# 高级时钟项目架构设计

## 服务端建立

（QT上位机做服务器），

## 终端设备设计

终端连接WIFI，然后传输心跳包，服务器下发GMT时间+天气状况。上位机发送指令操作，终端响应数据包。（LED的闪烁和屏幕的显示）Json数据的解析以及http的请求理解。

串口连接就是串口发送，WIFI连接WIFI发送，手机蓝牙的协议也做。

### WIFI收发数据机制（数据包解析）

#### 数据接收和包解析

#### ESP8266简介

#### ESP826初始化

#### ESP8266连接网络及连接服务器

#### ESP8266获取网络时间

#### ESP8266获取实时天气

### 蓝牙的字符串解析

### NB模组的使用

### NRF24L01模组的使用

### OLED时间和天气的显示



### 界面设计

## 协议设计

协议格式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头  （4B） | 帧地址  （2B） | 帧指令码  （2B） | 包序号（2B） | 数据区长度（2B） | 数据区  （N） | 校验  （4B） |

帧头：gjsz（高级时钟） 0x676A737A

帧地址：202011070001 时间+编号

帧指令码：0x01-0x10

包序号：每次+1

校验：和校验

### 心跳包（终端）0x01

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x01 | 包序号 | 24 | 数据区 | 校验 |

数据区长度：24字节

数据区：温度数据（1B）+湿度数据（1B）+保留字节（16B）

### 心跳包（服务器）0x01

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x01 | 包序号 | 1 | 数据区 | 校验 |

数据区长度：20字节

数据区：00：成功 其他：errcode

### 图片包（服务器）0x02

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x02 | 包序号 | 64 | 数据区 | 校验 |

数据区长度：64字节

包序号：每次+1

数据区：图片

### 图片包（终端）0x02

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x02 | 包序号 | 1 | 数据区 | 校验 |

数据区长度：1字节

数据区：00：成功 其他：errcode

### 升级包（服务器）0x03

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x03 | 包序号 | 64 | 数据区 | 校验 |

数据区长度：64字节

包序号：每次+1

数据区：升级包

### 升级包（终端）0x03

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x03 | 包序号 | 1 | 数据区 | 校验 |

数据区长度：1字节

数据区：00：成功 其他：errcode

### 跳转（服务器）0x04

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x04 | 包序号 | 0 | 数据区 | 校验 |

数据区长度：0字节

数据区：空

### 跳转（终端）0x04

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x676A737A | 帧地址 | 0x04 | 包序号 | 1 | 数据区 | 校验 |

数据区长度：1字节

数据区：00：成功 其他：errcode

## ZLG awtk GUI界面使用学习

### 编译工具链

Scons、cmake、vs2019(msvc)、nodejs、

**Nodejs：**基于Chrome V8引擎的JavaScript运行环境。

主要做web开发为，微服务，前端构建；开发前端库；开发低延迟网络应用。

安装：windows 64位 .msi格式安装包

Msi格式 是windows installer开发出来的程序安装文件，将所有安装文件的内容封装在一个包里。

**VS2019编译错误：**

error LNK2038: 检测到“RuntimeLibrary”的不匹配项: 值“MDd\_DynamicDebug”不匹配值“MD\_DynamicRelease”

原因：就是编译选项的debug和release没有选择正确，lib库与工程要选择一致。

多线程调试Dll (/MDd) 对应的是MD\_DynamicDebug

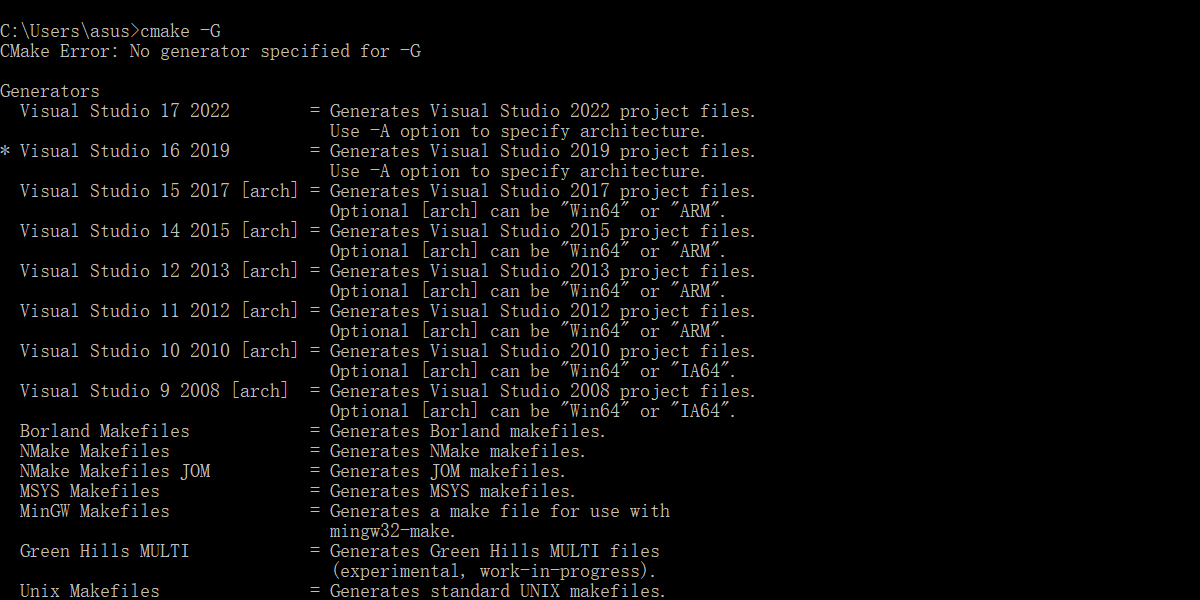
多线程Dll (/MD) 对应的是MD\_DynamicRelease

多线程(/MT) 对应的是MD\_StaticRelease

多线程(/MTd)对应的是MD\_StaticDebug

cmake .. -G "Visual Studio 16 2019 Win64"

cmake .. -G "Visual Studio 16 2019" -A x64



error MSB8020: 无法找到 v140 的生成工具(平台工具集 =“v140”)：版本选择错误。

是由于没有安装v142工具集，其实v142对应的是VS2019，我的项目从另外一台电脑copy过来的，之前的编译器VS2019，现在是VS2017，所以报的这个错误。

v142–>VS2019

v141–>VS2017

v140–>VS2015

v120–>VS2013

MS VC++ 14.0 \_MSC\_VER = 1900 vs2015

MS VC++ 12.0 \_MSC\_VER = 1800 vs2013的编译器的平台是v120

MS VC++ 11.0 \_MSC\_VER = 1700 vs2012的编译器的平台是v110

MS VC++ 10.0 \_MSC\_VER = 1600 Visual C++ 2010

MS VC++ 9.0 \_MSC\_VER = 1500 Visual C++ 2008

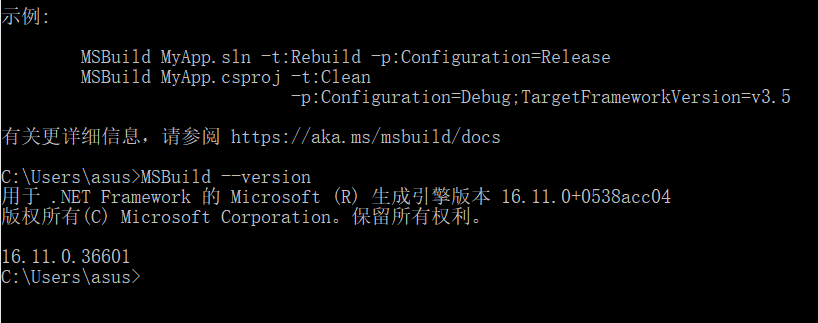
MS VC++ 8.0 \_MSC\_VER = 1400 Visual C++ 2005

MS VC++ 7.1 \_MSC\_VER = 1310

MS VC++ 7.0 \_MSC\_VER = 1300

MS VC++ 6.0 \_MSC\_VER = 1200

MS VC++ 5.0 \_MSC\_VER = 1100



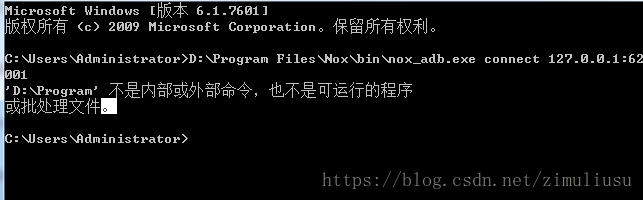
D:\LearningResource\DesigneProject\AdvancedClock\awtk\bin\themegen.exe

D:\LearningResource\DesigneProject\AdvancedClock\awtk-examples\HelloWorld.Xml-Demo\res\assets\default\raw\styles\keyboard.xml

D:\LearningResource\DesigneProject\AdvancedClock\awtk-examples\HelloWorld.Xml-Demo\res\assets\default\inc\styles\keyboard.data



如果文件名之间有空格，则会提示对应的exe程序找不到（不是内外部指令），windowws下面以空格来区分执行的指令和参数。



.Inc 文件 include file，通常在文件末尾包含进来，就相当于将该文件扩展进来。

### Scons

类似于Cmake，可以组织生成编译文件。

### MinGW

MinGW 的全称是：Minimalist GNU on Windows，GCC的windows版本。GCC的32位停留在4.8.1版本，MInGW-w64版本一直在更新。

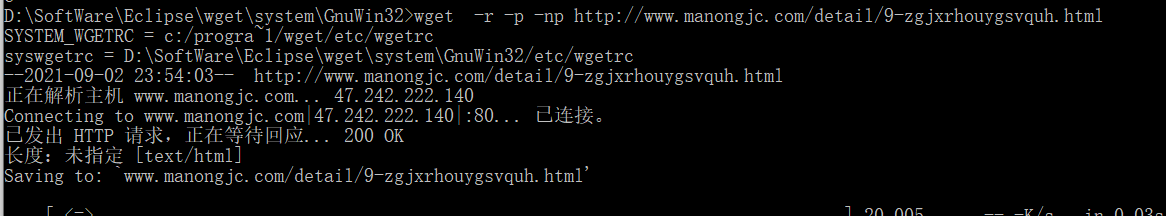
Msys：minimal GNU（POSIX） system on windows，小型的GNU环境，基本的bash和make。

利用Bash环境可以进行Linux指令操作。比如Wget操作：

wget命令格式详解：  
　　wget [参数列表] [目标软件、网页的网址]

　　-o,--output-file=FILE 将软件输出信息保存到文件  
　　-a,--append-output=FILE将软件输出信息追加到文件  
　　-d,--debug显示输出信息  
　　-q,--quiet 不显示输出信息  
　　-i,--input-file=FILE 从文件中获取URL

　　-O, --output-document=FILE下载文件保存为别的文件名  
　　-nc, --no-clobber 不要覆盖已经存在的文件  
　　-N,--timestamping 只下载比本地新的文件  
　　-T,--timeout=SECONDS 设置超时时间  
-nd, --no-directories 不建立目录  
　　-x,--force-directories 强制建立目录



### Cmake

### Awttk 界面生成

### Awtk运行

### Awtk 控件窗口建立