

基于区块链的智能水表应用

学 院（系）： 软件学院
专 业： 软件工程
班 级： 电计 1801
学 生 姓 名： 聂闻华
学 号： 201999903
完 成 日 期： 2020-06-18

大连理工大学

目 录

1	文献综述.....	1
1.1	论文相关背景	1
1.2	研究意义	1
2	相关技术介绍	3
2.1	区块链	3
2.1.1	区块链定义.....	3
2.1.2	区块链特征.....	3
2.2	LoRa 通信	4
2.2.1	LoRa 定义.....	4
2.2.2	LoRa 无线技术的主要特点.....	4
2.3	窄带物联网 NB-IoT	4
2.3.1	NB-IoT 定义	4
2.3.2	NB-IoT 优势	5
2.4	无线技术简要介绍与对比	6
3	具体功能实现.....	8
3.1	实验一区块链基本代码	8
3.1.1	代码分析.....	8
3.1.2	代码实现.....	8
3.2	实验二区块链数字签名代码	10
3.2.1	代码分析.....	10
3.2.2	代码实现.....	11
3.3	实验三基于区块链的智能水表代码	13
3.3.1	代码分析.....	13
3.3.2	代码实现.....	13
4	总结	19
	参 考 文 献.....	20

1 文献综述

1.1 论文相关背景

水表一直是每家每户必不可少的设备，通过抄表员上门抄表进行对当月用水量的统计，用户再进行缴费动作。

中国传统的水表是安装在家中，抄表员必须进到家中进行抄表工作，一来可能影响住户本身的隐私；二来有可能碰到住户不在家中，无法抄表的尴尬情况。甚至有可能因为抄水员的失误，造成水费计量错误，形成了不必要的麻烦。国外一些地方的水表安装在街道上，但这一般针对于住户较少的地区，同时也一样无法避免人工抄表这一环节。

目前国内外有很多类型的智能水表，主要集中在北美、西欧等发达国家市场，总体来说，由于人们对于水资源管理可持续化愈加重视，先进水表计量技术正受到越来越多的关注。智能水表技术的应用并不受区域限制，但是各区域市场的驱动及限制因素不尽相同。目前，该技术的主要市场集中在以北美和欧洲为首的发达地区。在发展中地区，智能水表设备的应用普遍还处于试点阶段。智能水表的增长很大程度上是由北美市场所拉动，智能水表的出货量仍然只占水表总出货量的一小部分。以 2010 年为例，全球智能水表出货量仅占全部水表的 6%，而预计到 2016 年年末，该比例仅上升至 8%。

然而，在国内市场上，智能水表改造需要巨额资金投入，资金来源问题是困扰智能水表发展的一大瓶颈。智能水表行业所面临的最大困难就是融资问题，大部分自来水厂资金并不充裕，通常其预算的很大一部分都用于供水网络的维修。要想投资新技术，无论从预算里拨款还是融资都具有相当的困难。而在投入大量资金引进智能水表之前，企业往往只是先启动小型的试点项目，智能水表市场的发展因此深受影响。现阶段广泛推广使用智能水表的条件还不成熟，尚需时日。

1.2 研究意义

实践证明，无论是供水管理部门员工还是委托、雇佣他人抄表，只要是人工方式都存在不按时、不到位，估抄、错抄、漏抄等现象，而且人工抄表需要挨家挨户上门抄表，工作效率十分低下。在这种背景下，如何提高管理水平，应用新技术，切实解决好查表、收费以及监控管理等等问题，合理使用水资源，对供水管理部门是一项新课题。

基于区块链+LoRa 或 NB-IoT 的模式，通过低耗远距的方法将用户的用水计量信息传送到收费单位，从而代替人工这一费事费力且容易出错的方式。并且因为区块链的篡改需要大量成本，所以水表被用户私自修改的情况基本可以避免。

同时对于日常的人工费用可以缩减，只需要支出维护人员及统计人员的费用，既节省了人力，还节省了支出。弊端是初期安装费用较大，需要挨家挨户进行水表替换工作，所以这是智能水表难以取代传统水表的原因之一。但主要原因仍是智能水表技术不够完善，水表安装位置难以控制等。

2 相关技术介绍

2.1 区块链

2.1.1 区块链定义

区块链（Blockchain）是一种由多方共同维护，使用密码学保证传输和访问安全，能够实现数据一致存储、难以篡改、防止抵赖的记账技术，也称为分布式账本技术（Distributed Ledger Technology）。

典型的区块链以块-链结构存储数据。作为一种在不可信的竞争环境中低成本建立信任的新型计算范式和协作模式，区块链凭借其独有的信任建立机制，正在改变诸多行业的应用场景和运行规则，是未来发展数字经济、构建新型信任体系不可或缺的技术之一。各类区块链虽然在具体实现上各有不同，但在功能架构存在共性。

因为区块链的特性，完美的契合了水表这一方面的需求，既可以将用户的用水信息进行统一管理，做成账本的形式，防止人工方面的疏漏；还可以防止占小便宜的用户对用水信息进行篡改，一举两得。目前我认为比较完善的做法是，收费单位使用公钥对用户用水信息进行加密，用户可以通过 app 等方式，使用自己的私钥解密，这样不仅可以防止信息被篡改，同时也让用户可以更加信任收费单位，从而避免不必要的麻烦。

2.1.2 区块链特征

去中心化。区块链技术不依赖额外的第三方管理机构或硬件设施，没有中心管制，除了自成一体的区块链本身，通过分布式核算和存储，各个节点实现了信息自我验证、传递和管理。去中心化是区块链最突出最本质的特征。

开放性。区块链技术基础是开源的，除了交易各方的私有信息被加密外，区块链的数据对所有人开放，任何人都可以通过公开的接口查询区块链数据和开发相关应用，因此整个系统信息高度透明。

独立性。基于协商一致的规范和协议(类似比特币采用的哈希算法等各种数学算法)，整个区块链系统不依赖其他第三方，所有节点能够在系统内自动安全地验证、交换数据，不需要任何人为的干预。

安全性。只要不能掌控全部数据节点的 51%，就无法肆意操控修改网络数据，这使区块链本身变得相对安全，避免了主观人为的数据变更。

匿名性。除非有法律规范要求，单从技术上来讲，各区块节点的身份信息不需要公开或验证，信息传递可以匿名进行。

因为这些特征，收费单位和用户可以互相信任，避免了传统人工可能会造成的问题。

2.2 LoRa 通信

2.2.1 LoRa 定义

LoRa 是一个基于开源的 MAC 层协议的低功耗广域网标准，基于 Sub-GHz 的频段使其更易以低功耗远距离通信，可以使用电池供电或者其他能量收集的方式供电。LoRa 是 LPWA 通信技术中的一种，是美国 Semtech 公司采用和推广的一种基于扩频技术的超远距离无线传输方案。这一方案改变了以往关于传输距离与功耗的折衷考虑方式，为用户提供一种简单的能实现远距离、长电池寿命、大容量的系统，进而扩展传感网路。目前，LoRa 主要在全球免费频段运行。

基于低功耗、广域网 LoRaWAN 协议的物联网应用级网关路由器，可以为物联网设备提供低功耗的、可移动的、安全的本地双向无线通信服务，支持 LoRa, 2G/3G/4G, WiFi 和 GPRS 多种无线通讯功能。采用星型架构设计，前端节点设备通过单跳给单个或多个网关，实现终端设备和云平台之间的无线通信，也可以通过标准的 IP 连接到云平台。可广泛应用于智慧城市、智慧社区、智慧消防、智慧农业、智慧水务等领域。

2.2.2 LoRa 无线技术的主要特点

- (1) 长距离：10~20km
- (2) 节点数：万级甚至百万级
- (3) 电池寿命：3~10 年
- (4) 数据速率：0.3~50kbps

2.3 窄带物联网 NB-IoT

2.3.1 NB-IoT 定义

NB-IoT 是基于蜂窝的窄带物联网，其构建于蜂窝网络，只消耗大约 180KHz 的带宽。可直接部署于 GSM 网络、UMTS 网络或 LTE 网络，以降低部署成本、实现平滑升级。NB-IoT 聚焦于低功耗广覆盖物联网市场，是一种可在全球范围内广泛应用的新兴技术。具有覆盖广、连接多、速率低、成本低、功耗低、架构优等特点。这些优势使其可以广泛应用于多种垂直行业，如远程抄表。此次疫情更加提现远程抄表这一方法的需求是巨大的。



2.3.2 NB-IoT 优势

一是覆盖广, NB-IoT 的覆盖范围比传统的 GSM 网络要好 20 个 db。如果按照覆盖范围计算, 一个基站可以提供 10 倍的覆盖范围;

二是能支持海量连接, 200KHz 频率下, 借助 NB-IoT 一个基站可以提供 10 万个连接;

三是低功耗, NB-IoT 通讯模组电池可以十年独立工作, 而不需要充电;

四是低成本, NB-IoT 模组的成目标小于 5 美金。

2.4 无线技术简要介绍与对比

无线技术简要介绍与对比

	Bluetooth	WiFi	ZigBee	Z-Wave	SigFox	LoRa	NB-IoT
频段	2.4GHz	2.4GHz 5GHz	868MHz / 915MHz, 2.4GHz	868.42MHz(欧洲) 908.42MHz(USA)	SubG免授权频段	SubG免授权频段	SubG授权频段
传输速率	1Mbps-24Mbps	11b: 11Mbps 11g: 54Mbps 11n: 600Mbps 11ac: 1Gbps	868MHz:20kbps 915MHz:40kbps 2.4GHz:250kbps	9.6kbps或40kbps	100bps	0.3-50kbps	<100kbps
典型距离	1-100m	50-100m	2.4GHz band: 10- 100m	30m (室内) ~100m (室外)	1-50km	1-20km	1-20km
发射功率	1-100mW	终端36mW, AP320mW	1-100mW	1mW	<100mW	<100mW	<100mW
典型应用	鼠标、无线耳机、 手机、电脑等邻近节点数据交换	无线局域网, 家庭, 室内场所高速上网	家庭自动化、楼宇 自动化、远程控制	智能家居, 监控 和控制	智慧家庭、智 能电表、移动 医疗、远程监 控、零售	智慧农业, 智 能建筑, 物流 追踪	水表、停车、宠 物跟踪、垃圾桶、 烟雾报警, 零售 终端

对于智能水表远程抄表这一环节，LoRa 和 NB-IoT 两者都可以实现，但两者仍有区别。

第一，频段，服务质量和成本。LoRa 工作在 1GHz 以下的非授权频段，在应用时不需要额外付费，NB-IoT 和蜂窝通信使用 1GHz 以下的频段是授权的，是需要收费的。LoRa 模块在处理干扰、网络重叠、可伸缩性等方面具有独特的特性，但却不能提供像蜂窝协议一样的服务质量。NB-IoT 出于对服务质量的考虑，不能提供类似 LoRa 一样的电池寿命。如果需要确保应用场景，推荐使用 NB-IoT，而低成本和大量连接是首选项的话 LoRa 是不错的选择。

第二，电池寿命。关于电池寿命方面有两个重要的因素需要考虑，节点的电流消耗以及协议内容。LoRa 是一种异步的基于 ALOHA 的协议，也就是说节点可以根据具体应用场景需求进行或长或短的睡眠;而蜂窝等同步协议的节点必须定期地联网,这样就额外的消耗了电池的电量。

所以对于需要频繁通信、较短的延迟或者较大数据量的应用来说 NB-IoT 或许是更好的选择，而对于需要较低的成本、较高的电池寿命和通信并不频繁的场景来说 LoRa 更好。

第三，网络覆盖和部署时间表。NB-IoT 标准在 2016 年公布，除网络部署之外，相应的商业化和产业链的建立还需要更长的时间和努力去探索。LoRa 的整个产业链相对已经较为成熟了，产品也处于“蓄势待发”的状态，同时全球很多国家正在进行或者已经完成了全国性的网络部署。

第四，设备成本。对终端节点来说，LoRa 协议比 NB-IoT 更简单，更容易开发并且对于微处理器的适用和兼容性更好。同时低成本、技术相对成熟的 LoRa 模块已经可以在市场上找到了，并且升级版还会接踵而至。

LoRa 可以利用传统的信号塔、工业基站甚至是便携式家庭网关来进行。构建基站和家庭网关价格便宜。但是对于 NB-IoT 来说，升级现有的 4G LTE 基站的价格保守估计每个不少于 15000 美元。

但是两者目前都是 5G 之前的技术，目前 5G 还在开发，或许基于 5G 的高速度、低时延、低功耗，可以完全取代 LoRa 和 NB-IoT。

3 具体功能实现

3.1 实验一区块链基本代码

3.1.1 代码分析

区块链是由很多个区块所构成的链条，每个区块都拥有自己存储的数据和哈希值，首先先产生一个创世区块，接下来的每个区块都存取自己和上一个区块的哈希值，如果哈希值被篡改则链条断裂。区块链还引入了 Proof of Work 工作量证明机制，区块链会控制计算哈希值的难度，难度动态改变会使得进行计算的时间是几乎不会改变的，所以要篡改时所需的工作量远超收益，避免了篡改区块数据的现象。当记录了转账信息时，区块链就变成了一种数字货币，然后矿工用穷举法计算哈希值，当一个答案被计算出来时就会广播到全网，挖到这个区块的矿工就获得一定的比特币奖励。每生成一个新区块时就会产生一个转账记录，而奖励的比特币每四年会减半。广播到全网后，所有的区块都会对这个新区块进行验证。验证通过后，新区块就会加到区块链上。

3.1.2 代码实现

```
12 function proofOfWork() {
13     let data = "luotuo";
14     let x = 1;
15     while (true) {
16         if (
17             sha256(data + x)
18             .toString()
19             .substring(0, 1) !== "0"
20         ) {
21             x = x + 1;
22         } else {
23             console.log(sha256(data + x).toString());
24             console.log(x);
25             break;
26         }
27     }
28 }
29
30 proofOfWork();
```

定义工作量证明机制

```
138   addBlockToChain(newBlock) {
139       newBlock.previousHash = this.getLatestBlock().hash;
140       newBlock.mine(this.difficulty);
141       this.chain.push(newBlock);
142   }
143   mineTransactionPool(minerRewardAddress) {
144       const minerRewardTransaction = new Transaction(
145           null,
146           minerRewardAddress,
147           this.minerReward
148       );
149       this.transactionPool.push(minerRewardTransaction);
150       const newBlock = new Block(
151           this.transactionPool,
152           this.getLatestBlock().hash
153       );
154       newBlock.mine(this.difficulty);
155
156       this.chain.push(newBlock);
157       this.transactionPool = [];
158   }
```

实现挖矿

```
• Console

console.log
挖矿结束 053ccee344d003be955ddd07e246d5665d8e9252e345df699715cd5d6b902da0

  at Block.mine (src/blockChain.js:87:13)

console.log
挖矿结束 042192386e0879769454b96c6bacf1ea0d86c03e71997f9b66f424c23d59e78b

  at Block.mine (src/blockChain.js:87:13)

console.log
[
  Block {
    transactions: '我是祖先',
    previousHash: '',
    timestamp: 1591604612441,
    nonce: 1,
    hash: '082af6f7f3fbccfea7b9fd637f628d55e37dea74f04588f54fa5ceca2da6c9f5'
  },
  Block {
    transactions: [ [Transaction], [Transaction] ],
    previousHash: '082af6f7f3fbccfea7b9fd637f628d55e37dea74f04588f54fa5ceca2da6c9f5',
    timestamp: 1591604612855,
    nonce: 45,
    hash: '042192386e0879769454b96c6bacf1ea0d86c03e71997f9b66f424c23d59e78b'
  }
]

  at Chain.validateChain (src/blockChain.js:187:15)

console.log
挖矿结束 08925a85f158443921d176f26703c5a66a859ad2af4db84a45117fdb8af897f8
```

代码运行后，模拟挖矿过程

3.2 实验二区块链数字签名代码

3.2.1 代码分析

首先生成 hash，之后用私钥对 hash 进行签名。当别人得到这份签名和公钥后，用同样的方法得到 hash，用公钥解密，两个 hash 进行对比就能确定文件是否篡改。私钥绝对私密，不能给予他人，否则文档可以被篡改。

3.2.2 代码实现

```
6  const key = ec.genKeyPair();
7  console.log('private', key.getPrivate('hex'))
8  console.log('public', key.getPublic('hex'))
9  const doc = "zhuanzhang20yuan"
10 const hashedDoc = sha256(doc).toString()
11 const hexSignature = key.sign(hashedDoc, 'base64').toDER('hex')
12 console.log('hashed doc', hashedDoc)
13 console.log('signature', hexSignature)
14 const publicKey = ec.keyFromPublic(key.getPublic('hex'), 'hex')
15 console.log(publicKey.verify(hashedDoc , hexSignature))
16 const tamperedDoc = 'zhuanzhang10yuan'
17 const hashedTamperedDoc = sha256(tamperedDoc).toString()
18 console.log(publicKey.verify(hashedTamperedDoc, hexSignature))
```

获取公钥私钥进行加密，然后验证签名内容是否被篡改

```
console.log
非法交易

    at Block.validateTransactions (src/blockChain.js:94:17)

console.log
非法交易

    at Chain.validateChain (src/blockChain.js:179:17)

console.log
挖矿结束 04813e547a23c32dbd59c6e3d0b2c30b7f933f035052b6e73f2d40e553c25734

    at Block.mine (src/blockChain.js:87:13)

console.log
数据篡改

    at Chain.validateChain (src/blockChain.js:184:17)

console.log
挖矿结束 0a43295cd0b8c92169792a7da8c37235885dbca4729f2ec914f459dab630676f

    at Block.mine (src/blockChain.js:87:13)

console.log
[
  Block {
    transactions: '我是祖先',
    previousHash: '',
    timestamp: 1591604615815,
    nonce: 1,
    hash: '80ea41ae2525e87e846ad3d134450cf1e136f034455077e4121fdd0048ec984d'
  },
  Block {
    transactions: [ [Transaction], [Transaction] ],
    previousHash: '94729d66f7eccf1aa8fa050b653cbec7e01b00d56f40c0e1d0154252c0f8ce3f',
    timestamp: 1591604615562,
    nonce: 7,
    hash: '0a43295cd0b8c92169792a7da8c37235885dbca4729f2ec914f459dab630676f'
  }
]

    at Chain.validateChain (src/blockChain.js:187:15)

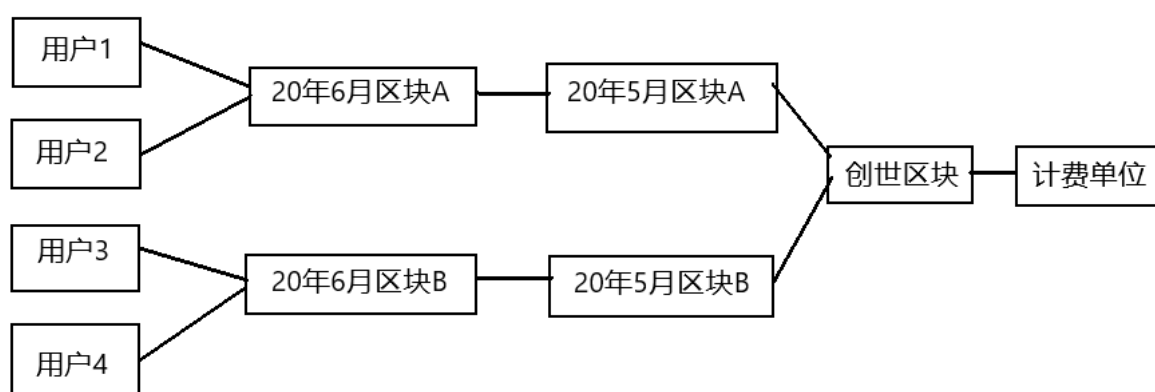
console.log
前后区块链接断裂
```

当检测到数据篡改时区块链断裂

3.3 实验三基于区块链的智能水表代码

3.3.1 代码分析

水表如果应用区块链，则创世水表需要被计费单位拥有，相当于总表的位置，每个地区相当于一个独立的区块进行链接，最后形成一个水表专用的区块链。



用户用水信息一般是以月计量，地区所有用户的用水信息形成一个小型的区块保存，再跟之前月份区块链接，形成链条，同时广播到所有用户，防止区块篡改。如果用户想要篡改自己的信息，则必须花费大量成本，避免了这种现象的发生。同时计费单位只需少量的人力对每个月的区块进行整理，方便统计。

利用微软 azure 服务+truffle 区块链框架实现将模拟树莓派的数据传输至区块链上，并保存数据为哈希值。

3.3.2 代码实现

```

10  abs_dagongiot_niewenhua_niewenhua: {
11    network_id: "*",
12    gas: 4700000,
13    gasPrice: 0,
14    provider: new HDWalletProvider(fs.readFileSync('d:\\xiangmu\\HelloBlockchain\\niewenhua.env', 'utf-8'), "https://niewenhua.blo:3")
15  }
16 }
  
```

对 truffle-config.js 文件进行修改，将.sol 配置文件的区块进行部署。truffle 使用迁移脚本将合同部署到 Ethereum 网络。迁移是项目的 migrations 目录中的 JavaScript 文件。

```

1  const Migrations = artifacts.require("Migrations");
2
3  module.exports = function(deployer) {
4    deployer.deploy(Migrations);
5  };
6

```

对 1_initial_migration.js 进行修改，使用 migrations 方法来部署合约

```

3  contract Migrations {
4    address public owner;
5    uint public last_completed_migration;
6
7    modifier restricted() {
8      if (msg.sender == owner) _;
9    }
10
11    constructor() public {
12      owner = msg.sender;
13    }
14
15    function setCompleted(uint completed) public restricted {
16      last_completed_migration = completed;
17    }
18
19    function upgrade(address new_address) public restricted {
20      Migrations upgraded = Migrations(new_address);
21      upgraded.setCompleted(last_completed_migration);
22    }
23

```

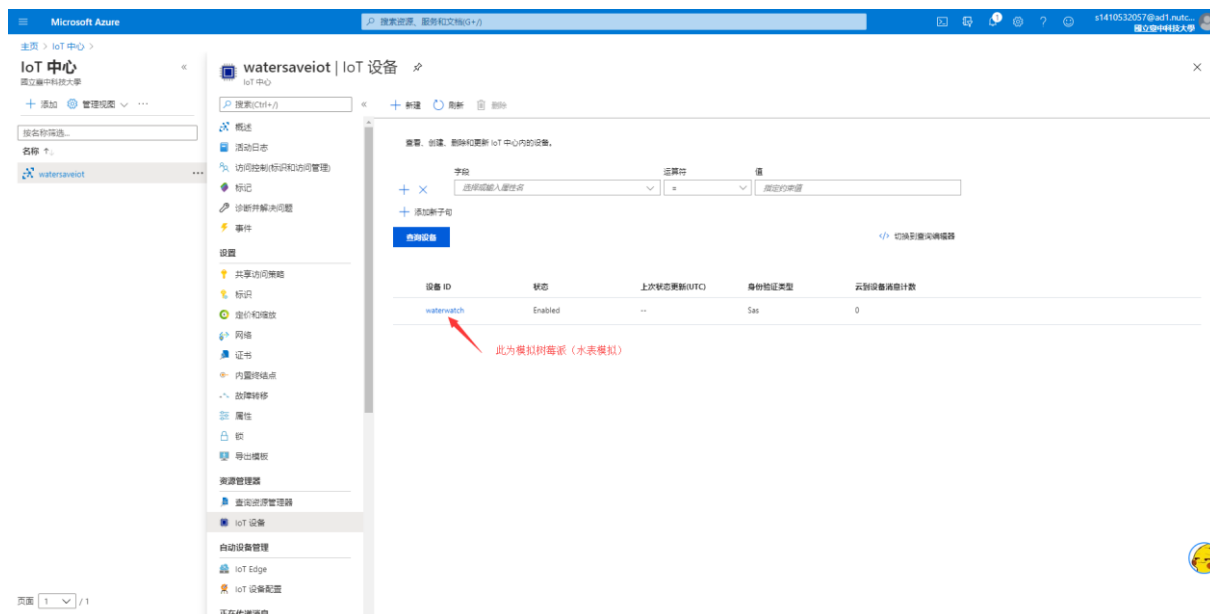
Migrations.sol 则是确保合同不会被重复部署，不会出现相同的合约。

```

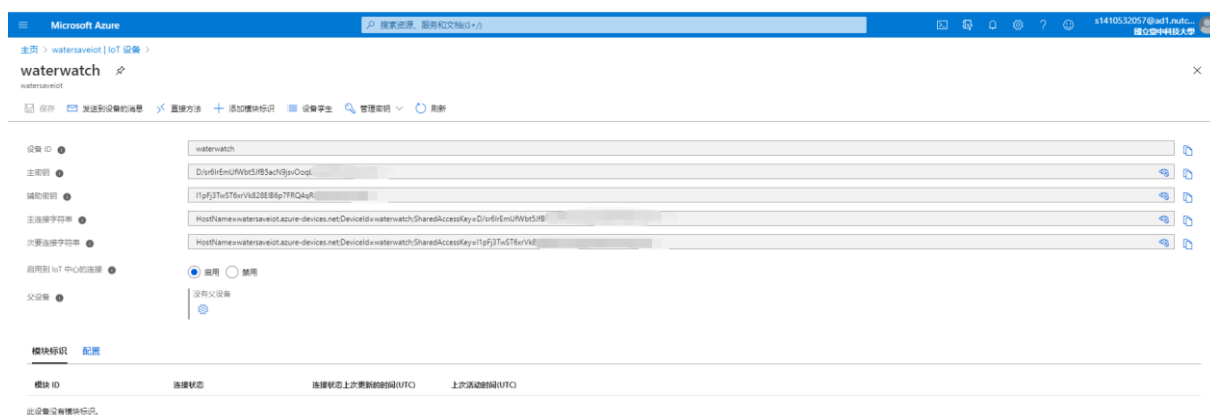
23  function getMessage(cb) {
24    messageId++;
25    sensor.readSensorData()
26      .then(function (data) {
27        cb(JSON.stringify({
28          messageId: messageId,
29          deviceId: 'Raspberry Pi Web Client',
30          wateruse: data.wateruse_C,
31          money: data.money

```

使用 getMessage 函数模拟获取水表的使用量和价钱信息。



在 iot 中心创建模拟水表，获取到设备的连接密钥，同时可以让 Azure 接收到 iot 中心所发来的数据，以便后面流分析工作时能将数据转出。



将水表的连接密钥填写到树莓派的连接函数上，可以让模拟树莓派连接到建立的模拟水表设备，同时发送各种数据给 Azure 测试使用。

```
14  
15 const connectionString = '  
16 const LEDPin = 4;  
17
```

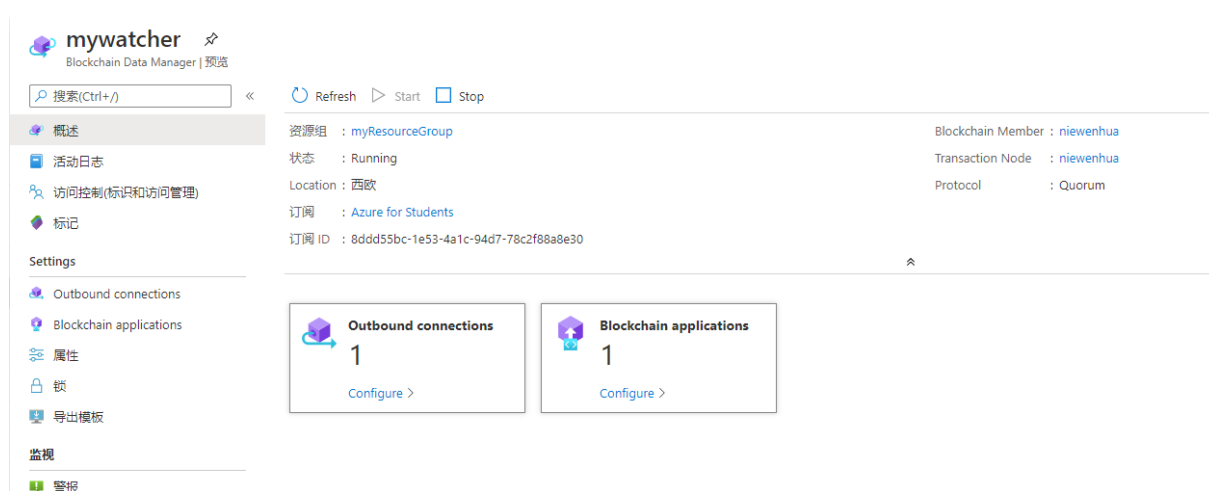
在此添加密钥，进行模拟树莓派和 azure iot 中心的连接

```

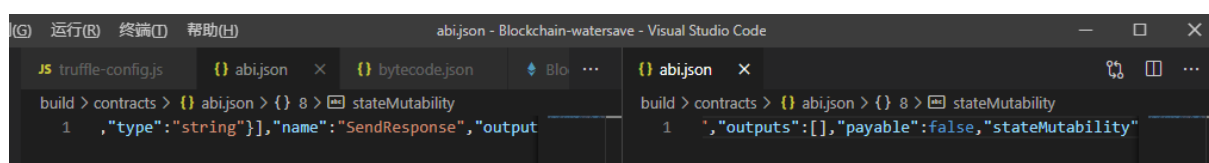
运行 重置
Message sent to Azure IoT Hub
>
Sending message: {"messageId":315,"deviceId":"Raspberry Pi Web Client","temperature":21.16554433202178,"humidity":62.02813913717651}
>
Message sent to Azure IoT Hub
>
Sending message: {"messageId":316,"deviceId":"Raspberry Pi Web Client","temperature":26.696422456684562,"humidity":62.11715601195242}
>
Message sent to Azure IoT Hub
>
示例停止。

```

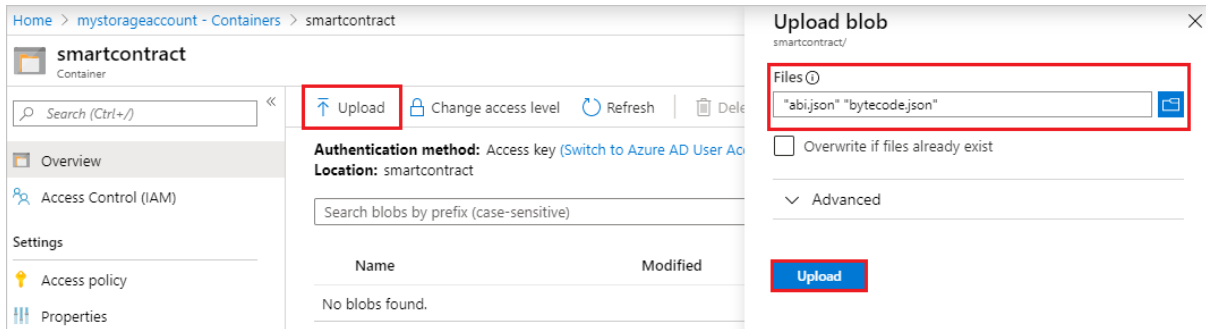
连接成功后，运行模拟设备，iot 中心会接受到数据，同时将数据发送到流分析工作。流分析工作会将数据进行处理，转成 json 码的形式发送至区块链数据管理器。



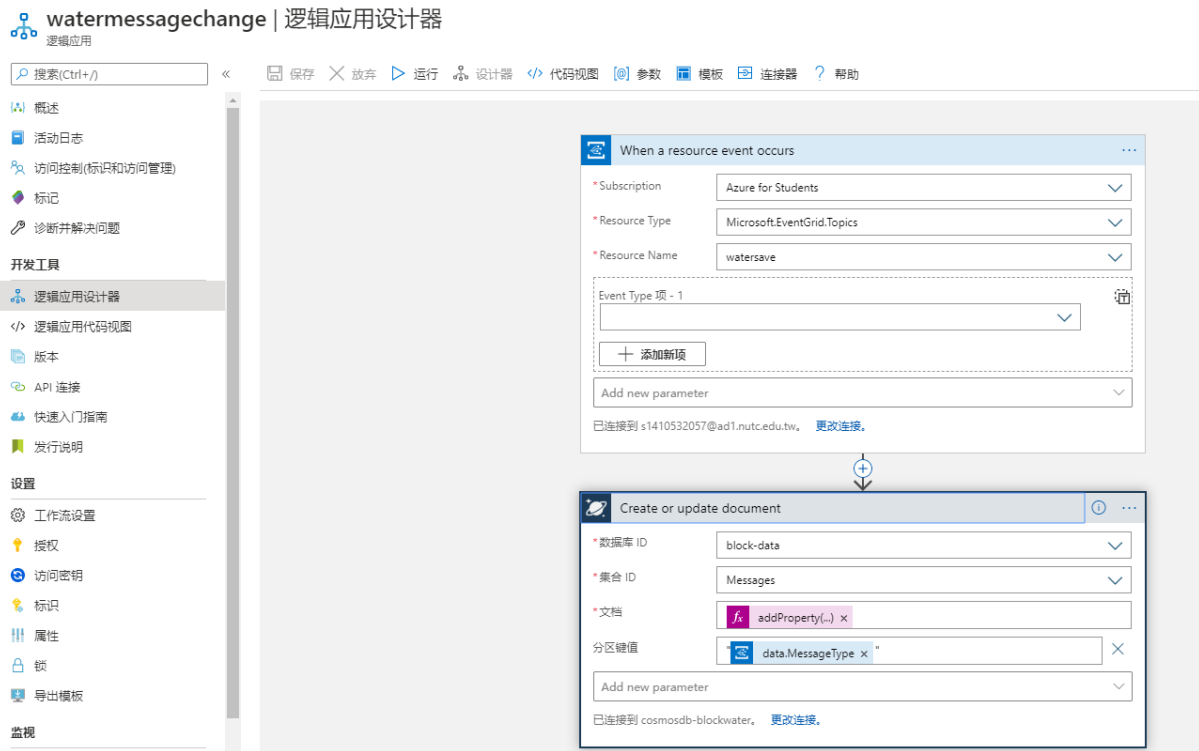
区块链数据管理器实例连接并监控 Azure 区块链服务事务节点。实例从事务节点捕获所有原始块和原始事务数据。出站连接将区块链数据发送到 Azure 事件网格。创建实例时，可以配置单个出站连接。



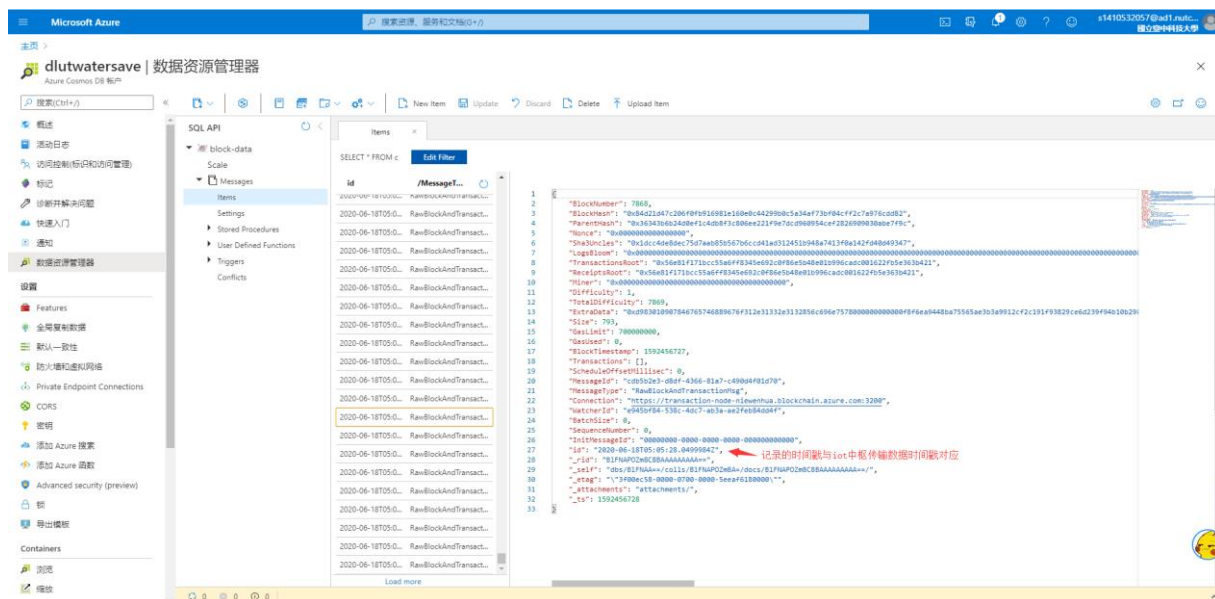
将“字节码”值保存为 JSON 文件。添加应用程序时，区块链数据管理器要求由 URL 访问合同 ABI 和字节码文件。可以使用 Azure 存储帐户提供可私下访问的 URL。



将两个 JSON 文件上传到容器后，生成对应的 URL。



每个逻辑应用都必须从触发器开始，该触发器在模拟水表的树莓派进行传输数据的情况下触发。每当触发器触发时，逻辑应用引擎就会创建一个逻辑应用实例来启动并运行工作流。使用 Azure 事件网格触发器将区块链事务数据从事件网格发送到 Cosmos DB。将 Cosmos DB 与使用 truffle 搭建成功的区块链服务连接，获取到水表发送的数据。



当树莓派将数据传送到 iot 中心后,通过流分析中心可以将 iot 中心的数据传送到区块链中,生成对应的哈希值进行保存。

通过时间戳对照,我们能够发现树莓派发出的数据能够成功被 iot 中心接收,并且发送至 Cosmoc DB 接收,最后转换为区块中的哈希值。

4 总结

水表基本上用于贸易结算，用户面大量广，无疑是最重要的法制计量器具之一。水表必须符合所使用的国家法规的要求。在我国水表列为重点管理计量器具，属强制性检定计量仪表并有具体的实施方法；国内很多地区开展实施的不入户抄表、一户一表、阶梯水价政策需要适用可靠的水表产品支持，电子远传水表、智能型水表有着广阔的前景。生活品质的提升会有更多的城市实现管道直饮水，高计量等级、流量测量范围宽的水表将会有更多的市场，水资源的紧张导致的水价提高是不争的趋势，也会使优质水表价有所值。

通过技术不断地发展和创新，不仅仅是应用在智能水表的 LoRa 和 NB-IoT，5G 时代的到来也预示着万物互联这一概念能够开始被广泛的应用在各处，中国也会更快的跨入万物互联时代。

参 考 文 献

- [1] HCNA-IoT 华为认证物联网工程师[C]. 华为, 2017.
- [2] 环球表计. 智能水表的海外布局与中国困境 [EB/OL]. (2016, 08, 17) [2020, 06, 01].
https://www.sohu.com/a/110829095_472929.
- [3] 中国信通院. 区块链白皮书 [C/OL]. 来源: 中国信息通信研究院和可信区块链推进计划 (2019, 10) [2020, 06, 01].
<http://www.caict.ac.cn/kxyj/qwfb/bps/201911/P020191108365460712077.pdf>.