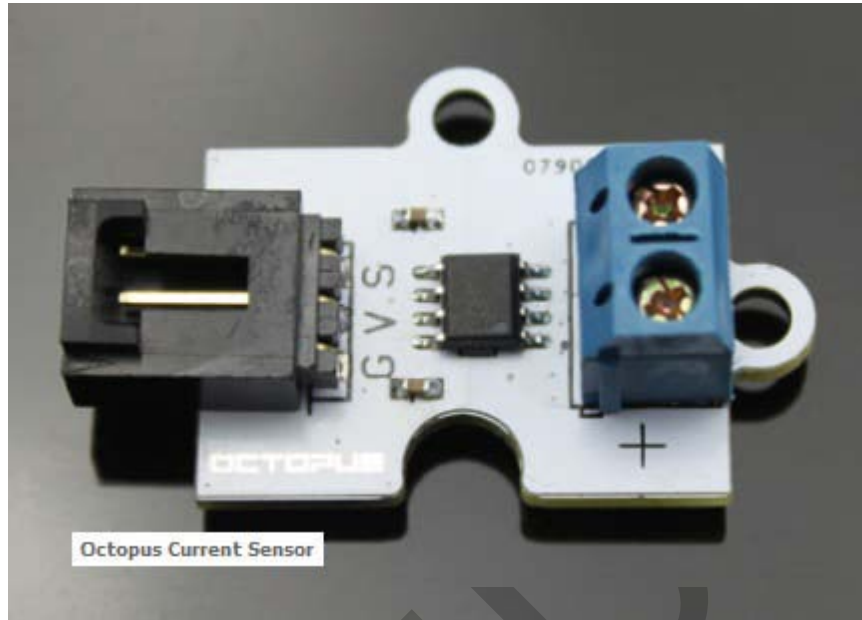


## ■ Current Sensor



### 1. 사양

#### 1.1. Description

Current Sensor는 current의 크기를 detecting 하기 위해 사용하며, ACS712 current sensor chip을 사용하는 sensor module이다. ACS712ELC-30A chip을 사용하며, 아날로그 출력(Sensitivity)은  $66\text{mA} / \text{V}$  이다

#### 1.2. Feature

Use Chip: ACS712ELC-30A

Measuring range:  $\pm 30\text{A}$

Analog output:  $66\text{mV} / \text{A}$

When current is not detected, the output voltage is  $VCC / 2$

Power supply needs: 5V

Interface type: Analog

Pin Definition: 1-Signal 2-VCC 3-GND

Fast response and High sensitivity

Simple drive circuit

## Current Sensor

### 1.3. 구성

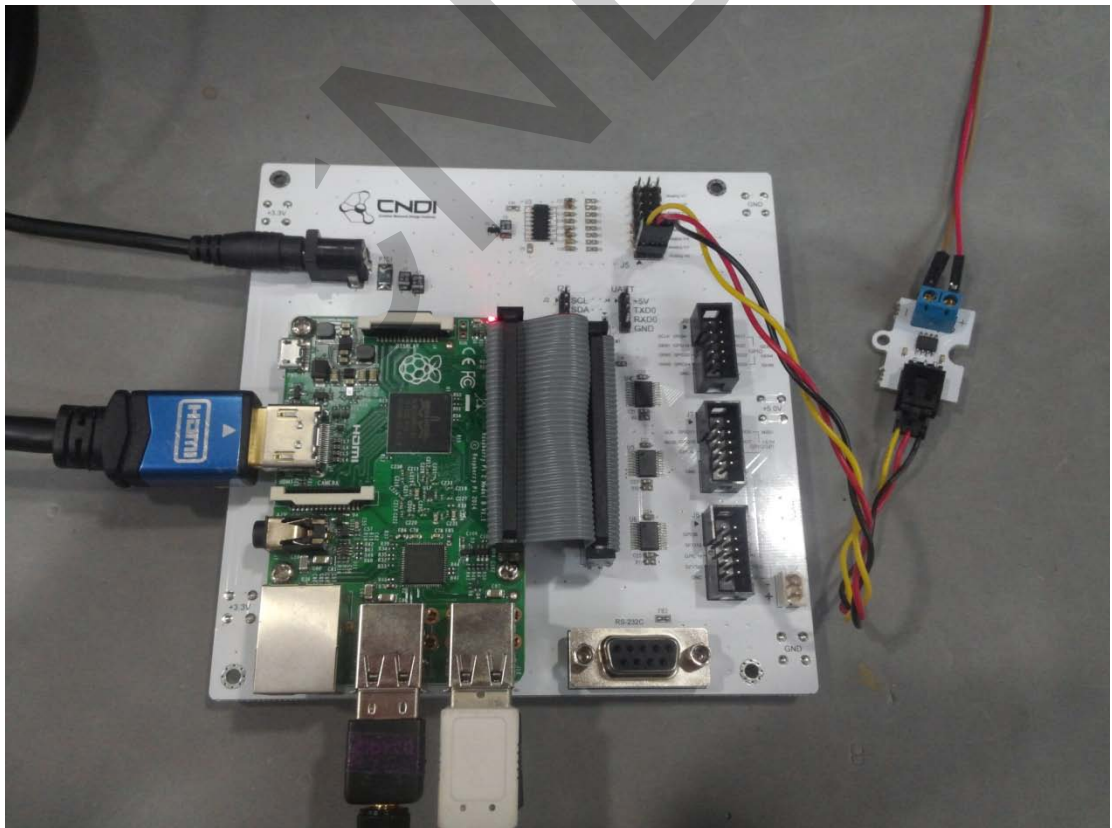
- 1 x Current Sensor
- 1 x Analog Sensor Cable

## 2. User Guide

### 2.1. 결선

모듈의 결선은 아래의 그림과 표를 이용하여 결선한다.

Raspberrypi	PIR Motion
Gnd	GND(검정)
5V	VCC(빨강)
Analong In 0	SIGNAL(노랑)



모듈의 장치 및 회로에 대한 상세한 내용은 데이터시트 및 회로도를 참고한다.

## 2.2. 예제프로그램

아래 예제 프로그램을 작성 후 컴파일 한 후 실행하면 Current 센서에 입력 값을 변환하여 전류 값을 디버그 메시지로 표시하는 동작을 한다.

컴파일 명령 : **gcc -o CurrentSensor CurrentSensor.c -l wiringPi**

실행 명령 : **./CurrentSensor**

- CurrentSensor.c

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringPiSPI.h>

#define CS_MCP3208      8           //GPIO 8
#define SPI_CHANNEL     0           //SPI Channel
#define SPI_SPEED       1000000    //spi speed

int ReadMcp3208ADC(unsigned char adcChannel)
{
    unsigned char buff[3];
    int nAdcValue = 0;

    buff[0] = 0x06 | ((adcChannel & 0x07) >> 2);
    buff[1] = ((adcChannel & 0x07) << 6);
    buff[2] = 0x00;

    digitalWrite(CS_MCP3208, 0);           //low cs Active

    wiringPiSPIDataRW(SPI_CHANNEL, buff, 3);
```

## Current Sensor

```
//8bit data
buff[1] = 0x0F & buff[1];
nAdcValue = (buff[1]<<8) | buff[2];

//spi chip Select command
digitalWrite(CS_MCP3208, 1);

return nAdcValue;
}

int main(void)
{
    int nAdcChannel = 0;
    int nAdcValue = 0;

    if(wiringPiSetupGpio() == -1)
    {
        fprintf(stdout,"Not start wiringPi: %s\n",strerror(errno));
        return 1;
    }

    //wiringpi init
    if(wiringPiSPISetup(SPI_CHANNEL, SPI_SPEED) == -1)
    {
        fprintf(stdout, "wiringPiSPISetup Failed: %s\n", strerror(errno));
        return 1;
    }

    pinMode(CS_MCP3208, OUTPUT);

    while(1)
    {
        nAdcValue = ReadMcp3208ADC(nAdcChannel);
        //sensorRead
```

```
        nAdcValue = ((nAdcValue-506)*5/1024/0.066);           //506 is
ADC Value of 0A
        printf("Current Sensor Value = %u\n", nAdcValue);
        delay(100);
    }

    return 0;
}
```