
2025 Project Report

무엇보다 화려한 화분 Colorful Flower

Smart FlowerPot

스마트 파머스 팻

한상진, 김도연, 신종섭, 전길수

목차

01

프로젝트 개요

구현하고자 하는 목표&일정표
개발 언어 및 기술 스택&개발 툴&개발
보드(FPGA)

04

구현 코드

프로젝트를 구성하는
코드&기능

02

주요 기능

프로젝트를 구성하는 부품&기능

05

프로젝트 결과

완성된 프로젝트
결과물

03

플로우 차트

구현하고자 하는 프로젝트의 흐름

06

마무리

초기 프로젝트 계획
&문제점/개선사항&정리 및 향후 계획

프로젝트 개요

구현 목표



스마트 화분

주변 온/습도와 토양의 습도를 측정하여 수분의 필요 여부를 수시로 표정으로 나타냄.

구현 목표

물 수위 센서와 온/습도 센서를 이용해 토양의 수분을 체크하고 알맞은 물주기를 알 수 있게 해주는 스마트 화분을 구현할 예정.

프로젝트 개요

개발 언어 및 기술 스택&개발 툴&개발 보드(FPGA)

개발 환경



개발 언어



개발 툴&개발 보드(FPGA)



프로젝트 개요

일정표

8/26	8/27	8/28	8/29	8/30	8/31	9/1
일정 조율& 파트 분배						
	코드 작성& 하드웨어 제작					
			PPT 발표 자료 준비			
						발표

주요 기능

프로젝트를 구성하는 부품&기능

(R/G/B) LED

(R/G/B) 세기를 조절해
색을 표시하는 LED 모듈.



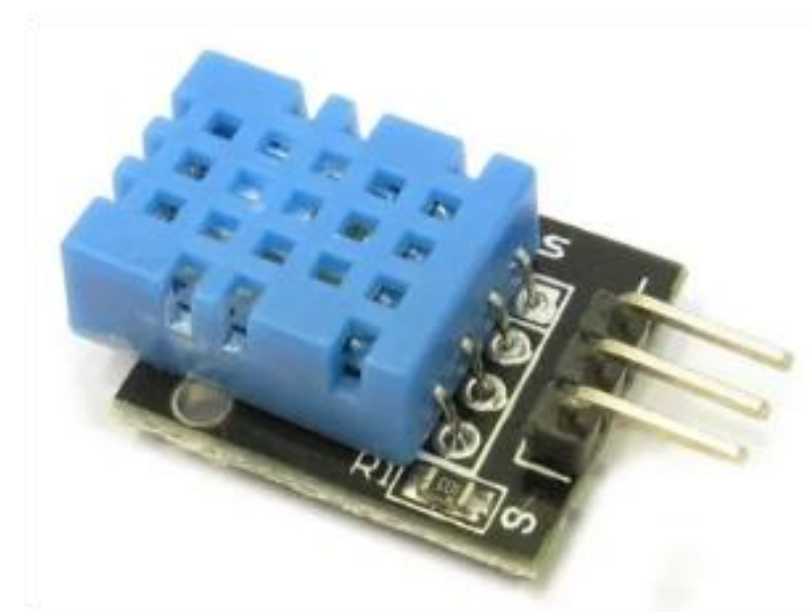
물수위감지센서

물의 수위에 따른 저항 변화를
이용 물의 수위를 감지하는 센서.



온/습도센서

온도와 습도를 측정해
디지털 신호로 출력하는 센서.



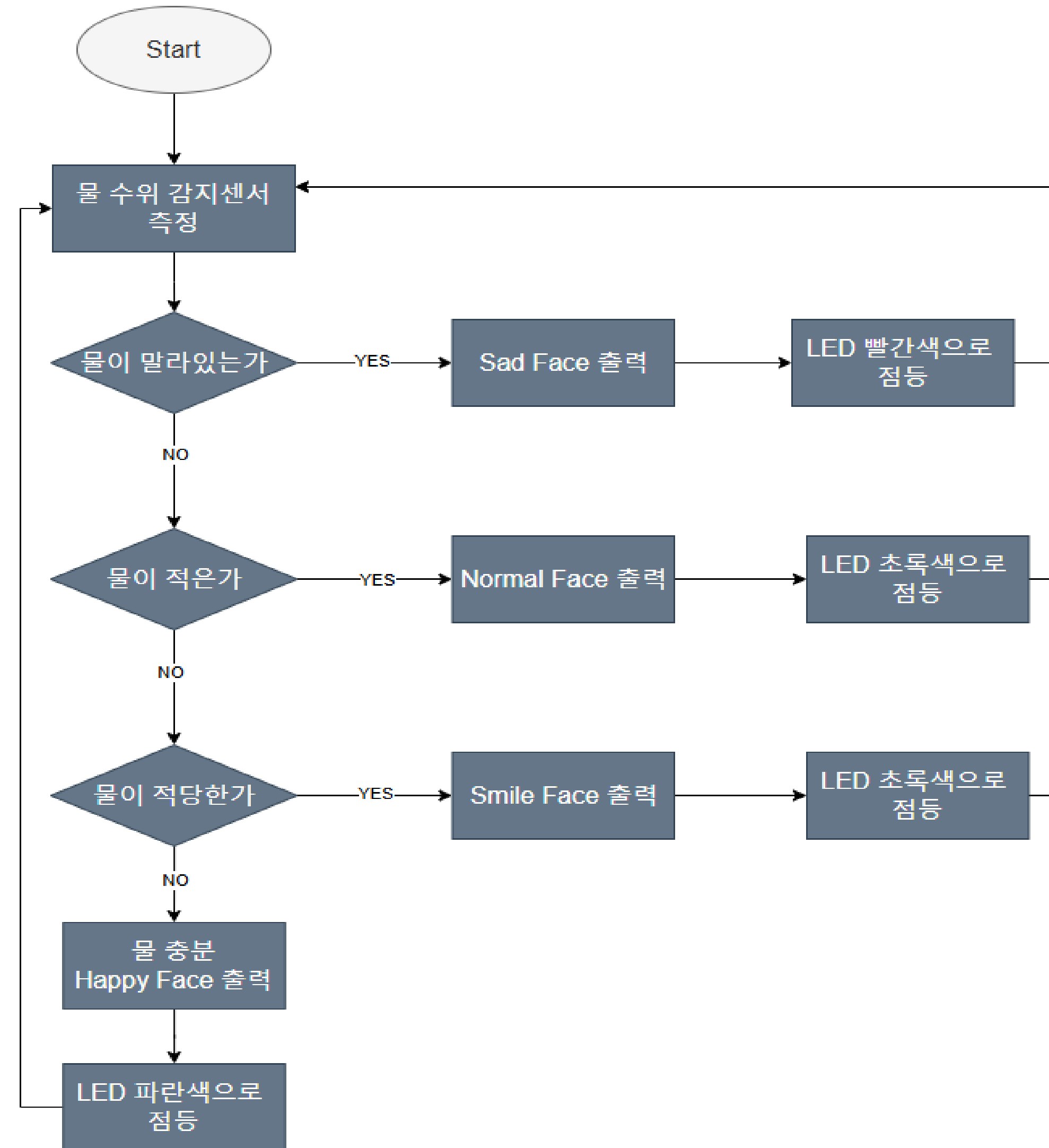
LCD

문자와 숫자 정보를
화면에 표시하는 LCD 모듈.



플로우 차트

구현하고자 하는 프로젝트의 흐름



(R/G/B) LED Module



RGB LED의 색상 제어

```
module sfp_led_rgb_cntr (
    input clk, reset_p,           // 클럭 신호와 비동기 리셋 신호
    input [2:0] color_sel,       // 색상 선택 신호 (3비트)
    output reg led_r,            // Red LED 출력
    output reg led_g,            // Green LED 출력
    output reg led_b             // Blue LED 출력
);
```

```
module sfp_led_rgb_cntr (
    input clk, reset_p,
    input [2:0] color_sel,
    output reg led_r,
    output reg led_g,
    output reg led_b
);

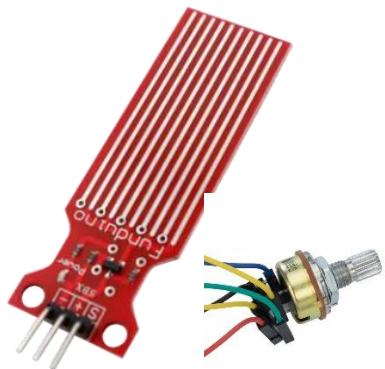
// 색상 선택 (예: 3비트, 0: Red, 1: Green, 2: Blue, 3: Yellow, 4: Cyan, 5: Magenta, 6: White)
always @(posedge clk, posedge reset_p) begin
    if (reset_p) begin
        led_r = 0; led_g = 0; led_b = 0;
    end else begin
        case (color_sel)
            3'b000: begin led_r = 1; led_g = 0; led_b = 0; end // Red
            3'b001: begin led_r = 0; led_g = 1; led_b = 0; end // Green
            3'b010: begin led_r = 0; led_g = 0; led_b = 1; end // Blue
            3'b011: begin led_r = 1; led_g = 1; led_b = 0; end // Yellow (Red + Green)
            3'b100: begin led_r = 0; led_g = 1; led_b = 1; end // Cyan (Green + Blue)
            3'b101: begin led_r = 1; led_g = 0; led_b = 1; end // Magenta (Red + Blue)
            3'b110: begin led_r = 1; led_g = 1; led_b = 1; end // White (Red + Green + Blue)
            default: begin led_r = 0; led_g = 0; led_b = 0; end // 모두 꺼짐
        endcase
    end
end

endmodule
```

코드 설명

입력 값 **color_sel** (색상 선택 신호)에 따라 RGB LED의 색상을 제어하는 모듈

* color_sel 선택 [3비트 (0: Red, 1: Green, 2: Blue, 3: Yellow, 4: Cyan, 5: Magenta, 6: White)]



Water Level Sensor Module

물수위감지센서를 사용하여 물 높이 측정

```
// 워터센서를 사용하여 물 높이 조절.
wire [4:0] channel_out;
wire eoc_out;
wire [15:0] do_out;

xadc_wiz_0 adc
(
    .daddr_in({2'b00, channel_out}), // Address bus for the dynamic reconfiguration port
    .dclk_in(clk), // Clock input for the dynamic reconfiguration port
    .den_in(eoc_out), // Enable Signal for the dynamic reconfiguration port
    .reset_in(reset_p), // Reset signal for the System Monitor control logic
    .vauxp6(vauxp6), // Auxiliary channel 6
    .vauxn6(vauxn6),
    .channel_out(channel_out), // Channel Selection Outputs
    .do_out(do_out), // Output data bus for dynamic reconfiguration port
    .eoc_out(eoc_out) // End of Conversion Signal
);

reg [11:0] adc_value;
wire eoc_pedge;
edge_detector_p echo_ed( .clk(clk), .reset_p(reset_p), .cp(eoc_out),.p_edge(eoc_pedge));

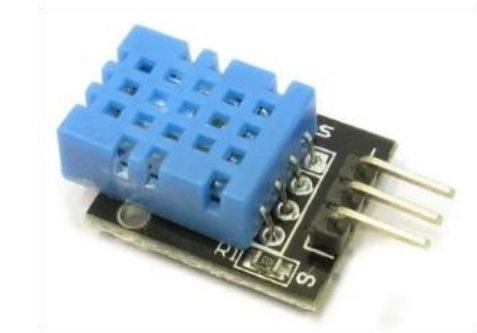
// 수위 센서 0 ~ 30 : step 1
//          31 ~ 40 : step 2
//          41 ~ 50 : step 3
//          51 ~ 55 : step 4

always @(posedge clk, posedge reset_p) begin
    if (reset_p) adc_value = 0;
    else if (eoc_pedge) begin
        adc_value = do_out[15:8];
    end
end
```

코드 설명

XADC로 아날로그 신호를 디지털(adc_value)로 변환하여 물 수위를 측정 하는 로직 추가

- * XADC 모듈로 물 높이를 측정해 디지털 값(adc_value)으로 변환
- * eoc_out 상승 엣지로 변환 완료를 검출해 adc_value 갱신
- * adc_value로 LED·텍스트·부저 등 수위 기반 동작 제어



dht11_cntr (온/습도 측정)

DHT11 센서 제어 모듈

```
module dht11_cntr(
    input clk, reset_p,
    inout dht11_data,      // 센서와의 양방향 데이터 통신 핀 (입출력 겸용)
    output reg [7:0] humidity, // 습도 데이터를 저장하는 8비트 레지스터 출력
    output reg [7:0] temperature, // 온도 데이터를 저장하는 8비트 레지스터 출력
    output [15:0] led       // LED 상태를 제어하는 16비트 출력 신호
);
```

```
// DHT11 센서를 제어하여 습도와 온도를 읽어오는 모듈
wire [7:0] humidity, temperature;
dht11_cntr dht11(
    .clk(clk), .reset_p(reset_p),
    .dht11_data(dht11_data),
    .humidity(humidity), .temperature(temperature));

// 8비트 와이어를 선언하여 습도와 온도를 BCD 형식으로 저장
wire [7:0] humi_bcd, tmpr_bcd;
bin_to_dec bcd_humi(.bin(humidity), .bcd(humi_bcd));
bin_to_dec bcd_tmpr(.bin(temperature), .bcd(tmpr_bcd));
// fnd_cntr 모듈을 인스턴스화하여 결합된 BCD 값을 FND 출력
fnd_cntr fnd(.clk(clk), .reset_p(reset_p),
    .fnd_value({humi_bcd, tmpr_bcd}),
    .hex_bcd(1),
    .seg_7(seg_7), .com(com));
```

코드 설명

DHT11 센서를 사용하여 습도와 온도를 측정하는 모듈

- * DHT11 센서에서 습도(humidity)와 온도(temperature) 데이터를 읽어옴.
- * dht11_data는 양방향 포트로, 센서와의 통신에 사용.
- * BCD 값을 결합하여({humi_bcd, tmpr_bcd}) FND 출력.



i2c_lcd_text_cntr

```
// text_cmd 명령 (1: Happy, 2: Smile, 3: Sad, 4: Normal, 5: Clear)
module sfa_i2c_lcd_text_cntr(
    input clk, reset_p,
    input [2:0] text_cmd,
    output scl, sda,
    output reg init_done      // 초기화 완료 신호
);

    ....

    i2c_lcd_send_byte send_byte(
        .clk(clk), .reset_p(reset_p),
        .addr(7'h27), .send_buffer(send_buffer),
        .send(send), .rs(rs),
        .scl(scl), .sda(sda), .busy(busy), .led()
    );

    ....
    ....

    // FSM 상태 정의
    localparam IDLE      = 7'b000_0001; // 0: 대기 상태
    localparam INIT      = 7'b000_0010; // 1: 초기화 상태
    localparam SEND_HAPPY = 7'b000_0100; // 2: Happy Face 전송 상태
    localparam CLEAR_DISPLAY= 7'b000_1000; // 3: 화면 지우기 상태
    localparam SEND_SMILE = 7'b001_0000; // 4: Smile Face 전송 상태
    localparam SEND_SAD   = 7'b010_0000; // 5: Sad Face 전송 상태
    localparam SEND_NORMAL = 7'b100_0000; // 6: Normal Face 전송 상태

    ....
    ....

    // FSM 로직
    always @(posedge clk or posedge reset_p) begin
        if (reset_p) begin
            ...
            ...
        end
        else begin

            case (state)
                IDLE: begin
                    ...
                    ...
            end
        end
    end
endmodule
```

코드 설명

I2C 를 통해 텍스트를 LCD 에 표시하는 모듈

- * Input (text_cmd)값에 따라 미리 정의된 문자열을 LCD 출력.
=> (1: Happy Face) (2:Smile Face) (3: Sad Face) (4: Normal Face) (5: LCD_CLEAR)
- * 수업시간에 실습한 "i2c_lcd_send_byte"을 활용하여 개발.

1) TOP 모듈 (smart_flower_pot_top)

```

module smart_flower_pot_top(
    input clk, reset_p,          // 클럭 신호와 리셋 신호 입력
    inout dht11_data,           // DHT11 센서 데이터 핀
    input vauxp6, vauxn6,        // XADC 입력 핀
    input [3:0] btn,             // 4개의 버튼 입력
    output [7:0] seg_7,          // 7-세그먼트 디스플레이 출력
    output [3:0] com,            // 7-세그먼트 공통 핀 출력
    output reg [15:0] led,       // 16개의 LED 출력
    output led_r, led_g, led_b,  // RGB LED 출력
    output buzz,                 // 부저 출력
    output scl, sda              // I2C 신호 (SCL, SDA)
);

```

2) 온/습도, 물 수위 센서 모듈 선언

```

// DHT11 센서를 제어하여 습도와 온도를 읽어오는 모듈
wire [7:0] humidity, temperature; // 습도와 온도 데이터
dht11_cntr dht11(
    .clk(clk), .reset_p(reset_p),
    .dht11_data(dht11_data),
    .humidity(humidity), .temperature(temperature)
);

// 워터 센서를 사용하여 물 높이 측정
wire [4:0] channel_out; // XADC 채널 출력
...
...
xadc_wiz_0 adc(
    ...
    ...
always @(posedge clk, posedge reset_p) begin
    if (reset_p) adc_value = 0; // 리셋 시 ADC 값 초기화
    else if (eoc_pedge) begin
        adc_value = do_out[15:8]; // 변환된 데이터 저장
    end
end
end

```

3) 물 수위 변화에 따른 상태 출력 구문

```

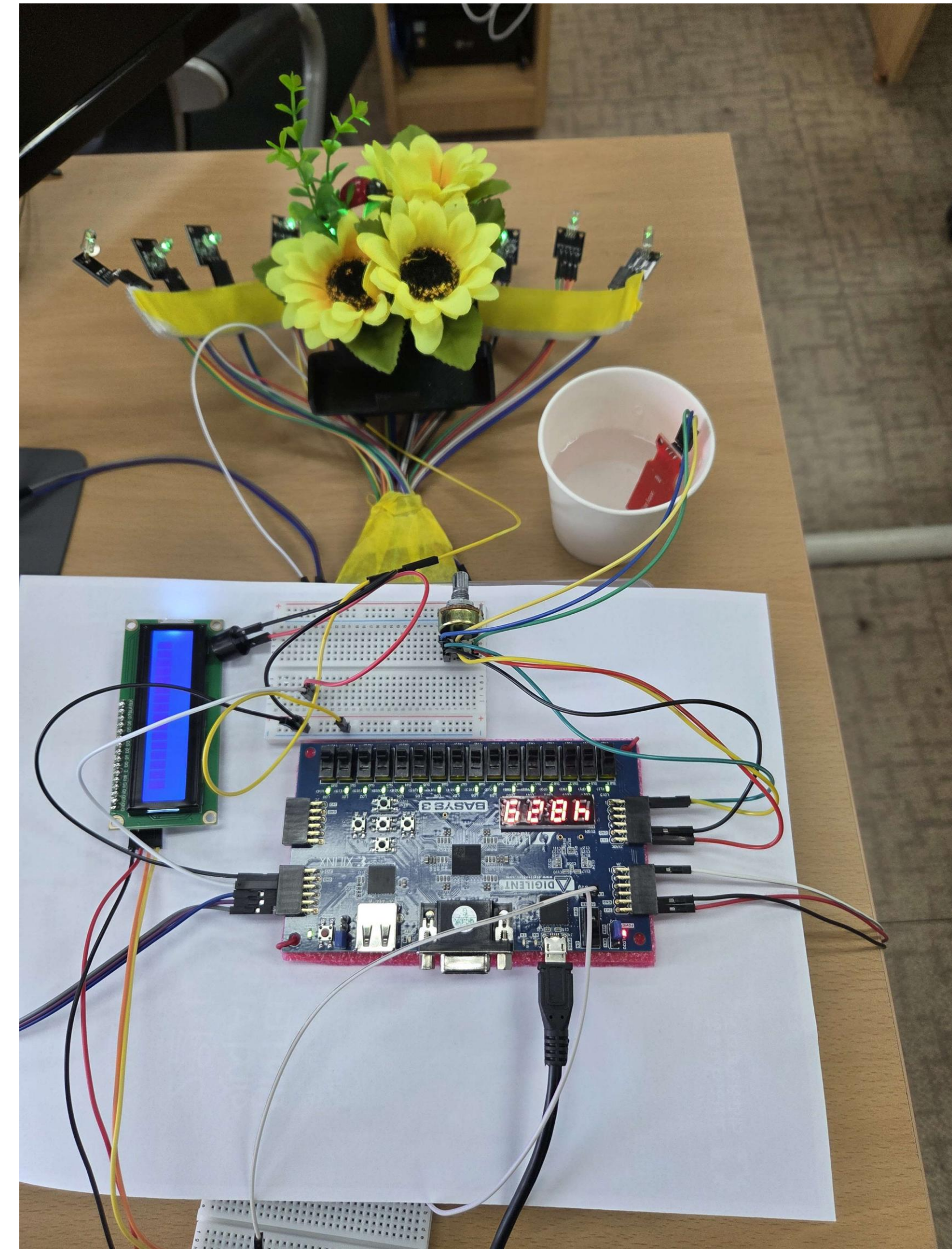
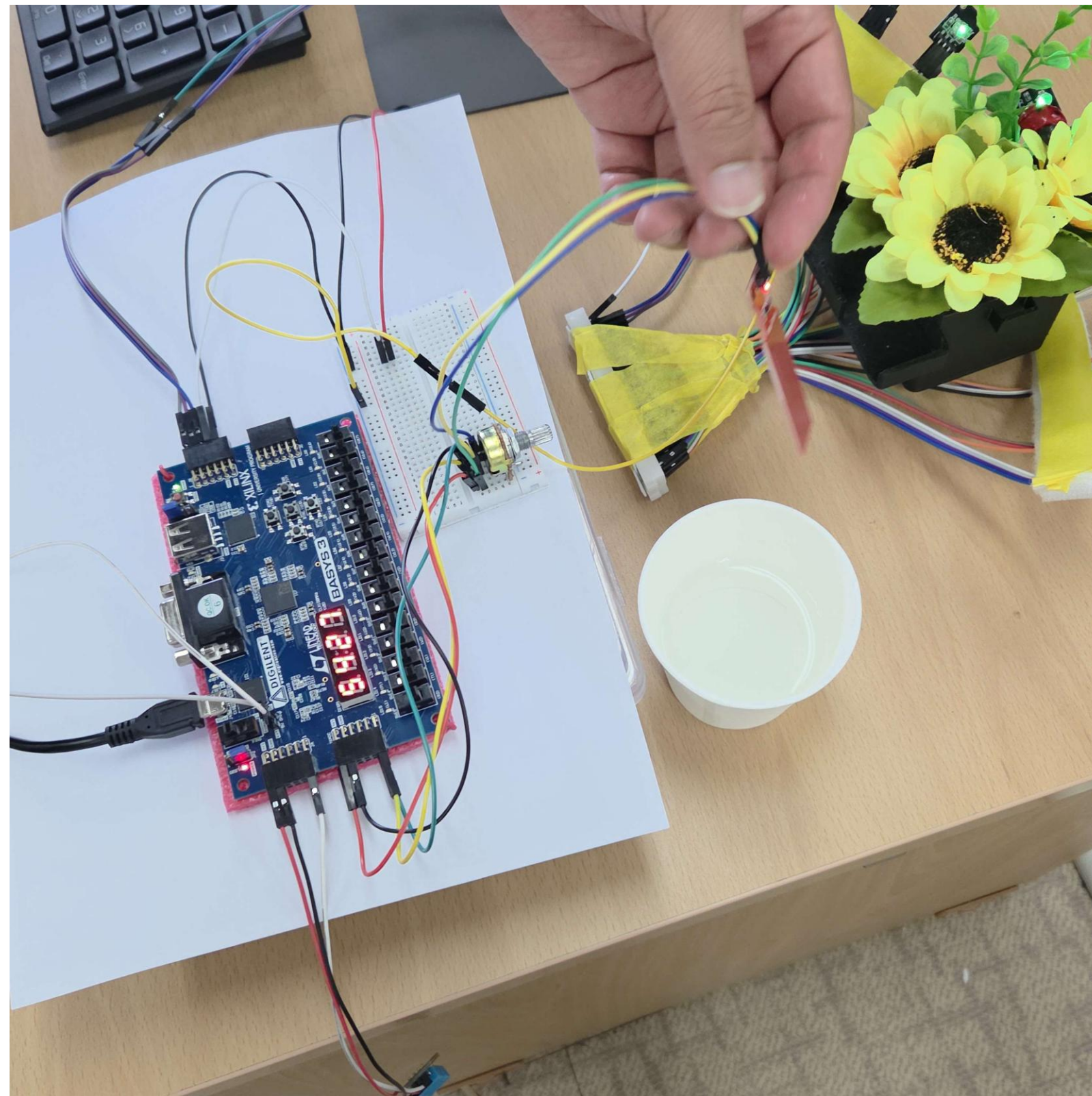
// ADC 값에 따라 LED 및 텍스트 명령 제어
always @(posedge clk or posedge reset_p) begin
    if (reset_p) begin // 리셋 로직

    end
    else begin
        if (clk_usec_pedge) begin
            // ADC 값에 따른 LED 및 텍스트 명령 처리
            if (adc_value <= 10) begin // Sad Face 표시 로직
                ...
                else begin
                    text_cmd <= 3; // Sad Face
                    led = 16'b0000_0000_0000_1111;
                end
            end
            else if (adc_value <= 20) begin // Normal Face 표시 로직
                ...
                end
                else begin
                    text_cmd <= 4; // Normal Face
                    led = 16'b0000_0000_1111_1111;
                end
            end
            else if (adc_value <= 30) begin // Smile Face 표시 로직
                ...
                else begin
                    text_cmd <= 2; // Smile Face
                    led = 16'b0000_1111_1111_1111;
                end
            end
            else begin // Happy Face 표시 로직
                ...
                ...
                else begin
                    text_cmd <= 1; // Happy Face
                    led = 16'b1111_1111_1111_1111;
                end
            end
        end
    end
end
end

```

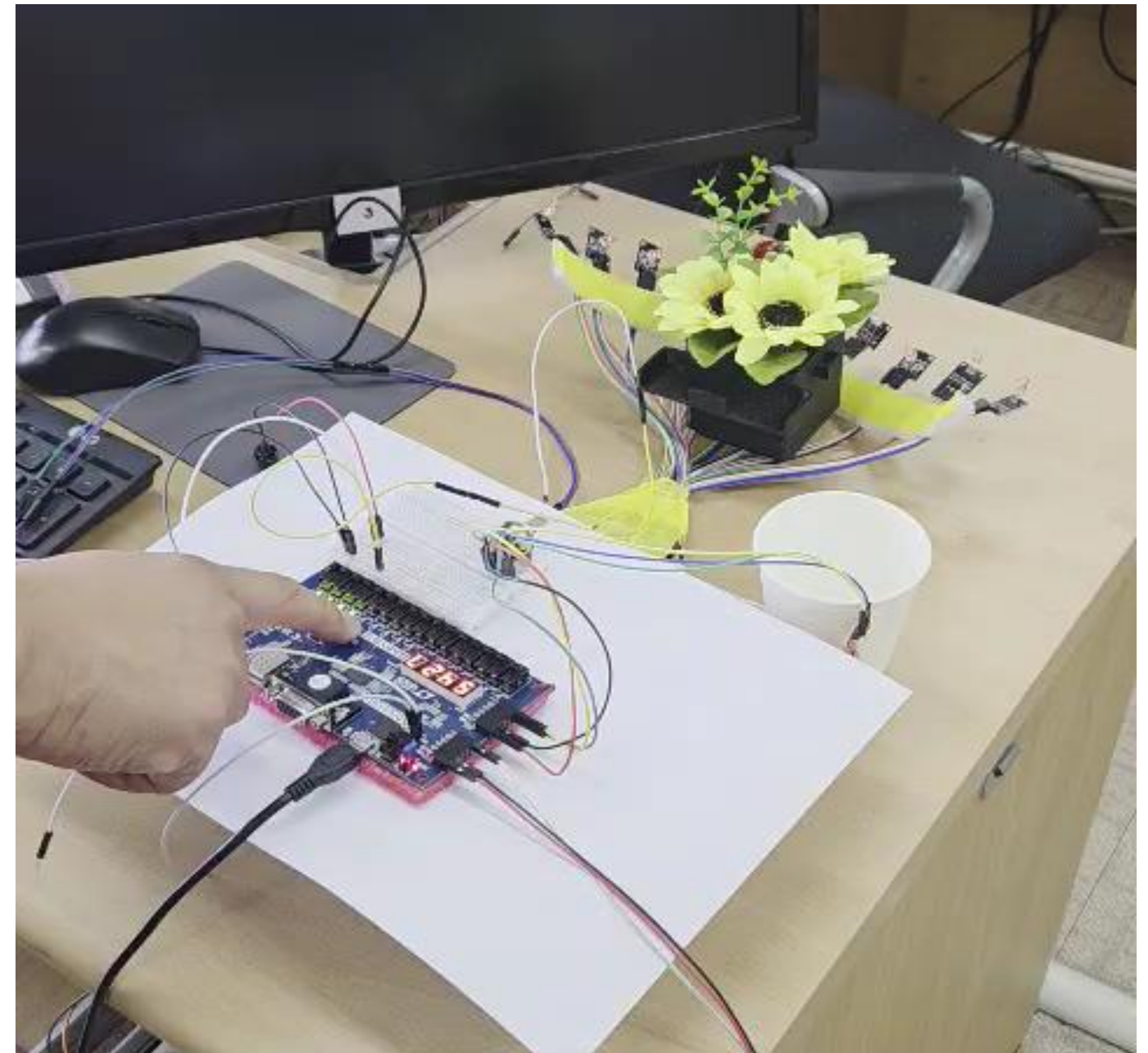
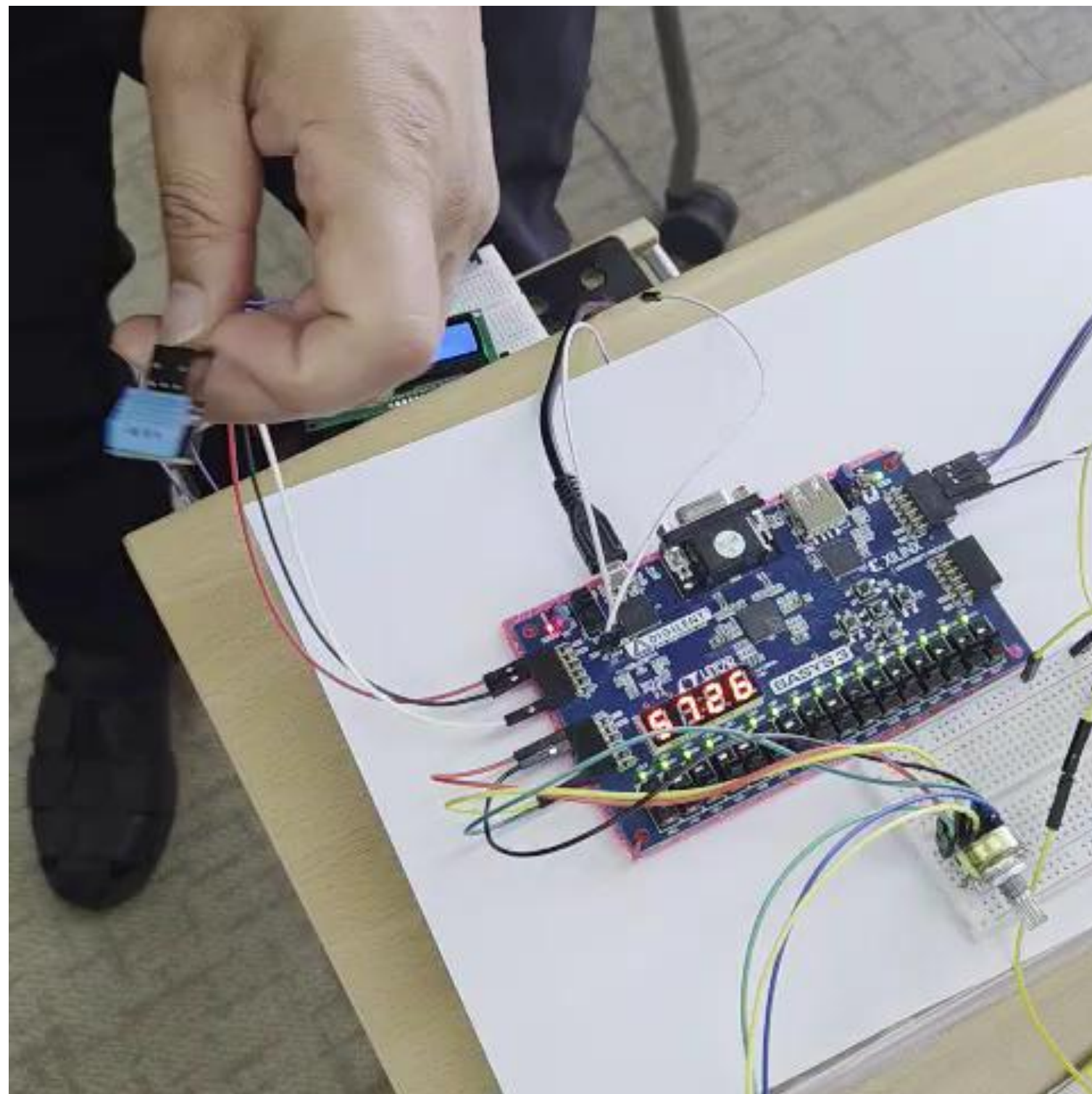

프로젝트 결과물

완성된 프로젝트 결과물



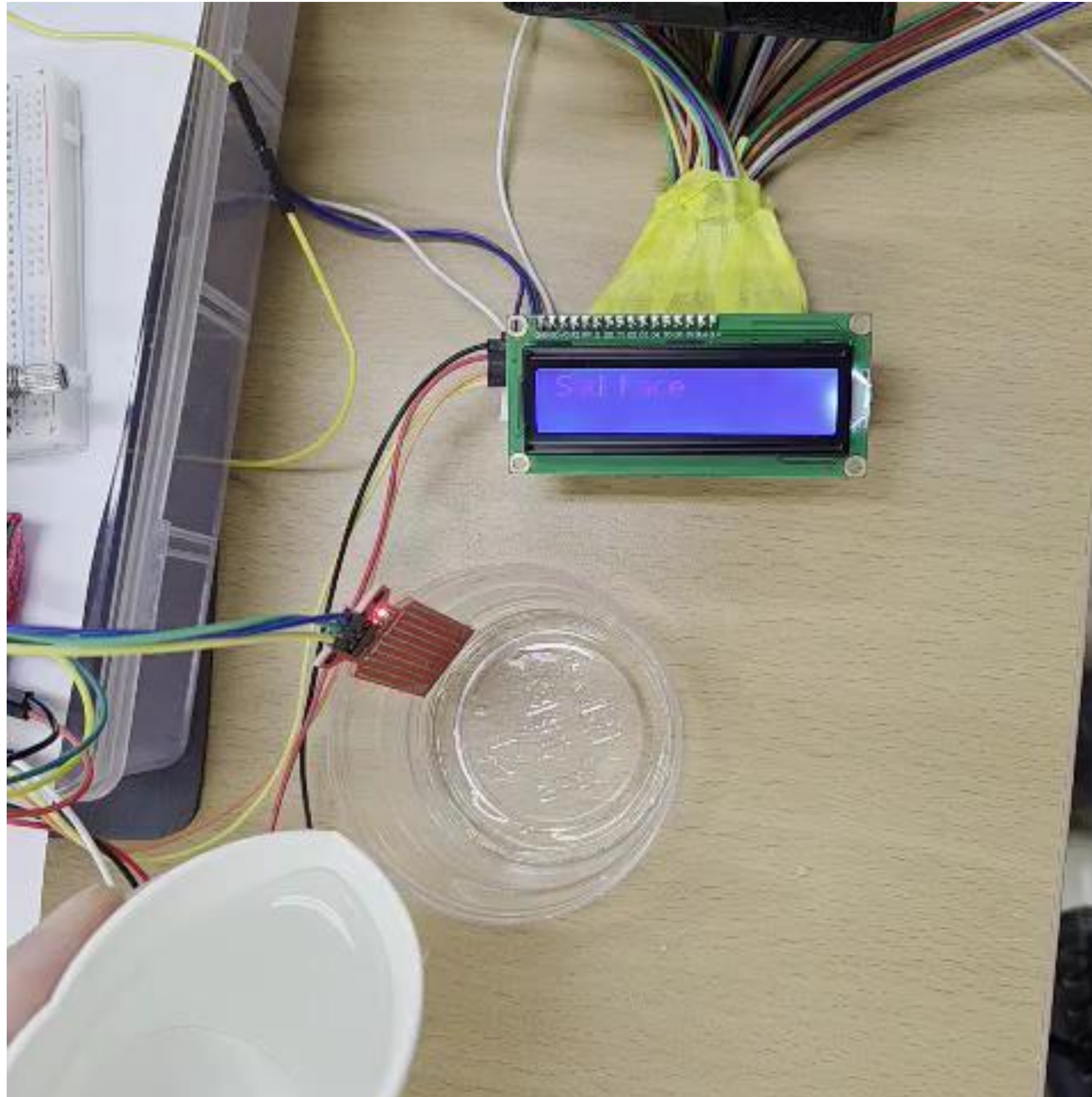
프로젝트 결과물

완성된 프로젝트 결과물



프로젝트 결과물

완성된 프로젝트 결과물



마무리

초기 프로젝트 계획&문제점/개선사항&추후 방향성

초기 프로젝트 방향성&계획

- 다양한 표정을 LCD로 구현
- LED 색상으로 화분 상태 확인
- 토양 온도 센서를 이용
- 물 습도 센서 측정
- 대기 온/습도 센서 측정

마무리

초기 프로젝트 계획&문제점/개선사항&추후 방향성

문제점

- 물 수위 센서 정밀도 문제
- LED 색 3가지만 나오는 문제
- LCD 표정/문구 출력 문제
- 온/습도 센서 기능X

개선 사항

- 기존 물 수위 센서 보다 고성능의 물 수위 센서 사용 & 물 수위 센서 장착 위치를 여러 개 사용하여 기존 센서의 정밀도 개선
- RGB LED 제어를 PWM 방식으로 변경하여 다양한 색 표현이 가능하도록 기능개선
- 온/습도 값을 Text LCD에 표시되도록 기능 개선 필요
- 물 수위 변경 시 Text LCD 문자가 튕는 현상 개선 필요

마무리

초기 프로젝트 계획&문제점/개선사항&정리 및 향후 계획

정리

- 물 수위 센서를 활용하여 다양한 표정을 LCD로 구현
- RGB LED 로 화분 상태 확인
- 물 수위 센서 에 대한 제어
- 대기 온/습도 센서 제어 방법

향후 계획

- 기존 대기중의 온/습도를 토양의 온/습도 측정하는 센서로 변경
- 블루투스/와이파이를 이용한 원격 제어 기능
- TFT LCD를 활용하여 다양한 이모티콘(표정) 기능