

## Arduino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

int touchPin=7;

LiquidCrystal_I2C lcd(0x27, 16, 2);
char image = '0';
char bef_image = '0';
char sum='0';
void setup()
{
    Serial.begin(9600);
    pinMode(touchPin, INPUT);
    lcd.init();
    lcd.backlight();
}

void loop(){

    if(Serial.available() > 0)
    {
        image = Serial.read();
        if (bef_image != image){
            switch(image)
            {
                case 'A':
                    lcd.print("A");
                    break;

                case 'B':
                    lcd.print("B");
                    break;

                case 'C':
                    lcd.print("C");
```

```

        break;

        case 'D':
            lcd.print("D");
            break;

        case 'E':
            lcd.print("E");
            break;

        case 'F':
            lcd.print("F");
            break;

        case 'G':
            lcd.print("G");
            break;

        default:

            break;
    }
    bef_image = image;
}

if(digitalRead(touchPin)==HIGH)
{
    lcd.clear();

}

delay(500);
}

```

## p5.js

```
/*
Modify modelURL and serialPort to your settings
*/
const modelURL = 'https://teachablemachine.withgoogle.com/models/bPM0UYeu0/'; // 모델
URL 설정
const serialPort = 'COM4'; // 시리얼 포트 설정

let classifier; // 이미지 분류기 변수
let serial; // 시리얼 포트 변수
let label = ''; // 분류된 라벨을 저장할 변수
let video; // 비디오 캡처를 위한 변수

function preload() {
  // 모델을 미리 로드하고 시리얼 포트를 초기화
  classifier = ml5.imageClassifier(modelURL + 'model.json'); // 이미지 분류기 초기화
  serial = new p5.SerialPort(); // 시리얼 포트 초기화
}

function setup() {
  // 설정 함수: 캔버스를 생성하고 비디오 캡처를 설정
  serial.open(serialPort); // 지정된 시리얼 포트 열기
  createCanvas(320, 240); // 캔버스 생성
  video = createCapture(VIDEO); // 비디오 캡처 설정
  video.hide(); // 비디오 요소 숨기기
  classifyVideo(); // 비디오 분류 시작
}

function classifyVideo() {
  // 비디오를 분류하는 함수
  classifier.classify(video, gotResult); // 비디오를 분류하고 결과를 gotResult 함수로 전달
}

function draw() {
  // 화면을 그리는 함수
  background(0); // 배경을 검정색으로 설정
  image(video, 0, 0); // 비디오를 캔버스에 표시
  fill(255); // 텍스트 색상을 흰색으로 설정
  textSize(32); // 텍스트 크기 설정
  textAlign(CENTER); // 텍스트 정렬을 중앙으로 설정
  text(label, width / 2, height - 4); // 분류된 라벨을 캔버스에 표시
}

function gotResult(error, results) {
  // 분류 결과를 처리하는 함수
  if (error) {
    console.error(error); // 오류가 발생하면 콘솔에 출력
  }
}
```

```
    return; // 오류 발생 시 함수 종료
}
label = results[0].label; // 첫 번째 결과의 라벨을 저장
console.log(label); // 라벨을 콘솔에 출력
serial.write(label); // 라벨을 시리얼 포트로 전송
setTimeout(classifyVideo, 2000);
}
```