

CLOUDCANDE

2012.03.24

日本ANDROIDの会原子力部

「NINJA」

オープンフォース総統

河野

NINJAとは？

- NUCLEAR INSPECTOR
NETWORK JAPAN

由来

- 日本ANDROIDの会
- 四国支部と会津若松支部
- 同様なことをやっていたので合同でワイヤレス
ジャパンに出展
- 2011.6 合同チームへ
- 2011.6 福島ガイガーカウンター勉強会開催
- 2012.2 福島ガイガーカウンター勉強会 IN
TOKYO #8 開催
- 他

福島ガイガーカウンター勉強 会

- ガイガーカウンターを自作する
- 標準線源を使って校正する
- 測定値の計算方法を開発
- 文献の翻訳

CLOUDCANDE

- NINJAの提案するマッピングソリューション
- 計測値をクラウドにマッピング
- 特徴
- 誤差があることを前提

基本コンセプト

- ゲノム解析システムの応用
- 断片をシーケンス、アセンブル、全貌を明らかに
- 個別のデータはあてにならない
- B 遮蔽しているかどうか？
- 車で測定？歩いて測定？
- 測定時間は？

SVを表示する測定器

- 大体、5分ぐらいかかるものである
- 移動平均を真面目にとると感度が低く見える
- なので突発的に大きな値になったときは
- 移動平均を小さく取り直す
- 感度が大きく見える

内部でどうなっているのかわ からない

- 換算係数はCO60? CE-137?
- 何倍も異なる
- SVを信頼することはできない
- ぴっと鳴るものをカウント→CPM
をアップロードする

今回話をすること

- 計測器→データ収集→スマートフォン
やPC→クラウド→DB→HTMLビュー
- 今回はクラウドにデータ投入するところ
まで
- 発表は4部作
- OSCTOKYO/OSCEHIME/ABC
/JAVAONE

計測器

- 自動でデータ収集できるもの
- 一般用途でサービスポートについているのはほとんどないので
- HACKする

予備知識

- 放射線の種類
- 汚染の種類
- 計測器の種類
- データ収集の種類
- マッピングソリューションの種類

放射線の種類

- α 線、 β 線、 γ 線
- α 線は肺胞への吸塵時に強い毒性があり、その他にはあまり影響がない
- β 線は今回の汚染で広範に検出するが透過力が弱く測定条件で100倍以上の変動が発生する
- γ 線はマッピング測定のためには最も手軽に測定できるが弱いエネルギーでの検出なので工夫が必要

汚染の種類

- ウラン系元素
 - 原発内部またはごく近辺のみ
- 分裂核系元素
 - セシウム、ストロンチウム、ヨウ素の同位体
- バックグラウンド
 - 宇宙線やカリウム
- 幸いにして、大放出が一度きりだったので、組成はどこでもほぼ同じ
- 半減期から見てCS-137をターゲットに

計測器の種類

- ガイガー
- 半導体
- シンチレーション

ガイガー

- 原理：高圧電場の中に放射線を入れると気体全体が放電する。その信号を利用。
- 400V以上の高圧が必要
- 本来はベータ線しか反応しないものが多い
- Γ 線は？
- 光電効果で β 線に変換したものを計測する
- なので感度が低い

半導体

- 原理：半導体のギャップを放射線で乗り越える
- 小型化できる
- 体積が少ないのでガイガーに比べると反応が弱い
- 回路上の問題で振動で誤作動することが多い
- 大体1時間ぐらい安置しておかないといけない

シンチレータ

- 原理：蛍光を発する結晶に放射線をあてると光を発するのでそれを計測する
- 安定、感度高い、値段も高い

データ収集の種類

- スマートフォン
- PC
- 接続方法
 - いくつも接続方法があるがどれも統一して使えるように
- バリエーション広くして間口を広く

マッピングソリューションの 種類

- みんなでつくる放射線量マップ
- 全国放射能情報
- 国、自治体による高さ1M,0.5M計測を中心とした放射線量マップ
- データ量が何よりものを言う
- いろいろ批判があるが
- 国の結果と同じような結果を描き出す
- 国よりも速い

我々の試みの紹介

計測器、接続方法、データ投入

ケース1 (ガイガー方式)

- 自作ガイガーカウンター
- ガイガー管輸入
- 高圧回路は使い捨てカメラのフラッシュ回路

ケース2（半導体方式）

- 半導体方式はBEEP出力が無いものが多い
（EX,エアカウンター）
- BUT
- 音声出力に特化したものはむしろ安価に入手可能
- EX,RADIATIONWATCH.ORG(フリスクガイガー)
- 音声入力なのでそれを受けるプログラムを開発

ケース3 (ガイガー、シンチ方式)

- 出来合いの放射線計測器を改造
- 例として、FUKUSHIMA GEIGER COUNTER
- この機種はオープンハードウェアなので回路図が公開されている
- 分解してスピーカ部分から信号を取り出し

ケース4（ガイガー、シンチ方式等）

- モニタリングポスト装置を改造
- MARK2、BLUEV2など
- 元々定置測定用なので、フォーマットを変更するだけで使える

接続方法1

- **USB有線接続**
 - ARDUINOを使った方法
 - 安定している
 - 電力供給も含めてOK
- **センサネットワーク**
 - XBEEを使った方法
 - SUNSPOT
 - 自律収集ネットが構築できる
- **BLUETOOTH**
 - 専用モジュールを開発中
 - 持ち歩きにはこれをお勧め

接続方法2

- WI-FI
 - 未開発
- 有線LAN
 - 単体でLANに接続
 - 専用機
- 音声有線接続
 - IPHONEなどに

クラウドアップロード

- データ形式：JSON形式
- IMAOCANDEフォーマット
- CF, IMAOCANDEマシン

クラウド側

- 乞ご期待！

事例紹介

SUNSPOTを使ったセコサ ネットワーク

SUNSPOT

- 無線センサーネットワークデバイス
- フィジカルコンピューティング
- フルJAVAプログラミング
- い！

ハードウェアスペック

- **USBから充電**
- 小さい
- オンボードセンサーがいっぱい
 - 温度センサー
 - 照度センサー
 - 3軸加速度センサー
- **フルカラーLED X 8**
- **IEEE802.15.4**
- デジタル入出力X4
- アナログ入出力X4
- スイッチ入力X2 (割り込み)

電源スペック

- 電池 3.7V 770MAH
- 3V ピン 電力供給可能
- 大電流出力ピンが4つある
- ソース、シンク300mA 合計
500mA 個別制御
- モーターコントロールなどもできる

センサネットワーク

- SLEEP $30\mu A$
- IEEE802.15.4
- メッシュネットワークが構築できる

コンセプト

- 半導体放射線検出器バージョンの
SUNSPOT
- 数百M程度のフィールドでメッシュ
ネットワークを構築
- データ収集はホストPCI台に
- ノードは太陽電池で駆動

太陽電池

- 2W級 SP80180
180X81X1.5MM
- 5.5V 360MA
- SUNSPOTは低消費電力なので太陽電池でも駆動可能
- 1.4V 240MA(エクトプラズム動作状態)

ただし！

- 太陽電池
- 震災
- 放射能
- これが3つ合わさると進化して何になるかな・・・

詐欺

- 太陽電池ソリューションは99%サギ
- それじゃ我々もサギ？

当然ながら

- 夜はダメよ
- 晴天時のみ
- 厳密なものではなく、ベストエフォートで
- ベストエフォートも詐欺だよなあ

1日のうち、昼間の数時間の み動作

- 数値を1時間安定するまで計測して
- 1時間たったら転送する
- 5つホップして順繰りにデータを転送
- 最大255ノードからPCにデータを収集
- 屋外のモニターをメッシュ状にしなければならない
- 農園などで動作

実験1

- 半導体検出器の信頼性試験
- セシウム137標準線源からの理論計測値
- 通常の計測限界を割り込む計測に成功

実験2

- 屋外ノイズ実験
- 風雨の振動を除外することに成功

実験3

- 太陽電池実験
- 太陽電池で満充電
- 4時間
- スリープ後23時間で動作に成功
- 繰り返し5回(5日間)に成功

実験4

- ホップ転送実験
- スリープ中の自律ノード同士で通信させる
- 5ホップに8時間
- 成功率88%

マッピング事例

- (図示)

応用へ

- 放射線以外への応用
- 屋外での長期モニタリング
- 月、年単位
- 農場の植物生育
- 植生変化
- メモリはフラッシュなので通信をせずに年単位のデータ収集もOK

発展形

- SUNSPOT入手できない場合
- ARDUINO+XBEE+シリアル通信
+JAVAで同様のものが動作するよ
うに
- モックアップ作成

今後の予定

- **CLOUDCANDE**サーバサイドアイコンプリメント、サービスアイコン
- 災害管理システム**SAHANA**プラグイン
- 開発ハードウェアのスピコンアウト

スピニアウト例

- BLUETOOTH汎用入出力装置
- INKSCAPEでG-CODE作成
- FT232

参考文献

- PROJECT2108.COM
CLOUDCANDE
- PROGRAMMING SUN
SPOT 2009.08
- もっと知りたい方はオープンフォー
スMLへ！

おしまい