[システム概要（仮）]

ラズパイに備え付けたセンサ（今回は温度を測定）により取得した温度、設定した地域、数値を取得した時間をブロックに保存する。

定期的に取得した温度を保存する方式では、手数料がかかってしまうので、ある一定の温度以上になったら、データ登録処理を行うといった工夫を行う。

また、他のノードはその値を参照することができる。

ダイアグラム

自動的に生成された説明

[使うもの（仮）]

* ラズベリーパイ4B(ストレージ：32GB、OS:raspbian)

-センサを備え付け、数値を取得できるようにする

-apt-getコマンドなどを利用し、数値を取得できるように設定

* Node.js+web3

-センサから取得した数値を定期的に取得し、スマコンの関数を使いデータの登録を行う

* JavaScript、HTML、web3

-スマコンの関数を呼び出し、参照を主に行う

-地域情報の登録もここで行う

* Solidity

-コントラクト処理を記述するために使用

* Ethereumネットワーク（テストネットワーク）

-今のところはパブリックを使用、状況によってはプライベート型も検討

[コントラクト概要（仮）]

* 地域を登録、参照できるコントラクト
* 関数１（引数１：地域名）： 送信者のIDと引数の地域名を結びつける
* 関数２（引数１：送信者ID）：送信者のIDに結び付けられた地域名を取得する
* 時間を取得できるコントラクト
* 関数１：現在時刻（年月日）を取得し、それを返す
* 温度を登録、参照できるコントラクト（地域、時間のコントラクトを継承）
* 温度クラス：地域名、時間、温度で構成
* 関数１（引数１：温度）：温度クラスにそれぞれの値を格納し、それを返す
* 関数２：関数１の返り値を配列に追加する
* 関数３（引数１：ID）：IDと一致するデータを返す
* 関数４（引数１：地域名）：地域名と一致するデータを返す

[問題点など]

* 処理が非常に遅く、リアルタイムには向かない（別ネットワークを使用？）
* 保存したデータは、全員が見れるようにするのか、特定の人が見れるようにするのか
* 定期的にデータを保存してしまうと、手数料がかかる