附件1

国家级大学生创新创业训练计划

项目申报表

（创新训练项目）

|  |  |
| --- | --- |
| 推荐学校： | 浙江大学城市学院（盖章） |
| 项目编号： |  |
| 项目名称： | 物联网可视化编程平台开发 |
| 所属一级学科名称： | 计算机应用 |
| 项目负责人： | 胡子阳 |
| 联系电话： | 15858260563 |
| 指导教师： | 侯宏仑 徐永哲 |
| 联系电话： | 13071858629 |
| 申报日期： | 2017.3 |

浙江省教育厅 制

二○一七年一月

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **项目名称** | | | | | 物联网可视化编程平台开发 | | | | | | | | |
| **项目所属**  **一级学科** | | | | | 计算机应用 | | | | | | | | |
| **项目类型** | | | | | 创新训练项目 | | | | | | | | |
| **项目实施时间** | | | | | 起始时间： 2017年4 月 完成时间： 2018年4月 | | | | | | | | |
| **项**  **目**  **简**  **介**  (100字以内） | 本项目将开发一套物联网可视化编程平台原型，通过平台集中部署智能硬件节点，能以可视化编程的理念快速建立物与物之间的逻辑联系，从而实现智能硬件的相互通信。最终可落地到一个智慧工厂的场景中去。 | | | | | | | | | | | | |
| **申请人或申请团队** |  | 姓名 | | | | 年级 | 学号 | 所在院系  /专业 | | 联系电话 | | E-mail | |
| 主  持  人 | 胡子阳 | | | | 2 | 31501333 | 计算学院  软件工程 | | 15858260563 | | 31501333@stu.zucc.edu.cn | |
| 陈董锴 | | | | 2 | 31501324 | 计算学院计算机科学与技术 | | 13587361291 | | 31501324@stu.zucc.edu.cn | |
| 成  员 | 颜哲锟 | | | | 2 | 31502373 | 信电学院自动化 | | 15858272186 | | 31502373@stu.zucc.edu.cn | |
| 朱兵 | | | | 1 | 31602402 | 信电学院自动化 | | 17376501924 | | 31602402@stu.zucc.edu.cn | |
| 程竞泽 | | | | 1 | 31601116 | 计算学院计算机科学与技术 | | 17367073061 | | 31601116@stu.zucc.edu.cn | |
| **指 导 教 师** | 第一指导教师 | | 姓名 | | | 侯宏仑 | | 单位 | | 计算机与计算科学学院 | | | |
| 年龄 | | | 44 | | 专业技术职务 | | 副教授、博士 | | | |
| 主要成果 | | | | | 1. 指导学生吴海波获得2012年度，全国云计算应用大赛二等奖 2. 指导学生MRFLO项目获得“挑战杯”2014度浙江省一等奖 3. 指导学生叶建州等获得2014年浙江省新苗计划项目 4. 指导学生钱雪峰等获得2013年国家大学生创新创业项目 5. 指导学生张以鹏等获得2014年国家大学生创新创业项目 6. 指导学生翟丹妮等获得2015年国家大学生创新创业项目 7. 指导学生陈榆等获得2016年国家大学生创新创业项目 8. 指导学生郑楠等获得2016年Google国家大学生创新创业项目 9. 指导学生江展翔等获得2016年国家大学生创新创业项目 | | | | | | | |
| 第二指导教师 | | 姓名 | | | 徐永哲 | | 单位 | | 计算机与计算科学学院 | | | |
| 年龄 | | | 35 | | 专业技术职务 | | 讲师、博士 | | | |
| 主要成果 | | | | | 1. 参与韩国应急管理局 NEMA科研 2. 主持谷歌高尔夫 LG U+ inforsystem科研 3. 主持乐视众乐乐电子琴科研 4. 主持CAMfit走势识别科研 5. 韩国专利，粒子烟雾流体模拟方法及装置，NP-13-1393 6. Apparatus and Method for Processing of Exercise Information 7. 보안 카메라 시스템 기반의 반려동물 공격성향 분석, 한국정보기술학회논문지 8. Multi Camera for Surveillance System Ground Detection and, International Journal of Smart Home 9. Simulation of Smoke Improve Computing Coordinate Performance， International Journal of Multimedia and Ubiquitous Engineering | | | | | | | |
| 1. **申请理由**（包括自身具备的知识条件、自己的特长、兴趣、已有的实践创新成果等）   项目组主要成员在创客项目、国创项目中已有过愉快的合作，完成过《可旋转跟踪太阳能板》、《智能电表》、《应用ThingWorx的物联网植物培养箱》等项目，所有项目都正常运行至今。在项目进行中深感目前物联网依旧不尽如意，在不断实践和反思中认为我们可以尝试着共同去优化目前的物联网，本项目亦是基于之前多项物联网项目的升华与延伸。  项目组主要成员熟悉一些基本传感器的通信协议以及使用方法，比如DS1307 时钟模块、GY-30光强传感模块、BMP180气压温湿度传感模块等传感器的 IIC 协议，DS18B20和DHT系列温湿度传感模块等传感器的单总线通信协议、电能表的 Modbus 协议等等，熟悉MQTT这一轻量化的物联网通信协议以及传统的WiFi、BlueTooth。能熟练使用Arduino IDE对基于Arduino的物联网编写程序，对各种传感器进行数据采集。能利用现在主流的ESP8266模块，利用MQTT协议将各种传感器数据通过WiFi发送给MQTT Broker，并使用联网的Android手机接收来自MQTT Broker的传感器信息。项目组大部分成员有一年以上 Arduino 编程经验和物联网建设经验,并能结合各种底层传感器、控制器和机械结构设计出一些有趣的作品，项目负责人曾经做出一台 Arduino 与Android通过BlueTooth通信的 3D 打印延时摄影滑轨。  项目组在本项目进行前期可行性分析期间，已经在本地计算机中配置好了node.js框架，并架设了基于node.js的Node-RED可视化工具平台，在服务器上架设了MQTT Broker，都可以正常运行。对JavaScript 语言、node.js 框架、HTML5 有初步的学习，能够使用JavaScript语言通过ESPruino WEB IDE对刷写ESPruino固件的NodeMCU编写程序，并制作了简单的DEMO，也能对 Arduino 刷 Firmata固件，通过WEB界面直接对 Arduino 的引脚进行远程控制。基本了解Node-RED可视化 工具平台的使用，用此实现了一些逻辑控制以及网络消息的传递功能，但Node-RED毕竟只是一个工具平台，还不能也不可能对物联网实现一站式解决，因此我们也会研究Node-RED可视化工具平台，并使用开源代码对其二次开发以实现我们建设交互式可编程IOT平台原型的目的。 | | | | | | | | | | | | | |
| 1. **项目方案** 2. **项目研究背景**    1. **国内外物联网现状**   现在已经有像Thingworx, Murano, Electric Imp等比较成熟的物联网平台，他们的产品可面向智能家居，智能商场，智能交通等领域，但是使用效果并不理想，原因在于他们的硬件大规模部署起来不方便，硬件的管理流程较复杂。它们组网的方式多种多样，并且大多采用ZigBee、BlueTooth、WiFi、Ethernet、RFID方式组网，它们所做的最多的是大数据分析，数据安全存储，远程检测与控制等功能，但很少能灵活提供合适的软硬件之间的API。  未来，物联网体系中必将会有大规模的传感器需要快速部署与管理，这就需要一个物联网平台，而市面上具有管理这些独立传感器使之成为一个有机整体的平台并不多见，市面上更多的只是简单开发一款APP，用APP去远程控制一两个电器，并且这些平台也都没有真正进入人们的日常生活，人们对物联网平台软件的依赖程度普遍较低。   * 1. **本项目已有基础**   项目成员会利用目前主流的物联网芯片进行编程，比如安信可GPRS模块，ESP8266模块。我们已经通过MQTT协议跑通了整个物联网神经回路。能利用手机APP远程控制各种电器电器开关，远程显示传感器数据。在平台的研究上，我们已经掌握Node-RED这一开源物联网平台的核心技术，并能在此基础上用node.js代码进行简单的二次开发，加入我们自己需要的功能。   * 1. **尚缺少的条件及方法等**   Node-RED平台技术属于前沿科技，我们虽然可以通过代码嵌入我们自己的功能，但是其余功能，比如安全性，大数据分析，硬件大规模部署等需要日后慢慢研究。最终落地的场景需要一定人力，时间和资金去布置。   1. **项目研究目标及主要内容**    1. **研究目标：**   借鉴Node-RED可视化 工具平台开发一个完整生态的软件，支持可视化编程、可视化控制、可视化数据显示。在本项目开发的物联网平台中嵌入的Google Blockly模块，通过增加各类不同功能的模块以支持用户只用拖拽代码块的方式实现控制并管理物件，降低物联网开发门槛，让逻辑关系快速的建立在物体与物体之间。对传统传感器与WiFi、GPRS等通信模块集合封装成智能传感器（智能执行器），基于MQTT通信协议接入我们的物联网平台。   * 1. **主要内容：**      1. **物联网平台的开发**   + 调用改进的智能传感器（智能执行器）库文件，批量调用、批量控制，实现大规模快速部署；   + 实现平台与硬件之间标准化数据的交互格式，如JSON格式；   + 改进Mashup、Dashboard、Freeboard等显示模块，将智能传感器反馈回来的数据以自定义的形式实现可视化显示；   + 封装好相应的函数模块，作为接口供用户调用，从而降低物联网开发和管理门槛。     1. **底层硬件的集合封装和库的改进**   + 采用最新的ESP8266模块（NodeMCU开发板）、ESP32模块、GPRS+GPS+GSM A7模块与传统传感器进行集成封装成智能传感器（智能执行器），都能借用WiFi或者GPRS通过MQTT协议（Message Queuing Telemetry Transport，消息队列遥测传输）与我们的物联网平台实现通信；   + 对库进行移植、优化和改进，并转换为JavaScript代码，并封装成本项目物联网平台中的模块，实现对改进库的调用。     1. **数据库（云）的开发**   + 数据库接收来自本项目开发的物联网平台的API的设计；   + C:\Users\PLANE\Desktop\微信图片_20170321214242.jpg数据库对JSON格式的属性的存储；   + 数据库发送数据到本项目开发的物联网平台的API的设计；     1. **场景的搭建**   + 如图所示我们将我们的平台用于控制工厂的生产流程，借用慧鱼模型将由平台操控的智能硬件接入，然后由用户自己去设计逻辑去操控一个工厂的完整生产过程。   C:\Users\PLANE\Desktop\微信图片_20170321215136.jpg   * + 在场景本地部署Android Things用于场景中各种智能传感器（智能执行器）的数据本地可视化。如计数传感器，实时提供给用户在仓库中的货物数量。机械手设备，抓取流水线上加工完成的货物放入仓库中。   本项目的项目组已经在Node-RED可视化工具平台上制作了一个简单的功能，Dashboard如下图：     1. **项目创新特色概述**    * **本物联网平台支持智能硬件大规模部署。**本项目是轻量级且高效的。最显著的一点是在对同类的传感器、执行器进行烧录程序时是批量的烧录而非单一的录入，从而实现大规模的硬件部署及管理。这一点是目前已有的物联网平台所做不到的，也是最为重要与突出的一部分。这样做的优点是对同类别的仪器可以直接烧写程序，当多个传感器同时发送数据到平台时，平台能够针对每一个设备做到点对点的命令回应，不仅不会发生指令的冲突还能大大加快工作的效率。    * **支持智能硬件连接和数据标准化。**智能硬件上将通信模块ESP8266芯片（ESP32、A7）和传统传感器芯片进行集成封装成智能传感器（智能执行器），直接成为可以入网的传感器（执行器），利用发布/订阅模式的MQTT协议，使数据实现有序、可靠、双向字节流的传输，能最大限度减小网络流量，并且在数据交互之前就已经完成了标准化。    * **支持普通用户模块化建设物联网。**软件上借鉴Node-RED可视化 工具平台开发出一个物联网平台使其足以适应特定场景的需求，通过编写HTML和JavaScript文件可以形成一个个函数的节点。这些节点都是封装好的，类似于一个个接口，用户只需要输入对应的参数，再加上几根连线，便可以完成对物件的操控与管理。从而用户可以在并不熟悉编程语言的情况下，依旧能够完成对物件的操控，下达相应的命令，轻便而又快捷。 2. **项目研究技术路线**   本项目分为网络层协议适配部分、感知层硬件改造部分、应用层WEB软件系统开发部分。  **网络层，**协议适配部分。协调智能硬件与物联网平台的通信，使用的是MQTT协议。大规模智能硬件接入本项目物联网平台时的初始化，基于该协议新增一个握手流程来实现。完成初始化后，大规模智能硬件即可与平台的节点一一对应，实现平台的交互可编程。  **感知层，**智能硬件改造部分。将对传统传感器或执行器等硬件新增ESP8266芯片或ESP32芯片或A7芯片封装成智能硬件。硬件部分的软件——固件，每个智能硬件烧录统一的通信功能的固件部分，原有的数据分析功能的固件部分将转移到本项目物联网平台上。  **应用层，**交互功能开发部分。将实现大规模部署和管理控制智能硬件，能模块化编写代码，可以友好地可视化显示数据，以数据流方式连接智能硬件节点，同时平台将自带MQTT Broker和MongoDB方便数据的传输和存储。我们将参考借鉴开源的Node-RED可视化工具平台，其有以数据流的形式实现硬件的连接、模块化方式修改逻辑等优点，但无法大规模部署和管理控制智能硬件，无法灵活友好地显示数据。   * 1. **网络层，即协议适配部分。**      1. **物联网通信协议MQTT**   如下图所示，MQTT协议发送的数据包含信息部分（Message）和主题部分（Topic）。通过MQTT Broker（MQTT服务器）以广播的方式用主题（Topic）向外发送信息（Message）。MQTT Client（计算机等终端）通过监听Topic：A，接收关于该Toipc的广播就能获取到Message：1234。如此不断往复即可接收到一系列的数据。     * + 1. **智能硬件与物联网平台的握手流程**   + 什么是与平台的握手？   通俗的来说，就是智能硬件将独一无二的名字（MAC地址，类似每个人的身份证号码）告诉平台，让平台能认识该智能硬件，同时用户在平台上自定义该智能硬件的名称（类似每个人的姓名），能够在平台进行匹配。当平台想让该智能硬件做出动作的时候，就喊它的名字+命令，这时候只有该智能硬件会做出相应的动作。  **智能硬件与物联网平台的握手原理图**    用专业的术语表述，就是因为现实场景中将会存在大规模型号相同的智能执行器（继电器，马达等执行命令的智能硬件），这时候我们平台需要对他们进行管理，而平台与智能硬件之间并没有实现大规模的对应关系，所以需要一个握手配对步骤，实现本物联网平台在后续发送命令与智能执行器一一对应。实施流程如上图所示，用户将一个需要新进入平台的智能执行器放入场景中需要进行初始化，将智能执行器切换为部署模式，即按下按钮，智能执行器将以meet为Topic向平台发送本智能执行器的MAC地址，通过MQTT Broker转发给平台，平台上的MQTT接收节点便可实现与该智能执行器的联系，同时用户自定义该智能执行器的名称，Function 节点便获得了该智能执行器的自定义名称和MAC地址，以JSON格式一同发送给MongoDB（数据库）存储。完成初始化后，本物联网平台就完成了对该智能执行器的识别。退出部署模式（松开按钮），此智能执行器停止发送MAC地址，于是平台即可继续与下一个需要新进入平台的智能执行器进行配对，智能传感器同上。   * + 1. **智能硬件与物联网平台的工作流程**   **智能硬件与物联网平台的工作原理图**    这里以场景运作时候智能传感器与智能执行器是怎么相互运作的为例。如上图所示，智能传感器得到数据，将自己的MAC地址与数据值通过MQTT协议发送至平台，此时平台上的MQTT接收节点与智能传感器的Topic相同即可。Function节点得到MQTT接收节点接收到的数据值以及MAC地址后，会进行数据分析（即原本固件中的数据分析功能部分的代码）并会在MongoDB中寻找到对应智能传感器需要执行动作的智能执行器的MAC地址，此时Function节点需要将Topic设为将要控制的智能执行器MAC地址，然后将MAC地址传送至MQTT发送节点。完成以上动作之后，MQTT发送节点的Topic（MAC地址）便与需要执行命令的智能执行器的Topic（MAC地址）一致，智能执行器利用MQTT协议广播订阅需要执行的命令的信息，即可开始执行命令。   * 1. **感知层，即智能硬件改造部分。**本项目将采用最新的ESP8266模块（NodeMCU开发板）或ESP32模块或GPRS+GPS+GSM的A7模块与传统传感器进行集成封装形成智能硬件节点。另外需要开发更新一套通信固件对智能硬件进行统一烧写，计划利用Arduino IDE开发和烧写这款通信固件，技术已经成熟。此固件在智能硬件接入本物联网平台前烧写，支持智能执行器通过MQTT协议接收Topic为自身MAC地址的的数据，根据部署模式或工作模式发送Topic为自身MAC地址或“meet”字符串的信息数据，以支持智能硬件与我们的物联网平台对接。另外，硬件中原有的逻辑和数据处理部分的固件代码转移到本项目物联网平台。     **图1**  如图1所示，是能通过WiFi联网的智能传感器（智能执行器同理）简要架构图，集成有一个传统传感器芯片和一个集成ESP8266的NodeMCU处理器，能够接电源，能无线联网，输入输出信息流，同时有小型数据处理能力。    **图2**  如图2所示，是支持WiFi联网和Bluetooth近程连接的智能传感器（智能执行器同理）简要架构图，集成有一个传统传感器芯片和一个ESP32芯片，能够接电源，支持利用Bluetooth近程连接，同时能通过WiFi无线联网，输入输出信息流。    **图3**  如图3所示，是嵌入A7模块的智能传感器（智能执行器同理）简要架构图，集成有一个传统传感器芯片和一个GSM+GPRS+GPS功能的A7芯片，能够接电源，支持利用GPRS无线入网，可以GPS定位，输入输出信息流。   * 1. **应用层，即交互功能开发部分。**我们计划在本地计算机上搭建node.js框架环境，参考借鉴开源的Node-RED可视化工具平台，使本项目物联网平台成为一款WEB端物联网传感器节点管理控制交互平台，支持超声波测距、工件计数、温湿度等智能传感器（智能执行器）大规模初始化接入、批量化管理。在本平台底层代码中，利用JavaScript、HTML、CSS为各种基于 MQTT 协议的智能硬件添加可视化逻辑编程模块和 Mashup、Dashboard 、freeboard等数据图表显示模块，实现对这些智能硬件的代码逻辑编写，并将反馈回来的数据实时显示分析并支持用户对这些数据的进行处理、利用、存储，之后MongoDB接入本平台，将反馈的数据以JSON格式标准化存入数据库，以备历史查询和备份。我们计划将node.js框架下的MQTT Broker——MQTT.js嵌入到本平台并包装到后台。      1. **对智能硬件的逻辑编写**     **自定义节点模块JavaScript代码图**  智能硬件中原有的逻辑和数据处理部分的固件代码转移到了本项目物联网平台的Function节点中。本物联网平台中的Function节点使用的是JavaScript语言，这需要我们在智能传感器（智能执行器）研制之后用JavaScript代码编写原本利用C和C++编写的底层逻辑，并以数据流节点的形式保存封装，同时编写对应的HTML文件保存在节点文件夹中作为页面的布置模块。待用户需要使用时，能够以拖拽的形式摆放并修改参数之后直接使用。这就是可视化逻辑编程，本物联网平台计划嵌入开源的Google Blockly可视化编程模块作为平台的其中一个节点，用户可以根据每个智能硬件节点进行自定义设计或更改逻辑，智能硬件将数据通过MQTT Broker和平台MQTT接收节点可以反馈给本平台上的各种任务执行节点。     * + 1. **数据库系统**   数据库的选择上我们将采用MongoDB，非关系型数据库的一种。关系型数据库中的表都是存储格式化的数据结构，每个元组字段的组成都相同，但非关系型数据库是采用键值对存储，它的结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加或减少一些自己的键值对，这样就不会局限于固定的结构。这对于我们处理用户自定义标签与智能硬件的MAC地址正好做到了键值对的匹配，同时还兼容感知层传回来的数据存储，极大地提高了用户的操作性，同时数据内容是存在缓存上而非普通的关系型数据库存放在硬盘上，这样增、删、改、查操作也可减少时间和空间的开销。  **主要流程**：   * **数据获取和预处理**   智能传感器（智能执行器）中的传感器芯片在场景中不断地获取数据，并利用电子电路滤波，对数据进行简单预处理，从而在最底层就防止出现失真数据。   * **数据的传输和标准化格式**   预处理好的数据通过ESP8266芯片（ESP32、A7），利用MQTT协议传输到MQTT Broker并转发。本平台的MQTT接收节点接收到MQTT Broker转发的数据并传输给Function节点，利用事先编写好的JavaScript代码对数据进行处理和标准化格式，得到以JSON等标准化格式的数据。   * **数据存储**   本平台将有数据库发送和接收节点，可以直接对接到MongoDB，得到的JSON数据格式对应数据库中的智能传感器（智能执行器）MAC地址进行存储；也可以将智能硬件反馈回来的数据进行存储。   * **数据查询分析**   利用数据库语言对数据进行查询和分析   * **数据可视化显示**   本物联网平台将会拥有Dashboard、Freeboard等数据可视化模块，支持即时将数据可视化显示。如果想要对历史数据进行可视化，可以调用本平台数据库接收节点，将接收到的数据重新传输给可视化显示节点，实现历史数据的可视化显示。    **数据可视化流程图**   1. **研究进度安排**      1. **项目组成员分工**    * 胡子阳：项目负责人,负责整个项目的管理与实施,重点负责借鉴Node-RED开发可视化工具平台；    * 陈董锴：主要负责实现平台与传感器通过MAC地址进行准确传输；    * 颜哲锟：主要负责修改和改进库文件，包装JavaScript的Function，使之能够通过MQTT协议与平台通信，实现可视化编程；    * 朱兵：重点负责集成传感器模块和ESP8266芯片、硬件场景的搭建；    * 程竞泽：重点负责解决以Mashup、Dashboard图标方式在平台中记录显示传感器数据，实现数据可视化交互，辅助硬件场景的搭建、辅助对库文件的包装。 | | | | | | | | | | | | | |
| **三、学校提供条件**（包括项目开展所需的实验实训情况、配套经费、相关扶持政策等）  ZUCC-Google-ARM 实验室提供场地用于该项目组落地场景的搭建，提供调试设备等实验器件。 | | | | | | | | | | | | | |
| **四、预期成果**  主要开发出一款可视化物联网逻辑平台，用慧鱼，乐高等电子积木搭建一套智能工厂的沙盘模型，将各种智能传感器（智能执行器）部署其中，大量传感器联网后接入由平台提供的大量接口中，模拟工件的加工分拣过程，并在平台上对整个体系进行模块可视化编程，供用户快速设计整个流水线。使这个智能工厂场景组成一个有机整体。   * 基本功能  1. 传感器多样化，能将各方面的数据实时地通过MQTT协议广播出去，对环境进行实时监测。执行器多样化，能实时接受平台发来的MQTT消息,对命令进行执行。 2. 平台能把固件统一的任意一种智能传感器（智能执行器）快速接入到平台为其自动开放的一对一接口中去，于此同时平台可生成只针对于这个传感器的模块对象，供用户的模块化编程。 3. 平台提供各种Dashboard数据可视化图表，远程监控各种传感器的数据。  * 拓展功能  1. Android Things本地实时显示传感器数据 2. 将平台移植到树莓派上运行，使之更轻量化，硬件成本更低 3. 远程通过Firmata协议进行对个别传感器或执行器硬件固件的修改，使平台保持对低层硬件的尽可能灵活的控制。 | | | | | | | | | | | | | |
| **五、经费预算** | | | | | | | | | | | | | |
| **总经费（元）** | | | | **20000** | | | **财政拨款（元）** | | **10000** | | **学校拨款（元）** | | **10000** |
| **注：**总经费、财政拨款、学校拨款由**学校**按照有关规定核定数目进行填写 | | | | | | | | | | | | | |
| 具体包括：  1、调研、差旅费；  2、用于项目研发的元器件、软硬件测试、小型硬件购置费等；  3、资料购置、打印、复印、印刷等费用；  4、学生撰写与项目有关的申请专利费等。 | | | | | | | | | | | | | |
| **六、导师推荐意见**  签名：  年 月 日 | | | | | | | | | | | | | |
| **七、院系推荐意见**  院系负责人签名： 学院盖章：  年 月 日 | | | | | | | | | | | | | |
| **八、评审专家组意见：**  负责人签名：  年 月 日 | | | | | | | | | | | | | |
| **八、学校推荐意见：**  学校负责人签名： 学校公章  年 月 日 | | | | | | | | | | | | | |

注：表格栏高不够可增加。