

## 8. 전용 라이브러리 사용하기

### 1) 개요

아두이노로 프로젝트를 진행할 때 어려운 점들 중 하나는 장치를 하나씩 제어할 때와는 달리 여러 개의 장치들을 사용하고자 할 때 동일한 하드웨어 자원을 사용하는 서로 다른 장치들이 충돌을 일으켜 문제가 발생하는 경우이다. 이러한 경우 두 장치 중 하나를 포기하여 하는 상황이 발생할 수 있고 프로젝트 수행에 큰 차질을 가져오게 된다.

그러나 스마트 인벤터 보드 전용 라이브러리가 지원하는 인스턴스를 사용하게 되면 스마트 인벤터 보드가 지원 가능한 모든 장치를 동시에 사용한다 하더라도 장치간의 충돌 없이 본 제어를 사용할 수 있다. 따라서 스마트 인벤터 보드로 여러 개의 장치들을 함께 제어할 때에는 전용 라이브러리를 사용하는 것이 좋다.

### 2) 사용방법

아두이노 스케치의 메뉴 상단에서 스케치 -> include library 에서 SmartInventor 라이브러리를 추가하도록 한다. 주의해야 할 것은 전용 라이브러리를 사용할 경우 전용 라이브러리에서 지원하는 동일한 기능을 지닌 함수나 라이브러리를 함께 사용해서는 안된다는 것이다. 예를 들어 tone함수는 전용 라이브러리에서 Buzz 인스턴스로 대체되었는데 tone을 여전히 사용한다면 문제가 생길 수 있다. 또한 같은 이유에서 앞에서 언급한 TVRemocon과 DCM 라이브러리, RFRemocon라이브러리를 본 라이브러리와 함께 사용해서는 안된다.

### 3)인스턴스 설명

`void DCMotorUse()`

DC모터를 사용할 수 있도록 초기화 하는 함수이다. DCM라이브러리를 사용할 때처럼 핀을 지정하는 파라미터가 없다는 것에 유의하자.

`void DCMotorExtension()`

내장된 2개의 DC모터 드라이버 이외에 추가로 2개의 모터를 더 사용하고자 할 때 추가 DC모터들을 초기화하는 함수이다. 이 함수를 사용하기 위해서는 외부 DCM 모터 드라이버를 구입하여 PORTD의 28~31번에 체결한 다음 사용하여야 한다. 이렇게 하면 총 4개 DC모터를 각각 제어할 수 있게 된다.

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> DCMotorFour

`void DCMove(byte direction, byte speed)`

2륜 구동 로봇일 때에는 M1(M3)과 M2(M4)를 동시에 제어해야 하는데 이러한 경우 편리하게 사용할 수 있는 인스턴스이다.

direction: FORWARD / BACKWARD / LEFT / RIGHT / STOP / LOOSE의 파라미터가 입력될 수 있으며 입력된 값에 따라 이륜 구동 로봇의 방향이 제어된다.

speed: 모터의 속도로서 0 ~ 100까지의 값이 사용된다.

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> DCMove

void DCMotor(char selection, char direction, char speed)

모터를 1개씩 제어할 때 사용한다. selection에는 M1, M2, M3, M4이 입력될 수 있으며 M3와 M4는 추가 DC모터 드라이버를 장착했을 때에만 제어 가능하다.

Direction : CW/CCW/STOP/LOOSE

speed: 0 ~ 100

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> DCMotor

void Buzz(int frequency, int duration);

frequency에는 음계(피치)가, duration에는 음 길이를 입력한다. 내장형이므로 버저에 연결된 핀 번호는 이미 정해져 있으므로 따로 파라미터로 입력할 필요가 없으며 pitches.h를 아두이노 폴더에 추가할 필요 없이 음계 이름을 frequency에 사용할 수 있다.

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> Buzz

void LedArray(unsigned char num);

제어기 상단의 8개 LED를 제어할 수 있는 인스턴스이다. 파라미터 값은 0부터 255까지 LED의 점멸을 2진수로 보고 입력하면 된다. 예를 들어 4를 입력하면 왼쪽에서 3번째 LED만 켜지고 16을 입력하면 왼쪽부터 4개의 LED가 켜지는 식이다.

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> LedArray

void RFRemoconUse(long speed)

RF리모컨의 신호를 받을 수 있도록 초기화 한다. 스마트 인벤터 보드의 ZIG소켓에 블루투스 모듈이나 지그비 모듈을 연결한 후에 사용할 수 있다. 로보티즈사의 RC-100 리모컨 패킷 형식으로 데이터를 송수신 하므로 동일한 패킷을 전송하는 리모컨이라면 모두 사용할 수 있다. speed에 입력할 값은 무선 시리얼 통신 전송 속도(bps)로서 4800, 9600, 19200, 38400, 57600의 속도가 사용될 수 있다.

RFData

RFData 인스턴스는 사용자의 리모컨 신호가 RF리모컨에서 온 것인지 아니면 IR리모컨에서 온 것인지 판단한다. RF리모컨인 경우에는 0보다 큰 값을 갖고 그렇지 않은 경우엔 0의 값을 갖는다

int RFRemconData()

수신된 리모컨 값을 반환한다. 리모컨의 버튼 마다 고유의 값을 갖는데 어떤 키를 누르느냐에 따라서 수신된 값은 달라진다. 버튼을 눌렀을 때 제어기로 송신되는 RF리모컨의 값은 아래와 같다.

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> RFRemcon



void TVRemoconUse()

IR리모컨을 사용할 수 있도록 초기화 한다. 5장에서 살펴본 IR리모컨 수신 함수 기능과 동일하다.

void TVRemoconData()

IR리모컨으로부터 수신 키 값을 반환한다. 이 인스턴스를 이용하여 어떤 버튼이 눌러졌는지 알 수 있다.

\* 참조 예제 : 파일 -> 예제 -> SmartInventor -> TVRemocon

#### 4)프로그래밍

##### 예제1: 스마트 1단계 밀어내기 로봇 프로그래밍하기

밀어내기 로봇은 스마트 인벤터 보드의 아랫면의 7개의 IR센서로 낭떠러지를 감지하며 자율 주행하는 로봇이다. 전용 라이브러리에서 DC모터 구동, 버저를 사용하였다.

\*예제 위치: 파일 -> 스케치북 -> Rokit -> smartInventor -> model -> smart1 -> sumoBot

##### 예제2: 스마트 1단계 축구 로봇 프로그래밍하기

축구로봇은 IR리모컨이나 RF리모컨으로 조종을 하면서 즐길 수 있는 로봇이다. RF리모컨과 TV리모컨 둘 중에 어떤 것을 사용해도 조종이 가능하다. 전용 라이브러리에서 DC모터 구동, 버저, RF리모컨, IR리모컨 그리고 내장 LED 제어 기능을 함께 사용하였다.

\*예제 위치: 파일 -> 스케치북 -> Rokit -> smartInventor -> model -> smart1 -> SoccerBot