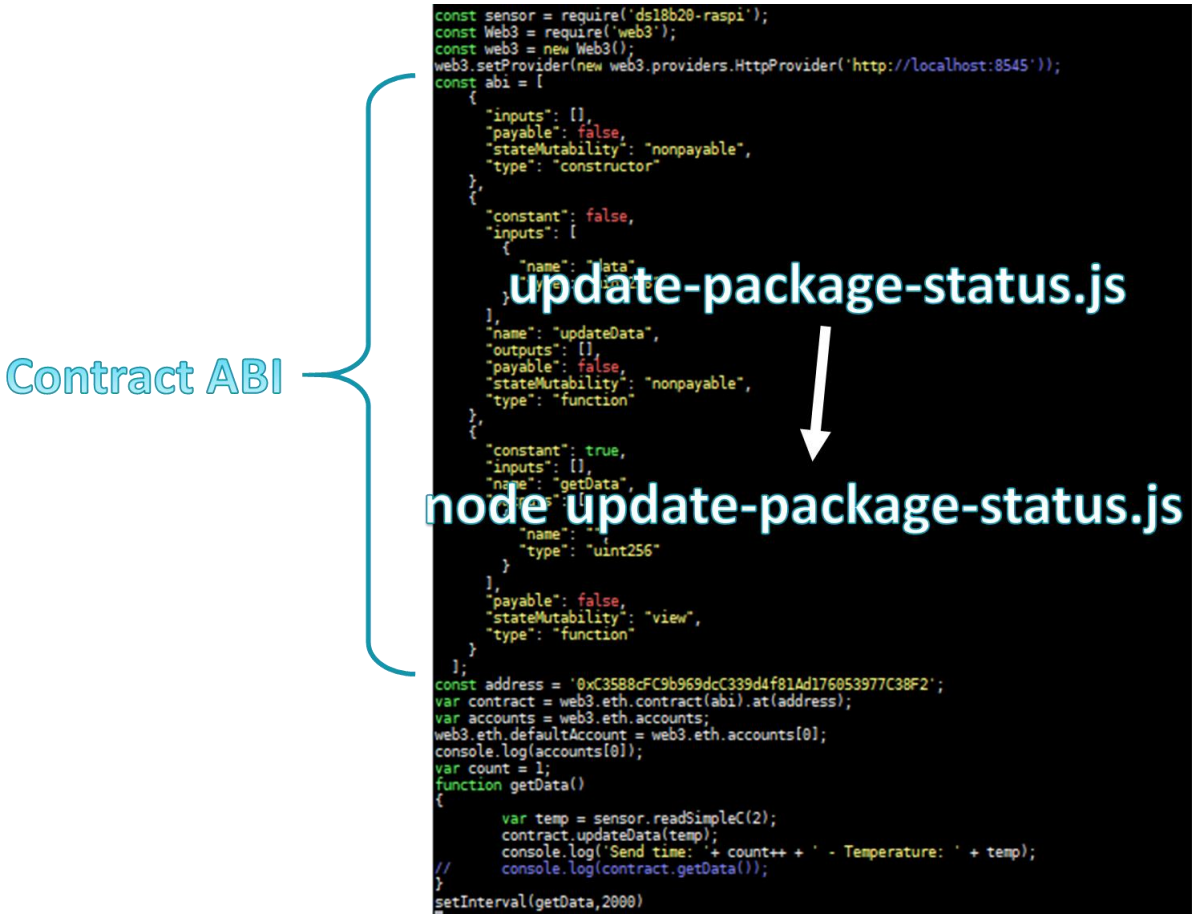
Phần vận chuyển, các source code nằm trên Raspberry số 2 (mật khẩu là raspi - nếu cần)

Một số gói cần cung cấp để thực hiện việc cập nhật từ Raspberry Pi bao gồm: **node, nodejs, npm,** các thư viện cần thiết cho cảm biến. Đặc biệt là package của ***Web3js*** để PI tự chạy được code và có thể đẩy dữ liệu lên Blockchain thông qua thông số lấy nhập vào Web3:

https://web3js.readthedocs.io/en/v1.2.0/getting-started.html



Source code lưu tại địa chỉ folder: **~/Desktop/LogisticsJS/**

tại folder này có 2 folder và 2 file bao gồm:

- **/First-Test-Sensor** : folder chứa các file test đọc các loại cảm biến viết bằng .js

- **/build** : là folder thư viện của các cảm biến.

- **check_balance_infura.js** : file nodejs để kiểm tra kết nối giữa Pi và tài khoản được chạy trên Infura (liên kết với Rinkeby)

- **update-package-status.js** : là file chính thức để cập nhật dữ liệu từ cảm biến lên sản phẩm nhờ barcode.

Trong phần code chính này có 4 phần code cần lưu ý:

```
web3.setProvider(new  web3.providers.HttpProvider("https://rinkeby.infura.io/v3/22e36e35c19148adbdd7d769761f0259"));
```

➔ Đây là phần cung cấp địa chỉ API của Infura vào code.

```
const contractAddress = '0x55b2900b87b2eda7e3cbe61af513d737f4980106';
```

➔ Đây là phần cung cấp địa chỉ Smart Contract đã được Deploy lên Ethereum, lấy địa chỉ này trên: https://rinkeby.etherscan.io/

```
const accountAdress = "0x13f4d28863c5Cd9f53E0c4F8f6CaD3C921b648fc";
```

➔ Đây là phần cung cấp địa chỉ của Account Ethereum hiện tại.

```
const privateKey = '0x9CAD73E956F9304D5392E27411B3F4B1C8C590B71DE495DC53AD6D77F68DCC1A';
```

➔ Đi đôi với Account Ethereum là privateKey, ta có thể lấy được privateKey này tại MetaMask, khi muốn xuất ra

privateKey, hệ thống sẽ yêu cầu nhập mật khẩu ví, với mật khẩu là "***iotblockchain***".

Cú pháp chạy: ***node update-package-status.js <barcode món hàng>***

VD: ***node update-package-status.js 113***