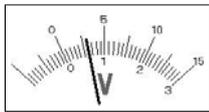


【実習1】

センサ・カメラにさわってみよう

センサの出力あれこれ

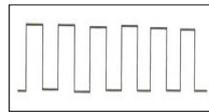
センサは計測した値に応じて電氣的な出力を変化



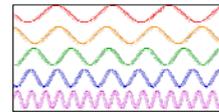
アナログ電圧出力



アナログ電流出力
(風速センサなど)



パルスカウント
(雨量センサなど)



周波数変化
(ECセンサなど)

出力形式によって計測器(受信回路・プログラムなど)が異なる

電圧出力のセンサで色々なものを測ってみよう

圃場計測で使用する主なセンサ2

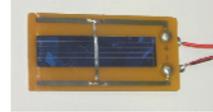


光量子センサ (IKS-27)

$$(\mu\text{mol}/\text{m}^2\text{s}) = V_{\text{out}}(\text{mV}) \times 10$$

価格: 60000円 (小糸工業) →

入力電圧なしで駆動



ソーラパネル(200円)
の発電量で代用

CO2センサ (K30)

$$(\text{ppm}) = V_{\text{out}}(\text{mV}) / 2.135$$

価格: 15000円 (SenseAir)



V_{out} GND 5V

土壌水分センサ (10HS)

$$(\text{m}^3/\text{m}^3) = 5.8 \times 10^{-7} \text{mV}^2$$

$$-2 \times 10^{-4} \text{mV} - 5.8 \times 10^{-2}$$

価格: 15000円 (AINEX)

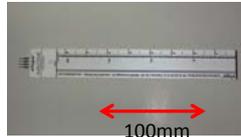


圃場計測で使用する主なセンサ3

水位センサ (eTape)

$$\text{抵抗値} = 1.5(\text{k}\Omega) - 56(\Omega) \times (\text{cm} - 2.54)$$

価格: 5000円 (千石電商)



葉濡れセンサ (Decagon LWS)

濡れ具合に応じて0.32V~1V出力

価格: 24000円 (AINEX)



他にも

変位センサ、距離センサ、風速センサ、荷重センサ、
加速度センサ、超音波センサ、アルコールセンサ…

センサ出力をモニタリング



Arduino UNO (3000円)

A/D: 10bit 6ch (0~1V, 0~5V)

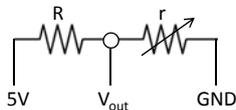
- ・ 入力電圧を $0 \sim 2^{10}-1 (=1023)$ で表示
- ・ 計測範囲は 0~1V か 0~5V に切替可

すなわち

電圧(V) = 生値 / 1023 × 5 (0~5V時)

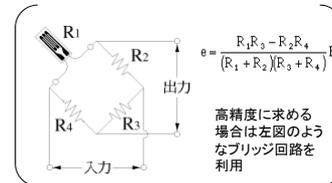
約5mVより小さい変化は認識できない (分解能)

抵抗変化(r)をみる場合は



$$V_{out} = r / (R+r) \times 5 (V)$$

として求める (流れる電流が
大きくならないようRを調整)



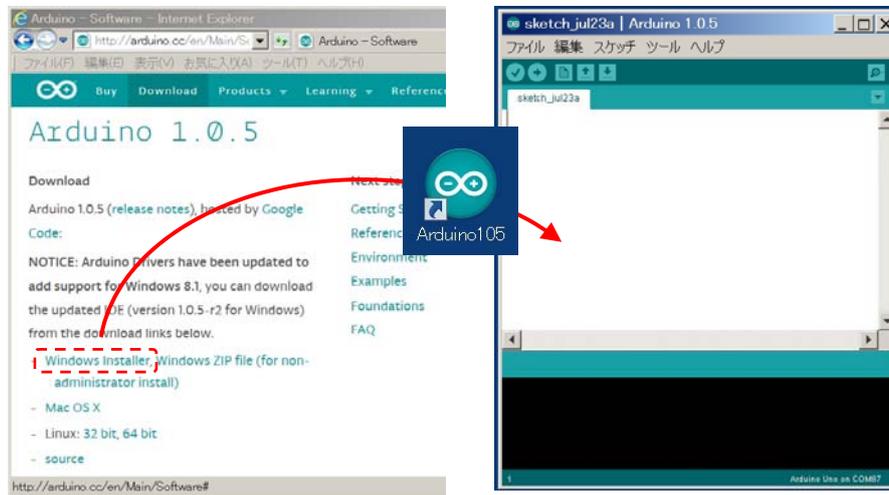
$$e = \frac{R_1 R_3 - R_2 R_4}{(R_1 + R_2)(R_3 + R_4)} E$$

高精度に求める
場合は左図のよう
なブリッジ回路を
利用

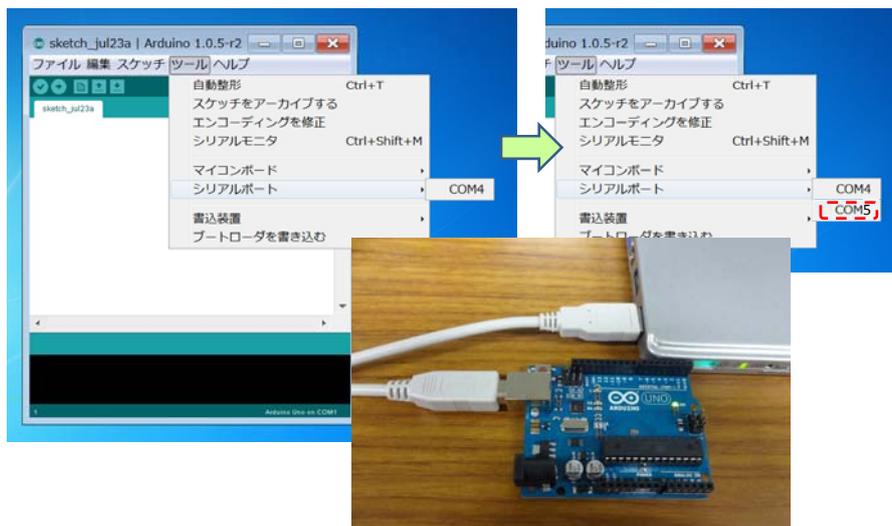
Arduinoの使い方

1. Arduinoのソフト(IDE)をインストール
(<http://arduino.cc/en/Main/Software> 2014/6時点でver.1.05)
2. ArduinoをUSBケーブルでPCに接続
(ArduinoはUSBから電源供給される)
3. Arduino IDEを立ち上げ、Serial Portを設定
(Arduinoを外したときに表示されないCOM*を選択)
4. File - Examples - 1.Basics - Blinkを選択・実行
(→マークのボタンを押すと実行。ArduinoのLEDが1秒間隔で点滅)
5. 分からない部分はGoogleで検索してみよう
(http://www.slideshare.net/slideshow/embed_code/29408197 など)

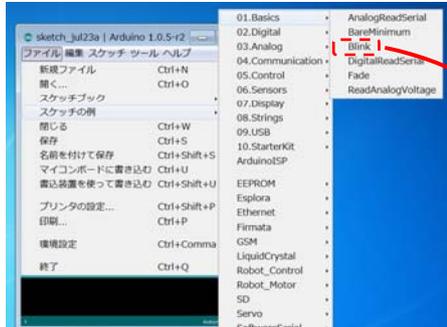
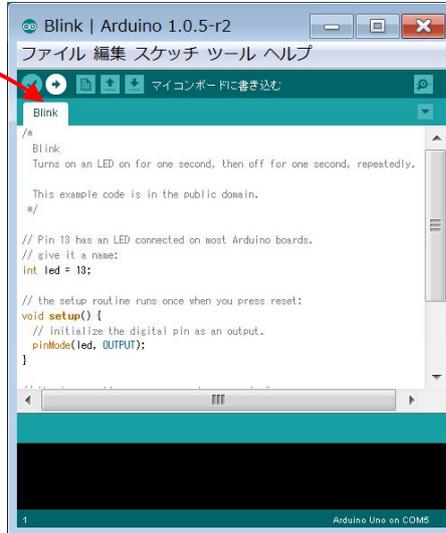
Arduinoの下準備



シリアルポートの調べ方



シリアルポートの調べ方

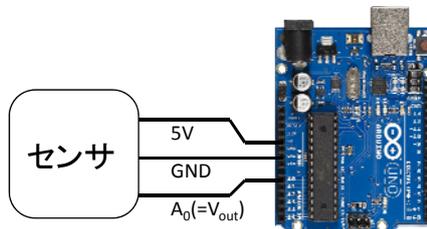

ここが点滅

Arduinoを使ってセンサの値をよむ

1. File—Examples—1.Basics—AnalogReadSerialを選択

```
void setup(){
  Serial.begin(9600);
}
void loop(){
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(1);
}
```

2. センサをケーブルで接続

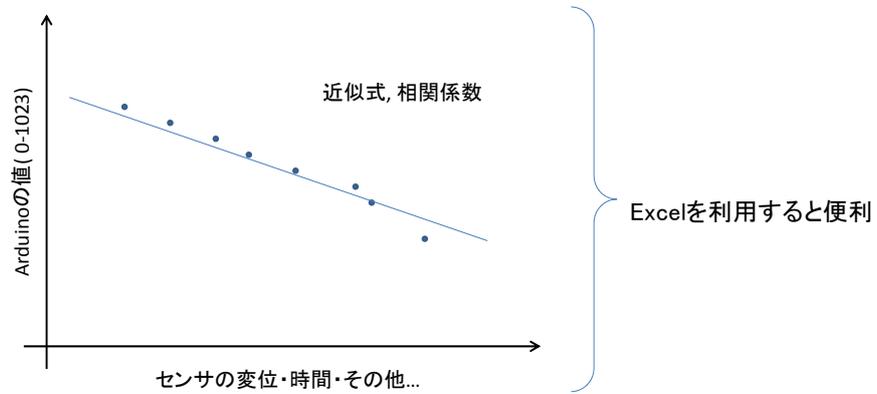


3. 実行してセンサの値を表示

「ツール—シリアルモニタ」で Arduinoから送られるセンサ値を表示



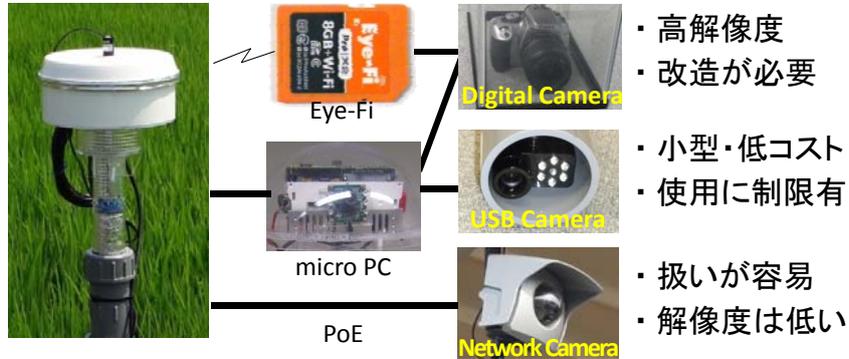
センサ値を変化させてグラフ化しよう



データから何が分かるか考えてみよう

— メモ —

カメラで計測をしてみよう



フィールドサーバでは様々な種類のカメラを接続可能

目的・設置場所・カメラの種類によって扱いが異なる

デジタルカメラで定点観測するには

電源をどうやって供給するか

- ・ 外部電源アダプタのあるカメラが望ましい
- ・ 電源ボタンを押さないと起動しないものはNG
- ・ 電圧が特殊なので電圧変換回路が必要



シャッターをどう制御するか

- ・ インターバル撮影は数分～数時間で自動OFF
- ・ リモコン撮影/レリーズ撮影機能があれば可
- ・ 上記を制御する電子回路が必要



撮影画像をどう取得するか

- ・ Eye-Fi SDカードを利用すれば無線で自動送信
- ・ 他の無線SDカードだと、画像削除ができない
- ・ 一部のカメラはgphoto2(@linuxでUSB接続)利用可



Eye-Fiで定点撮影をしてみよう

画像を送信するクラウドサービスを登録する

1. <http://www.flickr.com>でアカウントを作成
(下の方にあるJoin Flickrをクリック。Yahoo.comのアカウントが作成)
2. もう一度flickrのページにアクセスしてSign In
(Yahoo!のアカウントでログインし、Take me to flickrをクリック)
3. flickrで検索にひっかかるようセキュリティを設定
(5枚以上の写真をアップし、public設定して承認されるまで数日待つ)
4. Eye-Fiに入っているソフトをインストール
(D:¥Autorun¥Aurorun.exeをダブルクリックしてEye-Fi Managerをインストール)
5. Eye-Fiで撮影画像をflickrにアップするよう設定
(Eye-Fi Managerはハングしやすいので定期的に強制終了・再起動)

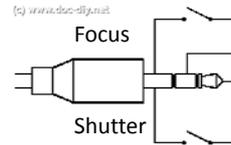
Eye-Fi Managerでの設定

The screenshot displays the Eye-Fi Manager application interface. At the top, there's a navigation bar with 'ファイル', 'アカウント', and 'ヘルプ'. Below it, the 'Eye-Fi' logo and 'Eye-Fiカード' are visible. The main area is divided into two panels: 'ネットワークの設定' (Network Settings) on the left and 'Eye-Fiの設定' (Eye-Fi Settings) on the right. The network settings panel shows 'Wi-Fiネットワーク' set to 'P2-lecture' and a password field. The Eye-Fi settings panel includes a '管理' (Management) section with a checkbox for 'エンドレスモードを使用する' (Use Endless Mode) which is checked. Below this is a progress bar showing 60% usage. A note at the bottom states: '次の場合に領域を空けてください。 60 カードの領域のうち % が使用中です。送信が完了した古い画像から削除されます。' (Please free up space in the next case. 60% of the card's space is in use. Old images whose transmission is completed will be deleted from the card.)

デジカメのインターバル撮影

レリーズ端子がある場合 (ex.Canon EOS kiss)

1. 右図のような信号で撮影制御
(短絡させることでフォーカス/シャッターが動作)
2. 短絡回路をArduinoで制御させる
(フォトカプラAQV212などを利用)



リモコン端子がある場合

1. インターバルタイマリモコンを作製
(SWITCH SCIENCE ¥2,500。分単位になるよう変更)
2. リモコン受光部に固定し動作確認
(カメラ側のオートパワーオフに注意)



ネットワークカメラを利用するには

ネットワークカメラとは (ex. AXIS M1014)

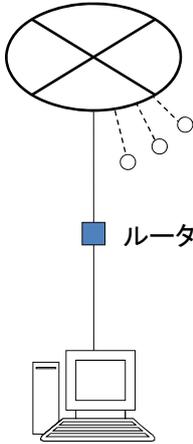


- ・ LANケーブルで接続してWeb上で画像表示
- ・ ネットワーク設定の知識が必要
- ・ 画像の自動取得にはちょっとしたコツが必要

ネットワークの基礎知識

--- IPアドレスとは ---

インターネット



www.google.co.jp ↔ 173.194.126.255

DNSがURL(名前)とIP(番号)を変換

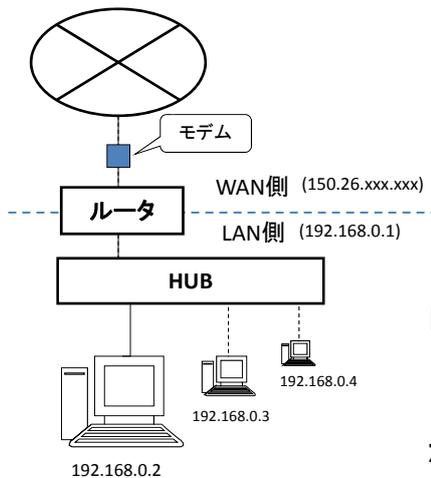
- ・グローバルIP : XXX . XXX . XXX . XXX
(以下のローカルIPを除く XXX: 0~255)
- ↑↓
- ・ローカルIP : 192.168.0.0~192.168.255.255
172.16.0.0~172.31.255.255
10.0.0.0~10.255.255.255

固定IP ↔ 非固定IP (DHCP)

Dynamic-DNS

ネットワークの基礎知識

--- ルータ ---



- ・ DHCPを設定することでルータからIPが自動割振
- ・ 外からはルータしか見えない



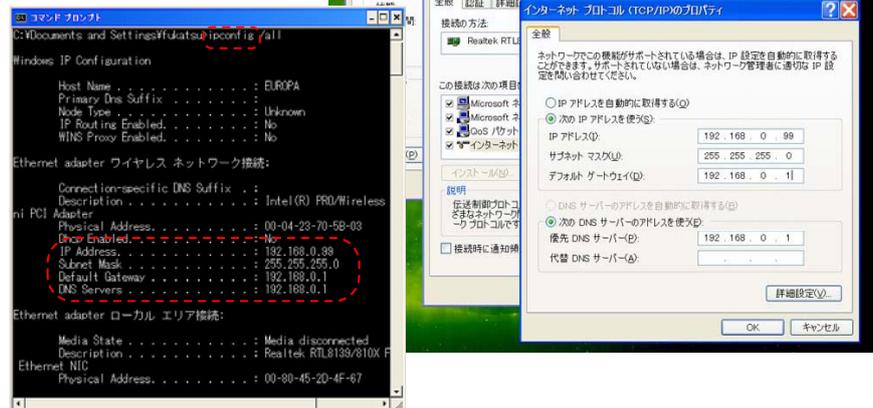
DMZ: ローカルIPのどれかを代表として見えるように

or

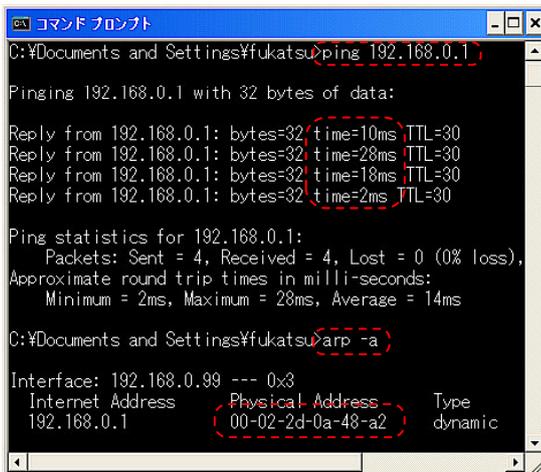
ポートフォワード : ポート毎に設定したローカルIPを表示

PCのネットワーク設定の確認

コマンドプロンプト上で
ipconfig (/all)



相手先の接続チェック



• ping xx.xx.xx.xx

→ 接続可/不可

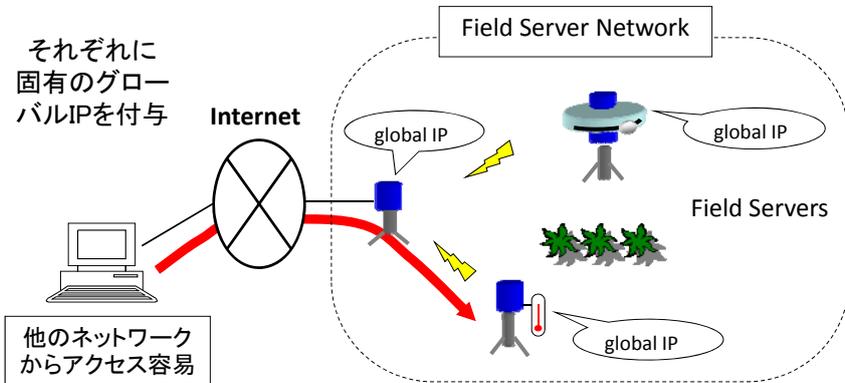
→ 通信状態

• arp -a

→ 接続先MAC

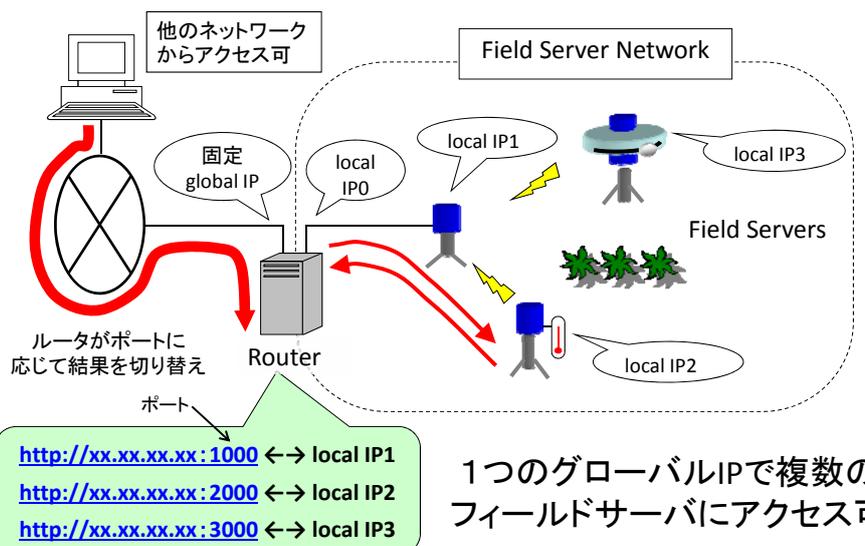
(• arp -d
→ 履歴クリア)

ネットワーク構築(固定グローバルIP)

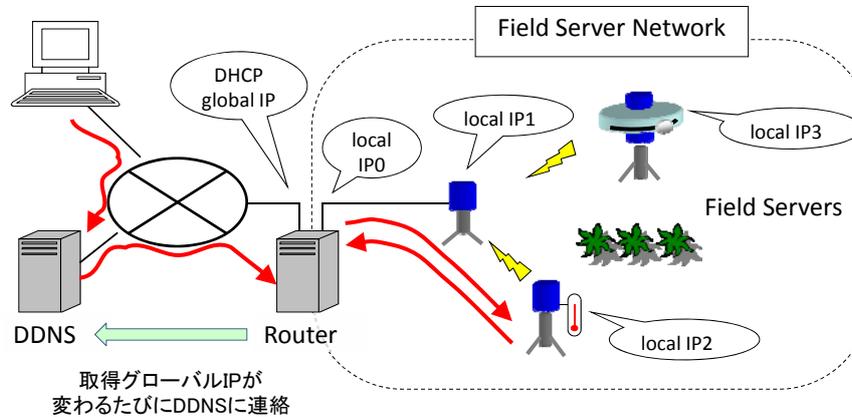


多数の固定グローバルIPの取得は難しい

ネットワーク構築(ポートフォワード)



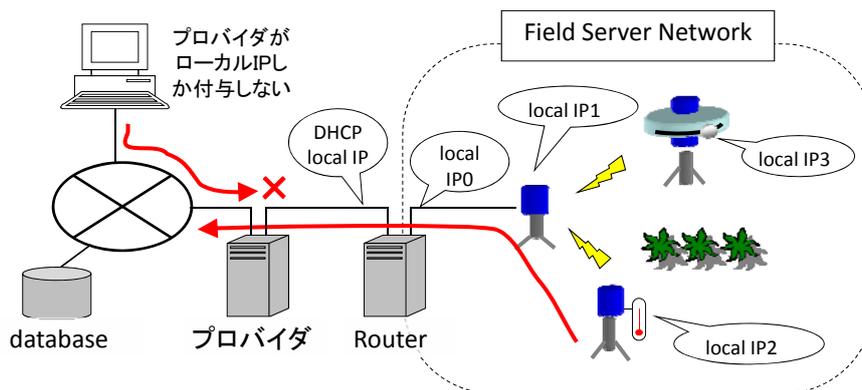
ネットワーク構築(DDNS利用)



<http://fs.iobb.net> ← <http://xx.xx.xx.xx>
<http://fs.iobb.net> ← <http://yy.yy.yy.yy>
<http://fs.iobb.net> ← <http://zz.zz.zz.zz>

DDNSに登録することでIPが変わっても同じURLでアクセス可

ネットワーク構築(ローカルIP取得の場合)



- ・ そのままでは外部からフィールドサーバにアクセスできない
- ・ フィールドサーバ側からデータをPUSHするのは可

ネットワークカメラの画像取得

画像取得のURLを調べて直接アクセス



AXIS: http://**.**.**.*/jpg/image.jpg

松下: http://**.**.**.*/SnapshotJPEG?Resolution=640x480&Quality=Clarity

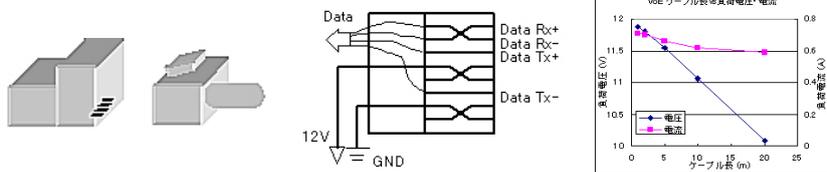
カメラの異常検知 & メール転送機能を利用

- ・ 閾値を下げて異常がなくとも反応するように設定
- ・ メール転送はサーバ側のセキュリティに注意

[SMTP :smtp.mail.yahoo.co.jp/587, SMTP Authenticationでログイン]

ネットワークカメラの設置ノウハウ

PoEで電源ケーブルとLANケーブルを一体化



防水ケースで屋外設置



簡易画像モニタリングシステム



- ・ 既存製品の組み合わせのみで動作するシステムの提供
- ・ 12Vタイムスイッチで省エネ・USB出力コンバータで5V出力

カメラの画像を取得してみよう



- ・ 電源を入れて40秒ほど待つ
- ・ 初期設定ではIPはDHCPで配布 (192.168.20.80-99のどれか)
- ・ 初回にパスワードを登録 (ユーザ名:root/パスワード:passwordとする)

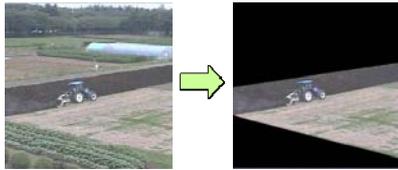
カメラを好きな場所に置いて画像を取得

時間があれば別のカメラでも試してみよう

基本的な画像処理1

画像の切り抜き

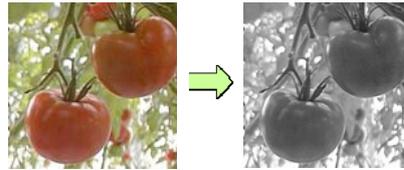
マスク画像の利用や範囲を指定して不要な部分を削除



- ・他の画像処理を行う前処理として
- ・差分処理を行う場合に利用
- ・回転・縮小などの基本処理と併用

グレースケール処理

カラー画像を0-255段階のモノクロ(グレイ)画像に変換

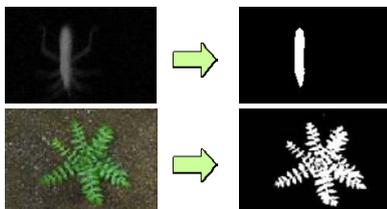


- ・ $\text{gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B$
- ・差分処理・2値化処理の前段階として
- ・濃淡の分布はヒストグラムで確認

基本的な画像処理2

2値化処理

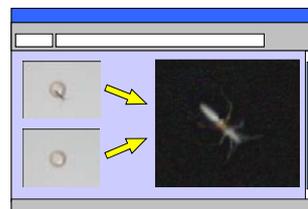
閾値を設定して画像の各点を白か黒のどちらかに変換



- ・対象を抽出し、大きさ・形状が分かる
- ・カラー画像は各RGBの閾値を設定
- ・代表的な閾値設定: 大津の2値化

差分処理

2枚の画像の各点を引き算した画像を生成する

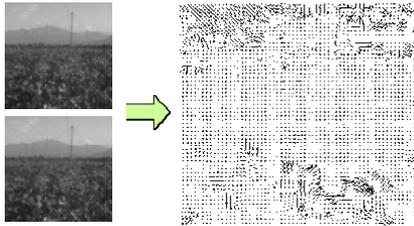


- ・基本はグレースケール画像同士で行う
- ・変化のある部分が明るく表示される
- ・2枚の画像の位置がずれないように注意

基本的な画像処理3

オプティカルフロー

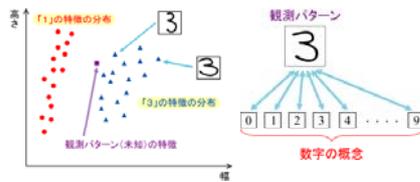
2枚の画像の対応点がどれくらい移動したかベクトル表記



- ・ 対応点の検出範囲をうまく調整
- ・ 画像処理にはOpenCVなどを利用

パターン認識

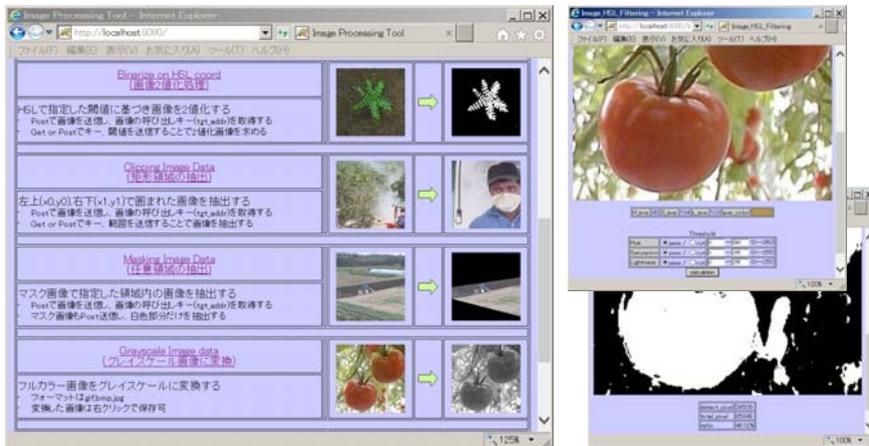
学習データに基づき、どの登録したクラスに近いか判定



- ・ 教師画像で各クラスの特徴を算出
- ・ 判定を行う識別器は多くの種類が存在
- ・ 特定の対象(虫・開花・果実)の検出に

画像処理アプリケーション

<http://192.168.20.2:8080/> でWeb上で画像処理が可能



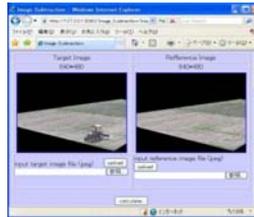
画像処理で大きさを測ってみよう

色々な画像処理を組み合わせて対象の大きさを測る

切り抜き処理



差分処理



2値化処理



- ・最初に大きさが既知の対象で行う(キャリブレーション)
- ・目的の対象で2値化処理後のpixel数を数えて比較

— メモ —