

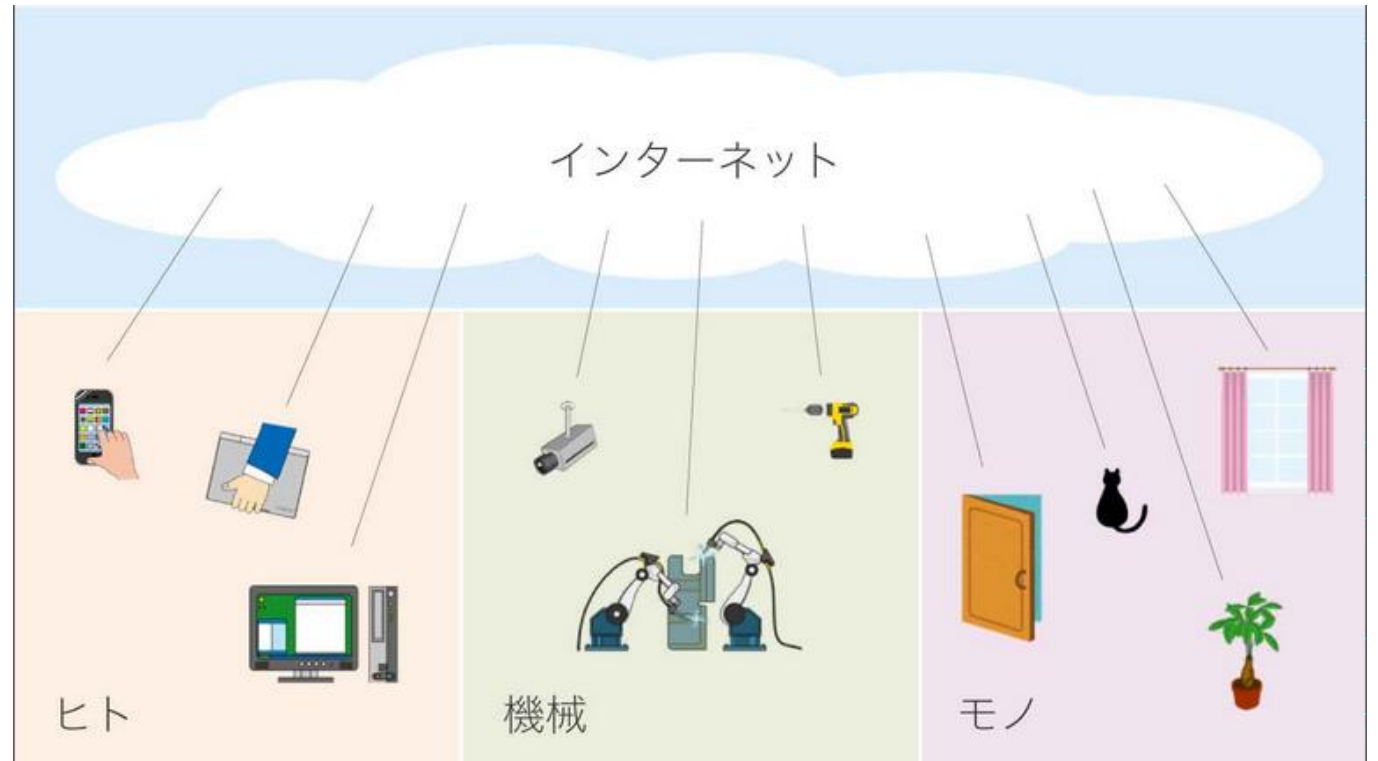
全体ゼミ

中国で卒業論文の簡単な紹介

研究生 郭一凡

モノのインターネットとは (Internet of Things, IoT)

- モノがインターネット経由で通信すること
 - モノを操作する。
 - モノの状態を知る。
 - モノ同士で対話する。
- 計測データ、センサーデータ、制御データの交換をすることで様々な課題解決




去年の研究テーマ

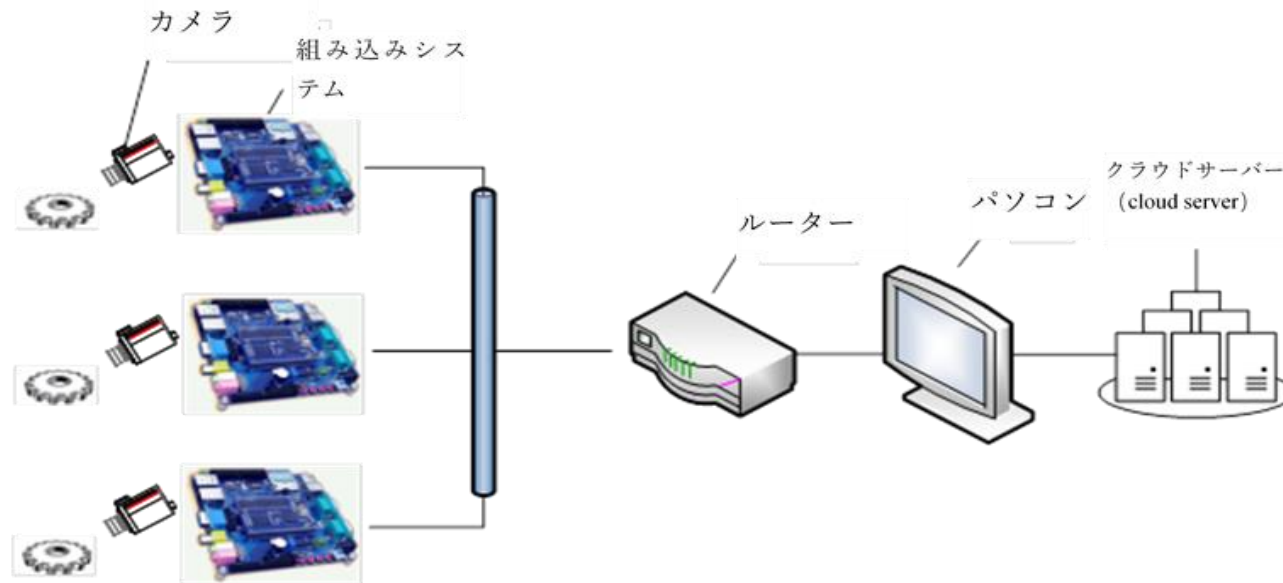
組み込みシステムを用いた産業用画像検出システムの設計

(Design of industrial image detection system based on the embedded system)

なぜこのテーマを選んだ？

- Nanjing Kangni Mechanical & Electrical Co., Ltd.(以下KN会社)からの求め
 - KN会社は中国での地下鉄自動ドアシステムの主要製造者（ほかの主要な製造者はシーメンス(Siemens  実際はライバル)
 - 中国国内市場占有率は50%を超え、グローバル市場占有率は25%（出典：SCI Report,2016年）。
- 自分の専攻と興味に基づいて
 - ハードウェアの実践的なチャンスが多く
 - 信号処理または画像処理に興味がある
- 時間：2017年2月から5月まで

産業用画像検出システム～理想的な場合



- カメラ：部品に撮影
- 組み込みシステム：動画の簡単に処理
- ルーター：データの転送
- パソコン：画像処理と出力
- クラウドサーバー：データベースに入れる

目的

- K N会社生産した部品での傷があるかどうかを確認する、できれば傷の長さを計測する。
- 既知情報:金属製、円筒形
- 「実際の部品は機密であり、現時点では公開されていません」と言われて
- 自分でいくつかのサンプルを模倣した

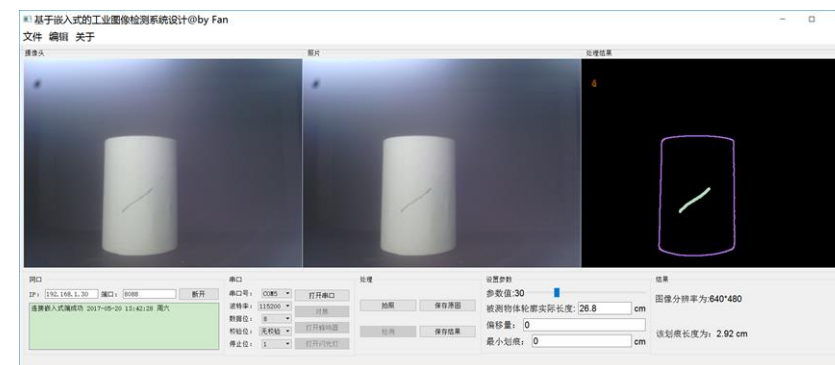
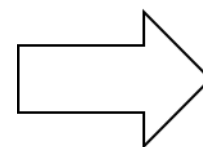
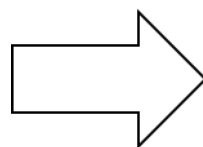
- こんな感じ……
 - (紙とプラスチック製)



- 残念ながら、最後まで本物も見えない。

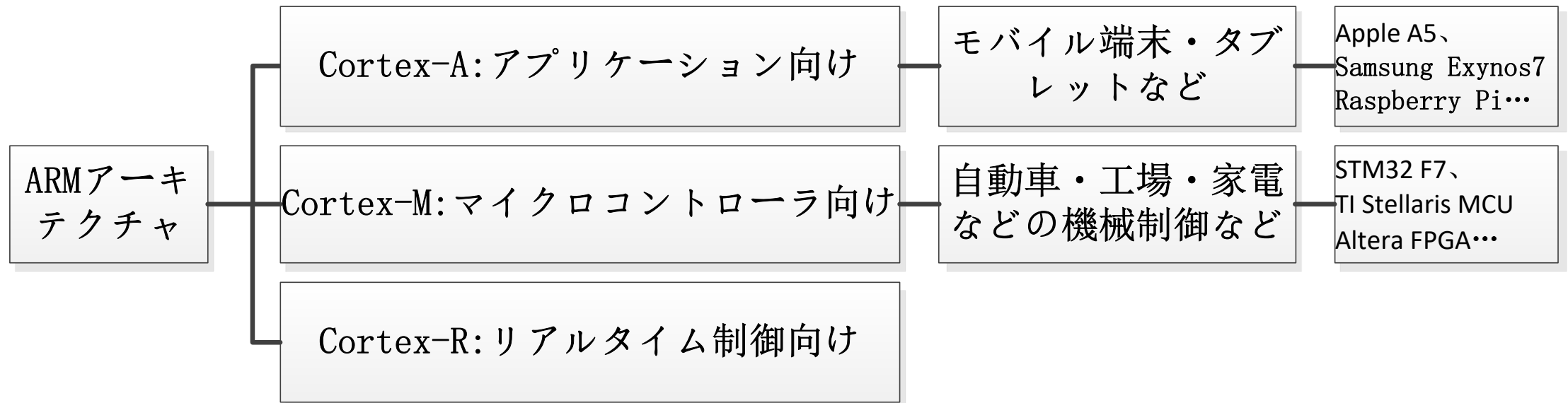
システムの構成

- ハードウェアまたはホストコンピュータ(host computer)
- ソフトウェアまたはマスターコンピュータ(master computer)
- ソフトウェアによる画像処理



ハードウェアの設計～ARMアーキテクチャ

- 2016年ソフトバンク（Softbank）はイギリスのARM会社を買収して240億ポンド（3.3兆円）かかった。
- ARM Cortex-Aシリーズ:携帯の90%以上占有率
- ARM Cortex-Mシリーズ:低電力、IoT向け



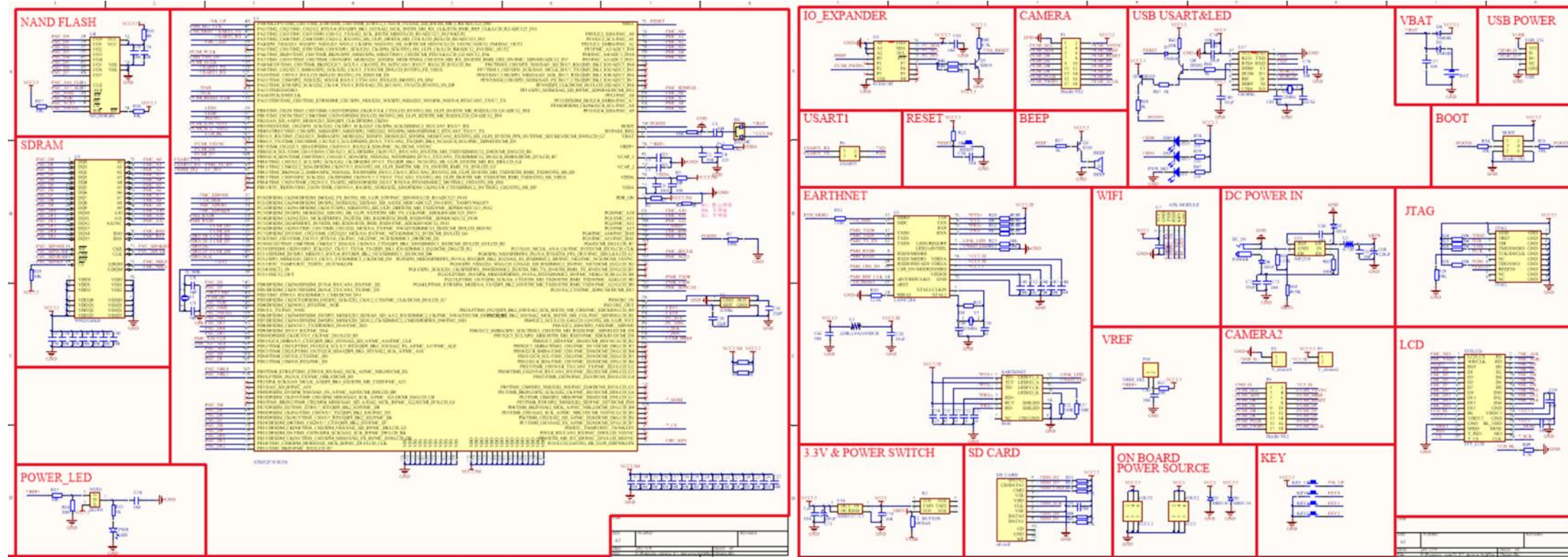
ハードウェアの設計～X86系とARM系の比較(2016年)

	X86系(例Intel i7-7700)	ARM系(例STM32F767IGT6)
周波数(Frequency)	@ 3.60GHz(3601 MHz)	@216MHz
消費電力(Power consumption)	110W	0.5W(0.5ワットぐらい)
比率(Ratio)	0.03W/MHz	0.002W/MHz
値段(Price) (2018/06/21)	40826円	1105円
半導体デバイス製造	14nm	90nm(改善の余地がある)

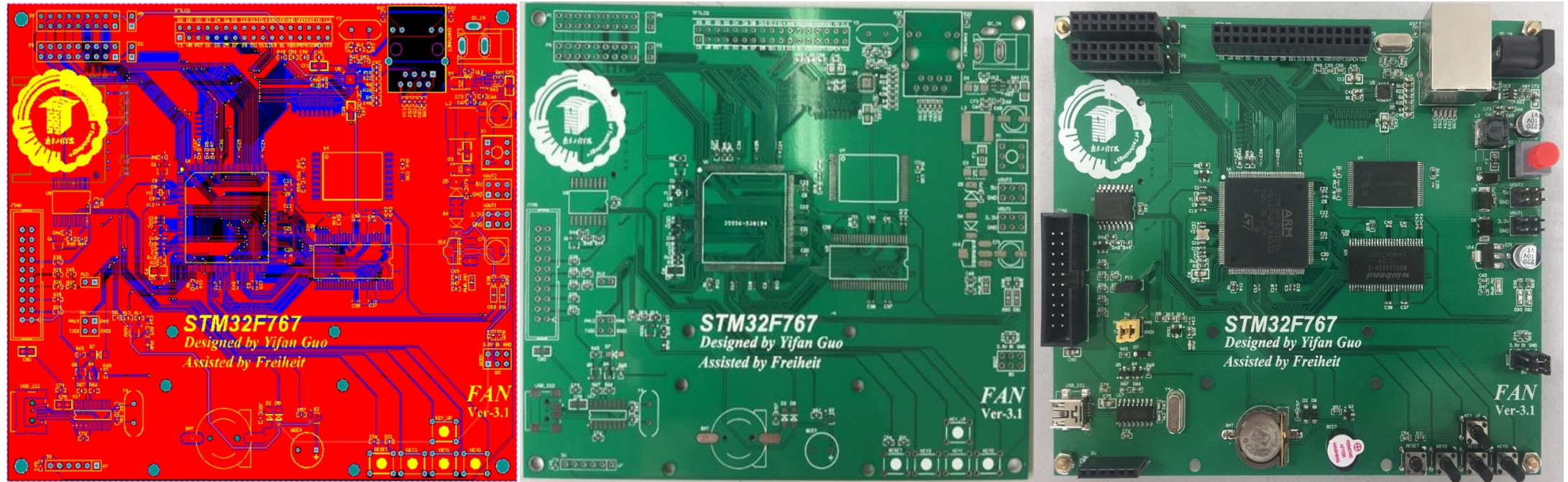
なぜARM系を選んだ??

- Intelと比べてARM Cortex-MシリーズのSTM32F7電力効率に優れる。
- ARM よりIntelの周波数すごく高いが、値段も36倍、比率も15倍！
- カメラ多い、マルチノード(Multi-node)の場合はARM系の32-bit MCU(STM32F767IGT6,2016)を選んだ。

ハードウェアの設計~回路図



ハードウェアの設計~プリント基板(Printed Circuit Board)

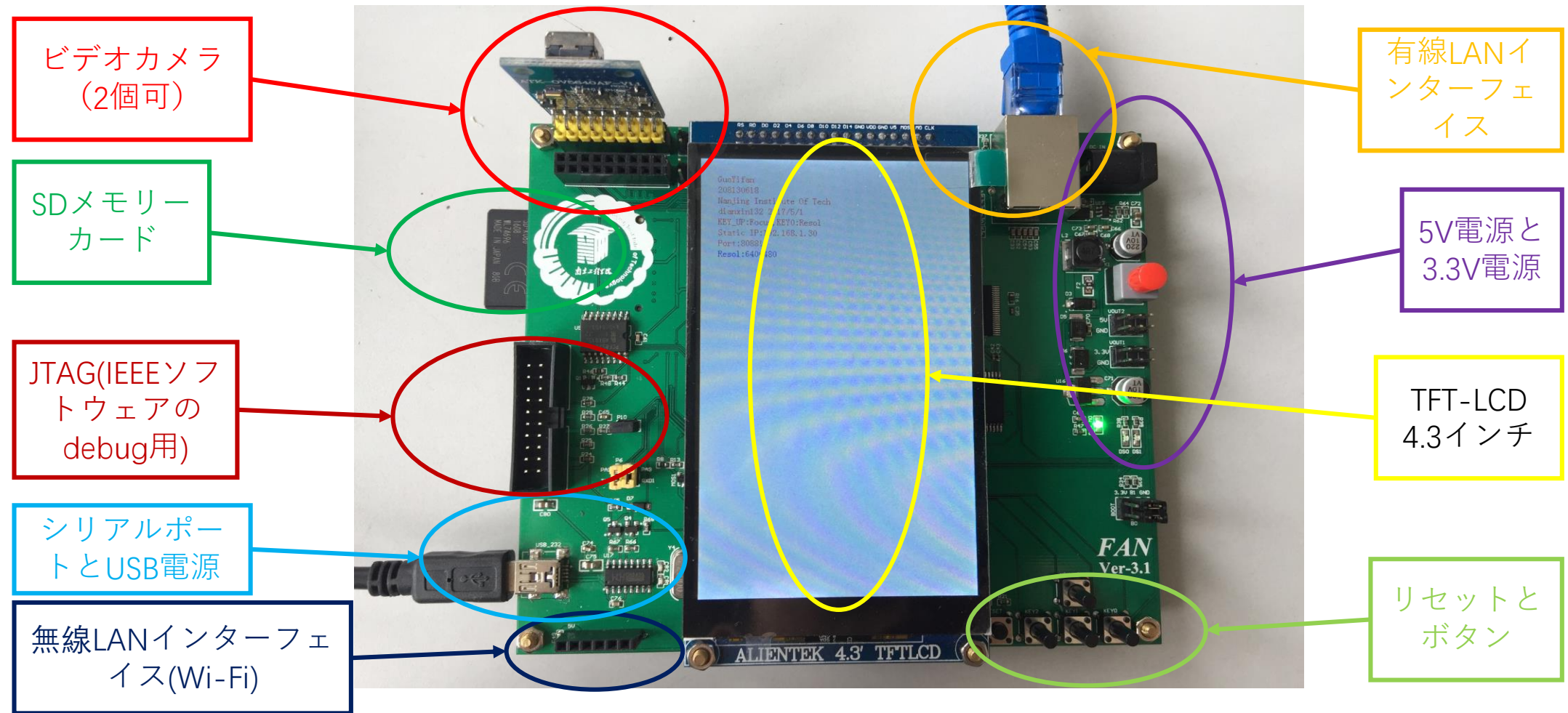


自分で設計したプリント基板
基本的な情報：
長さ：14.0cm
幅さ：12.1cm

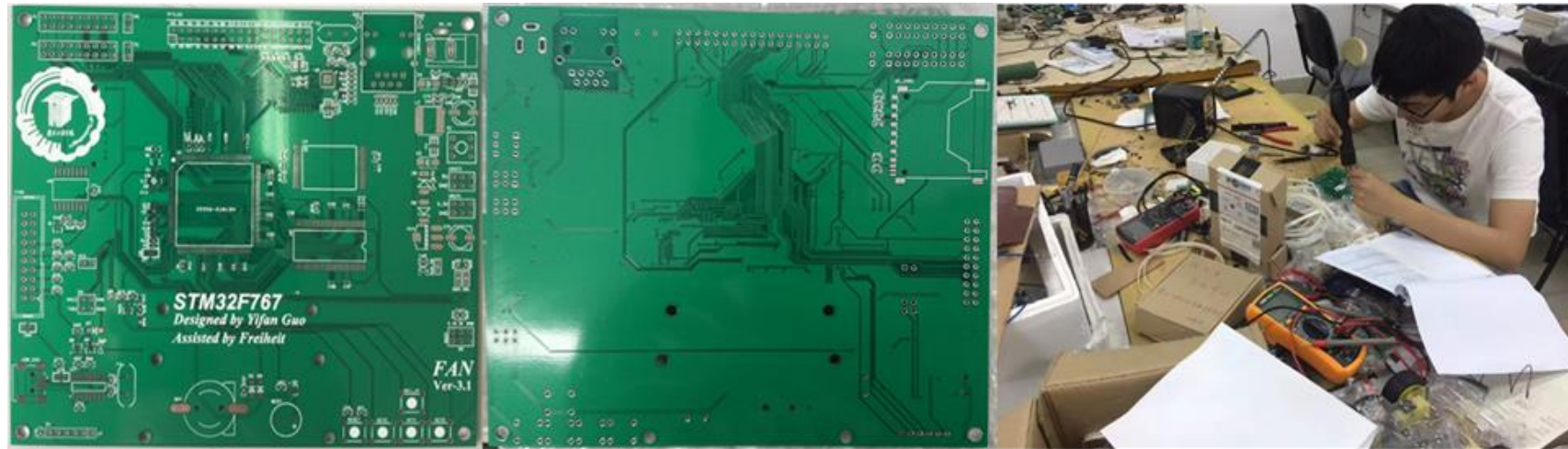
厚さ：1.6mm
層数：2層
費用：8000円ぐらい

注意すべきこと：スケーラビリティ(scalability)が不可欠である。

ハードウェアの設計~ホストコンピュータの構成

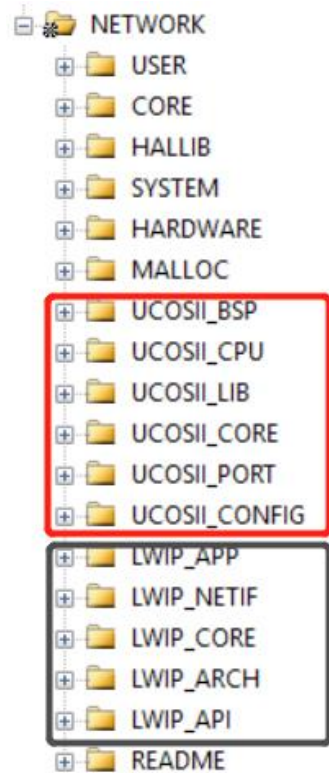


記念写真



溶接用の道具：
ホットエアガン、
はんだ

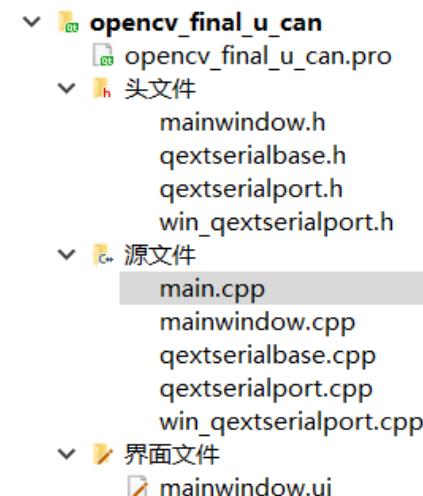
ハードウェア上のソフトウェアの設計~ホストコンピュータ



- 赤いところ：回路基板にリアルタイムオペレーティングシステム(RTOS)をインストールする。
 - μ C/OS IIはマクロコントロールOS(Micro-Controller OS)
 - リアルタイム
 - C言語で書いて,最小限のカーネルは2KBにコンパイルできる
 - MMU(メモリ管理ユニット)がないので、STM32はLinuxを搭載できない
- 黒いところ：回路基板にlwIP (lightweight IP)をインストールする
 - フルスケール(full scale)のTCPにおいてもリソースの使用を削減する
 - 今回オンボードイーサネットチップ(Onboard Ethernet chip) 最大100Mbpsをスポットできる
- 注意すること：タスクの優先順位

ソフトウェアの設計~マスターコンピューター

- QtのMainWindow classに基づいた
 - Qtはプログラム開発フレームワーク (右側の写真)
- ホストコンピュータ(host computer)との通信
 - ビデオカメラは自動的にJPEG画像を出す
 - ホストコンピュータから毎秒30フレーム(30fps)に画像転送
 - 画像の解像度は640*480(変更可能)
 - Qtのソケット(socket)は一度に65536バイトしか受信できません
- 最も重要なのはJPEGデータストリームを抽出する
 - JPEGファイルのマークコード(0xff,0xd8,……,0xff,0xd9)



最初のパッケージ

……,ff,d8,……

複数の中間パッケージ

マーク/テールなし

最終のパッケージ

……,ff,d9,……

ユーザインタフェース(UI)

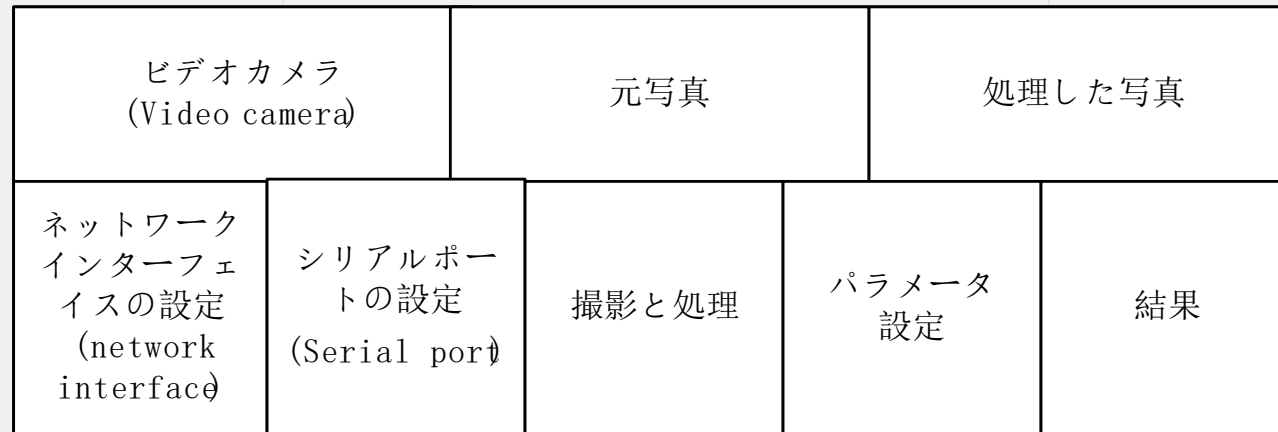
組み込みシステムを用いた産業用画像検出システムの設計 (日本語版) @by Fan

file edit about

ビデオカメラ

写真

処理結果



ネットワークポート

IP: 2.168.1.30 ポート番号: 8088 つながる

シリアルポート

番号: ポート開く
速度(bps): 115200 焦点合わせ
データビット: 8 ブザー開く
パリティ: 無し
ストップビット 1 スピードライト開く

処理

写真を撮る 元画像を保存する
検出 結果を保存する

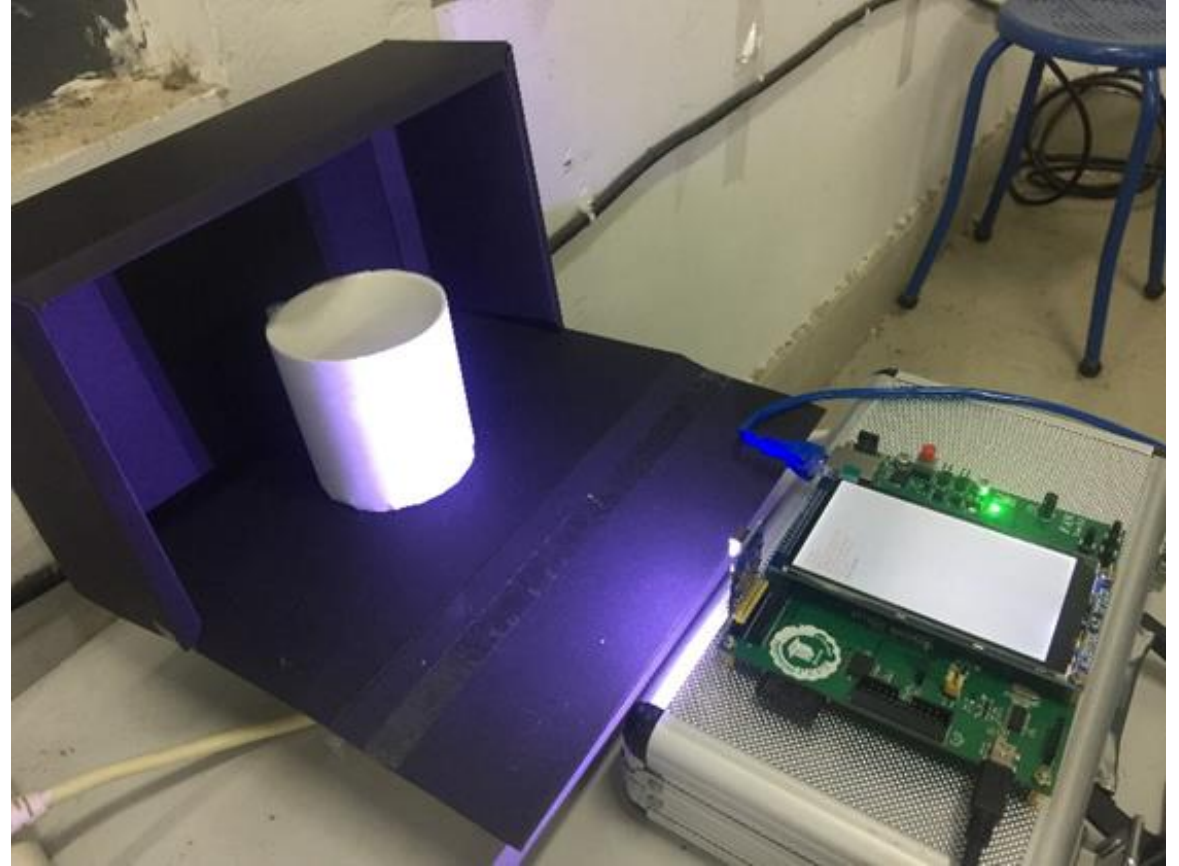
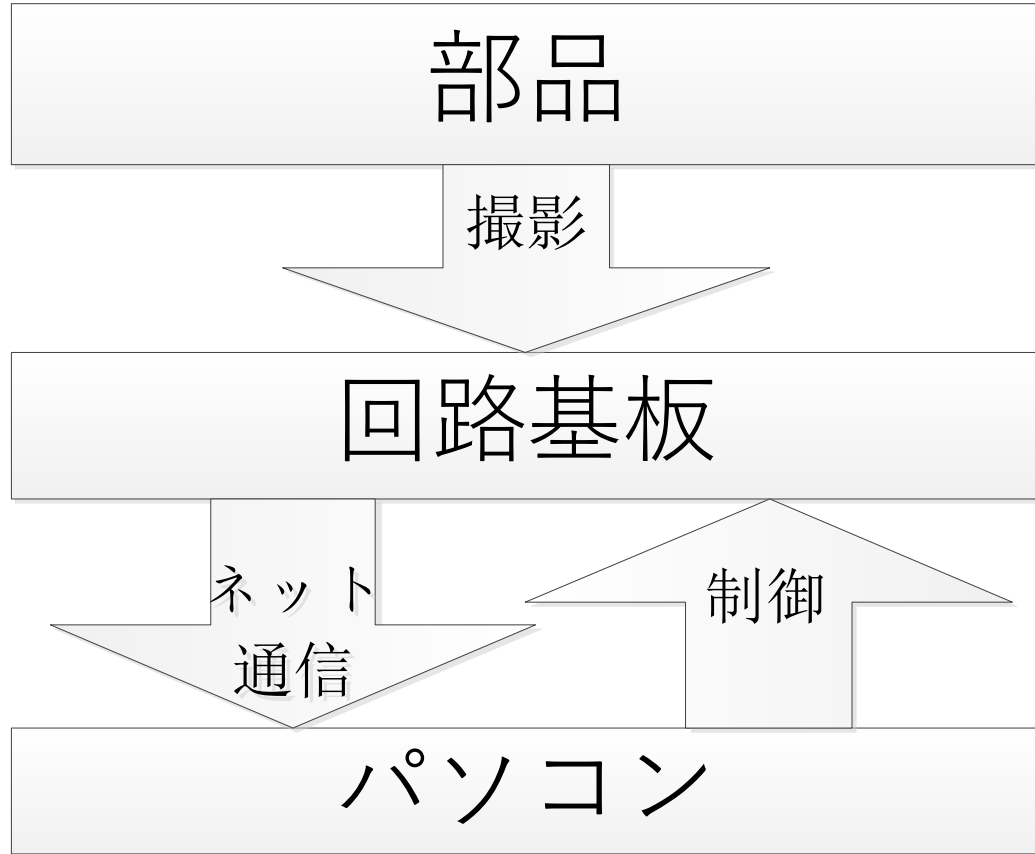
パラメータ

しきい値: 18
カップの実際の輪郭: 26.8 cm
偏り: 0
傷の最小値: 0 cm

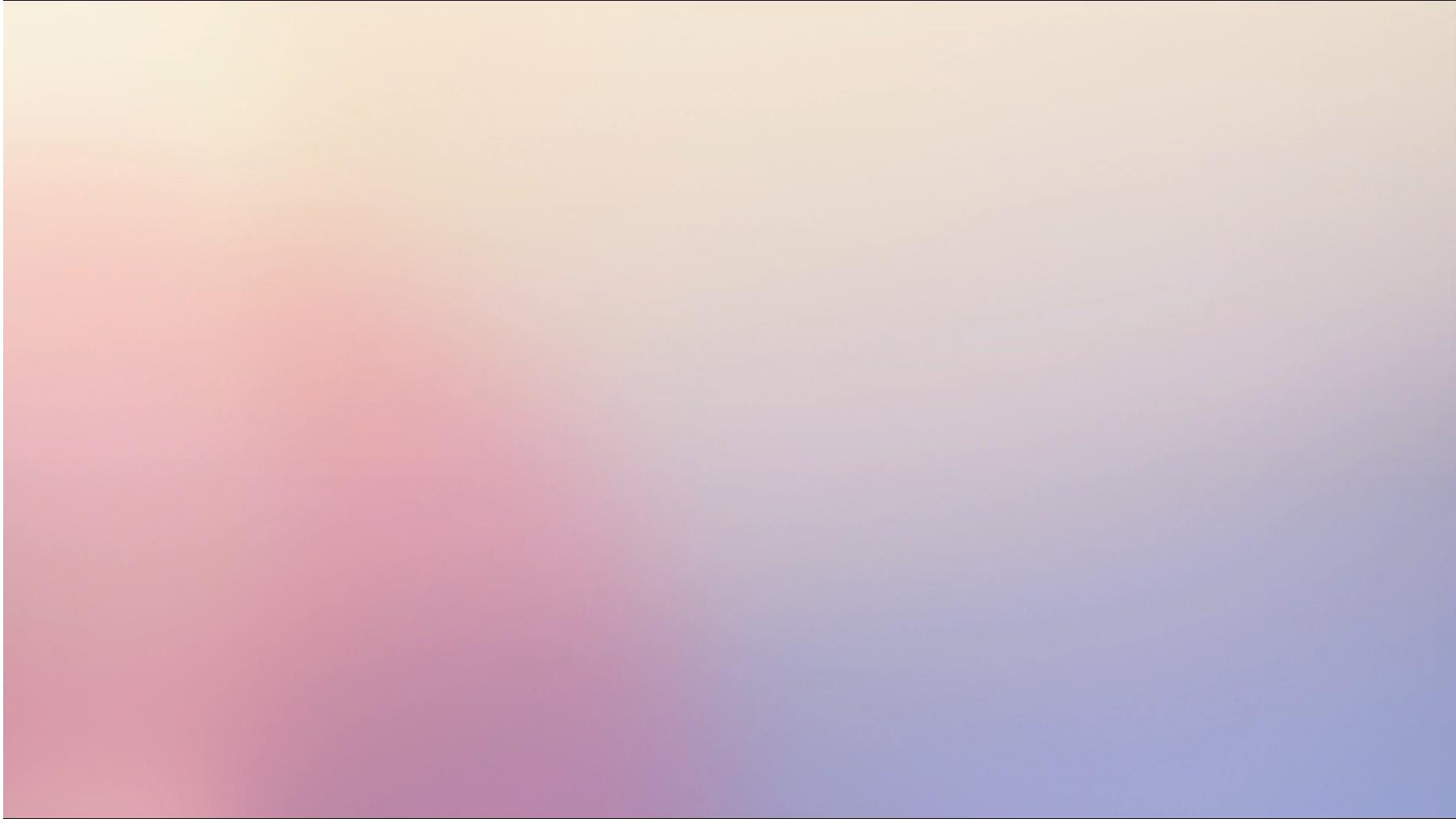
結果

解像度:
傷の長さ:

システムのテスト



テスト



画像処理について

- 二値化、ノイズを低減するための3x3コア（サイズは小さい場合もっと小さい変化を計測できる）
- エッジ検出
 - OpenCVのCannyエッジ検出演算子
 - 勾配（gradient）を利用して、グレースケールの強度が画像内で最も変化する位置を見つける
 - 各輪郭の長さを検出することができる
- 傷の長さを計算する公式
 - $$\frac{\text{実際の部品の周囲長 } P1}{\text{実際の傷の長さ } L1} = \frac{\text{画像の部品の周囲長 } P2}{\text{画像の傷の長さ } L2}$$
 - P2とL2は画像から利用可能であり、P1は（標準化された部品であるため）事前に測定する必要があり、L1は取得したい最終的なものである。

パラメータの説明

パラメータ

しきい値: 18

カップの実際な輪郭: cm

偏り:

傷の最小値: cm

公式：

$$\frac{\text{実際の部品の周囲長 } P1}{\text{実際の傷の長さ } L1} = \frac{\text{画像の部品の周囲長 } P2}{\text{画像の傷の長さ } L2}$$

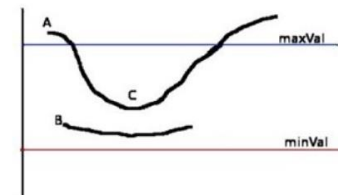
```
Canny( grayImage, g_cannyMat_output, edgeValue, edgeValue*3, 3 );
```

閾値：ピクセル(pixel)位置の振幅が低閾値よりも小さい場合、ピクセルは除外される。edgeValueは低しきい値、edgeValue*3は高しきい値。

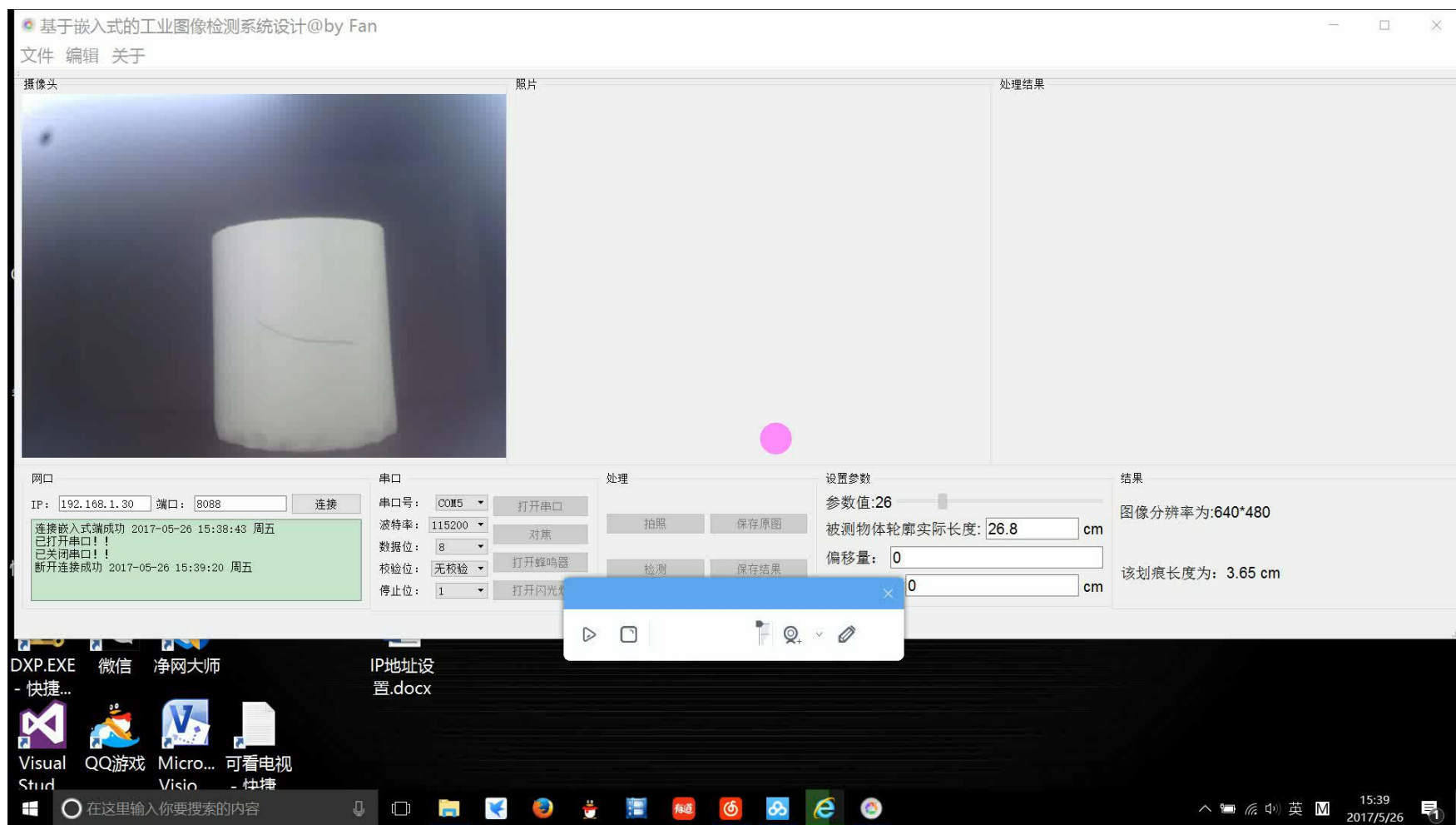
カップの実際な輪郭：公式の中で実際の部品の周囲長 $P1$

偏り：誤差を減らすために使用されること

傷の最小値：実際の傷がこの最小値より小さい場合は、傷が無視される。



もし、カップの実際の輪郭は33.3cm,実際の傷の長さは4.00cmの場合、システム全体のエラーをどのように調整するのか？



まとめ（各開発環境）

	ハードウェア	ホストコンピュータ	マスターコンピュータ	OpenCV テスト
IDE	Altium Designer 	Keil 	Qt 	Visual Studio 
言語	/	C	C++	C++/Python
難しい 部分	高周波回路における非等長導体による信号遅延	OSにおけるタスクの優先順位の割り当て	JPEG形式の画像の分割と受信	画像処理に慣れていない
解決策	蛇行経路 	ネットワーク送信 タスクの優先度を上げる	<div>最初のパッケージ</div> <div>.....ff,d8,.....</div> <div>複数の中間パッケージ</div> <div>マーク/テールなし</div> <div>最終のパッケージ</div> <div>.....ff,d9,.....</div>	公式発表 （制限あり） $\frac{\text{実際の部品の周囲長 } P1}{\text{実際の傷の長さ } L1} = \frac{\text{画像の部品の周囲長 } P2}{\text{画像の傷の長さ } L2}$

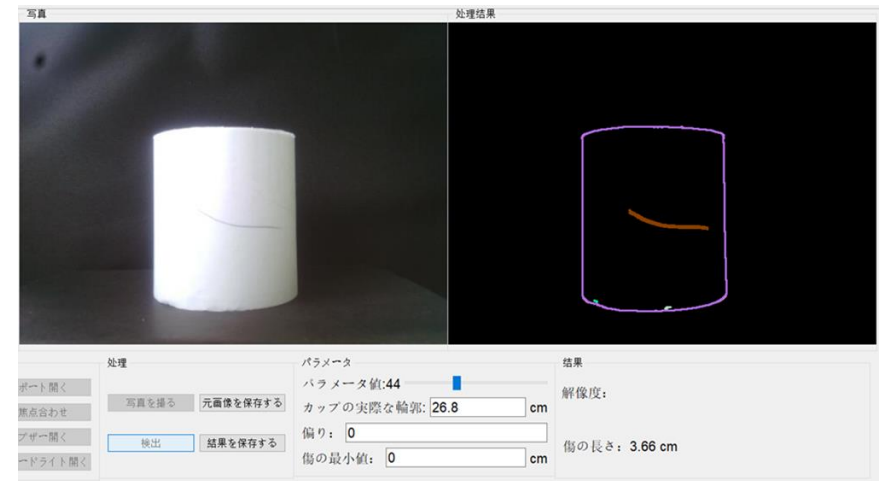
まとめ(欠点と未完成の部分)

- 欠点

- 手動操作が多すぎ、スマートではない
- 色々な制限がある（黒い背景）
- 傷の深さが計測できない
- 複数の傷が計測できない

- 未完成の部分

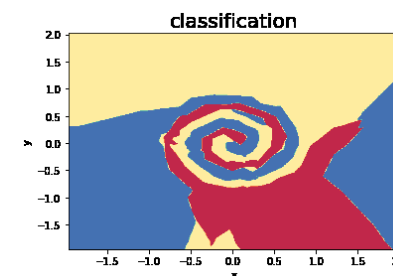
- マルチノード(Multi-node)
- 無線(Wireless)
- 機械式アーム(Mechanical Arm)のインストール



- 完全にそれを応用したければ、より多くの時間とお金がかかる。

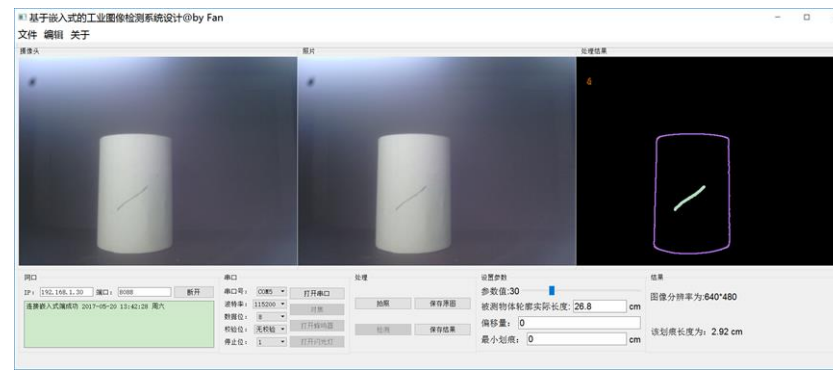
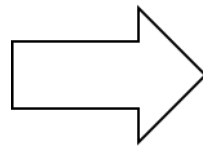
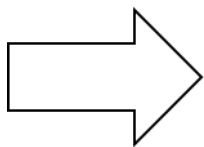
最近やったこと

- NumPy library と数学で作った N 層のニューラルネットワーク
 - 例えば、5 層はReLU,出力層はsigmoid,三分類のとき
- ```
#NN layers
layers_dims = [Xtrain.shape[1],10,10,10,10,3]
```
- 2700 training data, 300 test data, 10000 epochs
- テスト正答率98.33%
- PyTorchを勉強する
- MNIST dataset にCNNを訓練した(正答率98.46 %, Batch size 100, 学習率0.001, epoch 2回)、自分で作った数字の画像に入れて、正答率は90%ぐらい。
- 修士になるために専門知識、日本語と作文を復習する。





質問



ご覧いただきありがとうございます。