Report: RFID Scanner for Blockchain

Assignment Member:

Shashwat Khare

2017A8PS0249P

Introduction:

This project uses a Raspberry Pi along with an RFID scanner as an IoT device and the EOS blockchain as the global distributed platform. The IoT device scans RFID chips in keycards, tags, etc. The tags can be used to uniquely identify the items they are attached to in order to provide proof of location and time in supply chain, manufacturing, asset tracking and access control applications.

EOS was selected for the project because it offers these key features:

- EOS offers fast transaction times: a scanned tag appears on-chain within 2 seconds typical.
- EOS has no transaction fees. The blockchain resources are purchased up-front and one-time only. The IoT device can then transact within allocated resources without incurring additional fees.
- EOS transactions are efficiently packed binary structures minimizing bandwidth requirements.
- EOS uses the same globally recognized and hardened security standard as other blockchains like Bitcoin and Ethereum.

We'll connect the MFRC522 RFID chip/antenna board and buzzer to the Pi's expansion header, download our device software repository to a directory on our Raspberry Pi, run `npm install`, and then `sudo node rfid-scanner-eos-rpi.js`. Place tags within 2 cm of the antenna and view the web application (at https://github.com/EOSIoT/rfid-html) to see the scanned tag UID appear with seconds.

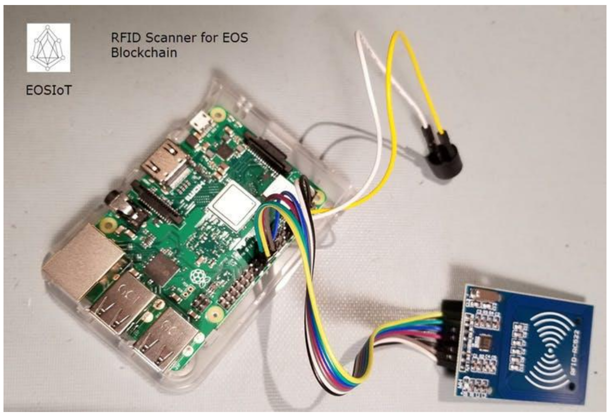
Hardware used:

- Raspberry Pi 3 Model B
- MFRC522 (RFID Reader)
- Active buzzer. These are simple two-wire tone-generators, like this- https://components101.com/buzzer-pinout-working-datasheet.

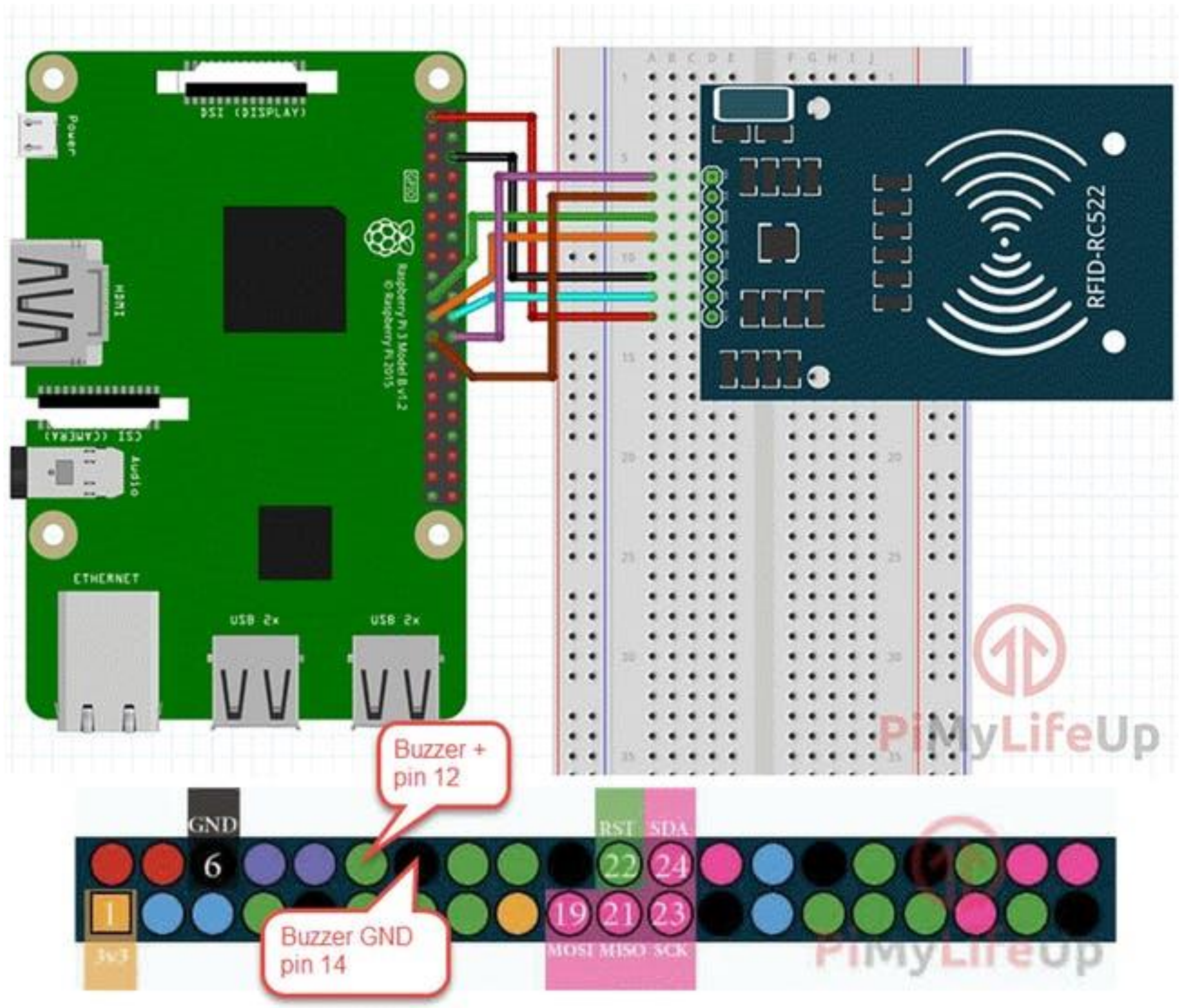
Hardware setup:

Get the Raspberry Pi's up and running.

The hardware setup should look like:



Wire it up like:



Software Setup:

- Latest Raspbian OS
- Recent Node.js. The version (8.11.1) that comes with Raspbian works.
- NPM package manager. Install it then update it:
  - sudo apt-get install npm
  - sudo npm i -g npm
- Git (already installed)

Clone the repository onto your Pi as follows and go to the location of the cloned folder:

- $ git clone https://github.com/EOSIoT/rfid-scanner-node.git
- $ cd rfid-scanner-node

Install project dependencies:

- $ npm install

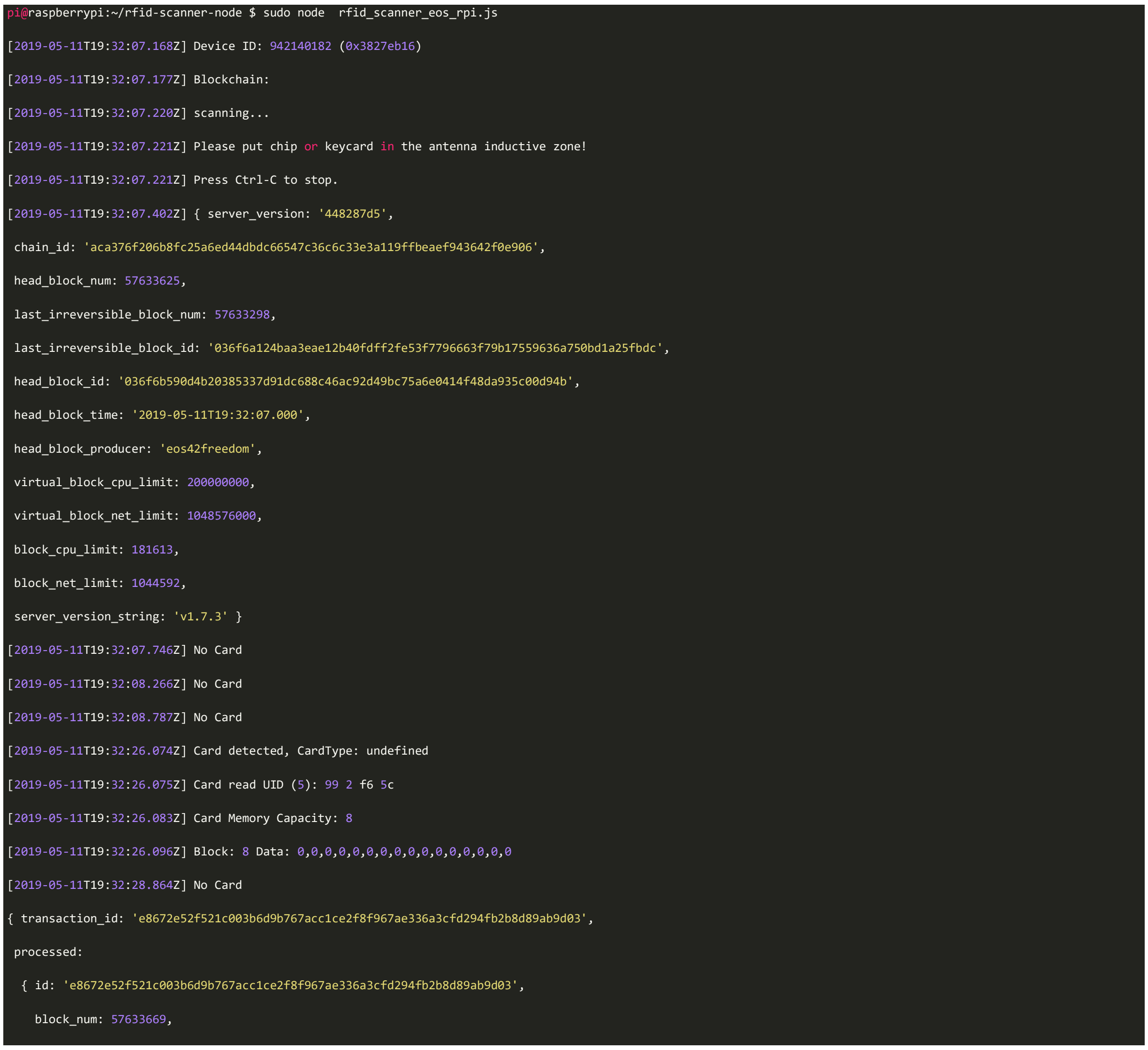
That's it for the software. The private key to sign transactions to the EOS blockchain dApp is already encoded in the software. Through EOS' flexible permission management system, a custom permission was created just for the task of submitting RFID data.

Using the IoT device:

Run the application with super-user privileges. Take note of the unique device ID to reference the scanner's data in the demo web application. In the example below, the device ID is 942140182. Write this down as you'll need it to filter for your scanner's data in the web application.

If the application was setup correctly, you will see the results of an initial blockchain info request, showing that communication with the designated EOS API endpoint is working.

Example:



```
pi@raspberrypi:~/rfid-scanner-node $ sudo node  rfid_scanner_eos_rpi.js

[2019-05-11T19:32:07.168Z] Device ID: 942140182 (0x3827eb16)

[2019-05-11T19:32:07.177Z] Blockchain:

[2019-05-11T19:32:07.220Z] scanning...

[2019-05-11T19:32:07.221Z] Please put chip or keycard in the antenna inductive zone!

[2019-05-11T19:32:07.221Z] Press Ctrl-C to stop.

[2019-05-11T19:32:07.402Z] { server_version: '448287d5',

 chain_id: 'aca376f206b8fc25a6ed44dbdc66547c36c6c33e3a119ffbeaef943642f0e906',

 head_block_num: 57633625,

 last_irreversible_block_num: 57633298,

 last_irreversible_block_id: '036f6a124baa3eae12b40fdff2fe53f7796663f79b17559636a750bd1a25fbdc',

 head_block_id: '036f6b590d4b20385337d91dc688c46ac92d49bc75a6e0414f48da935c00d94b',

 head_block_time: '2019-05-11T19:32:07.000',

 head_block_producer: 'eos42freedom',

 virtual_block_cpu_limit: 200000000,

 virtual_block_net_limit: 1048576000,

 block_cpu_limit: 181613,

 block_net_limit: 1044592,

 server_version_string: 'v1.7.3' }

[2019-05-11T19:32:07.746Z] No Card

[2019-05-11T19:32:08.266Z] No Card

[2019-05-11T19:32:08.787Z] No Card

[2019-05-11T19:32:26.074Z] Card detected, CardType: undefined

[2019-05-11T19:32:26.075Z] Card read UID (5): 99 2 f6 5c

[2019-05-11T19:32:26.083Z] Card Memory Capacity: 8

[2019-05-11T19:32:26.096Z] Block: 8 Data: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

[2019-05-11T19:32:28.864Z] No Card

{ transaction_id: 'e8672e52f521c003b6d9b767acc1ce2f8f967ae336a3cfd294fb2b8d89ab9d03',

 processed:

  { id: 'e8672e52f521c003b6d9b767acc1ce2f8f967ae336a3cfd294fb2b8d89ab9d03',

    block_num: 57633669,
    block_time: '2019-05-11T19:32:29.000',

    producer_block_id: null,

    receipt: { status: 'executed', cpu_usage_us: 345, net_usage_words: 15 },

    elapsed: 345,

    net_usage: 120,

    scheduled: false,

    action_traces: [ [Object] ],

    except: null } }

[2019-05-11T19:32:29.382Z] No Card
```

To scan a card or tag, place it near (within 2 cm) the mfrc522 reader's top (antenna) side. You'll hear a beep sound from the buzzer confirming the scan. The tag's UID and the current time on the Raspberry Pi IoT device is then bundled up into a transaction and sent to the EOS blockchain where it is quickly absorbed into a block.

The RFID scanner software has a private key enabling it to submit tag data on behalf of the eosiot11node account. The tag's UID data is placed into a database hosted by a smart contract under the eosiot11rfid account.

The data generated by the RFID scanner is accessed via simple REST API (https://developers.eos.io/eosio-nodeos/reference) calls or a JavaScript library (https://github.com/EOSIO/eosjs). A simple web application (https://github.com/EOSIoT/rfid-html) can be used to see the scanned tag data appear on the EOS blockchain in near real-time.

A Note on the EOS Blockchain:

A blockchain smart contract eosiot11rfid and a device account eosiot11node is already setup and waiting to receive the RFID scanner data, so nothing has to be done on the blockchain side to see the scanner data appear on the chain. The account being "billed" for transaction bandwidth and the account hosting the smart contract are both located on the mainnet chain - the EOS flagship blockchain (real-time status).

The blockchain account created for this project (eosiot11node) is shared between every IoT device that uses the software. There is a limited amount of bandwidth and CPU allocated to it, and with enough users the daily limits will be reached. However, we can setup our own device account(s) where we can control access and allocate resources, and even deploy our own dApp (to see the scanned tag data appear on the EOS blockchain in near real-time).

A finite amount of resources has been pre-allocated to support this project (requiring EOS tokens with real value). The demonstration device account eosiot11node can support a limited number of transactions per day and is therefore suitable for demonstration purposes only. However, note that this limit is only a function of how much EOS is staked to the account bandwidth.

A separate node account can be created just for your RFID scanner node(s) and an appropriate amount of EOS can be allocated to support your expected transaction requirements.

References:

https://www.hackster.io/firmwareguru/build-an-rfid-scanner-for-blockchain-1fbdb3

https://github.com/EOSIoT/rfid-scanner-node

demo web application: https://github.com/EOSIoT/rfid-html