

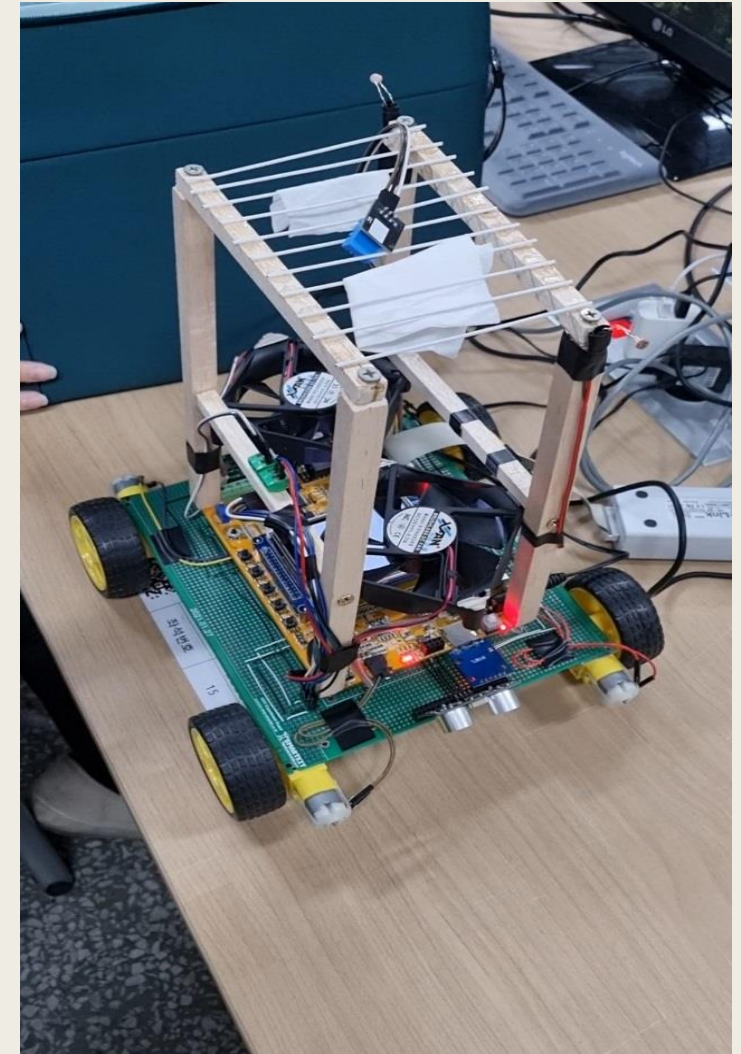
임베디드 시스템 설계 및 실험

4조 TERM PROJECT

Smart Drying Rack – 스마트 빨래 건조대

개발 목적

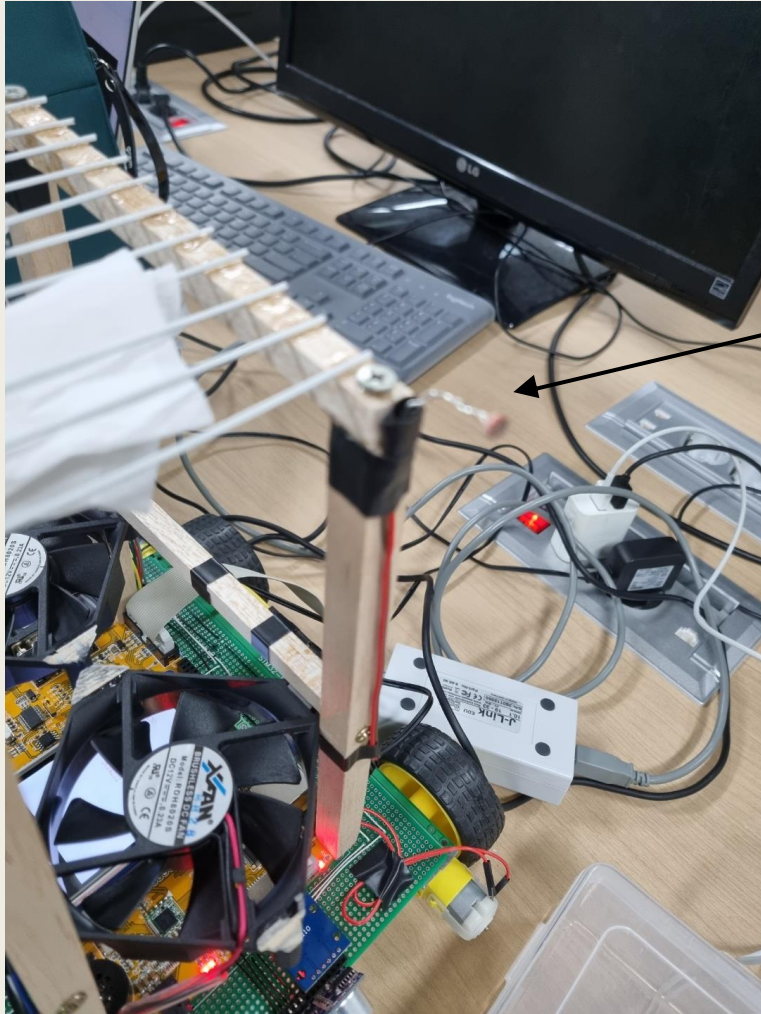
- 햇빛이 좋은 장소에 빨래를 널어도 시간이 지나면 그늘이 저 빨래가 잘 마르지 않게 됨
- 만약, 건조대가 빛 강도를 측정해서 빛이 강한 곳으로 이동한다면 빨래를 더 빠르고 효율적으로 말릴 수 있음



기능

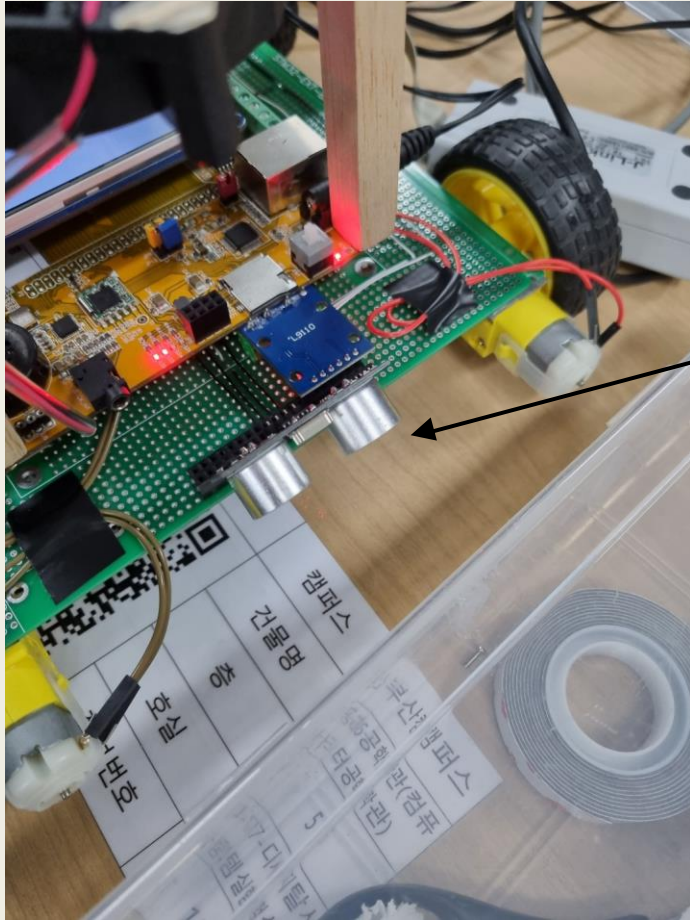
- 조도센서를 이용한 빛 강도 측정
- 초음파 거리 센서를 이용해 이동 방향에 방해물이 있는지 확인
- 온 습도 센서를 이용해 빨래 건조 여부 확인
- 모터 드라이버를 이용해 바퀴 및 팬 가동
- 블루투스를 이용해 빨래가 다 말랐다는 메시지 전송

세부 기능 - 빛 감지



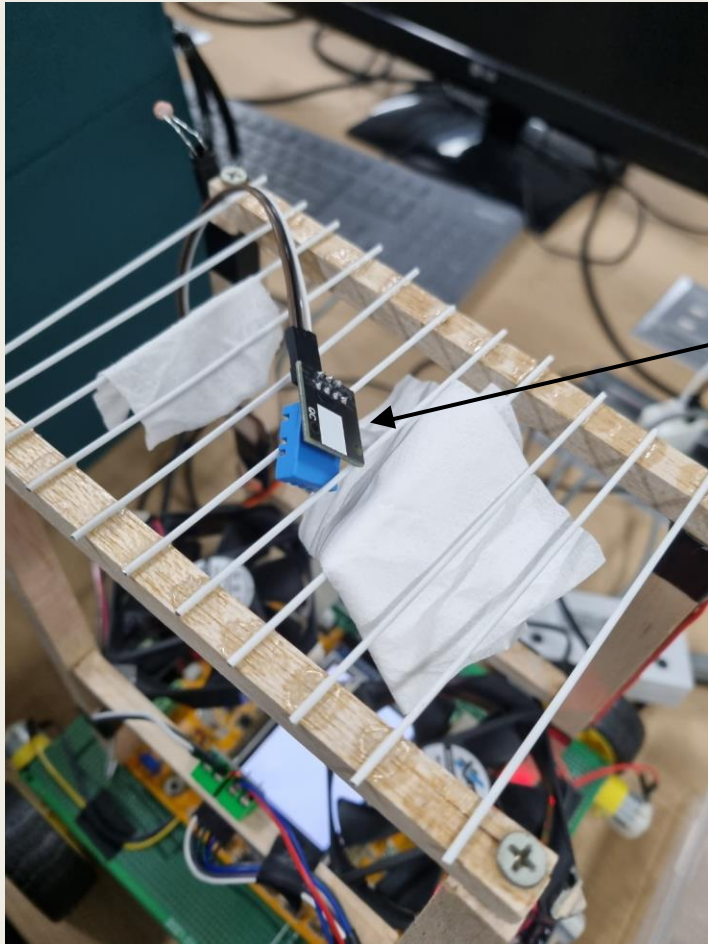
- 건조대 양단에 부착된 조도센서로 주변 환경의 빛 강도 측정
- 2개의 측정 값으로 이동 방향 제어 - 빛이 더 강한 방향으로 이동
- Direct Memory Access를 통해 Analog - Digital convert 방식으로 연결

세부 기능 - 충돌 방지



- 양단에 연결된 초음파 센서를 이용해 앞 물체와 거리 측정
- 조도 센서 값을 따라 움직이고 있더라도, 초음파 센서가 방해물을 감지하면 움직이지 않음
- TIM1,3을 통해 연결

세부기능 - 빨래 건조 여부 확인/메시지 전송

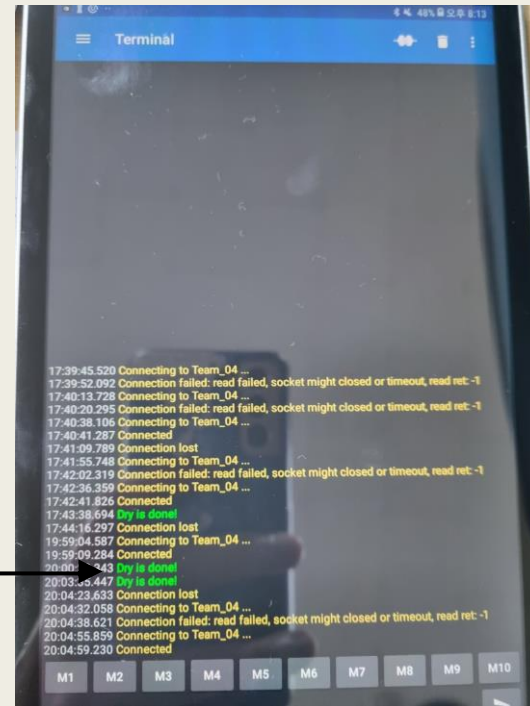


- 건조대 위에 부착된 온 습도 센서로 빨래 습도 측정
- 센서로 측정한 습도 값이 설정된 값보다 낮으면 블루투스로 빨래가 다 말랐다는 메시지 전송
- TIM2를 통해 연결되고, 주기적으로 interrupt를 일으켜 습도를 측정하는 방식

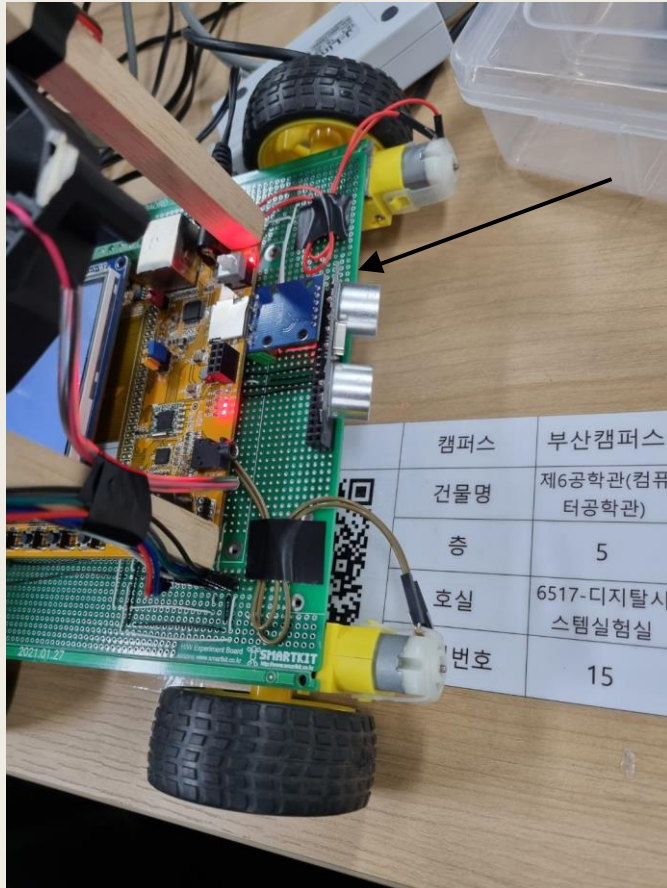
```
EXTI_GenerateSWInterrupt(EXTI_Line0);  
  
void EXTI0_IRQHandler(void) {  
    if (EXTI_GetITStatus(EXTI_Line0) != RESET) {  
        GPIO_SetBits(GPIOD,GPIO_Pin_2);  
        if(count > 5000000){  
            res=DHT11_read(&dev);  
            if(dev.humidity<45)  
                dry_flag = 1;  
            else  
                dry_flag = 0;  
        }  
        EXTI_ClearITPendingBit(EXTI_Line0);  
    }  
}
```

```
if(dry_flag==1 && previous_flag ==0){  
    for(int i=0; i<14; i++){  
        sendDataUART2(msg_dry[i]);  
        delay();  
    }  
}
```

“Dry is done”메시지를
전송한 화면



세부기능 - 바퀴 및 팬 가동

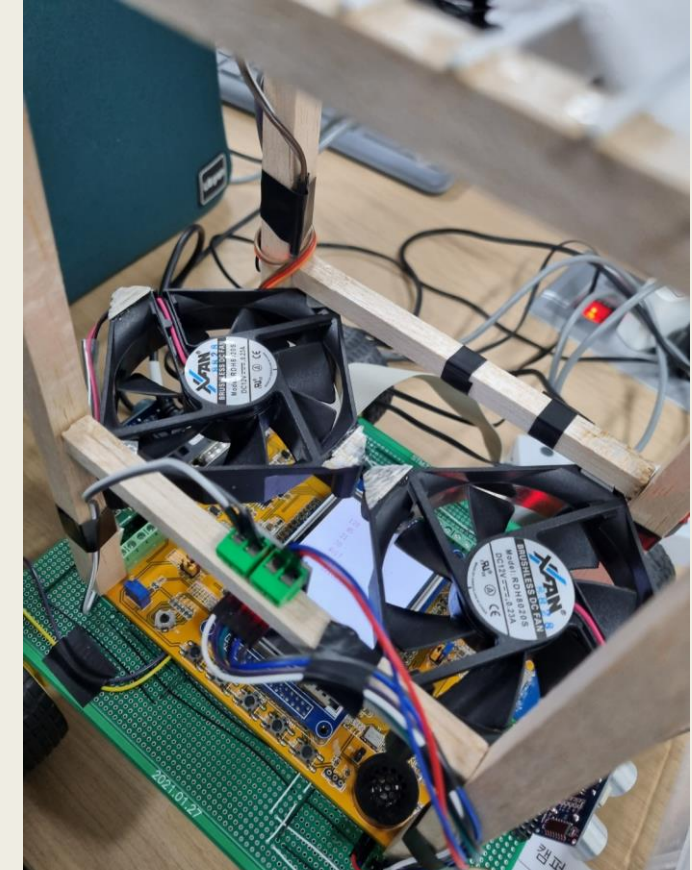


❖ 바퀴

- TIM4를 통해 연결한 모터 드라이버로 바퀴 2개씩 연결, CH3,4 사용
- 조도 센서와 초음파 센서 값을 관찰해 방향 결정

❖ 팬

- TIM4의 CH1,2를 이용해 가동
- 사용자가 임의로 켜고 끌 수 있음
- User Button1로 켜고 끄기 가능 - interrupt 방식



```
//fan on-off
void EXTI15_10_IRQHandler(void) { // when the user S1 is pressed
    if (EXTI_GetITStatus(EXTI_Line11) != RESET) {
        if (GPIO_ReadInputDataBit(GPIOD, GPIO_Pin_11) == Bit_RESET) {
            if(fan_flag==0)
            {
                Fan_on();
                fan_flag=1;
            }
            else
            {
                Fan_off();
                fan_flag=0;
            }

            GPIO_SetBits(GPIOD, GPIO_Pin_3);
        }
        EXTI_ClearITPendingBit(EXTI_Line11);
    }
}
```