



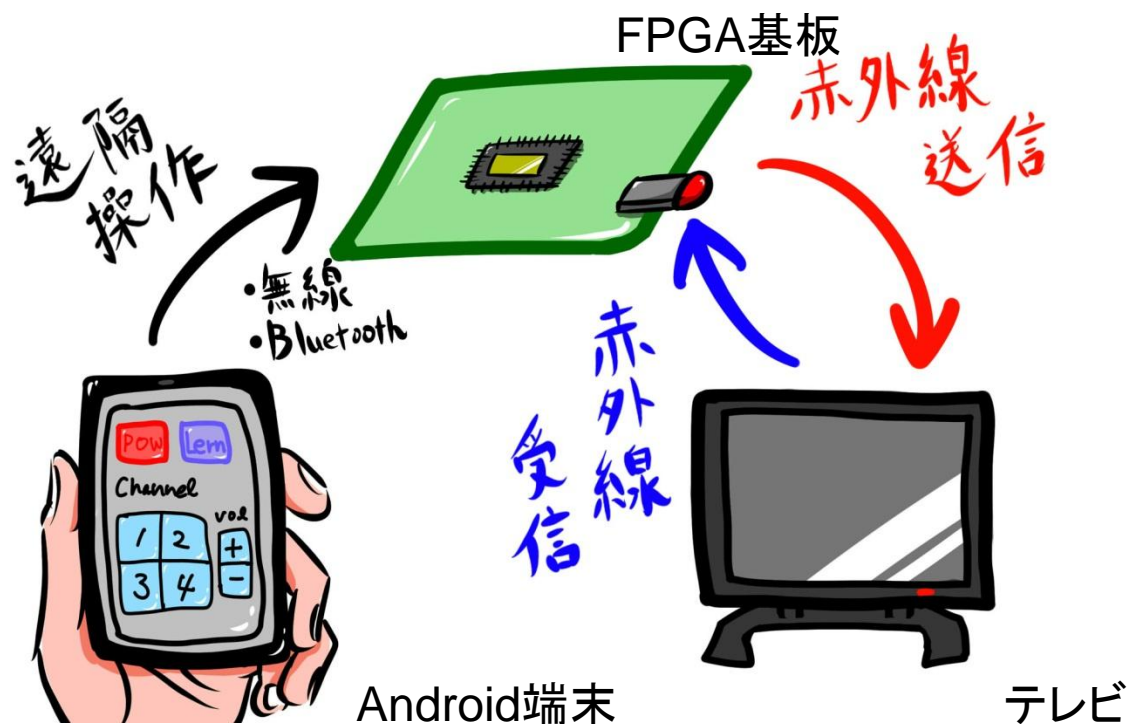
ANDROID × FPGA

赤外線学習リモコン 中間発表

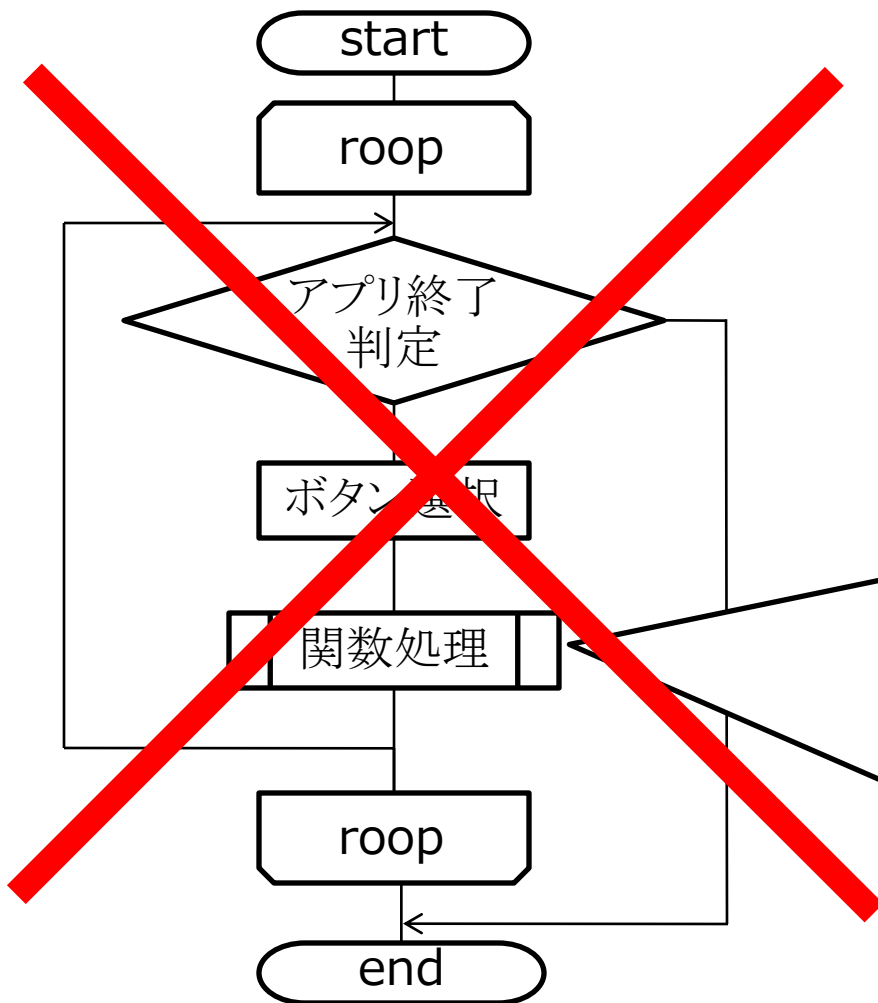
宇都宮大学工学部情報工学科3年 高橋惇

➤ 計画時全体構想

専用リモコンではなく生活と密接に関わるAndroid端末を用いてFPGAを中継しTVを操作する。



➤ 計画時全体構



```
sendPow(int pow){  
    if(pow==1)  
        電源ON;  
    else  
        電源OFF;}
```

```
sendCh(int ch){  
    if(ch==1){  
        ch1の信号送信;}  
    else if(ch==2){  
        ch2の信号送信;}  
    ...}
```

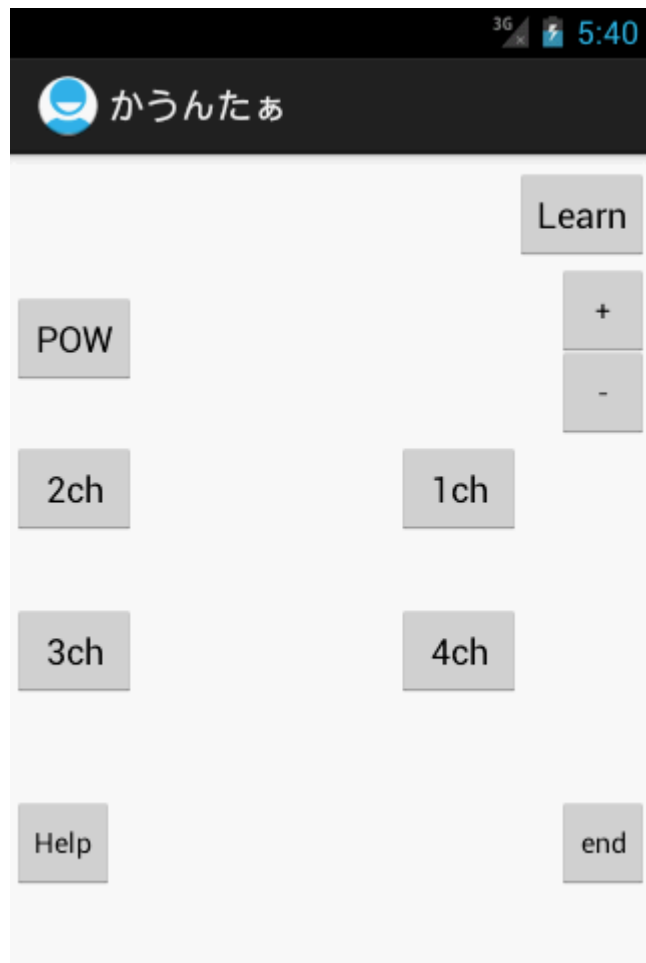
```
sendVol(double v){  
    vの値により音量変更;}
```

```
learnCh(int ch){  
    if(ch==1){  
        ch1の設定;}  
    else if(ch==2){  
        ch2の設定;}  
    ...}
```

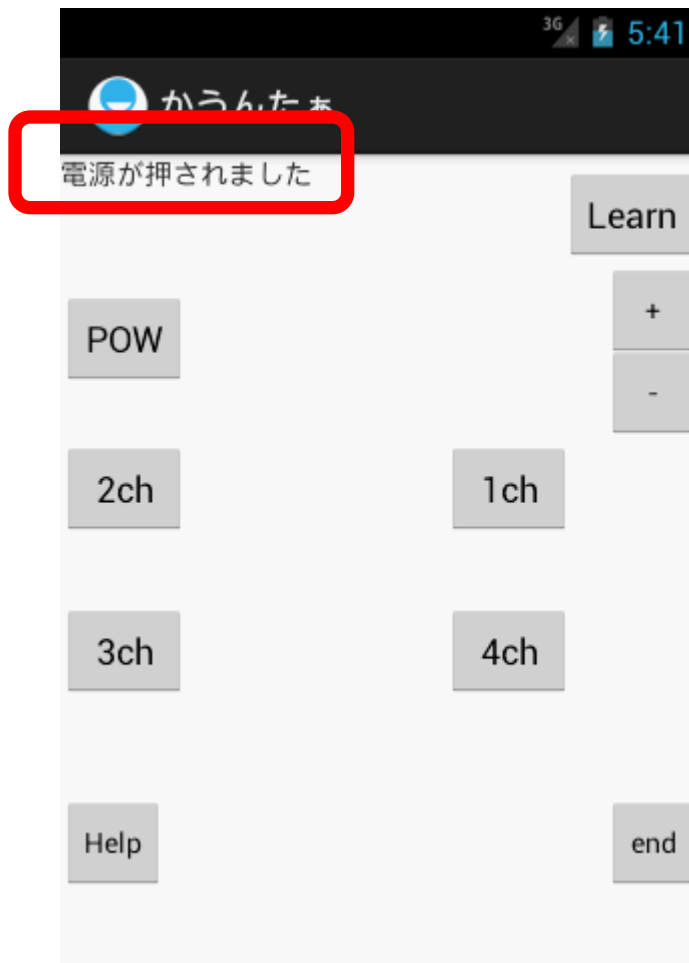
```
End(){  
    アプリ終了フラグ設定;}
```

➤現状

起動時

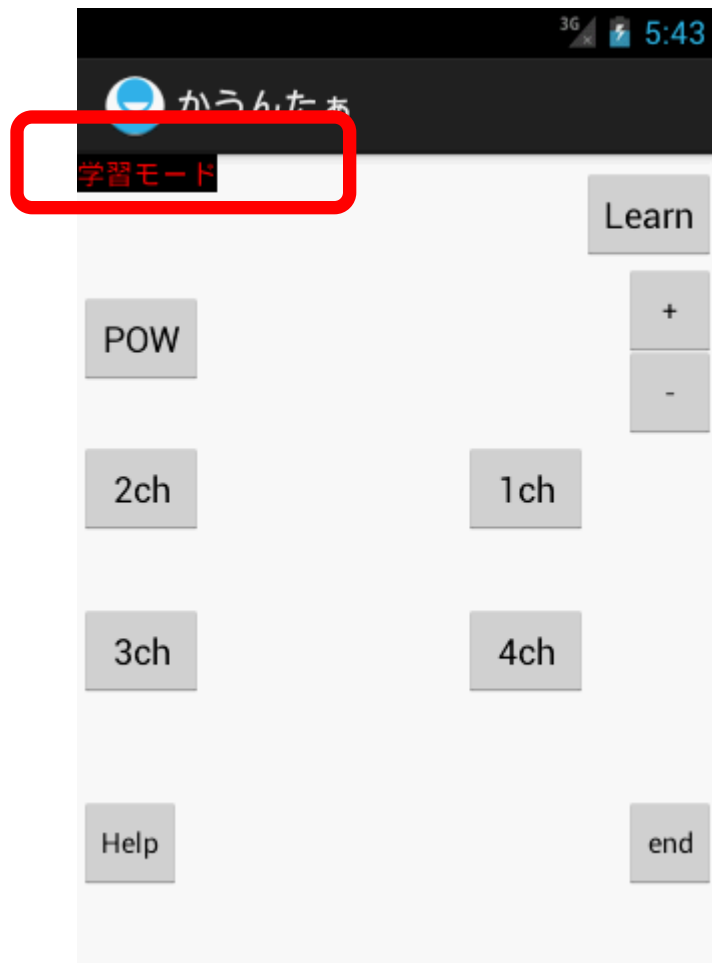


電源を押すと…

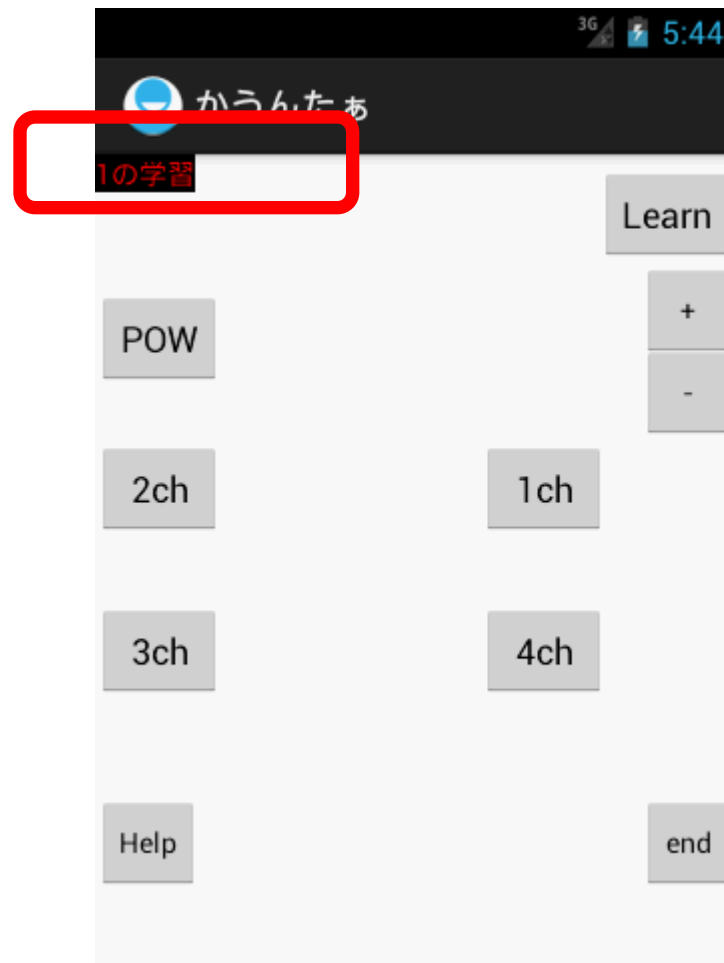


➤現状

Learnを押すと…



さらに1chを押すと…



➤機能仕様

○ クラス一覧

| アクセス | クラス名 | extends(継承) | implements(実装) |
|--------|--------------|-------------|---|
| public | MainActivity | Activity | OnClickListener OnLongClickListener OnTouchListener |
| public | IRdata | - | - |



リモコンコードを管理するクラス



➤機能仕様

○ MainActivityクラス フィールド一覧

| アクセス | 型名 | フィールド名 | 機能 |
|---------|-----------|-----------|--------------|
| private | Button | button1 | リモコン1ch |
| private | Button | button2 | リモコン2ch |
| private | Button | button3 | リモコン3ch |
| private | Button | button4 | リモコン4ch |
| private | Button | button5 | アプリ終了ボタン |
| private | Button | button6 | ヘルプ表示 |
| private | Button | button7 | リモコン音量+ |
| private | Button | button8 | リモコン音量- |
| private | Button | button9 | リモコン電源 |
| private | Button | button10 | 学習モード切り替え |
| private | TextView | textView1 | テキスト表示 |
| private | boolean | longclick | 長押しフラグ |
| private | boolean | learnf | 学習モードフラグ |
| private | S2012FPGA | fpga | インターフェイスクラス |
| private | IRdata | irdata | リモコンコード格納クラス |
| private | int | count | カウント変数 |

➤機能仕様

○ MainActivityクラス メソッド一覧

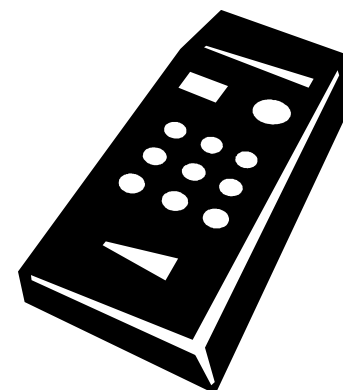
| アクセス | 型名 | メソッド名 | 機能 |
|--------|---------|-------------------------------------|--|
| public | void | onCreate(Bundle savedInstanceState) | fpgaインタフェース 領域確保 リモコンデータ初期化 ボタンリスナー設定 パラメータ: Bundle saveInstanceState: インスタンスの状態 |
| public | boolean | onTouch(View v, MotionEvent event) | 長押しフラグ設定 パラメータ: View v:ビュー MotionEvent event: タッチ状態 |
| public | void | onClick(View v) | 各ボタンのクリック処理 パラメータ: View v:ビュー |
| public | boolean | onLongClick(final View v) | 長押し処理 パラメータ: final View v:ビュー |
| public | void | Learn(View v, char ch) | リモコン学習処理 パラメータ: View v:ビュー char ch:チャンネルの名前 |
| public | boolean | onCreateOptionsMenu(Menu menu) | オプションメニュー設定 パラメータ: Menu menu:メニュー |



➤機能仕様

○ IRdataクラス フィールド一覧

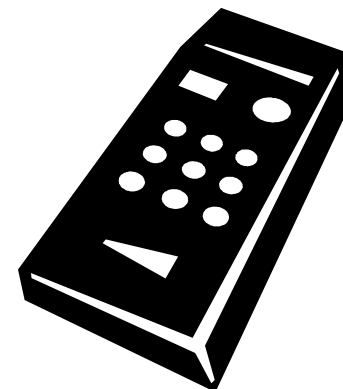
| アクセス | 型名 | フィールド名 | 機能 |
|---------|------|--------|--------------|
| private | long | ch1 | 1chリモコンコード変数 |
| private | long | ch2 | 2chリモコンコード変数 |
| private | long | ch3 | 3chリモコンコード変数 |
| private | long | ch4 | 4chリモコンコード変数 |
| private | long | chp | 音量+リモコンコード変数 |
| private | long | chm | 音量-リモコンコード変数 |
| private | long | chP | 電源リモコンコード変数 |



➤機能仕様

○ IRdataクラス メソッド一覧

| アクセス | 型名 | フィールド名 | 機能 |
|--------|------|------------------------------------|---|
| public | 無し | IRdata() | コンストラクタ |
| public | long | getData(char chname) | リモコンコードのゲッター パラメータ: char chname:チャンネルの 名前 |
| public | void | setData(char chname ,long data) | リモコンコードのセッター パラメータ: char chname:チャンネルの 名前 long data:リモコンコード |



➤ FPGAとのI/F方法

本来は…

```
void send_data(int data){  
    /*data1,0 点灯する*/  
}
```

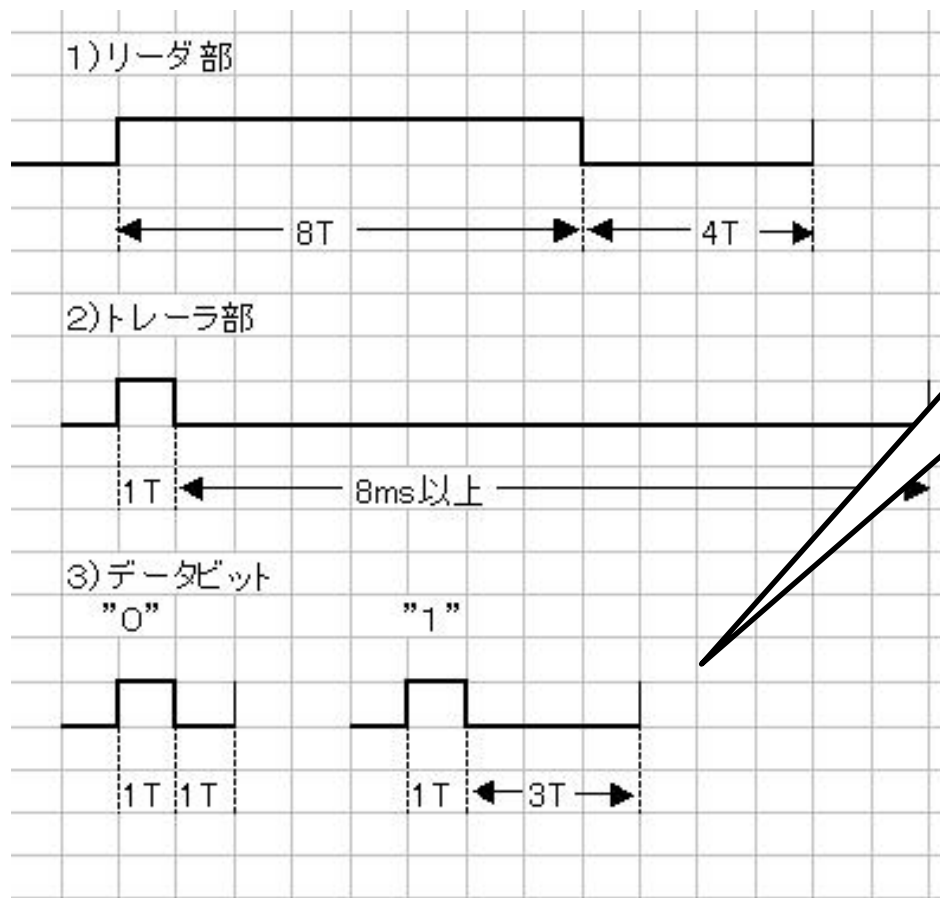
```
int receive_data(){  
    if(赤外線点灯)  
        return 1;  
    else  
        return 0;  
}
```

**FPGAとAndroid間の通信
に時間がかかり過ぎる!!**



➤ FPGAとのI/F方法

リモコンコードをひとまとめにしよう！



リモコンコードは0,1から成り立っている。

例えば...

1001 1111 0010 1010

のとき

9F2A

に変換してコードとして扱う



➤ FPGAとのI/F方法

ということで...

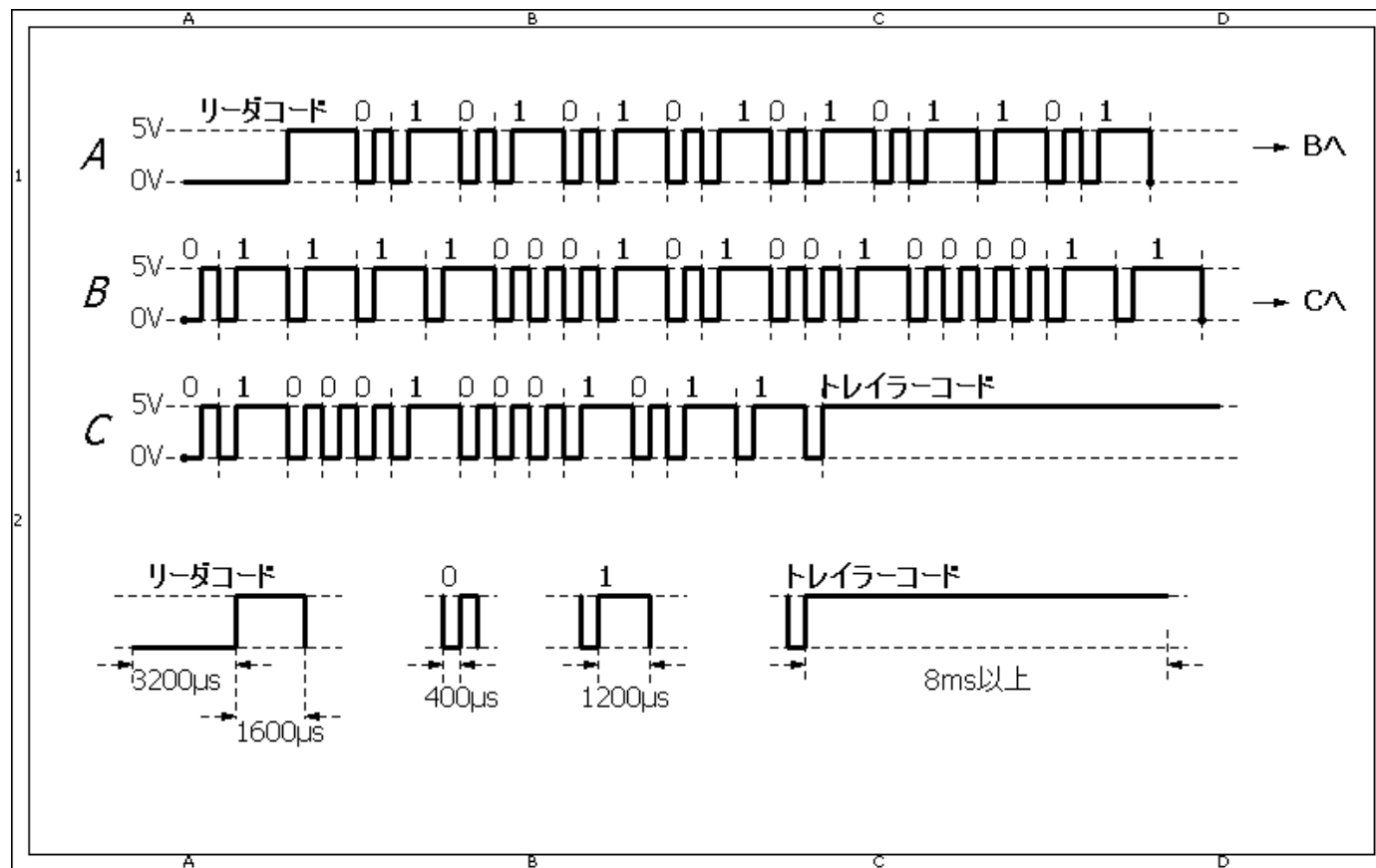
```
void sendIrDAdata(long data){  
    /*dataに応じて赤外線モジュールを点灯消灯する*/  
}
```

```
long receiveIrDAdata(){  
    return (16進数リモコンコード);  
}
```



➤ 赤外線リモコン調査

先日購入した赤外線トランシーバで研究室のTVのリモコンコードを解析してみた。



➤ 赤外線リモコン調査

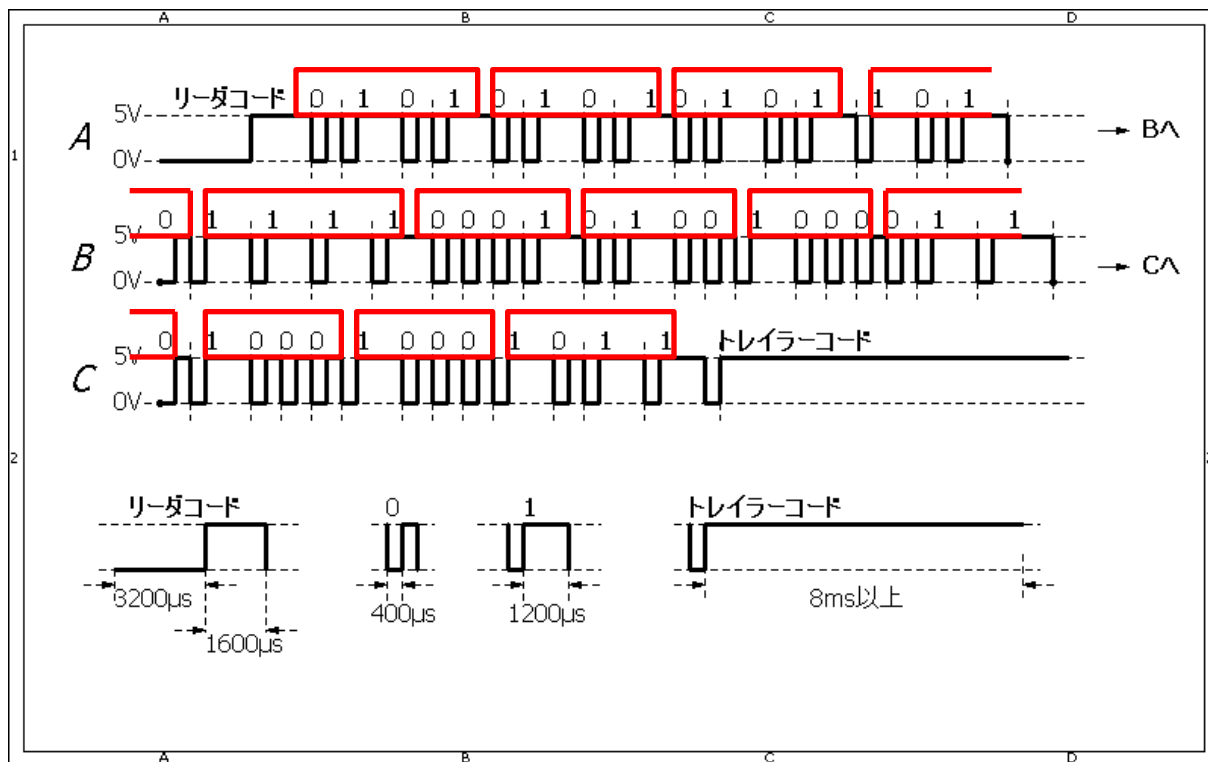
解析結果を2進数で表すと…

0101 0101 0101 1010 1111 0001 0100 1000 0110 1000 1000 1011

これを16進数に変換すると…

555AF148688B

になる。



➤ 今後の予定...

第**11**回: 中間発表

第**12**回: ソフトウェア改良・動作テスト

第**13**回: 動作テスト・システム統合

第**14**回: システム統合

第**15**回: 最終発表

