

# HASC Tutorial 2012

## Part 2



<http://hasc.jp/>

### スライドコンテンツ

HASC Loggerの使い方

HASC Toolのセットアップ

HASC Toolの機能と使い方

名古屋大学 河口研究室  
修士1年 渡辺 穂高

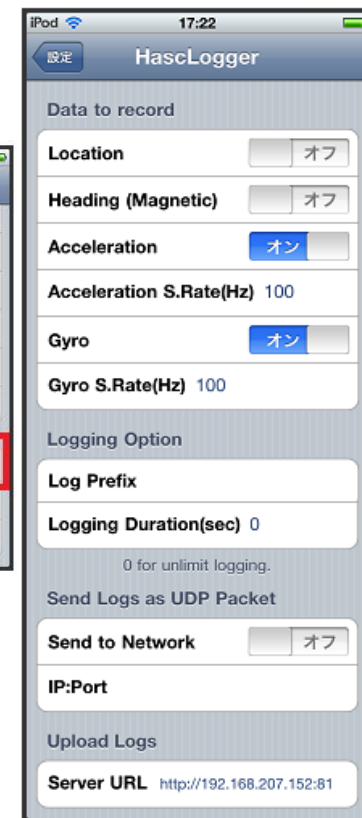
# HASC Loggerの入手

- iPhone/iPod touch
  - App StoreでHASC Loggerを検索
  - HASC Loggerをインストール
- Android端末
  - Google Play(Market)でHASC Loggerを検索
  - HASC Logger (Nobuo Kawaguchi版)をインストール



# HASC Loggerの設定

- HASC Challenge 2012 > HASC Loggerを参照
  - <http://hasc.jp/hc2012/hasclogger.html>
- 設定アプリ内のHASC Loggerを選択
  - 収集対象のセンサ
  - サンプリングレート
  - データ収集時間
  - 行動データのリアルタイム転送先
  - HASC LoggerServerのデータ転送先



# 行動データ収集設定

- Data to record
  - Location : 位置情報
  - Heading(Magnetic) : 地磁気情報
  - Acceleration : 加速度情報
  - Acceleration S.Rate(Hz) : 加速度計測周波数
  - Gyro : 角速度情報
  - Gyro S.Rate(Hz) : 角速度計測周波数
  - Audio : 音声情報
- Logging Option
  - Logging Duration(sec) : データ計測時間
    - 0の場合 : 制限なし

# 行動データの送信

- Send Logs as UDP Packet
  - HASCToolへ行動データをリアルタイム送信
  - Send to Network
  - IP:Port
    - HASCToolを実行するPCのIPアドレス+ ":" +ポート番号6666
    - Ex) 192.168.100.10:6666
- Upload Logs
  - 収集した行動データをHASCToolにアップロード
  - Server URL
    - "http://" + HASCToolを実行するPCのIPアドレス + ":" +ポート番号 + "/"
    - Ex) http://192.168.100.10:80/

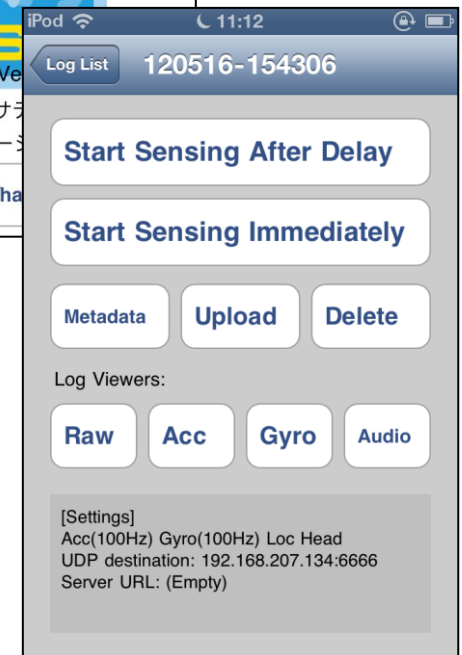
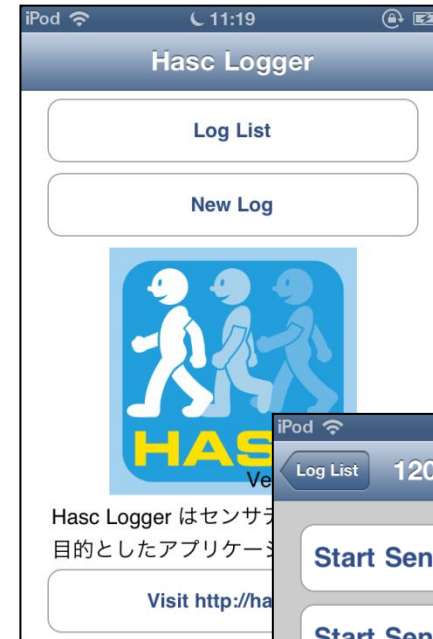
# HASC Logger 起動画面

- HASC Loggerの起動

- Log List
  - ログの一覧を表示
- New Log
  - 新規ログを作成

- ログ情報画面

- Start Sensing After Delay
- Start Sensing Immediately
- Metadata
- Upload
- Delete
- Log Viewers: Raw, Acc, Gyro, Audio



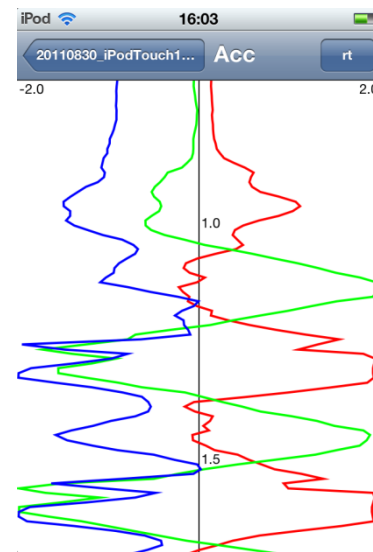
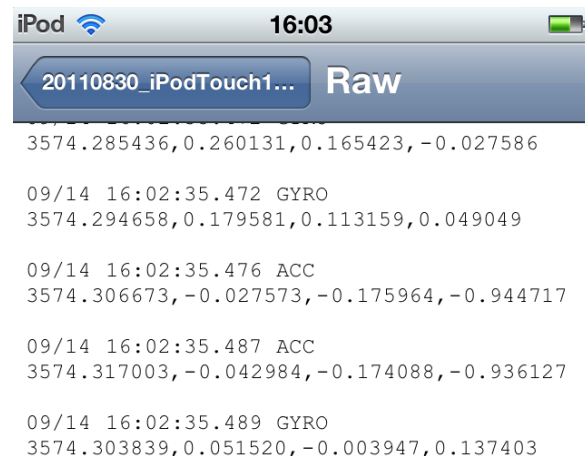
# メタ情報の設定

- ログ情報画面 > Metadata
- データ収集環境情報を入力
  - Comment
    - 直接打ち込み
    - 自由にコメントを書けます
  - Tags
    - リストから選択・追加
    - 直接打ち込みも可能
    - カンマ“, ”で区切る



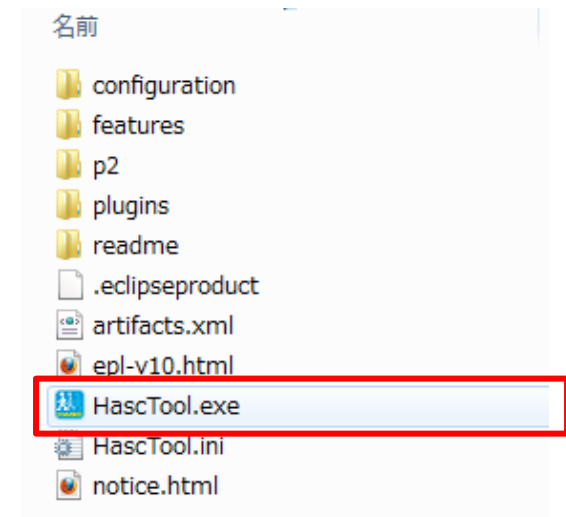
# データの収集と閲覧

- データ収集の開始
  - ログ情報画面 > Start Sensing ~
- データ収集中
  - 現在のセンサ値を表示
- データ収集の終了
  - 手動：Stop Sensingボタンを押す
  - 自動：事前にデータ収集時間を指定
- 収集データの形式
  - 日時 センサの種類  
時間(sec), センサ値 1, センサ値 2, ...
- 収集データの閲覧
  - ログ情報画面 > Log Viewers項目で  
**Raw, Acc, Gyro** ボタンを選択



# HASC Tool (ダウンロード版)

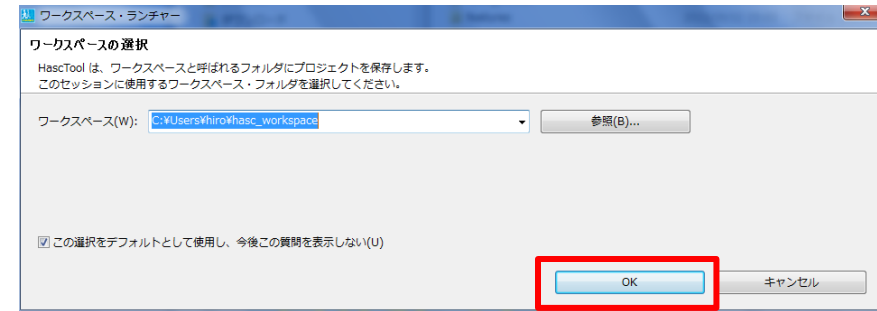
- HASC Challenge 2012 > ダウンロード
  - <http://hasc.jp/hc2012/download.html>
- HASC Tool・SampleProjectの入手
  - HASC Tool
    - Mac版
    - Windows版
  - Sample Data Project
  - RealWorldActivityData
- それぞれのファイルを解凍
- HASC Toolディレクトリを開く
- HASC Toolの起動



# HASC Toolの起動

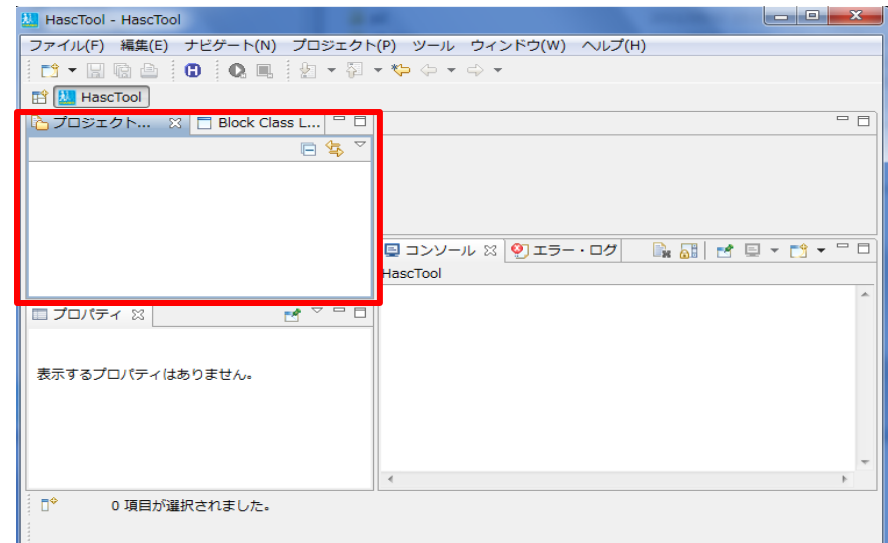
- ワークスペースの選択

- ◻ 任意に設定可
- ◻ デフォルトでもOK



- 新規プロジェクトの作成

- ◻ プロジェクト・エクスプローラ内で右クリック
- ◻ 新規 > プロジェクト
- ◻ 一般 > プロジェクト > 次へ
- ◻ プロジェクト名
  - Ex) HASCtestPrj

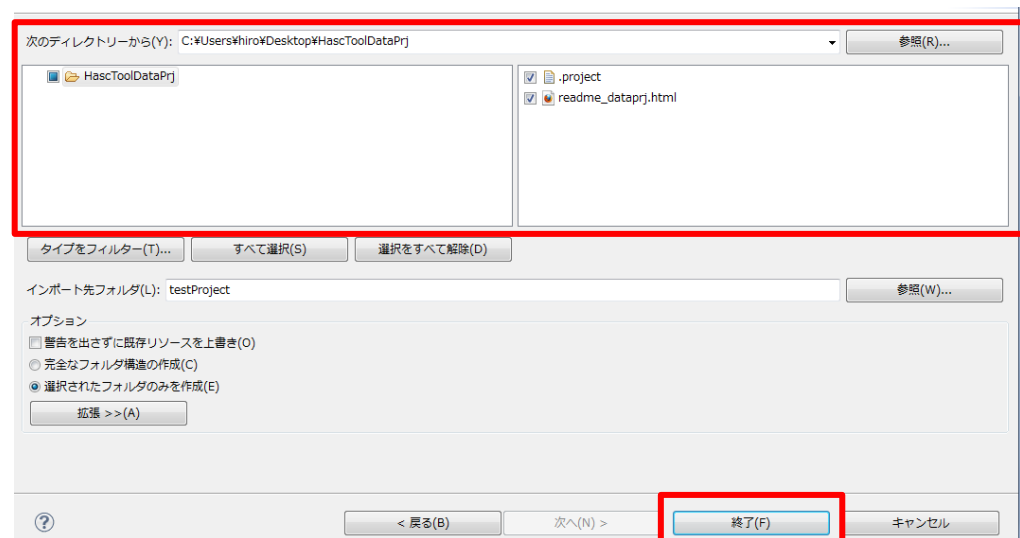


# Sample Data Projectのインポート

## SampleDataProject : サンプル行動データセット + 処理スクリプト

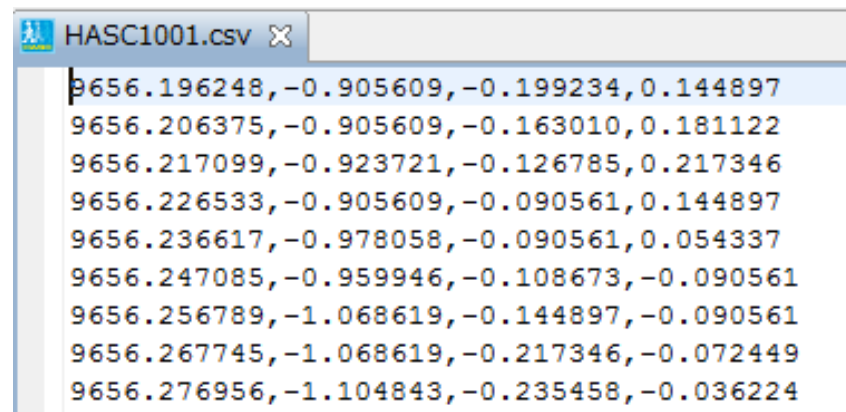
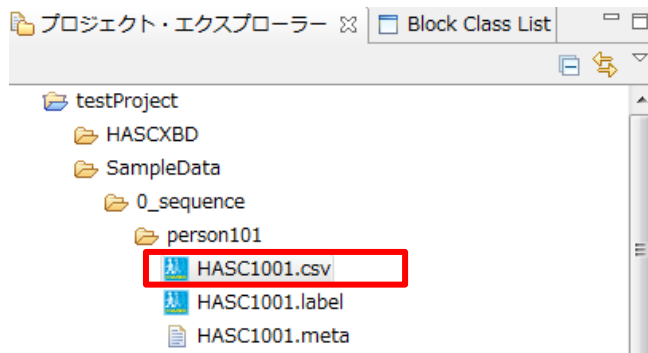
### • 作成したプロジェクト

- 右クリック> インポート> 一般> ファイルシステム
- 参照元の選択 (SampleDataProjectを選択)
- 出てきたすべての項目にチェック
- 終了を選択
- インポート完了



# 行動データ(.csv)の閲覧

- データ構成
  - 加速度データ
  - [時間(sec)],[x軸],[y軸],[z軸]
  - データフォーマットについては、  
HASC Challenge2012 > データ形式  
(<http://hasc.jp/hc2012/dataFormat.html>)  
に記載されています。



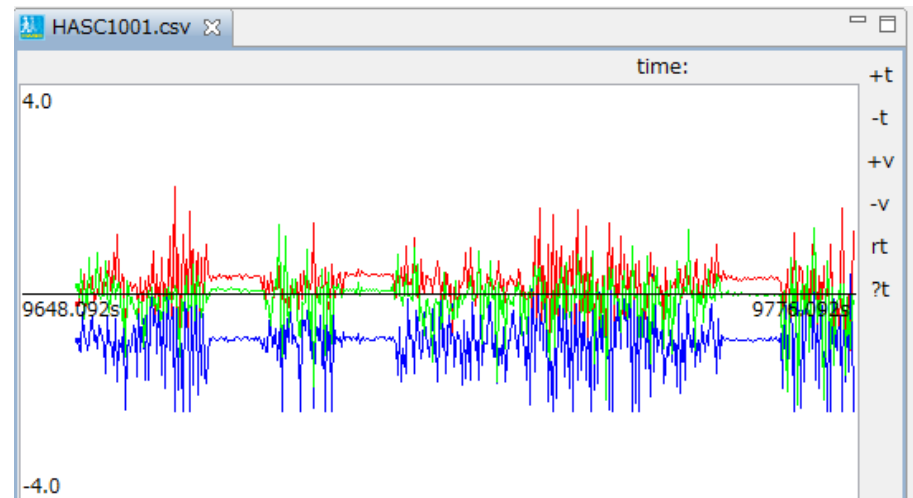
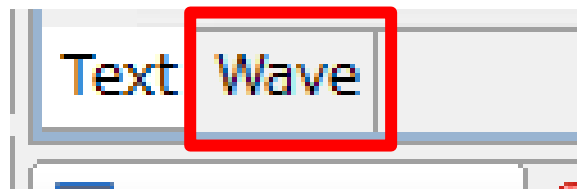
# 行動データのViewer

## • 波形の表示

- 縦軸：加速度(G), 横軸：時間(sec)
- センサX軸：青色, センサY軸：赤色, センサZ軸：緑色

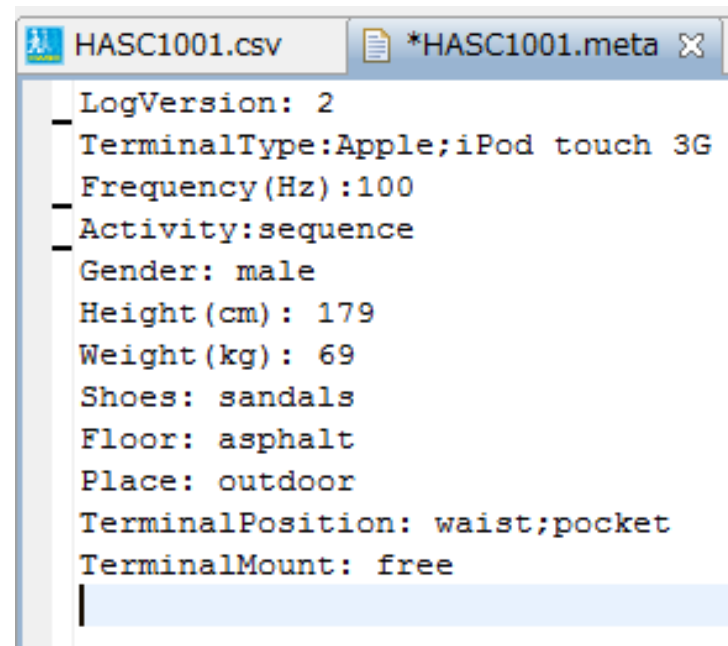
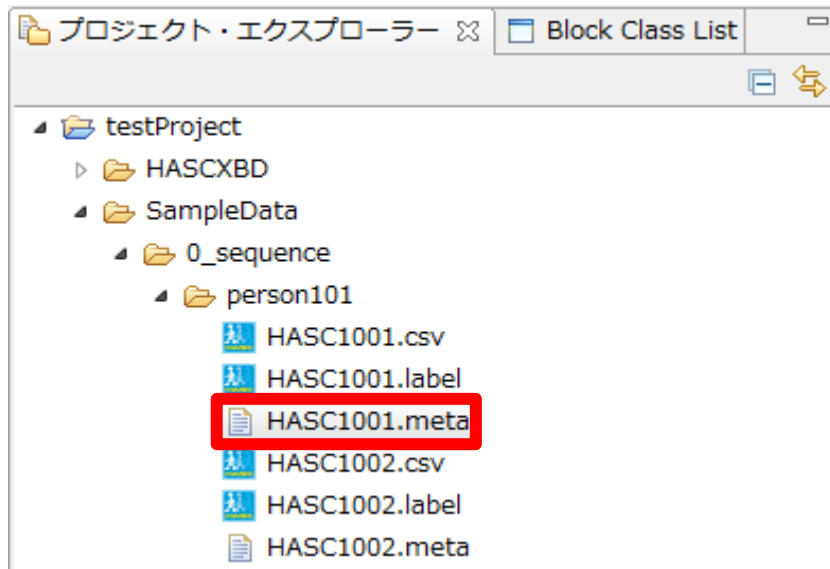
## • 波形ビューの操作

- 表示する範囲の変更：“+t”、“-t”、“+v”、“-v”
- 時刻の左端を0とする：“rt”
- 選択した時間範囲をテキストエディタ上に表示：“?t”



# メタデータ(.meta)

- データの要素
  - TerminalType, Frequency(Hz), Activity, Gender, Height(cm) など
  - key: value



# ラベルデータ(.label)

- データ構成
  - 1行目：対象とするcsvデータのパス
- 音声データ有りの場合
  - 2行目に音声データのパス
  - 下図を参考
  - `#audiofile:[ファイルパス]; offsettime:[遅延時間(s)]`

```
hasc-111018-165936.label x
#targetfile:$(projectRoot)/rwActData/person671/hasc-111018-165936-acc.csv
#audiofile:$(projectRoot)/rwActData/person671/hasc-111018-165936-audio.wav; offsettime:10.0
5057.661,5091.26,walk;floor;B2F;steps;44
5071.934,,walk;turn
5098.502,5126.499,escalatorUp;stay
5125.117,,move;floor;B1F
5127.665,5143.411,walk;floor;B1F;steps;22
5154.309,5162.703,escalatorUp;stair
5165.451,,move;floor;1F
```

音声データを再生する  
場合の遅延オフセット

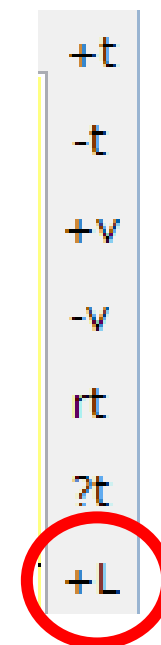
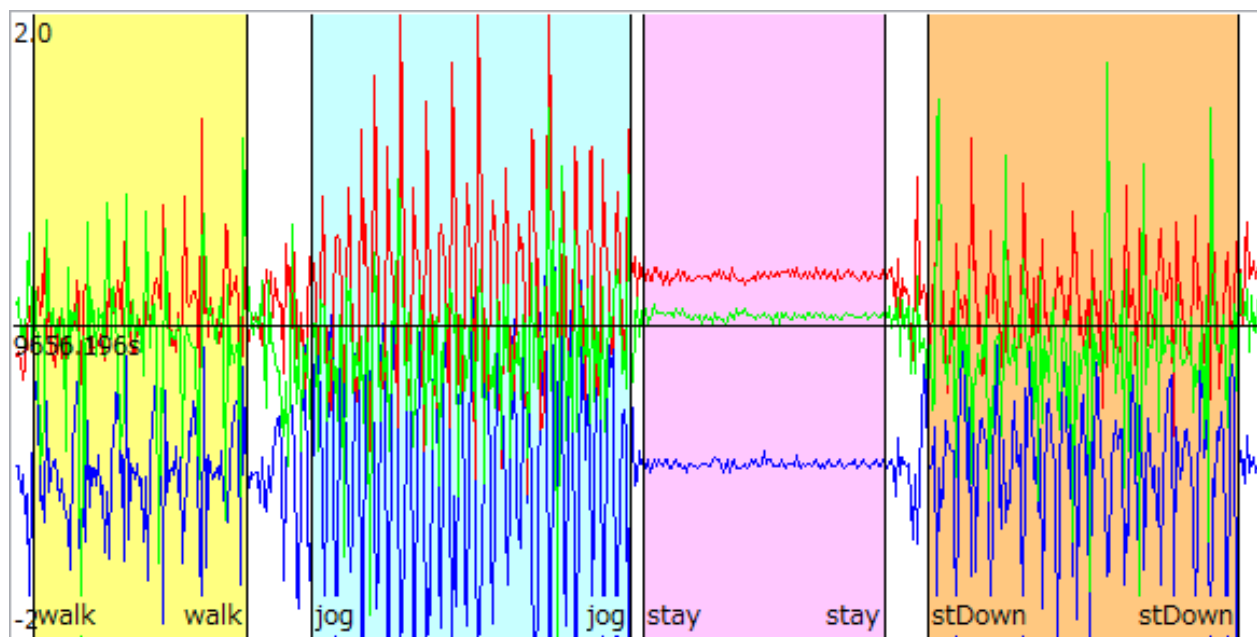
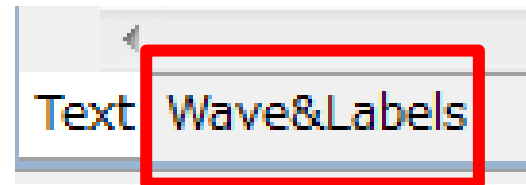
# ラベルデータ

- ラベル情報
  - [開始時刻(sec)], [終了時刻(sec)], [ラベル]
- イベント情報
  - [イベント時刻(sec)],,, [ラベル]
  - 下図 4 行目を参考
  - Ex) 6071.934,,,move;floor;B1F

```
hasc-111018-165936.label x
#targetfile:$(projectRoot)/rwActData/person671/hasc-111018-165936-acc.csv
#audiofile:$(projectRoot)/rwActData/person671/hasc-111018-165936-audio.wav; offsettime:10.0
5057.661,5091.26,walk;floor;B2F;steps;44
5071.934,,walk;turn
5098.502,5126.499,escalatorUp;stay
5125.117,,move;floor;B1F
5127.665,5143.411,walk;floor;B1F;steps;22
5154.309,5162.703,escalatorUp;stair
5165.451,,move;floor;1F
```

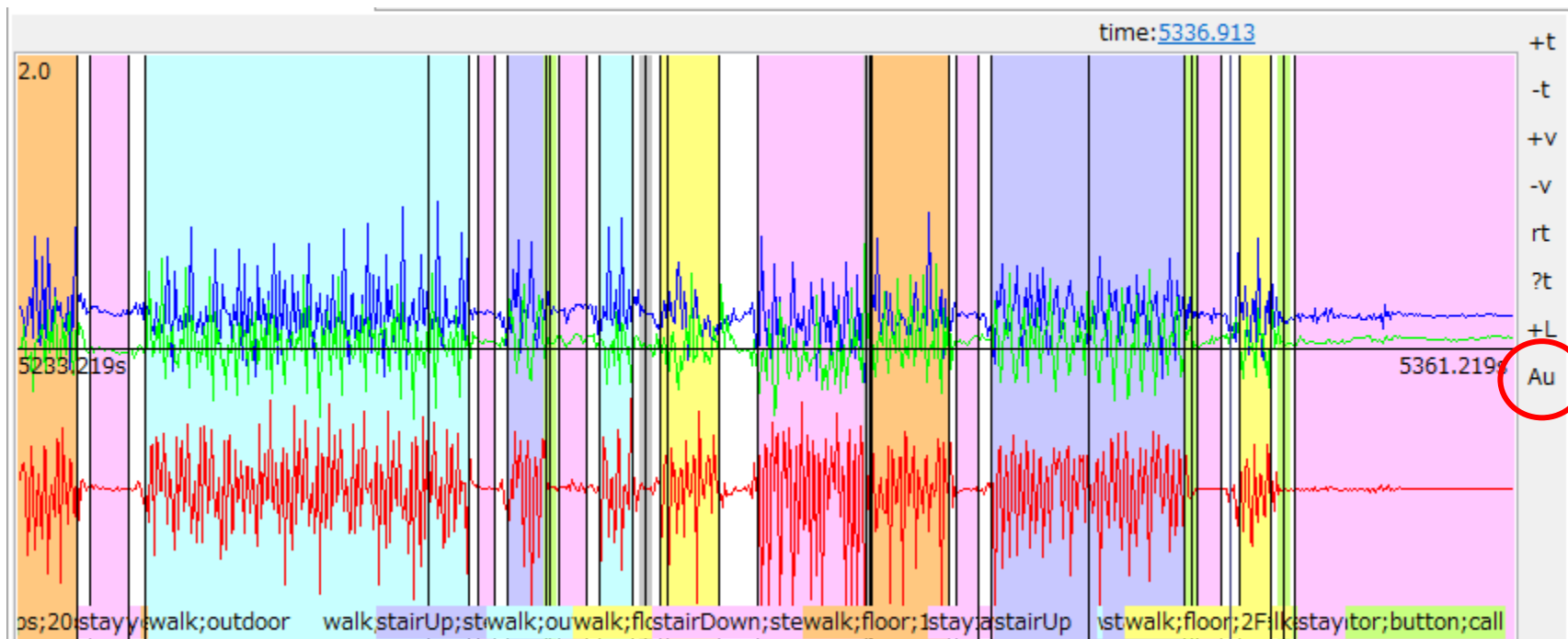
# ラベルデータのViewer

- ラベル付き波形表示
  - Wave&Labelsをクリック
- ラベルの付与
  - ドラッグで範囲を指定 > "+L"



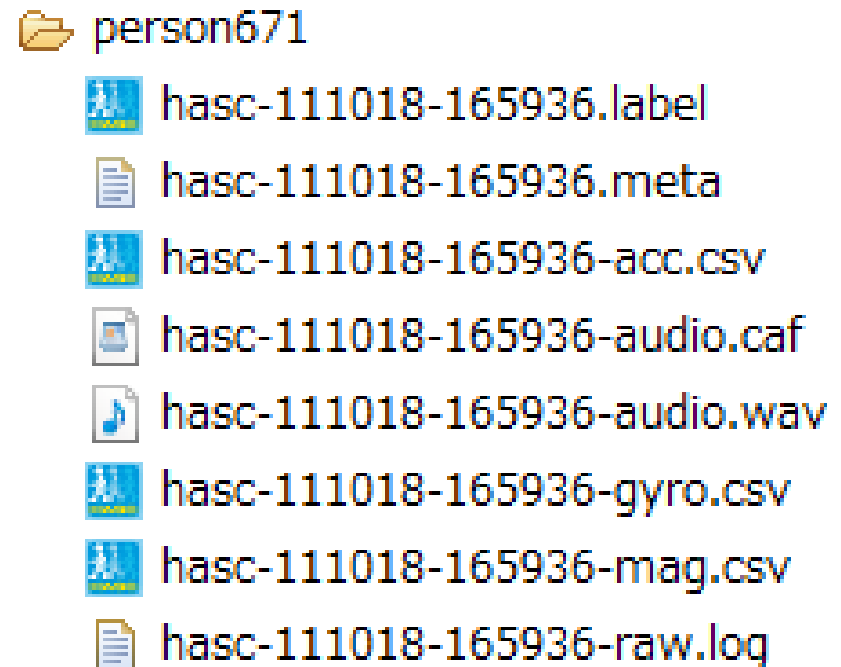
# ラベルデータのViewer

- 音声情報と行動波形の同期再生が可能
- 音の再生：“Au”



# 行動データセット

- 行動データ
  - 拡張子：.CSV
  - 加速度情報，角速度情報，地磁気情報など
- メタデータ
  - 拡張子：.meta
  - 計測環境の情報
- ラベルデータ
  - 拡張子：.label
- 音データ
  - 拡張子：caf,wav



# HASC Toolの機能説明

- HASC Loggerとの連携機能
  - HASCLoggerServerへデータ送信
- ラベル付与機能
  - ラベルファイルの編集
  - ラベル付の手順紹介
- 行動データ処理機能
  - HASCXBDディレクトリの内部構成
  - HASCXBDファイルの説明

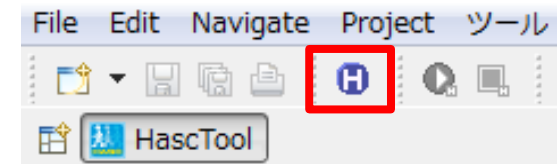
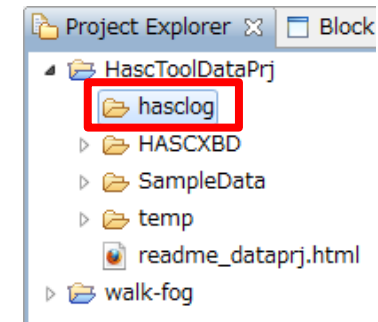
# HASCLoggerServerへデータ送信

- HASC Tool側

- ログの保存先ディレクトリを選択
- ツールバーの(H)ボタン
  - 起動, 終了
  - bindするTCPポートの設定
    - 任意選択 (デフォルトのままでもOK)

- HASC Logger側

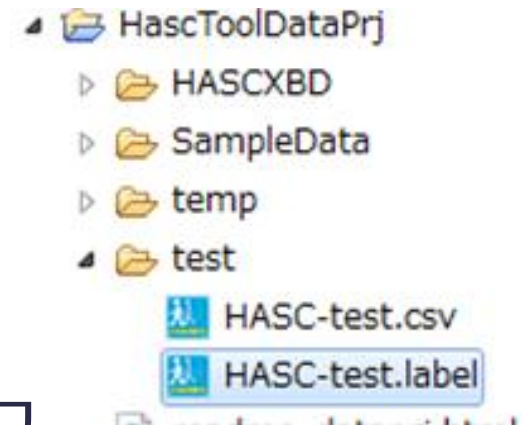
- Uploadボタン
- 確認画面 (右図)
- アップロード内容
  - 各種センサデータ, メタデータ



# ラベル付与機能

- ラベル付手順

- ラベルファイルを用意する
  - 拡張子: .label
- ターゲットファイルのPathを入力
  - ラベルファイルのフォーマットは以下に記載  
<http://www.hasc.jp/hc2012/dataFormat.html>
- ラベルファイルを編集



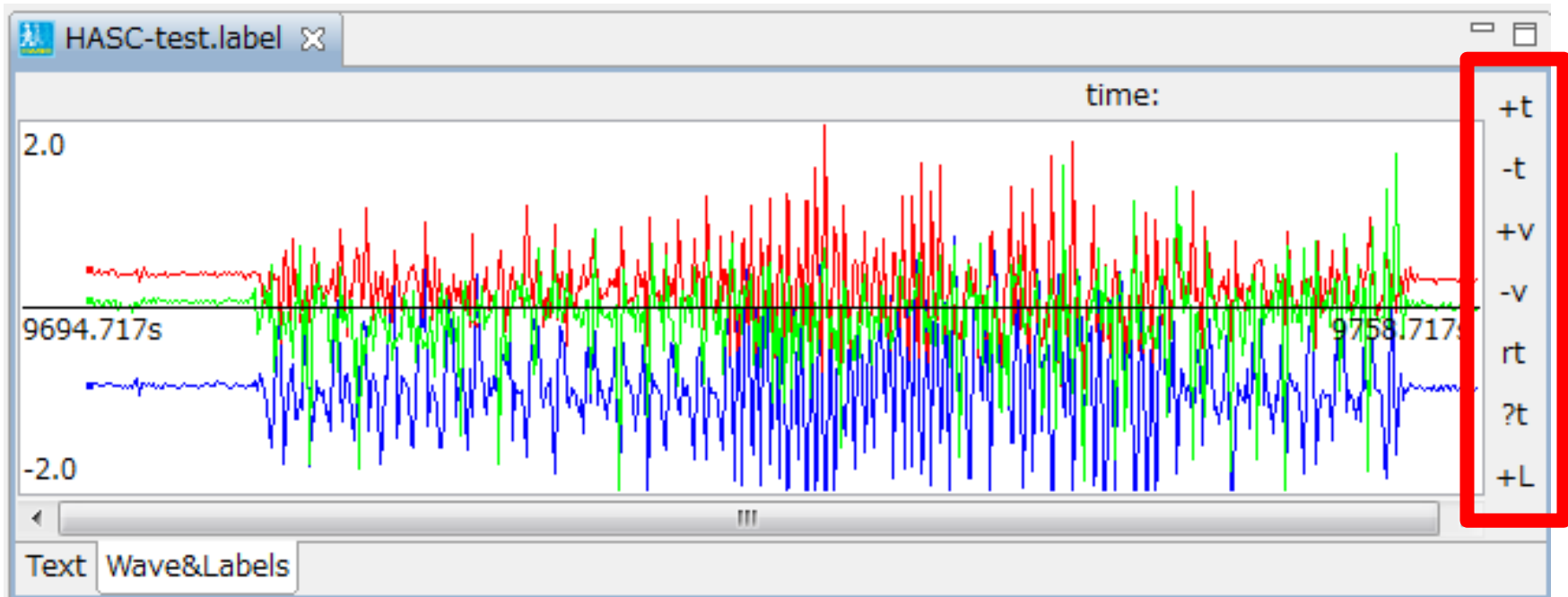
HASC-test.label ✖

```
#targetfile:$(projectRoot)/test/HASC-test.csv
```

# ラベルファイルの編集

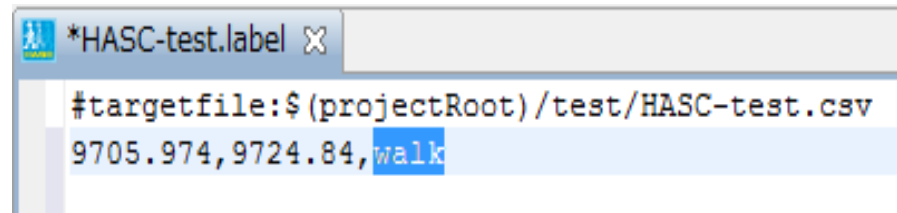
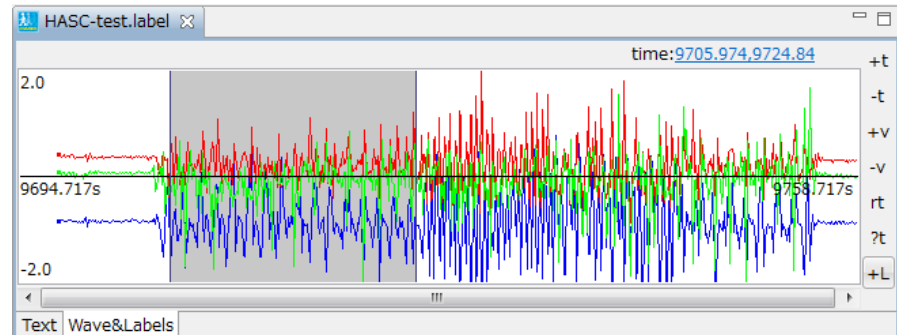
## • 編集ボタンは右側に配置

- ◻ 表示する範囲の変更：“+t”、“-t”、“+v”、“-v”
- ◻ 時刻の左端を0とする：“rt”
- ◻ 選択した時間範囲をテキストエディタ上に表示：“?t”
- ◻ ラベル付ボタン：“+L”

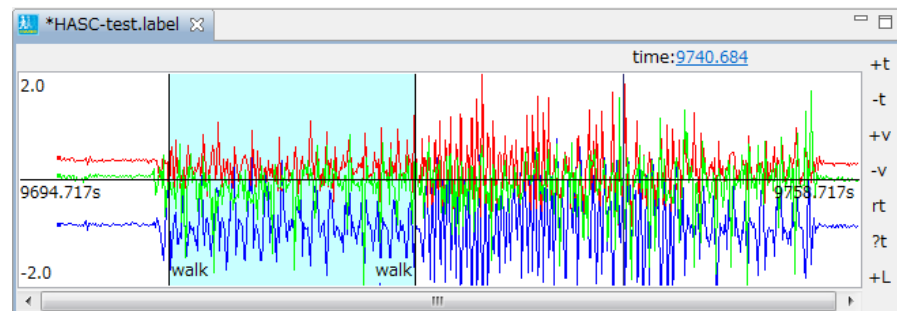


# ラベル付与の手順

- ラベル範囲を選択
  - 範囲をドラッグ
- +L ボタンを押す
- テキスト編集
  - ラベル情報の入力
  - 画像ではwalk
- ラベル付け完了



The screenshot shows a text editor window titled "\*HASC-test.label". The text content is: `#targetfile:$(projectRoot)/test/HASC-test.csv` followed by `9705.974,9724.84,walk`. The word "walk" is highlighted in blue.



上記作業を繰り返す

# 行動データ処理機能

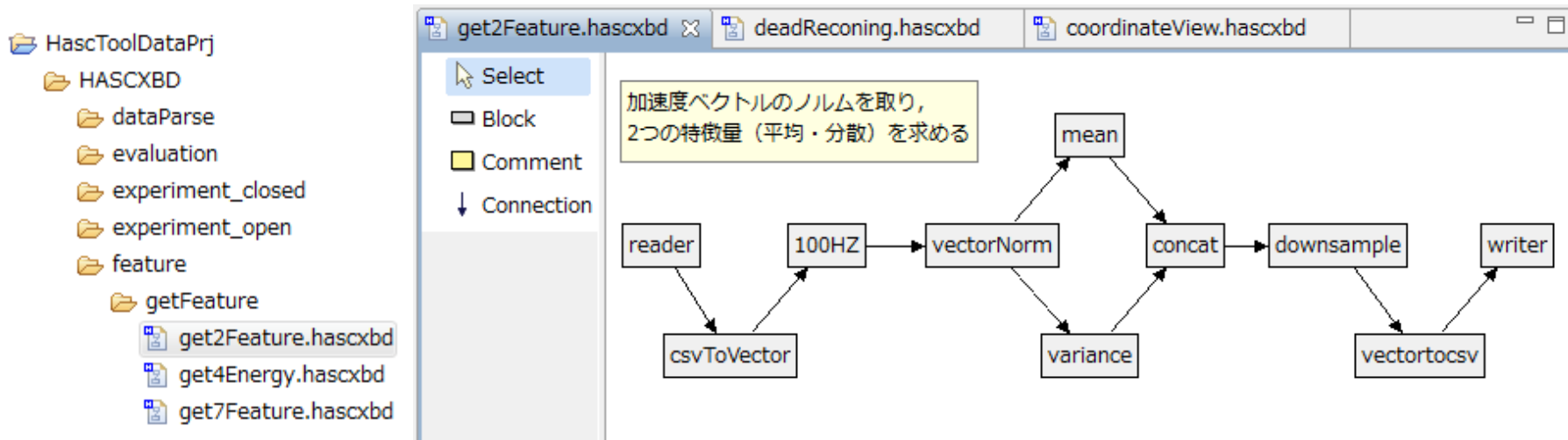
- HASCXBDディレクトリ
  - 特徴量抽出を実行
    - feature
  - 機械学習 & 評価を実行
    - experiment\_~
    - weka(機械学習)
    - evaluation(評価)
  - 閾値処理
    - Threshold
  - HASC Loggerからリアルタイムデータ送信
    - Udp
  - ファイルのカウント & メタデータテーブルの作成
    - file

- 📁 HASCXBD 405
  - 📁 dataParse 191
  - 📁 evaluation 379
  - 📁 experiment\_closed 405
  - 📁 experiment\_open 405
  - 📁 feature 405
  - 📁 file 283
  - 📁 frequency 253
  - 📁 label 191
  - 📁 sample 381
  - 📁 table 253
  - 📁 threshold 377
  - 📁 udp 294
  - 📁 view 200
  - 📁 weka 385

# HASCXBDファイル

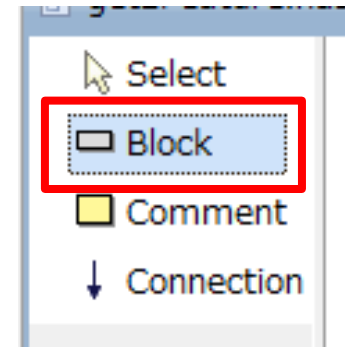
- HASC Tool上で様々なデータ処理を実行
- ブロック図による編集
  - ブロック（フィルタ）
  - コネクタ
  - コメント
  - ダイアグラムの階層化

実行/停止



# ブロック (フィルタ)

- 実際にデータ処理を行います
- 左メニューからブロック作成モード
  - エディタ上で配置したい場所をクリック
  - クラスファイルの場所を入力
- プロパティに値をセット

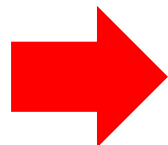


作成するブロックのクラス名を入力して下さい

```
jp.hasc.hasctool.core.runtime.filter.MeanFilter|
```

Property	Value
▾ _block	
_Type	BeanBlock
Class	jp.hasc.hasctool.core.runtime.filter.MeanFilter
Comment	
Name	mean
▾ bean properties	
sampleNum	256

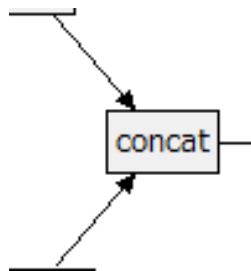
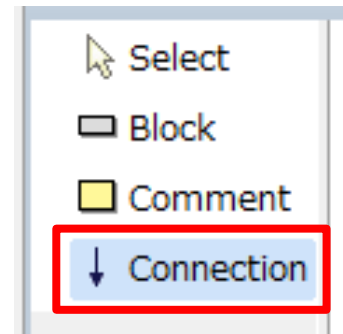
参考



<http://hasc.jp/hc2012/hasctool.html>  
“HASC Toolの説明”

# コネクタ

- ブロック間のデータ送信
- 左メニューからコネクタ作成モード
  - 出力側のブロックをクリックで選択
  - 入力側のブロックをクリックでコネクタ完成
- 複数入力ブロック
  - コネクタのプロパティを設定
    - それぞれにinputPort[0]~inputPort[n]を割り当てる

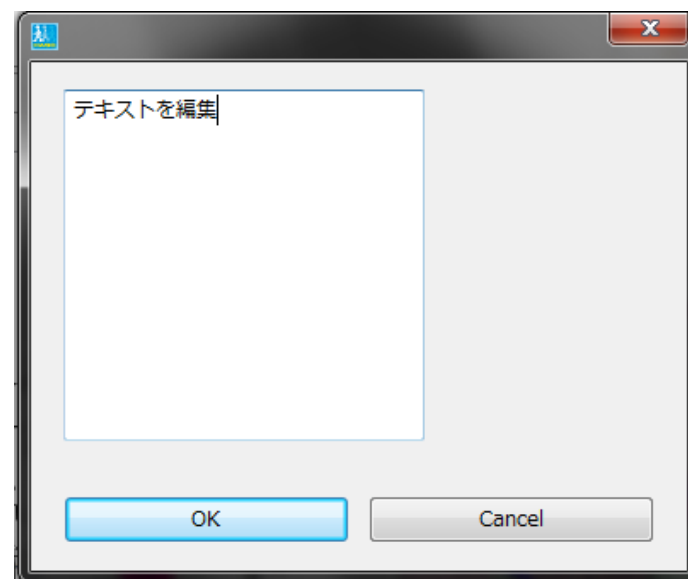
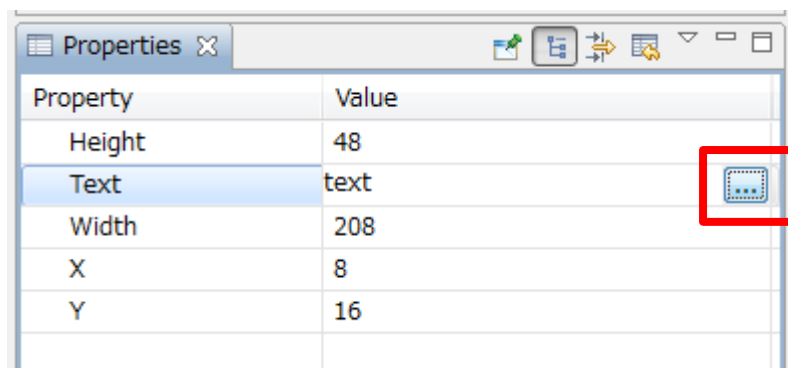
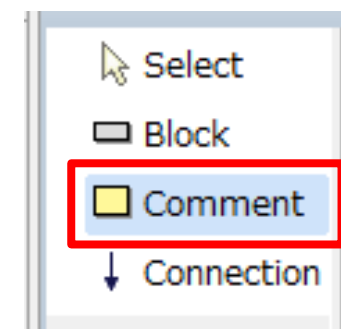


_Type	Connection
Comment	
InputPort	inputPort[1]
OutputPort	outputPort

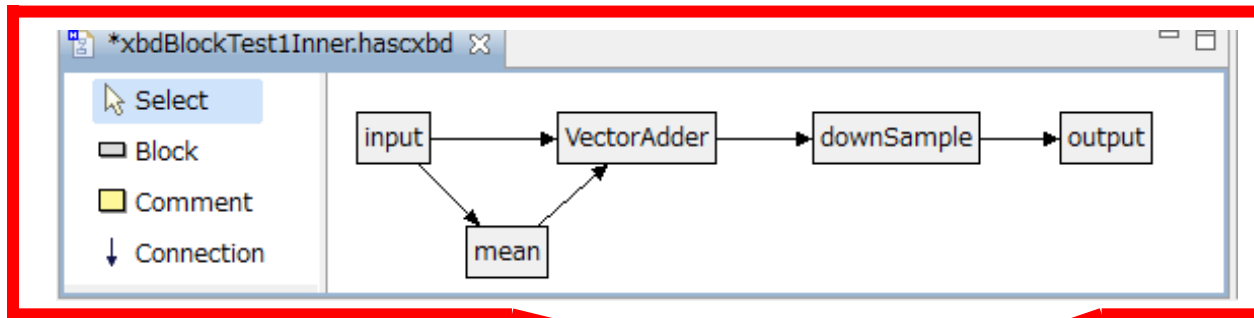
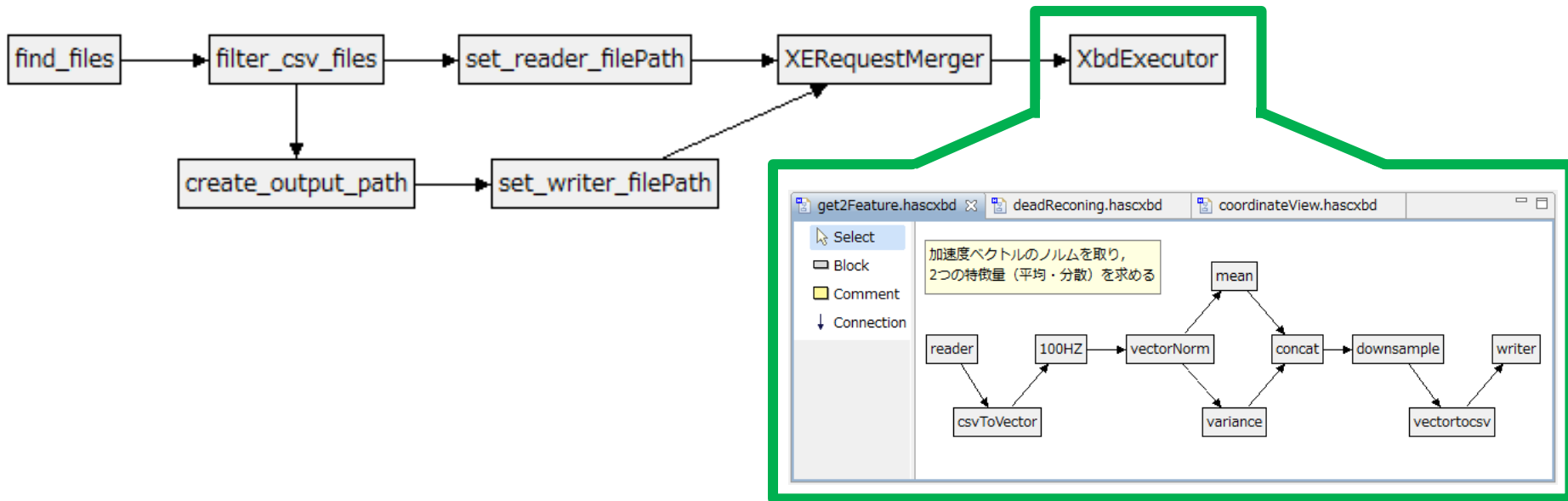
# コメント

- 補足説明など自由なテキストを記述
- 左メニューからコメント作成モード
  - エディタ上をドラッグでコメントボックス作成
- プロパティタブのTextを選択
  - [...]をクリック
- テキストを編集し, OK

加速度ベクトルのノルムを取り,  
2つの特徴量(平均・分散)を求める

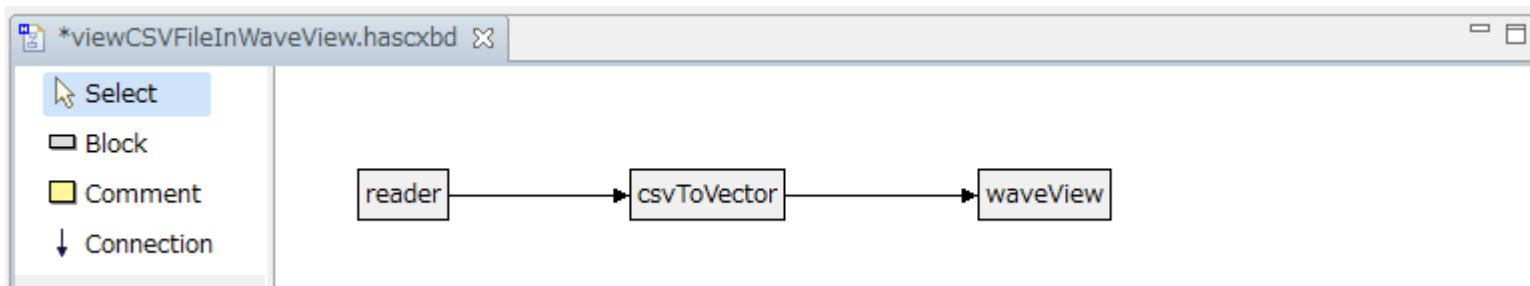
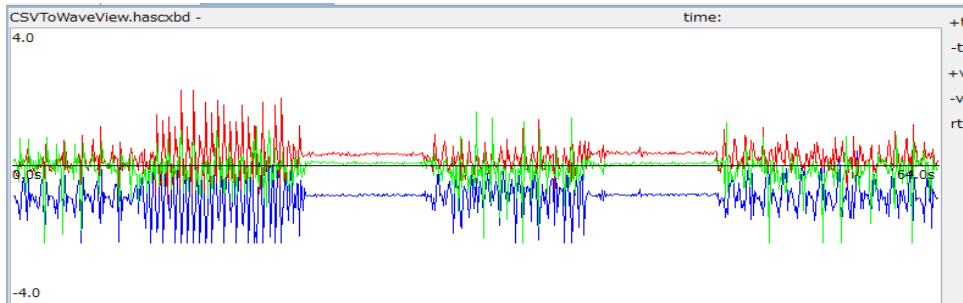
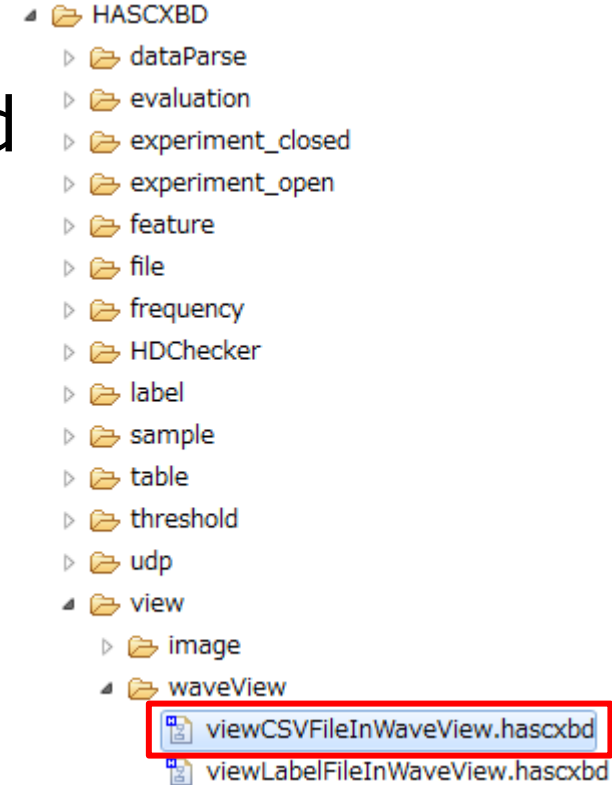


# ダイアグラムの階層化



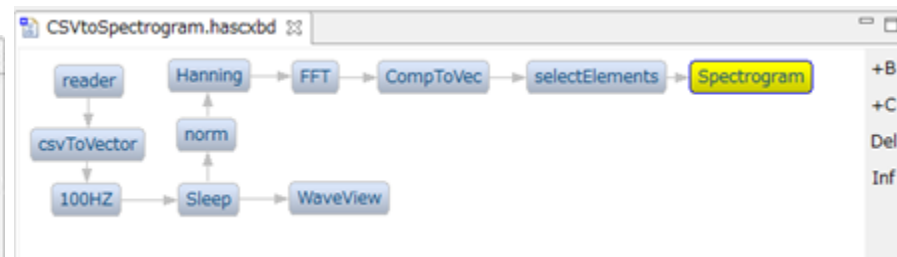
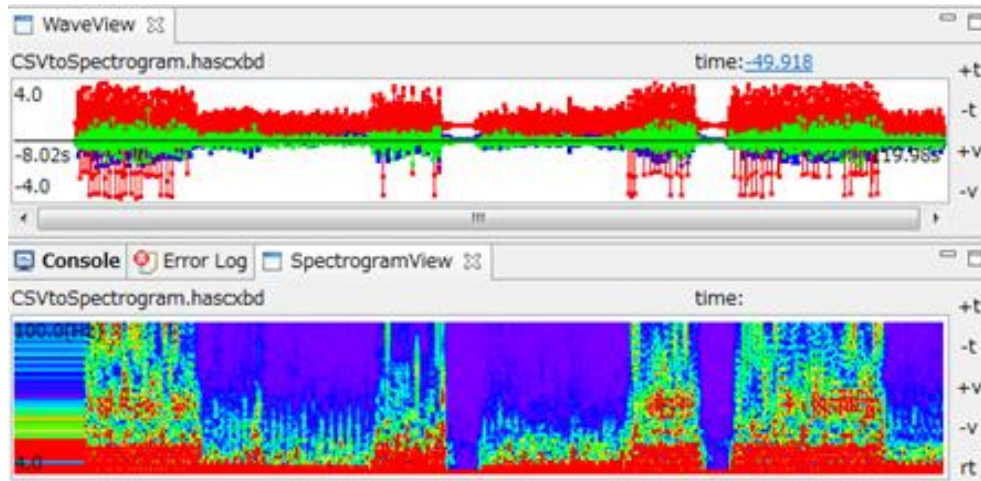
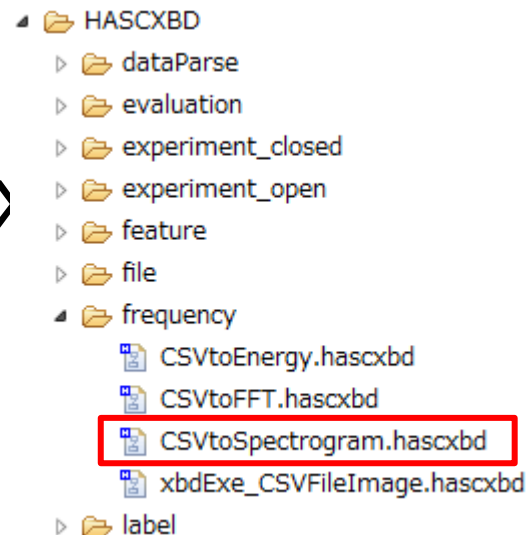
# 行動データの読込

- HASCXBD>view>waveView>viewCSVFileInWaveView.hascxbd
- csvファイルを読み込む
- 行動データ⇒時系列波形



# スペクトログラムの表示

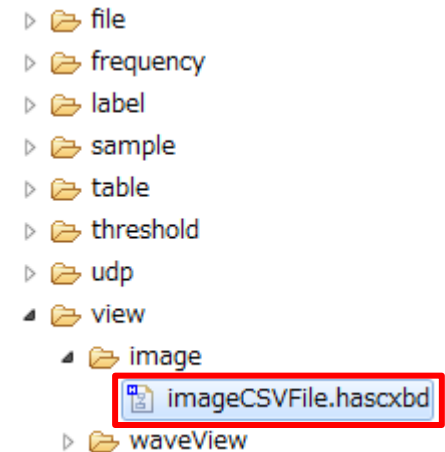
- HASCXBD > frequency > CSVtoSpectrogram.hascx
- csvファイルを読み込む
  - 行動データ  
⇒ スペクトログラム



**Hanning : ハニング窓**  
**FFT : フーリエ変換**  
**CompToVec : 複素数⇒絶対値**

# 画像ファイルの取得

- HASCXBD>view>image>  
imageCSVFile.hascxbd
- csvファイルを読み込む
  - 行動データ  
⇒時系列波形  
⇒画像ファイル



# フィルタの追加

- 追加するフィルタの決定

- Block Class Listから選択
- 右図ではMeanFilterを選択

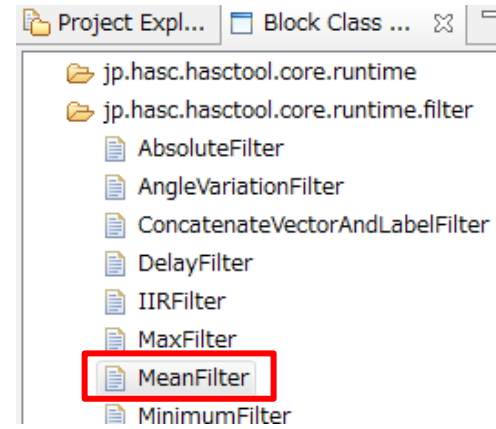
- 平均を計算するフィルタ

- 場所：

jp.hasc.hasctool.core.runtime.filter

- 選択したらブロックダイアグラム内にできます

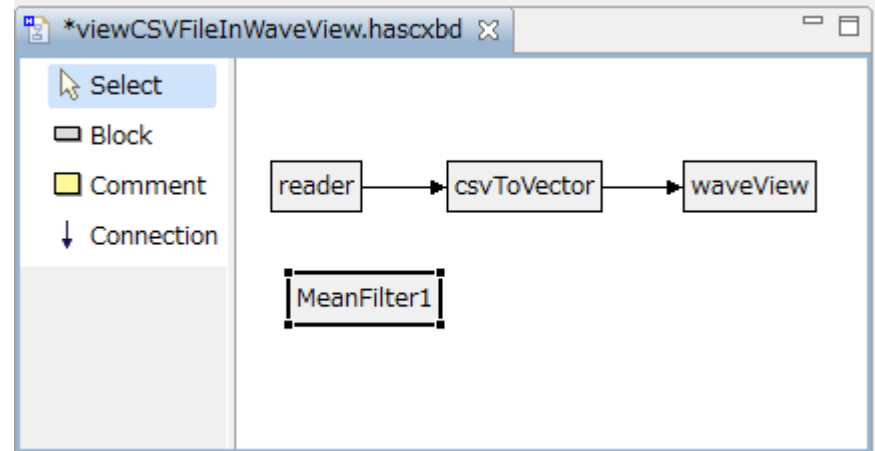
- 出てきたブロックを選択
- プロパティを設定



Property	Value
_block	
_Type	BeanBlock
Class	jp.hasc.hasctool.core.runtime.filter.MeanFilter
Comment	
Name	MeanFilter1
bean properties	
sampleNum	100

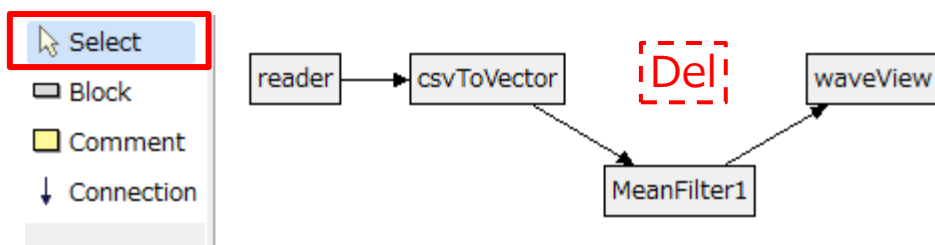
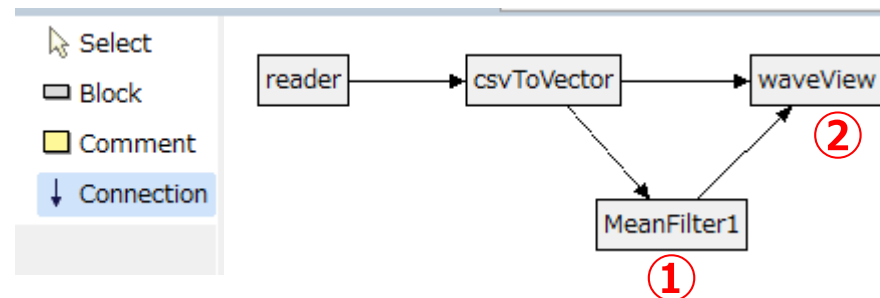
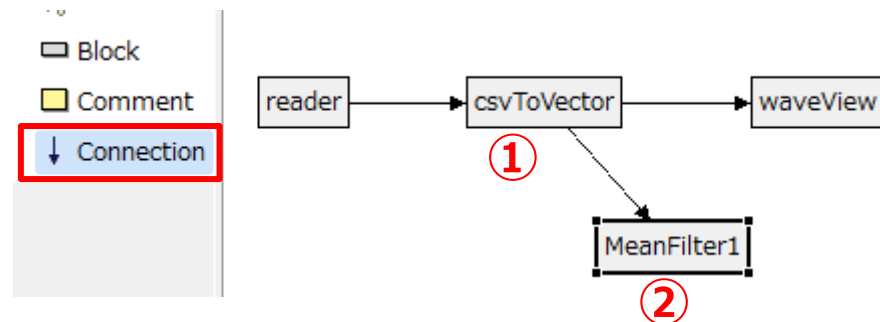
名前の編集可

対象データの数を設定



# ブロックダイアグラムの編集

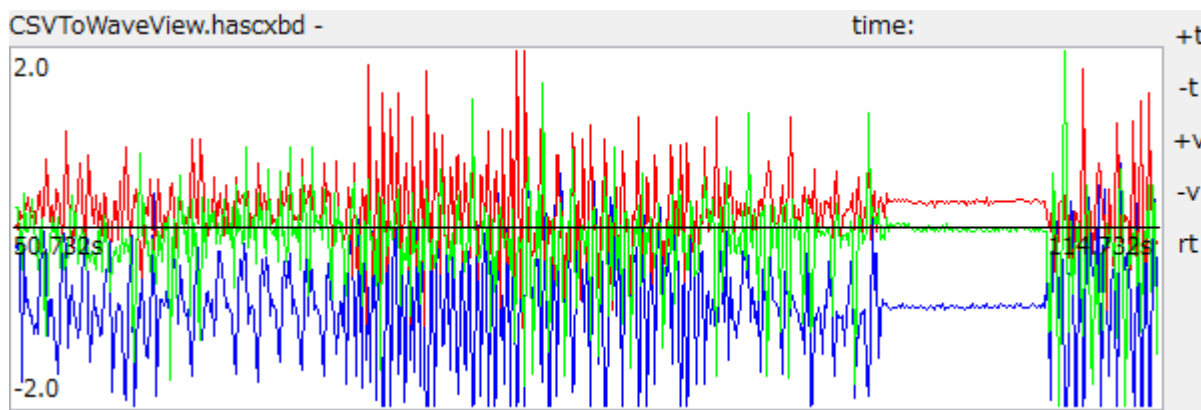
- csvToVectorとwaveViewの間にMeanFilterをつなげる
- 編集手順
  - コネクタ作成モードへ
  - 連結するブロックを順に選択
  - 右図のように連結すると警告がでるが、OKボタンを押す
  - セレクトモードに戻り余分なコネクタを消去



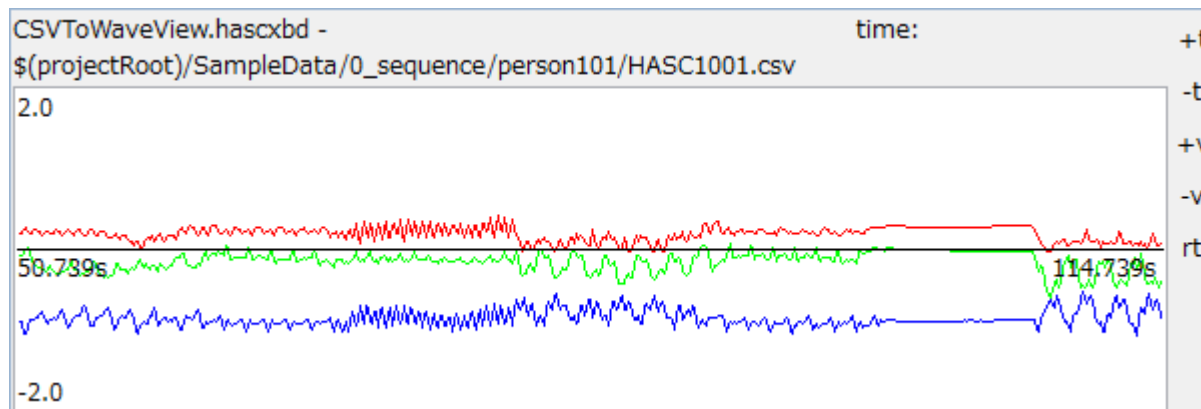
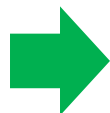
# 波形の変換

- 作成したブロックダイアグラムを保存
- HASCXBDファイルの実行

波形変換前



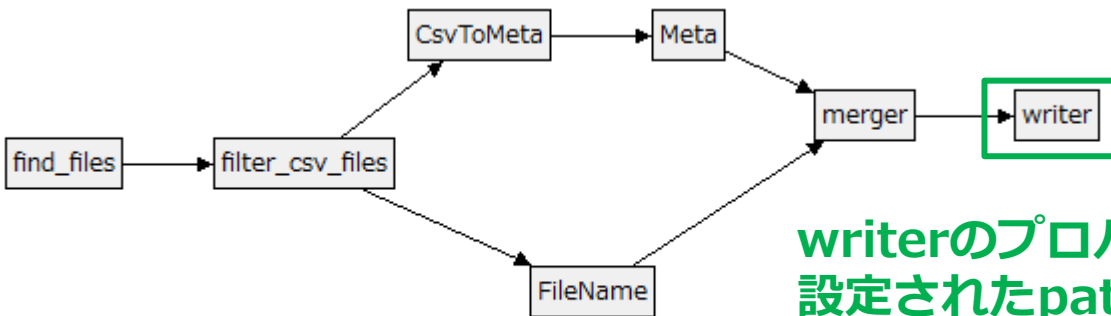
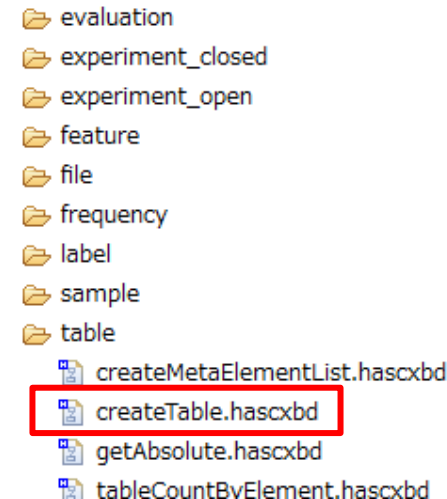
波形変換後



# SampleDataテーブルの作成

- HASCXBD>table>createTable.hascxbd

- 行動データファイル毎の計測環境が一覧できる



writerのプロパティで  
設定されたpathにテーブル  
ファイルが作成される

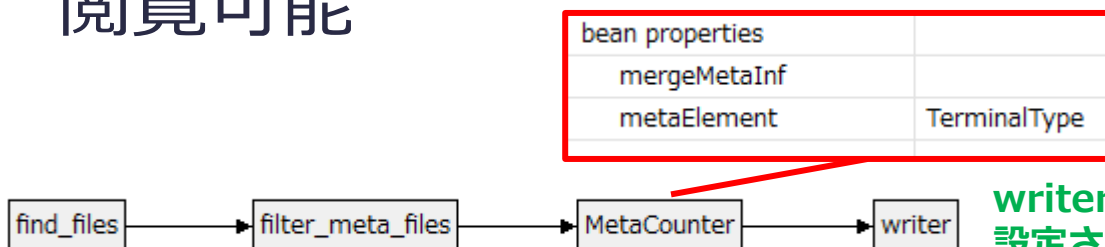
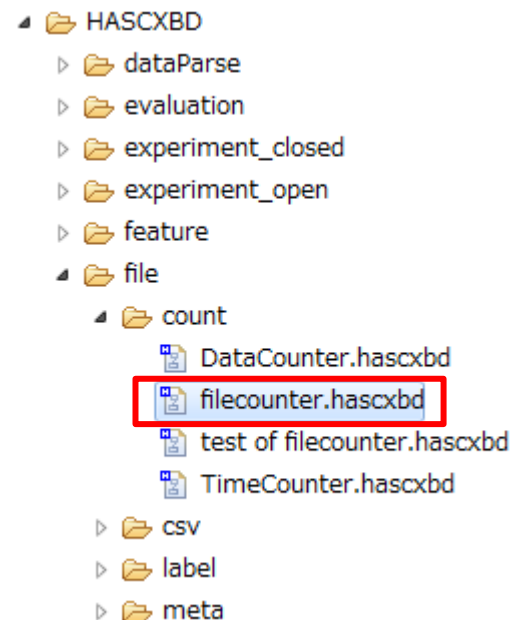
```
createTable.hascxbd | table_output.csv x
```

FilePath	FileName	Person	Activity	Activity	TerminalType	Frequency (Hz)	Gender	Height (cm)	Weight (kg)	Floor	Place	Shoes	TerminalMount	TerminalPos
\$(projectRoot)/SampleData/0_sequence/person101/HASC1001	HASC1001.csv	person101	0_sequence	sequence	Apple;iPod touch 3G	100	male	179	69	asphalt				
\$(projectRoot)/SampleData/0_sequence/person101/HASC1002	HASC1002.csv	person101	0_sequence	sequence	Apple;iPhone3Gs	100	male	179	69	asphalt	outc			
\$(projectRoot)/SampleData/0_sequence/person102/HASC1003	HASC1003.csv	person102	0_sequence	sequence	ATR;WAA-001	100	male	171	56	asphalt	outdoor			
\$(projectRoot)/SampleData/0_sequence/person102/HASC1004	HASC1004.csv	person102	0_sequence	sequence	Apple;iPod touch 3G	100	male	171	56	asphalt				
\$(projectRoot)/SampleData/0_sequence/person102/HASC1005	HASC1005.csv	person102	0_sequence	sequence	Apple;iPhone3Gs	100	male	171	56	asphalt	outc			
\$(projectRoot)/SampleData/0_sequence/person103/HASC1006	HASC1006.csv	person103	0_sequence	sequence	Apple;iPod touch 3G	100	male	170	54	asphalt				
\$(projectRoot)/SampleData/0_sequence/person103/HASC1007	HASC1007.csv	person103	0_sequence	sequence	Apple;iPhone3Gs	100	male	170	54	asphalt	outc			
\$(projectRoot)/SampleData/0_sequence/person104/HASC1008	HASC1008.csv	person104	0_sequence	sequence	Apple;iPod touch 3G	100	male	166	54	asphalt				
\$(projectRoot)/SampleData/0_sequence/person104/HASC1009	HASC1009.csv	person104	0_sequence	sequence	Apple;iPhone3Gs	100	male	166	54	asphalt	outc			
\$(projectRoot)/SampleData/0_sequence/person105/HASC1010	HASC1010.csv	person105	0_sequence	sequence	ATR;WAA-001	100	male	169	46	asphalt	outdoor			
\$(projectRoot)/SampleData/0_sequence/person105/HASC1011	HASC1011.csv	person105	0_sequence	sequence	Apple;iPod touch 3G	100	male	169	46	asphalt				
\$(projectRoot)/SampleData/0_sequence/person105/HASC1012	HASC1012.csv	person105	0_sequence	sequence	Apple;iPhone3Gs	100	male	169	46	asphalt	outc			
\$(projectRoot)/SampleData/0_sequence/person106/HASC1013	HASC1013.csv	person106	0_sequence	sequence	ATR;WAA-001	100	male	167	67	asphalt	outdoor			

# データファイル数のカウント

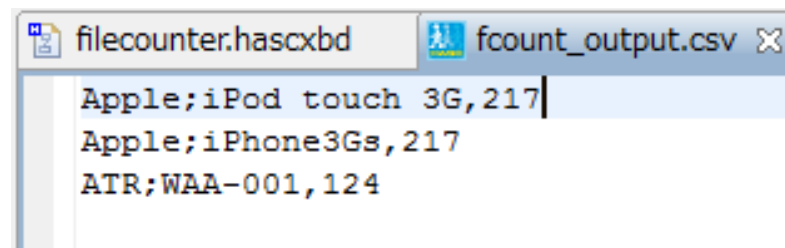
- HASCXBD>file>count>filecounter.hascxbd

- コーパス内に含まれる行動データファイル数をカウント
- 設定したメタ情報の分布も閲覧可能



writerのプロパティで設定されたpathにテーブルファイルが作成される

TerminalTypeに関するメタ情報のファイル数分布が閲覧できる



# HASCToolへリアルタイム送信

## HASC Logger側

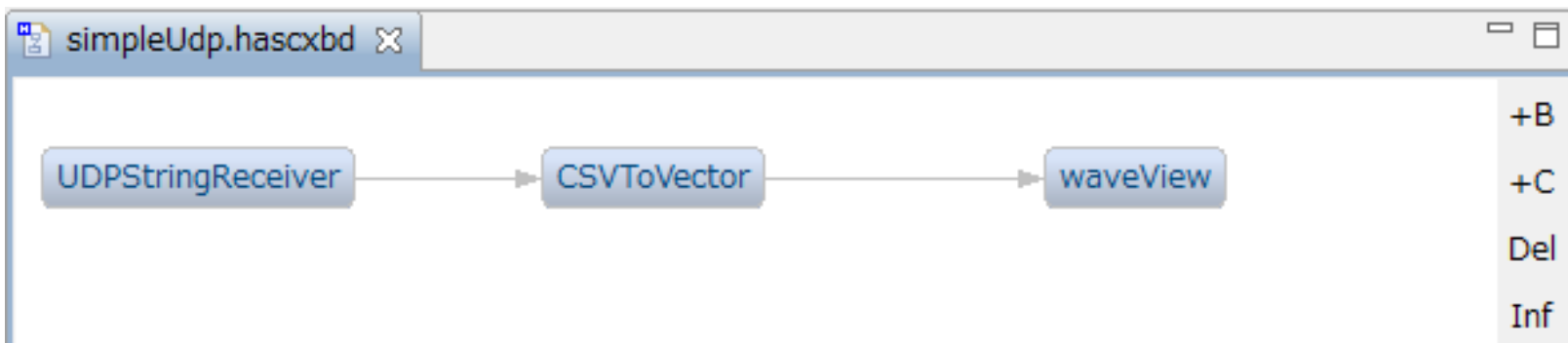
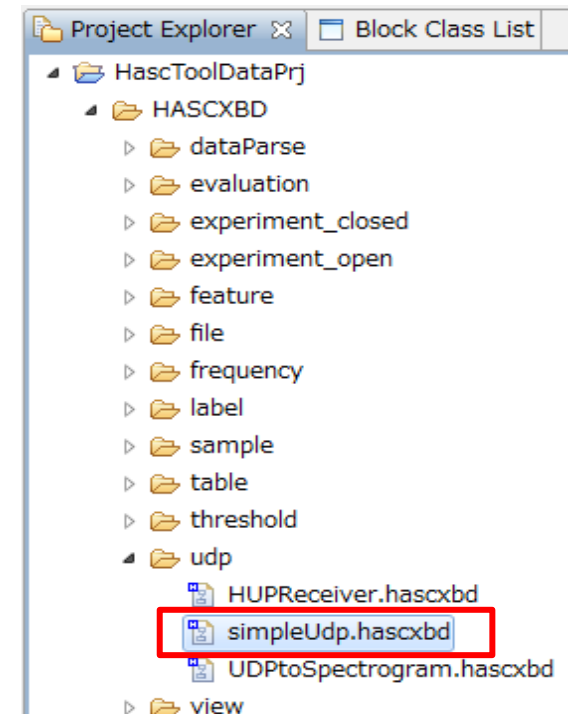
- iPhone版：設定アプリ> HASCLogger  
Android版：HASC Loggerを起動> Setting
- Send Logs as UDP Packet> IP:Port
  - HASC Toolを実行するPCのIPアドレス+":" +ポート番号6666
  - Ex)192.168.100.10:6666
- Start Sensing



# HASCToolへリアルタイム送信

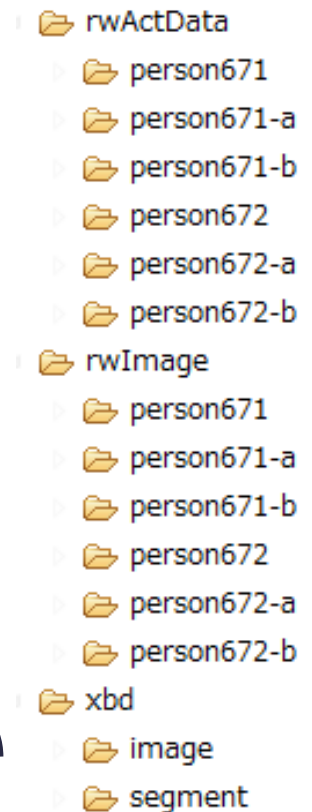
## HASC Tool側

- HASCXBD>udp>  
simpleUdp.hascxbd
  - BD実行
- UDPでHASCLoggerから  
行動データを受信
  - リアルタイムに波形を表示














# RealWorldActivityDataのインポート







- プロジェクトを新規作成
  - プロジェクト名：HASCrwPrj
- 作成したプロジェクト
  - 右クリック>インポート>一般>ファイルシステム
  - 参照元の選択（HASCrwdataを選択）
  - 出てきたすべての項目にチェック
  - アラートがでたら“OK”を押してください
  - 終了を選択
  - インポート完了



# 実環境サンプルデータセット

- rwActData
  - 2人分の実環境データ
  - -a,-bはラベル情報を変更
- rwImage
  - イメージ画像セット
- xbd
  - 処理スクリプト

- ▲  rwActData
  - ▷  person671
  - ▲  person671-a
    - ▲  hasc-111018-165936-acc
      - ▷  move;floor;1F
      - ▷  move;floor;2F
      - ▷  move;floor;4F
      - ▷  move;floor;B1F
      - ▶  move;indoor
      - ▷  move;outdoor
    - ▷  hasc-111018-165936-gyro

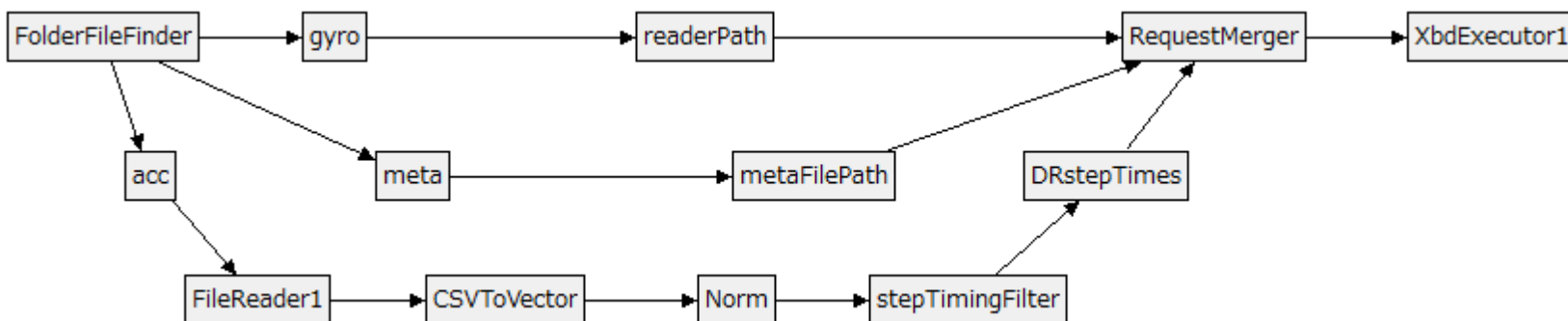
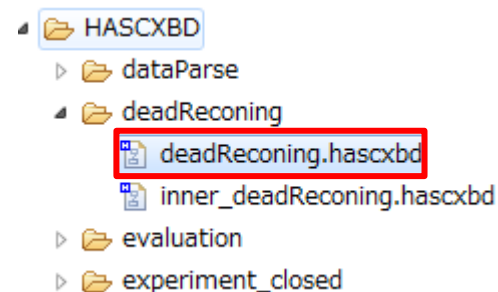
- ▲  hasc-111018-165936-acc
  - ▶  move;floor;1F
  - ▲  move;floor;2F
    -  seg1004.csv
    -  seg1004.label
  - ▷  move;floor;4F

# 処理スクリプト

- segment
  - label
    - xbdExe\_SegmentLabel.hascxbd
    - ラベルファイルをイベント情報で区切る
  - signal
    - xbdExe\_SegmentCSV.hascxbd
    - 行動データファイル(csvファイル)をイベント情報で区切る
- image
  - xbdExe\_imageLabelFile
  - 区切ったデータのイメージを作成

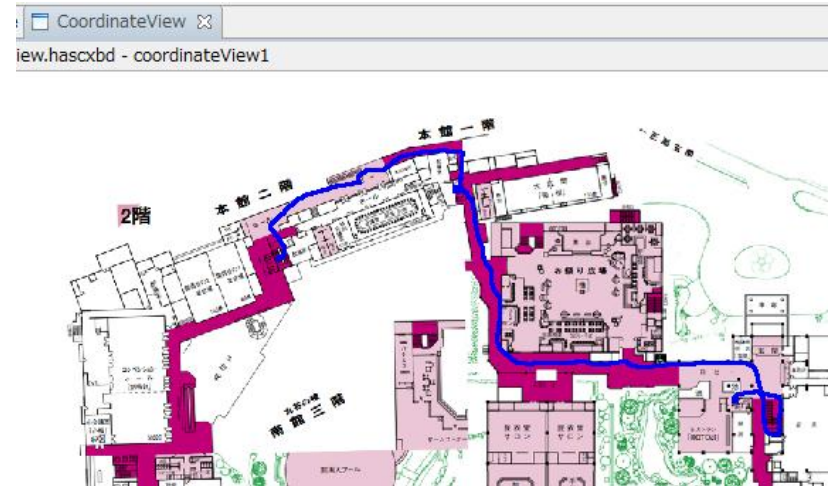
# 歩行者デッドレコニング

- デッドレコニング（自己位置推定）
  - センシング情報等から自分の位置を相対的に推定
- HASCXBD>deadReconing  
>deadReconing.hascxbd
  - 加速度情報から歩数を推定
  - 角速度・メタ情報から歩行経路を2次元座標列で出力



# 歩行経路のViewer

- 歩行経路の表示
- `view > coordinateView > coordinateView.hascxbd`
  - 出力されたファイルを選択



※画像ファイルが配布できないためsampleDataのみではviewは表示できません.