

시스템 프로그래밍 실습 팀 과제 보고서

Team싹 (3조)

201420894 김성규

201720723 박수린

201720743 정소희

1. 과제 개요

☆ 2개의 RBP 통신

이번 과제는 두 개의 라즈베리 파이에서 GPIO 14번, 15번핀을 이용한 유선 통신을 위한 디바이스 드라이버를 제작하는 것이었다. 먼저 과제 요구 사항에 나와 있는 대로 두 개의 라즈베리 파이의 GPIO 14번과 GPIO 15번핀이 교차되게 연결하였다. 디바이스 드라이버에서는 open, release, write, read 함수를 구현하였다.

write 함수에서는 GPFSEL1 레지스터를 통해 GPIO 14번핀을 output(tx)모드로 설정하였다. 사용자가 입력한 string을 copy_from_user를 사용하여 kernel level로 내려 buffer에 저장하고 각각의 character에 대해 소문자 a를 0(5bits 2진수 0000), z를 25(5bits 2진수 11001)로 변환하여 정수형으로 계산하여 주었다. 시작 bit에 1, 마지막 bit에 0을 추가하여 하여 각각의 character를 총 7bits로 변환하여 전송하였다. 중간의 데이터는 bit연산을 통해 1이면 GPIO 14번핀의 output을 GPSET0 레지스터를 통해 set 하고, 0이면 GPCLR0 레지스터를 통해 output을 clear 시켜 주어 데이터를 전송하였다. 각각의 데이터 전송 후 5msec 딜레이를 주었다. 입력 받은 문자열의 송신이 끝나면 마지막에 '{'(5bits 2진수11010)을 전송하여, 수신 측에서 이를 통해 입력이 끝났음을 확인하도록 했다.

read 함수에서는 GPIO 15번핀의 초기값이 0으로, Input모드이기 때문에 따로 설정해 주지 않았다. GPLEV0 레지스터를 사용하여 current_value, pre_value 변수와의 값 비교를 통해 값이 변하는지 확인하였고, 중간에 5msec 딜레이를 주었다. 1이 처음 수신되면 이후의 5bits 읽은 후 마지막 7번째 bit로 0이 수신됨을 확인하고, 중간 데이터 5bits를 일련의 과정을 통해 character로 변환하여 buffer에 저

장해 주었다. 시작 bit가 1이 아니거나 마지막 bit가 0이 아니면 반복문을 종료해 더 이상 읽지 않도록 하였다. 또한 마지막 글자가 (1)11010(0)이 왔을 때 '{'가 수신된 경우이므로 송신이 끝났다고 판단하고 buffer에 값을 추가하지 않고 반복문을 종료한다. 마지막 글자에 Null value를 넣어준 뒤 copy_to_user를 통해 buffer에 있는 string을 user level로 전송한다.

※ 과제 수행 전제 조건: 두개의 라즈베리 파이에서 송수신을 하기 위해서는 송신 측에서 처음에 아무 문자열이나 수신 측에 보내 수신 측 GPIO 15번 핀의 GPLEV를 0으로 초기화 시켜줘야 한다.

2. 팀원별 역할 분담

- 1) 김성규 : 알고리즘 구축, 디바이스 드라이버 제작(read 함수 구현), app.c 구현
- 2) 박수린 : 알고리즘 구축, 디바이스 드라이버 제작(write 함수 구현), app.c 구현
- 3) 정소희 : 알고리즘 구축, 디바이스 드라이버 제작(기타 함수 구현), app.c 구현

3. 고찰

처음에는 UART 레지스터를 이용하여 하드웨어 통신을 계획했지만 과제 개요에 맞지 않다는 것을 깨닫고, 소프트웨어 통신으로 방향을 바꾸었다. GPLEV0 레지스터를 사용하여 과제를 해결하면서 한가지 의문점이 생겼다. GPIO 핀에 전류가 흐르면 HIGH가 되어 해당 GPIO 핀의 GPLEV가 1이 될 것이라고 생각하였는데, 송신 측에서 GPCLR을 사용하여 GPIO 핀의 output을 clear하자 수신 측의 GPLEV가 0이 되었다. 아직 GPIO 레지스터에 대한 이해도가 미흡한 것 같다. 그래서 이번 과제에 한가지 해결하지 못한 부분이 존재한다. 수신 측에서 문자열을 받기 전에 송신 측에서 아무런 문자열은 보내 수신 측의 GPLEV를 0으로 만들어 줘야 한다는 것이다. 이 과정을 거치지 않으면 수신 측의 GPLEV0가 1로 설정되어 있어서 알고리즘이 제대로 작동하지 않는다. 이 부분은 해결하지 못했지만 가산점이 부여되는 3개의 핀만을 사용하고, 하나의 어플리케이션과 하나의 디바이스 드라이버에 송/수신 기능을 구현하였다.