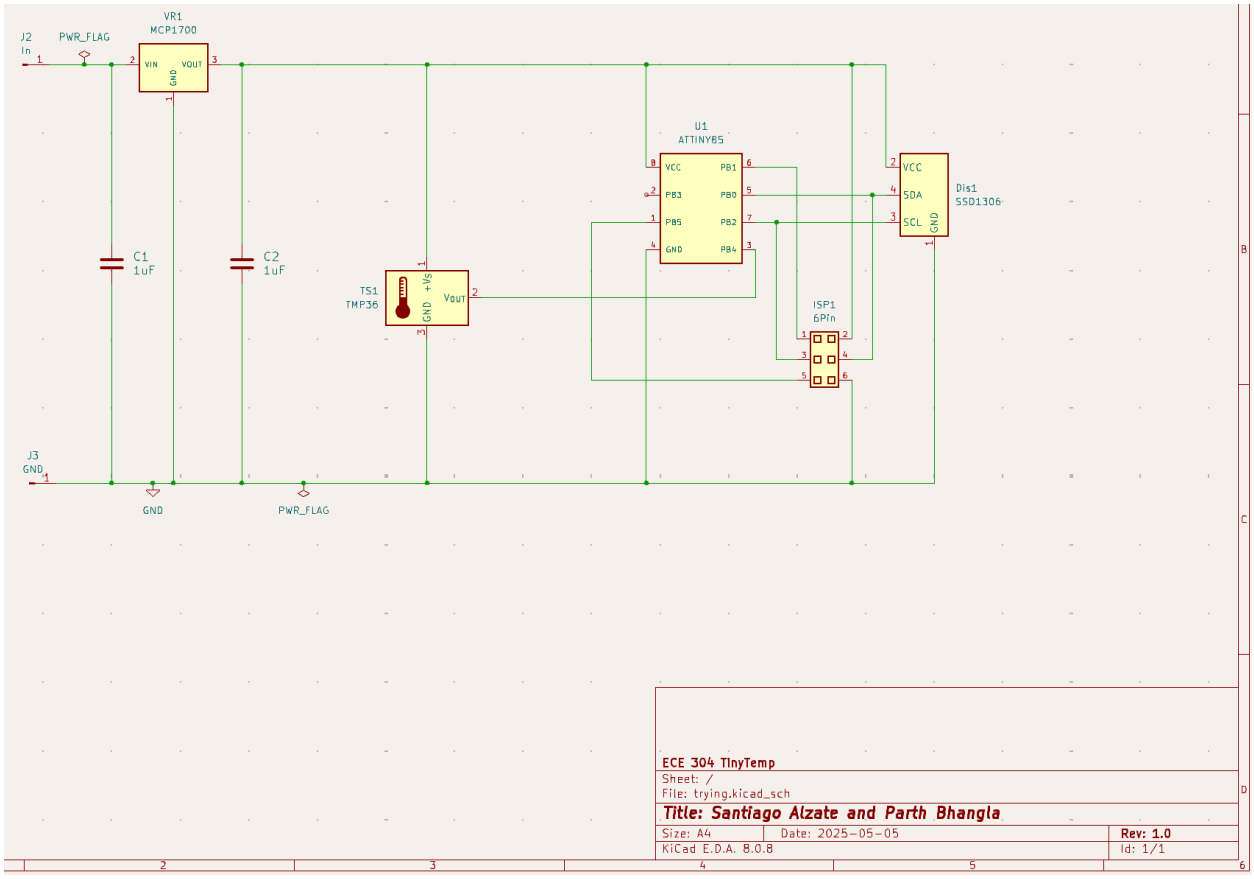
E&C-ENG 304: Junior Design Course

TinyTemp Project Report

Name(s):
Parth Bhangla
Santiago Alzate

Design Documentation

Schematic from KiCad



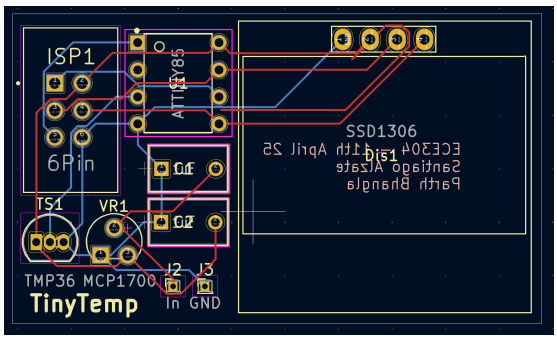
Discussion:

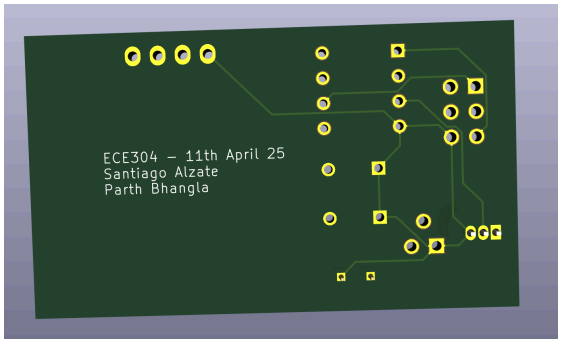
This schematic was designed using KiCad and serves as the foundation of our thermometer system. At the core of the circuit is the ATtiny85 microcontroller, which reads analog temperature data from the TMP-36 sensor connected to the ADC2 input (PB4).

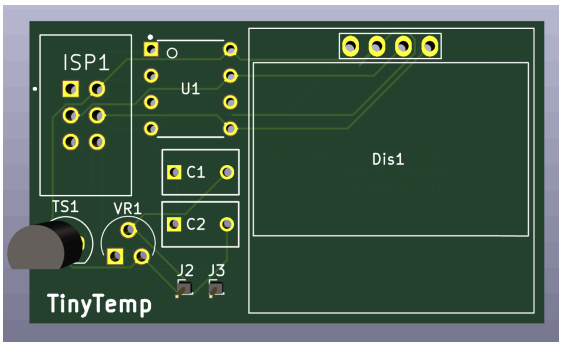
Temperature readings are shown on a 128x64 SSD1306 OLED display, alerting the user with a “TOO HOT” message when the temperature exceeds 18°C.

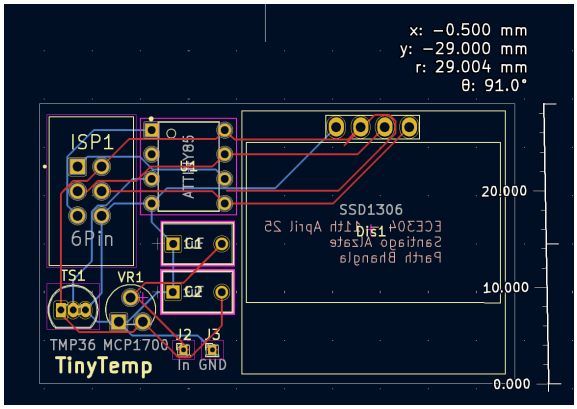
The system is powered by four AA batteries supplying 6 volts. This voltage is regulated down to 3.3V using a linear voltage regulator. To improve power stability and reduce voltage ripple, we placed two capacitors — one on the input side and one on the output — as the regulator’s datasheet recommended. These capacitors help smooth out fluctuations and ensure a consistent power supply to the microcontroller and display.

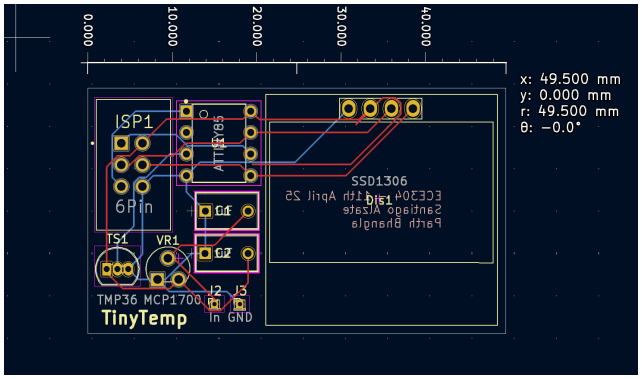
PCB Layout from KiCad









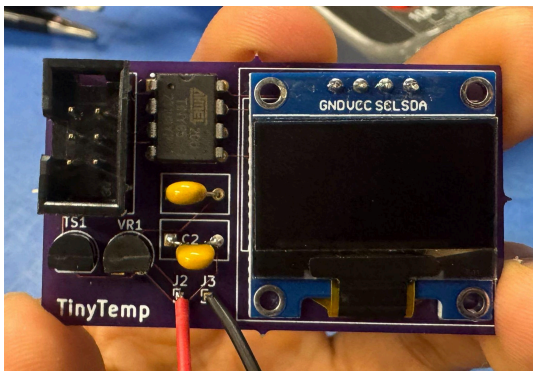


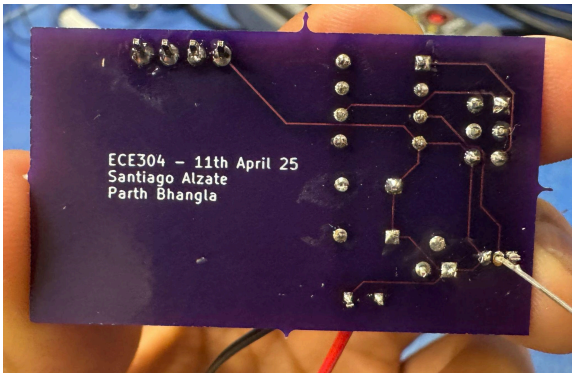
Discussion:

The schematic was designed on KiCad, the main goal was to minimize the board area while maintaining reliable connections between the components. The ATtiny, the voltage regulator, and the OLED Display were all placed within a short distance, to allow direct routing and minimize the connection distance, which helps reduce noise and improve stability.

The board's dimensions are 2.9cm x 4.95cm, which gives us an area of 2.23 square inches, which is below the 3-inch requirement.

Assembled PCB Photos





Discussion:

Soldering and assembling the PCB was quite straightforward except the part where we had to solder the temperature sensor. We ended up using the inline footprint for the temperature sensor which meant we had a really high chance of shorting the solder joints which would mean that the temperature sensor wouldn't work. We soldered this part really carefully trying to use the least amount of solder possible. It came out right.

Source Code

Code:

```
#include "test.h"

#include <avr/io.h>

#include <util/delay.h>

#include <stdlib.h>

#include <string.h>


#define NUMSAMPLES 25

#define VREF 1.1            // Internal reference voltage

#define MAXTEMP_C 18        // Too Hot warning threshold
```

```
// TMP36 connected to ADC2 (PB4)
#define TEMP_SENSOR_CHANNEL 0x02

// Strings to display
const char msgC[] PROGMEM = "TEMP C:";
const char msgF[] PROGMEM = "TEMP F:";
const char tooHotMsg[] PROGMEM = "TOO HOT!";

// Function declarations
void adc_init(void);
uint16_t get_adc(void);

int main(void) {
    char buffer[6];
    float tempC, tempF, voltage;
    uint32_t totalValue;
    uint16_t digitalValue;
    uint8_t xPos;

    OLED_init();
    adc_init();
    OLED_clear();

    while (1) {
        // Take multiple ADC samples and average
        totalValue = 0;
        for (int i = 0; i < NUMSAMPLES; i++) {
            totalValue += get_adc();
            _delay_ms(5);
        }
```

```
    digitalValue = totalValue / NUMSAMPLES;

    // Convert digitalValue to voltage in mV
    voltage = digitalValue * VREF / 1024.0;
    voltage *= 1000;  // in mV

    // TMP36: Temp (°C) = (mV - 500) / 10
    tempC = (voltage - 500.0) / 10.0;
    tempF = tempC * 9.0 / 5.0 + 32.0;

    OLED_clear();

    // Display Temperature in C (Line 0)
    dtostrf(tempC, 4, 1, buffer);
    xPos = (128 - (strlen_P(msgC) + strlen(buffer)) * 6) / 2;
    OLED_cursor(xPos, 0);
    OLED_printP(msgC);
    for (uint8_t i = 0; i < strlen(buffer); i++) {
        OLED_printC(buffer[i]);
    }

    // Display Temperature in F (Line 1)
    dtostrf(tempF, 4, 1, buffer);
    xPos = (128 - (strlen_P(msgF) + strlen(buffer)) * 6) / 2;
    OLED_cursor(xPos, 1);
    OLED_printP(msgF);
    for (uint8_t i = 0; i < strlen(buffer); i++) {
        OLED_printC(buffer[i]);
    }
```

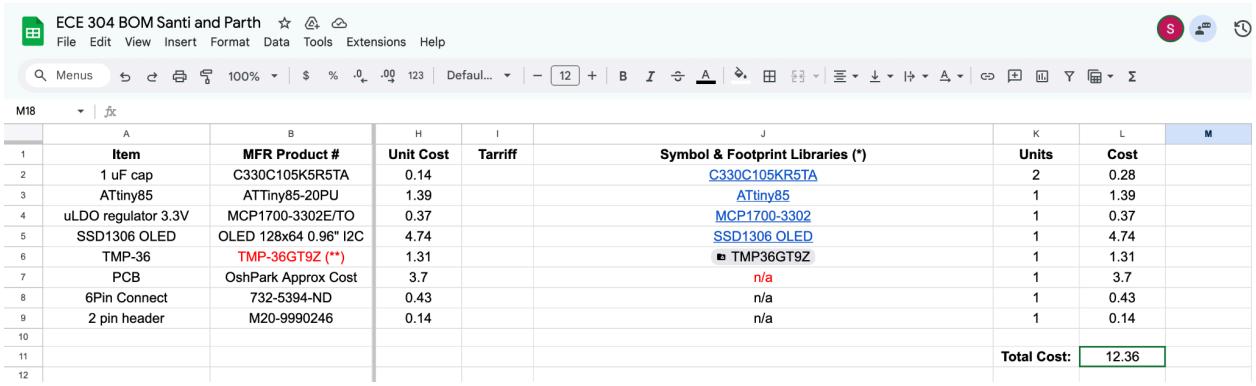
```
        // Show Too Hot warning (Line 2)
        xPos = (128 - strlen_P(tooHotMsg) * 6) / 2;
        OLED_cursor(xPos, 2);
        if (tempC >= MAXTEMP_C) {
            OLED_printP(tooHotMsg);
        } else {
            for (uint8_t i = 0; i < strlen_P(tooHotMsg); i++) {
                OLED_printC(' ');
            }
        }

        _delay_ms(1000);
    }
    return 0;
}

// ADC setup: use Vref = 1.1V, ADC2
void adc_init(void) {
    ADMUX = (1 << REFS1) | TEMP_SENSOR_CHANNEL; // ADC2, Vref=1.1V
     ADCSRA = (1 << ADEN) | (1 << ADPS1) | (1 << ADPS0); // Enable ADC, prescaler 8 for ATtiny85 @ 1MHz
}

// Read from ADC
uint16_t get_adc(void) {
    ADCSRA |= (1 << ADSC); // Start conversion
    while (!(ADCSRA & (1 << ADIF))); // Wait for conversion
    ADCSRA |= (1 << ADIF); // Clear interrupt flag
    return ADC;
}
```

Cost



ECE 304 BOM Santi and Parth

File Edit View Insert Format Data Tools Extensions Help

M18

	A	B	H	I	J	K	L	M
1	**Item**	**MFR Product #**	**Unit Cost**	**Tarriff**	**Symbol & Footprint Libraries (*)**	**Units**	**Cost**	
2	1 uF cap	C330C105K5R5TA	0.14		C330C105KR5TA	2	0.28	
3	ATtiny85	ATTiny85-20PU	1.39		ATtiny85	1	1.39	
4	uLDO regulator 3.3V	MCP1700-3302E/TO	0.37		MCP1700-3302	1	0.37	
5	SSD1306 OLED	OLED 128x64 0.96" I2C	4.74		SSD1306 OLED	1	4.74	
6	TMP-36	TMP-36GT9Z (**)	1.31		TMP36GT9Z	1	1.31	
7	PCB	OshPark Approx Cost	3.7		n/a	1	3.7	
8	6Pin Connect	732-5394-ND	0.43		n/a	1	0.43	
9	2 pin header	M20-9990246	0.14		n/a	1	0.14	
10								
11						**Total Cost:**	12.36	
12								

Requirements Verification

Requirement	Requirement Met	Verification
1.1 Measure room temp without user interaction	Yes	Shown in video demo
1.2 Measure room temp over range 0C/32F - 45C/113F	Yes	Based on TMP-36 spec
1.3 Measure room temp with accuracy +/- 4 deg C	Yes	Based on TMP-36 spec
2.1 Display temp in both F and C	Yes	Shown in video demo
2.2 Display precision 0.1 degrees	Yes	Shown in video demo
2.3 Display temperature at all times	Yes	Shown in video demo
3.0 Provide an indication that it is too hot when temp exceeds a pre-set value	Yes	Shown in video demo
4.1 Battery lifetime 2 months (threshold requirement)	Yes	Analysis: Current drawn from the device is 3.3 mA. Battery capacity is 6000 mAh corresponding to lifetime of 1818.19 hours = 2.49 months.
4.2 Battery lifetime >3 months (design goal)	No	Undercuts 3 months battery life design goal by 15 days.
5.0 Use ATtiny85MCU, SSD 1306 OLED and TMP-36 sensor	Yes	Inspect schematic, KiCad layout, and photos of PCB.
6. Cost <$15 for parts and PCB (in quantities of 1000)	Yes	See BoM and Cost estimate (uploaded as another attachment)
7. Install on PCB < 3 square inches (threshold requirement)	Yes	PCB measures 4.95cm by 2.9cm for a total area of 2.23 square inches.
8. Install on PCB < 2 square inches (design goal)	No	Overshoots the value by 0.23 square inches.
9. Show name, date, and “ECE-304” on silk screen	Yes	Check the back 3D view of the PCB image attached above.