

Android はチヨロQを遠隔操作するか

Yutaka Yasuda, Kyoto Sangyo Univeristy



EMMA Mobile 1

- NEC Electronics のマルチメディアプロセッサ

ARM1176 500MHz + DSP + Graphics Accelerator

512Mbit DDR SDRAM を内蔵

SD, NTSC/PAL, LCD, PCM音源, Camera 等 IF 豊富

H.264 engine もある

現在は Renesas から

see: http://www2.renesas.com/mobile/ja/emma_mobile/index.html

EMMA Mobile-1D HDK

- Hardware Development Kit
with LCD, 2GB Flash, etc.
- Android SDK 1.6 (Donut)



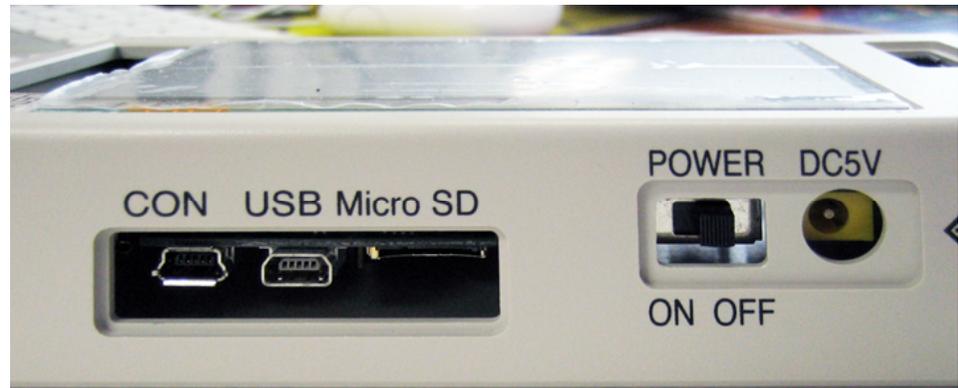
現在は Renesas から

see: http://www2.renesas.com/mobile/ja/emma_mobile/hdk_sdk/index.html

EMMA Mobile-1D HDK



12 keys + 4 LEDs



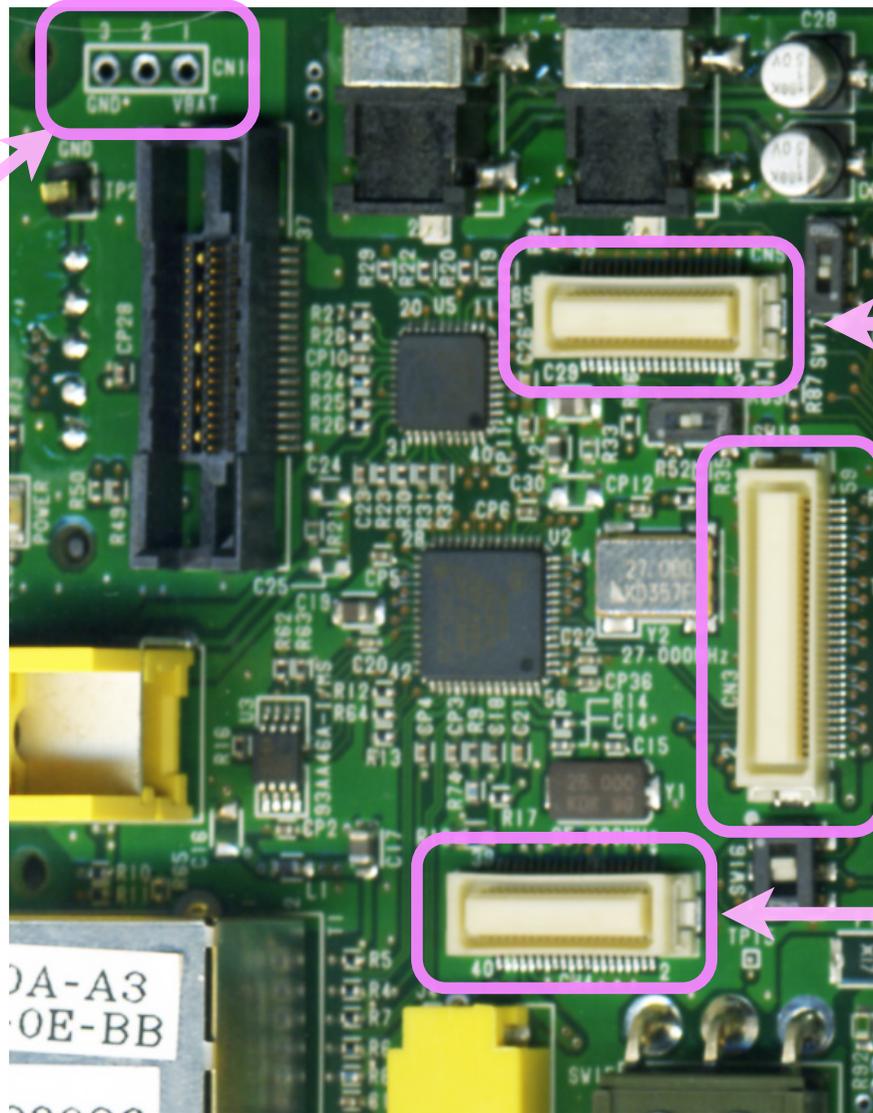
console + USB + MicroSD

いろいろ手が入られそう・・・

他にも Ether, TV-out, MIC in, LINE out, LED x2, push button x4, SD IF

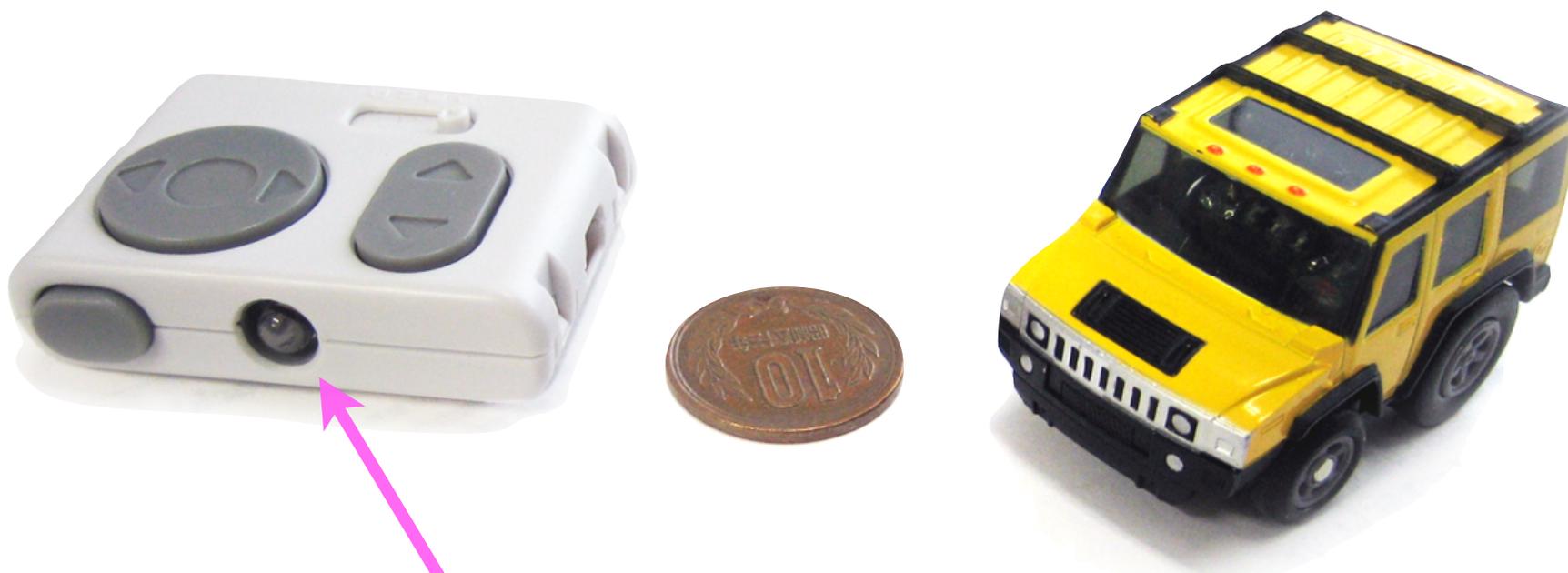
他にもいろいろ...

ピン無し
ホール



数々の
コネクタ

赤外線リモコンチョロQ



赤外線LED

チヨロQ側インタフェイス検討



ターボボタン
は今回使わず

赤外線LED

ボタンは 4 つ
前後
左右
の二系統

バラしてボタンを押し
たことにする回路をつ
ければ良からう

さて EM1 側は？

EM1側インタフェース検討

- 半田付け

ヤメてくださいと

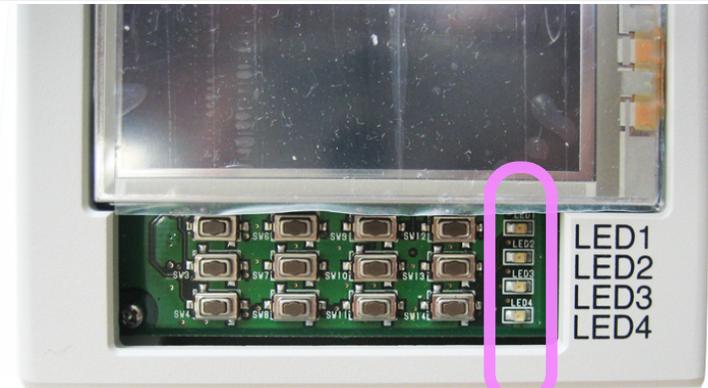
- コネクタから取り出す

フィルムケーブルの作業ができるか？

こちらで用意してもその先で壊すかも

- LED があるではないか

光カップリングでやろう

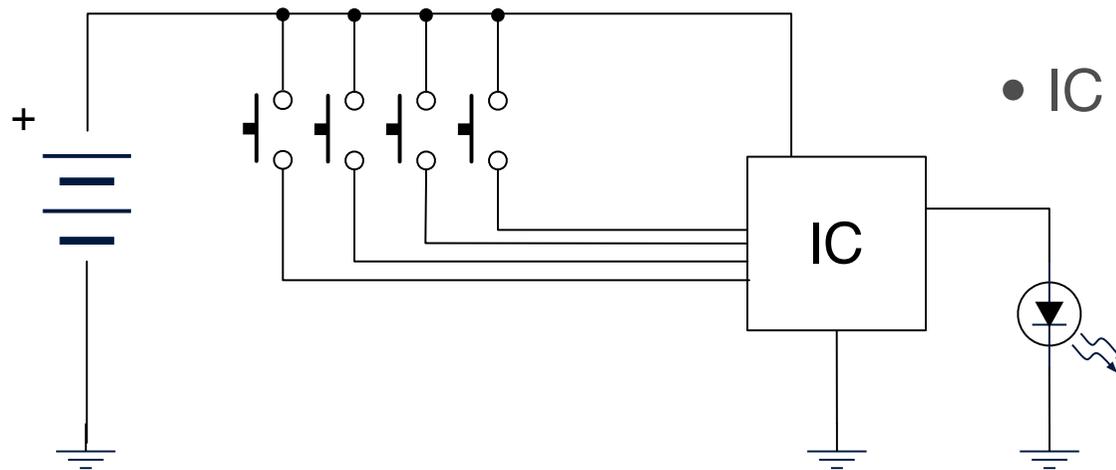


楽しい電子工作

step 1

どのくらいの抵抗値まで switch on
と認識するだろう？

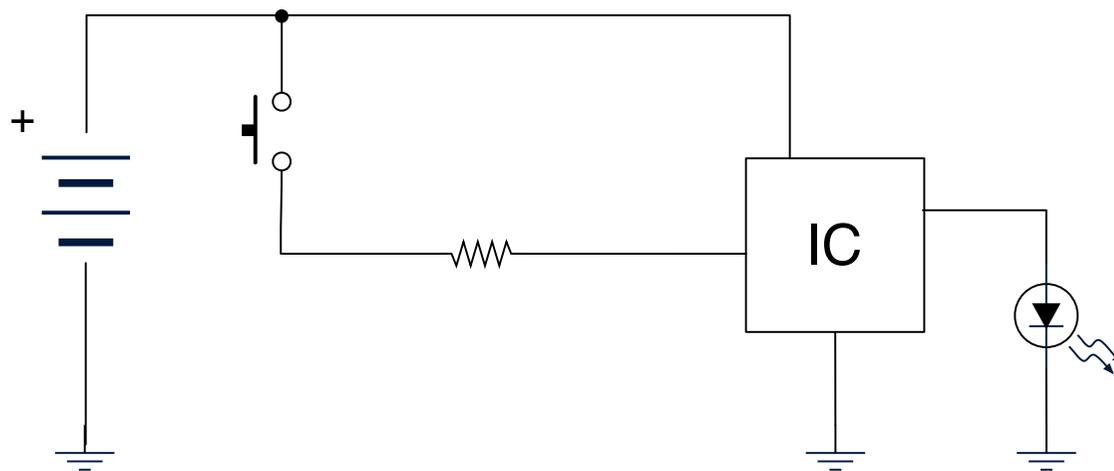
- 電源3V (LR44 x 2)
- 出力は LED ひとつ
- 入力はスイッチ 4つ
- 一方は電源直結
- 反対側は IC に消える
- IC は樹脂封止で詳細不明



step 2

では半導体でも導通と認識させられるであろう

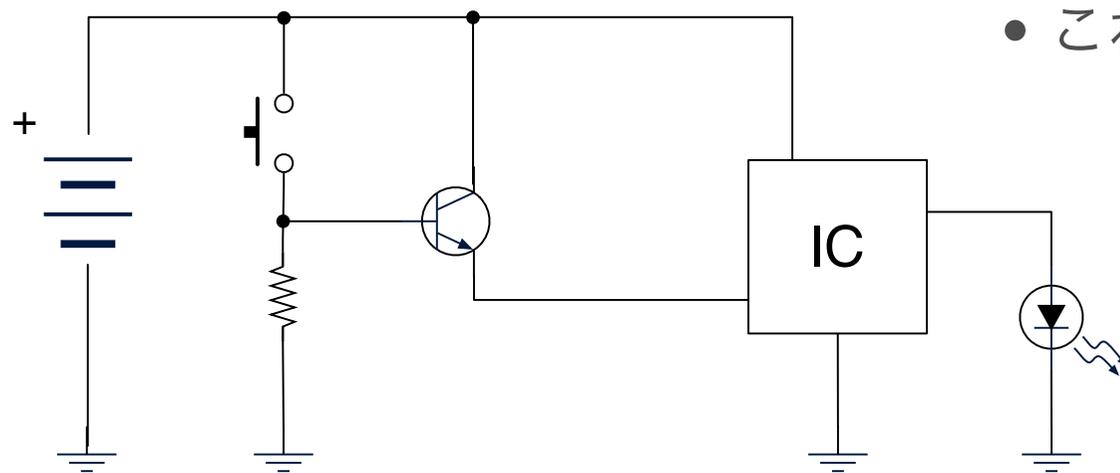
- 適当な抵抗を入れて実験
- $500\text{K}\Omega$ では絶縁と認識
- $200\text{K}\Omega$ では通電と認識



step 3

ではフォトトランジスタ直結でも動作するのではないか

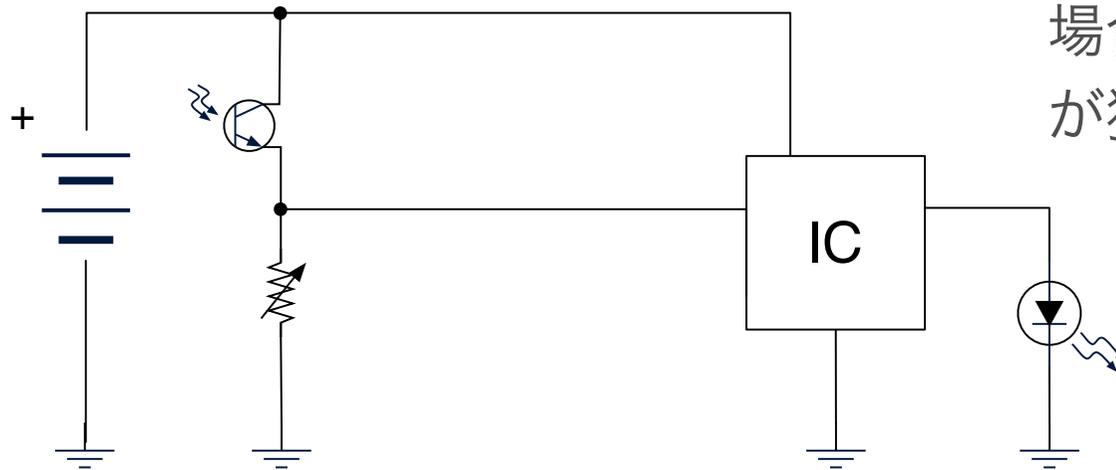
- 適当なトランジスタを挟む
- IC が負荷抵抗になるだろう（焼き切れないと推定）
- トランジスタ出力は直結で良い
- これで正しく認識



step 4

出力を上げて調整範囲を上げよう

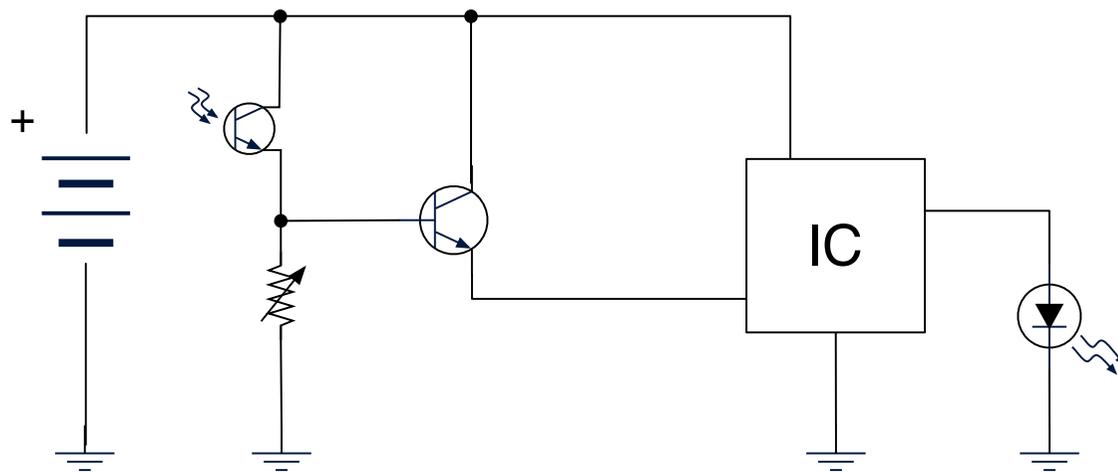
- スイッチとトランジスタをフォトトランジスタに置換
- 感度調整用に半固定抵抗
- 一応これで反応するが、
- 外光の影響などで動かない場合がある（調整可能範囲が狭い）



step 5

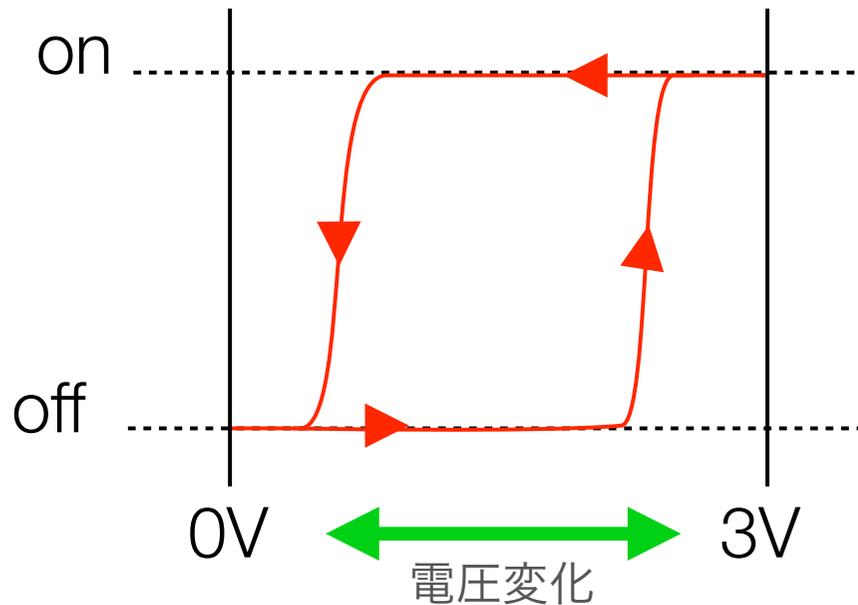
むむむそんな罫が・・・

- ましにはなるが、
- off → on はできても
on → off できなかったり
- その逆とか
- ヒステリシス特性がある



ヒステリシス特性

フォトトランジスタ由来の出力が、この上限・下限を往復するようにアナログ回路で調整するのは面倒すぎる・・・



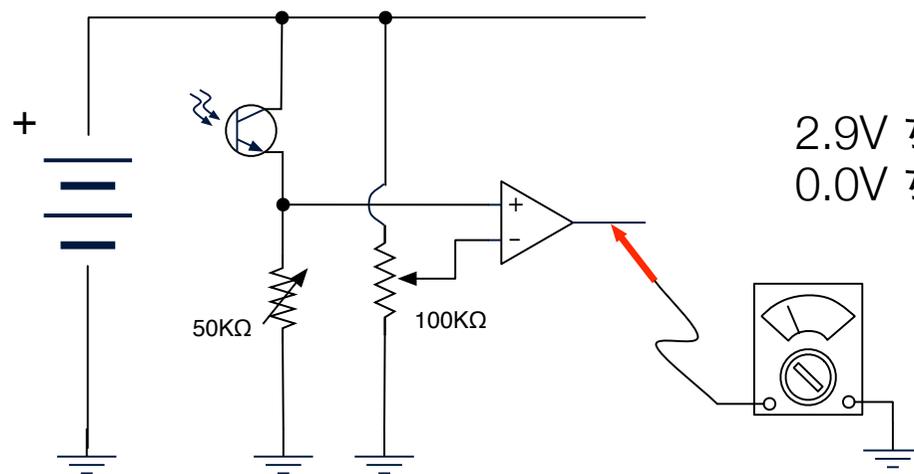
チャタリング対策としては普通（元はスイッチ直結だし）

- 非常に電源電圧に近いところまで上がらないと on になったと認識しない
- 非常にグラウンドに近づかないと off と認識しない
- 明らかな変化があるまでは、以前の状態を保持する
- しかし振れ幅がデカ過ぎ（0.2V-2.8V くらい？）

step 6

しかし・・・

- コンパレータ万歳！
- 3V 単一電源
- フルスイング
- これできっちり往復



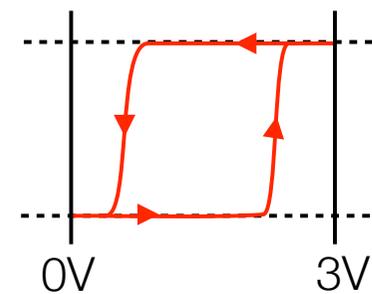
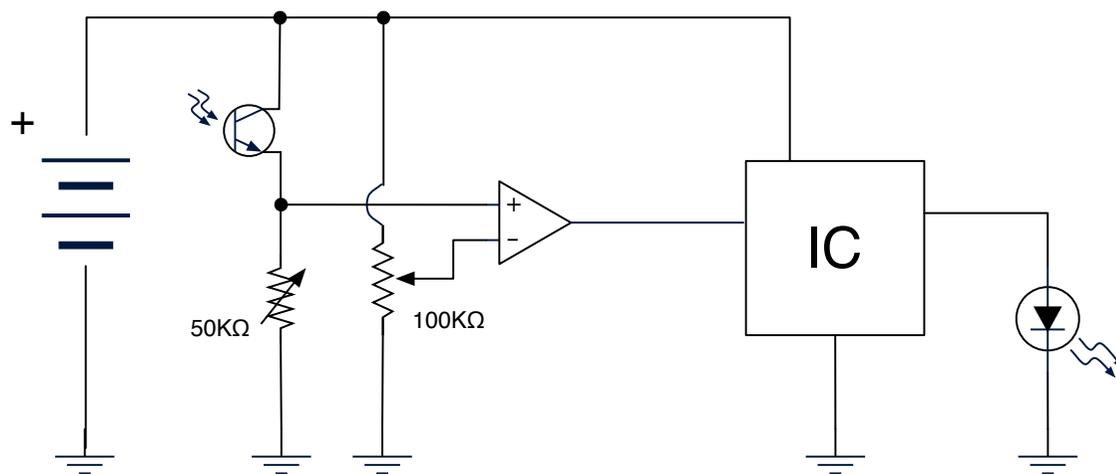
2.9V ちよいと
0.0V ちよいをきっちり往復

step 6

出力を上げて調整範囲を上げよう

- IC につなぐと出力の電圧が降下
(1V 程度に)

するとヒステリシスの罠が...



データシートを読もう (いまさら)



NJU7014/15/16

低電圧動作デュアル C-MOS オペアンプ

■ 概要

NJU7014/15/16 は、低電圧動作 2 回路入りの C-MOS オペアンプです。

バイアス電流は 1pA と低く、グランド電位近辺の微小信号を増幅することができます。

また、動作電圧は 1V (min) と低電圧駆動が可能で、出力は電源電圧範囲内でフルスイングが可能です。

さらに、小型パッケージのラインアップが充実されており、ポータブル機器やバッテリー駆動機器に幅広く応用することができます。

■ 外形



NJU701XD



NJU701XM



NJU701XV



NJU701XR

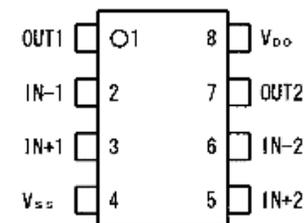


NJU701XRB1

■ 特徴

- 単電源動作
- 動作電源電圧 ($V_{DD}=1\sim 5.5V$)
- 高出力電圧振幅 ($V_{OL}=2.9V \text{ min @ } V_{DD}=3.0V$)
- 低消費電流
- 低入力バイアス電流 ($I_{IB}=1pA \text{ typ}$)
- 位相補償回路内蔵
- C-MOS 構造
- 外形 D1P8, DMP8, SSOP8, VSP8, TVSP8

■ 端子配列

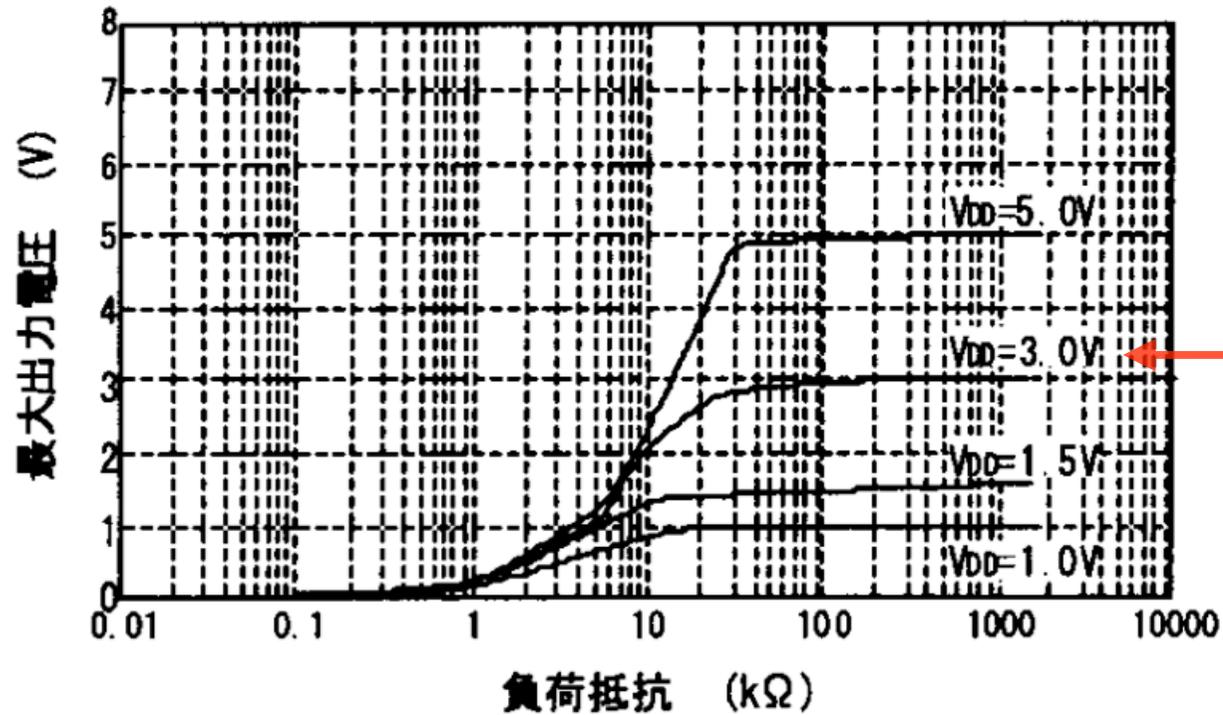


■ 製品構成 (Ta=25°C, V_{DD}=3.0V, 1 回路あたり)

項目	NJU7014	NJU7015	NJU7016	単位
消費電流	15	80	200	μA (typ)
スルーレート	0.1	1.0	2.4	V/μs (typ)
利得帯域幅	0.2	1.0	1.0	MHz (typ)

NJU7016

最大出力電圧 - 負荷抵抗特性



今回の電源は 3V

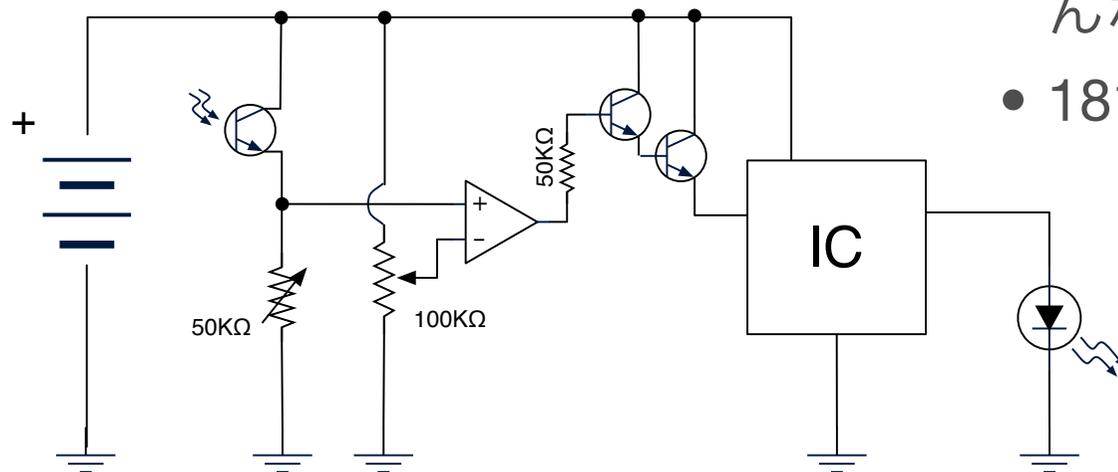
負荷抵抗は 50kΩ 程度が限界か
最初の予備実験では 400kΩ 程度で導通と認識されたが、、、
IC はブラックボックスなので、どのくらい引っ張るかが判らない

負荷抵抗で調整せず、出力を上げてしまえ

step 7

動作した、が、テスターが抜けない

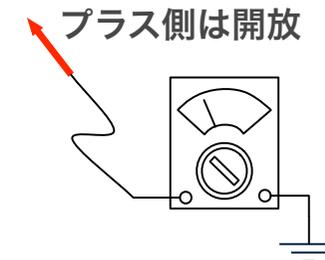
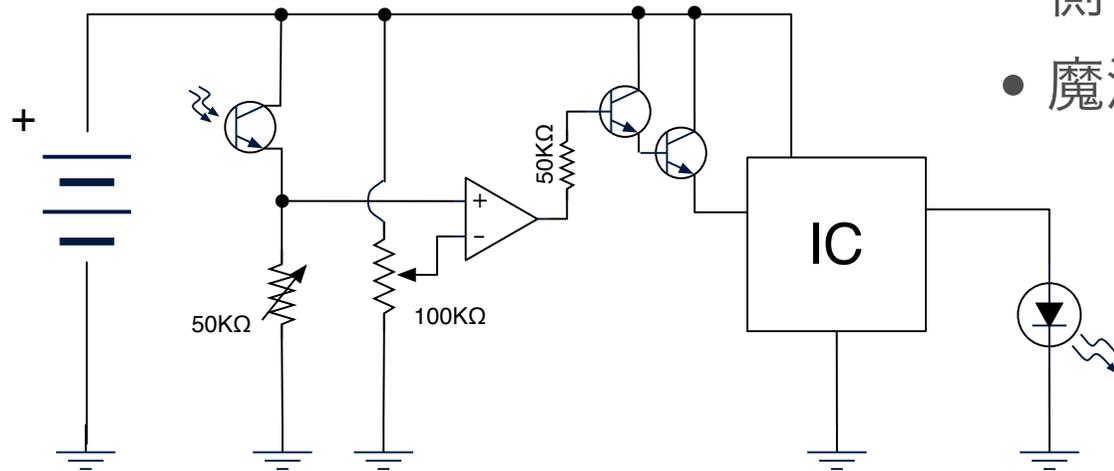
- $50\text{K}\Omega$ の出力負荷を用意
- $60\mu\text{A}$ では勿論足りない
- 2SC1815 (増幅率 100) 一段で 6mA (実測 2mA)
- 若干足らずもう一段
- 最大 200mA 、かつ IC はそんなに引かない筈
- 1815の定格範囲に収まる



step 8

電線の LC 成分の何かが効いている

- プラス側プローブは接続せず
- マイナス側プローブが接続されておれば安定動作
- 外すとダメ
- テスターを外してマイナス側プローブだけで同じ挙動
- 魔法の電線？

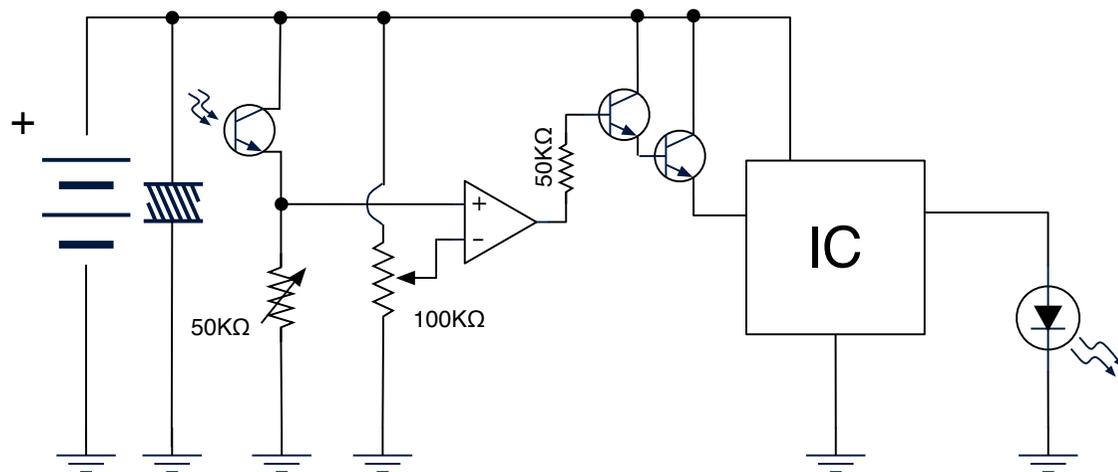


マイナス側を接続するだけで安定

step 9

だからアナログは嫌い

- 手近な基板から部品取り
- 0.1 μ F パスコンではダメ
- 大きめの電解コンデンサを電源側につけて解決
- これで今のところ安定



と言うかね・・・

データシートに書いてあるし

		32U (1VSP-8)	
動作温度範囲	T_{opr}	-40~+85	
保存温度範囲	T_{stg}	-55~+125	

(注1) 入力電圧は、 V_{DD} または7(V)より小さい方の値を超えて印加しないで下さい。

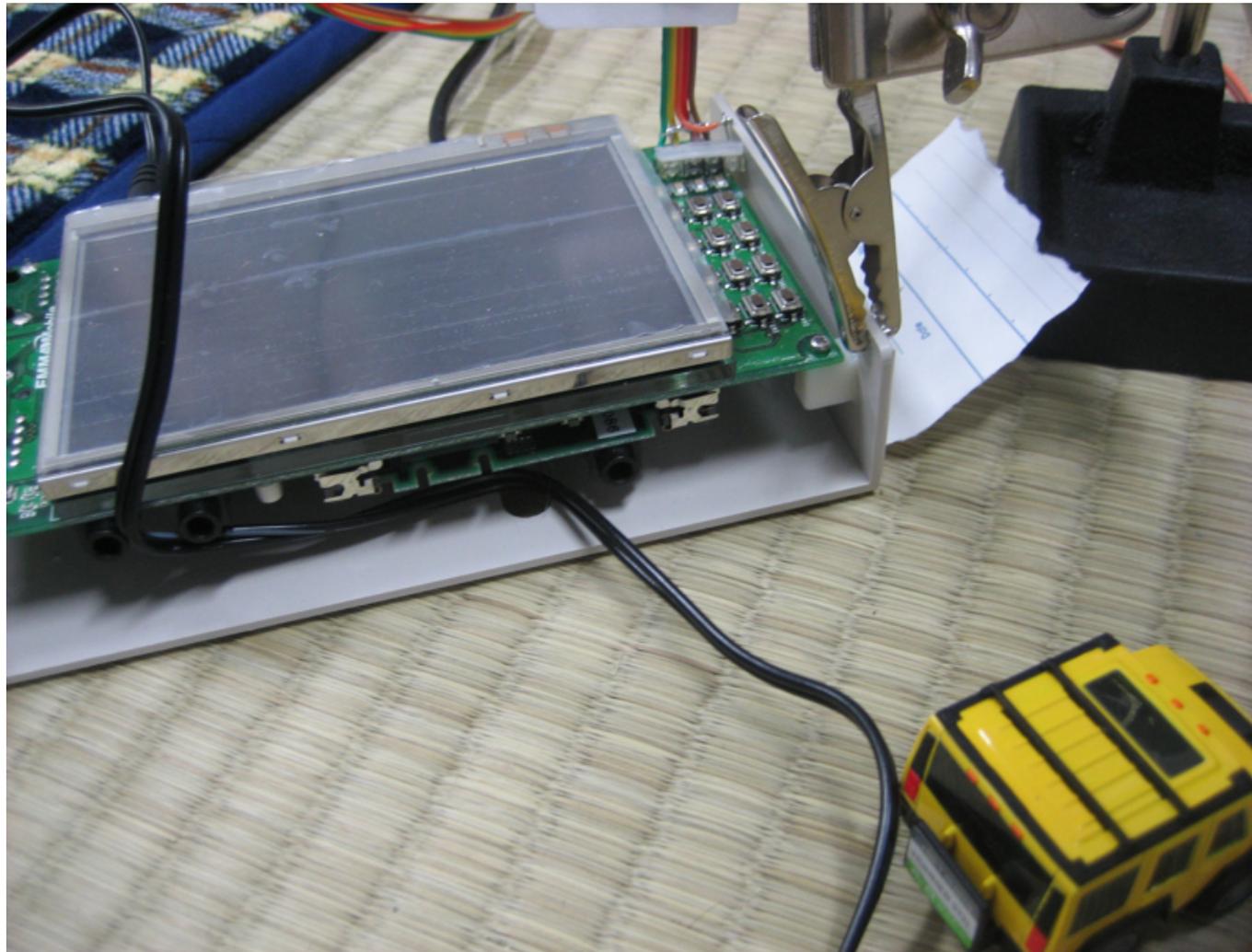
(注2) ICを安定して動作させるために、 V_{DD} - V_{SS} 間にデカップリングコンデンサを挿入して下さい。

■ 電気的特性 ($T_a=25^\circ\text{C}$, $V_{DD}=3.0\text{V}$, $R_L=\infty$)

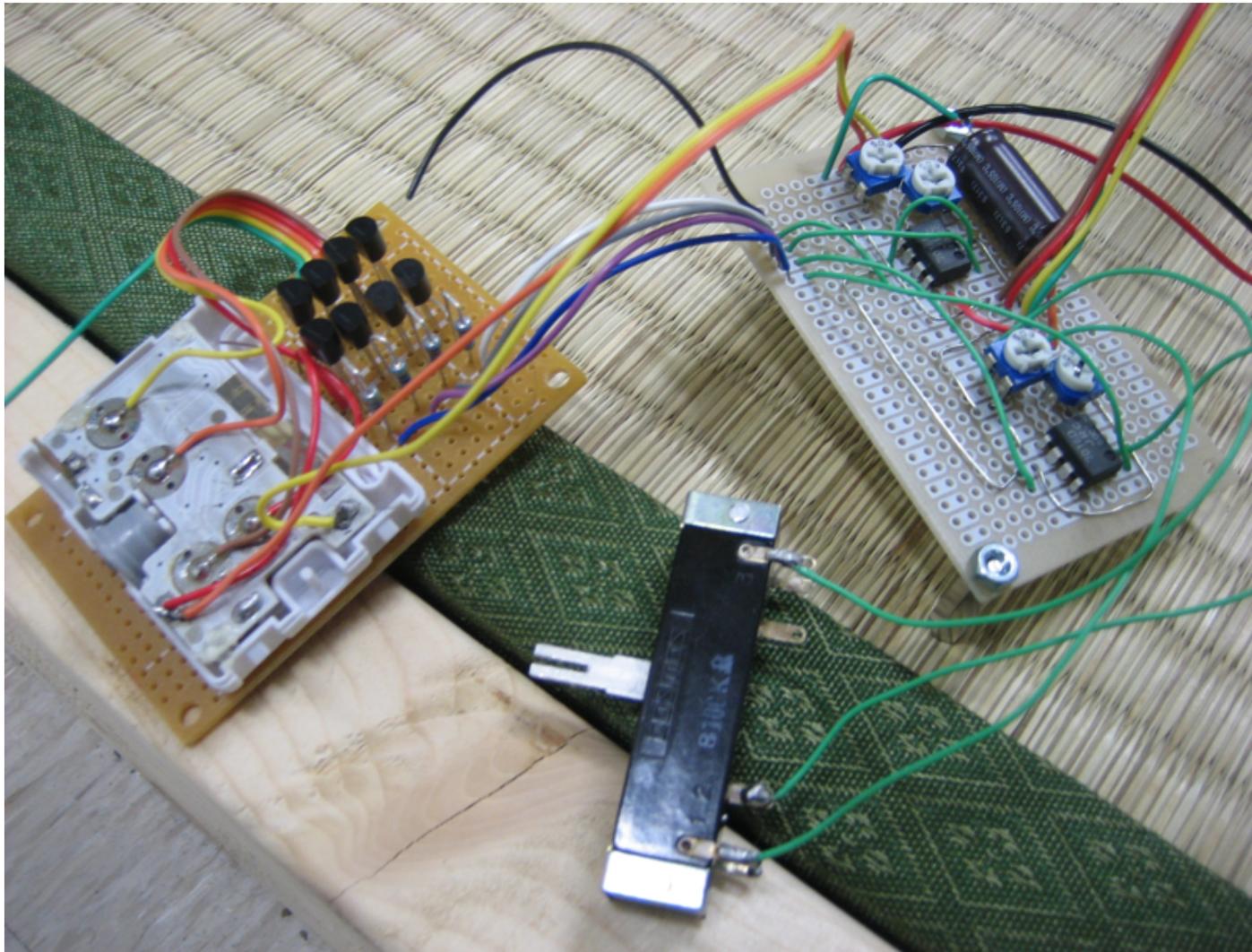
それ以前に常識です と、言われそう・・・

外觀

カップリング部



左：ドライバ 右：判定回路



ソフトウェア

Android SDK

- ホスト PC 側に Ubuntu 8 をインストール
- developer.android.com から SDK 1.6 を取得
- EM1 用のドライバソース等を組み込んで target build
- 生成された圧縮イメージを MicroSD にコピー
- EM1 の boot mode スイッチを変更して起動、MicroSD のイメージを本体側 Flash に複写
- USB/シリアル接続のコンソールを用意
- ROM monitor の起動パラメタを Wind River Linux から Android 用に書き替える

LED クラス

- LEDクラスドライバによるインタフェイス有り
- `/sys/class/leds/LED_NAME/brightness`
 - 消灯：0
 - 点灯：1（0以外）
- 単に `echo 1 > /sys/class/leds/led1/brightness` でも可
- パーミッションの問題あり
(`others writable` でない：後述)

実走風景

動画の編集

動画の編集

アノテーション

AudioSwap

キャプションと字幕

インサイト

EM1 でチョコQをドライブする(1)

yutakayasuda

6 件の動画

チャンネル登録



Video player controls including a play/pause button, a volume icon, a progress bar showing 0:04 / 0:09, a resolution indicator for 360p, and icons for full screen, share, and settings.

動画の編集

動画の編集

アノテーション

AudioSwap

キャプションと字幕

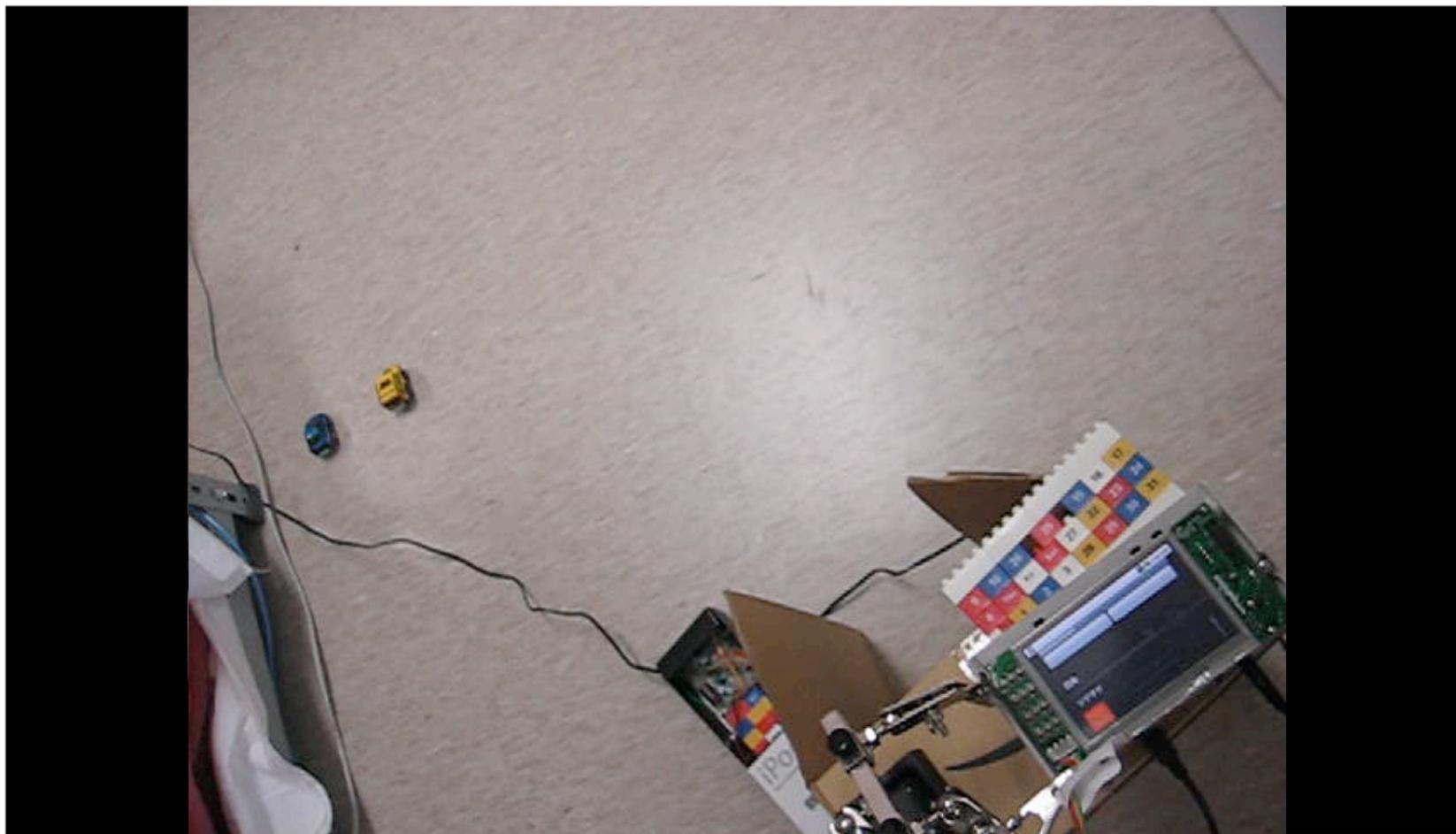
インサイト

EM1 でチョコQをドライブする(2)

yutakayasuda

6 件の動画

チャンネル登録



0:03 / 0:10 360p

yutakayasuda — 2010年06月16日 — NEC アンドロイド開発機 EMMA Mobile 1 でチョコQをドラ

15

まとめ： ちよろQドライブ開発は何だったのか

擬似組み込み開発体験である

- 本物の組み込みでは恐らく手が出ない

Linux ベースなのでトラブルに対処できる
(システムの動きを追える)

- それでも一通りのことはする

本体ハードの仕様、チップの型番を調べる

ブロック図を見る、スイッチ・ピンの機能を見る

経験の浅い情報系学生にとって難しすぎない

非ハード系の教員にとっても負担が軽い

ex. USB serial console

1. コンソールはUSB serial (機体とマニュアルを見る)
2. チップは FTDI の FT232R (ブロック図を見る)
3. Linux 用ドライバがある (FT232のデータシートを見る)
4. Linux kernel 2.6 以降はメインツリーにマージされたと知る (Google)
5. 自ホストが 2.6 ベースだったので差してみると lsmod で usbserial に ftdi_sio ドライバがロードされた (Linux)
6. /dev/ttyUSB* を確認 (Linux)
7. シリアル対応コンソールアプリ minicom を探してインストール (Google)
8. 仕様に従って baud rate などを設定 (マニュアル)
9. コンソール表示が見えた！

組み込み的な作業と上層の作業が適度に被っていて経験の浅い情報系学生に丁度良い

適度に被っているので教員がフォローしやすい

ex. LED クラスのpermission

- LED点灯は /sys/class/leds/LED_NAME/brightness
- しかし other writable でない (ls -l で確認)
- システム初期化スクリプトに入れたいが rc.local は無い
- EM1のSDKを漁って材料探し (find, grep 等)
- EM1 固有部分の初期化スクリプトを発見
(change-091202/vendor/necel/em1 以下の init.mp200.sh)
- 実機上では SDK_WORK/android-fs/system/etc 以下に init.mp200.sh あり
- chmod を追加、システムイメージを再ビルド、インストールして動作確認

Linux 的経験だけでもシステムの見通しが良く自由度が高い

ちよるQドライブ開発は何だったのか

- 情報系学生にとって良い教材である

本物に触れる

ハード：チップレベルからインタフェースまで

ソフト：ドライバからアプリまで（ソース込み）

上下共に天井無し

広範囲な経験が得られる

- 情報系教員にとっても扱いやすい教材である

従来的 Linux オペレーションでかなり追える

本職領域まで踏み込まずに済む

おすすめです

