

基于 RK3588 的光谱共焦光学轮廓仪

摘要

本项目基于 RK3588 芯片设计了一款光谱共焦光学轮廓仪，采用宽光谱卤素灯与线性色散镜头实现光轴方向色散，通过 Y 型光纤构建共焦探测结构，利用彩色工业相机提取反射光斑的色相信息以获取微小位移。为替代传统光谱仪，本系统创新使用相机采集拜耳格式图像，并基于 HSV 色彩空间中的 H 分量构建色相与位移的插值模型，实现三维表面形貌重建。系统软硬件采用模块化设计，内核层通过设备树和驱动模块实现电机控制，用户层基于 QT 实现图形界面和数据可视化，支持 CSV 导出与三维网格显示。整体方案兼具高分辨率、低成本、可扩展等优点，适用于工业检测等精密测量场景。

第一部分 作品概述

1.1 功能与特性

本作品开发的测量设备基于光谱共聚焦原理，通过将连续光谱沿光轴方向展开实现三维形貌测量。具有分辨率较高，稳定可靠，检测快速的特点。

1.2 应用领域

物体三维表面形态的检测，工业生产中具有微小构造的非透明较高反射率工件的表面检测。

1.3 主要技术特点

该系统光学部分采用光纤耦合卤素灯光源，其具备优异的光谱均匀性及较小的光学孔径。配合线性色散镜头，将宽光谱的白光色散成在光轴轴向波长线性分布的准单色光，再配合光纤口作为光学微孔实现共聚焦，即只有焦点处的准单色光能有效地反射回光纤被采集到。

传统光谱共焦技术采用的是光谱仪作为反射光信号检测装置，但光谱仪往往价格高昂且对使用环境有一定要求，考虑作品设计目标，认为换用彩色工业相机用来分辨颜色具有可行性与易用性。

通过采用彩色工业相机采集物体表面反射光斑经过 Y 型光纤出光口的图像，在光斑中心区域提取像素点色相数据。通过采集相机原始拜耳格式 (Bayer RG12) 而非 RGB，避免相机本身的 ISP 算法优化导致颜色出现偏差。再基于图像转换算法转换到 HSV 颜色空间，分析 H 色相分量分辨颜色 (Hue)，从而进行标定，得出在输入光源稳定下的色相和位移的关系。

1.4 主要技术指标

指标名称	参数值
x,y (水平) 分辨率	0.01 mm
z (深度) 分辨率	0.05mm
x,y 量程	40mm
z 量程	2.5mm

表 1 主要技术指标

1.5 主要创新点

- a. 与传统激光共聚焦扫描系统相比，本设备采用轴向色散补偿技术，消除了 Z 轴机械扫描需求，大大缩短了扫描时间。
- b. 使用光纤减小了光学孔径，从而基本消除球差，可以测量大深径比微孔目标。
- c. 光谱共焦位移传感器结构简单，相对于激光干涉仪，其尺寸小且质量轻，可在多种测量场合进行快速安装和集成。

1.6 设计流程

1.6.1 设计目标

设计目标为开发一款基于 RK3588 芯片的光谱共焦光学轮廓仪，能够实现高精度的三维形貌测量。系统应具备稳定可靠的性能，适用于工业生产中具有微小构造的非透明较高反射率工件的表面检测。

1.6.2 光学系统设计

光学系统采用宽光谱卤素灯作为光源，该光源具有优异的光谱均匀性特性。光学路径设计包含三个关键组件：直径不超过 5mm 的 Y 型光纤耦合器、覆盖 400-700nm 可见光波段的线性色散物镜，以及专门设计的反射光采集方案。检测方案尝试使用彩色工业相机替代传统光谱仪进行光学信号采集，降低了成本。

1.6.3 机械系统设计

机械系统由高精度 XY 平移台和 Z 轴位移台组成，共同实现 xOy 平面内的高精度位移测量功能。机械装配采用模块化设计理念，其中 XY 轴采用独立模块化结构设计，并通过专用连接板实现两轴联动控制。

1.6.4 电路系统设计

电源方案采用两级转换架构：第一级为 220V 转 24V 开关电源模块，第二级为 DCDC 转换模块。控制接口设计包含两个关键部分：步进电机驱动器的信号定义接口和光电开关的检测接口。

1.6.5 软件系统设计

内核层开发包含三项核心技术：设备树修改（新增电机控制节点）、PWM 驱动改造（支持 OneShot 工作模式）以及内核模块开发（提供 sysfs 接口）。用户层实现基于 QT6.4 框架的图形界面，集成海康 SDK 的相机控制模块，并开发了运动控制逻辑算法。数据处理层采用 OpenCV 进行光斑中心提取，并应用 GSL 科学计算库实现色相-位移的插值计算。

1.6.6 数据处理与标定

图像采集优化采用 RAW 格式 (RG12) 进行数据采集，同时关闭自动白平衡和黑电平功能。标定流程建立位移与色相值的对应关系模型，并构建插值映射函数。三维重构过程通过 XY 平面扫描控制获取空间数据，并利用 QT DataVisualization 模块生成三维网格模型。

1.6.7 测试与验证

硬件组装阶段完成光学、机械和电子系统的联合调试。软件测试实施从内核驱动到应用层的全链路验证流程。性能测试包含重复性测试、精度测试和稳定性评估三个关键指标。

1.6.8 总结

整个设计流程采用模块化开发策略，先完成光学原型验证，再逐步集成机械和电子系统，最后通过软件系统实现自动化测量与可视化。

第二部分 系统组成及功能说明

2.1 整体介绍

系统总体分为硬件系统和用户终端，硬件系统由光学模块、机械模块、电路模块组成，用户终端为 RK3588 开发板。

光学模块负责实现高精度的深度信息采集，机械模块用于在 xOy 平面内实现样品的精确移动与扫描，电路模块为各子系统提供稳定的电源支持，用户终端则承担运动控制与图像数据处理等核心功能。

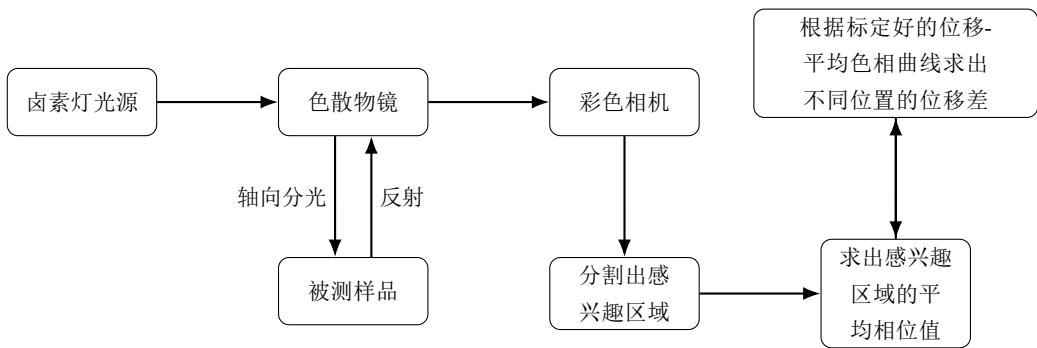


图 1 系统工作原理示意图

2.2 硬件系统介绍

2.2.1 硬件整体介绍

光学模块由宽光谱卤素灯，线性色散物镜，工业相机和y型光纤组成，主要利用光的色散对镜头和工件表面起伏导致的相对位移按颜色编码获取深度信息；机械模块由高精度XY平移台和Z轴位移台组成，实现xOy平面的高精度的位移测量；电路模块由输入220V输出24V的开关电源，DCDC组成，负责给各个模块提供稳定的电源支持。信号连接板负责将RK3588的GPIO和PWM分配给步进电机控制器，并将光电开关的信号转成3.3V输入到ELF2开发板的40PIN引脚上。

2.2.2 光学模块介绍

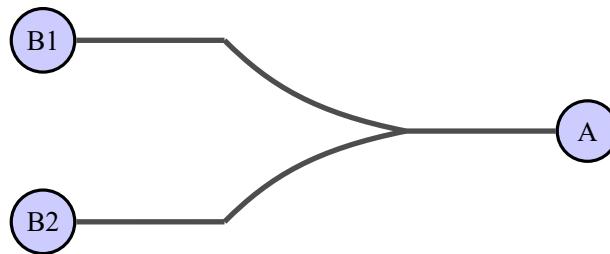


图 2 Y型光纤耦合器结构示意图

卤素灯光源具有显著的光谱均匀性与较小的光学孔径（直径 $\leq 5mm$ ）。通过y型光纤的B端接口连接卤素灯，其发出的宽光谱光线被高效耦合至光纤内部，继而从A端接口出射至预配置的色散物镜。该筒形镜组基于棱栅分光原理，在可见光波段（400 – 700nm）实现工作。经其色散作用，不同波长的光产生轴向色差，焦点以 $\pm 0.5\%$ 的均匀性误差沿光轴分布为连续聚焦区域（光学探针）。需满足样品表面反射率 $>5\%$ （标准白板参考值）的技术要求。

当样品表面置于光学探针对应聚焦位置时，特定波长的反射光强度显著增强（约3倍于离焦状态），该光信号经原路返回色散物镜，最终由光纤另一B端传输至彩色相机。在相机获取的HSV色彩空间图像中提取，分割出预定感兴趣区

域（ROI）并计算其 H 通道平均色相值。基于标定后插值所得的位移-色相值映射曲线，可导出微位移量。通过驱动 xy 位移平台（定位精度 $1\mu m$ ）执行区域扫描，遂重构样品表面三维形貌。

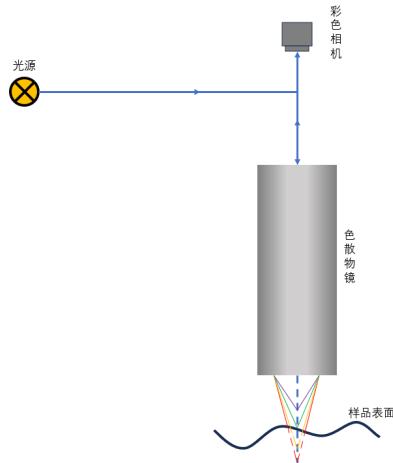


图 3 光路示意图

2.2.3 机械设计介绍

主体使用 PLA3D 打印和 2020 铝型材

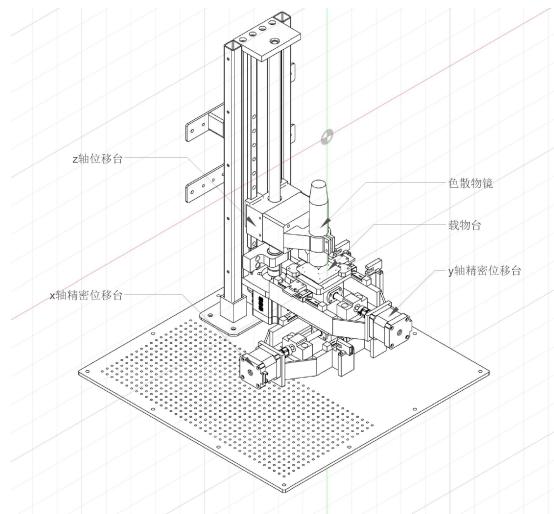


图 4 总体机械装配图

其中 Z 轴位移台基于 1205 丝杆，精度较低，因为其目的只是使色散物镜工作在工作区间，无需使用高精度位移台，这样做可以降低成本。

x、y 轴精密位移台基于 0801 滚珠丝杆和 MGN9 导轨滑块，二者结构完全一致，只是 y 轴位移台的载物台处在 x 轴位移台上安装了连接板，用于连接 x 轴位移台与 y 轴位移台

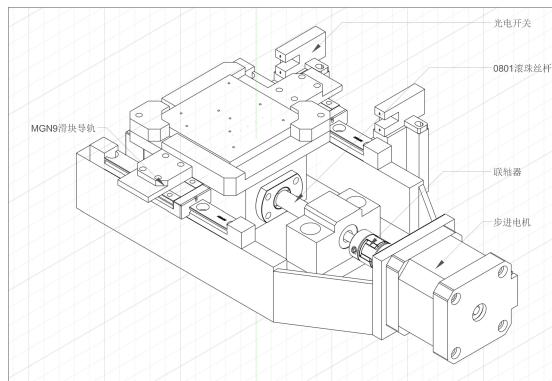


图 5 × 轴精密位移台结构图

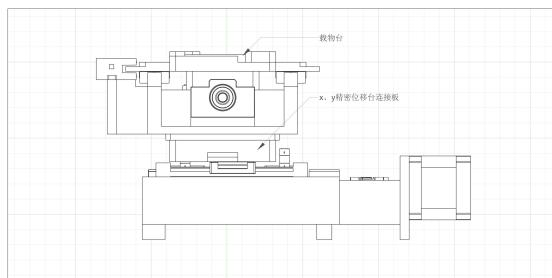


图 6 ×、y 轴精密位移台连接示意图

基于高精度 xy 位移台，可以实现 xOy 平面的高精度的位移测量。

2.2.4 电路设计介绍

电路设计主要是设计两种电路板。一种是作为分电板兼 DCDC 降压板 ($24V \rightarrow 12V$) 给 3 个 DM542 步进电机控制器 ($24V$) 卤素光源供电 ($12V 48W$)，由于卤素灯开关对电源有较大冲击，会再单独使用同一种 DCDC 降压模块 ($24V \rightarrow 12V$) 给 RK3588 供电。另一块板是作为步进电机控制器的信号连接板，主要是将 RK3588 的 GPIO 和 PWM 分配给给 3 个 DM542 步进电机控制器，并将光电开关的信号转成 $3.3V$ 输入到 ELF2 开发板的 40PIN 引脚上。

DCDC 降压基于 XL4016 芯片设计，输入 $24V$ 输出 $12V$ ，最大输出电流 $6A$ 。设计图如下：

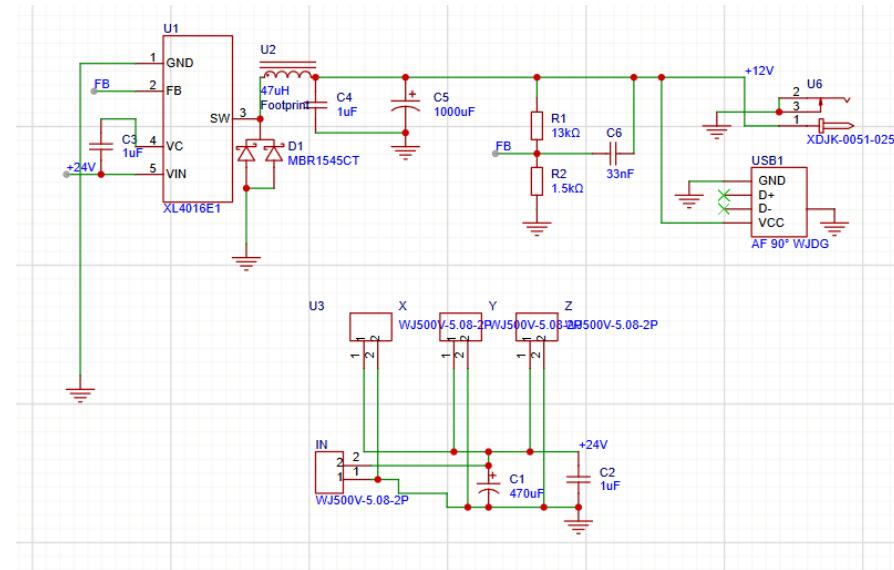


图 7 DCDC 降压电路原理图

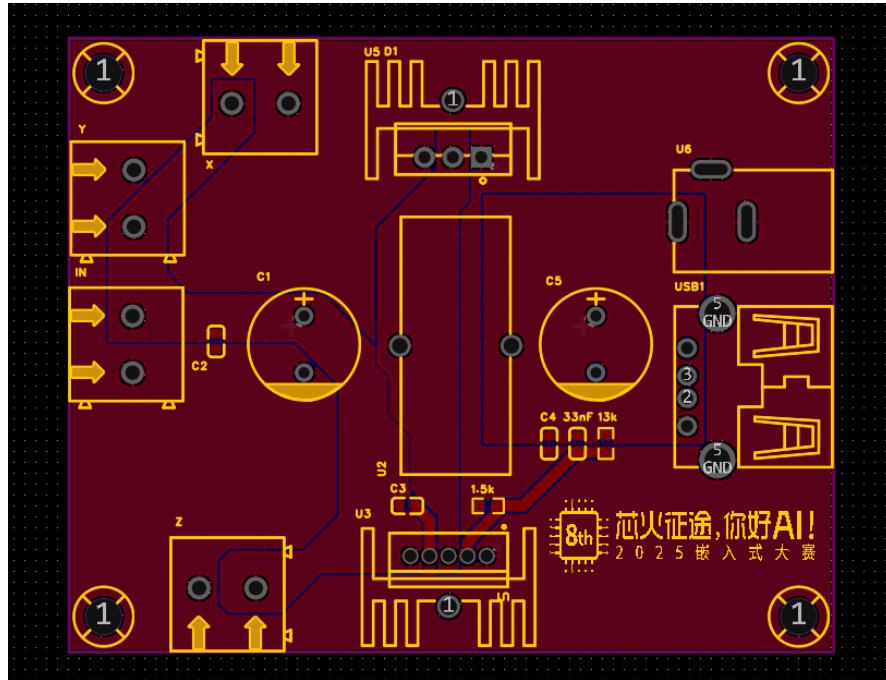


图 8 DCDC 降压电路 Layout

信号连接板基于SN74LVC4245APWR5V 转 3.3 芯片设计，设计图如下：

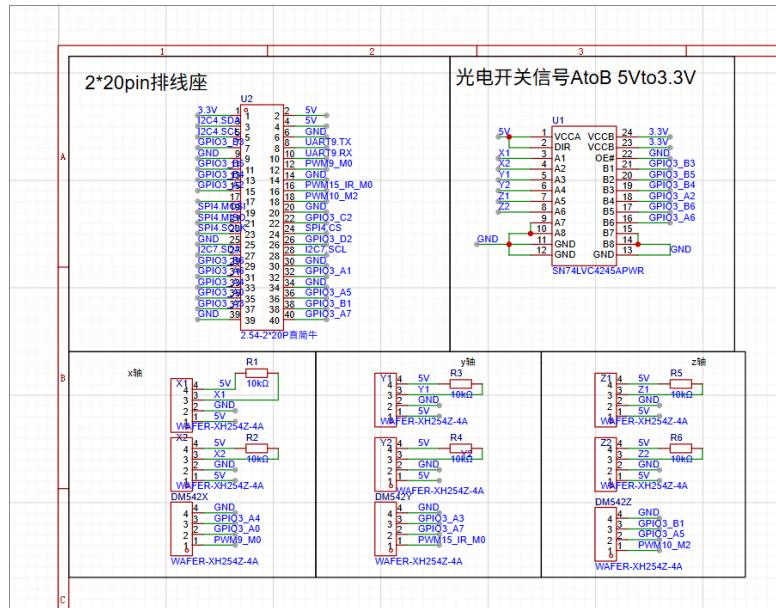


图 9 信号连接电路原理图

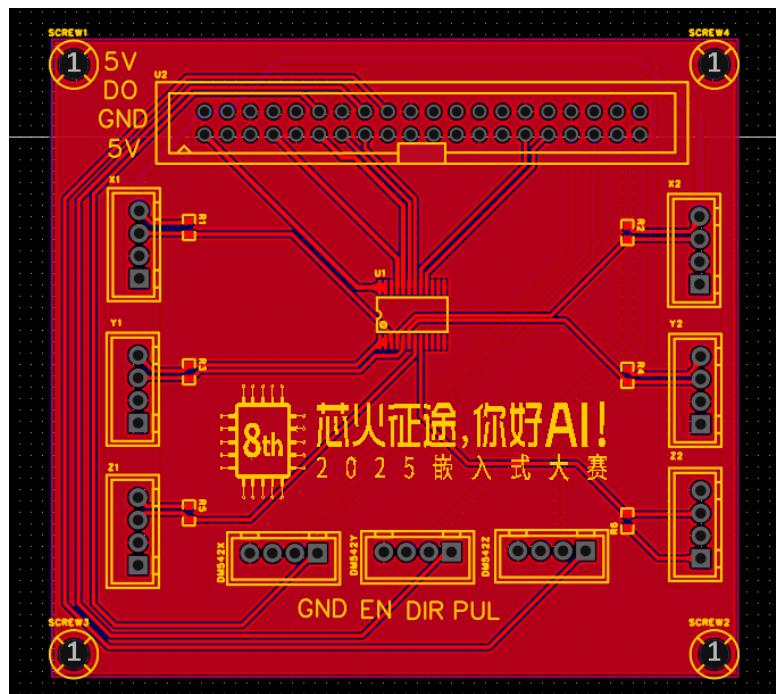


图 10 信号连接电路 Layout

2.3 软件系统介绍

2.3.1 软件整体介绍

软件由所用硬件决定，我们使用的 ELF2 开发板基于 RK3588 芯片，不同于开发板所给的 SDK（linux Kernel 5.10 Ubuntu 22.04）。我们使用的内核是 rockchip-linux 的 Kernel 6.1 分支修改移植原有设备树（Device Tree）。再基于 Joshua-Riek

的 ubuntu-rockchip 项目构建基于 Ubuntu 24.04 的根文件系统（Root Filesystem）。再依靠运行环境，内核态（Kernel Space）修改设备树和部分硬件驱动，编写驱动电机驱动器的内核模块并设计相关 sysfs 接口给用户态。用户态（User Space）选择 QT6.4 作为图形界面开发框架，使用 C++ 作为主要编程语言。测量数据可导出成 CSV 文件处理或通过 QT 的 Data Visualization 查看。

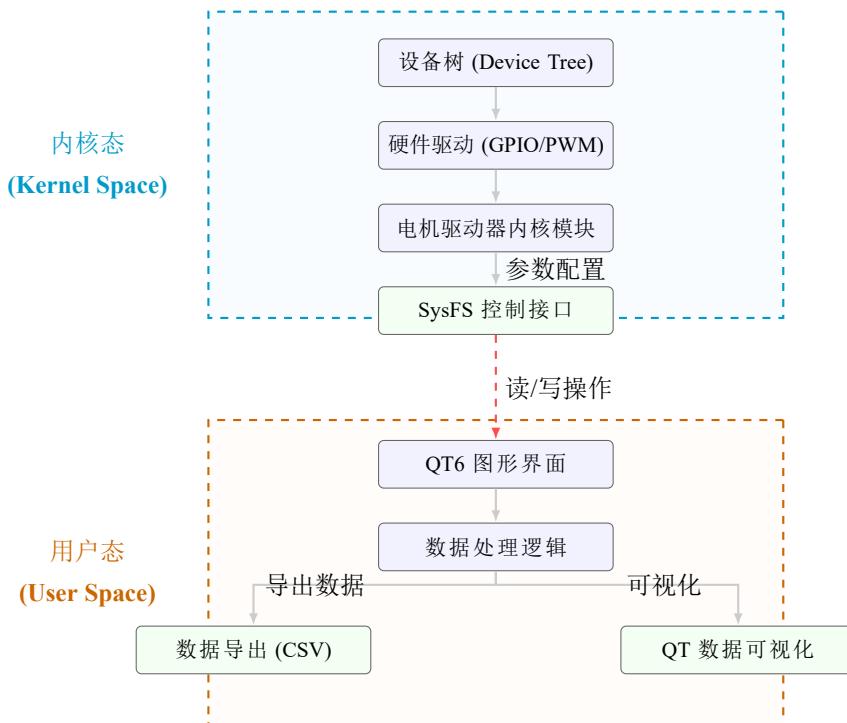


图 11 软件系统架构图

2.3.2 运行和开发环境介绍

运行环境为 RK3588 芯片的 Ubuntu 24.04 系统，内核版本 6.1。使用 QT6.4 作为图形界面开发框架。开发使用 WSL2+Docker 开发环境，方便区分内核态和用户态开发。内核态驱动使用 C 语言编写用 Makefile 编译，用户态程序使用 C++ 编写用 Cmake 编译 QT 程序。为方便交叉编译直接用 dpkg 添加 ARM64 架构的 Ubuntu 24.04 软件源，使用 apt 安装所需的开发工具和库，并采用动态链接库的方式编译 QT 程序。减少编译工作量和依赖问题。

2.3.3 设备树修改

设备树是 Linux 内核用于描述硬件设备的结构化数据。我们在 RK3588 的设备树中添加了步进电机控制器的相关节点并启用所需的 PWM 节点，以便内核能够识别并正确配置这些硬件。

```

/ {
    dm542_x:dm542-x{//X轴步进电机控制器
        compatible = "dm542";
        status = "okay";
        ena{//控制使能引脚
            gpios = <&gpio3 RK_PA4 GPIO_ACTIVE_HIGH>;
            pinctrl-names = "default";
            pinctrl-0 = <&dm542_x_ena>;
        };
        dir{//方向引脚
            gpios = <&gpio3 RK_PA0 GPIO_ACTIVE_HIGH>;
            pinctrl-names = "default";
            pinctrl-0 = <&dm542_x_dir>;
        };
        pul{//脉冲输入
            pwms = <&pwm9 0 50000 0>;
        };
        sw1{//滑台两侧光电开关
            gpios = <&gpio3 RK_PB3 GPIO_ACTIVE_HIGH>;
            pinctrl-names = "default";
            pinctrl-0 = <&dm542_x_sw1>;
            interrupt-parent = <&gpio3>;
            interrupts = <RK_PB3 IRQ_TYPE_LEVEL_MASK>;
        };
        sw2{
            gpios = <&gpio3 RK_PB5 GPIO_ACTIVE_HIGH>;
            pinctrl-names = "default";
            pinctrl-0 = <&dm542_x_sw2>;
            interrupt-parent = <&gpio3>;
            interrupts = <RK_PB5 IRQ_TYPE_LEVEL_MASK>;
        };
    };
};

&pwm9 {
    pinctrl-0 = <&pwm9m0_pins>;
    status = "okay";//启用PWM9
};

```

同时需要配置 Pinctrl 以设置 GPIO 引脚的工作模式和属性。以下为 X 轴配置：

```

&pinctrl {
    dm542_x_ena: dm542_x_ena {
        rockchip,pins = <RK_PA4 RK_FUNC_GPIO &pcfg_pull_none>;
    };
}

```

```

dm542_x_dir: dm542_x_dir {
    rockchip,pins = <RK_PA0 RK_FUNC_GPIO &pcfg_pull_none>;
};

dm542_x_sw1: dm542_x_sw1 {
    rockchip,pins = <RK_PB3 RK_FUNC_GPIO &pcfg_pull_none>;
};

dm542_x_sw2: dm542_x_sw2 {
    rockchip,pins = <RK_PB5 RK_FUNC_GPIO &pcfg_pull_none>;
};

}

```

通过上述两段代码，我们在设备树中定义了步进电机控制器的相关节点，并启用了所需的 PWM 节点和 GPIO 引脚。

2.3.4 PWM 驱动修改

由步进电机控制器的驱动方式，即移动距离

$$L = \frac{P}{P_r} \times D$$

其中 L 为移动距离， P 为脉冲数， P_r 为每转所需脉冲数， D 为螺距。需要能输出指定个数的脉冲信号才能实现步进电机的精确控制。因此启用 RK3588 的 PWM 的 OneShot 模式，但该模式下仅允许一次向寄存器写入 256 个脉冲个数。所以需要再配合 Linux 的 PWM 结构体的其他空闲参数与 RK3588 本身 OneShot 结束的回调函数实现任意脉冲个数的控制。以下是对回调函数的修改：

```

//pwm-rockchip-irq-callbacks.h
static void rockchip_pwm_oneshot_callback(struct pwm_device *pwm, struct pwm_state *state)
{
    if (!state->oneshot_repeat) {
        if (pwm->last.oneshot_count) {
            state->oneshot_count = pwm->last.oneshot_count;
            pwm->last.oneshot_count = 0;
            state->enabled = true;
            pwm_apply_state(pwm, state);
        } else {
            state->enabled = false;
            pwm_apply_state(pwm, state);
            // Notify the completion of oneshot mode
            blocking_notifier_call_chain(
                &rockchip_pwm_oneshot_done_notifier_list, 0,
                pwm);
        }
    }
    return;
}

```

```

    }
    state->oneshot_repeat--;
    state->enabled = true;
    pwm_apply_state(pwm, state);
}

```

2.3.5 电机驱动器内核驱动模块

内核驱动的主要目的是通过 Sysfs 接口提供给用户态 QT 程序控制步进电机的功能。因此需要设计 Sysfs 接口，提供对步进电机的控制功能且能在光电开关触发时或定数脉冲输出完成时停止电机并提供反馈给用户态。所有接口在用户态都可以通过文件系统访问。内核通知链（Notifier Chain）用于在定数脉冲输出完成或光电开关触发调用 `sysfs_notify` 以文件修改信号通知用户态。用户态通过监听文件修改信号来获取通知。相比轮询方式，通知链机制可以减少 CPU 资源的浪费，提高系统效率。因此设计了如下接口：

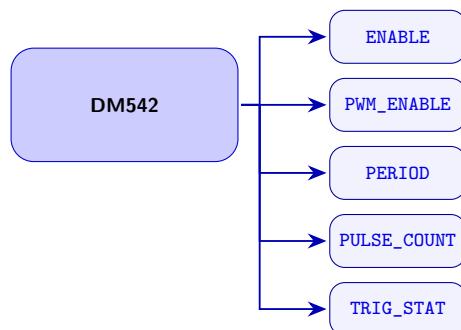


图 12 SysFs 接口设计示意图

- ENABLE 对应步进电机控制器的 ENA 信号输入口，负责电机的上下力。
- PWM_ENABLE 在设置完脉冲数时启动 PWM 输出来驱动电机，依靠通知链机制来实现定数脉冲完成时发送文件修改信号通知用户态。
- PERIOD 设置 PWM 输出周期，从而控制电机运行速度
- PULSE_COUNT 设置 PWM 输出的脉冲数，控制电机运行距离，其正负对应电机正反转。
- TRIG_STAT 为光电开关触发状态，若触发则停止电机并发出文件修改信号通知用户态。

2.3.6 用户态 QT 程序设计

用户态 QT 程序主要负责图形界面显示和用户交互，使用 C++ 编写。程序通过 Sysfs 接口与内核驱动进行通信，实现步进电机的控制和状态查询。同时需要结合海康相机的 SDK 实现相机的连接，采集和参数设置。获取图像信息后再利用 OpenCV 和 GNU Scientific Library (GSL) 进行图像处理和数据分析。用户界面

如图所示：

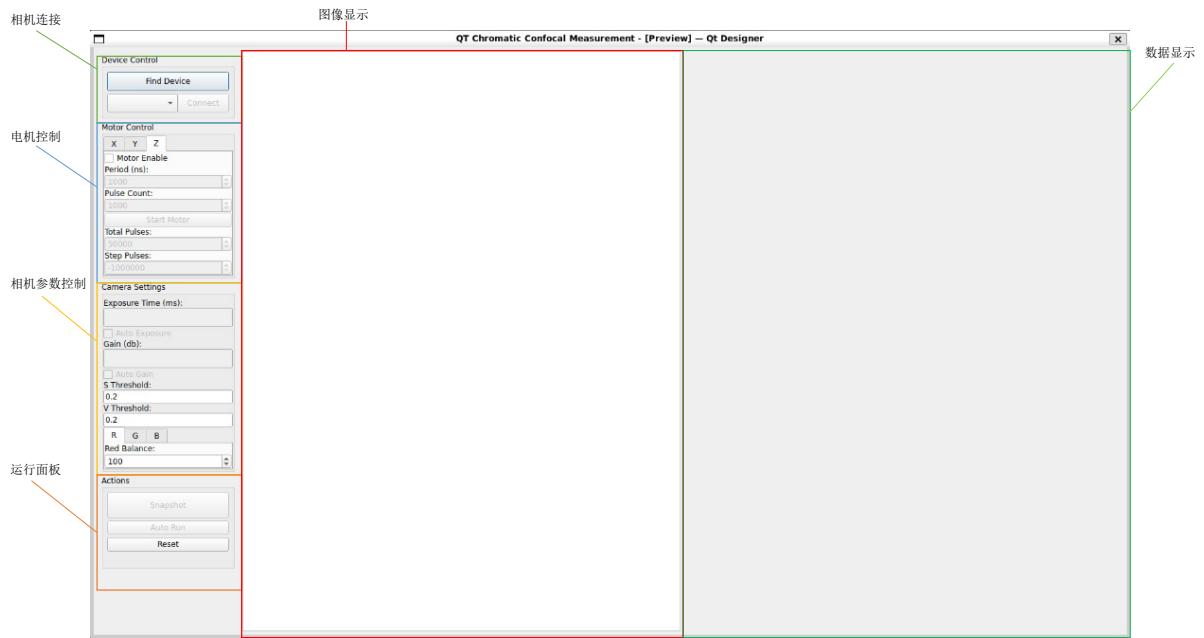


图 13 用户界面

相机连接主要是基于海康 SDK 提供的 API 实现，主要用来寻找接在 USB 上的海康相机并连接。连接相机后相机参数控制面板才会解锁，同时关闭黑电平和自动白平衡避免相机 ISP 算法造成的色差。相机参数面板用来设置相机的曝光时间和增益及 RGB 通道的增益。

电机连接依靠 Sysfs 接口实现，主要是通过文件读写来控制电机的使能，PWM 输出周期和脉冲数设置。通过 Tab 组件切换不同方向的电机进行设置。同时电机页面通过写入 XY 方向的自动运行的总脉冲数和步长设置实际扫描范围。

图像显示区用来显示每次拍照时的光斑图案，数据显示基于 QT DataVisualization 显示形貌扫描的结果。

UI 整体基于 QT Designer 设计，并完成了相关简单联动，如需连接相机才能拍照，电机运动时除了电机控制按钮外电机相关设置不可用等。

2.3.7 数据处理与可视化

根据光谱光谱共焦的原理，数据处理主要目的是获取稳定的色相值，进而可以得出位移值。其中则有两个问题，一是需要有足够稳定的采样数据和处理算法，二是位移和色相值的关系由于不使用光谱仪而是使用彩色相机采集，因此其值不够线性，且标定只能获取到色相值和位移的离散的对应值。

对于前者我们用 C-Mount 转 SMA905 的转接口将光纤口直接照射于工业相机的 CMOS 芯片上，同时相机格式数据使用 RG 12 并关闭自动白平衡和黑电平，这样可以最小化环境和相机算法的影响。对于采集的图像用大津二值法分割出

光斑区域，计算得出光斑中心。由于光斑中心的色相值最稳定，因此只取光斑中心区域的色相平均值作为测量结果。重复这个过程多次，当连续数次采集的色相值变化小于 5% 时，认为采样稳定，取其平均值作为最终结果。这样可以获得比较稳定的色相值。

对于后者则需要对色相值和位移的关系进行标定。由于色相值和位移的关系不够线性，因此需要通过多次采集比对不同位移下的色相值，得到离散的色相值和位移的对应关系。由色相值和位移的对应关系可以得到一个插值函数，使用 GSL 库的插值函数在标定完测量时由色相值获取位移值。通过该插值函数可以将采集到的色相值转换为位移值。

最后当用户点击开始测量时，程序会自动控制电机在 XY 轴上移动指定的步长和总脉冲数，并在每次移动后拍照采集图像，处理图像获取色相值并转换为位移值。最后以 Mesh 网格的形式显示在 QT DataVisualization 中，用户可以通过鼠标滚轮缩放和拖动查看不同角度的三维形貌。同时在运行文件夹下也会创建 CSV 文件保存测量结果，方便后续处理。

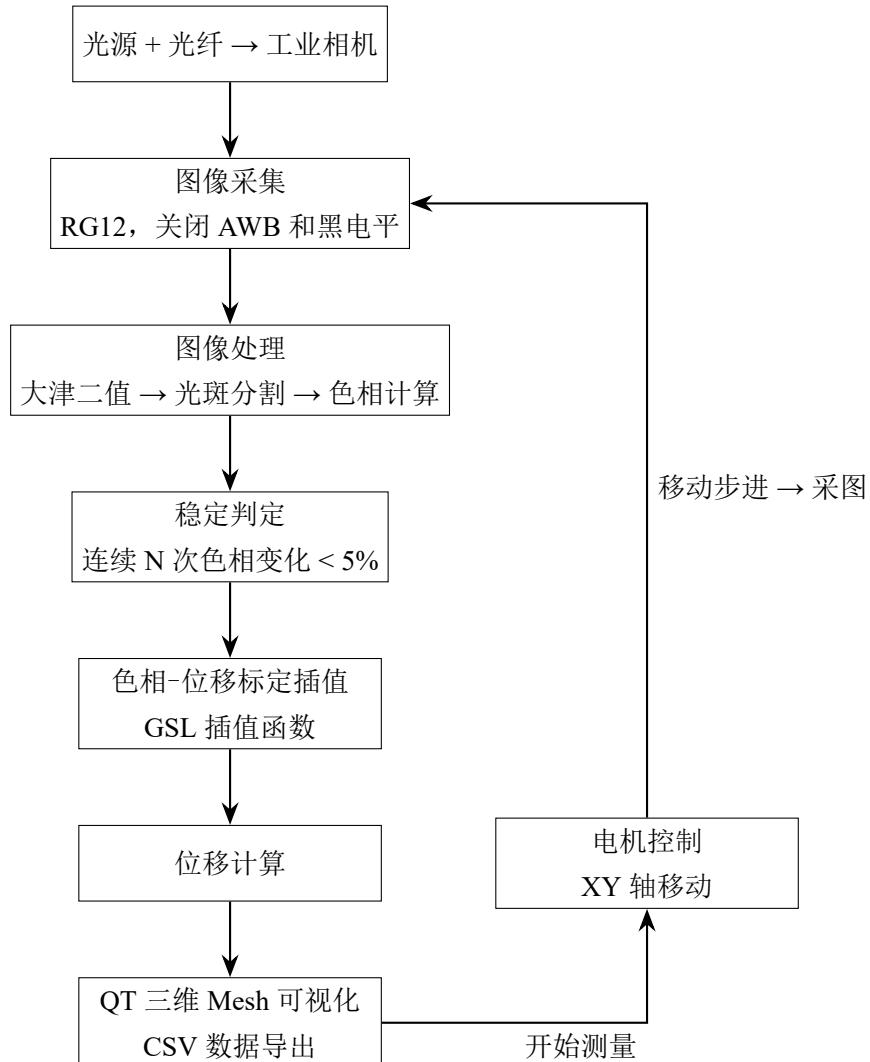


图 14 软件测量流程

第三部分 完成情况及性能参数

3.1 整体介绍

组装后，用一带有孔径 0.2mm，深度 1mm 孔的不锈钢试件测试，孔径结果受自动标注算法影响在 0.01mm 左右浮动，深度在 0.05mm 左右浮动。

3.2 工程成果

3.2.1 机械成果

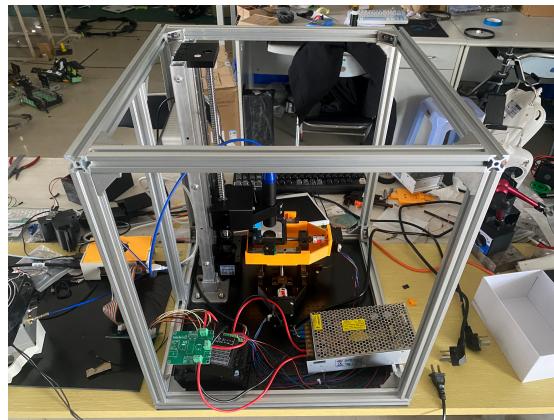


图 15 机械实物

3.2.2 电路成果

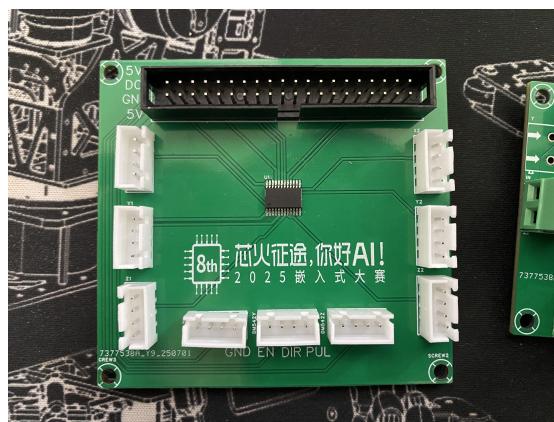


图 16 信号连接板实物

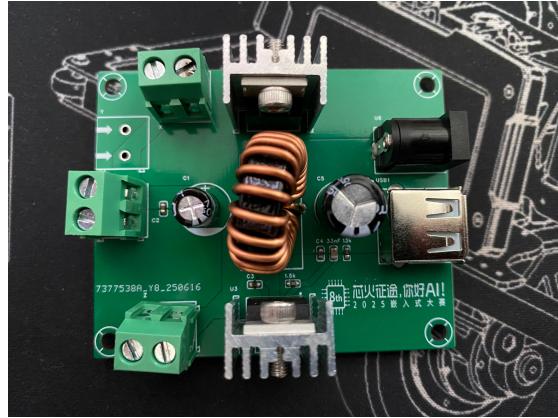


图 17 DCDC 实物

3.2.3 软件成果

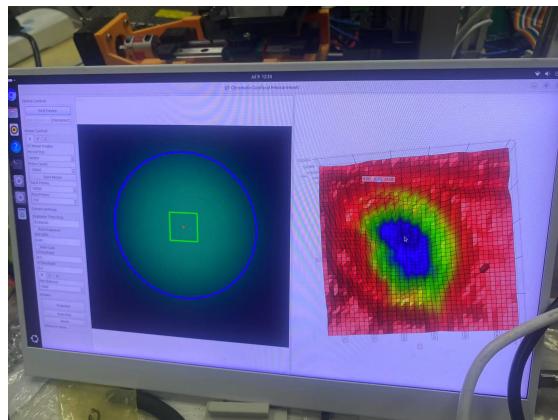


图 18 用户界面

以上为用户界面，可以操作位移台，显示 3 维的扫描结果。

第四部分 总结

4.1 可拓展之处

- a. 现在采用相机方案误差仍较大，后面考虑自组简单的光谱仪或其他分光仪器用来替代工业相机分辨颜色。
- b. 完善机械设计将元件固定在合理的位置，加强强度
- c. 修改电路接口，使其更方便接线
- d. 针对特殊形状优化检测算法和自定义扫描路线

4.2 心得体会

这个项目初期主要是为了实现快速检测深微孔，期间研究过非常多的方案，从立体视觉到白光干涉均有涉猎，花费了不少时间最后比对各种方案，选择了光谱共聚焦。关于各种细节，如机械结构，数据展示形式，设备选型等很多事项又讨论了很久，发现很多原理上很清楚的问题，实际实施的时候总是会有各种问题，不过从里面又学到很多。整个项目涉及到机械，电路，光学，代码队里每个人都学到以前不了解的知识。理解了我们专业常说“光机电算”究竟是什么。不过整个仪器现在看起来还是太粗糙了，后面会继续优化。