**《 智能系统 》实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学号** | | **姓名** | **承担任务** | | | **贡献度** | **得分** |
| 20182292 | | 张博钧 | **UI界面设计及实现、推理机实现** | | | **9** |  |
| 20181778 | | 胡栋月 | **仿真模块实现** | | | **8** |  |
| 20181719 | | 彭燃 | **数据库实现** | | | **7** |  |
| 20182587 | | 胡卫 | **代码调试、文档编写和组织** | | | **8** |  |
| 20184212 | | 代志豪 | **下位机与传感器连接实现** | | | **6.5** |  |
| 20184360 | | 张宇 | **下位机与百度地图api连接实现** | | | **6.5** |  |
| **实验题目** | **系统集成** | | | | | | |
| **实验时间** | **2021.06.12 – 2021.06.26** | | | **实验地点** | **DS3 401** | | |
| **实验成绩** |  | | | **实验性质** | **□验证性 □设计性 ☑综合性** | | |
| 教师评价：  □算法/实验过程正确 □源程序/实验内容提交  □程序结构/实验步骤合理 □实验结果正确  □语法、语义正确 □报告规范  其他：  评价教师签名： | | | | | | | |
| 1. 实验目的   为实现十字路口红绿灯智能控制，本次实验的目的是：  （1）完成项目集成  （2）实现合理的界面 | | | | | | | |
| 1. 实验项目内容   1、项目集成  （1）传感器数据采集；  （2）下位机数据封装上传；  （3）上位机数据管理；  （4）知识库管理与维护；  （5）推理机与解释器设计和实现；  （6）数据封装下发；  （7）传感器响应；  （8）测试“传感器采集——传输——数据库——知识库——推理机——数据库——传输——传感器响应”的全过程。  2、系统可视化  （1）设计系统UI；  （2）实现系统UI；  （3）通过UI实现对系统的管理维护和运行仿真。  3、实时交通路口仿真（附加）  （1）获取地图软件（高德、百度等）中的路况信息；  （2）解析路况信息；  （3）通过UI实现交通路口红绿灯仿真。  \* 系统UI参考《程序基本界面.exe》 | | | | | | | |
| 三、实验过程或算法（代码）  要求：  按实验项目内容要求逐条编写   1. 项目集成 2. 传感器数据采集   在下位机使用避障传感器来采集在一定的时间内（15秒）经过的车辆的数目。根据避障传感器的特点——当距离红外探头3-30cm内有障碍物时（车辆通过时），避障传感器的输出位为高电平；当距离红外探头3-30cm内没有障碍物时（车辆通过后），避障传感器的输出位为低电平。所以，在下位机中，使用一个int变量，来存储避障传感器的输出位电位变化次数，将这个次数除以二则得到通过的车辆数。  然后，规定每过一定的时间，就将这个测得的通过车辆数上传到上位机。  实现数据采集函数，使用Metro库，实现类多线程  *#include "Metro.h"*  声明控制数据采集的Metro对象  Metro countMetro = Metro(15000);  在loop()函数中，记录点位变化次数   1. tempVal = digitalRead(buttonpin); 2. if(val != tempVal) { 3. val = tempVal; 4. count++; 5. } 6. 下位机数据封装上传   下位机直接定时将count发送给上位机，每次发送使用一个空格分隔。  下位机设计和实现的数据包编码和传输函数   1. if (countMetro.check()) {  *//检查blink1Metro间隔* 2. Serial.print(count/2 + 1, DEC); 3. Serial.print(" "); 4. count = 0; 5. } 6. 上位机数据管理 7. 存储下位机上传的数据  |  |  |  |  | | --- | --- | --- | --- | | 名称 | 时间戳 | 东西 | 南北 | | 类型 | TEXT | TEXT | TEXT |      1. 存储上位机输出的数据  |  |  |  |  | | --- | --- | --- | --- | | 名称 | 时间戳 | 灯选择 | 持续时间 | | 类型 | TEXT | TEXT | TEXT |  1. 知识库管理与维护 2. 可信度知识  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 名称 | 编号 | 前提 | 结论 | 可信度 | 阈值 | 更新时间 | | 类型 | NUMBER | TEXT | TEXT | NUMBER | NUMBER | TEXT |      1. 模糊知识  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 名称 | 编号 | 前提论域 | 前提 | 结论论域 | 结论 | 模糊矩阵 | 可信度 | 阈值 | 更新时间 | | 类型 | NUMBER | TEXT | TEXT | TEXT | TEXT | TEXT | NUMBER | NUMBER | TEXT |   模糊矩阵的计算过程    模糊知识存入知识库  根据id删除模糊知识  获取模糊知识   1. 推理机与解释器设计和实现 2. 可信度推理机的设计与实现   设计原则  由于各个时间段的车流量不一样，有些时段车辆流会更高。参考了西安友谊西路的车流量统计，于是我们设置早高峰和晚高峰两个时间段，来延长绿灯的时间。延长的时间我们设计为5秒。将该数据添加到数据库的表中  计算可信度知识是否可用  这里的plusTime指的是是否需要额外增加时间，来缓冲高峰时间段的压力。  通过获取当前的时间与我们自己设定的早上8~10，下午17~19时间段进行比较，如果相等则，当前绿灯或黄等时间增加5秒，否则，就不增加时间。判断是否相等，调用useReliableKnoledge（）函数，返回值如果等于-1，则表明可信度知识不可用，不属于车流量高峰时段。反之，需要增加额外的时间。  这里的ledTime需要正常计算车流量的个数。如果该车流量的个数，满足模糊知识设计的要求，则返回的时间是经模糊知识设计规则处理过的时间。反之，按照我们原始的假设，按照一车一秒来计时。   1. 模糊推理机的设计与实现   设计原则  首先得到经过的车辆数，然后调用useObescureKnowledge()函数，根据其返回值,判断模糊知识是否可用。若返回值大于0，则绿灯和黄灯的时间为返回的值。若返回值等于-1，则绿灯和黄灯的时间为原始设定的一车一秒。  useObescureKnowledge()函数  首先获取数据库中的所有与模糊知识相关的信息，并调用transformObscureKnowledge()函数，进行相关格式的处理。然后调用triangleFuzzy()函数，进行模糊化，之后调用closseness()函数，计算两个函数的贴近度。  IMG_256得到贴近度之后， 通过假言推理，得出结论的模糊集合。最后调用defuzzification()函数去模糊化。  IMG_256TransformObscurekonwledge()函数：主要是进行格式的处理  TriangleFuzzy()函数: 模糊化方法采用三角法，即将精确的数值用如下的隶属函数转化为模糊集合。理论知识： 设x\*为实测的精确值，A为转换后的模糊集合：  IMG_256屏幕截图 2021-05-29 163006IMG_256 Closeness()函数：计算两个模糊数的贴近度理论知识：贴进度计算方法如前文所述，为C(F, G)=(F·G+(1-F G))/2。   1. 解释器的设计与实现   基于传感器得到的事实数据，跟踪并记录系统的推理过程和推理中使用的知识，本实验中，解释器解释说明的格式为：  因为有事实……（事实数据），应用…….（知识库中所用知识ID），所以……（得出的结论）  D:\文档\1099646580\Image\C2C\Z}4)E$_Y%$]66Z3[~Q8GE]1.png  D:\文档\1099646580\Image\C2C\@4UZHQ0501O9`~T}%@Q{{Z5.png   1. 数据封装下发   上位机发送给下位机的控制数据，第一位控制对象位，当控制对象位为[1,3]之间时，为有效。第二位持续时间位。并在之后的程序中根据这两位来更新对应灯亮的持续时间。     1. 传感器响应   下位机控制信息    信号灯响应显示  使用Arduino 的 millis()函数来获取自开机后经过的毫秒数，从而计算出持续时间，并以此来循环更换红绿灯。    使用一个timeSlot数组，数组大小为3，分别存储着三个灯各自亮持续时常。  使用一个全局变量i，i决定当前是哪一个灯正在亮，每当持续时间超过了timeSlot[i]，i就会加一并取模3。     1. 测试“传感器采集——传输——数据库——知识库——推理机——数据库——传输——传感器响应”的全过程 2. 传感器采集   采集到南北车辆数     1. 传输   车辆数传输到下位机，再传输到上位机   1. 数据库   将得到的事实数据“车辆数”存储到数据库   1. 知识库   根据已知事实选取匹配的知识     1. 推理机   基于事实和选取的知识应用推理机进行推理，得到结论    直接找到结论并使用     1. 数据库   将得到的结论存进数据库，以便下次使用时直接查找，避免重复计算     1. 传输   将得到的结论“南北方向的红灯时间”传输到下位机。再传输到信号灯   1. 传感器响应   信号灯的红绿灯持续时长更改为推理得到的结论  2.系统可视化  1）主界面  本系统主要包括4个功能，系统仿真、知识维护、统计  分析和数据维护。    2）系统设置界面  设置页面可以对数据来源进行选择，也可以设置红绿灯的周期时长、最短红灯时长、最短黄灯时长和最短绿灯时长。    下拉框可以切换端口    3)数据管理界面  数据管理界面实现了对数据库所有数据的查询。可以通过输入两个时间点来实现对指定时间段的数据的查询    4）统计分析界面  实现了对数据库中所有数据的统计分析。可以通过输入两个时间点来实现对指定时间段各方向车流量数据和绿灯时长数据的统计和展示。    5）知识维护界面  知识维护界面分为两个部分，分别负责可信度知识和模糊知识的增删改查操作。通过双击表中的数据可以直接将该行数据填入到输入框中，方便对已有数据进行修改。    模糊知识库    可信度知识库  3.实时交通路口仿真  仿真分析界面包括两个部分，一个是对当前测量的车流量以及绿灯  时长 统计，包括启动、暂停、刷新操作，方便进行数据的对比。  第二个部分是根据检测数据对当前实际道路车况的模拟，以动画形式呈现。      数据获取代码，url为指定的链接，通过修改url的内容可以获取不同的数据。 | | | | | | | |
| 1. 实验结果及分析   2. 系统UI测试  1)系统设置测试：      2）数据查询测试：  直接点击查询按钮查询所有数据：  E4AC2148-9AD7-436E-8C7D-DE1251360260  对某一时间段内的数据进行查询：  CF3047D1-1769-4A95-BF6D-8DFA373327CA  仅设定起始时间进行查询：  B562C5A2-1718-4155-ADFD-B8A2141858B2  3）知识维护测试：  双击一条知识，输入框中自动填入相应数据      对知识进行修改    增加一条新知识      可以看出，id为5的一条新知识被加入到知识库中  删除已有知识      可以看出，id为5的知识被删除  4)统计分析测试    5)解释器测试    可以看到，通过从百度地图api获取当前道路信息，并使用推理机对数据进行分析处理  6）仿真界面测试    根据数据绘制图表    刷新后图表清空，重新开始绘制 | | | | | | | |
| 1. 完成时间   （1）实验时间：2021.6.19，2021.6.26  （2）检查时间：2021.6.26  （3）2021年6月26日23:59之前提交实验报告 | | | | | | | |