 태스크들에게 ‘L’ 이란 문자를 저장한다. 이때, 최소 값을 찾기 위해 전에 배운 알고리즘 코드를 이용한다. (Ready List, Mapping Table 이용)

그 후에, 최소값을 보낸 taskNumber이 몇번인지 찾아서 receive\_array에 ‘W’를 입력하고, 최소값이 아닌 값을 보낸 task의 receive\_array에는 ‘L’을 입력한다.

마지막에는 OSFlagPost()를 통해 플래그 정보를 RandomTask의 OSFlagPend()로 보낸다. 이때, 옵션을 통해, 위와 같이 4개의 결과를 다 받을 때 까지 기다리고 조건이 만족되면 플래그의 비트들을 clear 시킨다.

Receive\_array에 저장된 character에 따라 해당 색깔을 화면에 출력한다.

<스크린샷>

