

승인 된 버전. 승인 된 날짜 : 2020 년 1 월 24 일 09:51

# 사용자 공간에서 GPIO를 제어하는 방법

## 내용

- 1 목적
- libgpiod를 통한 2 개의 GPIO 제어
- 자신의 응용 프로그램을 통한 3 GPIO 제어
  - 3.1 목적
  - 3.2 코드
  - 3.3 빌드 애플리케이션
- 4 참고

## 1 목적

이 기사는 사용자 공간 에서 GPIO 를 제어하는 두 가지 방법을 보여줍니다 .

- libgpiod 사용
- 신청서 작성

## libgpiod를 통한 2 개의 GPIO 제어

libgpiod는 Linux GPIO 문자 장치 (gpiod는 GPIO 장치를 나타냄) 와 상호 작용하기위한 C 라이브러리 및 도구를 제공합니다 . 자세한 설명은 libgpiod 저장소 <sup>[1]</sup> 을 참조하십시오 .

- **gpiodetect**
  - 시스템에 존재하는 모든 gpiochip 목록
  - 용법:

```
보드 $> gpiodetect
gpiochip11 [GPIOZ] (16 줄)

...
gpiochip0 [GPIOA] (16 줄)
```

- **gpioinfo**

- 지정된 **gpiochip**의 모든 라인, 이름, 소비자 및 설정을 나열합니다.
- 용법:

```
보드 $> gpioinfo
gpiochip11-16 줄 :
라인 0 : 명명되지 않은 사용되지 않은 입력 액티브-하이
1 행 : 명명되지 않은 사용되지 않은 입력 액티브-하이
...
```

또는

#### 코멘트

```
보드 $> gpioinfo gpiochip0- > gpiochip0 라인 만 인쇄
```

- **gpioget**
  - 지정된 GPIO 라인 의 값을 읽습니다 (라인이 이미 요청 된 경우 유효하지 않음)

#### 코멘트

```
보드 $> gpioget gpiochip0 5- > GPIOA5의 값을 얻는 것
0- > 회선이 낮게 구동됨을 의미
```

- **gpioset**
  - 지정된 GPIO 라인 의 값을 설정하여 라인을 계속 내보내고 시간 종료, 사용자 입력 또는 신호까지 기다립니다.

```
보드 $> gpioset gpiochip3 8 = 1- > GPIO8을 높게 설정
```

## 자신의 응용 프로그램을 통한 3 GPIO 제어

### 3.1 목적

이 응용 프로그램은 GPIO \_A\_14 ( GPIO 뱅크 A, 14 행)를 토글 합니다. 예 [STM32MP15\\_Evaluation\\_boards](#) 또는 [STM32MP15\\_Discovery\\_kits](#) GPIO \_A\_14 녹색에 연결되어 LED .

이 애플리케이션은 커널과 동일한 툴체인으로 크로스 컴파일되어야 합니다.

## 3.2 코드

```
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys / ioctl.h>
#include <unistd.h>

#include <linux / gpio.h>

int main ( int argc , char ** argv )
{
    struct gpiohandle_request 요청 ;
    구조체 gpiohandle_data 데이터 ;
    char chrdev_name [ 20 ];
    int fd , ret ;

    strcpy(chrdev_name, "/dev/gpiochip0");
```

## 3.3 빌드 응용 프로그램

이 애플리케이션을 빌드하려면 [Linux\\_user\\_space\\_applications 추가](#) 를 참조하십시오 .

## 4 참고 자료

1. [libgpiod 저장소](#)