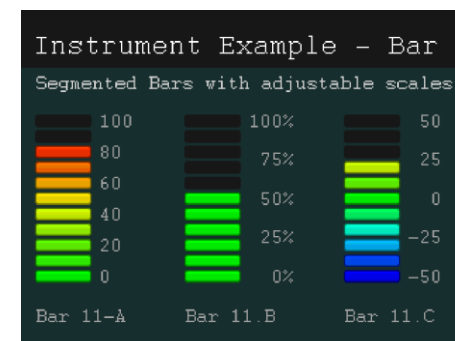


# PCAN-MiniDisplay Case 仕様 (1)

## □ CAN 2ch ディスプレイ / トレース / プレイバック

- CAN High-speed 2ch (40 kbit/s ~ 1 Mbit/s), CAN 2.0A/B
- アナログ入力 x 2: 0 ~ 16.5 V (10ビットAD, 0 = 0 V, 1023 = 16.5 V)
- デジタル入力 x 1: 10 kΩプルアップ (0: 1.8 V以下, 1: 6.6V 以上)
- プラスチックケース: 116.5 x 72.5 x 24.5 mm (W x D x H)
- TFTディスプレイ: 320 x 240 ピクセル (2.8 インチ相当)
- メモリ: マイクロSD 出荷時 512 MB (max. 32GB)
- USB2.0 x 1 (Mini-B)
- 電源: DC 7~30V (Typical: 135 mA@12V )
- 動作温度: -20~70°C
- IP保護クラス: IP20
- 重量: 138g



設定例: Instrument\_Bars



# PCAN-MiniDisplay Case 仕様 (2)

Slide 2

## □ 3つのモード(ディスプレイ/トレース/プレイバック)

- どれか1つのモードを実行 (**注意**: トレースしながらディスプレイ機能は動作しません)

### □ ディスプレイ

- ミニUSBを介してメモリ(マイクロSD)内に編集したファイルをコピー
  - テキストエディタでファイル編集: 設定例が付属 — 例. シグナルを変数(variable)で指定
  - 画像(ユーザ任意): ビットマップ(bmpファイル)作成(シグナル値だけの表示には bmp は不要)
- 複数のScene設定を設置可能 (表示中に実行する設定をボタン操作で切替)
- CANメッセージ送信
  - 3タイプ: Scene開始時、Scene終了時、Scene実行中に周期送信(各30メッセージまで)

### □ トレース(CANロウ・データのみ)

- トレースデータ: btr ファイル (PEAK-Systemバイナリフォーマット)
- フィルター設定可能: 通過させるCAN-IDの設定ファイルを指定 (例. 0x083, 0x7F0-0x7FE)
- コンバータ: PEAK-Converterでテキストベースのフォーマットに変換
  - csv、asc (Vectortレースフォーマット)、trc (PEAK-Systemトレースフォーマット)

### □ プレイバック

- トレースデータの送信 — ループ回数を指定可能: 1~9, 無限

# 外観(コネクタ等)

Slide 3

## 10ピンコネクタ

1. Vb (DC 7-30V)
2. GND
3. (Dout1 \*1)
4. Din4
5. Ain1
6. Ain2
7. CAN2\_L
8. CAN2\_H
9. CAN1\_L
10. CAN1\_H

\*1 Dout1はサポートされていません。  
(2019.09.18現在 FW V1.3.2)

プッシュボタン [Up]

プッシュボタン [Down]

ミニUSB (Mini-B)

## 4ピンコネクタ

1. Din1: Down
2. Din2: Up
3. Din3: Enter
4. GND

プッシュボタン [Enter]

## マイクロSDカードの交換

1. 電源が入っていないことを確認します。
2. プラスチックキャップ(4個)を外します。
3. ネジ(4個)を外します。
4. ケースを開けます。
5. マイクロSDスロットがここにあります。

PCAN-MiniDisplay

IPEH-002262-KSM01 10066

IPEH-002262-KSM01 10066

型番

シリアルNo.

# 設定 / 使用

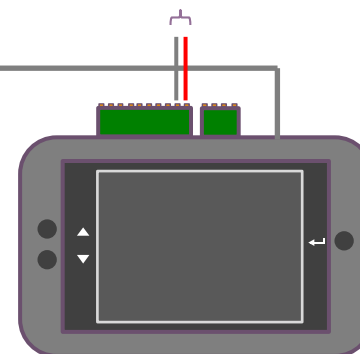
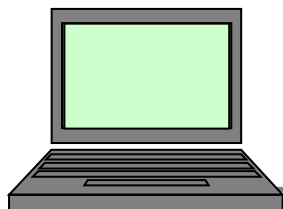
Slide 4

## □ 設定時

PC (Windows 10, 8.1)

ミニUSBケーブル

電源 DC 7~30V



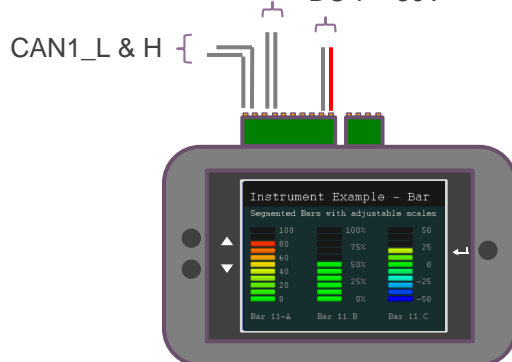
PCAN-MiniDisplay Case

- ・insファイルを編集、画像はbmpファイルを作成
- ・ミニUSBケーブルを接続すると、大容量デバイスとして検出
- ・新規フォルダ作成、ins / fon / bmp ファイルをコピー

**注意:** PC、ミニUSBケーブル、線材 はお客様でご用意ください。

## □ 使用時

CAN2\_L & H  
CAN1\_L & H { 電源 DC 7~30V



最低限必要な接続は、電源(Vb, GND)とCAN(CAN\_L, CAN\_H)です。  
アナログ入力(Ain1, Ain2,)、デジタル入力(Din4) は必要に応じて接続してください。

# 画像サンプル

Slide 5



Instrument\_Bars



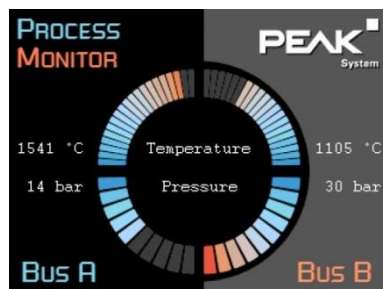
Instrument\_LEDs



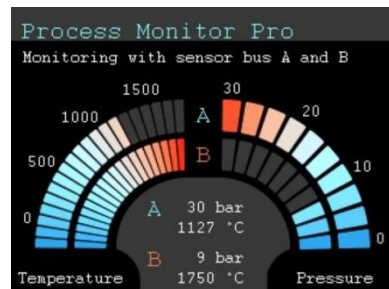
Instrument\_Tacho\_A-240



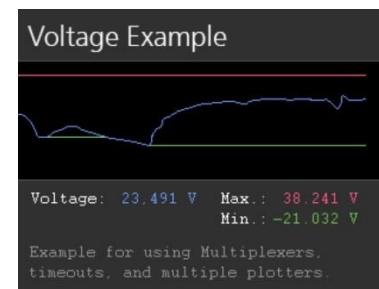
Instrument\_Tacho\_A-120



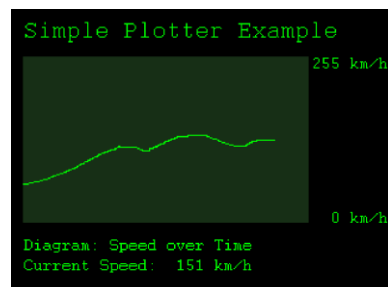
Process-Monitor



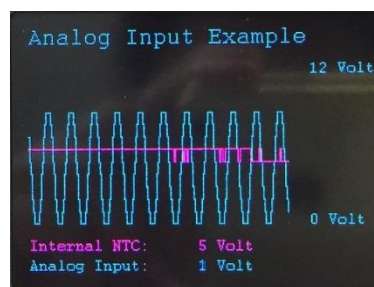
Process-Monitor-Pro



Voltage



Simple-Plotter



Analog Input

Voltage、Simple-Plotter、Analog Inputは、plotter指定で線を描いています(他のサンプルのような、多くのbmpファイルを、値によって差し替えての表示ではありません)。

Simple-PlotterとAnalog inputは、insファイルの設定だけです。Bmpファイルは使用していません。

# 関連ファイル

Slide 6

- 下記のURLからダウンロードし、PCAN-MiniDisplay.zipを解凍

<https://www.peak-system.com/quick/DLP2262>

PCAN-MiniDisplayディレクトリ下

- Documentation: ユーザーマニュアル
- Filters: フィルターサンプル
- Help: ヘルプファイル
- Resources: リソース
  - サンプルのファイル、PCAN-Explorer(PEAK-System社製CANアナライザ)プロジェクトファイル
- Scenes: 設定サンプル
  - 最新の設定サンプルがあり、このファイルを参照して編集にかかります
- Traces: トレースデータ格納場所
- Tools: PEAK-Converter(コンバータ)



# メインメニュー

Slide 7

- Device Settings
  - CAN 2ch ビットレート、リッスンオンリ、フィルターファイル指定
  - Beeper, Display rotation, Screensaver Timeout, Date & Time etc.
- Startup Options
  - Startup scene, Select scene, Start tracer, Tracer Timeout
- Show Scene
  - ディスプレイ設定(各Sceneのins指定で実行)
- Trace Messages
  - トレース設定、実行
- Play Back Trace
  - プレイバックのためのトレースファイル設定、実行
- Internal Statistics
  - 情報:ファームウェアVer., 日時, コンパイルデータ
  - Update Firmware, Factory Default, Bootloader

```
MiniDisplay V1.3.2          TR act  TR act
-----
Device Settings
Startup Options
Show Scene
Trace Messages
Play Back Trace
Internal Statistics
Memory Card
Help
Power Off
```

# メニュー Device Settings, Startup Options

Slide 8

## □ Device Settings

```
Device settings          TR act  TR act
-----
Ch.1 Detect CAN bitrate: Start
    CAN bitrate.....: 500k
    Listen-only mode..: Off
    Filter.....: none
Ch.2 Detect CAN bitrate: Start
    CAN bitrate.....: 33.3k
    Listen-only mode..: Off
    Transceiver mode..: normal
    Filter.....: None
User CAN bitrate.....: Edit
Beeper.....: Off
Screensaver Timeout....: never
Show startscreen.....: on
Date & time.....: Set
Reset file index (2) ..: Reset
                                ↓
OK Save&OK Cancel Help
-----
```

## □ Startup Options

```
Startup                  TR act  TR act
-----
Start scene...: ☒
Select scene...: .. Instrument_LEDs.ins
Start Tracer...: ☐
Tracer timeout: Never
OK Exit Help
```



# メニュー Show Scene, Trace Messages

Slide 9

## □ Show Scene

```
Scene                                TR act  TR act
-----
Minidisplay/Sence
..
Transmit-Lists
Voltage
All-In-One
Analog Input
Default
Font_Example
Instrument_Bars
Instrument_LEDs
Instrument_Tachos
Process-Monitor
Process-Monitor-Pro
Simple-Plotter

438 MB free of total 470 MB
Exit
```

## □ Trace Messages

```
Trace                                TR act  TR act
-----
File: trc00002.btr

Trace CAN1 : Data+Err
Trace CAN2 : Off
Logfile size: 10 Mbyte
Tracer timeout: Never
Exit Start Help
```

# メニュー Internal Statistics, Memory Card, Help

Slide 10

## □ Internal Statistics

```
Internal Statistics      TR act  TR act
-----
MiniDisplay V1.3.2
Date/Time: 2018-06-20 15:35
SerNo: IPEH002262-100661
Compiledate: 2018-05-24
Heap:0xfe0000,used(max):0x258(0x1a3818)
Userstack Size:0x1800, used:0x0b7c
IRQ-Stack Size:0x0140, used:0x0014
FIQ-Stack Size:0x0020, used:0x0000
Exit Update Firmware Factory Defaults
Bootloader
```

## □ Help

```
MiniDisplay V1.3.2      TR act  TR act
-----
Device Settings
Startup Options
Status line on top
Shows for each CAN channel:
- TR for CAN transmission/reception
  * blinking during CAN traffic
  * yellow or red in case of errors
  * green otherwise
- Bus status: active, passive, bus-off
- L for activated listen-only mode

Device Settings
Press button to exit
```

## □ Memory Card

```
Mem Card      TR act  TR act
-----
Show Directory
View Bitmap
Help
Exit
```

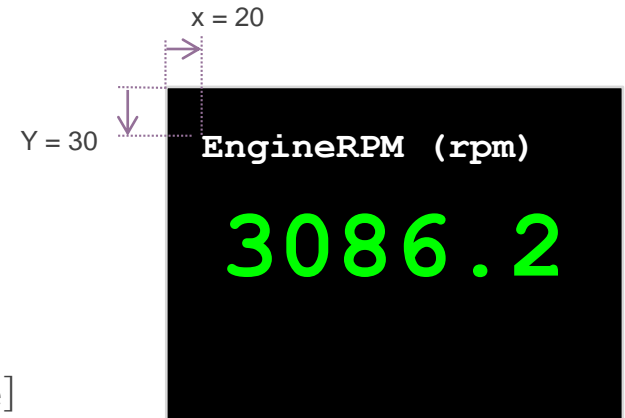
# シンプルScene [Value\_RPM] (1)

Slide 11

## □ シグナル(EngineRPM)情報

- High-speed CAN 500 kbit/s
- CAN-ID: 0x083 [message - canid]
- CAN 2.0A(11ビットCAN-ID) [message - frametype]
- 変数名: EngineRPM [variable - name]
- スタートビット, レングス: 0, 16 [variable - position]
- フォーマット(バイト順): Intel [variable - byteorder]
- ファクタ: 0.1 [variable - scale]
- オフセット: 0 [variable - offset]
- 変数データ: unsigned(符号なし整数) [variable - vartype]
- 表示データ: float(小数点以下がある場合) [variable - datatype]

Label(ラベル)で表示設定  
label - position=20,30



EngineRPMという1個のシグナルを表示するために必要な記述が、「シンプルScene [Value\_RPM] (2) & (3)」です。  
このファイル (Value\_RPM.insとする) と必要な2個のフォントファイルをPCAN-MiniDisplay CaseのScenesフォルダ下に新しいフォルダ (例. Value\_RPM) を作成して入れます。

### MiniDisplay¥Scenes¥Value\_RPM

- Value\_RPM.ins
- courier\_13\_18\_bold.fon
- courier\_40\_48.fon

# シンプルScene [Value\_RPM] (2)

Slide 12

Value\_RPM.ins

```
[global] // ===== Global Settings (グローバル設定) =====
format=1          // 変更しないでください
compress=1        // 1: 最初のロード後に圧縮。2回目以降は早く起動される。
messages=1        // CANメッセージ数 (max 96)
variables=1        // 変数 (シグナル) の数 (max 96)
labels=2          // ラベル数 (max 64)
fonts=2           // フォント数 (max 27)

[message1] // ===== Messages (メッセージ) =====
enable=1          // 0: メッセージ処理 ディセーブル, 1: メッセージ処理 イネーブル
name="EngineInfo" // メッセージ名 (任意の名前)
canid=0x083       // CAN-ID (例. Hex 16進数 0x083, Decimal 10進数 131)
frametype=0       // 0: 11 ビット, 1: 29 ビット
channel=0         // 0: CANチャンネル 1, 1: CANチャンネル 2

[variable1] // ===== Variables (変数 = シグナル) =====
name="EngineRPM"  // 変数名 (シグナル名)
messagename="EngineInfo" // メッセージ指定
position=0,16     // スタートビット, レングス (ビット数)
byteorder=0       // 0: Intel, 1: Motorola
muxtype=0         // 0: None, 1: Multiplexer, 2: Multiplexed
scale=0.1          // スケール (y = ax ± b の a)
offset=0           // オフセット (y = ax ± b の b) , y = 0.1 x ± 0
vartype=0         // 変数データ -- 0: unsigned, 1: signed, 2: float
datatype=2        // 表示データ -- 0: unsigned, 1: signed, 2: float
```

# シンプルScene [Value\_RPM] (3)

Slide 13

```
[font1] // ===== Fonts (フォント) =====
font="courier_13_18_bold.fon" // フォントファイル名
name="DemoFont"             // フォント定義 (任意の名前)
type=0                      // 変更しないでください

[font2]
font="courier_40_48.fon"     // フォントファイル名
name="DemoFont2"            // フォント定義 (任意の名前)
type=0                      // 変更しないでください

[label1] // ===== Labels (ラベル) TFT display x: 320, y: 240 =====
name="label1"               // ラベル名 (max 15文字)
position=20,30              // 左上から x 位置 0~319, y 位置 0~239 ピクセル
font_name="DemoFont"        // フォント指定
length=15                   // 文字数
initval="EngineRPM (rpm)"    // 初期値 (max 39文字)
fontcolor=255,255,255        // 文字色 RGB指定 白色
bgcolor=0,0,0               // 背景色 RGB指定 黒色

[label2]
name="label2"               // ラベル名 (max 15文字)
position=40,90              // 左上から x 位置 0~319, y 位置 0~239 ピクセル
font_name="DemoFont2"        // フォント指定
fmt_string="%04.1f"          // フォーマット定義 (printf参照)
var_name="EngineRPM"         // 変数指定
length=6                    // 文字数
fontcolor=0,255,0            // 文字色 RGB指定 緑色
bgcolor=0,0,0               // 背景色 RGB指定 黒色
```

# グラフィカルScene [Inst\_Bar] (1)

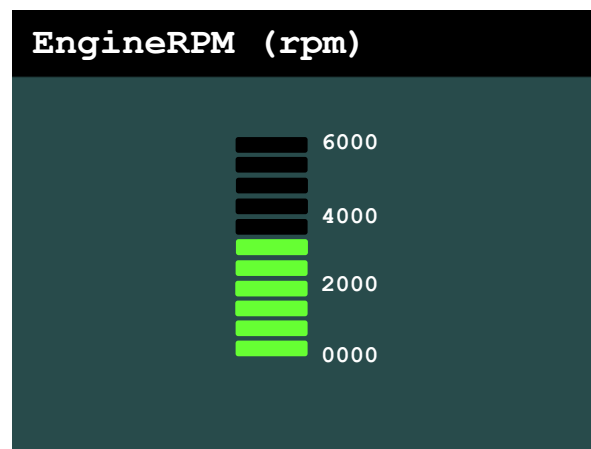
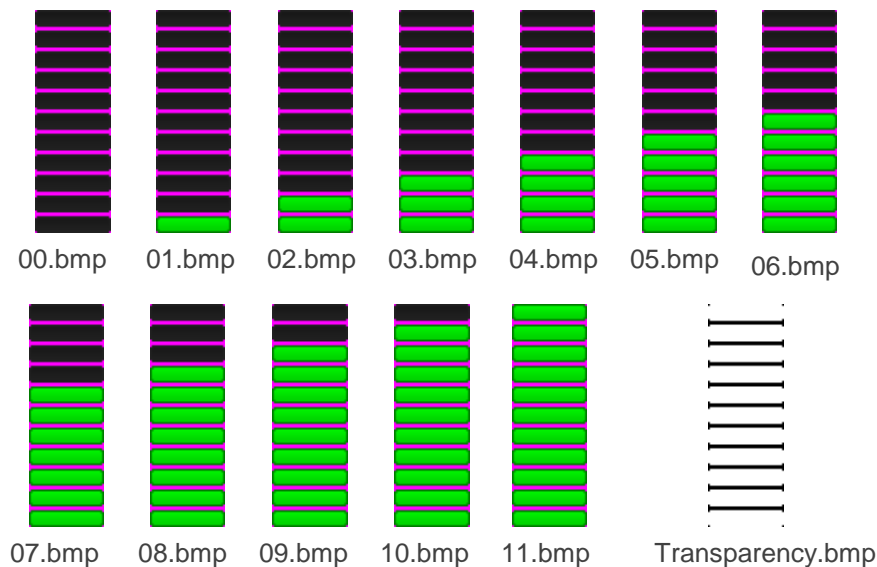
Slide 14

## グラフィカル表示

Inst\_Bar.insと14個のbmpファイルでEngineRPMという1個のシグナルを棒グラフで表示します（右図参照）。

Inst\_Bar.insは「グラフィカルScene [Inst\_Bar] (2) , (3), (4)」を参照してください。

このSceneでは、font\_idxで定義しているので、フォントファイルを配置する必要はありません。



Background.bmp

# グラフィカルScene [Inst\_Bar] (2)

Slide 15

Inst\_Bar.ins

```
[global] // ===== Global Settings (グローバル設定) =====
format=1 // 変更しないでください
compress=1 // 1: 最初のロード後に圧縮。2回目以降は早く起動される。
instruments=2 // インストルメンツ (ビットマップ構成数 max 96)
variables=1 // 変数 (シグナル) の数 (max 32)
labels=5 // ラベル数 (max 64)

[variable1] // ===== Variables (変数 = シグナル) =====
name="EngineRPM" // 変数名 (シグナル名)
canid=0x83 // CAN-ID
position=0,16 // スタートビット, レンクス (ビット数)
frametype=0 // 0: 11 ビット, 1: 29 ビット
byteorder=0 // 0: Intel, 1: Motorola
muxtype=0 // 0 => None, 1 => Multiplexor, 2 => Multiplexed
muxval=0 // muxtype=2で有効:このマルチプレクサ値の場合に変数を更新 (muxtype=0:影響なし)
scale=0.1 // スケール (y = ax ± b の a)
offset=0 // オフセット (y = ax ± b の b) , y = 0.1 x ± 0
vartype=0 // 変数データ -- 0: unsigned, 1: signed, 2: float
datatype=0 // 表示データ -- 0: unsigned, 1: signed, 2: float

[label1] // ===== Labels (ラベル) TFT display x: 320, y: 240 =====
name="label1" // ラベル名 (max 15文字)
position=10,10 // 左上から x 位置 0~319, y 位置 0~239 ピクセル
font_idx=2 // フォント指定 (2: Courier_12_18)
length=16 // 文字数
initval="EngineRPM (rpm)" // 初期値 (max 39文字) 「EngineRPM (rpm)」表示
fontcolor=255,255,255 // 文字色 RGB指定 白色
bgcolor=0,0,0 // 背景色 RGB指定 黒色
```

# グラフィカルScene [Inst\_Bar] (3)

Slide 16

```
[label2]
name="label2"
position=161,183
font_idx=0           // フォント指定 (0: Courier_8_12)
length=4
initval="0000"       // 初期値 (max 39文字) 「0000」表示
fontcolor=255,255,255 // 文字色 RGB指定 白色
bgcolor=40,75,75     // 背景色 RGB指定 濃灰色

[label3]
name="label3"
position=161,146
font_idx=0
length=4
initval="2000"       // 初期値 (max 39文字) 「2000」表示
fontcolor=255,255,255
bgcolor=40,75,75

[label4]
name="label4"
position=161,109
font_idx=0
length=4
initval="4000"       // 初期値 (max 39文字) 「4000」表示
fontcolor=255,255,255
bgcolor=40,75,75

[label5]
name="label5"
position=161,72
font_idx=0
length=4
initval="6000"       // 初期値 (max 39文字) 「6000」表示
fontcolor=255,255,255
bgcolor=40,75,75
```



# グラフィカルScene [Inst\_Bar] (4)

Slide 17

```
[instrument1] // ===== Instruments (インスツルメンツ) =====
pictures=1           // ビットマップファイル数
name="Background"    // インスツルメンツ名
no_restore=1         // バックグラウンド扱い
0="Background.bmp","",0,0,0,0 // 番号=bmpファイル名,透過ファイル名,x位置,y位置,値下限,値上限

[instrument2]
name="EngineRPM"      // インスツルメンツ名
pictures=12           // ビットマップファイル数
no_restore=1         // バックグラウンド扱い
transparency=1        // 透過ファイル数
offset=0,0            // x,y 位置のオフセット
var_name="EngineRPM"  // 変数指定
initval=0             // 初期値
0="00.bmp","Transparency.bmp",116,74,0,0 // 番号=bmpファイル名,透過ファイル名,x位置,y位置,値下限,値上限
1="01.bmp","Transparency.bmp",116,74,1,499 // EngineRPM が 1~499 rpm のときに 01.bmp を表示
2="02.bmp","Transparency.bmp",116,74,500,999
3="03.bmp","Transparency.bmp",116,74,1000,1499
4="04.bmp","Transparency.bmp",116,74,1500,1999
5="05.bmp","Transparency.bmp",116,74,2000,2499
6="06.bmp","Transparency.bmp",116,74,2500,2999
7="07.bmp","Transparency.bmp",116,74,3000,3499
8="08.bmp","Transparency.bmp",116,74,3500,3999
9="09.bmp","Transparency.bmp",116,74,4000,4499
10="10.bmp","Transparency.bmp",116,74,4500,4999
11="11.bmp","Transparency.bmp",116,74,5000,5499
```

# Scene切替

Slide 18

- inpファイルを作成
  - ▣ inpファイル: 開始シーン番号と複数のinsファイルを記述(例. Example.inp)
- 必要なファイルを1個のフォルダ下に配置
  - ▣ 複数のScene(ins / fon / bmp ファイル)とinpファイルを1個のフォルダ下に配置
- inpファイルを実行
- プッシュボタン [Up] / [Down] 押下

Example.inp

```
[global] // ===== グローバル設定 =====  
version=1           // 変更しないでください  
  
[scenes] // ===== シーン =====  
scenes=4            // Scene (シーン) 総数  
startscene=1        // 開始シーン番号 (1: Value_RPM.ins)  
1="Value_RPM.ins"    // 値表示 - エンジン回転数  
2="Inst_Bar.ins"      // 棒グラフ - エンジン回転数  
3="Inst_Tacho.ins"    // タコメータ  
4="Value.ins"         // 値表示
```

