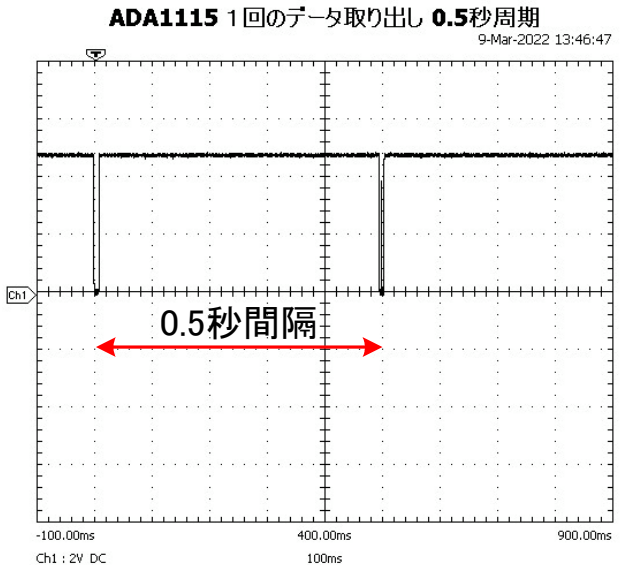
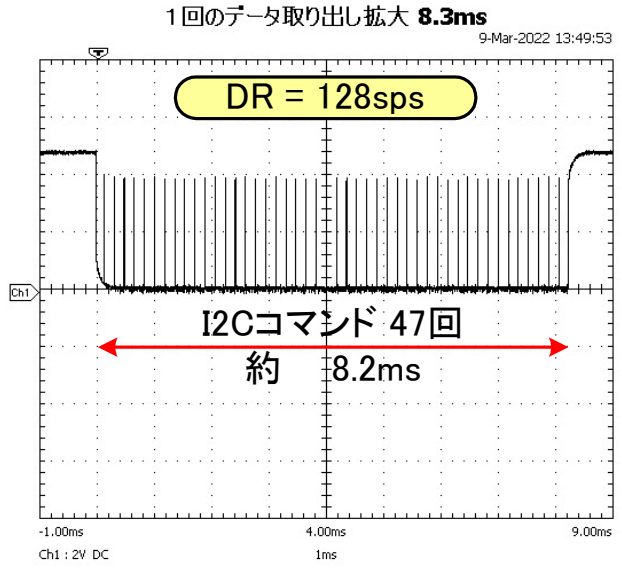
ADS1115の DR(SPS値)に関して

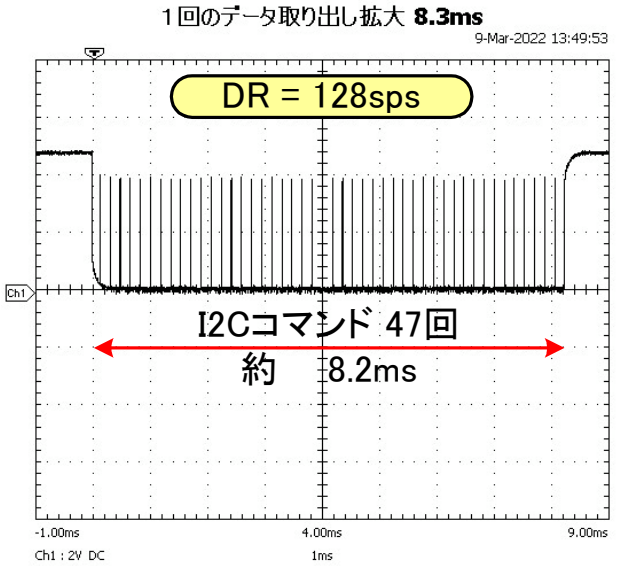
ADS1115の DR(データレイト)の SPS設定値ですが、最初、サンプルレイトの事と勘違いしてました。 サンプルレイトとは、一定のタイムインターバルで、データを A/D変換する事で、通常インターバルタイマで、正確な時間間隔でサンプリングする事を意味します。

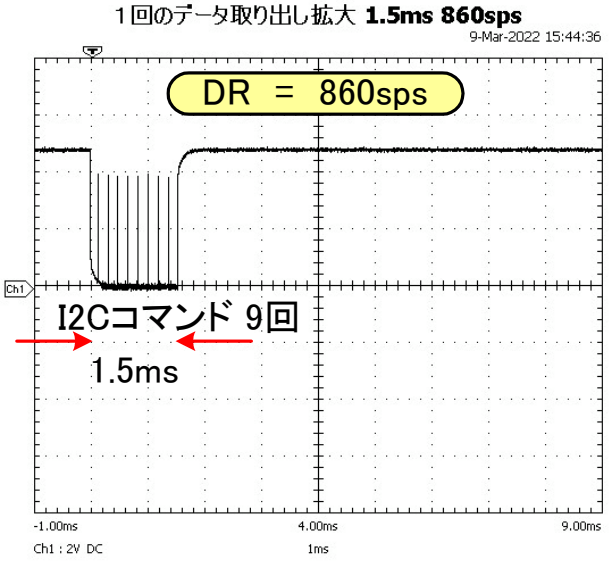
今回、DR=100: 128sps(デフォルト)に設定していました。 1回のA/D変換で、A/D変換スタートのコマンドを、送ってから、A/D変換値を 取り込むまで、どのくらい時間がかかるのか、気になっていました。 よって、I2Cの 電文を出すたびに、LEDを、ON OFFするようにプログラムして、そのLEDの信号タイミングを、オシロで測定しました。 右の オシロ波形上は、タイムインターバルは、百円マイコン側のインターバルタイマで、0.5秒間隔で A/Dデータを取り込んでいます。 で、右の オシロ波形下は、1回のA/D変換に必要なI2Cコマンドのやり取りです。 全体で、47回コマンドのやり取りを、していました。 47回全体で 約 8.2ms でした。





最初の予想より遅いと思ったので、試しに DRに 111 (860sps)を設定して、他は全く同じ条件で、繰り返し A/D変換を行ってみました。 処理時間の結果は、何と 8.2ms が 1.5ms に 短くなりました。 5.47倍 程度早くなりました。128sps が 860sps なる倍率は 6.72倍ぐらいなので、sps値の比率ほどは、早くなりません。 原因は、このsps値は ⊿Σ型A/D変換器の速度であり、その前段のマルチプレクサや、サンプルホルダの時間は、別で前段の回路の処理時間は固定で変わらないのではと、思います。 ちなみに、1チャネル1個のA/D変換データを取り込む I2Cコマンドのシーケンスは、最初にConfig Regにて、各種設定と A/Dスタートを 行います。 次に、Config Regの、OS bitを読み、A/D変換終了を確認します。 変換中であれば、終了になるまで、何度でも読み直します。 変換終了が確認出来たら、変換データを読み出します。
つまり、先頭と最後の I2Cコマンド以外は、変換終了の確認コマンドを、繰り返しているのです。
この測定を行った時の、動画をお見せします。





この実験で測定したデータの傾向は、？

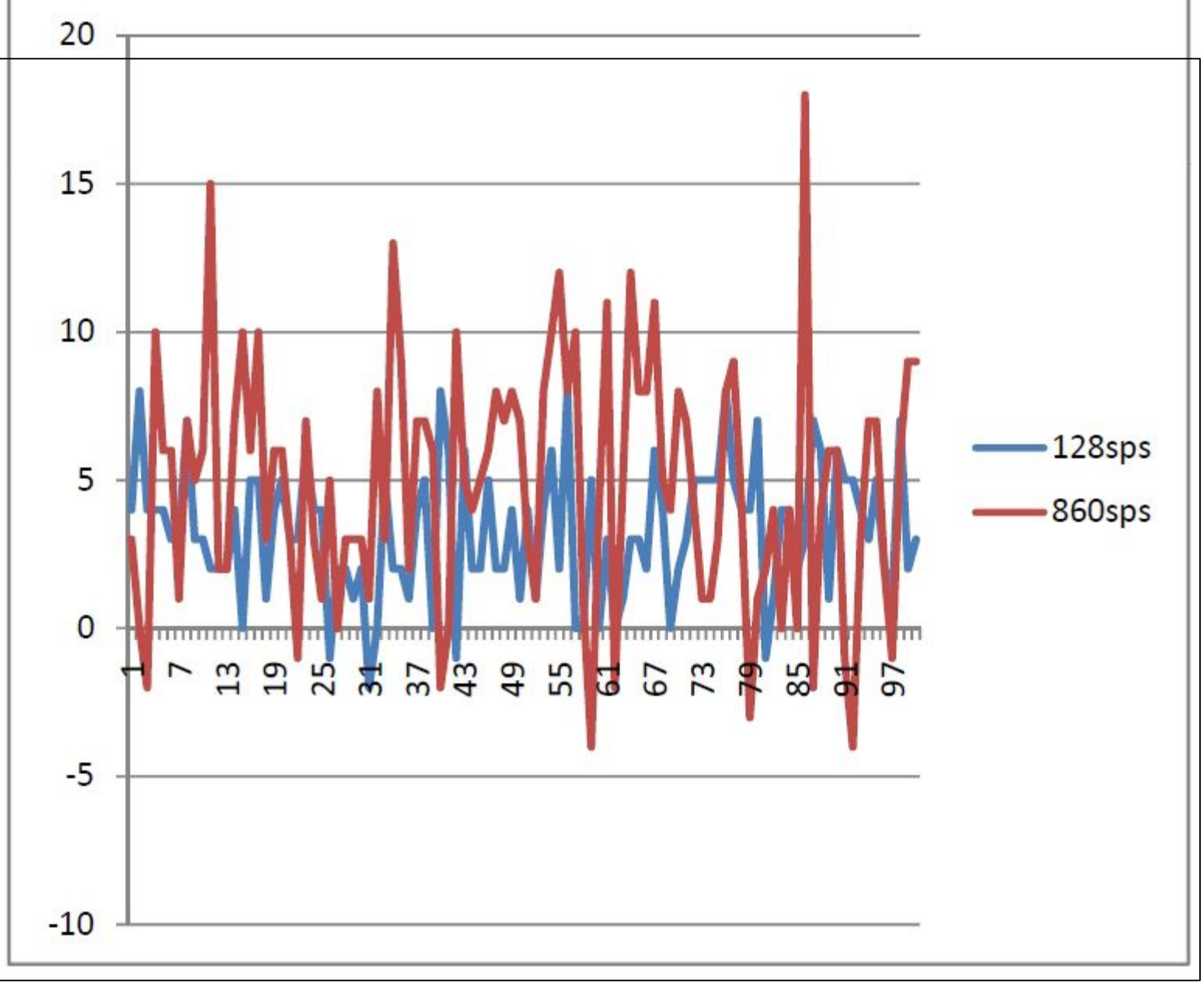
という事で、DR=100 （ 128sps)よりも、DR=111 (860sps)に設定した方が、A/D変換速度が速い事が、分かりました。

何事も、早いに越したことはなさそうですが、そうとばかりは、限りません。 例えば、パソコンの CPUクロックが 4GHzぐらいで稼働したら、その分 電力を消費し発熱が大きくなります。

A/Dコンバータの場合は、より高速で動かそうとすると、変換精度が、悪くなる傾向があります。 その事も気になっていたので、DR=128spsの 場合の GNDを計った A/D変換データ 100個と、860spsで GNDを計った A/D変換データ 100個の 計測値を、テラタームからコピペで、テキストエディタに移しておきました。

コンマ区切りファイルとして、Excelに読み込ませ、128spsと 860spsそれぞれ 100個の平均値と、グラフ表示を行いました。
次のページで、お見せします。 やはり速度が速くなると、ノイズの影響が、増える傾向になるようです。

連番	128sps	860sps
81	-1	2
84	4	4
85	2	0
86	3	18
87	7	-2
88	6	4
89	1	6
90	6	6
91	5	-1
92	5	-4
93	4	3
94	3	7
95	5	7
96	2	2
97	1	-1
98	7	6
99	2	9
100	3	9
合計	328	487
平均	3.28	4.87



ALERT/RDY信号を、A/D変換完了信号として使えるか。？

それと、もう一つ ADS1115の端子で、ALERT/RDY信号を、A/D変換完了の、信号として使えるかの確認です。 これは、実際やってみないと分からないので実験してみます。

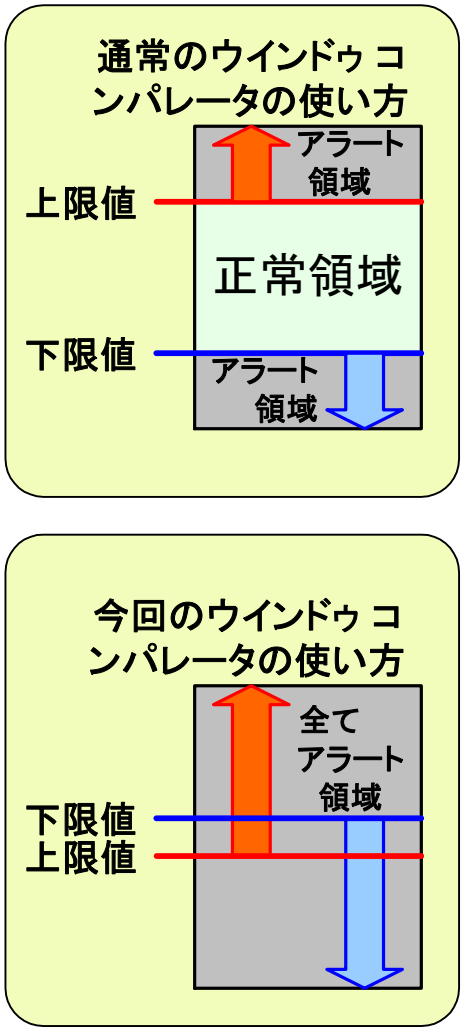
今回、PGAは、2/3倍（ FS=±6.144V ）のレンジに設定して そこに 0～5Vの電圧を入れる予定です。 コンパレータのモードですが、ウインドゥコンパレータモードで、使用する予定です。

通常の ウインドゥコンパレータモードとは、Lo_thresh Reg （ 下限値 ）レジスタより、信号が 下回るか、 Hi_thresh reg（ 上限値 ）レジスタより 信号が、上回った時、アラート信号が 出ます。（ 右図上 参照 ）

仮に 上限値を、下限値より低くして、下限値を、上限値より高くして、設定すれば、どうなるかという事です。（ 右図下 参照 ）

この設定が可能であれば、A/D変換後、変換値の値に関係なく、常時アラート信号が、出ます。 このように設定する事で、アラート信号を、A/D変換完了信号とみなして使えるのでは、ないかと考えます。

閾値の値は、上限値＝0x3300 、下限値＝0x3500 を 仮に想定します。



マイコンとパソコン間の通信仕様

マイコン側の ADS1115アクセス処理は、だいたい完成しました。　次に、ロガーを作成する上で重要なマイコン、パソコン間通信の コマンド、データ転送の通信仕様を 決めて行きます。 今回は簡易ロガーと、いう事で、あまり凝らずに作りやすい方法で開発を行います。

★ テキストベース（文字列）で、コマンド、データのやり取りを行う。 こうする事で、マイコン側の通信機能開発時に、テラターム等の端末ソフトを使ってデバッグが、行える。

一般的に、文字列より バイナリデータで 通信を行う方が、転送するデータ Byte数が少なくなり、より高速なデータ通信が行えます。

しかし、パソコン側、マイコン側の両方のプログラムが、完成しないとデバッグに入れないし、デバッグにやや、手間が かかると思われます。

★ マイコン側の実現したい機能一覧:

① サンプルレイトを 設定できる事。
（ 0.1、0.2、0.5、
1、2、5、10、20、30、60 秒 ）

② チャネル数を 設定できる事。
（ 1、2、3、4 チャネル指定
先頭チャネル ch.0 から順次使用する ）

③ A/Dコンバータの DR指定（ 128sps か 860sps ）

④ 平均化数の指定出来る事。（ 1、3、5、7 回 ）

⑤ リードリレーの制御（サンプルレイト依存）:
1秒未満の指定では、ONしたまま
1秒以上の指定では、測定時のみONする

リード リレー メーカーの 資料では、
Operate Time： 0.5ms Max
Release Time： 0.2ms Max と 書いてあります。
一応、リードリレー ONから 20ms待ってから
A/Dスタートをかける事に します。

パソコンから マイコンへ送るコマンド

① サンプルレイト設定コマンド：
asr=0[Cr]　（ [Cr]は、Crコード =0Dh ）
=0 は、0 ～ 9 の値を取り、0=0.1、1=0.2、2=0.5、3=1、4=2、5=5、6=10、7=20、8=30、9=60 秒を設定する。
（ デフォルト 3で １秒 ）

② チャネル数 設定コマンド：
ach=1[Cr]
=1 は、1 ～ 4 の値を取り チャネル数を意味する。　（ デフォルト ４チャネル ）

③ ADコンバータ変換速度 設定コマンド：
asp=0[Cr]　（ 128spsの設定/ 8.2ms ）
asp=1[Cr]　（ 860spsの設定/ 1.5ms ）
（ デフォルト 128sps ）

④ 平均化処理の設定コマンド：
avr=0[Cr]　（ 平均化処理 無し ）
avr=1[Cr]　（ 3個の 平均化処理 ）
avr=2[Cr]　（ 5個の 平均化処理 ）
avr=3[Cr]　（ 7個の 平均化処理 ）
（ デフォルト 平均化処理 無し ）

⑤ 測定開始コマンド：
run[Cr]（ 定周期で 連続的に取込み ）

⑥ 測定停止コマンド：
[esc]　（ ESC Code = 1Bh ）

⑦ 単発測定コマンド：
get[Cr]（ コマンド発行時、単発取込み ）

マイコンからパソコンへ送信されるデータ

① チャネル数１のデータ：
>0123[Cr][Lf]
0123が、ch.0 のデータ （ 16進４桁表現 ）

② チャネル数２のデータ：
>01234567[Cr][Lf]
0123が、ch.0 のデータ （ 16進４桁表現 ）
4567が、ch.1 のデータ （ 16進４桁表現 ）

③ チャネル数３のデータ：
>0123456789AB[Cr][Lf]
0123が、ch.0 のデータ （ 16進４桁表現 ）
4567が、ch.1 のデータ （ 16進４桁表現 ）
89ABが、ch.2 のデータ （ 16進４桁表現 ）

④ チャネル数４のデータ：
>0123456789ABCDEF[Cr][Lf]
0123が、ch.0 のデータ （ 16進４桁表現 ）
4567が、ch.1 のデータ （ 16進４桁表現 ）
89ABが、ch.2 のデータ （ 16進４桁表現 ）
CDEFが、ch.3 のデータ （ 16進４桁表現 ）

※ マイコンから パソコンへ送る文字列の 終端コードが、[Cr][Lf]（ 0Dh, 0Ah ）と なっているのは、ターミナルソフト側にて、Crコードだけだと、左端に復帰するだけで、改行を行わないため、[Cr][Lf]を、パソコン側にて送信している。 という事です。

パソコン側のロガープログラムの作成

今回も、またマイコン側に 時間をかなり割いてしまったので、パソコン側は、必要最低限の機能で作成します。

パソコン側のプログラム開発環境は、訳あって古いバージョンの Delphiを使用します。

Delphiは、1995年ごろに発売され始めたWoindows環境のプログラム開発環境です。
VBによく似た ビジュアルコンポーネントをフォームに貼り付けて機能を実装します。

VBと異なるのは、言語が Object PASCALである事。 Delphiは、基本ネィティブコンパイラなので実行速度が速い。というメリットがあります。
但し、ネィティブコンパイラなので、変数の型宣言は、厳密に行う必要があります。

パソコン側プログラム開発の説明を、やっているとまた、時間が かかるので 申し訳ありませんが パソコン側のプログラム説明は 省略します。

データファイルの ファイル構成

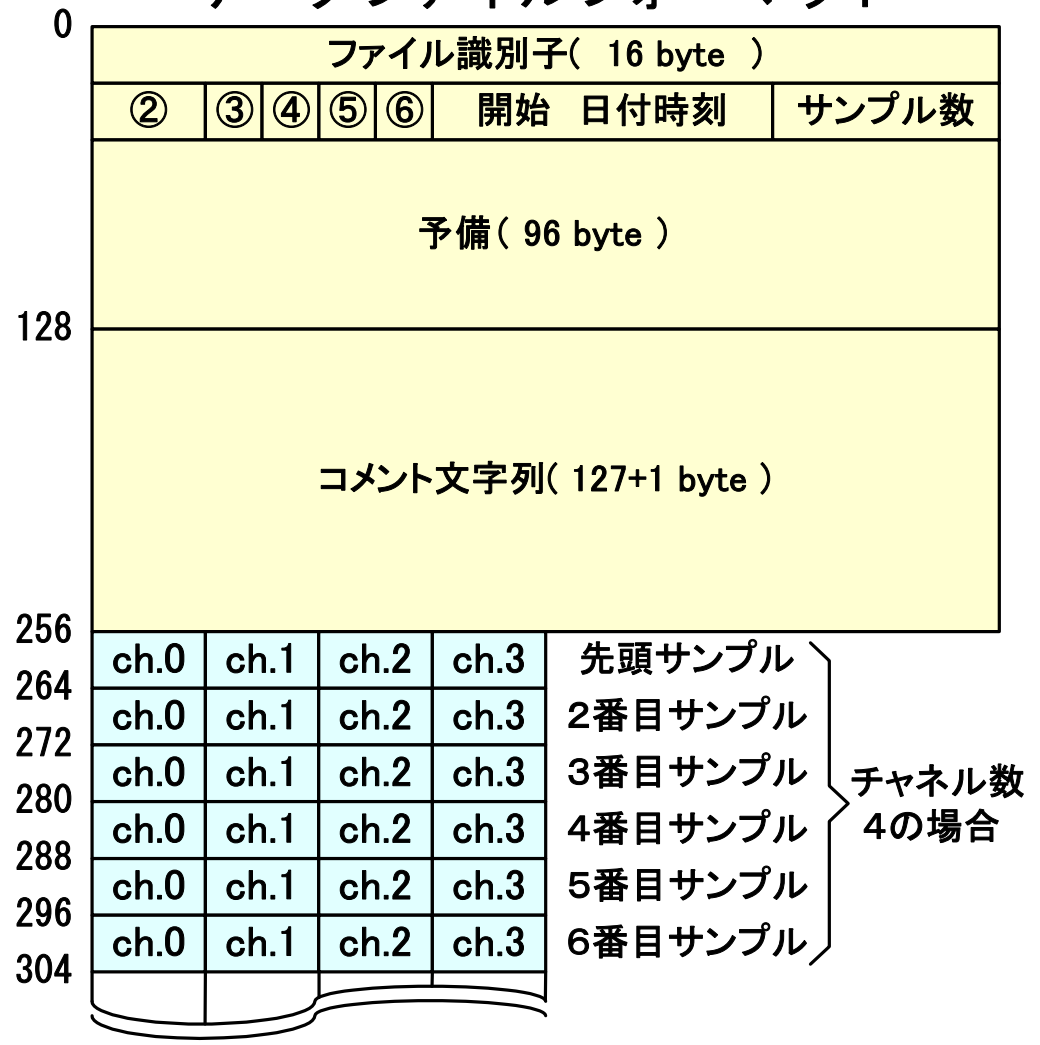
パソコン側プログラム説明を 省略するにしても
データを保存するファイル内の フォーマットは
最低限決めておく必要がありますね。

ファイル先頭部分にヘッダー部分を置く。

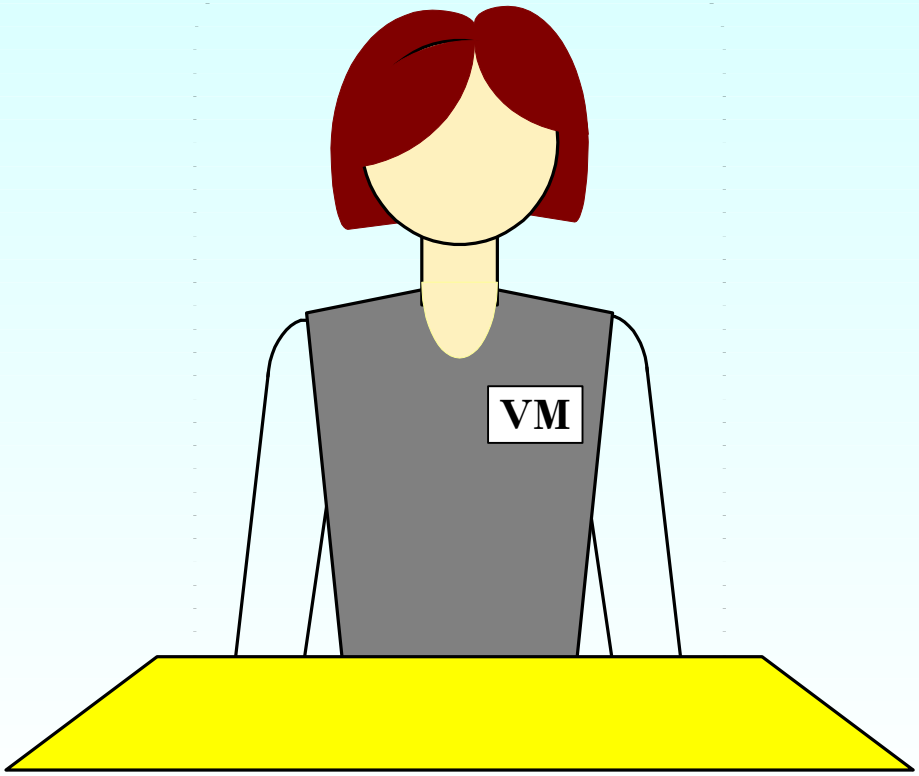
① ファイル識別子：（ 16 byte ）
② サンプルレイト：（ 0.01秒単位/ 2 byte ）
③ チャネル数：（ 1 byte ）
④ A/D変換速度：（ 1 byte ）
⑤ 平均化数：（ 1 byte ）
⑥ ファイル分割単位：（ 1 byte ）
（ 日 or 時 単位 ）
⑦ 作成開始 日付、時刻：（ 6 byte ）
⑧ 記録サンプル数：（ 4 byte ）
⑨ コメント：（ 半角 127 文字 ）

次に、データを順次 格納していく事にします。

データファイルフォーマット



これは、何　？



さしあたり、直線と 円と 円弧で描いた雑な絵で すみません。

　ネット上で。アバターというのでしょうか。？
自分の化身として アニメ的なキャラクターで声も変えて話する動画を 見かけますよね。

新しい試みで、やってみようと思います。

一つには、最初の挨拶で 私のような高齢者が出て来ると、若い人は、それだけで引いてしまう事もある　と思ったからです。

　音声は、あるソフトを購入して、聞き易い声で説明を行う事は、可能になりました。

試行錯誤しながら、やってみます。
ちなみに VM は、バーチャルモードです。