# 基于STM32H750VBT6智能车的图形化编程平台

# 需求分析与概要设计

## 项目说明

## 项目目标

项目主要针对硬件初学者，实现一个可以图形化编程的软件，用于控制STM32H750VBT6智能车运动。同时，利用AI语音识别帮助用户编程，加快开发效率；支持YOLO本地化部署到K210摄像头模块中，实现基础的图像识别功能。

## 软硬件环境需求

1. 硬件环境需求：
   1. 开发板：**自研**STM32H750VBT6群光开发板；
   2. 摄像头模块：K210；
   3. 车模及摄像头支架：**自研**InnoLegend车模；
   4. 电机：MG310；
   5. 电池：轮趣科技12V锂电池；
2. 软件环境需求：
   1. 内嵌Keil嵌入式开发IDE；
   2. 嵌入式FreeRTOS操作系统；
   3. PC端Windows操作系统；
   4. 百度语音识别API；
   5. Deepseek API；

## 使用的关键技术

1. 关键技术
   1. 自定义数据结构及转换：
   2. AI语音识别和数据结构转换：
   3. 底层bsp协议实现：根据芯片选型，自定义底层库文件，读写寄存器；
2. 难点
   1. FreeRTOS及系统配置：统共上百个配置引脚，以及数十个参数，需要深入了解硬件结构。
   2. 群光开发板原理图及PCB绘制：为实现专用的智能车，自研了群光开发板，是高速六层PCB板，对信号完整性及EMC要求高。
   3. 图形化编程错误分析：
3. 亮点
   1. 图形化界面优美便于编程：
   2. 直观显示编程错误位置：

## 需求分析

## 系统用例

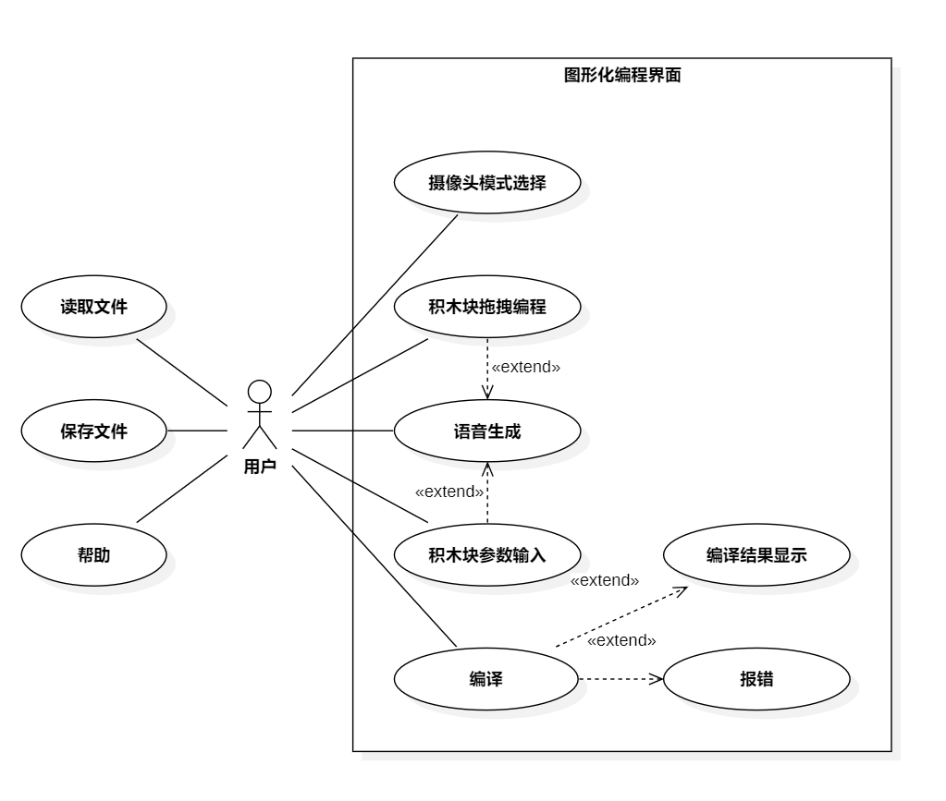
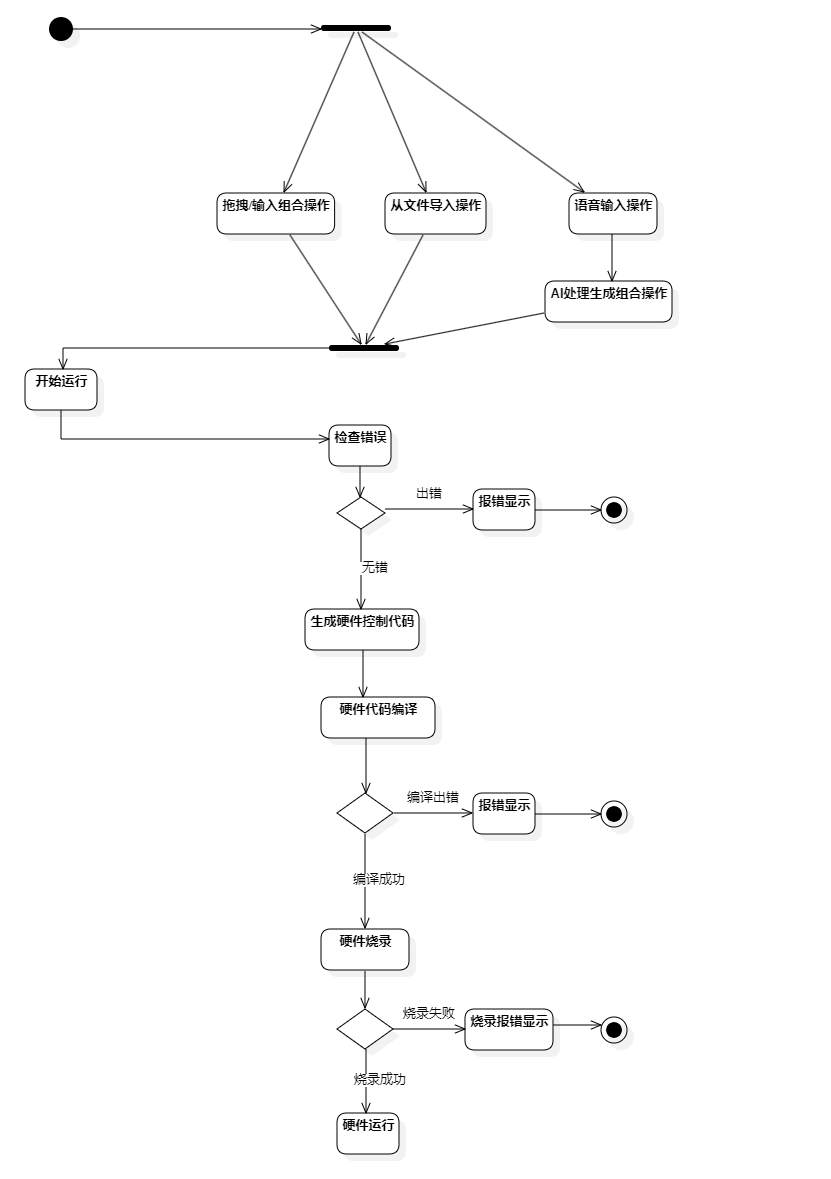


图1 系统用例图

1. 摄像头模式选择
   1. 参与者：用户
   2. 基本事件流：用户进入图形化编程界面，点击切换模式按钮，点击后程序切换摄像头模式并显示切换后的模式
2. 积木块拖拽编程
   1. 参与者：用户
   2. 基本事件流：用户进入图形化编程界面，点击左侧积木选择栏，程序根据用户点击的积木栏生成对应可拖拽的编程积木块
3. 积木块参数输入
   1. 参与者：用户
   2. 基本事件流：用户点击编程积木块的文本框进行输入
4. 语音生成
   1. 参与者：用户
   2. 基本事件流：用户点击录制按钮，程序开始录制用户语音音频，用户再次点击结束录制，程序根据音频自动生成编程积木块
5. 编译
   1. 参与者：用户
   2. 基本事件流：用户点击导出，程序检查编程积木块，如果有参数缺失等问题给出相应报错，如果没问题则生成对应目标代码并进行进一步烧录
6. 读取文件
   1. 参与者：用户
   2. 基本事件流：用户点击读取，程序弹出文件选择框并根据用户给出的路径读取文件
7. 保存文件
   1. 参与者：用户
   2. 基本事件流：用户点击保存，程序根据文件位置保存文件
8. 帮助
   1. 参与者：用户
   2. 基本事件流：用户点击帮助，程序显示基本的说明文本

## 业务流程



**图2 业务流程图**

1. **操作定义阶段：**

在操作定义阶段，系统提供了三种人机交互通道，以便用户能够便捷地配置和操作系统。首先，可视化拖拽功能通过图形用户界面（GUI）让用户能够直观地进行操作编排和组合。其次，文件接口支持导入.json和.xml格式的配置文件，方便用户通过文件批量导入操作组合。最后，智能语音功能通过自然语言处理（NLP）引擎解析用户的语音指令，并通过 AI 算法生成符合规则的操作组合。特别是语音通道内置了语义理解层，能够自动补全操作逻辑，从而提升语音交互的准确性和便捷性。

1. **代码构建阶段：**

在代码构建阶段，系统首先进行 语法校验，确保操作组合符合正确的语法规则，并且没有错误。若发现语法问题，系统会检测异常，并返回具体的错误信息，提示用户进行修正。这一阶段的目标是保证生成的操作代码准确无误，避免因代码错误导致后续环节出现问题。

1. **硬件实现阶段：**

硬件实现阶段首先通过 Keil工具链进行交叉编译，将编写好的代码生成机器码。若编译过程中出现问题，系统会返回报错显示，帮助开发人员快速定位问题。编译通过后，系统会输出标准的hex烧录文件，用于将固件写入硬件。在烧录过程中，系统会进行烧录验证，确认固件是否成功写入。如果烧录失败，系统会返回报错显示。烧录成功后，硬件将会激活并完成初始化引导，进入正常的运行状态，标志着整个开发流程的完成。

## 概要设计

## 功能模块设计

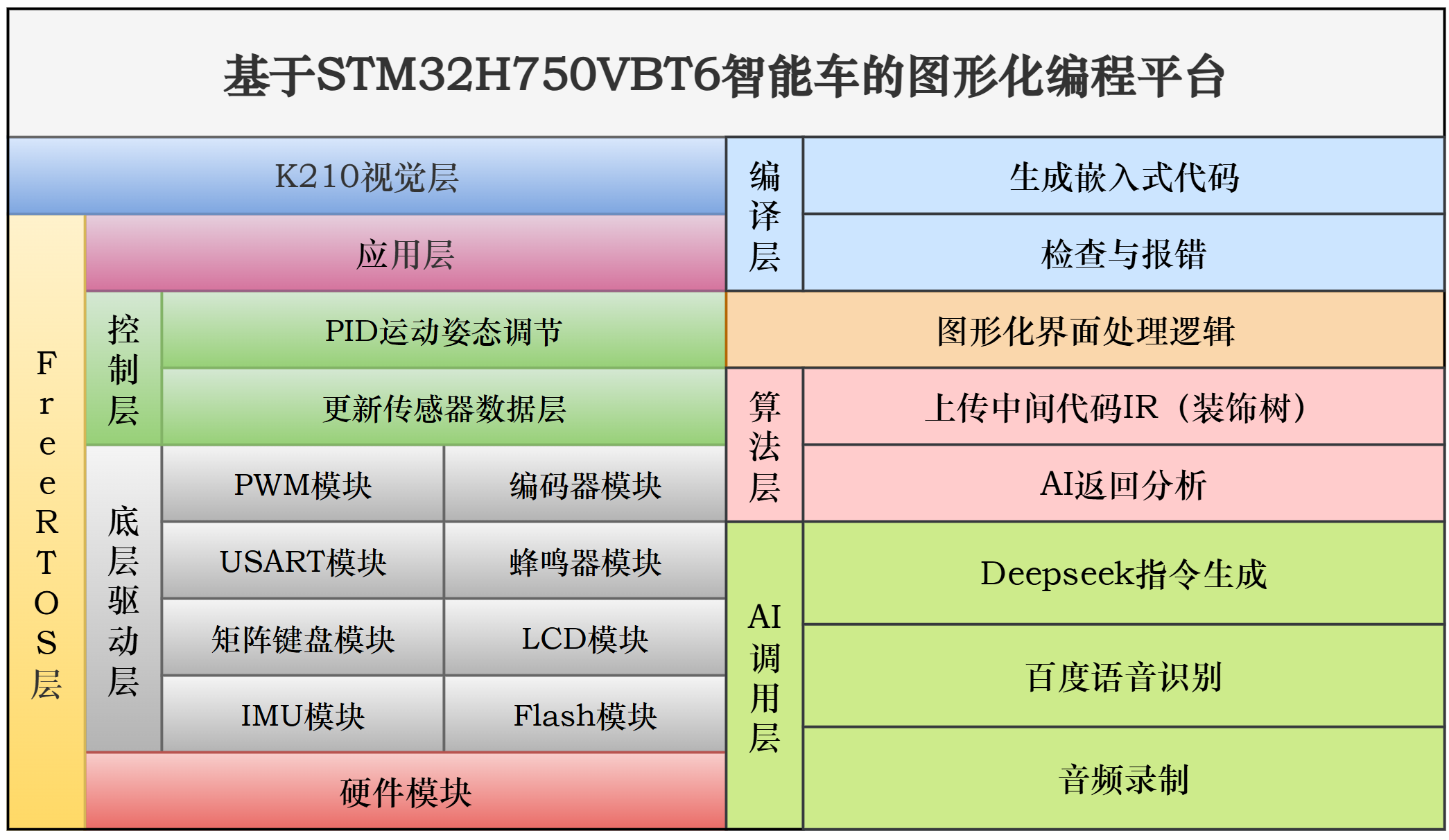


图3 功能模块图

1. **左侧硬件模块功能**

* K210视觉层：K210是一个AI视觉芯片，通常用于图像识别、目标检测、视觉跟踪等应用。其功能包括从摄像头获取图像数据，进行实时处理，并传输给后续处理单元。
* 应用层：用于调用底层接口，实现小车的运动，API如：go-line, circle, spin等。
* 控制层：用于获取硬件数据，将数据通过PID调节生成真实控制信号。
* PWM模块：PWM（脉宽调制）模块用于生成控制信号，通常用于控制电机的速度、调光LED灯光强度等。通过改变信号的占空比来控制输出的功率和能量。
* USART模块：USART（通用同步异步接收发送器）模块用于串行通信，支持全双工数据传输。
* 蜂鸣器模块：蜂鸣器模块通常用于产生声响提示。
* 编码器模块：编码器用于测量旋转角度，编码器提供的脉冲信号用于计算旋转位置或转速。
* LCD模块：LCD显示模块用于显示信息，可以用于车载显示屏、状态指示等。支持显示字符、图像和图形。
* Flash模块：Flash存储模块用于数据存储，如程序存储、参数保存等。
* IMU模块：IMU（惯性测量单元）模块用于获取车辆的加速度、角速度和方向信息。

1. **右侧软件模块功能扩充：**

* 编译层：检测装饰树，确定是否存在错误，遇到错误则抛出显示，正确则生成最终代码。
* 图形化界面处理逻辑：用于处理与用户交互的图形界面。包括图形界面元素的显示和响应，如拖拽、调参等。
* 算法层：先将AI返回的数据结构转化为中间代码IR（装饰树），并将其上传到图形化界面进行更新显示。
* AI调用层：先将人的语音保存成.wav文件，再将其放入百度语音转换生成文本，最后将其放入Deepseek转换为目标数据格式。

## 核心类图

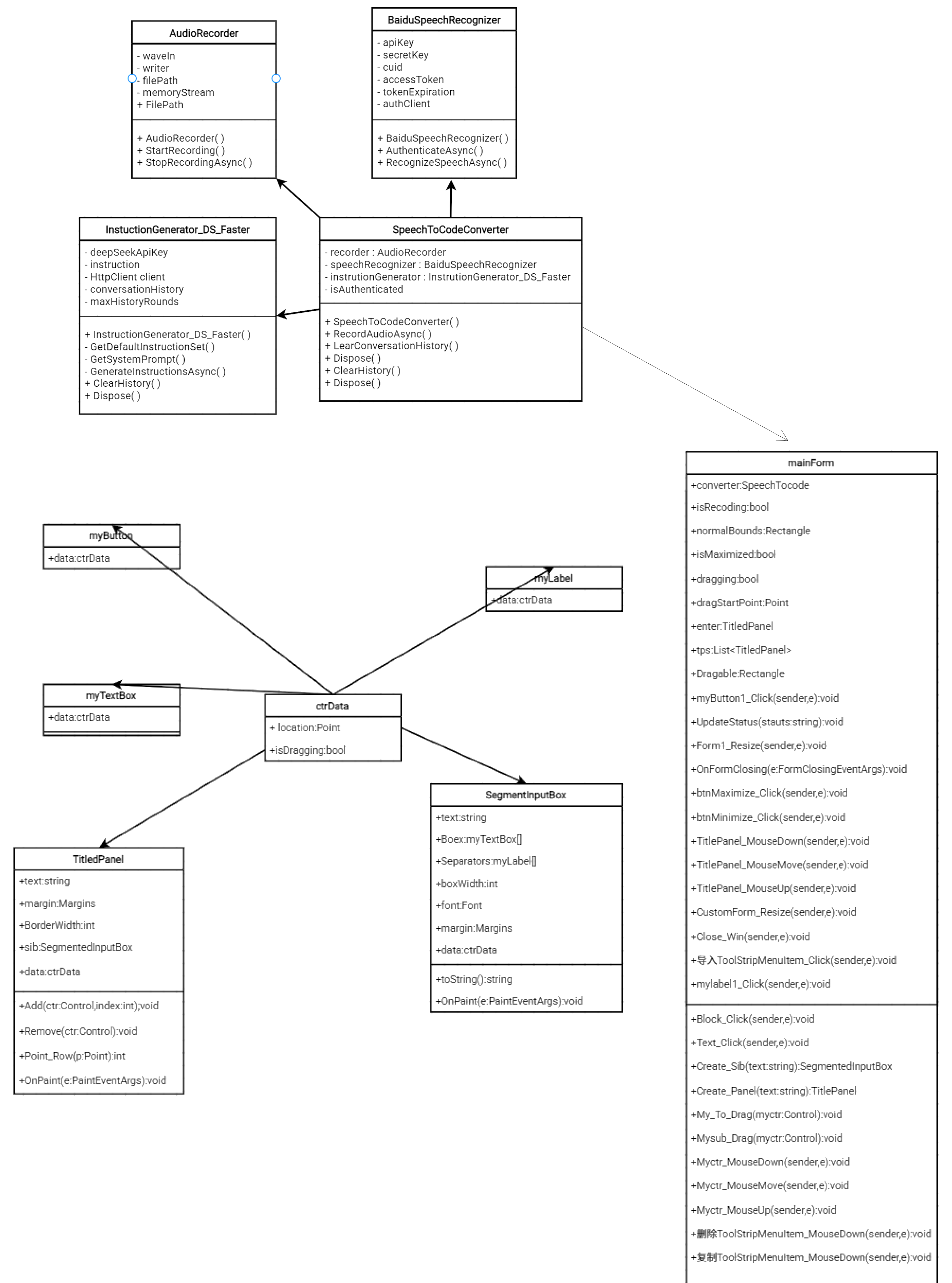


图4 核心类图

* 实体类：可拖拽的积木控件，储存参数与结构信息；
* 控制类：程序调用语音识别api与deepseek api实现语音生成，调用算法层ai分析模块生成编程积木，调用编译层模块生成嵌入式代码
* 边界类：即按钮等可交互的控件

## 界面设计

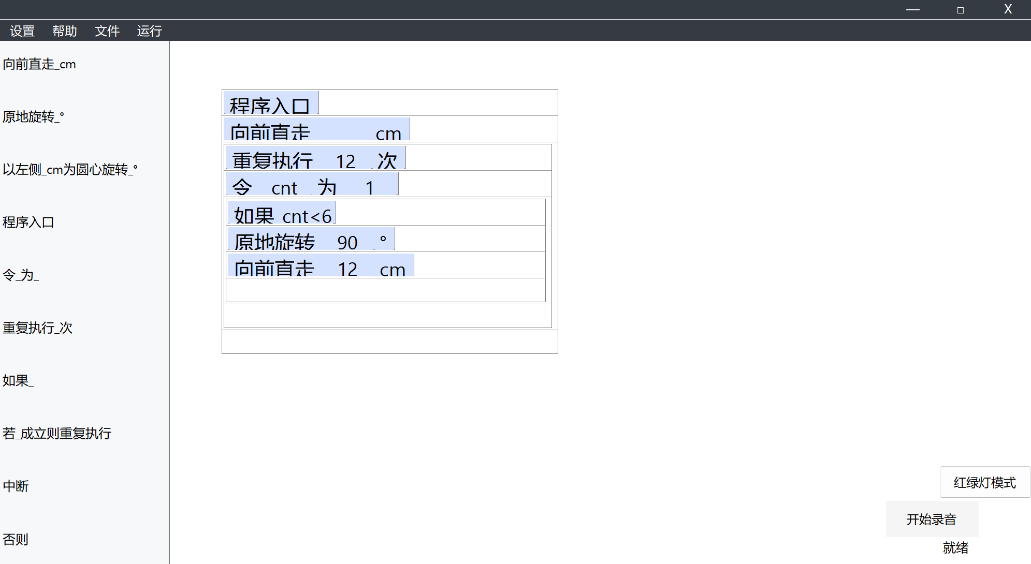


图 5 系统界面

1. 顶部菜单栏

包含四个下拉菜单，分别是设置、帮助、文件和运行，这些菜单提供了程序设置和操作功能，方便用户管理程序的不同方面。

1. 左侧命令面板

左侧命令面板展示了一系列可拖拽的命令模板，采用竖直列表的形式，每一项都是一个可编辑的命令块。例如，向前直走、原地旋转、以左侧为圆心旋转等。这些命令块可以轻松地拖拽到右侧的工作区，用户通过这些命令块来设计程序流程。

1. 中央工作区

中央的工作区是一个大面积的空白区域，用于摆放和组合从左侧命令面板拖入的命令块和容器块。每个命令块和容器块都可以嵌套和按层级组织，这样的设计让程序流程的构建更加直观和清晰。

1. 右下角操作按钮区域

在右下角的操作按钮区域，用户可以找到“红绿灯模式”按钮，用于切换程序的执行模式，分别通过红色和绿色指示不同的运行状态。此外，还提供了“开始录音/停止录音”按钮，点击后可以启用麦克风录音并将录音内容转化为指令流。同时，界面还会显示状态标签，实时更新录音和转换的进度。

1. 窗口装饰

窗口的装饰部分包括标准的最小化、最大化和关闭按钮。用户可以通过“程序入口”面板和标题栏进行窗口的拖动、最大化或还原，操作方便，提升了用户的使用体验。整体而言，这个界面通过可视化、拖拽式的操作方式，使得编程任务的构建和管理变得更加简便和直观。