

MICROCHIP 2019 IoT MASTERS

中国技术精英年会(第二十届)

C20L15 IoT45

使用基于PIC®/AVR® MCU 的 IoT
解决方案快速地实现云端连接



MICROCHIP

© 2019 Microchip Technology Incorporated版权所有。

C20L15 IoT45



SMART | CONNECTED | SECURE

幻灯片1

课程目标

完成本课程之后，您将能够

- 了解物联网解决方案的云端架构
- 了解基于Microchip PIC[®]/AVR[®] MCU的IoT解决方案如何快速进行IoT原型开发
- 使用MCC生成PIC/AVR IoT参考代码
- 连接IoT节点至阿里云
- IoT节点与阿里云交换传感器数据和控制数据



课程安排

- **IoT简介**
- **IoT的基本原理**
- **基于Microchip PIC[®]/AVR[®] MCU的IoT解决方案**
- **使用PIC/AVR IoT开发板连接云**
- **推送数据至阿里云**
- **订阅数据并从云端接收数据**
- **总结与答疑**



课程安排

- **IoT简介**
- IoT的基本原理
- 基于Microchip PIC®/AVR® MCU的IoT解决方案
- 使用PIC/AVR IoT开发板连接云
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑

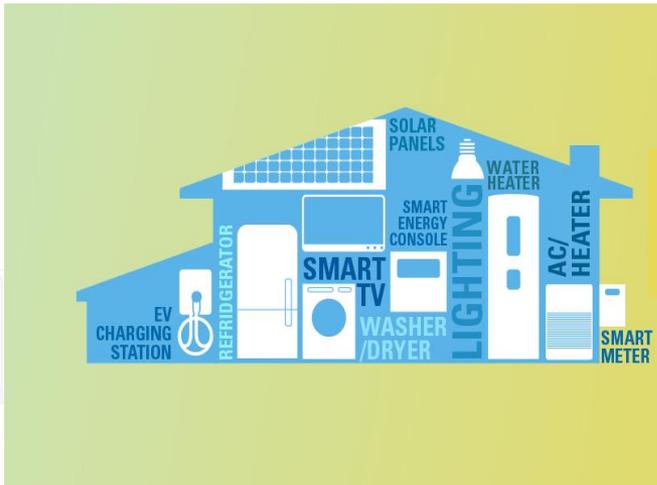
什么是IoT?



IoT应用示例



智慧城市



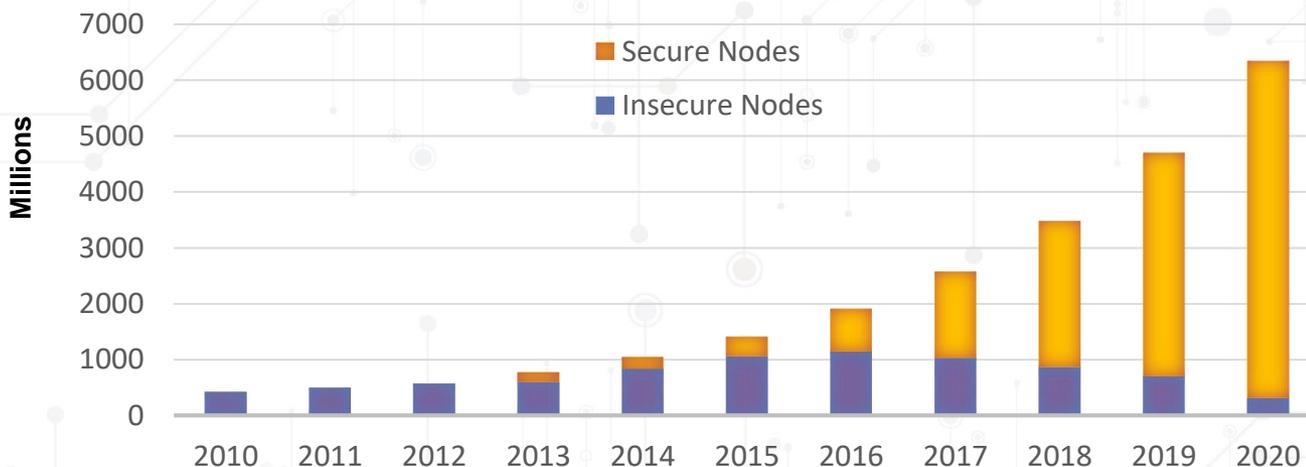
智能家居



智能医疗



IoT节点的需求



IoT节点出货量（单位：百万）

对安全的IoT节点需求日益增长

来源: Cisco, Ericsson, Gartner, IDC, Harbor, ABI, IHS, Strategy Analytics, BI Intelligence



为什么需要安全的IoT节点？

8.4B

The number of connected “things”
in use in 2017, up 31% from 2016*

攻击面巨大



Google Cloud Platform

Confidential & Proprietary

8

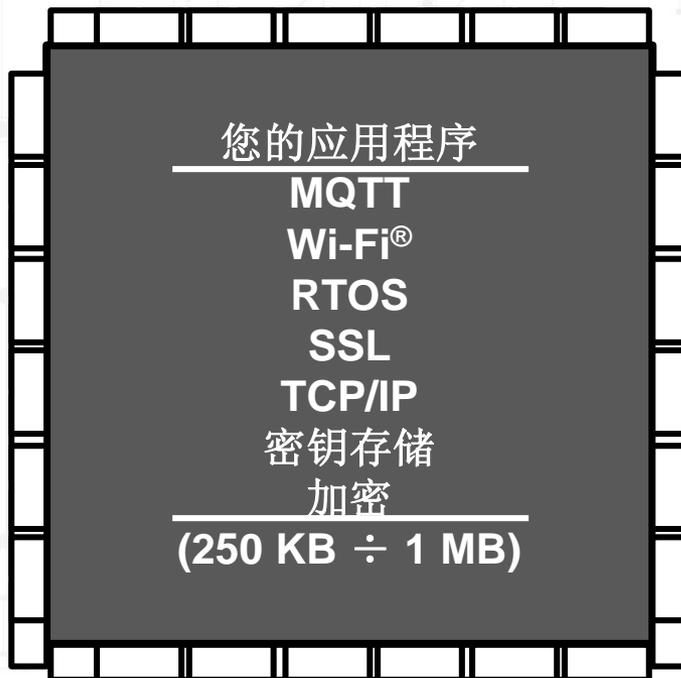


课程安排

- IoT简介
- **IoT的基本原理**
- 基于Microchip PIC®/AVR® MCU的IoT解决方案
- 使用PIC/AVR IoT开发板连接云
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑



IoT嵌入式应用的挑战—— 网络多层协议



准备开始调试**500K**的代码？

分而治之—— 减少MCU的负担并增加安全性



安全
元件

密码存储
身份认证
加密

智能
单片机

用户应用程序
MQTT
传感器控制

连接
WINC

Wi-Fi®
TCP/IP

aws



Google
Cloud IoT Core



AliCloud

- 为什么要选择IoT云服务商？
 - 可定制
 - 全球服务器部署
 - 低延迟，高可靠性
 - 众多的IoT参与者
 - 产品快速上市



网络数据传输示例

No.	Time	Source	Destination	Protocol	Length	Info
6	0.080559	192.168.42.2	139.196.67.150	TLSv1.2	303	Client Hello
8	0.142118	139.196.67.150	192.168.42.2	TLSv1.2	1334	Server Hello
12	0.143446	139.196.67.150	192.168.42.2	TLSv1.2	310	Certificate [TCP segment of a reassembled PDU]

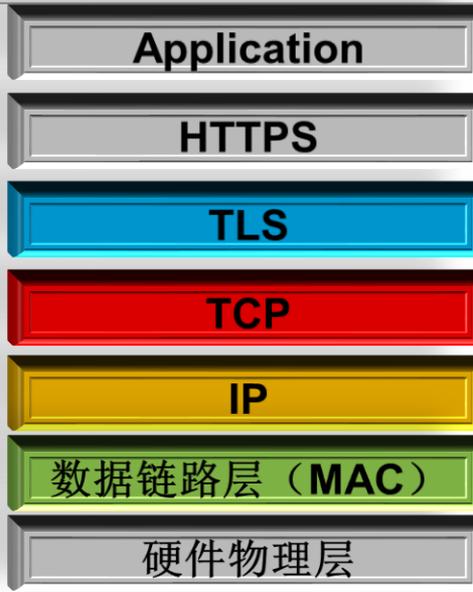
- > Frame 6: 303 bytes on wire (2424 bits), 303 bytes captured (2424 bits) on interface 0
- > Ethernet II, Src: MS-NLB-PhysServer-06 42:31:33:4f (02:06:42:31:33:4f), Dst: ba:dc:fc:98:31:af (ba:dc:fc:98:31:af)
- > Internet Protocol Version 4, Src: 192.168.42.2, Dst: 139.196.67.150
- > Transmission Control Protocol, Src Port: 61017, Dst Port: 443, Seq: 1, Ack: 1, Len: 249
- ▼ Transport Layer Security
 - ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 244
 - > Handshake Protocol: Client Hello

```

0000 ba dc fc 98 31 af 02 06 42 31 33 4f 08 00 45 00
0010 01 21 30 35 40 00 80 06 0f 9d c0 a8 2a 02 8b c4
0020 43 96 ee 59 01 bb d9 9c 18 b3 50 11 7c 64 50 18
0030 22 38 11 bb 00 00 16 03 03 00 f4 01 00 00 f0 03
0040 03 5c d3 eb 08 2b bc f9 67 6d a2 20 c7 59 91 f9
0050 7a 25 00 4a 48 6d fb 07 31 20 9e be 69 7d 69 a1
0060 fc 00 00 64 c0 24 c0 28 00 3d c0 26 c0 2a 00 6b
0070 00 6a c0 0a c0 14 00 35 c0 05 c0 0f 00 39 00 38
0080 c0 23 c0 27 00 3c c0 25 c0 29 00 67 00 40 c0 09
0090 c0 13 00 2f c0 04 c0 0e 00 33 00 32 c0 2c c0 2b
00a0 c0 30 00 9d c0 2e c0 32 00 9f 00 a3 c0 2f 00 9c
00b0 c0 2d c0 31 00 9e 00 a2 c0 08 c0 12 00 0a c0 03
00c0 c0 0d 00 16 00 13 00 ff 01 00 00 63 00 0a 00 16
00d0 00 14 00 17 00 18 00 19 00 09 00 0a 00 0b 00 0c
00e0 00 0d 00 0e 00 16 00 0b 00 02 01 00 00 0d 00 16
00f0 00 14 06 03 06 01 05 03 05 01 04 03 04 01 04 02
0100 02 03 02 01 02 02 00 17 00 00 00 00 00 21 00 1f
0110 00 00 1c 69 64 32 2e 63 6e 2d 73 68 61 6e 67 68
0120 61 69 2e 61 6c 69 79 75 6e 63 73 2e 63 6f 6d
  
```

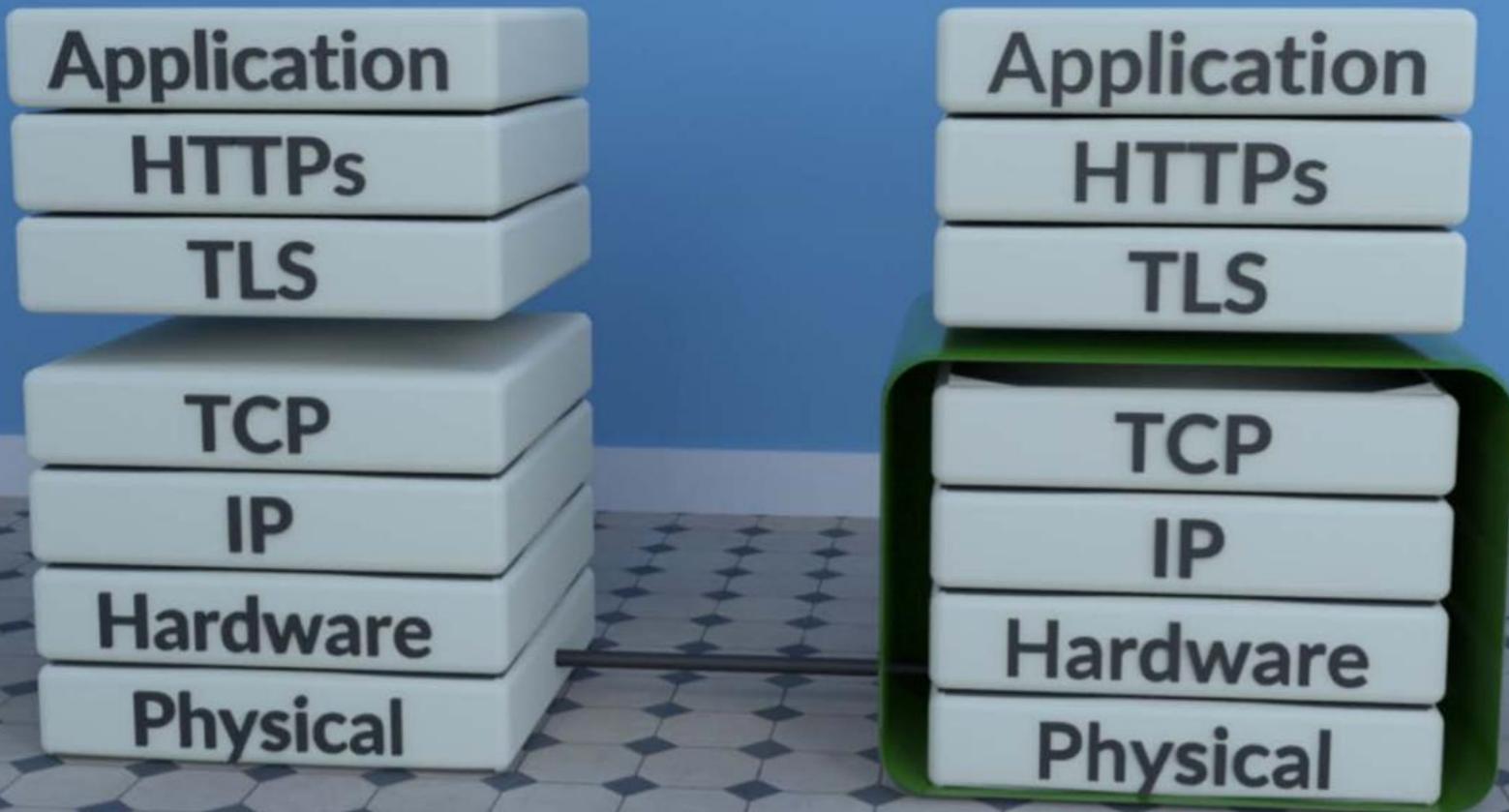
```

....1... B130..E-
!05@... ..*...
C..Y... ..P.|dP-
"8.....
.\...+... gm..Y..
z%.JHm.. 1 ..i}i-
...d.$.( =.&*.k
-j.....5 .....9.8
#.'<.% .)g@..
.../.... -3.2.,.+
-0....-2 ...../..
--1....
.....c....
.....!..
...id2.c n-shangh
ai.aliyu ncs.com
  
```



IoT网络层中的TLS是什么？

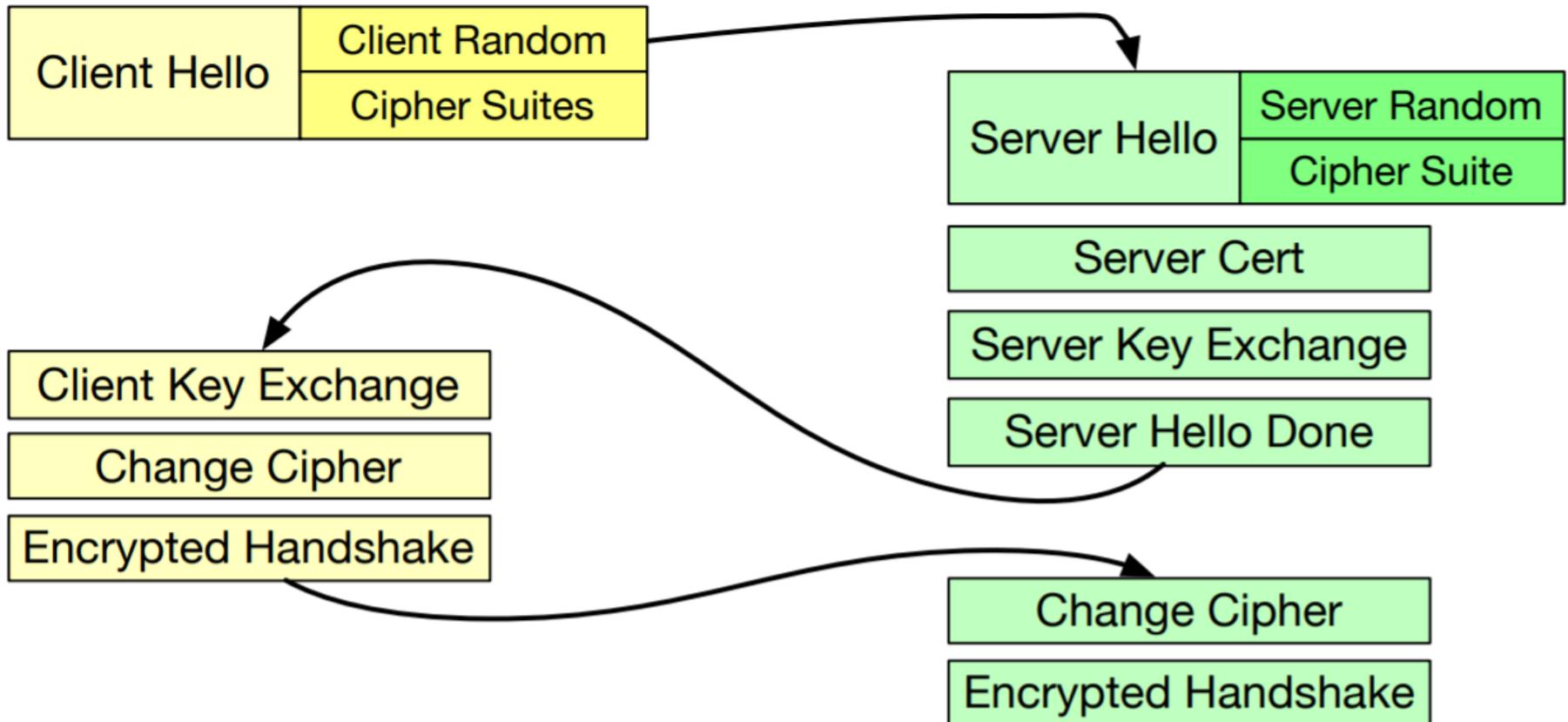
Network Layers + TLS





TLS的握手流程

TLS Handshake



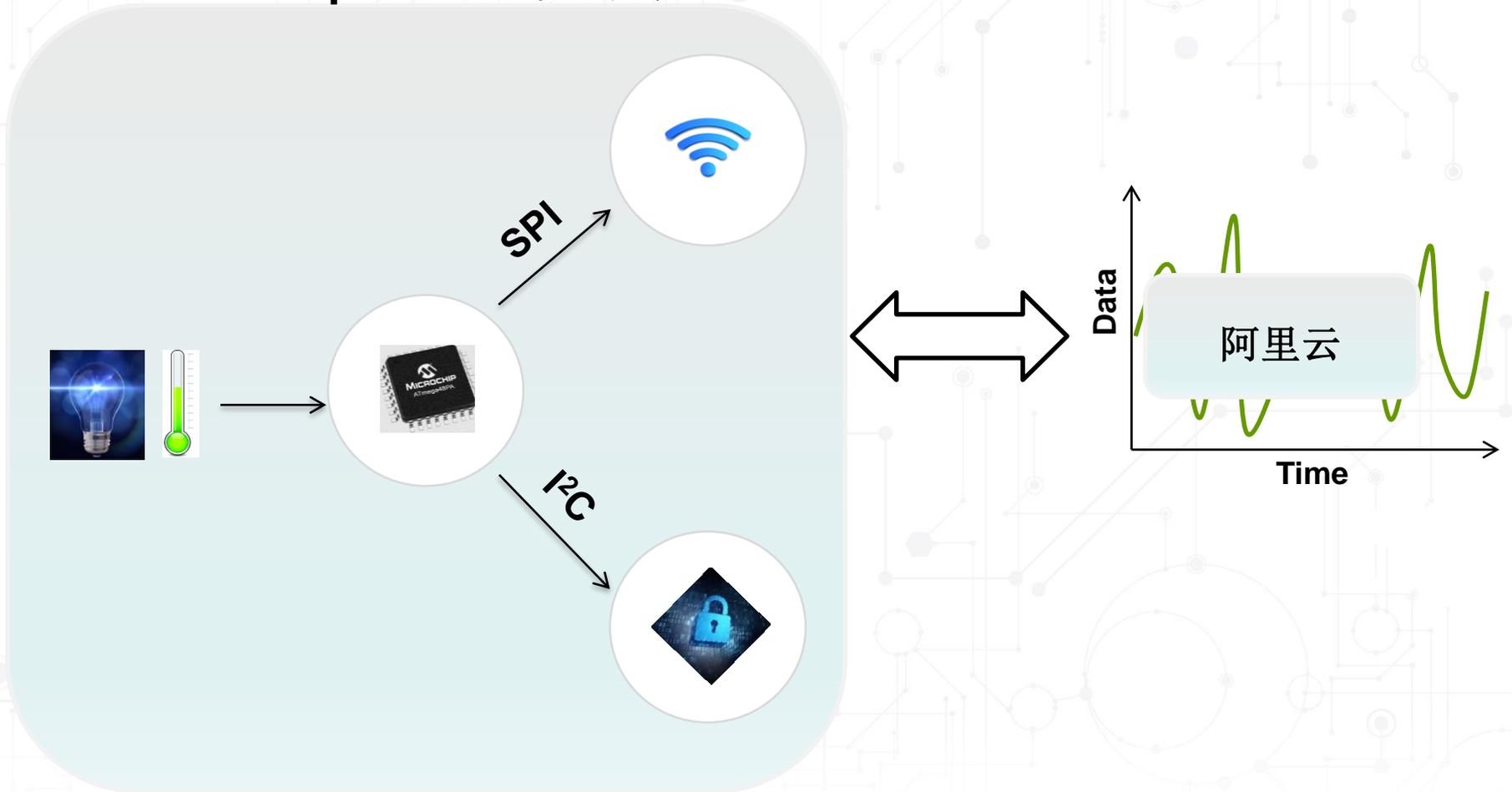


课程安排

- IoT简介
- IoT的基本原理
- **基于Microchip PIC[®]/AVR[®] MCU的IoT解决方案**
- 使用PIC/AVR IoT开发板连接云
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑

IoT传感器/执行器节点

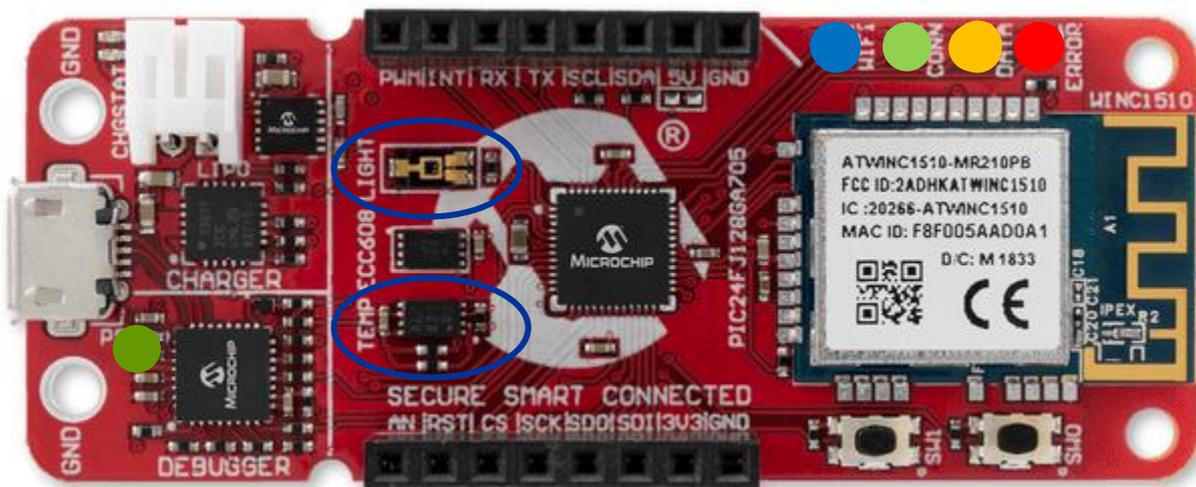
Microchip的IoT参考开发板





基于PIC®/AVR® MCU的IoT应用代码 之执行流程

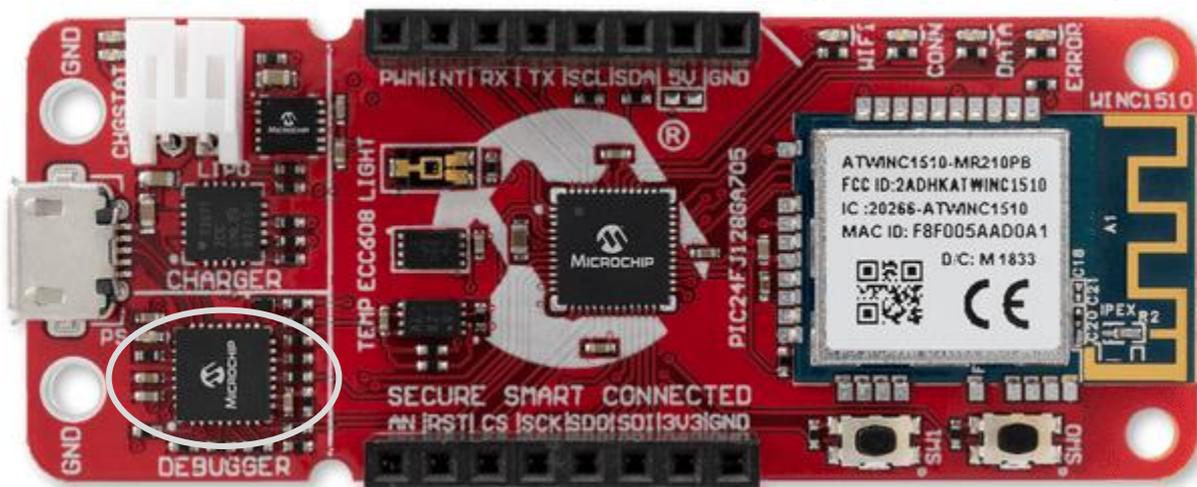
- 初始化MCU
- 连接Wi-Fi®至网络
- 打开一个TCP端口
- 与云端服务器建立MQTT连接
- 交换数据





PIC®/AVR® MCU IoT WG板 的调试

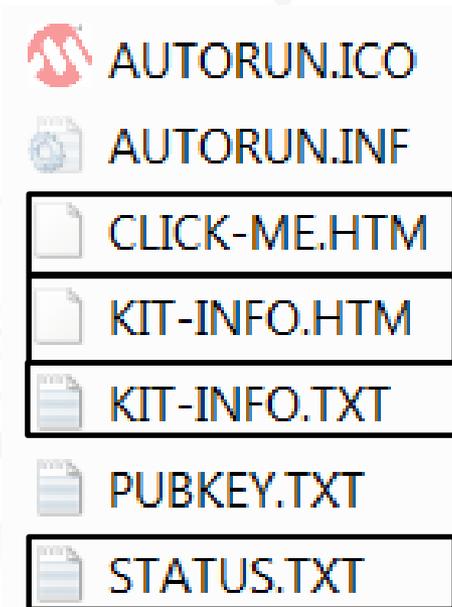
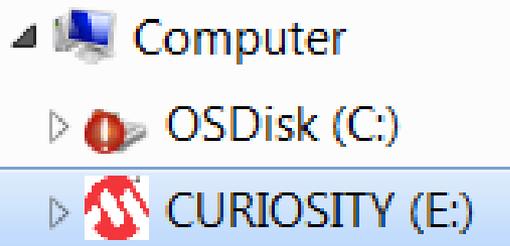
- **PKOB Nano**
 - 调试器/编程器
 - **USB转串口**
 - 模拟U盘





PIC/AVR-IoT开发板的 PKOB Nano

- PKOB Nano U盘存储



指向**PIC/AVR-IoT**网页配置界面

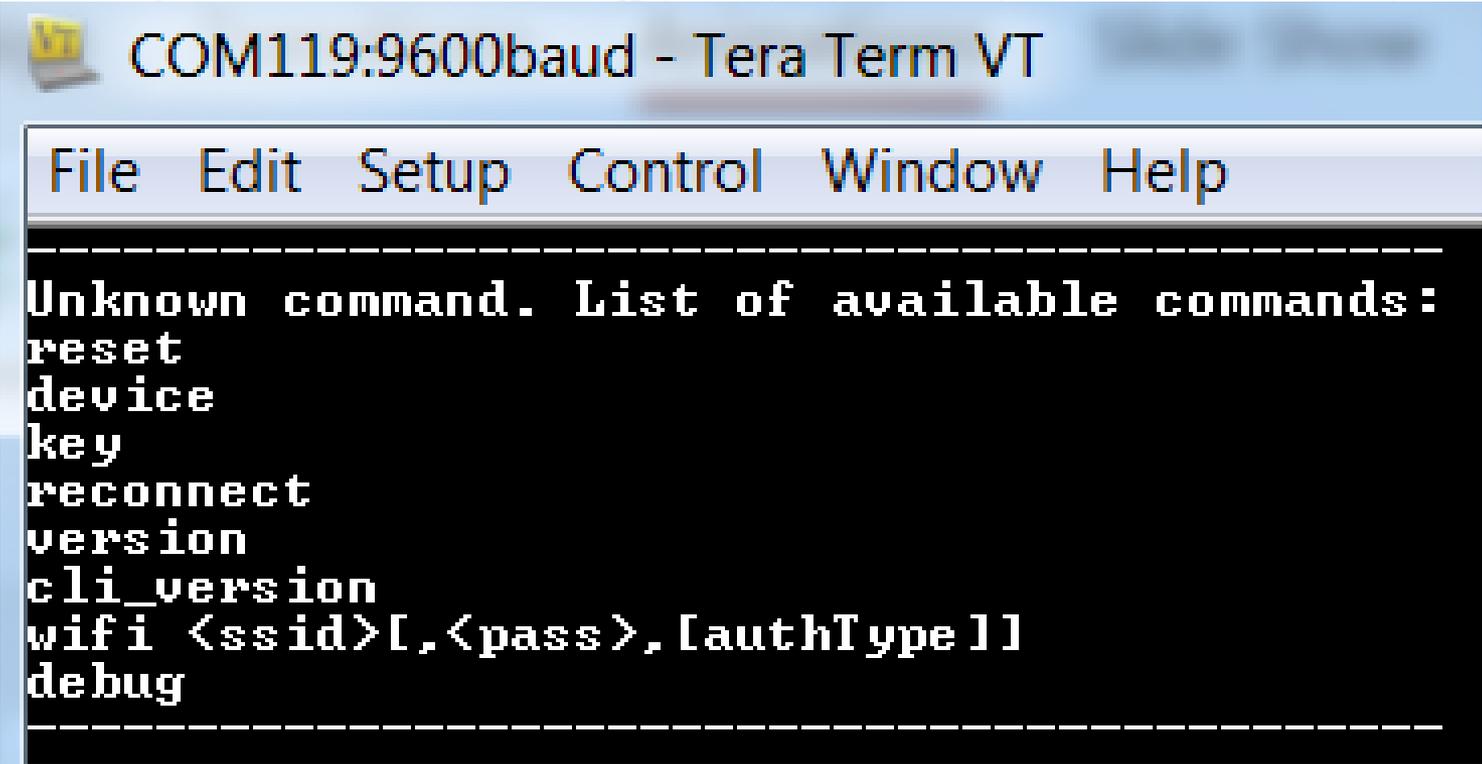
PIC/AVR-IoT开发板信息和资源

开发板序列号和固件信息

开发板状态

串口命令行 (CLI)

- **PKOB Nano串口命令行**



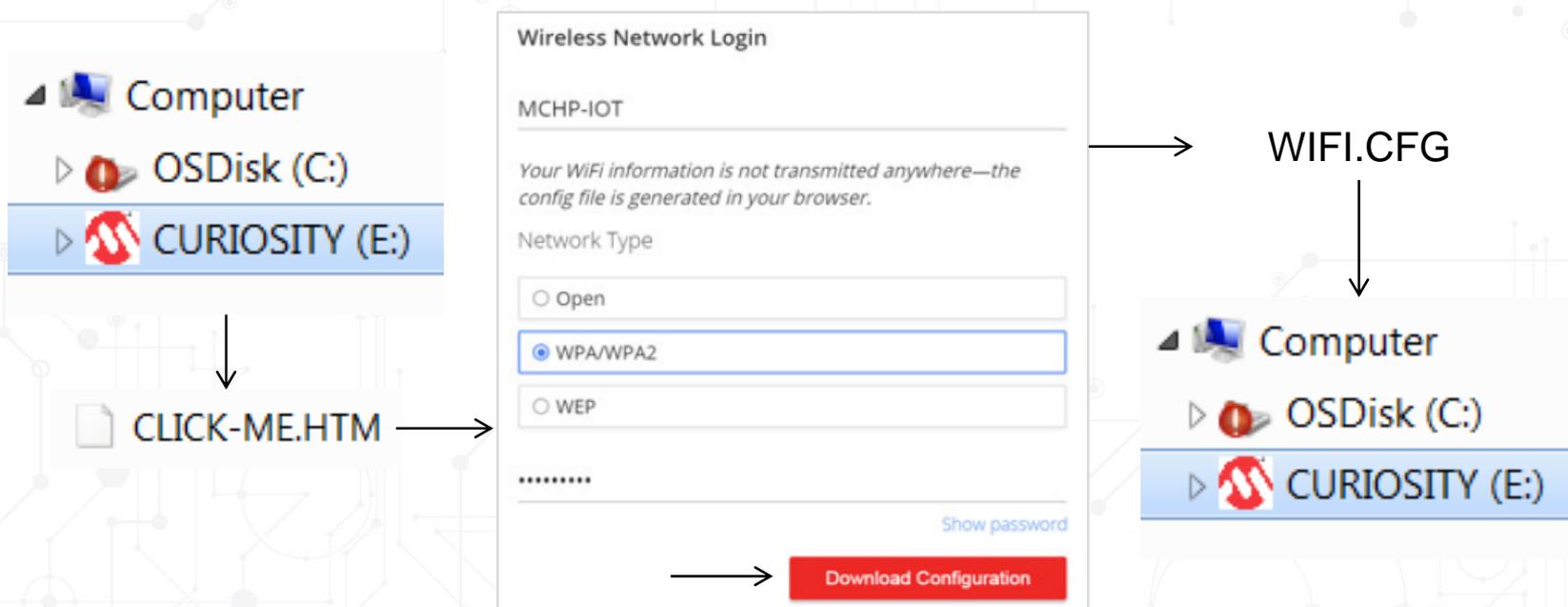
```
COM119:9600baud - Tera Term VT  
File Edit Setup Control Window Help  
-----  
Unknown command. List of available commands:  
reset  
device  
key  
reconnect  
version  
cli_version  
wifi <ssid>[,<pass>,[authType]]  
debug  
-----
```

连接至Wi-Fi®网络

- 通过PIC/AVR-IoT的配置网页
- 通过MPLAB®代码配置器（MCC）
- 通过软AP方式（SoftAP）
- 通过命令行（CLI）

连接至Wi-Fi®网络

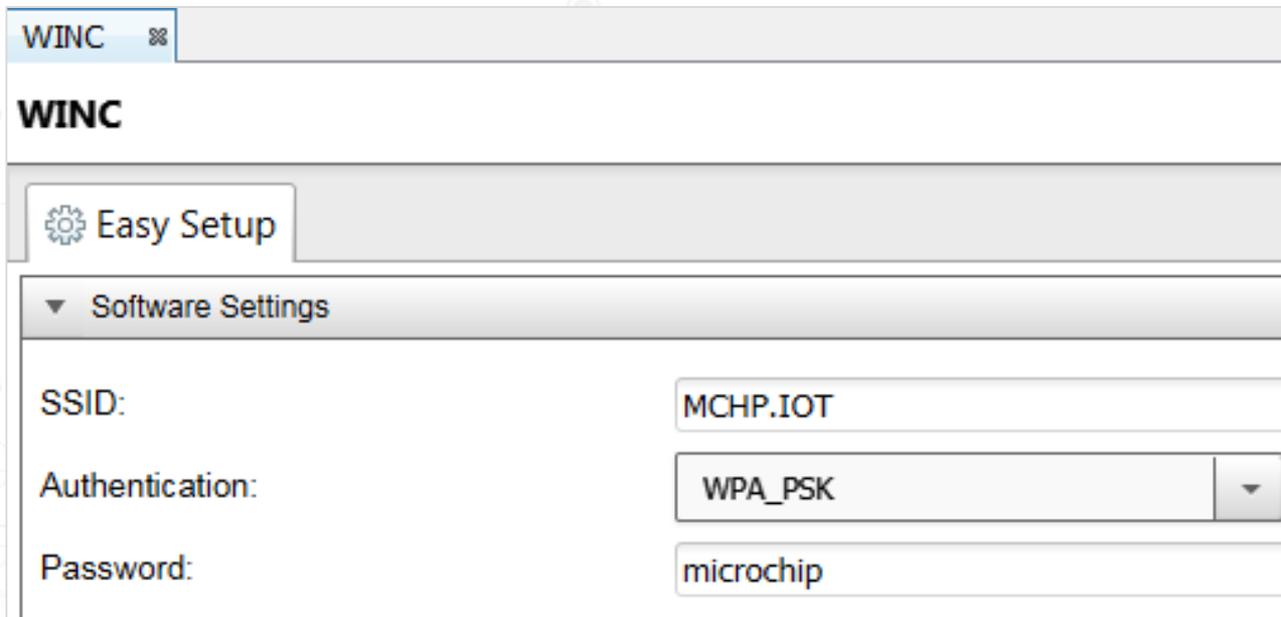
通过PIC/AVR-IoT的配置网页



连接至Wi-Fi®网络

通过MCC

- SSID
- 密码
- 网络类型



The screenshot shows the WINC Easy Setup interface. The title bar reads "WINC". Below the title bar, the text "WINC" is displayed. A tab labeled "Easy Setup" is selected. Underneath, a section titled "Software Settings" is expanded. It contains three input fields: "SSID:" with the value "MCHP.IOT", "Authentication:" with a dropdown menu set to "WPA_PSK", and "Password:" with the value "microchip".

连接至Wi-Fi®网络

- **通过软AP（SoftAP）**
 - 软AP功能由WINC1510提供
 - 可通过手机/平台的Web浏览器设置Wi-Fi用户名和密钥
 - 长按开发板SW0按钮进入软AP模式



Log In Cancel

Network Name
MCHP.IOT.ACCESSPOINT

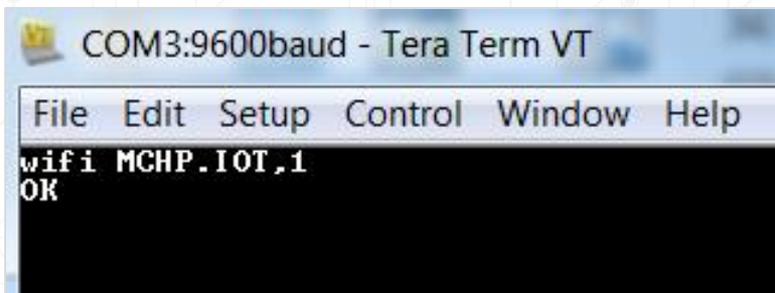
Pass phrase
.....

Device Name

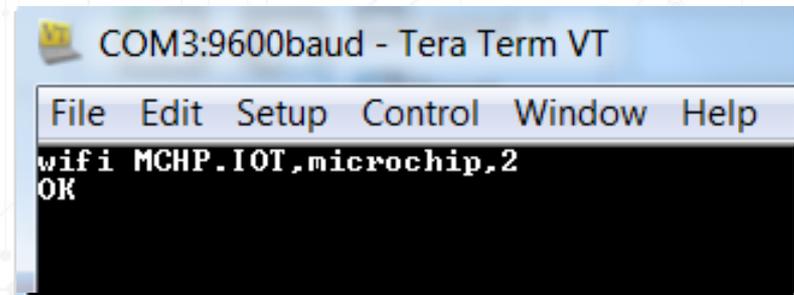
Connect

连接至Wi-Fi®网络

- 通过命令行方式（CLI）
 - SSID
 - 密码
 - 网络类型



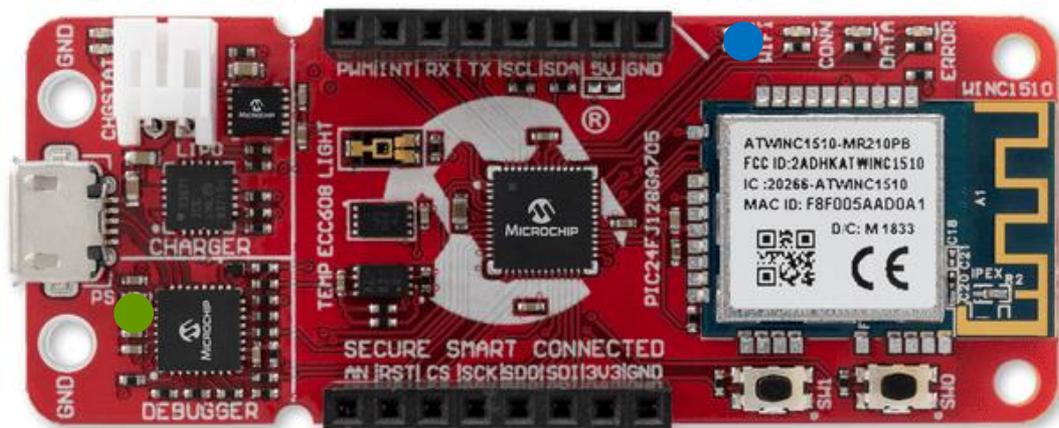
```
COM3:9600baud - Tera Term VT
File Edit Setup Control Window Help
wifi MCHP.IOT,1
OK
```



```
COM3:9600baud - Tera Term VT
File Edit Setup Control Window Help
wifi MCHP.IOT,microchip,2
OK
```

连接至Wi-Fi®网络

- 通过PIC/AVR-IoT的配置网页
- 通过MCC
- 通过SoftAP
- 通过CLI



阿里云物联网平台协议

- 支持多种协议
 - 消息队列遥测传输（MQTT）
 - 超文本传输协议（HTTP）

- TCP传输层

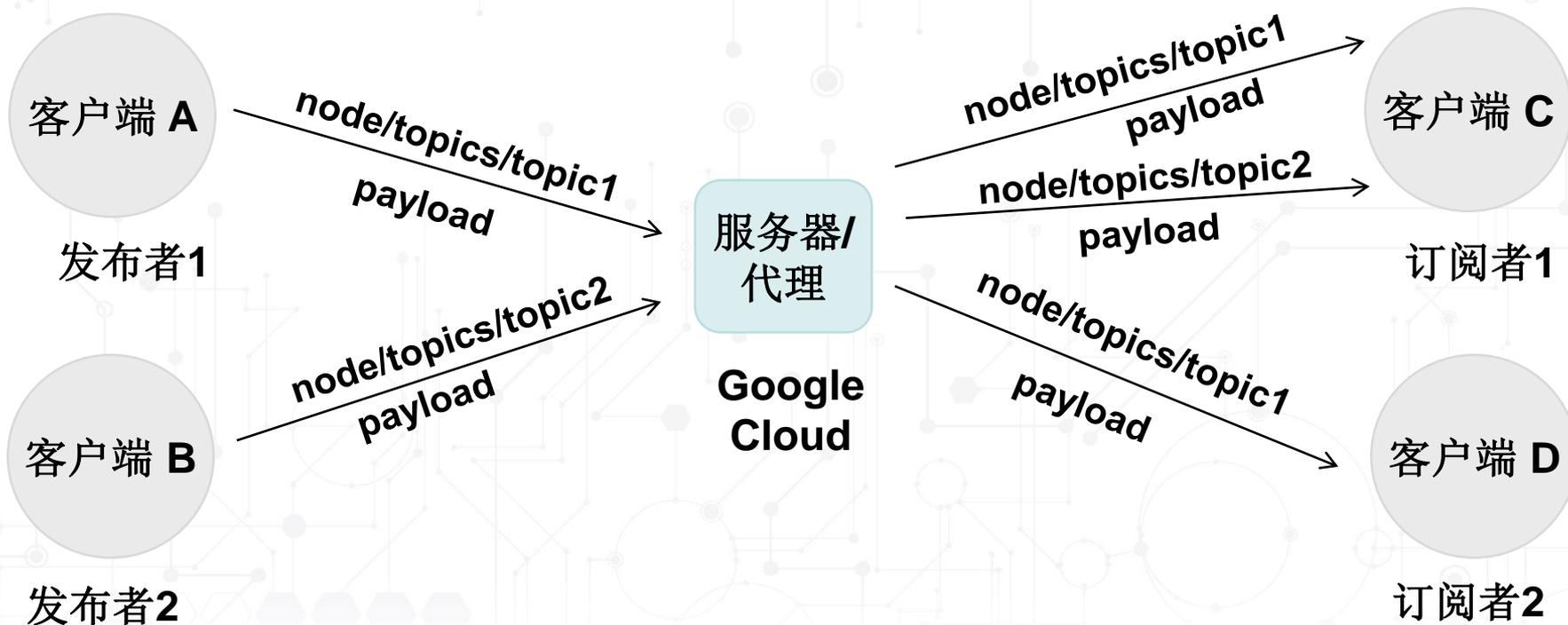
MQTT角色与内容

MQTT包含以下角色和内容：

- 客户端（Client）—— 消息处理节点
 - 发布者（Publisher）—— 发送消息
 - 订阅者（Subscriber）—— 接收消息
- 服务器（Server）
 - 代理（Broker）—— 消息中转（阿里云）
- 消息（Message）—— 数据结构体
- 主题（Topic）—— 消息分类

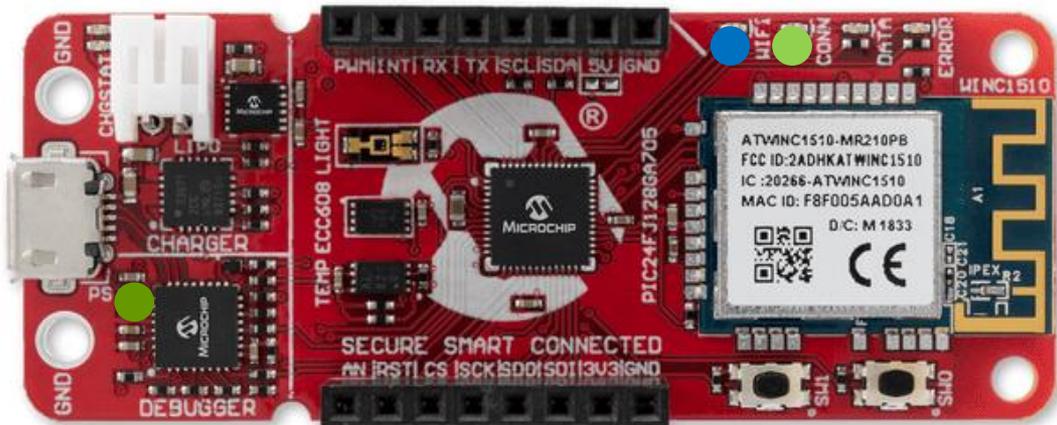
与阿里云物联网平台交换数据

- MQTT——轻量级IoT协议
- 发布/订阅方式



阿里云物联网平台协议

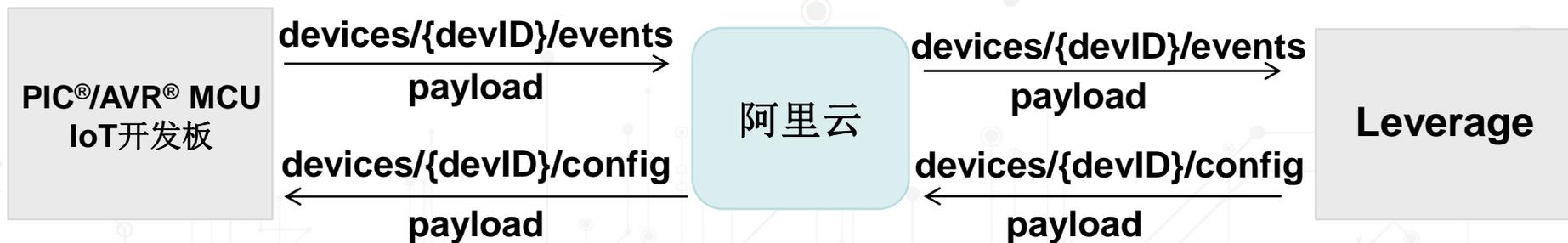
- 支持多种IoT连接协议
 - MQTT 
 - HTTP
- TCP作为传输层



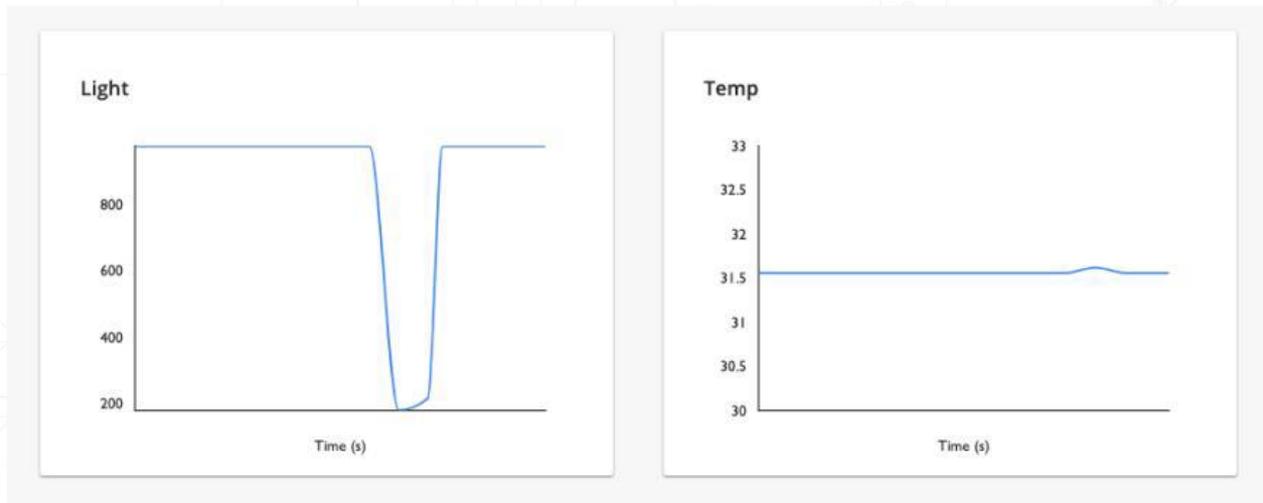
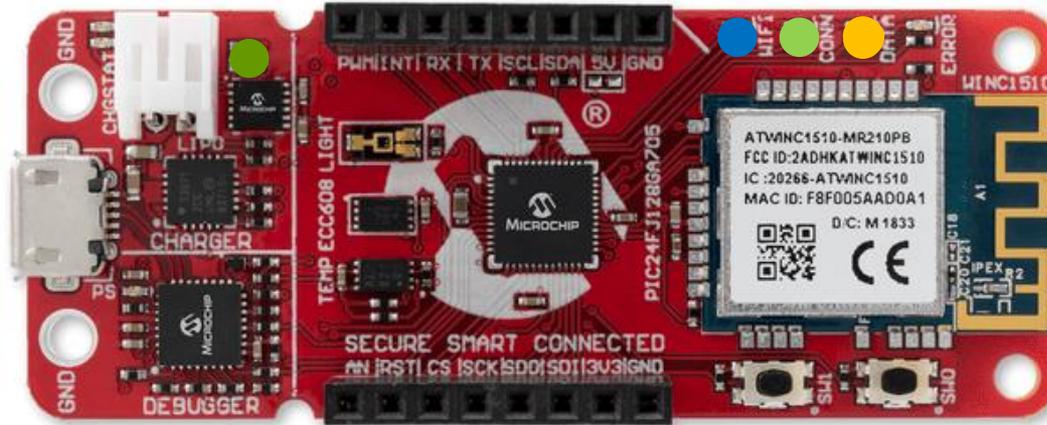


通过MQTT与阿里物联网平台 交换数据

发布/订阅



PIC[®]/AVR[®] MCU节点应用



PIC[®]/AVR[®] MCU IoT嵌入式应用

发布/订阅
交换数据

用户代码

HTTP MQTT

应用层

UDP TCP

以太网 Wi-Fi[®] Wi-Fi[®]

TCP/IP传输层

外设
传感器
执行器

PIC/AVR IoT WG硬件



问题

基于PIC®/AVR® MCU的IoT应用使用哪种协议与阿里云物联网平台进行数据交换？

MQTT——消息队列遥测传输



课程安排

- IoT简介
- IoT的基本原理
- 基于Microchip PIC®/AVR® MCU的IoT解决方案
- **使用PIC/AVR IoT开发板连接云**
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑

步骤

- 创建阿里云账号
- 创建云端IoT设备
- 通过MPLAB[®] X IDE编译基于MCC的AVR-IoT/PIC-IoT开发板工程
- 设备与云端连接

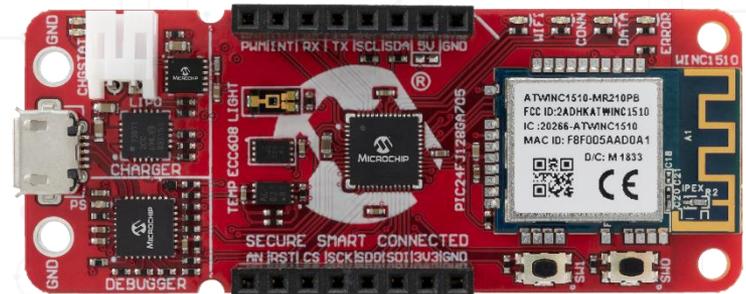
实验硬件

- **AVR-IoT/PIC-IoT WG开发板**
- **USB转接线**

AVR-IoT



PIC-IoT



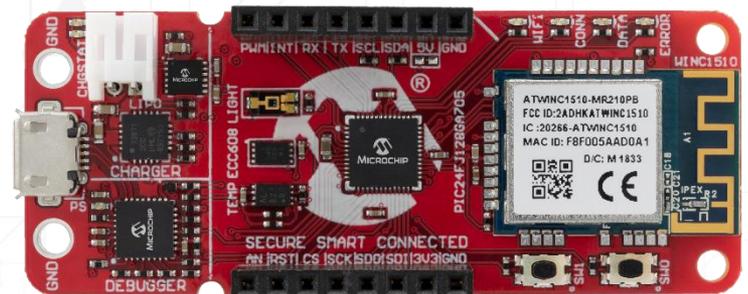
实验目标

- 熟悉在MCC中连接并使用AVR/PIC-IoT开发板
- 使用MCC创建AVR-IoT/PIC-IoT工程
- 工程编译
- 与阿里云进行数据交换

AVR-IoT



PIC-IoT



创建阿里云IoT Studio账号

- **阿里云IoT Studio账号**

可视化一站式开发平台，高效地完成设备、服务及应用开发

← → ↻ iot.aliyun.com/products/iotstudio

☰ 阿里云 IoT | 所知不止于感知

官方产品

特色行业

物联网市场

开发者

合作伙伴

最新活动

物联网应用开发 (IoT Studio)

物联网应用开发 (IoT Studio, 原 Link Develop), 是阿里云针对物联网场景提供的生产力工具, 可覆盖各个物联网行业核心应用场景, 帮助您高效经济地完成设备、服务及应用开发。物联网开发服务提供了移动可视化开发、Web 可视化开发、服务开发与设备开发等一系列便捷的物联网开发工具, 解决物联网开发领域开发链路长、技术栈复杂、协同成本高、方案移植困难的问题, 重新定义物联网应用开发。

立即使用

产品文档

IoT Studio创建新的项目

微芯IoT

项目概览

物联网开发1.6版本上线: [查看详情](#)

推荐

微芯IoT

Web可视化开发

项目产品

3

项目设备

4

Web应用

1

移动应用

2

移动应用开发

服务开发

快速入口



Web可视化开发

无需写代码, 通过可视化的方式搭建物联网网页应用



移动可视化开发

拖拽开发基于移动端的物联网APP或设备控制面板



服务开发

开发基于设备事件或接口请求触发的业务逻辑

设备管理

产品

设备

移动配置

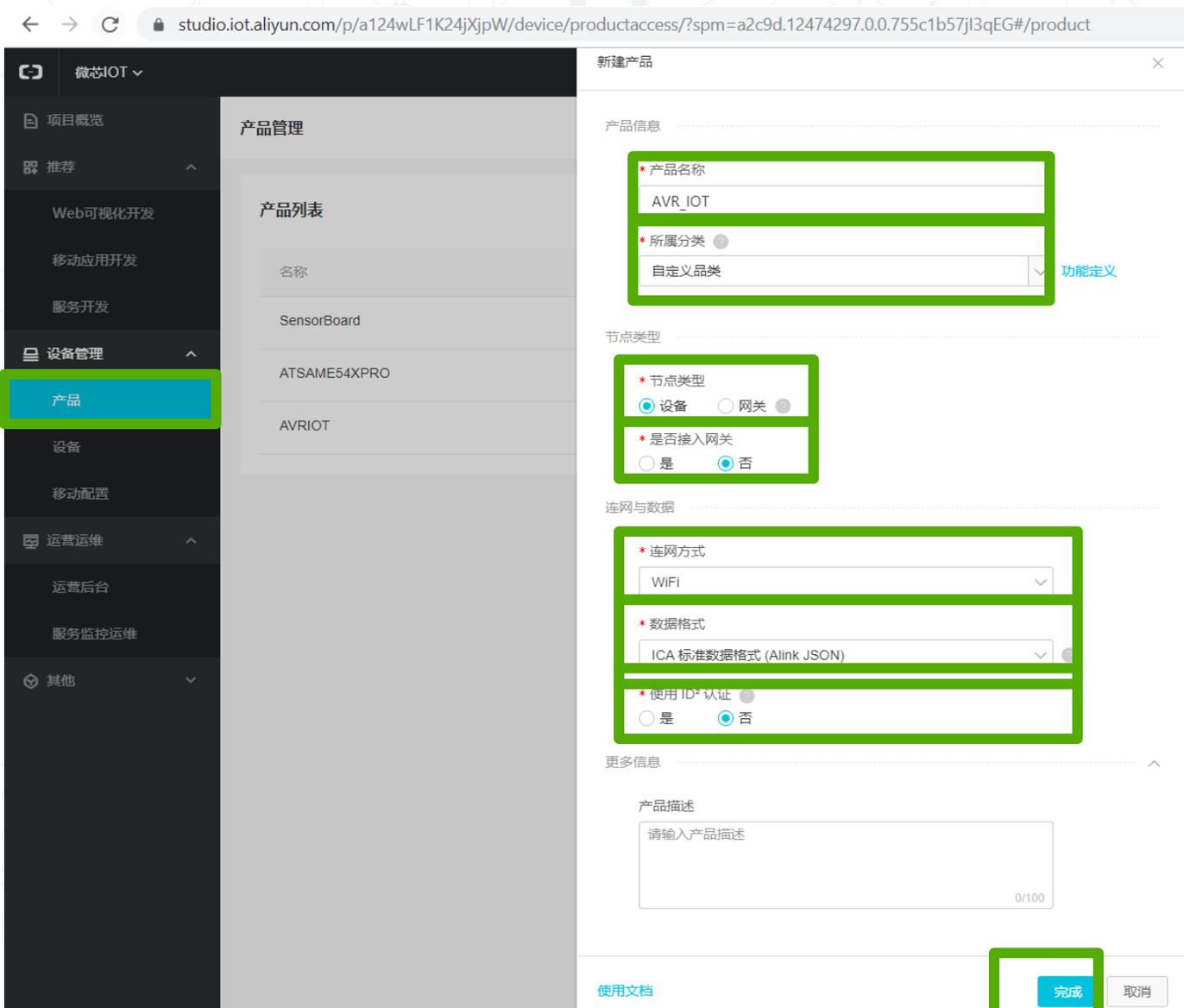
运营运维

运营后台

服务监控运维

其他

IoT Studio 创建新的产品



← → ↻ studio.iot.aliyun.com/p/a124wLF1K24jXjpW/device/productaccess/?spm=a2c9d.12474297.0.0.755c1b57jl3qEG#/product

新建产品

产品信息

- * 产品名称: AVR_IOT
- * 所属分类: 自定义品类 功能定义

节点类型

- * 节点类型: 设备 网关
- * 是否接入网关: 是 否

连网与数据

- * 连网方式: WIFI
- * 数据格式: ICA 标准数据格式 (Alink JSON)
- * 使用ID²认证: 是 否

更多信息

产品描述

请输入产品描述 0/100

使用文档 完成 取消

定义产品功能

微芯IoT

[开发文档](#)
[资源中心](#)
[开发者社区](#)

项目概览
推荐
Web可视化开发
移动应用开发
服务开发
设备管理
产品
设备
移动配置
运营运维
运营后台
服务监控运维
其他

产品管理 > 产品详情

AVRIOT 发布

ProductKey: a11v7xKKAWn [复制](#) ProductSecret: ***** [显示](#) 设备数: 3 [前往管理](#)

[产品信息](#) [Topic类列表](#) 功能定义 [服务端订阅](#) [日志服务](#) [在线调试](#)

标准功能
[导入物模型](#) [查看物模型](#) 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
<div style="text-align: center;">  <p>无标准功能</p> </div>					

自定义功能 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	亮度	Light	int32 (整数型)	取值范围: 0 ~ 1023	编辑 删除
属性	温度	Temp	text (字符串)	数据长度: 2048	编辑 删除
属性	灯光	YellowLEDSta...	bool (布尔型)	布尔值: 关 - 0; 开 - 1;	编辑 删除

IoT Studio创建新的设备



新增设备

* 产品: AVRIOT

* 添加方式: **自动生成** 批量上传

* 设备数量: 1

一次最多添加1000台, 系统会自动生成全局唯一的DeviceName。

提交 取消

当前在线 0

新增设备

DeviceName/备注名称	属产品	类型	态	线时	操
-----------------	-----	----	---	----	---

设备三元组信息

设备管理 > 设备详情

R1AjbfdhOGRuAiH9iDGk 离线

产品: AVRIOT

ProductKey: a11v7xKKAWn [复制](#)

DeviceSecret: ***** [显示](#)

设备信息 Topic列表 运行状态 事件管理 服务调用 设备影子 文件管理 日志服务 在线调试

设备信息

产品名称	AVRIOT	ProductKey	a11v7xKKAWn 复制	区域	华东2 (上海)
节点类型	设备	DeviceName	R1AjbfdhOG... 复制	DeviceSecret	***** 显示
备注名称	LeoPICIoT 编辑	IP地址	49.78.148.170	固件版本	-
添加时间	2019/09/16 23:32:49	激活时间	2019/09/16 23:50:58	最后上线时间	2019/09/27 17:35:35
当前状态	离线	实时延迟	测试		

阿里云物联网平台怎样标识设备

使用以下的标识符：

- 注册位置（Location）
- 产品ID（Product ID）
- 设备ID（Device ID）
- 设备密钥（Device Secure Key）



修改示例工程代码

✕ MPLAB X IDE v5.25 - AVR-IOT-Ali : default

File Edit View Navigate Source Refactor Production Debug Team Tools Window Help

The screenshot shows the MPLAB X IDE interface. On the left, the 'Projects - WSG' pane displays a tree view of the 'AVR-IOT-Ali' project, with 'cloud_config.h' selected under the 'config' directory. The main editor window shows the content of 'cloud_config.h'. The code defines several macros for cloud configuration, with three specific lines highlighted in green:

```
#ifndef CLOUD_CONFIG_H
#define CLOUD_CONFIG_H

// <h> Cloud Configuration

// <s> project id
// <i> Google Cloud Platform project id
// <id> project_id
#define CFG_PRODUCT_KEY "allv7xKKAwn"

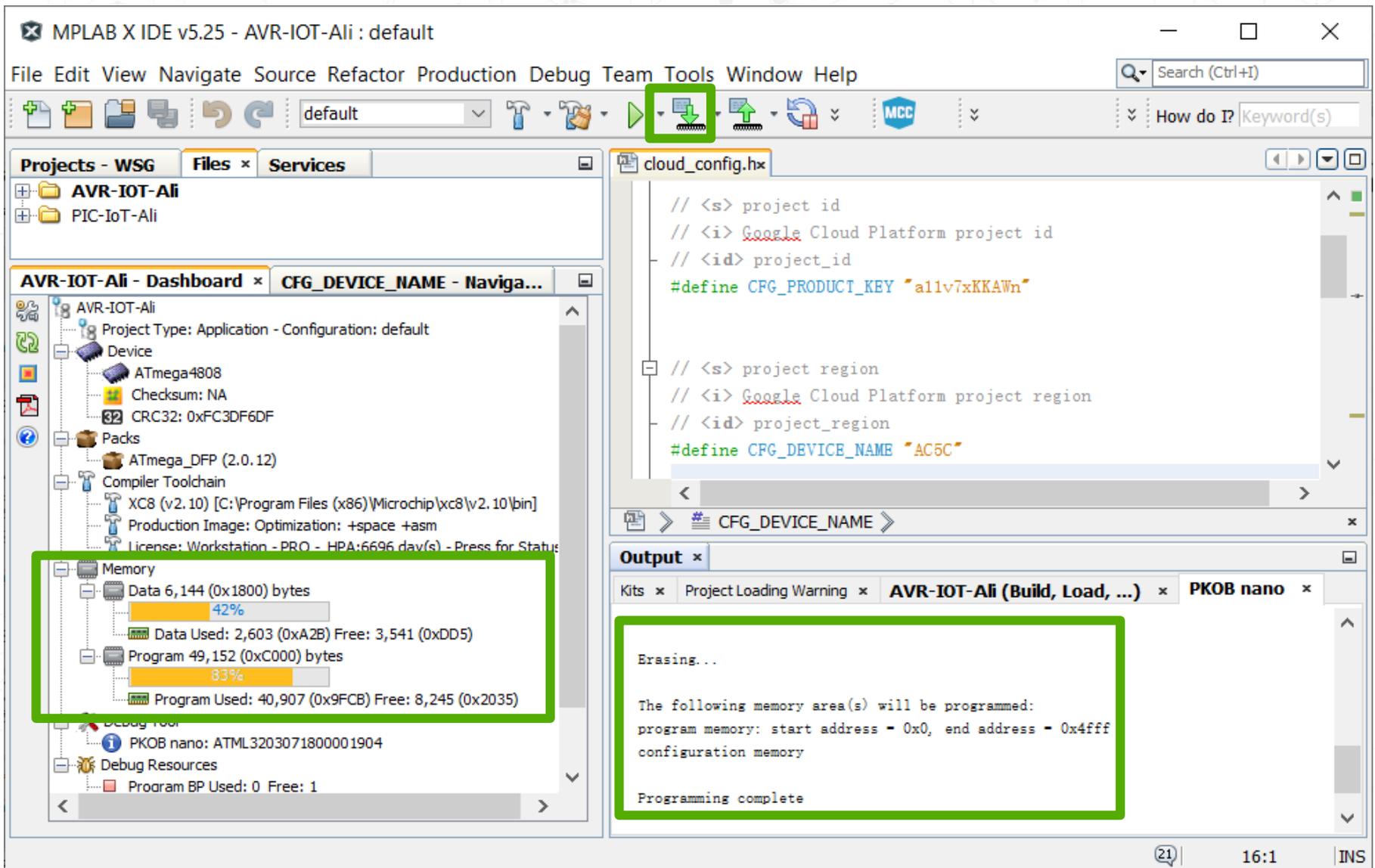
// <s> project region
// <i> Google Cloud Platform project region
// <id> project_region
#define CFG_DEVICE_NAME "AC5C"

#define CFG_DEVICE_SECRET "uzbhqsD8gRFuwX4HmUJiMShMmxclNMVu"
#define CFG_WRITE_DEVICE_SECRET 1

#define MAX_PRODUCT_KEY_LENGTH 11
#define MAX_DEVICE_NAME_LENGTH 30
#define DEVICE_SECRET_LENGTH 32

// <s> registry id
// <i> Google Cloud Platform registry id
// <id> registry_id
#define CFG_MQTT_HOST_SUFFIX ".iot-as-mqtt.cn-shanghai.aliyuncs.com"
```

与阿里云交换数据



The screenshot displays the MPLAB X IDE v5.25 interface for the project "AVR-IOT-Ali". The main window shows the "cloud_config.h" file with the following code:

```
// <s> project id  
// <i> Google Cloud Platform project id  
// <id> project_id  
#define CFG_PRODUCT_KEY "allv7xKKAwn"  
  
// <s> project region  
// <i> Google Cloud Platform project region  
// <id> project_region  
#define CFG_DEVICE_NAME "AC5C"
```

The left sidebar shows the project tree with the "Memory" section highlighted in green. The memory usage is as follows:

Memory Area	Used (bytes)	Free (bytes)	Usage %
Data	2,603 (0xA2B)	3,541 (0xDD5)	42%
Program	40,907 (0x9FCB)	8,245 (0x2035)	83%

The bottom right "Output" window shows the following text:

```
Erasing...  
  
The following memory area(s) will be programmed:  
program memory: start address = 0x0, end address = 0x4fff  
configuration memory  
  
Programming complete
```



与阿里云交换数据

设备管理 > 设备详情

R1AjbfdhOGRuAiH9iDGk 在线

产品: AVRIOT

ProductKey: a11v7xKKAWn [复制](#)

DeviceSecret: ***** [显示](#)

设备信息 Topic列表 运行状态 事件管理 服务调用 设备影子 文件管理 日志服务 在线调试

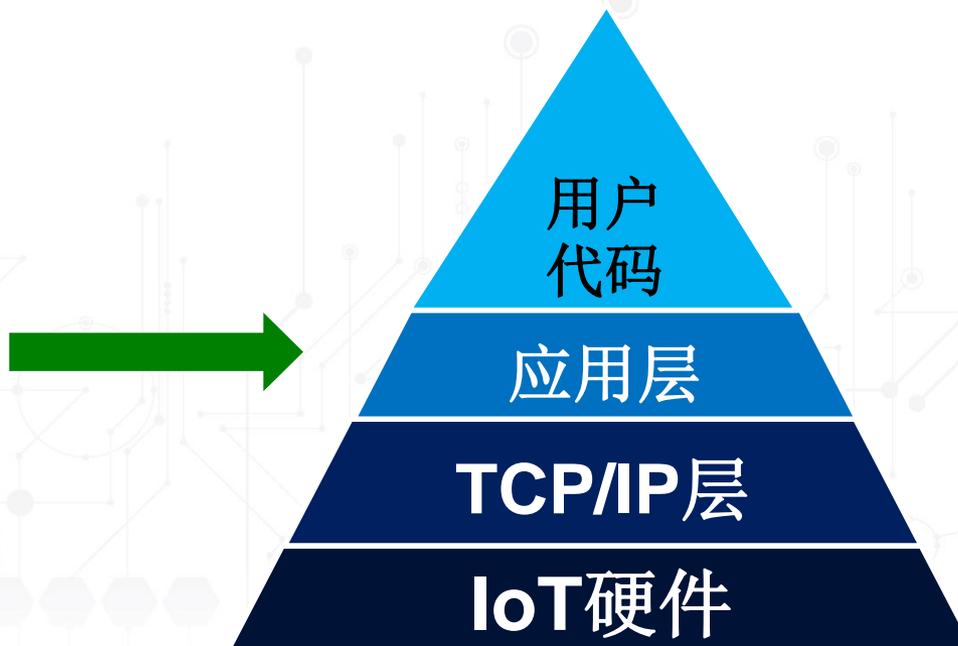
运行状态

实时刷新 [图表](#) [表格](#)

亮度 查看数据	温度 查看数据	灯光 查看数据
11	33.25	0 (关)
2019/10/02 01:13:53	2019/10/02 01:13:53	2019/10/02 01:13:53

应用层安全登录

- **MQTT 数据层**
 - 用户名，密码
 - 客户端ID (ClientID)



MQTT安全登录——密钥



- **连接报文：用户名、密码和ClientID**
 - MQTT客户端ID中包含登录时间戳
 - 密钥与ClientID绑定
 - 密钥的保存与计算由ATECC608A实现

MQTT安全登录—— MQTT ClientID的格式

`clientId|{signmethod},{securemode},{timestamp}`

- 仅在登录时使用
- 一个比较长的字符串，包含关键字与值
- 看起来像一个主题（topic）
- 但不是主题（topic）



MQTT安全登录—— MQTT ClientID

- ClientID: 客户端ID, 建议使用设备的MAC地址或SN码
- Signmethod: 签名算法类型, 支持hmacsha1和hmacsha256
- Timestamp: 当前时间 (单位为毫秒)
- Securemode: 2 (TLS直连模式) 和3 (TCP直连模式)

cliendId/{signmethod}/{securemode}/{timestamp}



MQTT安全登录—— MQTT登录报文

No.	Time	Source	Destination	Protocol	Length	Info
6	0.081153	192.168.43.244	106.15.100.2	MQTT	200	Connect Command
8	0.156484	106.15.100.2	192.168.43.244	MQTT	58	Connect Ack
9	0.159768	192.168.43.244	106.15.100.2	MQTT	164	Subscribe Request (id=1) [/sys/a1us7IMdhg4/High_LE
11	0.211421	106.15.100.2	192.168.43.244	MQTT	60	Subscribe Ack (id=1)

- > Frame 6: 200 bytes on wire (1600 bits), 200 bytes captured (1600 bits) on interface 0
- > Ethernet II, Src: IntelCor_a2:b5:20 (60:f6:77:a2:b5:20), Dst: HuaweiTe_74:14:9c (10:44:00:74:14:9c)
- > Internet Protocol Version 4, Src: 192.168.43.244, Dst: 106.15.100.2
- > Transmission Control Protocol, Src Port: 50416, Dst Port: 1883, Seq: 1, Ack: 1, Len: 146
- ▼ MQ Telemetry Transport Protocol, Connect Command
 - > Header Flags: 0x10, Message Type: Connect Command
 - Msg Len: 143
 - Protocol Name Length: 4
 - Protocol Name: MQTT
 - Version: MQTT v3.1.1 (4)
 - > Connect Flags: 0xc2, User Name Flag, Password Flag, QoS Level: At most once delivery (Fire and Forget), Clean Session Flag
 - Keep Alive: 60

```
Client ID Length: 67
Client ID: 10.0.75.1|securemode=3.signmethod=hmacsha1.timestamp=1538210758441|
User Name Length: 20
User Name: High LED&a1us7IMdhg4
Password Length: 40
Password: 3391B9CB48FA56B7F88C5E9E3B687644262B6A04
```

```
0000 10 44 00 74 14 9c 60 f6 77 a2 b5 20 08 00 45 00 .D.t..` w...E.
0010 00 ba 48 7e 40 00 80 06 f7 11 c0 a8 2b f4 6a 0f ..H~@... ..+..j.
0020 64 02 c4 f0 07 5b 57 81 50 b2 62 a1 22 9e 50 18 d....[W.P.b."P.
0030 44 70 2d f8 00 00 10 8f 01 00 04 d4 51 54 54 04 Dp-----..MQTT.
0040 c2 00 3c 00 43 31 30 2e 30 2e 37 35 2e 31 7c 73 ..<-C10. 0.75.1|s
0050 65 63 75 72 65 6d 6f 64 65 3d 33 2c 73 69 67 6e eecuremod e=3,sign
0060 6d 65 74 68 6f 64 3d 68 6d 61 63 73 68 61 31 2c method=h macsha1,
0070 74 69 6d 65 73 74 61 6d 70 3d 31 35 33 38 32 31 timestamp p=153821
0080 30 37 35 38 34 34 31 7c 00 14 48 69 67 68 5f 4c 0758441| ..High_L
0090 45 44 26 61 31 75 73 37 49 4d 64 68 67 34 00 28 ED&a1us7 IMdhg4.(
00a0 33 33 39 31 42 39 43 42 34 38 46 41 35 36 42 37 3391B9CB 48FA56B7
00b0 46 38 38 43 35 45 39 45 33 42 36 38 37 36 34 34 F88C5E9E 3B687644
00c0 32 36 32 42 36 41 30 34 262B6A04
```

MQTT安全登录——服务器验证



- 用户名和密码与ClientID正确



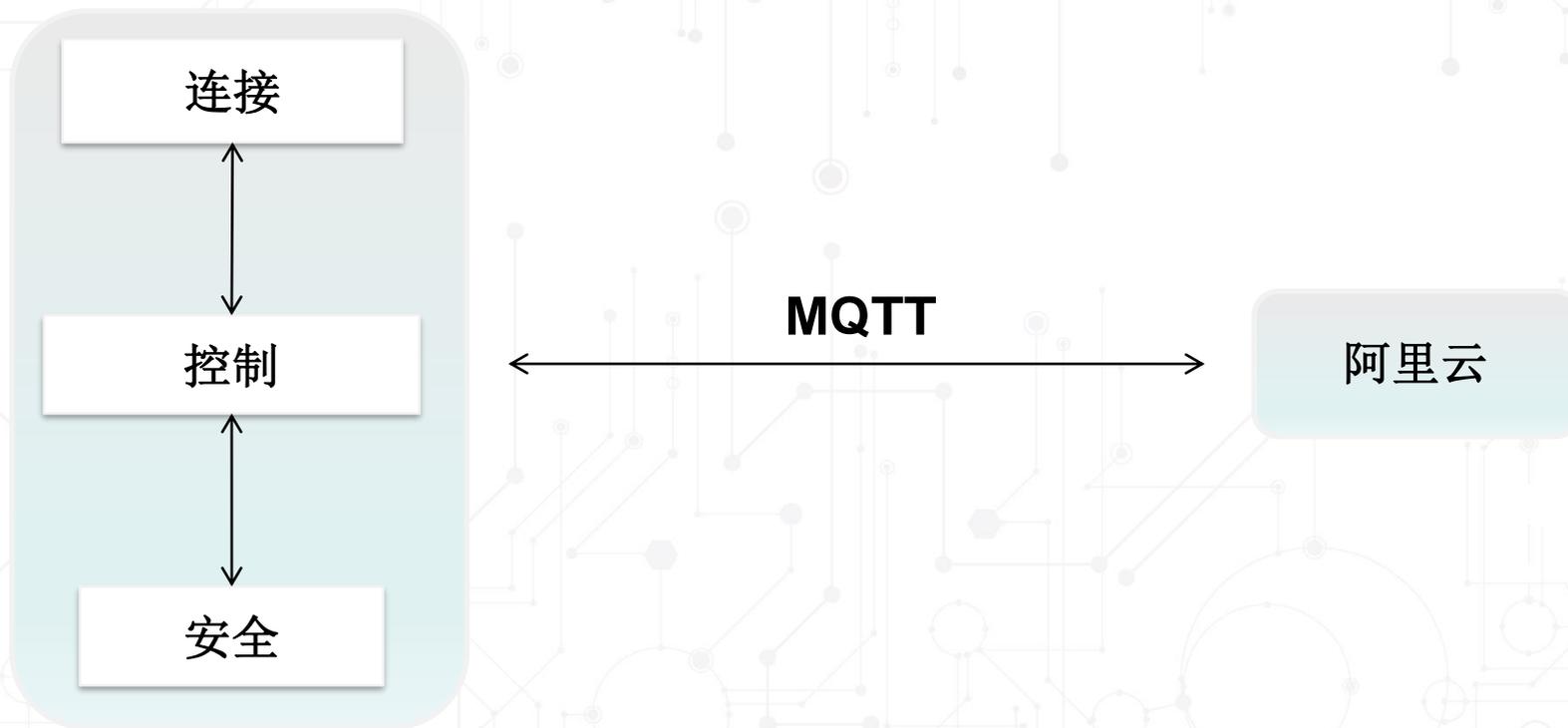


课程安排

- IoT简介
- IoT的基本原理
- 基于Microchip PIC®/AVR® MCU的IoT解决方案
- 使用PIC/AVR IoT开发板连接云
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑

与阿里云交换数据

IoT消息的发布订阅

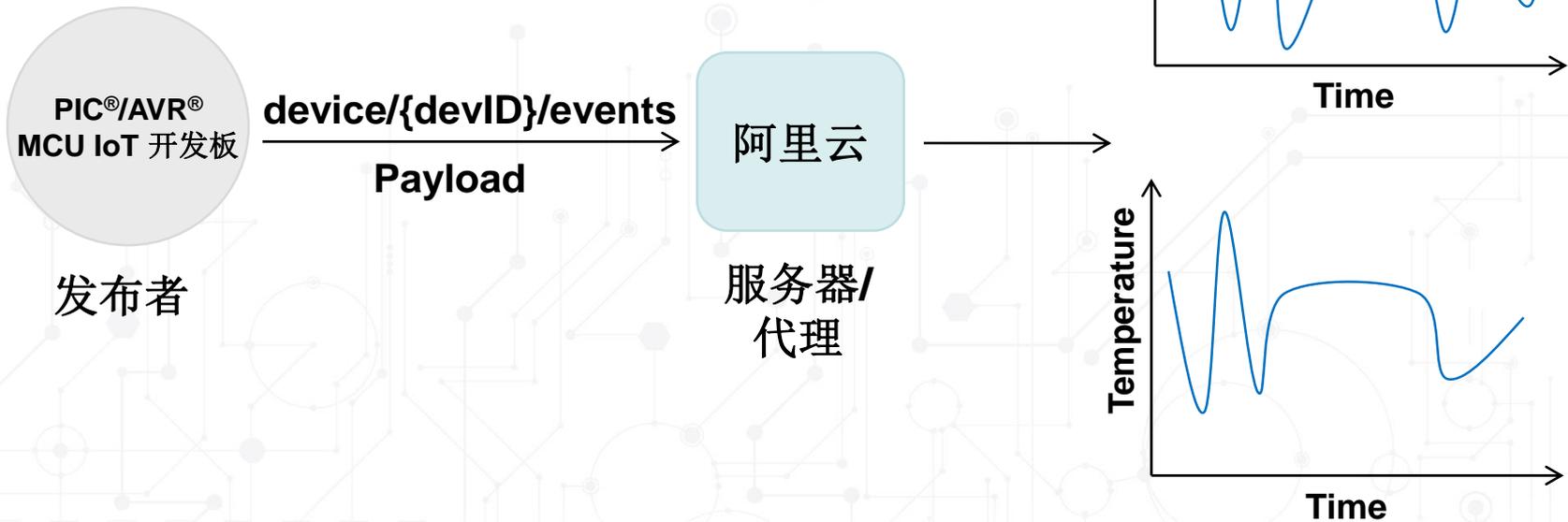


PIC/AVR-IoT 开发板



与阿里云交换数据

发布消息





发布消息

- 每条发布消息都有特定的主题 (**topic**)
- 使用主题 (**topic**) 分类消息
- 发布消息

主题 (topic)	<code>device/{deviceId}/events</code>
内容 (payload)	<code>light, temperature</code>

发布主题 (Topic)

- 大小写敏感的一个字符串
- 分层结构，使用正斜杠做为分隔符

sys/{deviceId}/events

- 允许使用通配符 (#)

sys/{deviceId}/events/#

阿里物联网平台——发布主题

- 阿里物联网平台——发布主题
 - **`/sys/{productID}/${deviceName}/thing/event/property/post`**
 - **`/sys/{productID}/${deviceName}/thing/service/property/set`**

- JavaScript Object Notation (JSON)字符串

topic	/devices/{device-id}/events
payload	Light: lightValue, Temp: tempValue



什么是JSON?

- **JavaScript Object Notation (JSON)**
- **密钥组——配对值**

```
{  
  "key1": value,  
  "key2": value  
}
```

```
{  
  "Light": lightValue,  
  "Temp": tempValue  
}
```

- **格式为字符串形式**



推送数据至云端

main.c

```
void sendToCloud (void)
```

```
{
```

```
    static char json[70];
```

```
    int rawTemperature = SENSORS_getTempValue();
```

```
    int light = SENSORS_getLightValue();
```

```
    uint8_t ledYellowStatus = LED_Yellow_Status();
```

```
    int len = sprintf(json,
```

```
        "{\"YellowLedStatus\":%d, \"Light\":%d, \"Temp\": \"%d.%02d\"}",
```

```
        ledYellowStatus, light, rawTemperature/100,
```

```
        abs(rawTemperature)%100);
```

```
    if (len > 0) {
```

```
        CLOUD_publishData((uint8_t*)json, len);
```

```
    }
```

```
}
```



推送数据至云端

 mqtt_packetPopulate.c

```
void MQTT_CLIENT_publish(uint8_t *data, uint16_t len)
{
    mqttPublishPacket cloudPublishPacket;
    // Topic
    cloudPublishPacket.topic = (uint8_t*)mqttTopic;

    // Payload
    cloudPublishPacket.payload = data;
    cloudPublishPacket.payloadLength = len;
    if(MQTT_CreatePublishPacket(&cloudPublishPacket) != true)
    {
        debug_printError("MQTT: Connection lost PUBLISH failed");
    }
}
```

与阿里云交换数据

微芯IoT 开发文档

项目概览 推荐 Web可视化开发 移动应用开发 服务开发 设备管理 产品 设备 移动配置 运营运维 运营后台 服务监控运维 其他

设备管理 > 设备详情

AC5C 在线

产品: AVRIOT ProductKey: a11v7xKKAWn [复制](#) DeviceSecret: ***** [显](#)

设备信息 Topic列表 运行状态 事件管理 服务调用 设备影子 文件管理 **日志服务** 在线调试

设备行为分析 物模型数据分析 **上行消息分析** 下行消息分析

请输入MessageID 请输入

全部状态 1小时

时间	MessageID
2019/10/02 01:23:04.883	117908432
2019/10/02 01:23:03.603	117908432
2019/10/02 01:23:02.323	117908431
2019/10/02 01:23:01.044	117908431
2019/10/02 01:22:59.763	117908430

查看详情

MessageID	1179084328433415168 复制
Topic	/sys/a11v7xKKAWn/AC5C/thing/event/property/post
时间	2019/10/02 01:23:04
内容	Text (UTF-8) <input type="text"/> {"id":"1","version":"1.0","params":{"YellowLEDStatus":0,"Light":84,"Temp":"37.56"},"method":"thing.event.property.post"} 复制

关闭

从IoT节点发布数据

- 自定义发布数据
- 支持多种mikroBUS™开发板



- 在MCC中支持多种mikroBus开发板！



问题

内容（比如传感器数据）是以什么格式发送到物联网平台的？

JSON——JavaScript Object Notation



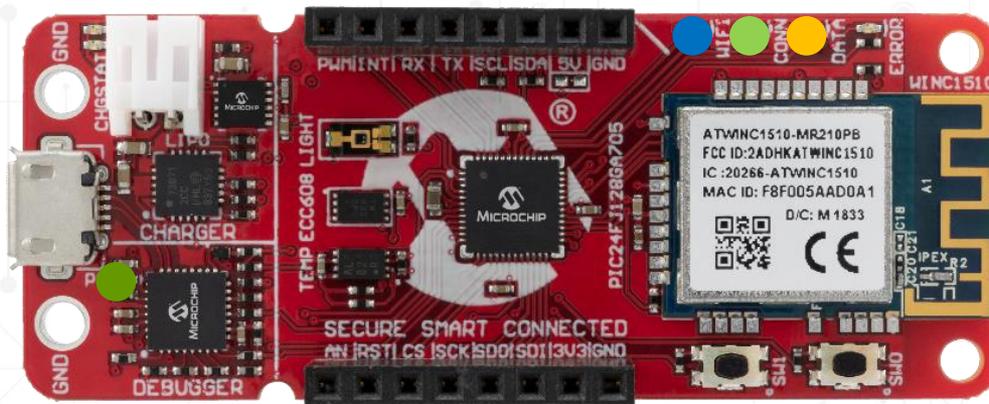
课程安排

- IoT简介
- IoT的基本原理
- 基于Microchip PIC[®]/AVR[®] MCU的IoT解决方案
- 使用PIC/AVR IoT开发板连接云
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑

步骤

- IoT开发板订阅YellowLEDStatus状态
- 云端通过下发YellowLEDStatus值控制IoT开发板的黄色LED

PIC-IoT





与阿里云交换数据

- 发送“订阅 (SUBSCRIBE)” 报文
- 接收“订阅 (SUBSCRIBE)” 应答



与阿里云交换数据

订阅数据





订阅消息

- IoT节点期望接收某类从云端发送的数据
- 通过一个主题（**topic**）获取数据
- 云端可灵活控制订阅主题



订阅消息

- 一个IoT节点可以订阅一个特定的主题
- 订阅消息
- 订阅成功的确认消息

payload

topic

payload

devices/{deviceId}/config

阿里云订阅主题

- **Google Cloud Core IoT订阅主题**
 - **`/devices/{deviceId}/config`**
- **IoT节点接收发布的有关
'`/devices/{deviceId}/config`'的消息**



订阅一个主题

mqtt_packetPopulate.c

```
void CLOUD_subscribe(void)
{
    // Payload
    cloudSubscribePacket.subscribePayload[topicCount].topic =
        "/devices/{deviceId}/config";
    // Processing packet received over config topic
    imqtt_publishReceiveCallBackTable[0].topic =
        "/devices/{deviceId}/config";
    imqtt_publishReceiveCallBackTable[0].mqttHandlePublishDataCallBack
        = receivedFromCloud;
    if(MQTT_CreateSubscribePacket(&cloudSubscribePacket) == true)
    {
        debug_printf("CLOUD: SUBSCRIBE packet created");
    }
}
```



通过订阅主题 (Subscribed Topic) 接收数据

The screenshot displays the Microchip IoT development console. The top navigation bar includes '开发文档', '资源中心', and '开发者社区'. The main interface is divided into several sections:

- 设备信息** (Device Information): Shows '调试设备' (Debug Device) set to 'AC5C'.
- Topic列表** (Topic List): Shows '调试设备' (Debug Device) and '虚拟真实设备' (Virtual Real Device).
- 运行状态** (Running Status): Shows '功能' (Function) set to '灯光(YellowLE...)' (Light(YellowLE...)) and '方法' (Method) set to '设置' (Settings).
- 实时日志** (Real-time Logs): Shows a log entry for '云端下发数据' (Cloud Downloaded Data) at '2019-10-02 01:50:07'. The log content includes a JSON object with 'params' and 'version' fields.

The '实时日志' section contains the following log entry:

```
云端下发数据
2019-10-02 01:50:07
{"code":200,"message":success,"topic="/sys/a11v7xKKAWn/AC5C/thing/service/property/set,response":{"code":200,"data":{},"id":"1626006927","message":"success","version":"1.0"},"device":{"activeTime":1555509270000,"aliyunCommodityCode":"iothub_senior","deviceKey":"qvxd7LSGsKE1sgXF2er","deviceSecret":"***","gmtCreate":155550917000,"gmtModified":1557496452000,"id":37301001,"iotId":"qvxd7LSGsKE1sgXF2er000100","name":"AC5C","nickname":"BrianLin","productKey":"a11v7xKKAWn","rbacTenantId":"8444610A90644C57BED6050A3FBD208F","region":"cn-shanghai","status":3,"statusLast":1,"thingType":"DEVICE"},scriptData={},useTime=5,rrpc=false,traceId=0bc1606315699522078713086d06e7
```



通过订阅主题接收数据

main.c

```
void receivedFromCloud(uint8_t *topic, uint8_t *payload)
{
    char *toggleToken = "\"toggle\": ";
    char *subString;
    if ((subString = strstr((char*)payload, toggleToken)))
    {
        LED_holdYellowOn( subString[strlen(toggleToken)] == '1' );
    }

    debug_printer(SEVERITY_NONE, LEVEL_NORMAL, "topic:
                %s", topic);
    debug_printer(SEVERITY_NONE, LEVEL_NORMAL, "payload:
                %s", payload);
}
```



通过订阅主题接收数据

VT COM15 - Tera Term VT



File Edit Setup Control Window KanjiCode Help

```
0123916E0015DDB3FE NONE NORMAL topic: /sys/a11v7xKKAWn/AC5C/thing/service/property/set
0123916E0015DDB3FE NONE NORMAL payload: {"method":"thing.service.property.set", "id":"1626098340", "params":{"YellowLEDStatus":1}, "version":"1.0.0"}
```



课程安排

- IoT简介
- IoT的基本原理
- 基于Microchip PIC®/AVR® MCU的IoT解决方案
- 使用PIC/AVR IoT开发板连接云
- 推送数据至阿里云
- 订阅数据并从云端接收数据
- 总结与答疑

总结

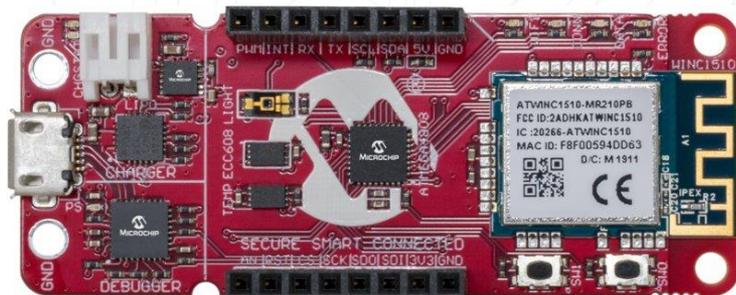
本节课程覆盖了以下几点：

- 基于**Microchip PIC[®]/AVR[®] MCU**的**IoT**解决方案
- **PIC/AVR-IoT WG**开发板的基本功能
- 通过**MCC**生成一个安全的**IoT**节点
- 开发板与**阿里云物联网平台**交换数据

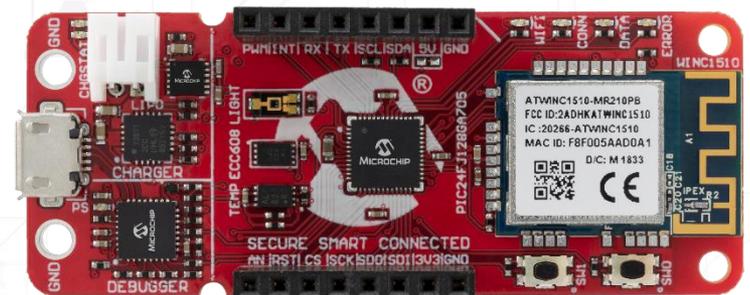
本节课使用的开发板

- **AC164160 AVR-IoT WG开发板**
- **AC164164 PIC-IoT WG开发板**

AVR-IoT



PIC-IoT





相关文档和资源

- [AVR-IoT WG Development Board](#)
- [PIC-IoT WG Development Board](#)
- [MCC-AVRIOT-ALI Source Code](#)
- [MCC-PICIOT-ALI Source Code](#)



提问？

法律声明



软件:

Microchip软件仅允许用于Microchip产品。此外，Microchip软件的使用受软件附带的版权声明、免责声明以及任何授权许可条款的限制，无论这些内容是在安装各个程序时阐明还是在头文件或文本文件中公告。

尽管有上述限制，但Microchip和第三方提供的软件的某些组件仍可能被“开源”软件许可覆盖，其中包括要求分发者提供软件源代码的许可。在开源软件许可要求的范围内，许可条款将起主导作用。

注意事项和免责声明:

这些材料和随附信息（例如，包括任何软件以及对第三方公司和第三方网站的引用）仅供参考，并且按“现状”提供。Microchip对第三方公司做出的声明或第三方可能提供的材料或信息不承担任何责任。

MICROCHIP不承担任何形式的保证，无论是明示的、暗示的或法定的，包括有关无侵权性、适销性和特定用途的暗示保证。在任何情况下，对于与MICROCHIP或其他第三方提供的材料或随附信息有关的任何直接或间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，MICROCHIP概不承担任何责任，即使MICROCHIP已被告知可能发生损害或损害可以预见。请注意，使用此处所述的知识产权时可能需要第三方许可。

商标:

Microchip的名称和徽标组合、Microchip徽标、Adaptec、AnyRate、AVR、AVR徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi徽标、MOST、MOST徽标、MPLAB、OptoLyzer、PackeTime、PIC、picoPower、PICSTART、PIC32徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron及XMEGA均为Microchip Technology Inc.在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、PrecisionEdge、ProASIC、ProASIC Plus、ProASIC Plus徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath和ZL均为Microchip Technology Inc.在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA和ZENA均为Microchip Technology Inc.在美国和其他国家或地区的商标。

SQTP为Microchip Technology Inc.在美国的服务标记。

Adaptec徽标、Frequency on Demand、Silicon Storage Technology和Symmcom为Microchip Technology Inc.在除美国外的国家或地区的注册商标。

GestIC为Microchip Technology Inc.的子公司Microchip Technology Germany II GmbH & Co. & KG在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2019, Microchip Technology Inc.版权所有。