移远通信培训周报

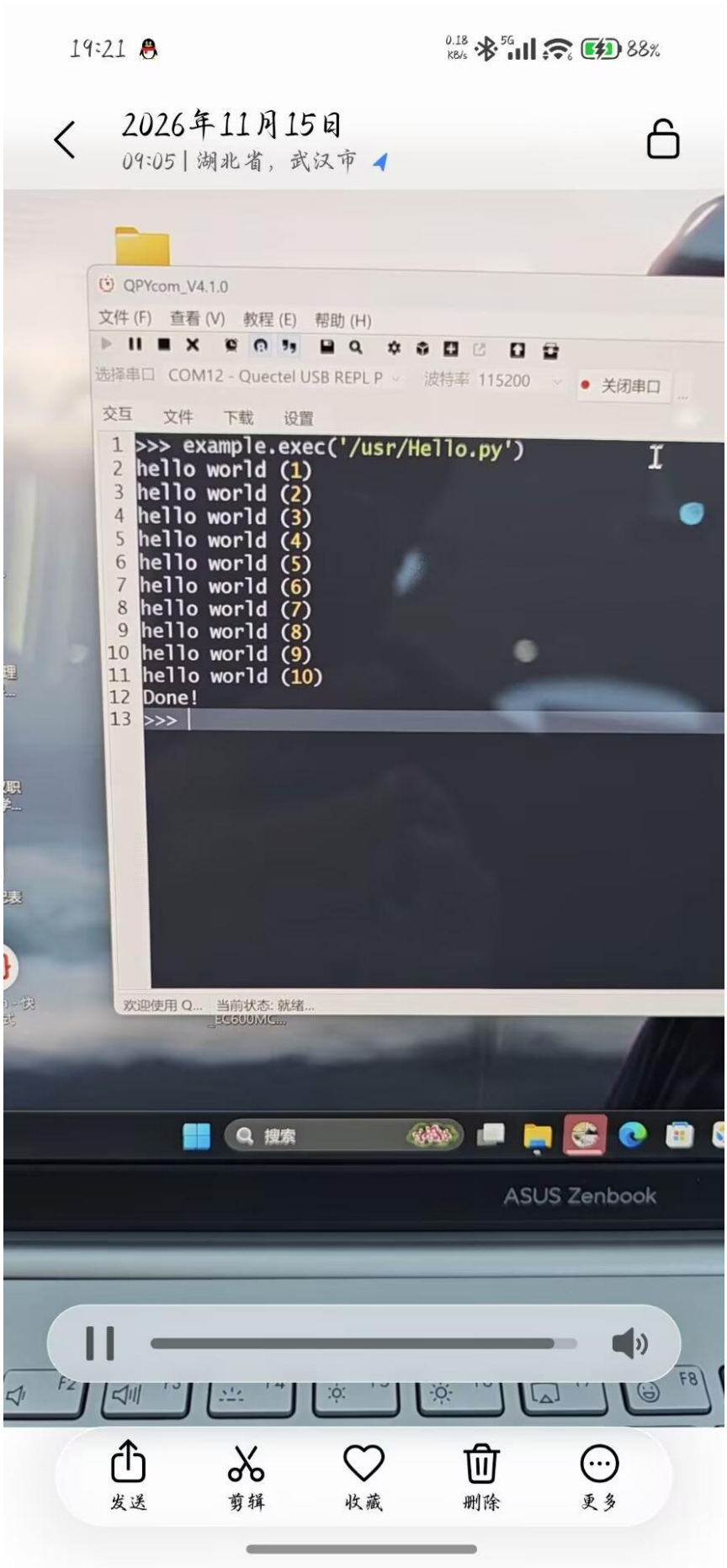
姓名： 徐昕　　学号： 2601406119　　时间：2026.11.19

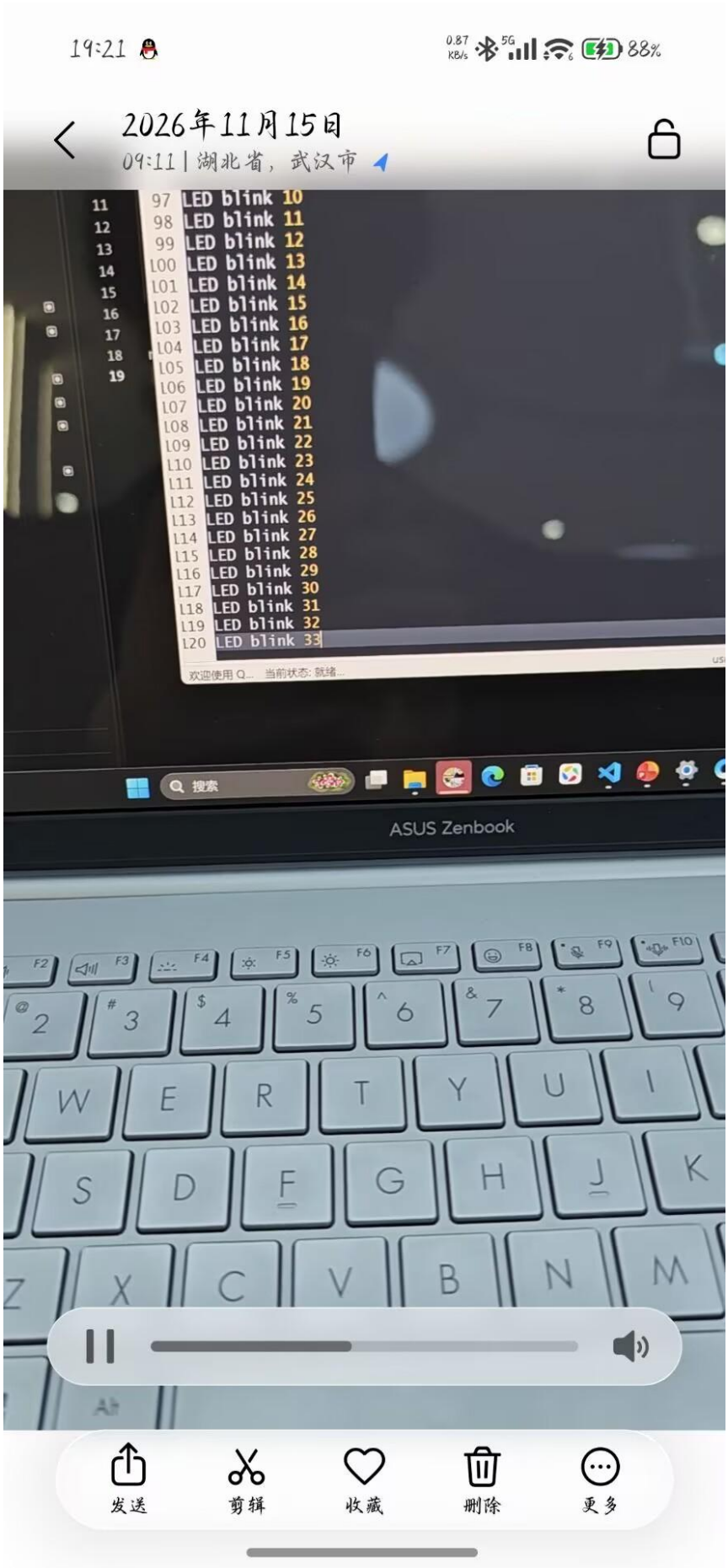
任务完成情况	任务 1：周期性打印 Hello World 完成度：100% 我的工作成果：完成了 VSCode 和 QuecPython 开发环境的配置，打通 AI 辅助开发流程，并使用 AI 生成全部基础代码。已在 EC600M 开发板上完成 Hello World 循环打印，共打印 10 次，每隔 2 秒输出一次，最后打印“Done!”。 任务 2：按键触发定时 LED 闪烁 完成度：100% 我的工作成果：完成按键、外部中断、软件定时器和 LED 定时翻转的联调。按下 S5（GPIO12）后，通过下降沿中断启动 50ms 消抖定时器；确认按下后，LED 每 200ms 翻转一次，10 秒后自动停止并重新使能按键中断。 任务 3：AHT20 温湿度采集与 ST7789 显示 完成度：100% 我的工作成果：完成 AHT20 I2C 驱动和温湿度公式转换，配置 I2C1，传感器地址为 0x38，转换完成后读取 6 字节原始数据并换算为摄氏度和相对湿度。同时完成 ST7789 240×240 显示屏初始化，循环在 LCD 上显示温度和湿度。 任务 4：智能环境监控综合系统 完成度：100% 我的工作成果：完成 GPIO、I2C、ADC、PWM、LCD、网络和 TTS 的综合联调。系统可采集 AHT20 温湿度、GL5528 光照，控制 LED 状态、SG90 窗户角度和异常报警；支持 S5 切换页面、S4 切换模式及三档开窗，并实现数据统计、低功耗、后台天气和语音播报等扩展功能。
知识总结	1.1 开发环境与 AI 辅助工具链搭建 本周完成 VSCode 及 AI 插件配置，通过 GitHub Copilot 建立代码生成、解释和调试能力，并注册 IoTcoder 账号、配置 MCP 服务器与相关 Skill。借助项目规范、官方文档和设备接口资料，我形成了“需求描述—代码生成—设备联调—问题分析”的 AI 辅助开发流程。 实践中，AI 提高了代码编写效率，也帮助梳理硬件引脚、模块接口和异常处理，但生成代码仍需结合官方文档检查参数、引脚和固件限制。最

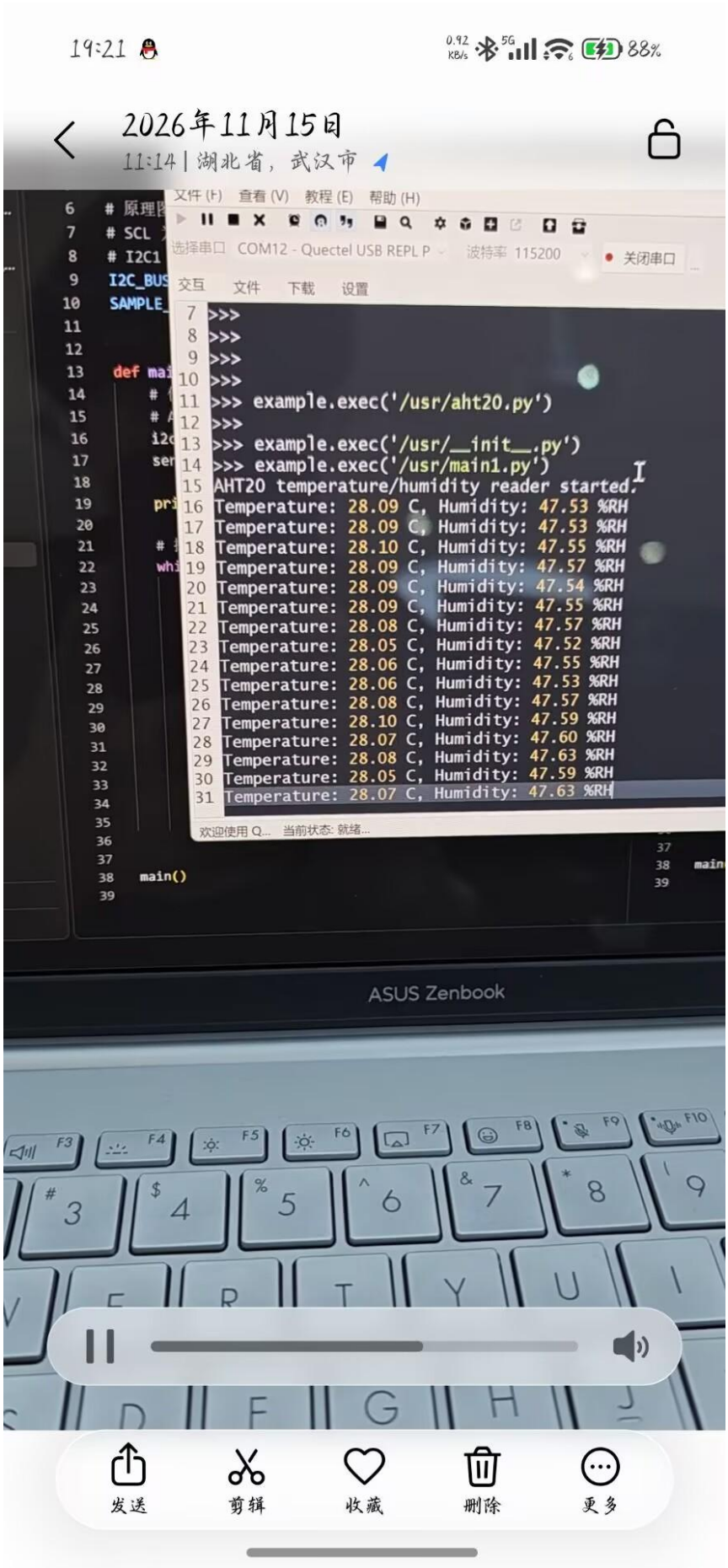
	终还要通过串口日志和硬件现象验证，不能只依赖代码是否能够编译。 1.2 QuecPython 基础编程与设备联调测试 首先完成周期性 Hello World 项目，使用 utime.sleep 实现两秒间隔。PC 端运行普通 Python 脚本与设备运行 QuecPython 不同，因此需要将 main.py 上传到开发板/usr 目录，再通过 QPYcom 运行并观察串口输出。 按键实验使用 GPIO24 控制 LED，GPIO12 连接按键并采用下降沿中断。中断回调禁用 ExtInt 并启动 50ms 单次消抖定时器；定时器确认按下后，Timer0 每 200ms 翻转 LED，Timer1 在 10 秒后停止闪烁、熄灭 LED 并重新使能按键。把中断、消抖和闪烁分开，使中断回调保持简短。 1.3 AHT20、LCD 及综合环境监控 AHT20 通过 I2C1 通信，SCL 为引脚 57，SDA 为引脚 56，地址为 0x38。发送测量命令后等待至少 80ms，再读取 6 字节原始数据并换算温湿度。ST7789 通过 machine.LCD 初始化，分辨率为 240×240；界面中的汉字使用 16×16 字模，数字和英文使用 8×16ASCII 字库。 智能环境监控系统整合了 AHT20、GL5528、SG90、LED、LCD、网络和 TTS。LED 按异常报警、温度警告、光照联动的优先级刷新；湿度超过阈值时舵机调整窗户角度；温度超过 35° C 或湿度超过 90%时进入报警。由于固件 PWM 参数只接受整数，舵机驱动依次尝试 1us、10us 和 ms 脉宽模式。 按键中断只设置事件标志，主循环进行 300ms 消抖后切换页面或模式；网络和天气请求放入独立线程；TTS 异步播放。系统还累计温湿度统计，在环境稳定且 20 秒内无按键时将采样周期由 2 秒降为 5 秒。

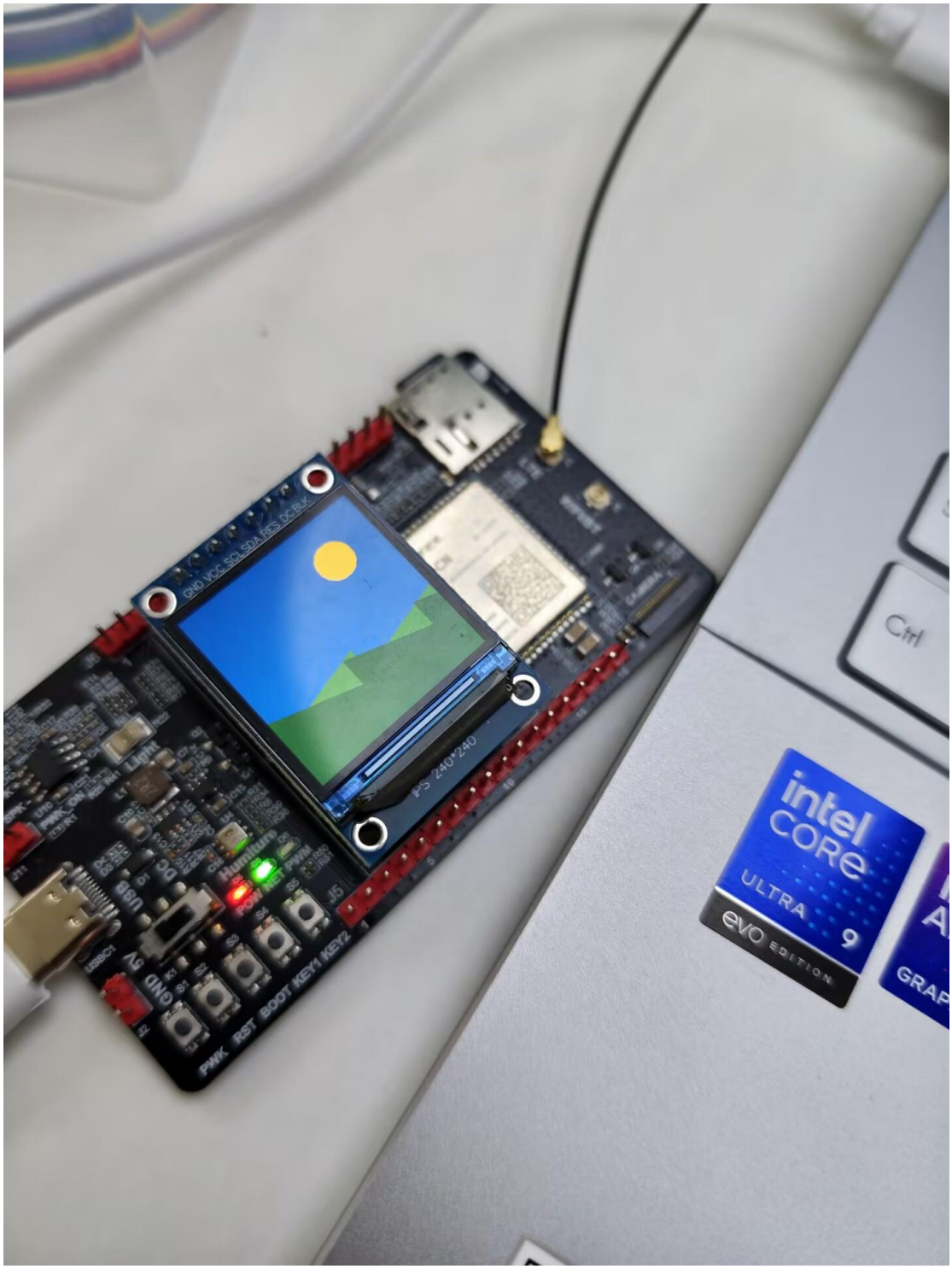
学习心得	这周主要完成了环境搭建、基础实验、传感器联调和综合系统开发，最大收获是理解了软硬件开发流程。 2.1 错误总结 开始时习惯在 VSCode 直接运行代码，发现 PC 端找不到硬件模块。确认后才明白 QuecPython 运行在 EC600M 固件环境中，main.py 和驱动文件应上传到开发板/usr 目录，再通过串口观察运行状态。 按键实验中，若把业务逻辑直接写入 ISR，虽然响应快，但会增加中断占用时间，还可能因抖动重复触发。改为 ISR 只禁用中断并启动定时器消抖后，逻辑更稳定。 AHT20 联调时，我曾因地址、I2C 引脚或等待时间不正确而得到异常数据；舵机联调时浮点脉宽参数会触发固件报错。通过串口日志、官方示例和整数脉宽单位逐步定位问题。LCD 项目还让我认识到，UTF-8 字符串按字节存储，增加汉字必须添加对应字模。 2.2 学习总结 Hello World 让我掌握设备入口和 utime 等待，按键实验让我理解 Pin、ExtInt、Timer 和 osTimer 的协作，环境监控让我逐步掌握 I2C、ADC、PWM、LCD、网络线程和 TTS。 项目不应一开始就堆在 main.py 中。配置、传感器驱动、显示驱动、舵机驱动、网络服务和核心业务应分别建模块，方便定位错误和复用代码。同一颗 LED 会同时承担异常、温度和光照状态，因此应由统一控制器按优先级刷新，不能让各模块直接争用引脚。 中断回调、采样、网络请求、天气解析和 TTS 播放都不能占用过长时间。只置标志、独立线程、定时器轮询和异步播报能够保证按键、采样和界面稳定。下一步我会继续测试断网、传感器异常、LCD 初始化失败等情况下的恢复能力。

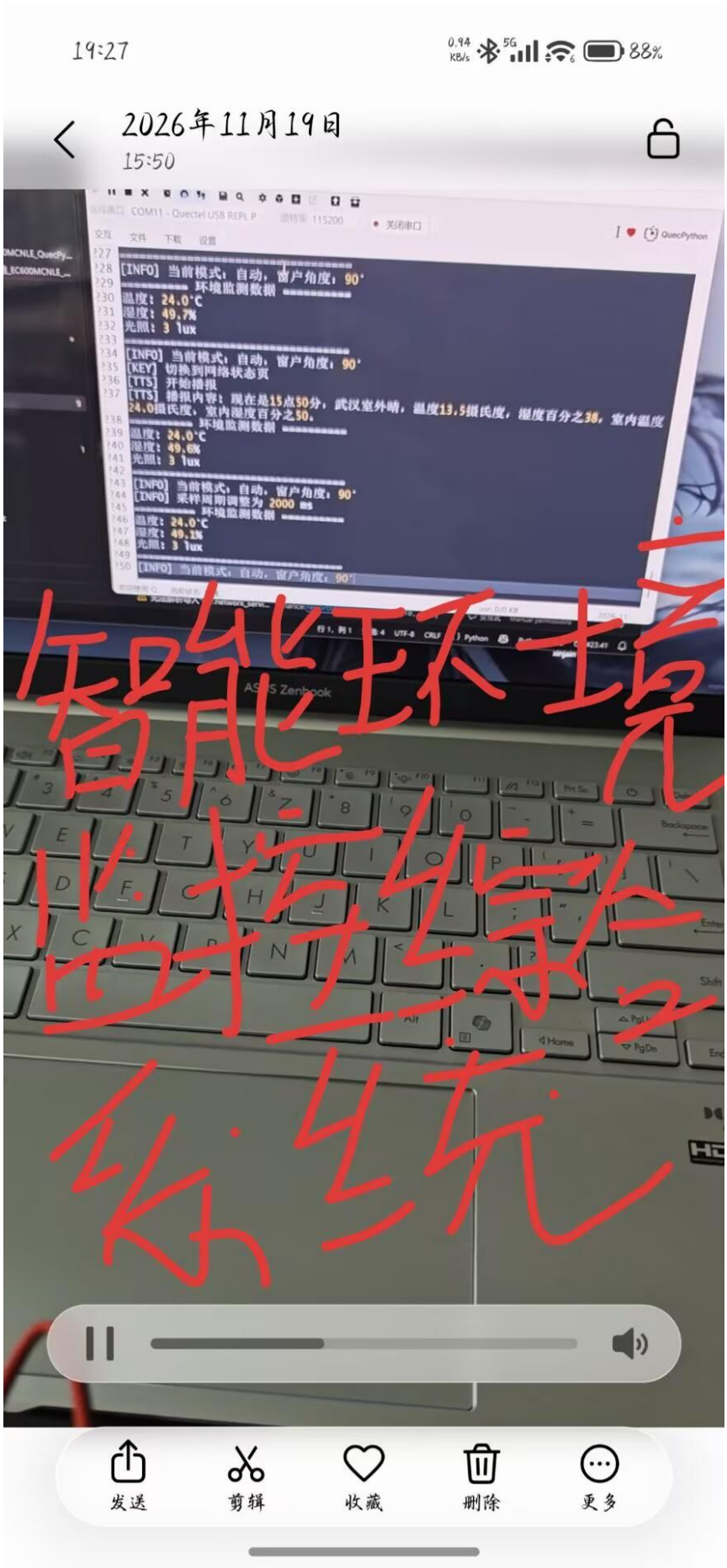
课后思考（作业）	思考题 1：为什么 ISR 中要先 exti.disable()再启动 osTimer，而不是直接执行业务逻辑？ 答：先禁用 ExtInt 可阻止按键抖动在按下过程中连续触发中断；再启动一次性 osTimer，在消抖时间结束后读取稳定电平并执行业务逻辑。ISR 中直接翻转 LED、启动定时器或访问复杂外设，会增加中断占用时间，影响采样和定时器。中断回调用于保存事件或启动定时器，不宜承载耗时业务。 思考题 2：debounce_cb 中为什么要重新读取引脚电平？直接用 ISR 的触发沿信息不行吗？ 答：触发沿只说明当时检测到下降沿，不能保证消抖结束后按键仍处于按下状态。机械抖动可能让引脚再次弹起或波动，因此消抖回调要重新读取电平。确认为低电平才启动 LED 和超时定时器，否则只重新使能中断等待下一次有效下降沿。 思考题 3：为什么中断回调中只置标志，页面切换和网络操作要放到主循环？ 答：中断回调应保持实时性和简短性。LCD 刷新、天气请求、数据解析和 TTS 播放都可能耗时，不宜直接放入 ISR，否则会影响采样、定时器和其他外设。主循环轮询标志后先完成消抖，再刷新页面或处理网络请求，逻辑更安全，也便于异常恢复。 思考题 4：综合系统中为什么要给 LED 和舵机设计优先级？ 答：LED 同时承担报警、温度和光照提示，舵机同时承担湿度联动和异常摆动。若模块直接覆盖状态，会出现报警结束后状态残留或正常控制打断报警等问题。核心逻辑应按“异常报警—温度警告—光照状态”刷新 LED；舵机报警时优先摆动，恢复后再按湿度或手动模式控制角度。


19:21
0.18 KB/s
88%
2026年11月15日
09:05 | 湖北省，武汉市
QPYcom_V4.1.0
文件 (F) 查看 (V) 教程 (E) 帮助 (H)
选择串口 COM12 - Quectel USB REPL P
波特率 115200
关闭串口
交互 文件 下载 设置
>>> example.exec('/usr/Hello.py')
hello world (1)
hello world (2)
hello world (3)
hello world (4)
hello world (5)
hello world (6)
hello world (7)
hello world (8)
hello world (9)
hello world (10)
Done!
>>>
欢迎使用 Q...
当前状态: 就绪...
搜索
ASUS Zenbook
发送
剪辑
收藏
删除
更多


19:21
88%
2026年11月15日
09:11 | 湖北省，武汉市
LED blink 10
LED blink 11
LED blink 12
LED blink 13
LED blink 14
LED blink 15
LED blink 16
LED blink 17
LED blink 18
LED blink 19
LED blink 20
LED blink 21
LED blink 22
LED blink 23
LED blink 24
LED blink 25
LED blink 26
LED blink 27
LED blink 28
LED blink 29
LED blink 30
LED blink 31
LED blink 32
LED blink 33
ASUS Zenbook
发送
剪辑
收藏
删除
更多


19:21
2026年11月15日
11:14 | 湖北省，武汉市
文件 (F) 查看 (V) 教程 (E) 帮助 (H)
选择串口 COM12 - Quectel USB REPL P
波特率 115200
关闭串口
交互 文件 下载 设置
>>> example.exec('/usr/aht20.py')
>>> example.exec('/usr/__init__.py')
>>> example.exec('/usr/main1.py')
AHT20 temperature/humidity reader started.
Temperature: 28.09 C, Humidity: 47.53 %RH
Temperature: 28.09 C, Humidity: 47.53 %RH
Temperature: 28.10 C, Humidity: 47.55 %RH
Temperature: 28.09 C, Humidity: 47.57 %RH
Temperature: 28.09 C, Humidity: 47.54 %RH
Temperature: 28.09 C, Humidity: 47.55 %RH
Temperature: 28.08 C, Humidity: 47.57 %RH
Temperature: 28.05 C, Humidity: 47.52 %RH
Temperature: 28.06 C, Humidity: 47.55 %RH
Temperature: 28.06 C, Humidity: 47.53 %RH
Temperature: 28.08 C, Humidity: 47.57 %RH
Temperature: 28.10 C, Humidity: 47.59 %RH
Temperature: 28.07 C, Humidity: 47.60 %RH
Temperature: 28.08 C, Humidity: 47.63 %RH
Temperature: 28.05 C, Humidity: 47.59 %RH
Temperature: 28.07 C, Humidity: 47.63 %RH
main()
ASUS Zenbook
发送
剪辑
收藏
删除
更多


GND VCC SCL SDA RES DC BLK
IPS 240*240
Light
Humiture
PWR RST BOOT KEY1 KEY2
Ctrl
intel
CORE
ULTRA
9
evo EDITION


19:27
0.94 KB/s
88%
2026年11月19日
15:50
[INFO] 当前模式：自动，窗户角度：90°
环境监测数据
温度：24.0°C
湿度：49.7%
光照：3 lux
[INFO] 当前模式：自动，窗户角度：90°
[KEY] 切换到网络状态页
[TTS] 开始播报
[TTS] 播报内容：现在是15点50分，武汉室外晴，温度13.5摄氏度，湿度百分之38，室内温度24.0摄氏度，室内湿度百分之30。
环境监测数据
温度：24.0°C
湿度：49.6%
光照：3 lux
[INFO] 当前模式：自动，窗户角度：90°
[INFO] 采样周期调整为 2000 ms
环境监测数据
温度：24.0°C
湿度：49.1%
光照：3 lux
[INFO] 当前模式：自动，窗户角度：90°
智能环境
监控系统
发送
剪辑
收藏
删除
更多


19:27
0.35 KB/s
88%
2026年11月19日
15:50
GND VCC SCL SDA RES DC BLK
网络状态
城市:武汉
时间:2026-11-19 15:50:17
SIM:就绪
湿度:49.7%
信号:-69dBm
数据:在线
室外温度:13.5C
室外湿度:38%
采样:83
更新:12:52
240*240
发送
剪辑
收藏
删除
更多


19:27
1.43 KB/s
88%
2026年11月19日
15:50
GND VCC SCL SDA RES DC BLK
环境监测
城市:武汉
时间:2026-11-19 15:50:22
温度:23.9C
湿度:49.5%
光照:31ux
模式:自动
室外温度:13.5C
室外湿度:38%
采样:89
周期:2000毫秒
IPS 240*240
发送
剪辑
收藏
删除
更多


19:27
0.31 KB/s
5G
88%
2026年11月19日
15:50
扬声器
发送
剪辑
收藏
删除
更多