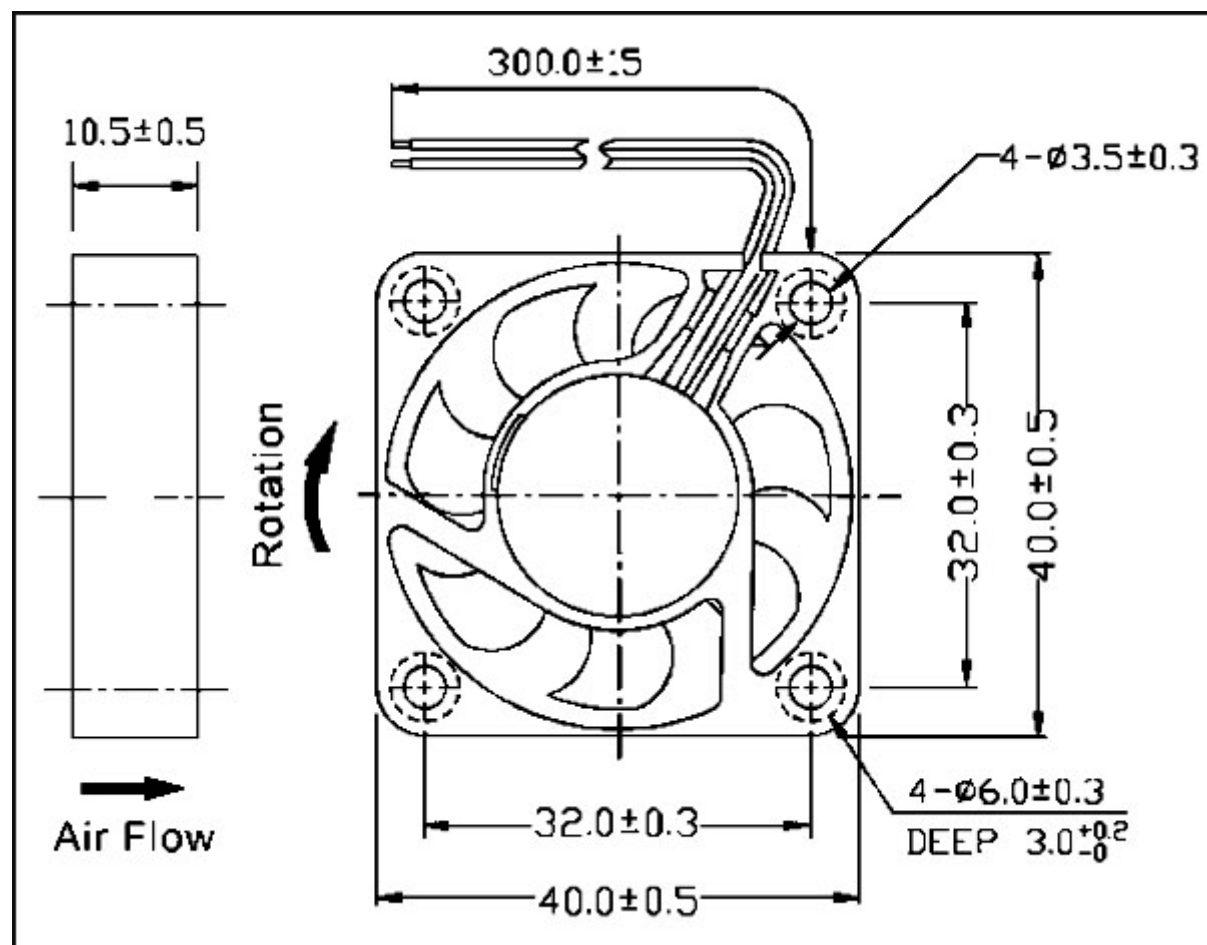


Skizzen

- [Fans](#)
- [Stepper motor](#)
- [ssd](#)
- [Raspberry Pi zero W 2](#)
- [Neue Seite](#)

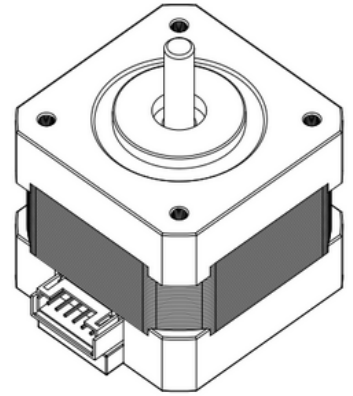
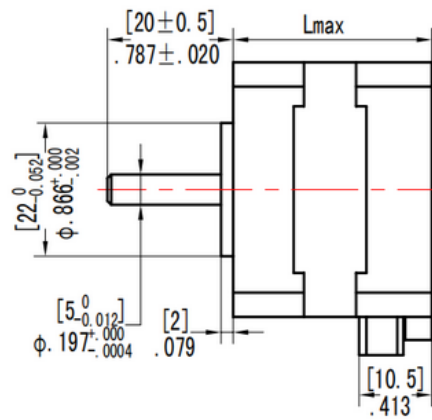
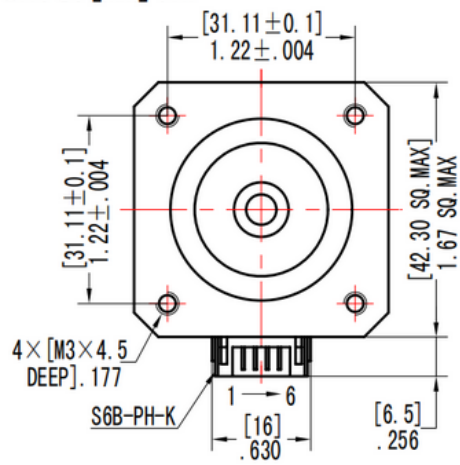
Fans

fan 40x40x10

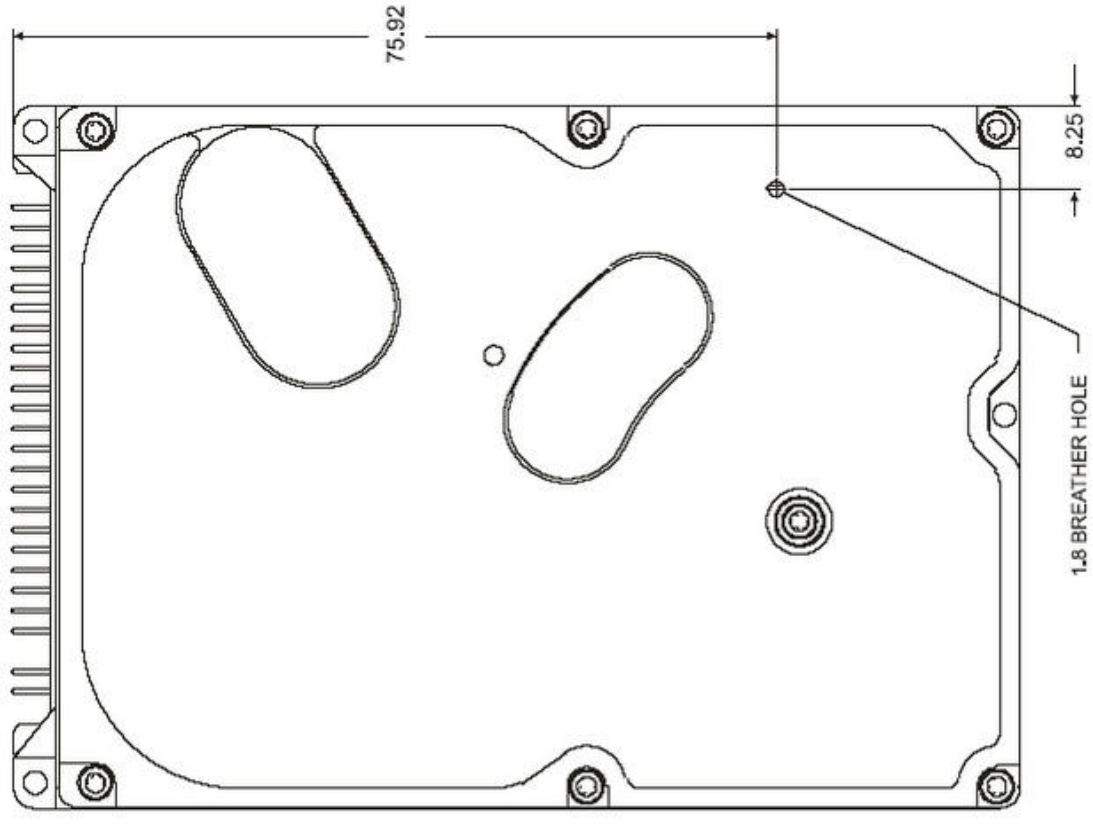
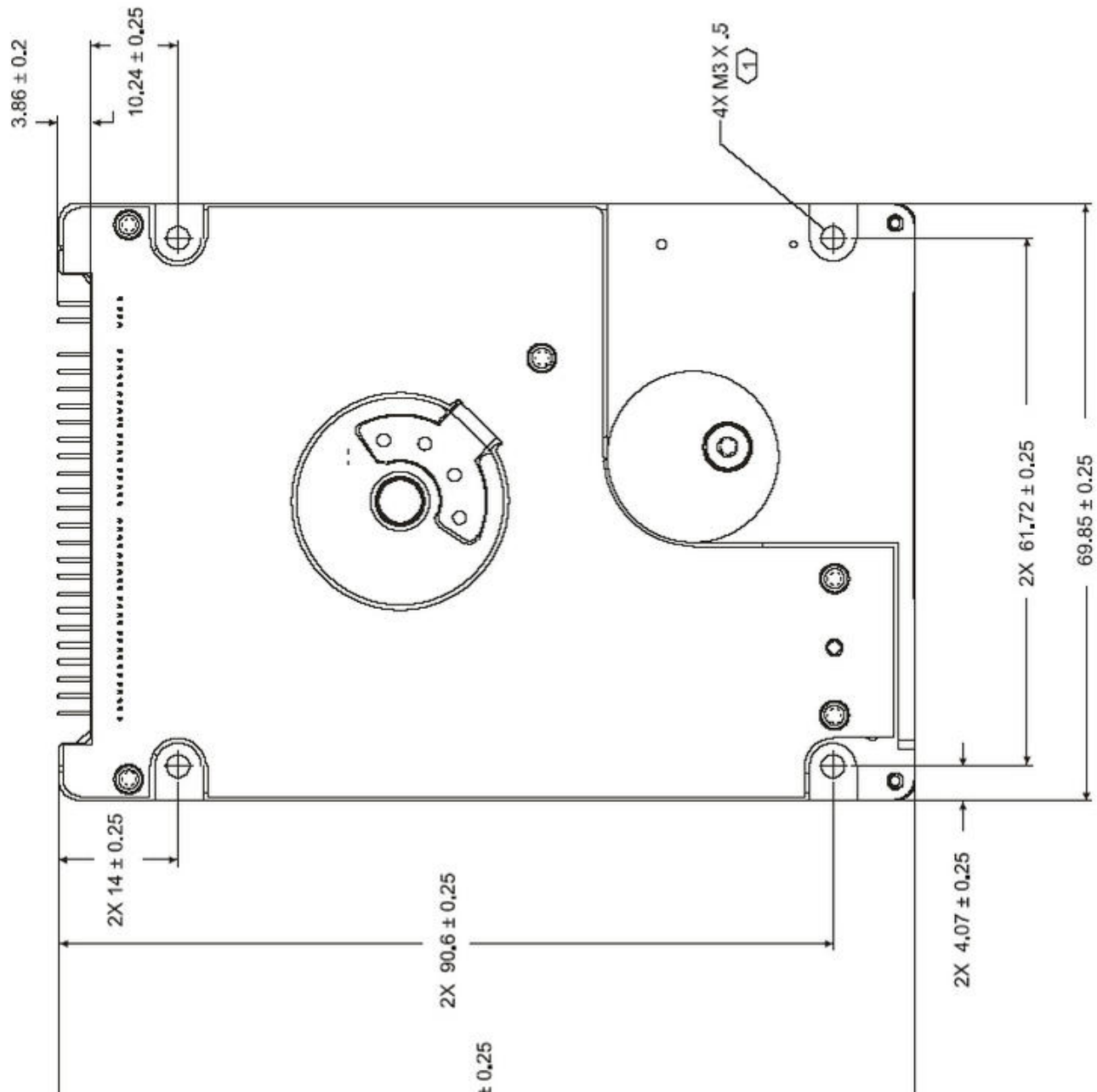


Stepper motor

Unit: [mm] in.

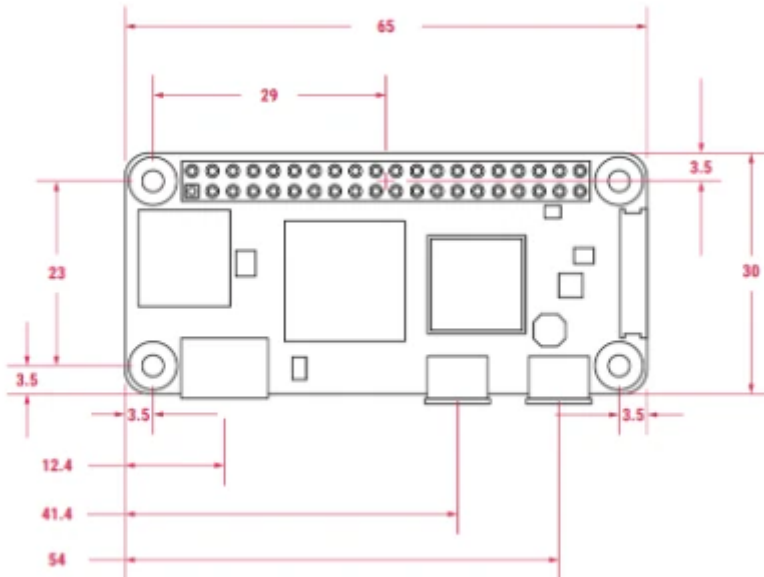


ssd



Raspberry Pi zero W 2

Foot Print



Note: all dimensions in mm

Neue Seite

```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <SPIFFS.h>
#include <Preferences.h>

Preferences preferences; // For saving Wi-Fi credentials

const char *ssid = "ESP32-AP"; // Access point SSID
const char *password = "123456789"; // Access point password (optional)

AsyncWebServer server(80); // Web server on port 80

// HTML code for the Wi-Fi configuration page
const char* htmlForm = R"(
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ESP32 Wi-Fi Configuration</title>
  <script>
    function submitForm() {
      var ssid = document.getElementById('ssid').value;
      var pass = document.getElementById('password').value;
      fetch('/savewifi', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/x-www-form-urlencoded',
        },
        body: 'ssid=' + ssid + '&password=' + pass
      }).then(response => response.text())
        .then(data => {
          alert(data);
        });
    }
  )
```



```

    </script>
</head>
<body>
    <h1>Wi-Fi Configuration</h1>
    <form onsubmit="event.preventDefault(); submitForm();">
        <label for="ssid">SSID:</label><br>
        <input type="text" id="ssid" name="ssid" required><br><br>
        <label for="password">Password:</label><br>
        <input type="password" id="password" name="password" required><br><br>
        <input type="submit" value="Save Wi-Fi">
    </form>
</body>
</html>
)";

void setup() {
    // Start the serial monitor
    Serial.begin(115200);

    // Mount SPIFFS filesystem
    if (!SPIFFS.begin(true)) {
        Serial.println("SPIFFS Mount Failed");
        return;
    }

    // Initialize Wi-Fi as Access Point (AP)
    WiFi.softAP(ssid, password);
    Serial.println("Access Point Started");
    Serial.print("IP Address: ");
    Serial.println(WiFi.softAPIP());

    // Handle root path for the Wi-Fi form
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
        request->send(200, "text/html", htmlForm);
    });

    // Handle the form POST request to save Wi-Fi credentials
    server.on("/savewifi", HTTP_POST, [](AsyncWebServerRequest *request){
        String ssid = request->arg("ssid");
        String password = request->arg("password");
    });
}

```

```

    // Save Wi-Fi credentials to Preferences
    preferences.begin("wifi", false);
    preferences.putString("ssid", ssid);
    preferences.putString("password", password);
    preferences.end();

    // Provide feedback to the user
    request->send(200, "text/plain", "Wi-Fi credentials saved. Rebooting...");

    // Reboot the ESP32 to attempt Wi-Fi connection
    delay(1000);
    ESP.restart();
});

// Start the web server
server.begin();
}

void loop() {
    // Check if Wi-Fi credentials are stored
    preferences.begin("wifi", true);
    String ssid = preferences.getString("ssid", "");
    String password = preferences.getString("password", "");
    preferences.end();

    // If credentials are stored, try to connect to Wi-Fi
    if (ssid != "" && password != "") {
        WiFi.begin(ssid.c_str(), password.c_str());
        Serial.print("Connecting to Wi-Fi");
        while (WiFi.status() != WL_CONNECTED) {
            delay(500);
            Serial.print(".");
        }
        Serial.println("\nConnected to Wi-Fi");
        Serial.print("IP Address: ");
        Serial.println(WiFi.localIP());
    }

    // Add any other functionality here

```

```
delay(1000);
```

```
}
```