目前遇到的问题很多

编译英豪push到gitlab上面的代码时总是编译不成功

原来是因为devel和build这两个文件

直接删掉后编译，有显示无法找到Person.h 其实这种情况下直接source一下就可以了

简单总结完成代码就是

rm devel/\* -rf

rm build/\* -rf

catkin\_make （失败是因为没有更新环境变量）

source devel/setup.bash 更新环境变量

catkin\_make 成功

1. 如何实现modbus和mqtt的转换？

首先删掉了build和devel文件

将libmodbus库移到3rdeparty里边并在其中的CMakelists中添加进去了，然后source后，重新编译

因为已经写过modbus的收发，与从机建立联系后是可以读取从机点位地址的，可执行文件为client

英豪写的发布文件是publish.cpp

我的调用函数接收结构体是写在client.cpp里边

现在想在publish.cpp里边直接将结构体内的变量取出来转成CJon格式完成发送

但是如何让client.cpp与publish.cpp建立联系？

或许应该将client.cpp 里边接受结构体调用接口函数放到publish.cpp 里边但是这样的话，原本client.cpp里边的与ip地址联系的函数怎么移植

而且对于发布mqtt暂时还没看所以应先把发布原理弄懂

感觉是libmodbus库的问题，因此在3r工程里边的CMakelists.txt添加了libmodbus

在publish.cpp管理的同级目录中的CMakelist.txt 中添加了libmodbus库文件

现在编译已经进行到100% 但是仍然报错如下：

/usr/bin/ld:CMakeFiles/publish.dir/src/interface.cpp.o: in function Fetch\_Device\_Id(\_modbus\*)':

interface.cpp:(.text+0x0):multiple definition of `Fetch\_Device\_Id(\_modbus\*)'; CMakeFiles/publish.dir/src/publish.cpp.o:publish.cpp:(.text+0x1f): first defined here

暂时不知如何解决

出现这个错误原来是因为我在 interface.cpp中添加了client.hpp 头文件造成了重复所以报错

由此总结一下：

若是添加进新的库时，需要在项目的三方库相应的CMakelists.txt中添加相应代码

同时在生成可执行文件的CMakelists.txt中也许加入库文件的索引

这也是最关键的一步

target\_link\_libraries(publish ${catkin\_LIBRARIES} ${MQTT\_OCC} cJSON /home/wangxudong/6.2/src/3rdparty/libmodbus/install/lib/libmodbus.so)就是因为红色字体刚开始没有写进去所以才会报错！！！！

现在已经能编译通过了但是想实现收发需要实验

还有一个疑问就是我的程序运行一次就可以讲数据储存进去了，在运行publish时是否还需要运行client

问题解决 ： 现在把client里边和从机联系的代码写进publish里边，并在publish里边写接收的函数直接调用，此时结构体变量应该是已经被赋值过的，然后将收到的结构体里边的变量转成CJon格式的，然后通过解析xml文件里边的地址与英豪电脑进行连接（地址使用的是英豪电脑的地址 端口是1883 有规定 接收那边是8083暂时不知道是为什么）

？？？？？？？？？？？

当发送的结构体内部变量是用数组存放的时候，又应该怎么办？？？？

之所以怎样都找不到函数的实现还是因为写的不规范，函数实现在生成可执行文件

add\_executable(publish src/publish.cpp src/interface.cpp src/tinyxml2.cpp /home/wxd/6.2/src/3rdparty/libmodbus/client.cpp)这条命令必须要添加进去client.cpp文件。因为函数是在client.h中声明的，但是添加client.cpp里边含有main函数

所以正确的写法是将main函数单独放一个文件。！！！！！！

写的modbus\_t \*ctx 是TCP建立连接的实例化对象

放在main函数中声明但是在写接口函数时内部也有 用到了ctx，所以要把ctx作为参数传到写的接口函数中比如

char \*func\_Device\_Id(modbus\_t \*ctx)

{

char \*Device\_Id\_msg = NULL;

Device\_Id Device\_id = Fetch\_Device\_Id(ctx);

cJSON \*Device\_id\_pRoot = cJSON\_CreateObject();

cJSON\_AddNumberToObject(Device\_id\_pRoot, "device\_id", Device\_id.device\_id);

Device\_Id\_msg = cJSON\_Print(Device\_id\_pRoot);

return Device\_Id\_msg;

}

一开始我传的是Device\_Id\_msg，但其实好像是没什么用 应该传ctx 这样问题解决了

下面需要写心跳包之类的

**7月3日**

现在是进行位移运算的操作

两个uint16要组合成一个uint32的

比如发送数值100000 一个寄存器只能存60000多

所以需要两个寄存器

结果一个数据是41094 一个是256 这就是100000拆分到每个寄存器后显示的值

100000的二进制可以表示为

10100000

10000110

00000001

00000000

41094为

10100000

10000110

256为

00000001

00000000

这样两个寄存器都有数，但是怎样把两个寄存器内的数还原成100000发送出去目前成为问题？？？？

<https://blog.csdn.net/weixin_44386927/article/details/114839080?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-114839080-blog-117185406.pc_relevant_multi_platform_whitelistv2&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-114839080-blog-117185406.pc_relevant_multi_platform_whitelistv2&utm_relevant_index=1>

这篇博客记录转换类型的

Modbus Tcp协议读取浮点型

参考博客

[https://blog.csdn.net/Polumgla/article/details/85254709?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165683400516781647562272%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=165683400516781647562272&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduend~default-1-85254709-null-null.142^v30^down\_rank,185^v2^control&utm\_term=modbus%E8%AF%BB%E6%B5%AE%E7%82%B9%E6%95%B0&spm=1018.2226.3001.4187](https://blog.csdn.net/Polumgla/article/details/85254709?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522165683400516781647562272%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=165683400516781647562272&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-85254709-null-null.142%5ev30%5edown_rank,185%5ev2%5econtrol&utm_term=modbus%E8%AF%BB%E6%B5%AE%E7%82%B9%E6%95%B0&spm=1018.2226.3001.4187)

目前已经将数据都读好了，但是接收中移数据还没有思路

DataUnit.cpp 里有个#include<random>

如果不包含就会报错：‘float\_t’ was not declared in this scope; did you mean ‘float’?

**7月4日**

现在需要用到回调函数

云端发布数据，用回调函数接收后返回message

用cjson解析后就可以取出来

这篇博客可以参考mosquitto回调函数 订阅 发布

<https://kunaly.blog.csdn.net/article/details/107466430?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-107466430-blog-119089359.pc_relevant_multi_platform_whitelistv1&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-107466430-blog-119089359.pc_relevant_multi_platform_whitelistv1&utm_relevant_index=1>

用完之后发现可以从eqmx往程序上发数据了，但是如何将数据提出来，转成modbus 写到保持寄存器里边

发现一篇博客用c解析josn的----

现在进行尝试

[https://blog.csdn.net/qq\_43742016/article/details/105393545?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165691592616782389490089%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=165691592616782389490089&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_ecpm\_v1~rank\_v31\_ecpm-1-105393545-null-null.142^v30^control,185^v2^control&utm\_term=%E8%A7%A3%E6%9E%90josn%E6%95%B0%E6%8D%AE%E5%8C%85&spm=1018.2226.3001.4187](https://blog.csdn.net/qq_43742016/article/details/105393545?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522165691592616782389490089%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=165691592616782389490089&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-1-105393545-null-null.142%5ev30%5econtrol,185%5ev2%5econtrol&utm_term=%E8%A7%A3%E6%9E%90josn%E6%95%B0%E6%8D%AE%E5%8C%85&spm=1018.2226.3001.4187)

回调函数的调用还没有搞太明白，但是

在创建客户端的时候 将要的数据写进去

Int value=10；

创建客户端

mosq = mosquitto\_new(NULL, session, &value); 填入地址

因为发数据就调用到

void my\_message\_callback(struct mosquitto \*mosq, void \*userdata, const struct mosquitto\_message \*message)

这个void \*userdata 就是传进来的指针

在内部接收一下

int \*value\_ptr = (int\*)userdata;

printf("This is %d\n", \*value\_ptr);

只要emqx一发送数据就会打印 “This is 10”

所以在其内部传入结构体并进行赋值

可以实现云端发出json数据包并解析后写入模拟的从机点位，但是在我运行程序的时候，从云端发的数据可以写进充电设备内，但是却读不到当前写入的值，读取寄存器的值是之前存放的数据，所以这可能涉及到线程问题，需要解决。

9.7

听说马上就要开始测试了，将之前的代码拿出来跑一下结果真的出问题了，emqx平台发数据始终发不过去，最后原来是因为发的数据格式不对 应该是这样的格式

{"Interval\_C": 50, "SCDControlMode": 100 }

切记！！！！

7.26

大概浏览了一边王乐乐Linux基础操作指令

想搞懂峰哥代码里边的3rdparty内的CMakelists.txt是如何被根目录下管理的？

每个三方库都有各自的CMakelists.txt 然后包含到3rdparty里边最后被根目录管理，始终是没搞懂。

代码模块化添加库文件可以参考博客

https://blog.csdn.net/chen1234520nnn/article/details/116933889?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522165879817016781667836736%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=165879817016781667836736&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~top\_click~default-1-116933889-null-null.142^v34^new\_blog\_fixed\_pos,185^v2^control&utm\_term=cmakelists.txt%E6%80%8E%E4%B9%88%E7%94%A8&spm=1018.2226.3001.4187