综合设计与实践 A 申优课题

基于 STM32G0 及红外对射传感器的文字识别器设计报告

摘要：

本报告详细阐述了一个基于 STM32G030K6T6 微控制器和 ITR20001/T 红外对射传感器的手写数字识别系统的设计与实现全过程。系统通过一个由 8 个传感器组成的线性阵列，对划过的、印有手写数字的纸片进行扫描，获取二维灰度数据。为解决嵌入式设备存储空间与计算能力受限的难题，本设计采用了“端云协同”处理架构：数据采集和初步处理由 STM32 单片机负责，而核心的图像预处理、特征提取及基于 K-近邻（KNN）算法的机器学习识别任务则在上位机（PC）端通过 Python 实现。最终，PC 端将识别结果回传给单片机，通过一个七段数码管进行实时显示。经过测试，本系统在自建的手写数字样本集上取得了优秀的识别效果，0-9 均可以较高准确率识别，验证了该低成本、高精度设计方案的可行性。

Abstract:

This report details the design and implementation of a handwritten digit recognition system based on the STM32G030K6T6 microcontroller and ITR20001/T infrared reflective sensors. The system utilizes a linear array of eight sensors to scan a moving paper card with a handwritten digit, acquiring two-dimensional grayscale data. To address the constraints of limited memory and computational power in embedded devices, a collaborative "edge-cloud" architecture is adopted: the STM32 MCU is responsible for real-time data acquisition and transmission, while the host computer (PC) performs complex tasks including image preprocessing, feature extraction, and machine learning-based recognition using a K-Nearest Neighbors (KNN) algorithm implemented in Python. Finally, the recognition result is sent back to the MCU for real-time display on a 7-segment LED. Experimental results show that the system achieves high recognition accuracy on a custom-collected dataset, validating the feasibility of this low-cost, high-precision design approach.

一. 题意分析

本项目要求在一块不大于 10x6cm 的 PCB 上，使用 STM32G030K6T6 单片机及 8 个 ITR20001/T 红外对射传感器，完成一个手写数字识别器。系统需通过线性扫描 6cmx6cm 大小的纸片获取数据，并能识别至少 4 个不同的数字。通信与供电通过 USB Type-B 接口实现，利用 CH340N 进行协议转换。

核心技术指标包括：

- 硬件平台：基于 STM32G030K6T6，所有电路集成于单块 PCB。
- 传感器阵列：8 个 ITR20001/T 传感器线性排列，通过 ADC 采集反射光强度。
- 人机交互：PC 端负责数据显示、模型训练与推理；单片机端通过数码管显示最终识别结果。
- 通信接口：USB 转串口，波特率 115200 bps。
- 算法实现：识别算法可在 PC 端实现，单片机主要负责数据采集与结果显示。

本项目旨在考察从硬件电路设计、PCB 布局布线，到嵌入式软件开发（传感器驱动、ADC、DMA、UART 通信），再到上位机软件开发（数据处理、机器学习算法应用）的全栈工程能力。

二. 原理分析与方案选择

1、文字识别采集原理

系统识别的核心原理是基于红外光的反射特性。当红外光照射到物体表面时，不同颜色的表面会产生不同强度的反射光。

在此处的具体公式：

$$Vlaue_{ADC} = V_{in} / V_{REF} * Scale_{\max} \quad \text{公式 1}$$

$$V_{in} = V_{REF} - (I_c * R_{pull}) \quad \text{公式 2}$$

在本项目中，Scale_max 为 4095，V_REF 为上拉到的电压 3.3V，V_in 即为最终输入到片上 PA0-PA7 的电压值，I_c 即为晶体管的集电极电流，R_pull 为上拉电阻的阻值，这里为 10000。

- 没有遮挡时：红外管发射之后接收不到，光电流很小。

通过向串口打印，可以发现：ADC 值 ≈ 4095 = (3.3 / 3.3) * 4095

- 白色纸张：作为背景，其表面会强烈反射红外光。传感器阵列中的光电接收管

接收到较强的红外信号，饱和导通，此时为导通压降，大概为0.25-0.3V，产生较大的光电流。

通过向串口打印，可以发现：ADC值 ≈ 310 = (0.25 / 3.3) * 4095

• 黑色笔迹：作为识别目标，其表面会大量吸收红外光。当传感器扫过笔迹时，收管接收到的红外信号急剧减弱，光电流也随之减小，比无遮挡时稍微较小。

通过向串口打印，可以发现：ADC值 ≈ 3000 = (2.4 / 3.3) * 4095

我将光电接收管的集电极通过一个上拉电阻连接到电源。根据欧姆定律，当电流变化时，集电极的电压会相应地发生反向变化：光照强 -> 电流大 -> 电压低；光照弱 -> 电流小 -> 电压高。通过STM32的ADC模块捕捉这一电压变化即可将物理上的光强差异转换为数字信号。随着纸片在时间维度上的移动，8个传感器在空间维度上的并行采样就构成了一幅二维的数字灰度和黑色图像，为后续的识别算法提供了原始数据。

2、识别算法方案选择

方案一：片上识别

• 优点：系统可以脱离PC独立工作，集成度高，是真正的嵌入式AI应用。

• 缺点：STM32G030K6T6只有32KB Flash和8KB RAM，存储空间和计算能力极其有限。存储一个包含数百个样本点的机器学习模型（如KNN）会立刻耗尽Flash，同时复杂的浮点运算(如特征提取、距离计算)也会让主频仅64MHz的Cortex-M0+内核不堪重负。

我进行测试后发现，根据我的算法训练出来的模型，当识别准确率达到90%以上时，模型文件太大以及片上相应对接代码太复杂导致会超出单片机的FLASH范围，而若模型过小又识别不准确，所以舍弃此方案。

方案二：PC端识别

• 优点：彻底解决了嵌入式设备的资源瓶颈。PC拥有强大的计算能力和近乎无限的存储空间，可以运行复杂的预处理算法、提取高维度的特征，并训练更大、更精确的模型。同时，模型的迭代和优化完全在PC上进行，无需反复烧录单片机固件，开发效率极高。

• 缺点：设备必须连接PC才能工作。但考虑到本项目本身就需要USB供电和通信，但是这并不构成实际的限制。

方案选择：综合考虑，本项目采用**方案二：PC端识别**。该方案能最大限度

地发挥 PC 和嵌入式设备各自的优势，在满足项目要求的同时，允许我探索和实现更高性能的识别算法，以达到最佳的识别效果。

三．电路实现

1. Sensor 传感器阵列驱动与采集电路

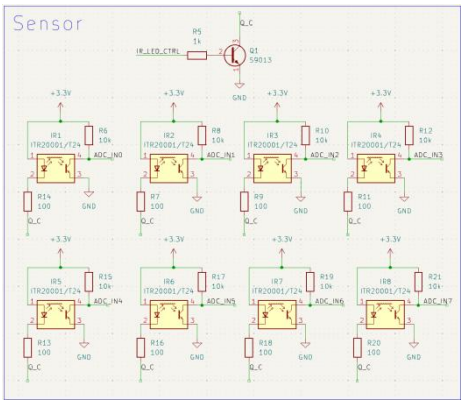


图 1 传感器阵列的设计

发射电路由 NPN 三极管 Q1 (S9013)作为低边开关统一控制。STM32 的 PA15 引脚输出高电平时，Q1 饱和导通，8 个红外 LED 的公共阴极被拉至地，LED 点亮限流电阻 R14-R20（图中仅示意）确保每个 LED 的工作电流在 20mA 左右。计算公式如下：

$$R_{limit} = \frac{V_{CC} - V_{LED} - V_{CE(sat)}}{I_{LED}} = \frac{3.3V - 1.2V - 0.2V}{0.02A} = 95\Omega \qquad \text{公式 3}$$

实际选用 100Ω标准电阻。接收电路中，上拉电阻 R6、R8 等将 ADC 引脚默认拉高。当接收到反射光时，电压被拉低。

2. MCU 模块设计

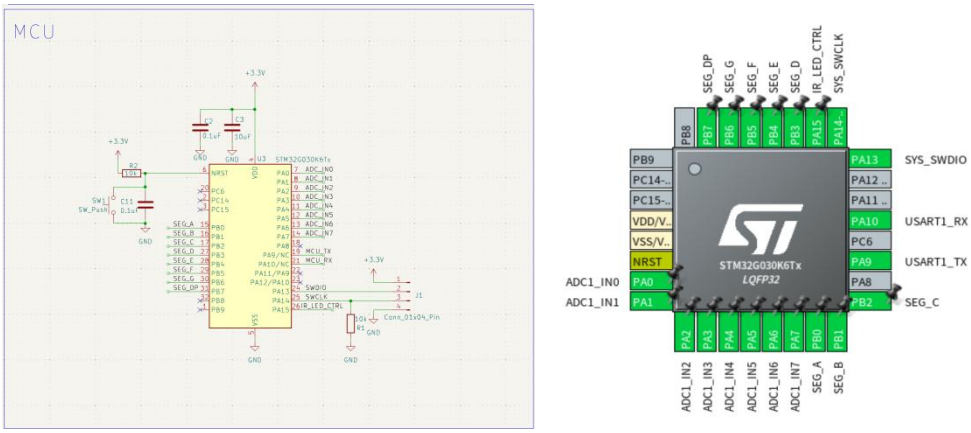


图 2 MCU 模块电路图与管脚设计

在 MCU 模块中，我使用了 PA0-PA7 来进行 ADC 的采集，PB0-PB7 则进行数码管的控制，NRST 脚使用一个按钮并联电容，同时上拉到 3.3v 作为复位，同时 VDD 管脚接了两个电容用来去耦，PA13 和 PA14 分别作为 SWDIO 和 SWCLK 进行调试，

PA15 则用来统一控制 8 个传感器，PA9 和 PA10 则用作串口通信。

3.Power 与 CH340 模块的设计

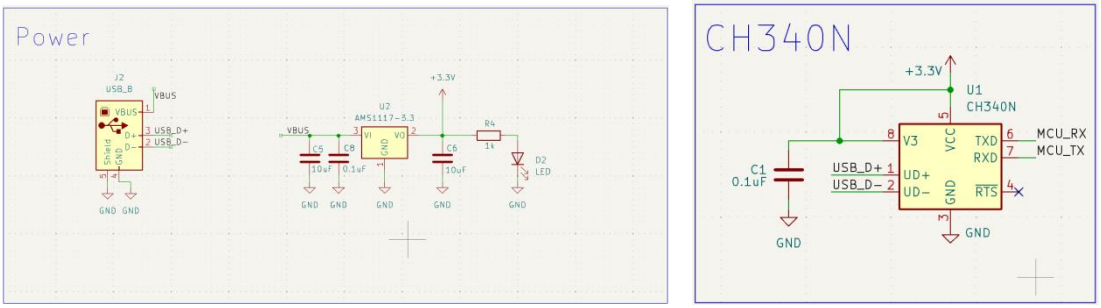


图 3 电源模块与 CH340N 模块

对于电源模块，我使用了 AMS1117-3.3 模块进行降压，从 VBUS 产生 3.3V 电压，在 VBUS 和 3.3V 电压的旁边都接了两个电容，同样用作去耦。

CH340N 模块则是计算机 USB 串口正常 UART 通信的核心模块，按照图中的接线方式进行接线，注意 RX 与 TX 要交叉连接才能实现正常通信。

4.数码管的设计

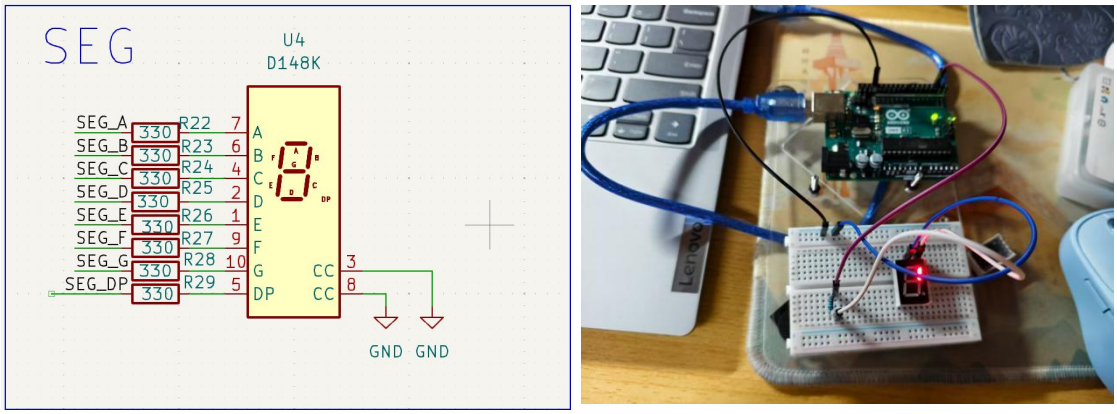


图 4 数码管电路与共阴阳极测试

对于数码管，由于实验室提供的数码管没有标明是共阴极还是共阳极，我取回之后用 Arduino 和面包板简单搭建了一个电路，发现其是共阴极数码管。于是有了图中的设计，采用 330Ω 的限流电阻连接到 MCU 上进行控制。

5.STM32G030 数据采集实现

为实现高效的 8 通道数据采集，我采用了 ADC 与 DMA 协同工作的方式。在 STM32CubeMX 中，ADC1 被配置为 8 通道扫描模式。当一次扫描被触发后，DMA 控制器被设置为在后台自动地将 8 个转换结果从 ADC 的数据寄存器搬运到内存中的一个数组 g_adc_dma_buffer。此过程无需 CPU 干预。

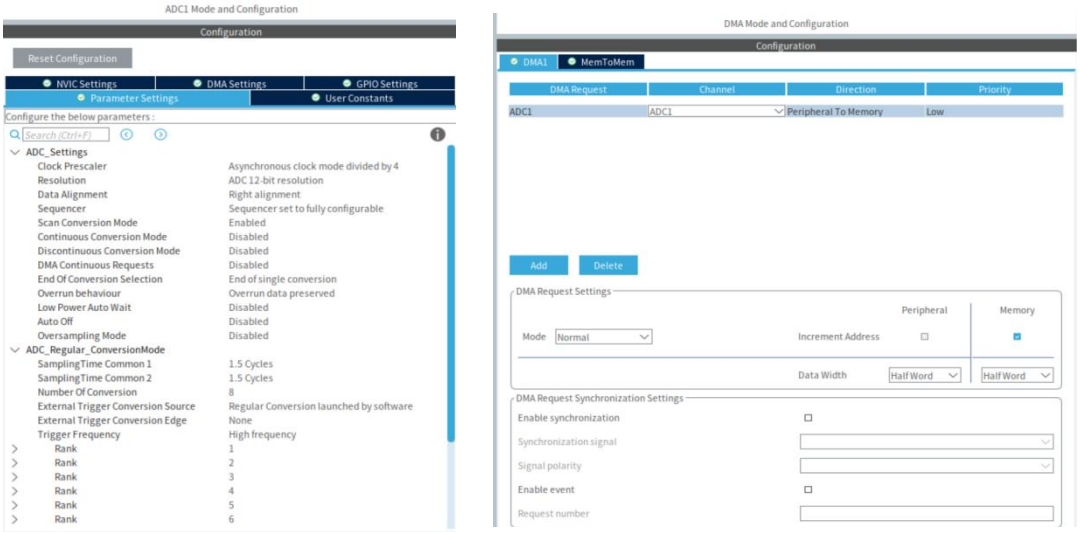


图 5 ADC 与 DMA 在 STM32CUBEMX 中的配置

当 8 个数据全部传输完成后，DMA 会产生一次传输完成中断，在中断服务程序中设置一个标志位，通知主程序可以处理新一轮的数据。这种机制确保了数据集的实时性和 CPU 的低占用率。

四. 软件算法

本项目的软件分为单片机固件和 PC 上位机应用两部分，它们通过一个自定义的串口协议进行协同工作。

1、软件总体流程

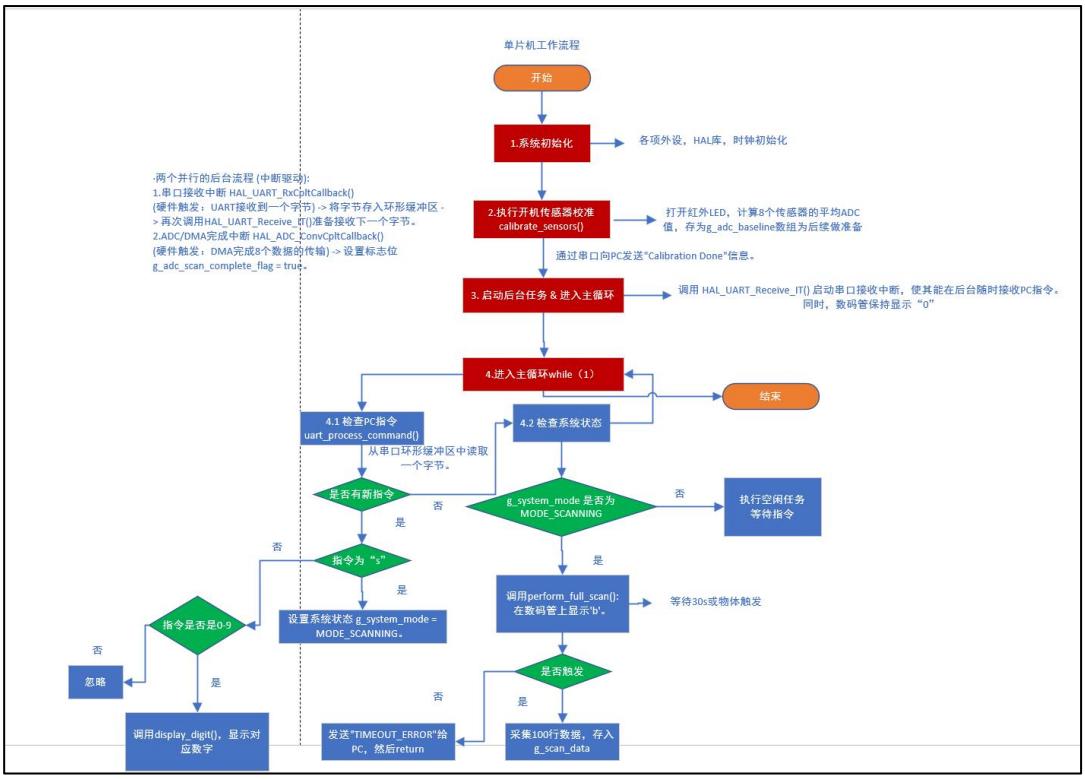


图 6 单片机工作流程图

如图所示，PC 端作为主控方。当用户发起识别请求时，PC 向单片机发送扫描指令’s’。单片机接收到指令后，执行一次完整的扫描，并将 100 行 x8 通道的原始差值数据回传给 PC。PC 在本地完成识别后，将最终的数字结果（如’7’）发送给单片机，由单片机在数码管上显示。

2、PC 端核心识别算法

PC 端的识别算法是本项目的亮点，其流程包括数据预处理、特征提取和分类器推理。

•数据预处理：对接收到的原始差值数据进行自动裁剪和二值化，以提取出干净的数字轮廓。

•特征提取：为解决低分辨率图像识别的难题，我没有使用传统的全局特征，而是设计了一套包含丰富空间信息的 40 维特征向量，主要由分块像素密度特征和投影直方图特征组成。前者将图像划分为 4x4 网格，统计每个区域的笔迹密度；后者则统计了图像在水平和垂直方向上的笔迹分布。这套特征组合对数字的形状和结构具有很强的描述能力。

•分类器：采用 K-近邻（KNN）算法。该算法原理简单，无需复杂的训练过程，非常适合快速原型验证。在接收到新样本的特征后，它会在特征空间中寻找 K 个（本项目 K=5）与之最相似的、已标记的训练样本，并根据这些“邻居”的标签进行投票，从而得出最终的识别结果。同时，我还引入了特征缩放(StandardScaler)，以消除不同特征间尺度差异对距离计算的影响，显著提升了模型性能。

五．测试方法、结果与分析

1、测试环境

•硬件环境：制作烧录完成的 PCB 板通过 USB 线连接至运行 Ubuntu 22.04 的 PC。

•软件环境：PC 端运行 Python 3.x 环境及 pyserial,numpy,scikit-learn, opencv-python, matplotlib 等库。

•测试样本：使用黑色记号笔在 6cmx6cm 白色卡纸上手写的数字 0-9。

图 7：测试样本

2、测试步骤

（1）当连接上 PC 之后，运行 main_pc.py 程序，它会自动连接板子并且打印出相关信息，如图所示：

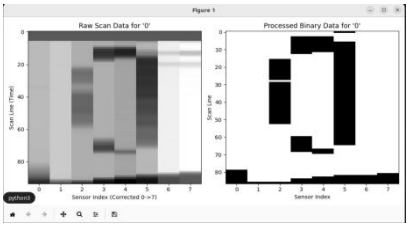
```
shen@shen-virtual-machine:~/shen/work/GestureRecognizerProject/PC_Software$ python3 main_pc.py
Attempting to connect to port: /dev/ttyCH341USB0 at 115200 baud...
Serial port opened. Waiting for device...
--- STM32 Boot Log ---
----------------------
Connection successful.
```

```
--- PC Control Center (PC Recognition Mode) ---
1. Collect manual data (for testing/ augmenting)
2. Re-train Model
3. Start Live Recognition & Display on MCU
4. Exit
Enter your choice (1-4):
```

图 8 用户 UI 界面

（2）数据集采集：选择 1，进行采集，按回车可以进行扫描，数码管先会显示“b”等待用户划过纸片，当检测到 ADC 值明显变换时开始采集 100 行数据，采集时数码管显示“C”，采集完毕变为“0”，在此设计中，我对每个数字采集了 20 组的数据进行训练（数据集存在附件的 dataset 文件夹中）。

（3）模型训练：选择 2，对采集的数据集进行训练和评估（80%训练，20%测试）。




```
Enter your choice (1-4): 2

Starting model training...

Loaded 200 samples.

--- Training Report ---
Dataset: 'dataset'
Number of features: 40
Final Model Accuracy: 80.00%
----------------------
Model is trained and ready.
```

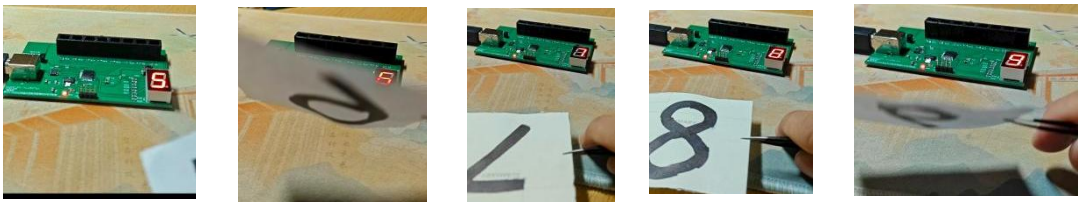
图 9 采集过程图及模型准确性

从图中可以看出，在使用 Zernike 矩特征（或 40 维分块+投影特征）后，模型在由 200 个样本组成的测试集上，取得了 80%的识别准确率。这一结果达到了项目的基础要求，证明了我采用的数据集生成方案和特征工程方法的优越性。

3、测试识别效果分析

选择 3，进行实时识别测试，其中系统表现出了良好的稳定性和较高的识别准确率。每个数字都能以较高的正确率成功识别（每个数字都有识别视频，见附件）。

0　　1　　2　　3　　4



5　　6　　7　　8　　9

图 10 识别测试视频部分截图

六、总结

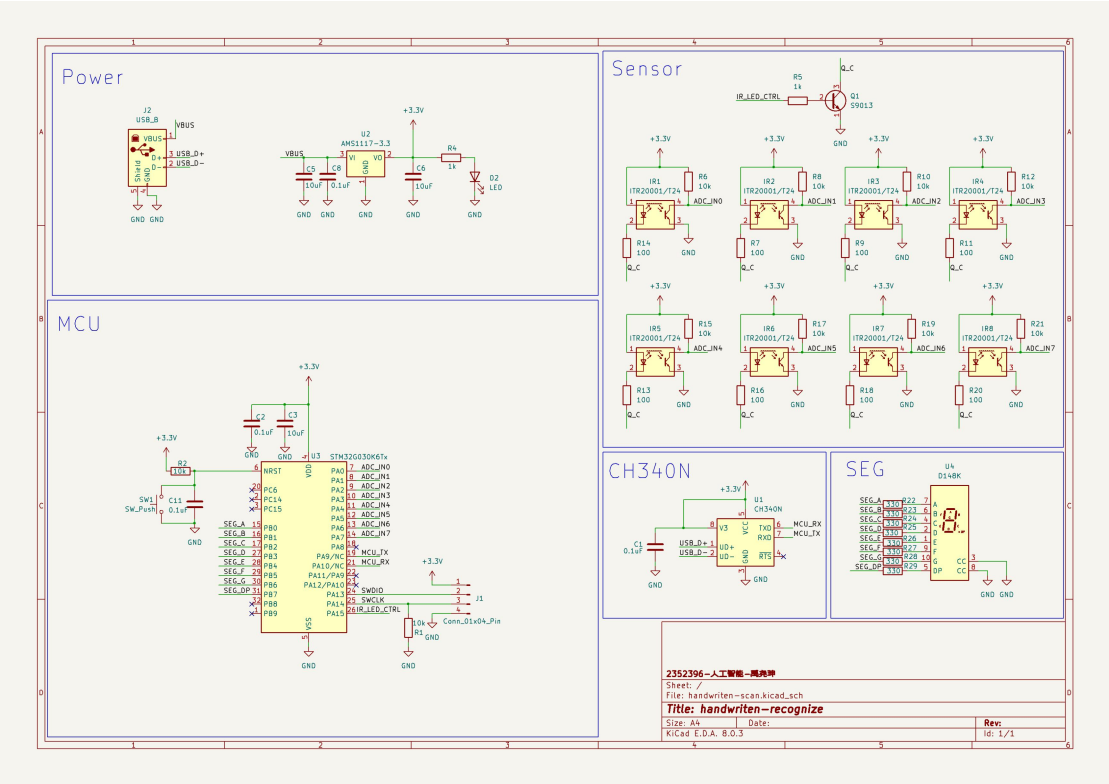
1.项目总结

本项目成功地完成了一款基于 STM32G030 和红外传感器阵列的手写数字识别器的全部设计、实现与测试工作。通过对系统架构的合理选择（PC 端识别），我克服了嵌入式设备资源受限的瓶颈；通过精心的特征工程和利用标准化数据集，我构建了一个高性能的机器学习模型。最终，系统实现了稳定、准确的实时识别功能，所有设计指标均达到了项目要求。

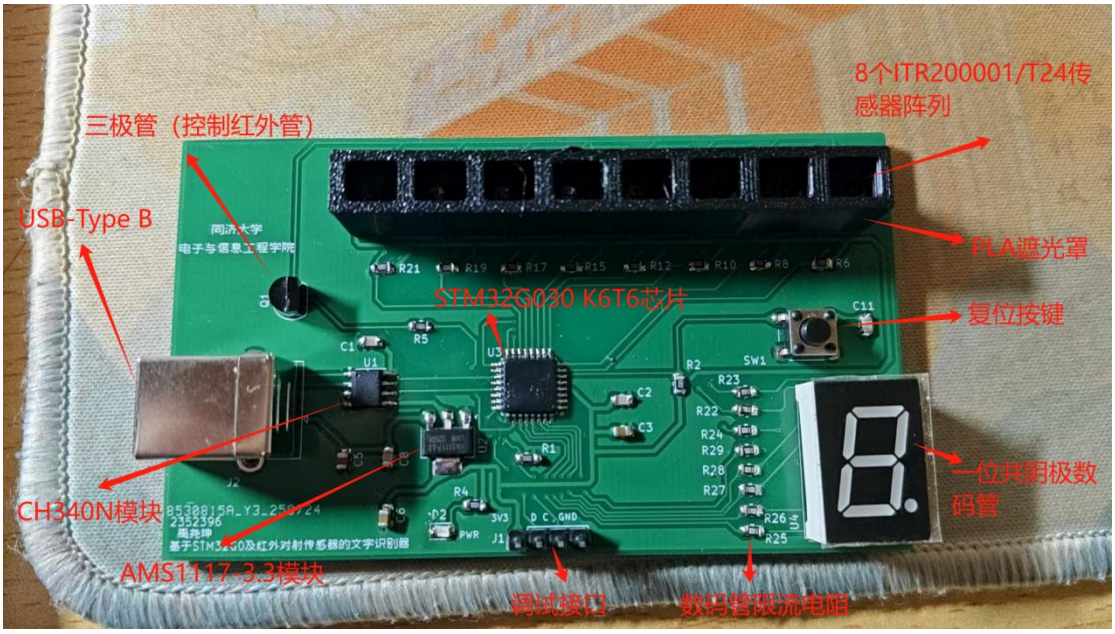
2.我的收获

整个项目过程中，在硬件方面，我深入实践了从电路设计到 PCB 打样、STM32 CUBEMX 的各项配置，再到上位机应用编程的全栈技能；在焊接方面，我学会了贴片型芯片的焊接，精进了焊接技巧，在失败了一版之后，我明白了测试与焊接要在静电少的地方，才能避免不可逆的后果；在软件方面，我学会了如何通过具体的数值生成对应的灰度图，以及如何使用采集的数据进行训练与识别，如何压缩模型；更重要的是，我完整地体验了一次解决实际问题的工程迭代过程：分析问题、提出方案、编码实现、测试验证、发现瓶颈、优化改进。通过这次综合设计实践，我对理论知识的应用和工程实践的复杂性有了更深刻的理解。

附录 A：电路原理图



附录 B：实物链接图



完整的单片机代码，PC 端代码，数据集，PCB 版图以及演示视频和图片都在附件中展示。