

NA2202/2203/2204

評価ソフトウェア ユーザマニュアル

1. 使用許諾、免責事項	2
2. サードパーティライブラリ	2
3. システム要件	3
4. インストール	3
4.1. ソフトウェアのインストール	3
4.2. ドライバのインストール	3
5. メインウィンドウ	4
6. USB デバイスの認識	5
7. レジスタ設定	6
7.1. 設定の保存と復元	6
7.2. その他の設定	7
8. モニタウィンドウ	8
8.1. 動作モード	8
8.1.1. 連続動作モード	9
8.1.2. 間欠動作モード	11
8.2. ズーム機能	12
8.3. ラベル表示	13
8.4. データ保存	13
8.5. ADC 変換コードへの演算処理	14
8.5.1. 定義済み関数, 定数	15
入力電圧換算のサンプル	17
9. ヒストグラムウィンドウ	18
9.1. ADC 変換データ取得間隔	19
9.2. 結果の保存と読み込み	20

1. 使用許諾、免責事項

本ソフトウェアは現状有姿で提供され、使用による損害に対して当社は一切の責任を負いません。

また、このソフトウェアは新日本無線の製品を評価する目的に限り使用できます。

本ソフトウェアの全ての権利は新日本無線に帰属し、再配布、改変、およびリバースエンジニアリングを禁じます。

2. サードパーティライブラリ

本ソフトウェアは“ChakraCore” (<https://github.com/Microsoft/ChakraCore>) を使用しています。

ChakraCore のライセンスについては [ChakraCore.LICENSE.txt](#) を参照してください。

3. システム要件

- ・ サポート OS: Windows10
- ・ Microsoft .NET Framework 4 Client Profile

4. インストール

4.1. ソフトウェアのインストール

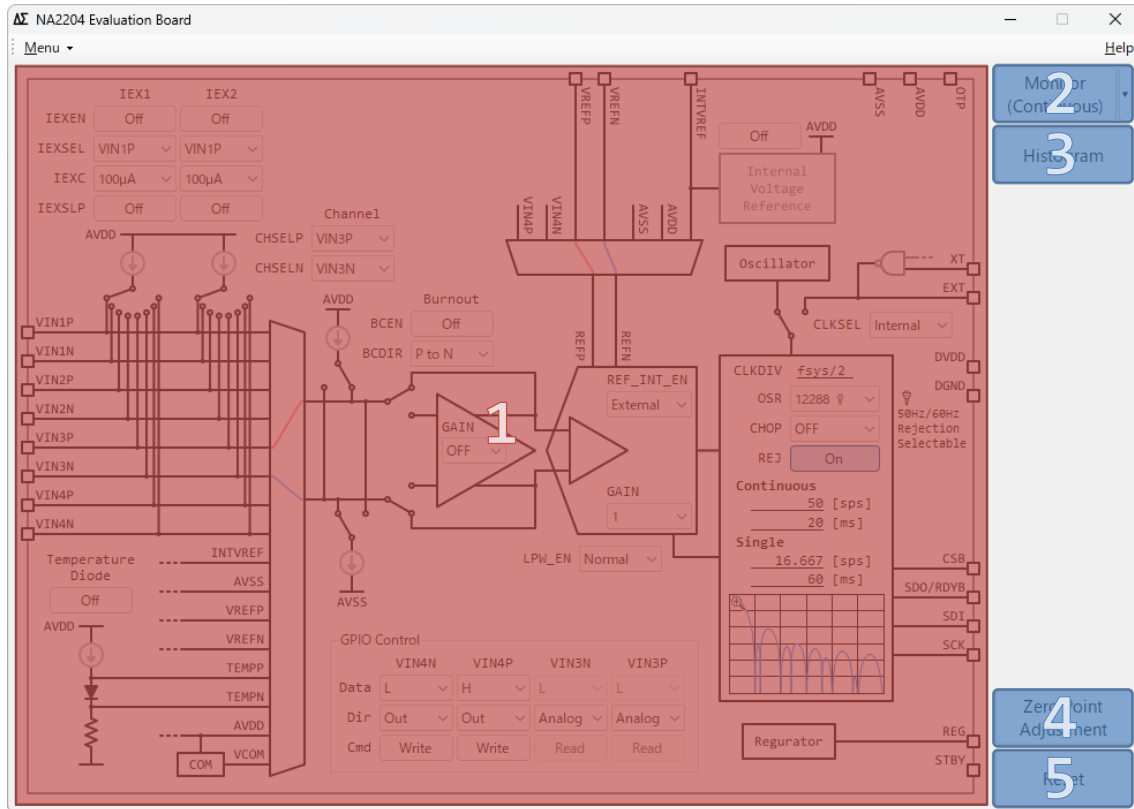
zip アーカイブを展開し、全てのファイルを同一のディレクトリに設置してください。

4.2. ドライバのインストール

親ボードとして STMicroelectronics 社のマイコンボード NUCLEO-F411RE を使用します。このマイコンボードと通信を行うため、ドライバのインストールが必要です。ドライバのインストールについては「[DriverInstallation_ja_note.pdf](#)」を参照してください。

5. メインウィンドウ

Fig. 1 メインウィンドウ



1. デバイスのレジスタ設定を行います
2. モニタウィンドウを表示します
3. ヒストグラムウィンドウを表示します
4. ゼロ点調整を実行します
5. デバイスをリセットします

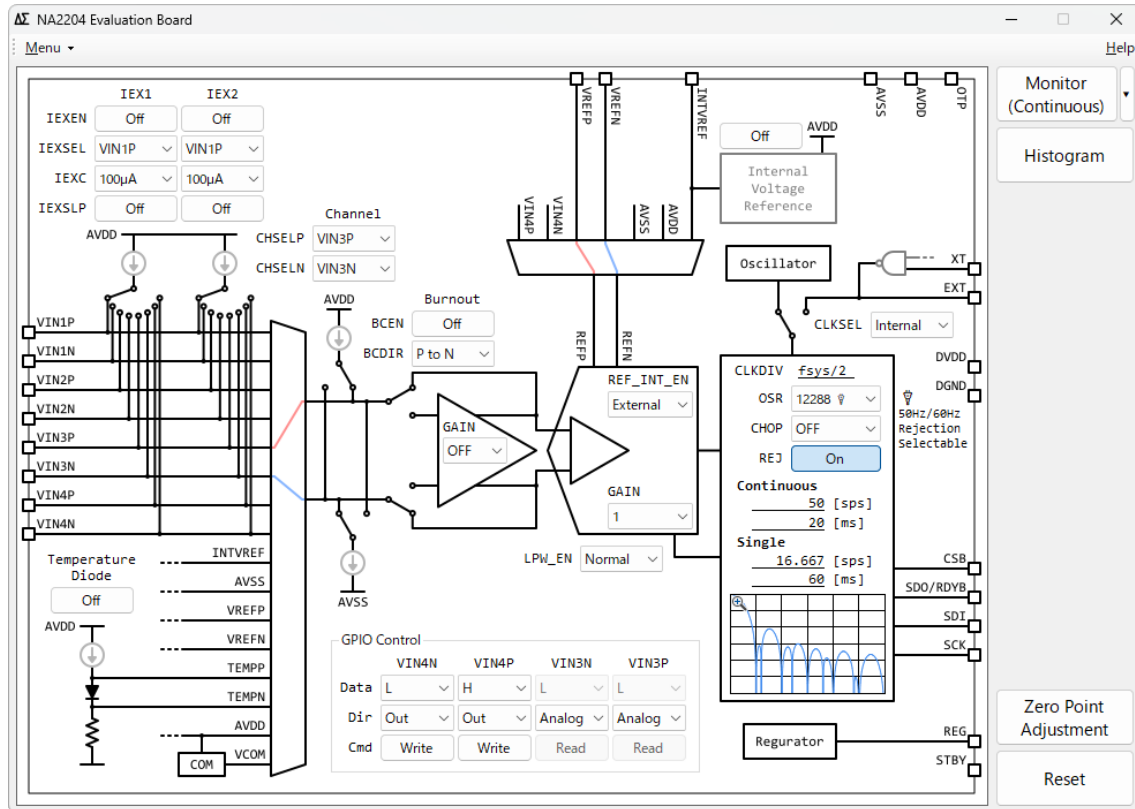
6.USB デバイスの認識

ソフトウェア起動前に評価ボードと PC を USB ケーブルで接続してください。

ソフトウェア起動時に USB デバイスの認識が自動的行われます。

7. レジスタ設定

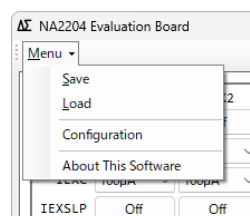
Fig. 2 レジスタ設定



7.1. 設定の保存と復元

ツールバーのメニューから、現在の設定状態を設定ファイルに保存、設定ファイルから設定状態を復元ができます。

Fig. 3 設定の保存・復元メニュー



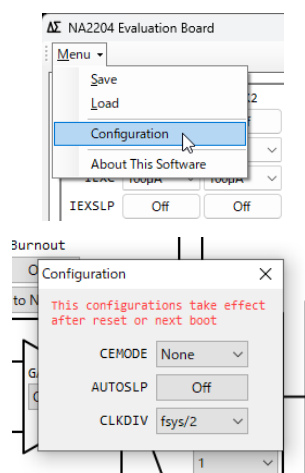
7.2. その他の設定

その他の設定は

Menu > Configuration

から設定可能です。通常はデフォルト設定のまま使用してください。
ここで設定した項目はリセット後または次回起動時に反映されます。
設定はユーザーフォルダに保存されます。

Fig. 4 その他の設定

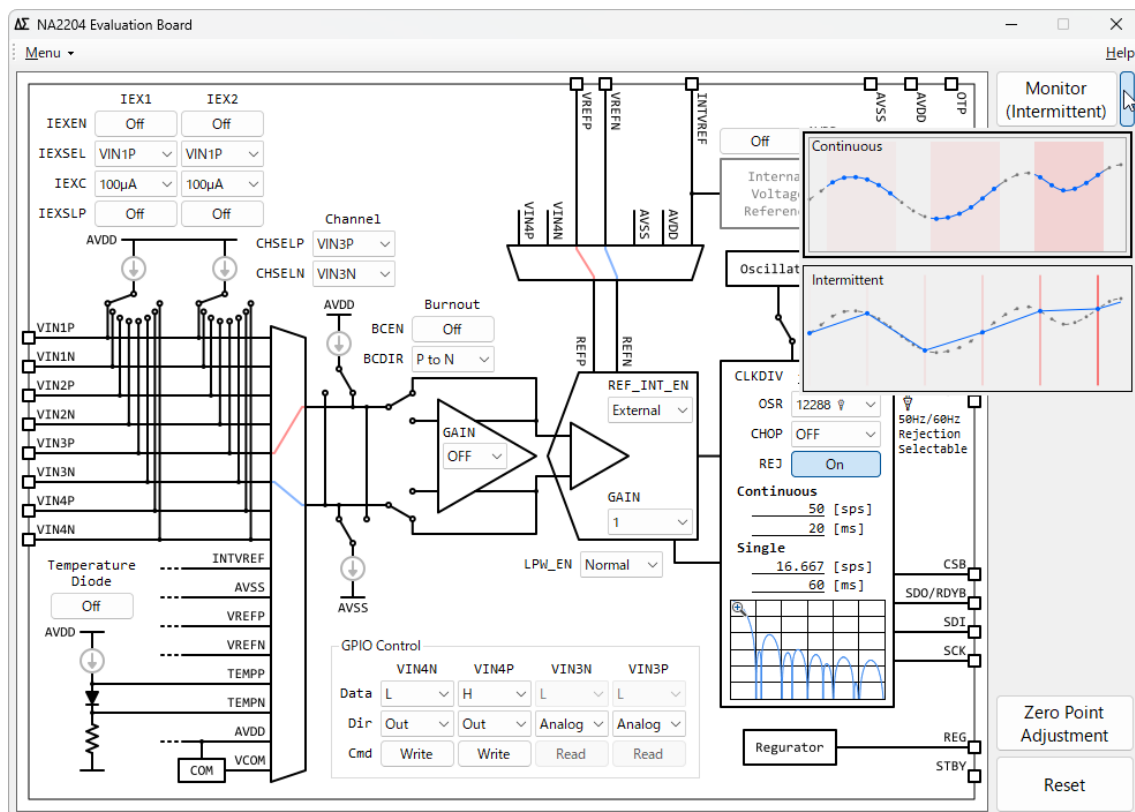


8. モニタウィンドウ

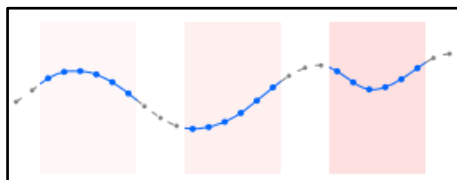
8.1. 動作モード

Monitor ボタン脇のプルダウンメニューから 2 つの動作モードを選択できます。

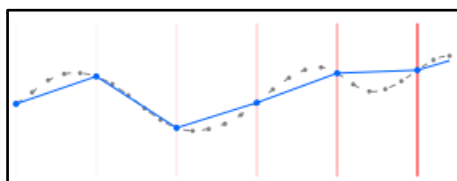
Fig. 5 モニタ動作モード選択



連続動作モード

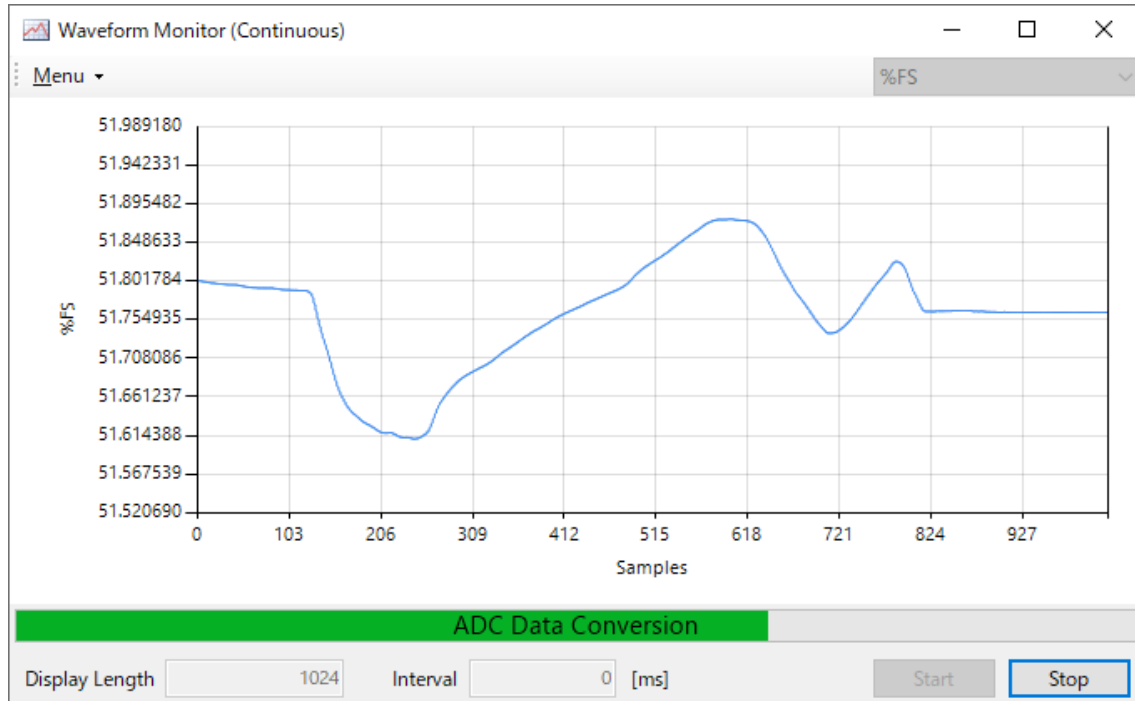


間欠動作モード

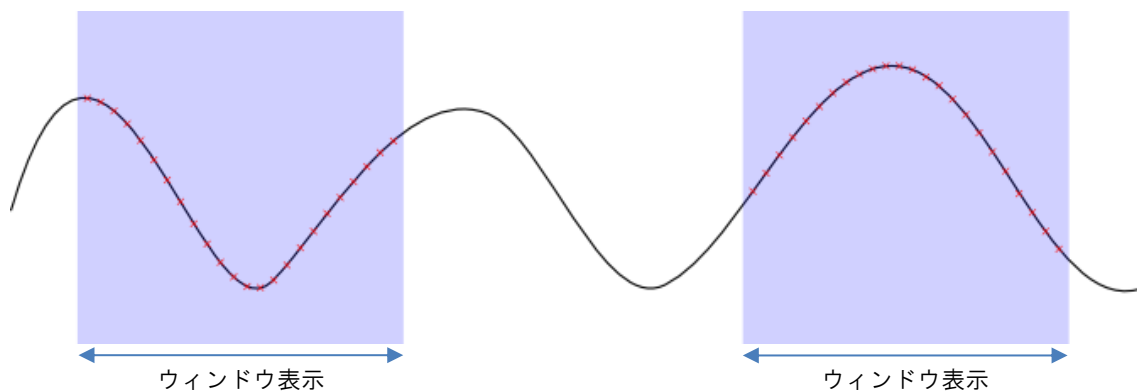


8.1.1. 連続動作モード

Fig. 6 連続動作モード



「Display Length」で設定された数の連続した ADC データを取得し、モニタウィンドウに表示します。表示の更新は設定数分の ADC データ取得時に一括で行われます。データ表示後、次のモニタ動作を開始します。



従ってモニタウィンドウに表示されている波形は連続した ADC 変換データですが、ウィンドウの更新間のデータは表示されません。

8.1.1.1. ADC 変換データ取得間隔

ADC 変換データを取得する時間間隔は「Interval」で設定します。「Interval」に 0 を設定した場合と、0 を超える値を設定した場合で動作が異なります。

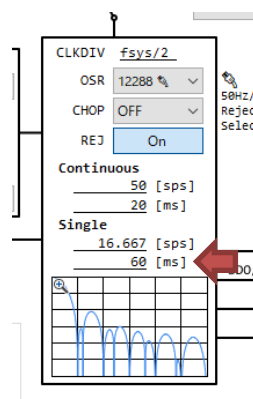
Interval == 0

デバイスは「連続変換モード」で動作し、OSR, CLKDIV, CHOP の有無で定まる最も短い時間間隔で ADC 変換データを取得します。

Interval > 0

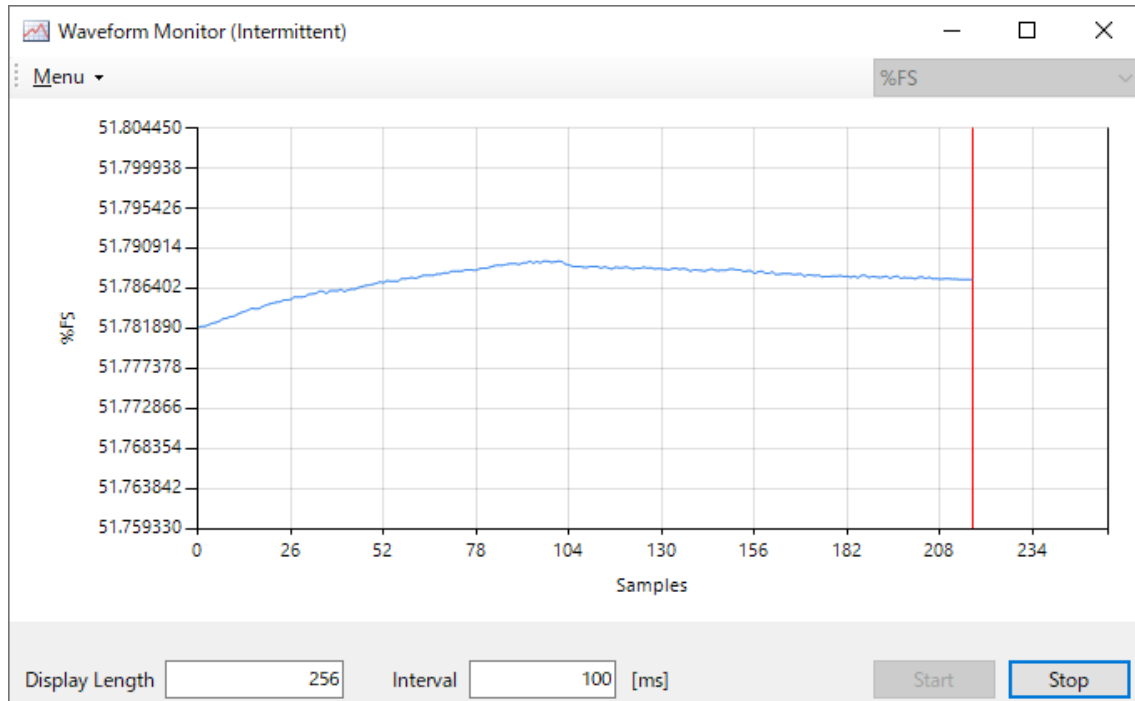
デバイスは「シングル変換モード」で動作し、Interval で設定された時間間隔毎に ADC 変換データを取得します。そのため、Interval には 1 回の ADC 変換データ取得にかかる時間よりも大きな値を設定する必要があります。デバイス設定ウィンドウの変換レート表示部分に表示されているシングル変換の変換時間の 2 倍程度の値を目安に Interval の値を設定してください。

Fig. 7 ADC 変換時間

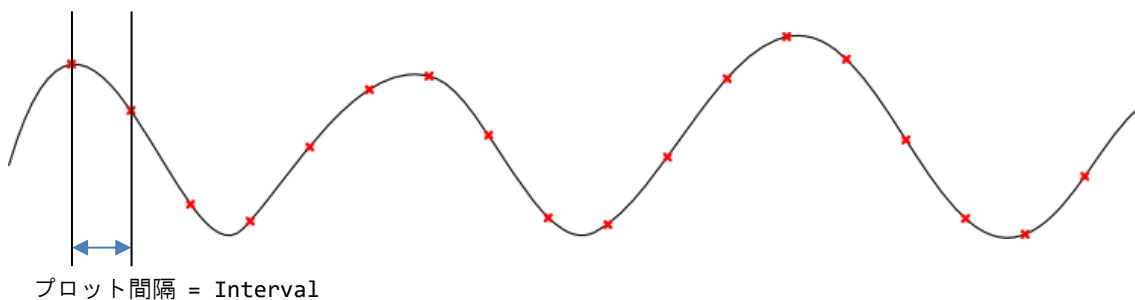


8.1.2. 間欠動作モード

Fig. 8 間欠動作モード



「Interval」で設定された時間間隔で ADC 変換データを取得し、モニタウィンドウに逐次プロットします。「Display Length」で指定された数の ADC 変換データを取得し終わるとプロット位置は原点に戻り、表示を上書きしながら ADC 変換データの取得を続けます。



「Interval」の値は最小でも 100[ms]程度に設定してください。ご使用の環境によりますが Interval を小さくし過ぎると、評価ボードとの通信が間に合わず正しい時間間隔でデータを取得できません。

また、データ取得の時間間隔は Interval 設定に対し数 10[ms] の誤差を含む場合があります。

8.2. ズーム機能

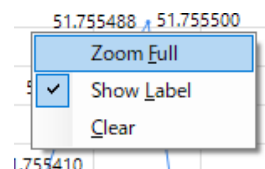
波形表示領域をマウスでドラッグすることで表示範囲を拡大できます。

Fig. 9 ズーム機能



ズーム表示を解除するには、波形表示領域の右クリックメニューから「Zoom Full」を選択してください。

Fig. 10 ズーム表示の解除



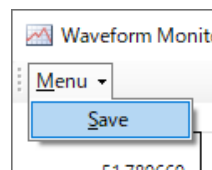
8.3. ラベル表示

プロットされたポイント上に値を表示する機能です。この機能は処理が重いため、モニタ動作が停止状態、且つ、表示されているポイント数が 300 以下の場合にのみ有効にできます。表示されているポイント数が多く、ラベルが表示できない場合はズーム機能を使用して表示されるポイント数を減らした上でラベル表示を有効にしてください。

8.4. データ保存

モニタ結果を CSV 形式で外部ファイルに保存できます。ツールバーのメニューから「Save」を選択してください。

Fig. 11 データ保存



尚、保存データの形式は「9.2.結果の保存と読み込み」のヒストグラムの Strip Chart の保存データと同じですので、「8.5.ADC 変換コードへの演算処理」による ADC コード以外への変換行われていないものに限る。波形モニタで保存したファイルをヒストグラムウィンドウで読み込み、ヒストグラム、統計データを表示することが可能です。

8.5. ADC 変換コードへの演算処理

取得した ADC 変換データに任意の演算処理を行う機能があります。この機能を使うことで例えば単位変換などを行うことが可能です。

演算処理を行う関数は JavaScript で定義され、以下のディレクトリにスクリプトファイルを追加する事でユーザーが任意の処理を施すことができます。

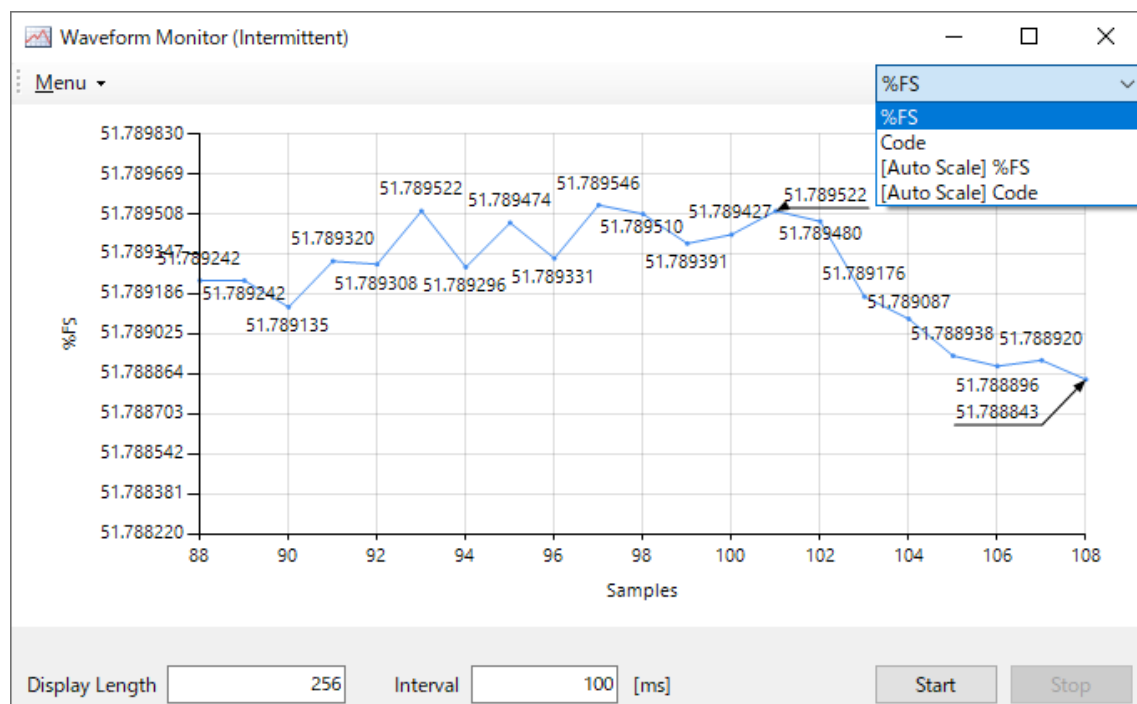
<プログラムファイルのディレクトリ>/js/<ファイル名>.js

スクリプトファイルを追加すると、モニタウィンドウ右上のリストに<ファイル名>の部分が表示されます。Fig. 12 は

- ・ <プログラムファイルのディレクトリ>/js/%FS.js
- ・ <プログラムファイルのディレクトリ>/js/Code.js
- ・ <プログラムファイルのディレクトリ>/js/[Auto Scale] %FS.js
- ・ <プログラムファイルのディレクトリ>/js/[Auto Scale] Code.js

のスクリプトファイルがある場合の例です。

Fig. 12 JavaScript の選択



8.5.1. 定義済み関数，定数

同一名の引数は各関数で同じ値なので、引数については「Table 2 引数」で説明します。

- `const CHIPID`

チップ固有の ID を表します。

「Table 1 定数」に各デバイスにおける CHIPID の値を示します。

Table 1 定数

デバイス	CHIPID
NA2202	0xA1
NA2203	0xA2
NA2204	0xA3

- `function data(adcData, pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C)`

ADC 変換データに演算処理を行います。

- `function min(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C)`

Y 軸の最小値を設定します。

- `function max(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C)`

Y 軸の最大値を設定します。

- `function unit(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C)`

Y 軸に表示するラベルを設定します。

- `function numberOfDecimals(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C)`

小数点第何位まで表示するかを設定します。

Table 2 引数

Name	Description	Variation
adcData	ADC 変換データ	-8388608~8388607
pga1Gain	PGA1 ゲイン	1,2,4,8,16,21.33,32 null(PGA1 が OFF の場合)
pga2Gain	PGA2 ゲイン	1,2,4
vrefSel	Vref 選択信号	0:外部 Vref 1:内部 Vref
iex1Sw	励起電流源 1 接続端子	"VIN1P","VIN1N","VIN2P",...,"VIN4N"
iex2Sw	励起電流源 2 接続端子	"VIN1P","VIN1N","VIN2P",...,"VIN4N"
iex1C	励起電流源 1 電流値	0.00010,0.00025,0.00050,0.00100, null(IEX1C が OFF の場合)
iex2C	励起電流源 2 電流値	0.00010,0.00025,0.00050,0.00100, null(IEX2C が OFF の場合)

入力電圧換算のサンプル

```
// It's a constant value as VREFs
const extVref = 4.6;
const intVref = 2.4;

// Bit width of ADC data
const BitWidth = CHIPID == 0xA3 ? 24 :
                  CHIPID == 0xA2 ? 20 :
                  CHIPID == 0xA1 ? 16 :
                  24;

// Calculating plot value from ADC data
function data(adcData, pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C) {
  var vref = vrefSel == 0 ? extVref : intVref;
  var pgaGain = pga1Gain == null ? pga2Gain : pga1Gain * pga2Gain;
  return _round((adcData * vref / (1 << (BitWidth - 1))) / pgaGain,
    numberOfDecimals(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C));
}

// Determine minimum value of Y axis
function min(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C) {
  return data(-(1 << (BitWidth - 1)), pga1Gain, pga2Gain, vrefSel);
}

// Determine maximum value of Y axis
function max(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C) {
  return data((1 << (BitWidth - 1)) - 1, pga1Gain, pga2Gain, vrefSel);
}

// Determine unit label of Y axis
function unit(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C) {
  return "[V]";
}

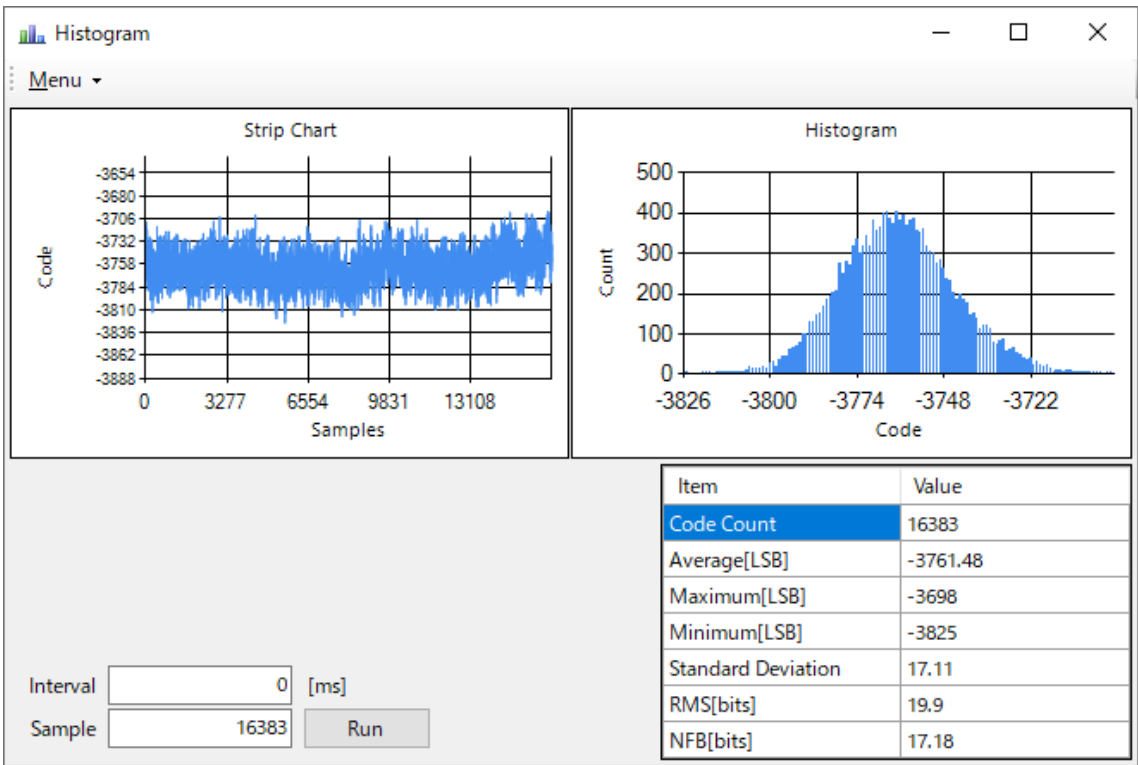
// Number of decimals
function numberOfDecimals(pga1Gain, pga2Gain, vrefSel, iex1Sw, iex2Sw, iex1C, iex2C) {
  return Number.parseInt(BitWidth * Math.log10(2));
}

// It's a local utility function
function _round(value, precision) {
  var s = Math.pow(10, precision);
  return Math.round(value * s) / s;
}
```

9. ヒストグラムウィンドウ

「Sample」で指定した回数分 ADC 変換データを取得し、ヒストグラムを作成します。

Fig. 13 ヒストグラムウィンドウ



9.1. ADC 変換データ取得間隔

ADC 変換データを取得する時間間隔は「Interval」で設定します。「Interval」に 0 を設定した場合と、0 を超える値を設定した場合で動作が異なります。

Interval == 0

デバイスは「連続変換モード」で動作し、OSR, CLKDIV, CHOP の有無で定まる最も短い時間間隔で ADC 変換データを取得します。

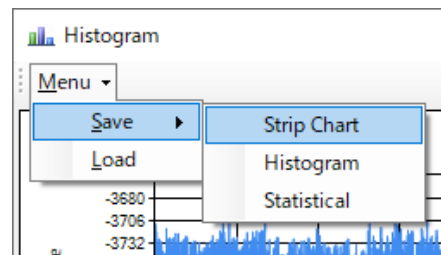
Interval > 0

デバイスは「シングル変換モード」で動作し、Interval で設定された時間間隔毎に ADC 変換データを取得します。そのため、Interval には 1 回の ADC 変換データ取得にかかる時間よりも大きな値を設定する必要があります。デバイス設定ウィンドウの変換レート表示部分に表示されているシングル変換の変換時間の 2 倍程度の値を目安に Interval の値を設定してください。

9.2. 結果の保存と読み込み

収集した ADC 変換データを CSV 形式で外部ファイルに保存できます。

Fig. 14 データ保存



保存可能なものはそれぞれ

- ・ Strip Chart : ADC 変換結果の生データ
- ・ Histogram : ヒストグラム
- ・ Statistical : 生データから作られる統計データ

です。

読み込みは、「Strip Chart」の結果のみ可能です。Strip Chart の結果を読み込むと、そのデータを元にヒストグラムと統計データを生成し、ウィンドウに表示します。

尚、Strip Chart の保存データの形式は波形モニタのデータ保存で出力されるファイルと同じですので、「8.5.ADC 変換コードへの演算処理」による ADC コード以外への変換行われていないものに関し、波形モニタで保存したファイルを読み込み、ヒストグラム、統計データを表示する事が可能です。