



실리카(Xilica) Lua 스크립트 왕초보 가이드

이 문서는 프로그래밍을 전혀 모르는 분들도 Xilica Designer 안에서 Lua 스크립트가 어떻게 작동하는지 이해하고, 간단한 기능을 직접 만들 수 있도록 돕기 위해 작성되었습니다.

1. 🤔 Lua 스크립트, 왜 쓰나요?

Xilica Designer에 있는 기본 블록(AND, OR, Delay 등)만으로는 해결하기 어려운 **복잡한 문제**를 해결하기 위해서입니다.

- "버튼을 3번 누르면 전광판을 켜고, 10초 뒤에 다시 꺼줘"
- "볼륨이 -10dB보다 커지면 경고등을 켜줘"

이런 "조건"과 "순서"가 있는 일을 ****글(Code)****로 적어서 시키는 것이 바로 Lua 스크립트입니다.

2. ⚙️ 핵심 원리: 요리사 모델 (Chef Model)

Lua 스크립트를 **"요리사"**라고 생각하면 아주 쉽습니다.

1. 재료 받기 (Inputs, `in_t`):

- 외부(터치패널 버튼, 센서)에서 들어오는 신호입니다.
- 요리사는 냉장고(`in_t` 테이블)를 열어 재료가 왔는지 확인합니다.

2. 요리 하기 (Logic):

- 조리법(스크립트 코드)에 따라 지지고 볶습니다.
- 예: "재료가 1(ON)이면 요리를 시작해라."

3. 서빙 하기 (Outputs, `out_t`):

- 완성된 요리를 접시(`out_t` 테이블)에 담아 밖(LED, 릴레이, 네트워크)으로 내보냅니다.

3. 📝 기본 문법 (이것만 알면 80% 가능)

실리카 스크립트는 항상 똑같은 모양(템플릿)을 가집니다.

◆ 1단계: 냉장고와 접시 준비 (기본 세팅)

```
-- 들어오는 신호(재료)를 담는 바구니
local in_t = ...
if type(in_t) ~= "table" then in_t = {} end

-- 나가는 신호(요리)를 담을 빈 접시
local out_t = {}
```

이 부분은 그냥 복사해서 맨 위에 붙여넣으세요! (마법의 주문입니다)

◆ 2단계: 재료 확인 (Input 읽기)

Xilica 모듈의 핀 번호를 대괄호 [] 안에 넣어서 확인합니다.

```
-- 1번 핀에 연결된 버튼 상태를 확인
local button_state = in_t[1]

-- 만약 버튼이 눌렸다면 (값이 1 또는 true 라면)
if button_state == 1 or button_state == true then
    -- 할 일을 적습니다
end
```

◆ 3단계: 요리 서빙 (Output 쓰기)

결과를 내보낼 핀 번호를 [] 안에 적고 값을 넣습니다.

```
-- 2번 핀(LED)을 켭니다
out_t[2] = 1

-- 3번 핀(네트워크 문자열)에 글자를 보냅니다
out_t[3] = "Hello World"
```

◆ 4단계: 마무리 (Return)

마지막에 접시를 내보냅니다.

```
return out_t
```

4. 🚀 실전 예제 (따라해 보세요)

예제 1: 단순 스위치 (버튼 누르면 LED 켜기)

가장 기초적인 "토스" 기능입니다. 1번으로 들어온 걸 2번으로 내보냅니다.

```
local in_t = ...
local out_t = {}

-- 1번 핀(버튼)의 값을 읽어서
local val = in_t[1]

-- 2번 핀(LED)에 그대로 넣는다
out_t[2] = val

return out_t
```

예제 2: 토글 스위치 (한 번 누르면 켜지고, 또 누르면 꺼짐)

기억력(메모리)이 필요합니다. 전역 변수(G_)를 사용합니다.

```
local in_t = ...
local out_t = {}
```

```

-- G_SwitchState가 없으면 0(꺼짐)으로 초기화 (기억력 생성)
if G_SwitchState == nil then G_SwitchState = 0 end

-- 1번 입력을 읽음
local btn = in_t[1]

-- 버튼이 눌렸을 때만 (떨 때는 무시)
if btn == 1 then
    -- 현재 상태를 뒤집음 (0이면 1로, 1이면 0으로)
    if G_SwitchState == 0 then
        G_SwitchState = 1
    else
        G_SwitchState = 0
    end
end

-- 최종 상태를 2번 핀으로 내보냄
out_t[2] = G_SwitchState

return out_t

```

5. 💡 꿀팁 (Tip)

1. **G_** 로 시작하는 변수:

- 일반 변수는 스크립트가 한 번 돌 때마다 사라지지만, **G_** 를 붙이면(**Global**) Xilica가 전원이 꺼질 때까지 **값** 을 기억합니다. 상태 저장(토글, 카운트)에 필수입니다.

2. 주석 (--):

- 코드 앞에 -- 를 쓰면 컴퓨터는 무시하고 사람만 읽습니다. 설명을 많이 적어주세요!

3. 안전장치 (SafeNum):

- 가끔 입력값이 비어있을(nil) 때가 있습니다. 이때 더하기 빼기를 하면 에러가 납니다.
- `if val == nil then val = 0 end` 처럼 항상 값이 있는지 확인하는 습관을 들이세요.

이 가이드가 실리카 Lua의 세계로 들어오시는 데 도움이 되기를 바랍니다! 궁금한 점은 언제든지 물어봐주세요. 😊