**Firmware for Value Reading from Sensor:**

// Well Documented code for better understanding.

//Referance 1. https://embeddedexplorer.com/esp32-gpio-tutorial/

//          2. https://my-esp-idf.readthedocs.io/en/latest/api-reference/peripherals/gpio.html

//          3. https://www.espressif.com/sites/default/files/documentation/esp32\_technical\_reference\_manual\_en.pdf

//     page: 26,32, 63,

// i've used above for making this code i haven't add gpio.h header, i've directly mapped resisters.

#include <stdint.h>

#include <stdio.h>

#define GPIO\_INPUT\_PIN 25 // Define GPIO pin number

// Register addresses

#define GPIO\_ENABLE\_REG 0x3FF44020  // GPIO enable register

#define GPIO\_OUT\_REG    0x3FF44004  // GPIO output register

#define GPIO\_IN\_REG     0x3FF4403C  // GPIO input register           --- Added for future use

#define GPIO\_PIN\_MUX\_REG 0x3FF49024 // Pin multiplexing register    --- Added for future use

// Function to set a GPIO pin as input

void gpio\_set\_direction(uint8\_t gpio\_num, uint8\_t mode) {

    if (mode) { // If mode is 1, set as output

        \*(volatile uint32\_t \*)(GPIO\_ENABLE\_REG) |= (1 << gpio\_num);

    } else { // Set as input if 0 is placed

        \*(volatile uint32\_t \*)(GPIO\_ENABLE\_REG) &= ~(1 << gpio\_num);

    }

}

// Function to read the level of a GPIO pin

uint8\_t gpio\_get\_level(uint8\_t gpio\_num) {

    return (\*(volatile uint32\_t \*)(GPIO\_IN\_REG) >> gpio\_num) & 0x01; // & 0x01 is used to isolate the least significant bit (LSB)

}                                                                    // of the value read from the GPIO input register, mask all, (to extract a single bit from a multi-bit value)

void main() {

    // Set GPIO\_INPUT\_PIN as input

    gpio\_set\_direction(GPIO\_INPUT\_PIN, 0);

    while (1) {

        // Read the level of the input pin

        uint8\_t level = gpio\_get\_level(GPIO\_INPUT\_PIN);

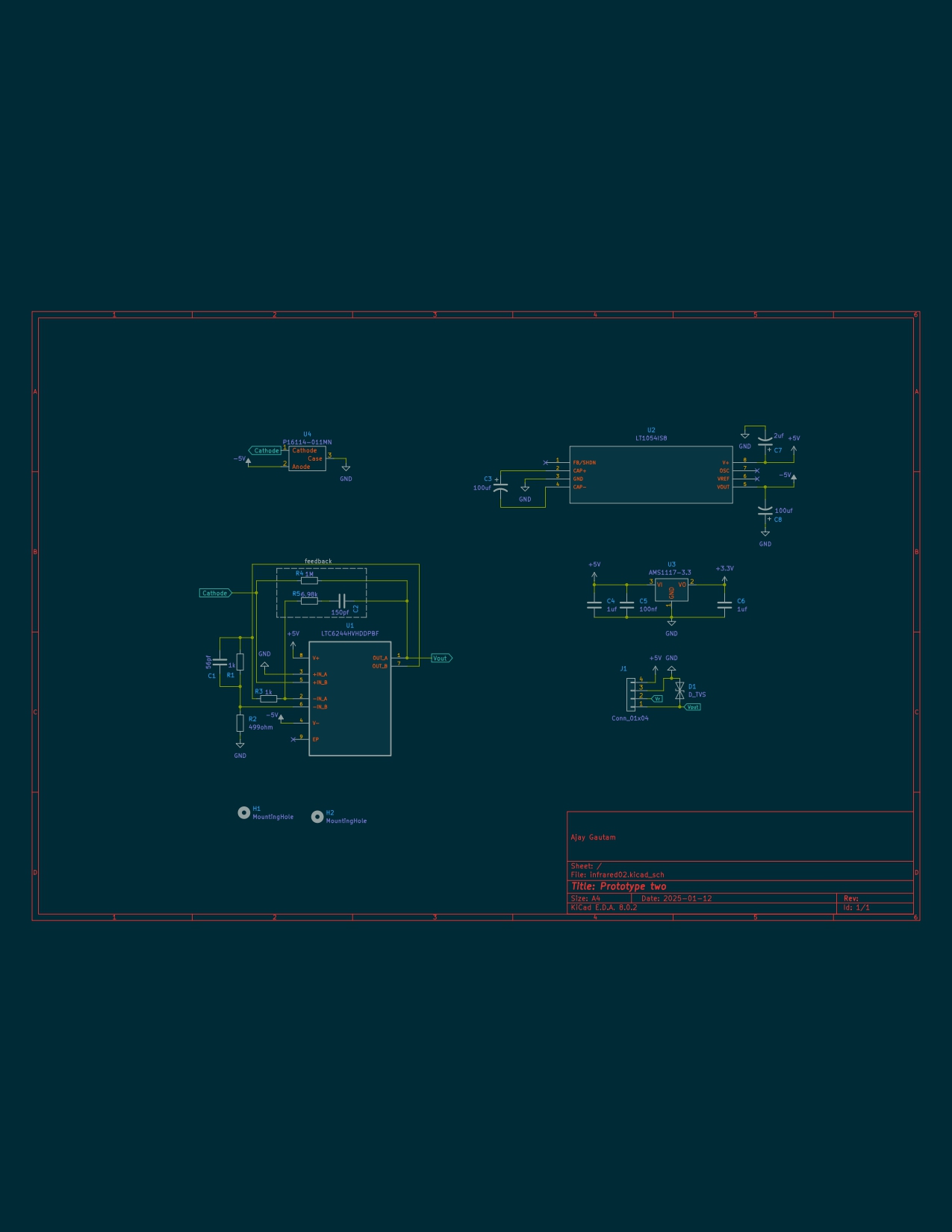
        // Print the level to Terminal or serial monitor. same time connect FFT Analyzer

        printf("GPIO %d Level: %d\n", GPIO\_INPUT\_PIN, level);

        // Add a delay if necessary (e.g., using vTaskDelay)

    }

}

**Updated Schematic:**

**Remark:**

1. **Sensor P16114-011MN is used.**
2. **Pcb layout routing is Pending**
3. **Firmware is Well Documented, Added citation for better understanding.**

**Pending work:**

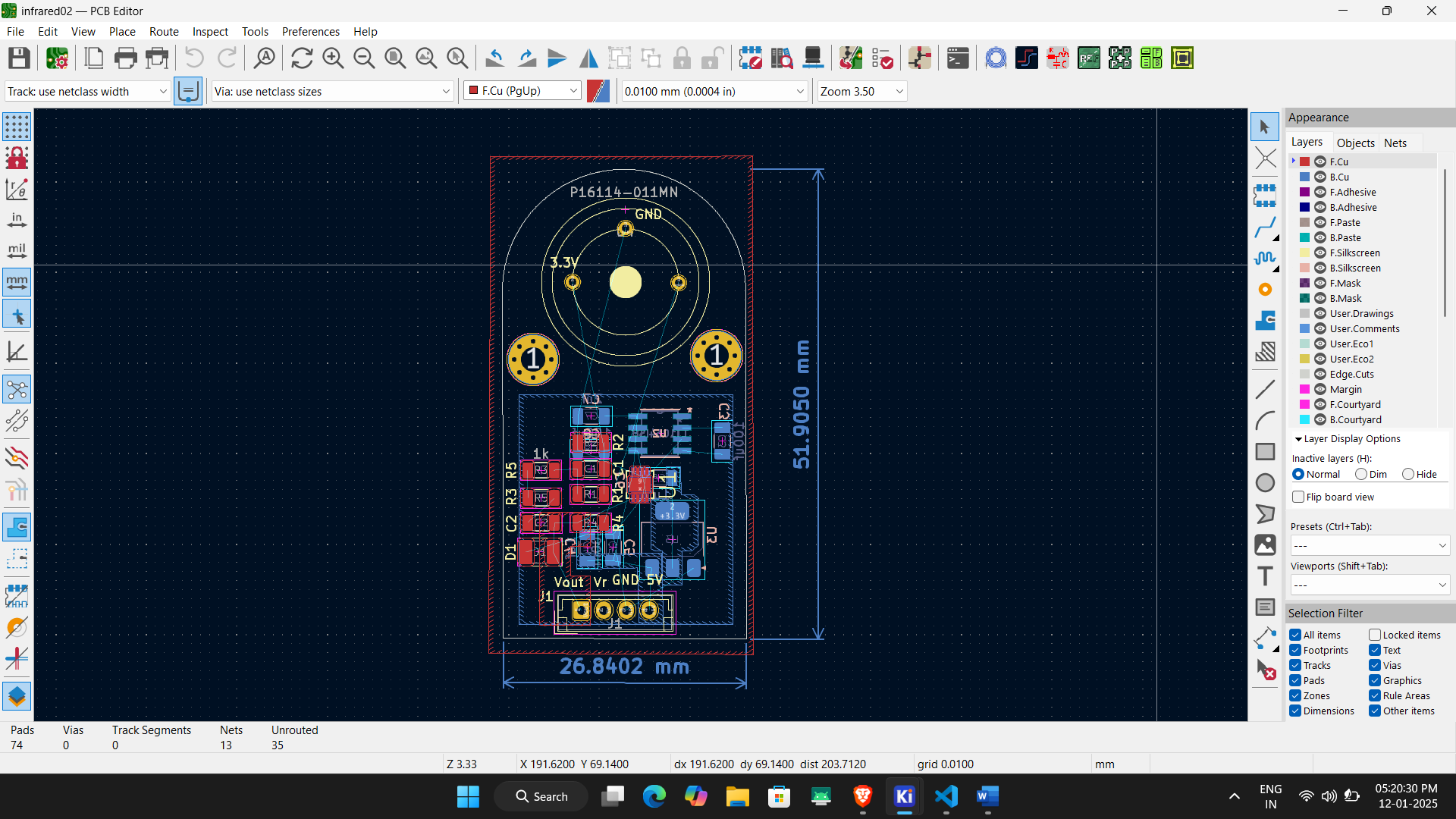


Figure: Pcb layout