

即插即用 demo 总体说明

创建时间： 2021/04/24 19: 30

作者： 1099023759@qq.com

一、环境配置

Goland、edgex-go、rt 内核、Igh Ethercat 主站、edgex 的 appservice-sdk-go 均在 Tower 上有文档。

二、前置知识

IO-Link，Igh EtherCAT，OPC UA，Edgex 设备微服务以及 APPservice，JS。

IO-Link：本工程仅仅利用 Ethercat 主站配置 IO-Link 主站为唤醒模式，IO-Link 主站和从站之间将建立通信。

以上知识不在本文件里详细描述，请从三份可执行文件的源码入手开始。

三、网络拓扑图

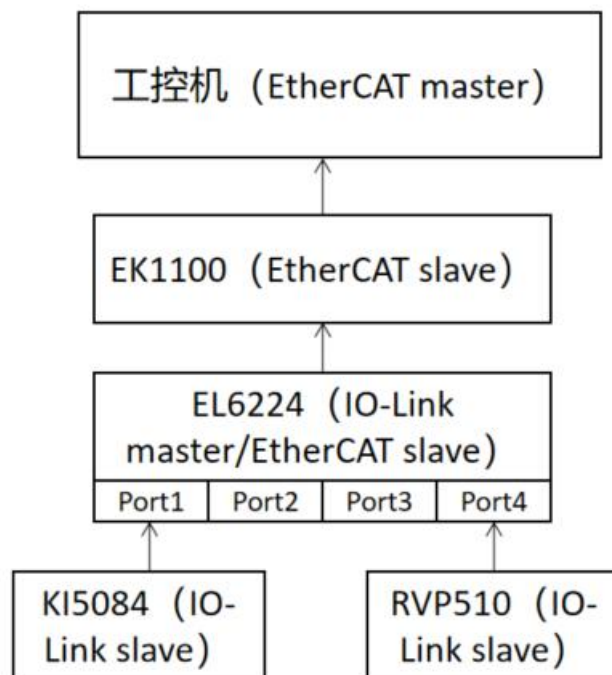


图 1-网络拓扑图

四、软件架构

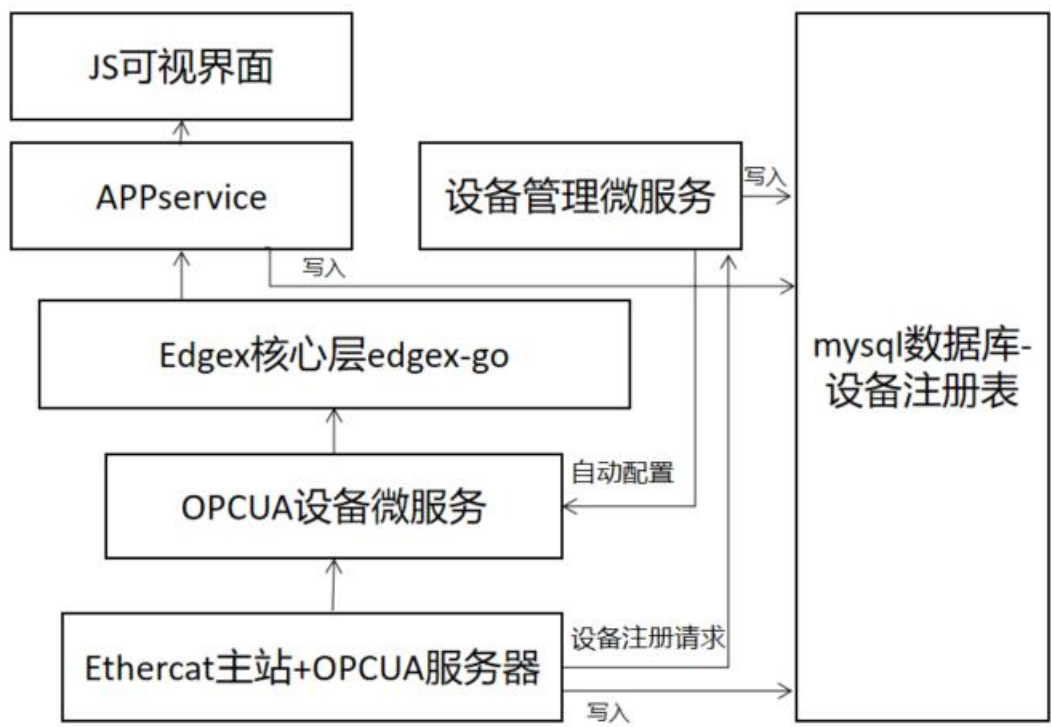


图 2-软件架构

五、Plug_and_play 文件包目录树

device-manage-service	设备管理微服务包
IODD	支持设备的IODD存放点
OPCUA调试客户端	和工程运行无关
PNP-Appservice	APPservice包
PNP-Ethercat-OPCUAserver	EtherCAT总线+OPCUA程序
sql	mysql创建表
设备微服务-单独	和工程运行无关

图 3-Plug_and_play 文件包目录树

六、文件及其说明

项目的根目录是 Plug_and_play，其中可执行文件作用如下表：

表 1-即插即用可执行文件说明

表名	角色	作用
PNP-Ethercat-OPCUAServer/EL6224	Ethercat 总线+OPCUA 服务器	Monitor 线程：监控 IO-Link 端口设备插入，加载 OPCUA 信息模型，配置 EtherCAT Ethercat_thread 线程：EtherCAT 总线数据读取

		Main 线程：运行 OPCUA 服务器
Device-manage-service/device-manage-service	设备管理模块	接口 1: IODD 解析 接口 2: 接受设备 edgex 注册请求，配置 toml 和 yaml，启动设备微服务。
PNP-Appservice/appservice	数据导出+可视化组件	线程 1: 过滤 Edgex 数据流，选择注册的 IO-Link 设备数据可视化。 向 js 界面提供： 接口 1: n 个注册设备的设备 id 接口 2: n 个 IO-Link 设备的过程数据

以上三个文件细节，请看源码完整注释。

七、演示即插即用流程

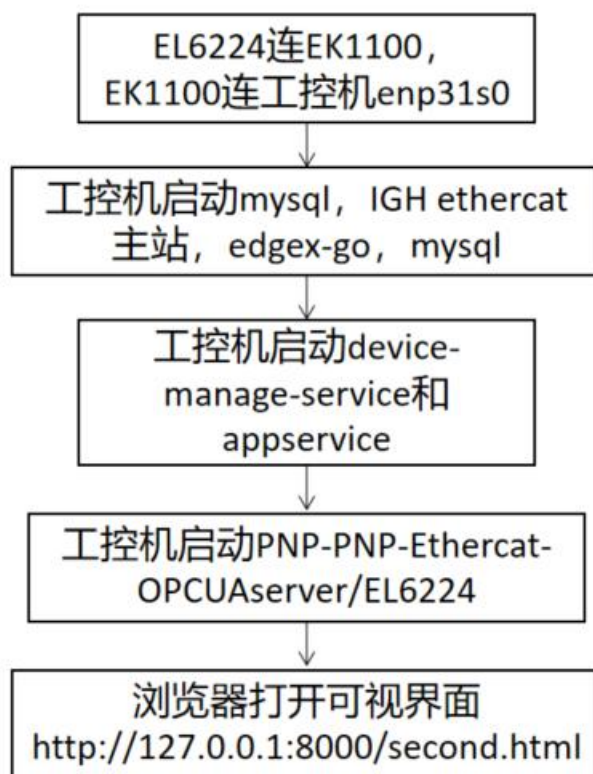


图 4-即插即用运行流程

具体运行指令可以参考目录下的 readme.txt 文件。

注意先在 EL6224 的 port4 上接入 RVP510，port1 不接入 KI5084。

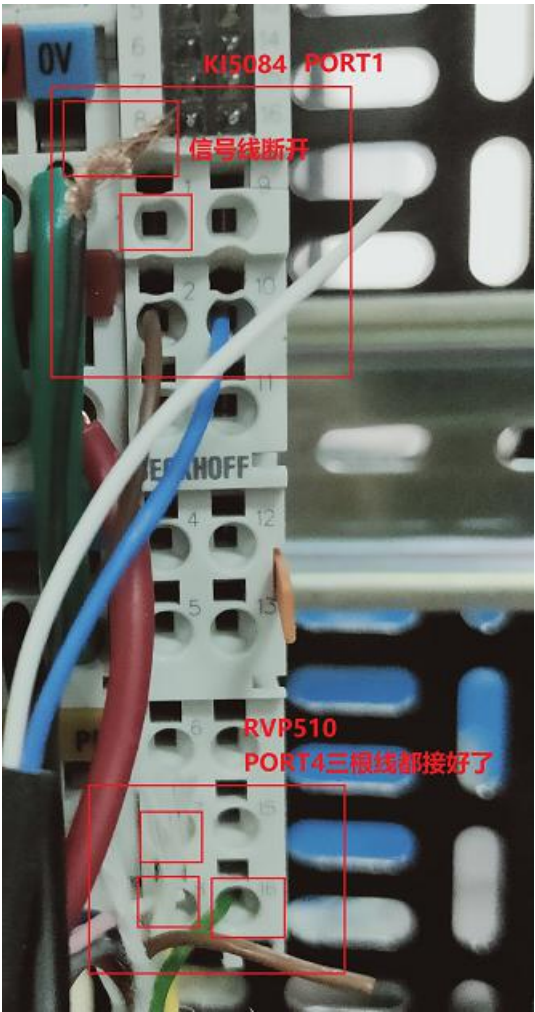


图 5-EL6224 与 IO-Link Slave 接线图

至此运行起来了程序环境。此时 EtherCAT 是 OP 状态。RVP5084 已经被自动化配置，在可视界面可以看到 RVP5084 的四个过程数据。

IO_Link_Device496_1	设备名 编码器	连接地址 62541:49601	数据名 过程数据1	数据值 = 0	断开
IO_Link_Device496_2	设备名	连接地址	数据名 过程数据2	数据值 = 0	断开
IO_Link_Device496_3	设备名	连接地址	数据名 过程数据3	数据值 = 8	断开
IO_Link_Device496_4	设备名	连接地址	数据名 过程数据4	数据值 44 =	断开

图 6-JS 可视界面

在 EL6224（IO-Link 主站）上的 port1 插口 1 号插入 KI5084 的信号线，本程序系统，将自动对其进行配置，在可视界面可以看到新增了 KI5084 的两个过程数据。

IO_Link_Device496_1	设备名 编码器	连接地址 62541:49601	数据名 过程数据1	数据值 = 0	断开
IO_Link_Device496_2	设备名 _____	连接地址 _____	数据名 过程数据2	数据值 = 0	断开
IO_Link_Device496_3	设备名 _____	连接地址 _____	数据名 过程数据3	数据值 = 8	断开
IO_Link_Device496_4	设备名 _____	连接地址 _____	数据名 过程数据4	数据值 44	断开
IO_Link_Device388_1	设备名 电容传感器	连接地址 62541:38801	数据名 过程数据1	数据值 = 0	断开
IO_Link_Device388_2	设备名 _____	连接地址 _____	数据名 过程数据2	数据值 112	断开

图 7-JS 可视界面-新设备 388 插入

即实现了设备的即插即用。

八、数据库表

为了实现设备即插即用系统中几个微服务的协作，设计了一张数据库表，如下。

```
CREATE TABLE IO_Link_Device(
  `device_id` BIGINT NOT NULL AUTO_INCREMENT COMMENT 'IO-Link 设备 id',
  `deviceservice_create_flag` int NOT NULL COMMENT '设备微服务开启标志',
  `appservice_monitor_flag` int NOT NULL COMMENT 'APP 微服务监控设备标志',
  `js_show_flag` int NOT NULL COMMENT 'JS 可视化标志',
  `io_link_port` int NOT NULL COMMENT 'IO-Link 主站 Port',
  `edgex_port` int NOT NULL COMMENT '设备微服务 Port',
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  PRIMARY KEY (device_id)
)ENGINE=INNODB AUTO_INCREMENT=1000 DEFAULT CHARSET=utf8 COMMENT='IO-Link 设备表';
```

设备插入时，总线程序将开启一行，设置设备 id 和创建时间，IO-Link 主站。

设备管理微服务获取注册请求，设置设备微服务标志，设备微服务 port。

APPservice 周期性查询数据库表，获得没被 APP 监控的设备 id，APP 微服务监控设备标开启。

九、程序交互时序图

以行时间递增的时序图。

表 2-程序交互时序图

程序外部	EL6224 内部	EL6224 与 device- manage- service 之 间	Device-manage- service	Appservice	Appservie 与 js 之间	JS 界面	Mysql 表
	启动 monitor 线程						
	初始化 ethercat 主 站						
	启动 ethercat_thread 线 程(OP)						
	启动 OPCUA 服务器						
设备插入							
	Monitor 线程提取出 设备 id						
	monitor 线程关闭 ethercat_thread 线 程（PREOP）						
	Monitor 线程	POST Deviceid 过程数据 OPCUA 节点- -->					
	数据库表插入设备 id 为主键的一行		解析 IODD				Deviceid

			1. 配置 toml、选择 yaml。 2. 以过程数据 OPCUA 节点为参数，启动设备微服务。 3. 数据库表设备 id 行置 deviceservice 标志为 1。	1. 查 mysql 表，启动对设备名的监控（选择 edgex 数据流）。 2. Mysql 表 APPservice 标志位置 1			Deviceid 和 deviceservice 标志位 1
							Deviceid、deviceservice 标志位 1 和 APPservice 标志位 1
				得到 IO-Link 设备的数据流			
				长度 n 的双向链表存储 n 个 IO-Link 设备的数据流			
					← 获取注册设备 id		
						可视化设备窗口	
					← 获取注册设备过程数据流		
						设备窗口呈现过程数据	

