

---

## 스마트 미러 설치 과정

### 1.라즈비안 패키지 시스템을 최신으로 업데이트 합니다.

```
sudo apt-get update
```

### 2.#MagicMirror2 설치

node.js #electron 으로 만들어진 MagicMirror2 프로그램을 설치합니다. MagicMirror2 는 #스마트미러 구현을 위한 기본 프레임워크 이며, 필요한 기능들을 모듈형태로 추가할 수 있는 구조 입니다.

스마트미러의 근간이 되는 프로그램 MagicMirror2 는 Node.js 기반의 Electron 으로 구현되어 있습니다. 라즈비안 운영체제에서 동작시키기 위해 Node.js 환경을 구성해줘야 합니다. node 명령어는 이미 설치가 되어 있어서, node 의 패키지 관리 시스템인 npm 을 최신버전으로 설치 합니다.

```
sudo apt-get install npm
sudo npm install -g npm@latest
```

오픈소스 MagicMirror2 에서 제공되는 스크립트로 설치 합니다. (최대 5분 소요)

```
bash -c "$(curl -sL
https://raw.githubusercontent.com/MichMich/MagicMirror/master/installers/raspberry
.sh)"
```

### 3.#MagicMirror2 에 핫워드와 구글어시스턴트 모듈 설치

node.js electron 으로 만들어진 MagicMirror2 프로그램을 설치하였고, 이번에는 MagicMirror2 에 음성인식과 말하기 기능을 넣기 위해 #핫워드 모듈과 구글어시스턴트 모듈을 설치합니다.

이전 단계에서 스마트미러의 근간이 되는 프로그램 MagicMirror2 설치완료 했습니다. 이번 단계에서는 음성인식 말하기 기능을 위한 추가 모듈을 설치 합니다. 우선, 모듈 설치전에 필요한 종속 프로그램과 라이브러리를 미리 설치 합니다.

```
sudo apt-get install libmagic-dev libatlas-base-dev sox libsox-fmt-all mpg321
libasound2-dev
```

modules 디렉토리 이동후, 핫워드 MMM-Hotword 설치를 수행합니다. 터미널 창을 열고, 다음의 명령어를 차근차근히 입력하시면 됩니다.

4. 기존 MMM-Hotword는 Submodule 들을 하나하나 직접 설치하고 빌드해야 했지만, 2.0.0버전 부터는 설치 스크립트에 통합되었습니다!

```
cd ~/MagicMirror/modules/  
git clone https://github.com/eouia/MMM-Hotword.git  
cd MMM-Hotword  
chmod +x ./installer/install.sh  
./installer/install.sh
```

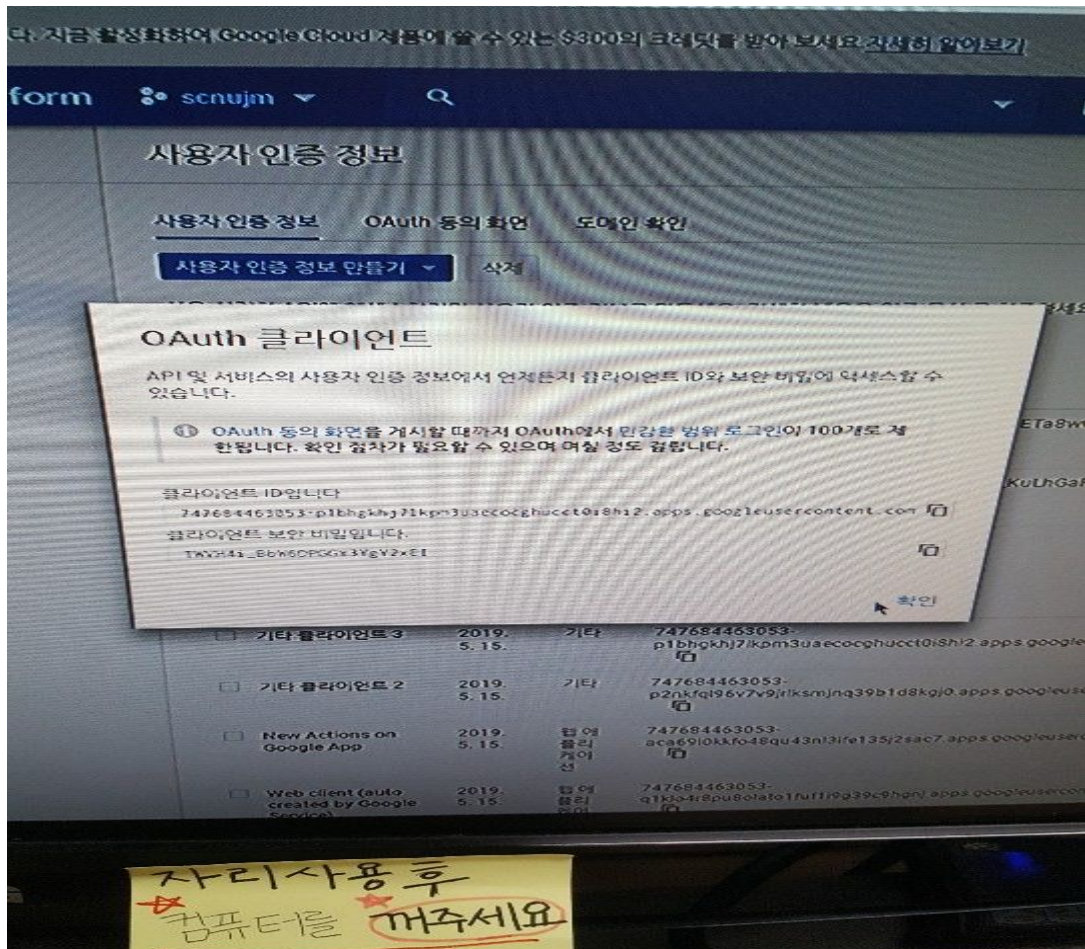
기존 튜토리얼에서 가장 실패가 많았던 Hotword 가 설치스크립트 하나로 금방 설치 되었습니다. 다들 잘 되셨는지요? 그럼 다시 modules 디렉토리 이동후, [구글어시스턴트 MMM-AssistantMk2](#) 설치를 수행합니다. (20~30분 소요)

5.#MagicMirror2 에 핫워드와 구글어시스턴트 모듈 설치

```
cd ~/MagicMirror/modules/  
git clone https://github.com/eouia/MMM-AssistantMk2  
cd MMM-AssistantMk2  
npm install  
cd scripts  
chmod +x *.sh  
cd ..  
npm install --save-dev electron-rebuild  
./node_modules/.bin/electron-rebuild
```

## 6.#구글어시스턴트 설정

구글어시스턴트 사용을 위해 백엔드 서버의 설정을 합니다.



구글 어시스턴트 모듈 설정

다음 단계는 설치된 **구글 어시스턴트** 를 동작시키기 위해서 **구글 백엔드** 서버의 설정을 해보도록 하겠습니다. 이작업을 하기 이전에 아래 접속되는 **구글클라우드플랫폼**에 사전에 회원가입이 되어 있어야 합니다.

**구글액션콘솔** Google Action Console 에 접속하여 새로운 프로젝트를 생성합니다. 검색창에서 **Google Assistant API** 를 검색하여 활성화(Enable)를 실행해줍니다. 그러면 생성된 프로젝트에서 **구글어시스턴트** 기능을 사용가능한 상태로 됩니다.

## 7.#구글어시스턴트 모듈 설정값 수정

스마트미러의 근간이 되는 프로그램 MagicMirror2 설치완료 하고, 음성인식 말하기 기능을 위한 추가 모듈을 설치하고 구글클라우드플랫폼 서비스에 연결을 했습니다. 설치된 모듈을 구성하기 위해서 설정파일인 config/config.js 에 내용을 추가해야합니다.

아래 추가될 코드를 넣었습니다. 추가되는 내용은 [하위드 구글어시스턴트](#) 동작에 대한 내용입니다

---

### Config.js 파일 소스코드

```
var config = {
  address: "localhost", // Address to listen on, can be:
    // - "localhost", "127.0.0.1", "::1" to listen on loopback interface
    // - another specific IPv4/6 to listen on a specific interface
    // - "", "0.0.0.0", "::" to listen on any interface
    // Default, when address config is left out, is "localhost"
  port: 8080,
  ipWhitelist: ["127.0.0.1", "::ffff:127.0.0.1", ":::1"], // Set [] to allow all IP addresses
    // or add a specific IPv4 of 192.168.1.5 :
    // ["127.0.0.1", "::ffff:127.0.0.1", ":::1",
    "::ffff:192.168.1.5"],
    // or IPv4 range of 192.168.3.0 --> 192.168.3.15
  use CIDR format :
    // ["127.0.0.1", "::ffff:127.0.0.1", ":::1",
    "::ffff:192.168.3.0/28"],

  //Magic Mirror에 나타나는 언어를 한국어로 변경
  language: "ko",
  //Magic Mirror에 나타나는 시간을 12시에서 24시로 변경
  timeFormat: 24,
  units: "metric",

  //Magic Mirror에 사용되는 모듈
  modules: [
    //경보에 사용하는 모듈
    {
      module: "alert",
    },
    //업데이트 표시를 나타내는 모듈
    {
      module: "updatenotification",
      position: "top_bar"
    }
  ]
}
```

```

},
//시간을 나타내는 시계 모듈
{
    module: "clock",
    position: "top_left"
},
//달력을 나타내는 캘린더 모듈 -> 구글 캘린더 url 파일 사용
{

```

```

    module: "calendar",
    header: "대한민국 공휴일",
    position: "top_left",
    config: {
        calendars: [
            {
                symbol: "calendar-check",
                url:
"https://calendar.google.com/calendar/ical/ko.south\_korea%23holiday%40group.v.calendar.google.com/public/basic.ics"
            }
        ]
    }
},

```

//매직 미리 하단에 나타나는 문구 모듈

```

{
    module: "compliments",
    position: "lower_third",
    config: {
        compliments: {
            anytime: [ //아무때나
"2019 소프트웨어 전시회"
            ],
            morning: [ //아침
                "좋은 아침!",
                "힘찬 아침!",
                "오신걸 환영합니다!"
            ],
            afternoon: [ //오후
                "안녕하세요!",
                "멋져요!",
                "오신걸 환영합니다!"
            ],
            evening: [ //저녁
                "와우! 잘 지냈나요?",
                "멋져 보이네요!",
                "반가워요!"
            ],
            day_sunny: [//맑은 낮

```

```
],
day_cloudy: [//흐린 낮

],
cloudy: [//흐림

],
cloudy_windy: [//흐리고 바람

],
showers: [//소나기

],
rain: [//비

],
thunderstorm: [//천둥번개

],
snow: [//눈

],
fog: [//안개

],
night_clear: [//맑은 밤

],
night_cloudy: [//흐린 밤

],
night_showers: [//소나기 밤

],
night_rain: [//비오는 밤

],
night_thunderstorm: [//천둥번개 밤

],
night_snow: [//눈오는 밤

],
night_alt_cloudy_windy: [//흐리고 바람부는 밤
```

```

    ],
  },
},
//날씨를 나타내는 모듈 -> Openweather의 api를 승인받아서 사용
{

```

```

  module: "currentweather",
  position: "top_right",
  config: {
    location: "Gwangju, KR",
    locationID: "", //ID from

```

```

    http://bulk.openweathermap.org/sample/city.list.json.gz; unzip the gz file and find your city
    appid: "6050bc695807684cf9594b35887a5464"
  }
},
{

```

```

  module: "weatherforecast",
  position: "top_right",
  header: "Weather Forecast",
  config: {
    location: "Gwangju, KR",
    locationID: "", //ID from

```

```

    http://bulk.openweathermap.org/sample/city.list.json.gz; unzip the gz file and find your city
    appid: "6050bc695807684cf9594b35887a5464"
  }
},

```

```

// 하단에 위치하는 뉴스피드를 뉴욕 타임즈 에서 jtbc 뉴스로 변경 -> jtbc rss 추출
{

```

```

  module: "newsfeed",
  position: "bottom_bar",
  config: {
    feeds: [
      {
        title: "jtbc",
        url:

```

```

        "http://fs.jtbc.joins.com/RSS/newsflash.xml"
      }
    ],
    showSourceTitle: true,
    showPublishDate: true,
    broadcastNewsFeeds: true,
    broadcastNewsUpdates: true
  }
},

```

```

//MagicMirror 에 음성인식과 말하기 기능을 사용하는 모듈
{

```

```

module: "MMM-Hotword",
position: "top_right",
config: {
    chimeOnFinish: null,
    mic: {
        recordProgram: "arecord",
        device: "plughw:1"
    },
    models: [
        {
            hotwords : "smart_mirror",
            file      : "smart_mirror.umdl",
            sensitivity : "0.5",
        },
    ],
    commands: {
        "smart_mirror": {
            notificationExec: {
                notification: "ASSISTANT_ACTIVATE",
                payload: (detected, afterRecord) => {
                    return {profile:"default"}
                }
            },
            restart:false,
            afterRecordLimit:0
        }
    }
},
},

{
    chimeOnFinish: null,
    mic: {
        recordProgram: "arecord",
        device: "plughw:1"
    },
    // 인식률이 떨어진 호출어를 computer로 변경하고 민감도 수정
    models: [
        {
            hotwords : "computer",
            file      : "computer.umdl",
            sensitivity : "0.5",
        },
    ],
    commands: {
        "smart_mirror": {
            notificationExec: {

```



```

        notification: "ASSISTANT_ACTIVATE",
        payload: (detected, afterRecord) => {
            return {profile:"default"}
        }
    },
    restart:false,
    afterRecordLimit:0
}
}
}
},
//Magic Mirror의 근간이 되는 구글 어시스턴트 모듈
{
    module: "MMM-AssistantMk2",
    position: "top_right",
    // 현재 위치를 구글맵스로 통해 위도 경도 수정
    config: {
        deviceLocation: {
            coordinates: {
                latitude: 34.969548, // -90.0 - +90.0
                longitude: 127.478610, // -180.0 - +180.0
            },
        },
        // 목소리를 입력받는 디바이스의 플러그를 설정
        record: {
            recordProgram : "arecord",
            device      : "plughw:1",
        },
        //Hotword가 정상 작동을 못할시 밀의 말을 호출
        notifications: {
            ASSISTANT_ACTIVATED:
                "HOTWORD_PAUSE",
            ASSISTANT_DEACTIVATED:
                "HOTWORD_RESUME",
        },
        // Hotword가 정상 작동시 밀의 말을 호출
        useWelcomeMessage: "Hellow, jiman",
        profiles: {
            "default" : {
                lang: "ko-KR"
            }
        },
    },
}
],
};

```

```
/****** DO NOT EDIT THE LINE BELOW *****/  
if (typeof module !== "undefined") {module.exports = config;}
```