

# 임베디드시스템설계 설계과제 보고서

20181536 엄석훈

## 1. 기본적인 변수 설명

font8x8: 8x8 숫자 폰트이 저장된 배열

font7x7: 7x7 숫자 폰트이 저장된 배열

displayFont: 화면에 표시할 내용에 대한 정보가 저장된 배열

fontQueue: 버튼 클릭시 설정되어야 하는 폰트를 넘겨주는 큐

rowSemPtr: rowShiftTask 가 반복문을 1 번 동작하기 위해 필요한 세마포어

rowTimer: 1ms 마다 row 와 해당 row 의 col 을 업데이트 하기 위한 타이머

colTimer: 1000/18ms 마다 colIdx 를 업데이트 하기 위한 타이머

BtnTimer: 버튼 디바운싱을 위한 타이머

student\_id: 학번이 저장된 문자열

displayMode: 현재 모드에 대한 정보

(0: 개별 숫자 모드, 1: scroll 모드, 2: scroll(speedup 모드)

colIdx: displayFont 에서 현재 출력하는 col 에 대한 정보

timestamp: colTimer 에서 모드에 따라 동작 여부를 결정하는 변수

flagBtn: 디바운싱을 위한 변수로 값이 0 인 경우에만 버튼 클릭에 대한 동작 수행

## 2. 콜백 함수 및 태스크 설명

rowTimerCallback: 1ms 마다 dotmatrix 의 다음 row 로 이동하라고 세마포어를 give 하는 콜백 함수

colTimerCallback: 1000/18ms 마다 현재 모드를 보고 displayMode 가 0 이라면 timestamp 를 통해 콜백 함수가 18 번 실행될 때 한번 colIdx 를 8 증가시키고, displayMode 가 1 이라면 2 번에 한번 colIdx 를 1 증가시키고, displayMode 가 2 라면 매번 colIdx 를 1 증가시키는 콜백 함수

BtnTimerCallback: 버튼 디바운싱을 위한 콜백함수로 버튼이 눌리고 200ms 이후

flagBtn 을 다시 0 으로 리셋하는 콜백 함수

fontInitTask: 큐에서 들어온 포인터에서 font 를 읽어들이 student\_id 의 내용으로 dot matrix 에 표시할 내용을 displayFont 에 업데이트 하는 태스크

rowShiftTask: 세마포어를 take 해서 dot matrix 에서 다음 row 만 0 을 출력하고 나머지 row 는 1 을 출력하도록 하고 displayFont 에서 해당 row 에 저장되어 있는 값을 읽어 dot matrix 의 col 에 해당 값으로 업데이트 하는 태스크

### 3. 기본 동작 시나리오

먼저 보드가 동작하기 위해 기본적인 세팅을 실행한다. 그 중에서 MX\_GPIO\_Init 함수에서 프로그램에서 필요한 USER\_BTN 에서 인터럽트를 받을 수 있도록 세팅하고 dot matrix 의 row 핀과 연결된 PD0-PD7 과 col 핀과 연결된 PE8-PE15 까지를 출력핀으로 사용할 수 있도록 세팅해준다.

다음으로 row 동작을 제어할 rowTimer 와 col 동작을 제어할 colTimer, 디바운싱을 위한 BtnTimer 를 각각 생성해준다. 그리고 폰트 전송을 위한 fontQueue 와 row 이동 동기화를 위한 rowSemPtr 을 생성한다.

마지막으로 fontInitTask, rowShiftTask 를 생성하고 타이머를 시작하고 fontDisplay 에 기본값으로 학번 데이터와 8x8 폰트를 기반으로 초기화한다. 그리고 스케줄러를 시작해 본격적인 동작을 시작한다.

매 1ms 마다 rowTimer 에 의해 rowTimerCallback 함수가 실행되어 rowSemPtr 세마포어를 기브한다. 그러면 rowShiftTask 가 해당 세마포어를 테이크해서 while 문을 한번 돌면서 row 를 1 칸 이동하고 displayFont 에서 해당 row 와 colIdx 를 기반으로 출력을 업데이트 한다. 그리고 다시 세마포어를 기다리면서 블럭 상태가 된다.

그리고 1000/18ms 마다 colTimer 에 의해 colTimerCallback 함수가 실행되어 displayMode 에 따라서 colIdx 를 업데이트 한다. 이때 3 가지 모드에서 하나의 타이머를 사용하기 위해서 가장 빠르게 업데이트하는 displayMode 2 를 기준으로 1000/18ms 마다 타이머가 반복되도록 하였고 나머지 모드는 timestamp 를 사용하여 각각 1 초 1000/9ms 마다 colIdx 가 업데이트 되도록 하였다.

그리고 중간에 유저버튼이 눌리면 인터럽트가 발생하고 HAL\_GPIO\_EXTI\_Callback 함수가 실행된다. 다만 이때 디바운싱을 위해서 flagBtn 이

0 인 경우에만 동작을 하게 된다. 먼저 flagBtn 을 1 로 세팅하고 displayMode 를 하나 이동시키고 colIdx 를 초기화 하고 큐를 통해서 사용할 폰트를 send 하게 된다. 큐에 폰트 정보가 send 되면 fontInitTask 태스크에서 해당 폰트 주소를 받아서 displayFont 에 student\_id 문자열을 기반으로 업데이트 하게 된다. 그리고 마지막으로 200ms 이후 BtnTimerCallback 이 실행되도록 타이머를 시작한다. 그리고 다시 다음 큐가 들어올 때 까지 block 되어 기다리게 된다.