阳礼宇个人研发日志

电控组第一天，今天的主要学习任务是学习单片机的基础知识。我了解到，单片机就像是我们机器车的“大脑”，它是一个微型计算机系统。

它的核心组成部分包括：中央处理器（CPU）、存储器（ROM和RAM）、输入/输出端口（I/O）以及定时/计数器等。这些部件集成在一块芯片上，共同协作。CPU负责执行程序指令，存储器用来存放代码和数据，而I/O口则是与外部传感器、电机等部件进行通信的桥梁。

通过今天的学习，我对这个核心控制器有了宏观的认识，为后续亲手编程控制小车打下了重要的理论基础。期待明天的学习与实践！

电控组第二天，今日的学习重点在于构建基于ARM Cortex-M内核的STM32单片机软件开发环境。主要实践内容为安装并配置：STM32CubeMX与Keil MDK-ARM（μVision5 IDE）两款核心工具链。在STM32CubeMX中，通过图形化界面完成了对微控制器引脚分配（Pinout）、时钟树（Clock Tree）的初步配置，并利用其HAL库初始化代码生成器，快速创建了工程框架。随后，在Keil μVision5中成功安装了目标器件的Device Family Pack，配置了编译器的优化选项，并设置了ST-Link调试器的Flash Download算法。最终，成功将生成的二进制文件烧录至芯片，完成了从环境搭建到程序下载的完整工作流，为后续的固件编写与调试奠定了基础。

电控组第三天，今日学习重心转向单片机对执行机构的电路驱动。深入理解了基于H桥结构的电机驱动电路原理，并通过集成驱动芯片（如TB6612FNG）进行实践。核心在于利用单片机的通用输入输出端口（GPIO）输出高低电平信号以控制电机的转向，同时通过其高级定时器（Timer）产生具有特定占空比的脉宽调制（PWM）波形，精确调控电机的平均电压与转速。实践中，我们使用STM32CubeMX配置了相应的GPIO为推挽输出模式，并初始化了定时器在PWM模式下的自动重装载值（ARR）与预分频器（PSC）。随后在Keil环境中编写代码，通过修改捕获/比较寄存器（CCR）的值，动态调整PWM占空比，成功实现了对直流减速电机的双向调速控制，完成了驱动电路理论与软件实践的闭环。

电控组第四天，今日聚焦于项目主控芯片STM32F103C8T6的硬件资源与电路设计。系统学习了该基于ARM Cortex-M3内核的MCU引脚定义与内部功能复用。我们详细分析了其LQFP48封装下的引脚分布，重点梳理了电源域（VDD/VSS、VDDA/VSSA）、复位（NRST）、启动模式（BOOT0/1）以及外部高速/低速晶振（OSC\_IN/OUT）等基础电路。同时，深入理解了其多功能复用引脚（GPIO）如何通过AFIO重映射，配置为不同的外设功能，如USART串行通信、I2C/SPI总线接口、ADC模数转换通道以及高级定时器（TIM1/2）的PWM输出通道。通过学习原理图，明确了最小系统电路与外设模块的连接关系。

电控组第五天，今日专题研究了机器人小车的全向移动平台核心——麦克纳姆轮。深入分析了其基于中间轮毂与周边呈特定角度（通常为45°）排列的辊子（Rollers）的机械结构原理。核心在于理解其运动学模型：通过控制四个轮子端线的速度矢量，利用辊子与地面接触产生的法向摩擦力进行矢量分解与合成，从而在笛卡尔坐标系下实现平台质心在X、Y轴方向的平移与绕Z轴的自转（即三自由度运动）。我们重点探讨了如何通过建立运动学逆解模型，将期望的车体速度与自转角速度，解算为四个独立直流电轮的期望转速。这为后续编写基于STM32的Mecanum轮系协同控制算法，实现精准的全向移动奠定了理论基础。

电控组第六天，今日重点研究了机器人小车的无线控制方案，学习了PS2手柄的通信原理与嵌入式集成应用。该手柄采用SPI（串行外设接口）协议与主控制器进行异步串行通信。其控制端作为主机（Master），我们的STM32微控制器需配置为从机（Slave），通过片选信号（CS/ATT）、时钟线（CLK）、命令线（CMD）与数据线（DAT）实现全双工数据交换。核心在于解析其数据帧结构，手柄会持续返回一个包含按键状态（数字量）和摇杆位置（模拟量，由内部电位器经ADC转换）的数据包。我们成功利用STM32的硬件SPI外设，配合DMA（直接存储器访问）控制器，高效读取了手柄数据。通过编写特定的数据解码算法，将获取的原始数据映射为小车运动的速度与方向指令，为下一步实现远控全向移动打下了坚实基础。

电控组第七天，今日任务是基于PS2通信协议，编写并调试手柄的操控代码。我们利用STM32的硬件SPI外设在从机模式下，通过DMA（直接存储器访问）循环接收来自手柄的数据帧。代码核心是解析该数据帧结构，包括起始字节、按键数据段及两个摇杆的模拟量AD值。

我们实现了基于状态机的数据包校验机制，通过计算校验和（Checksum）来确保通信数据的完整性。随后，编写了数据映射算法：将左侧摇杆的Y轴模拟量经线性变换后，解算为机器人的前进/后退速度；将右侧摇杆的X轴模拟量，解算为机器人的旋转角速度。最终，将这些控制量归一化后，作为参数传递给底层麦克纳姆轮的运动学逆解函数，成功实现了通过手柄对小车全向移动的实时、精确操控。

电控组第八天，今日专注于单片机固件的代码烧录流程与实践。我们使用ST-Link Debugger作为编程器，通过JTAG/SWD（串行线调试）接口与目标板上的STM32F103C8T6建立通信连接。在Keil μVision5 IDE中，我们配置了Flash Download算法，指定了片上Flash存储器的起始地址及容量。编译工程后，通过集成开发环境内的Loader命令，将链接器生成的Intel HEX格式可执行文件，经由调试适配器下载至微控制器的程序存储器中。此过程包含了擦除（Erase）、编程（Program）及校验（Verify）等关键步骤，并设置了复位并运行（Reset and Run）选项，确保固件在烧录完成后能自动开始执行。我们成功验证了整个“编辑-编译-烧录-调试”的开发闭环，为后续的持续集成与在线调试奠定了基础。

电控组第九天，今日进入硬件实操阶段，核心任务是完成直流减速电机与导线的可靠焊接。我们选用了线径与电机额定电流匹配的多股镀锡铜导线，并使用可调温电烙铁，将温度设定在约350°C，配合含松香芯的锡铅焊锡丝进行操作。焊接前，对电机原有的电极片进行了细致的氧化层刮除与预上锡处理，以确保良好的可焊性。焊接过程中，严格把控送锡量与加热时间，力求形成饱满、光亮且呈凹形圆滑过渡的圆锥状焊点，避免出现虚焊或冷焊。完成后，使用万用表的导通档对每个焊点进行了电气连接性测试，并确认正负极间无短路。最后，对裸露焊点部位套覆热缩管进行绝缘防护，为后续与电机驱动板的连接奠定了可靠的硬件基础。

电控组第十天，今日实操核心为构建主控系统供电与驱动链路。我们使用AMS1117-3.3V LDO降压模块将锂电池组电压降至3.3V，为STM32F103C8T6微控制器提供逻辑电源。同时，采用另一支持大电流输出的DC-DC降压模块（如MP1584EN）将总压降至5V，作为TB6612FNG电机驱动模块的VCC逻辑供电连接上，将STM32的通用IO口（PA0-PA3）与驱动模块的AIN1/AIN2、BIN1/BIN2控制输入端相连，以配置电机转向。将高级定时器（TIM1）的PWM通道（CH1-CH4）接至驱动模块的PWMA/PWMB端，实现占空比调速。此外，严格遵循电机驱动电源（VM）与逻辑电源（VCC）的共地原则，并使用万用表监测各节点电压，确保电平匹配与链路可靠，为系统上电调试做好了准备。