

代码重构

概述

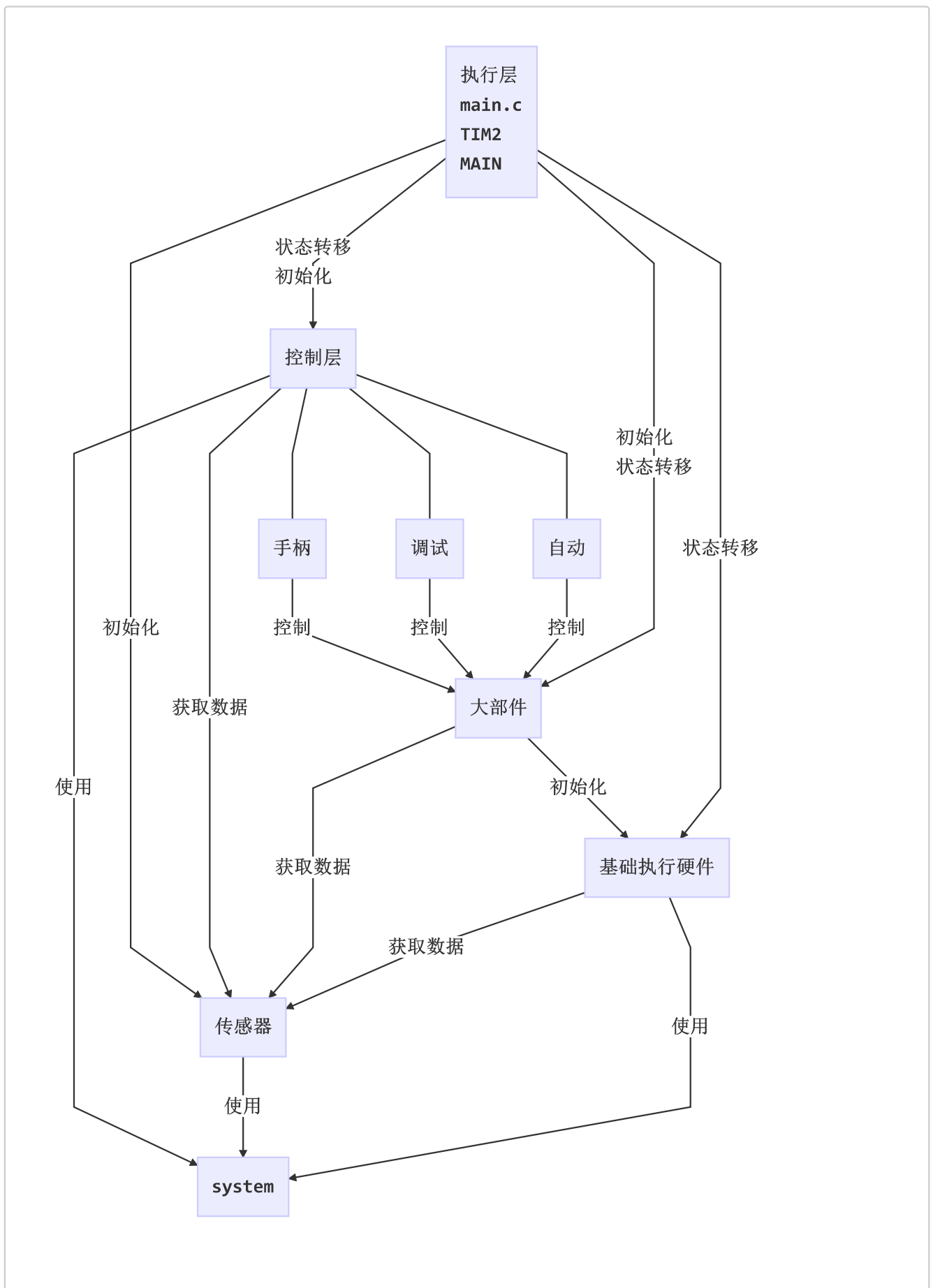
经过一个学期的对机器人代码的书写，有一定感悟，现在把代码的系统框架重新设计一下，方便之后的代码扩展和升级。

整体架构

基本思想是分层，stm32提供的包括串口，时钟，can总线，spi，iic，flash等底层为最底层，将封装为一个一个初始化函数，例如定时器的设置，要把每一种方式都写成一个初始化函数，这样上层使用的时候不用再使用结构体初始化。这个部分一步一步来，使用的时候就写一个，一步一步的完善。再往上一层是机器人的硬件驱动。这一层分为三小层，下面一层是对底层的使用，底层初始化函数的调用和使用在这一层，这一层向上提供最基本的硬件驱动，如摄像头，传感器等获取数据的设备。中间一层，初始化电机，步进电机，控制气缸的继电器等执行部件。在执行部件中可能会使用获取数据的设备的的数据。上面一层是对下层执行部件的进一步封装，组成机器人的各个大部件，例如底盘，各个发射机构。为了精确控制，这一层也需要检测数据。对于传感器的初始化建议放在主函数一开始，因为以上几层都有对数据的使用。

最上面一层是控制层和执行层，控制层直接控制部件的执行，分为两部分，一是调试部分，完全由命令控制机器各个部件的运动，具有最高权限，第二部分是自动部分，内置自动的策略，整个系统的状态转移。和控制层平行的一层是执行层，其实就是main.c中的大部件初始化、main函数大循环和tim2的中断函数。main函数的大循环中用于自动控制的状态转移，和各部件半自动的状态转移，tim2用于定时，延时，计时功能，主要用于没有反馈的部件延时，例如博创电机的位置环，不能稳定的反馈到达位置的信息，之后不用博创的驱动器的话，该功能可能会被废除。执行部件或大块部件需要写位于主循环的检查函数和tim2的一些函数。

架构图



其他问题

有一些stm底层较为特殊，例如can总线，显示屏的配置等和外界交互的，几乎全局都有可能使用到的，需要在main函数中配置初始化。