ADC 入力を行うNUCLEO-F401RE のプロジェクトサンプルです。ADC CH0 ~ ADC CH5 の6本の入力が可能です。

NUCLEO-F401RE は STMicroelectronics 社製の Cortex-M4 ARM CPU である STM32F401RET6 を搭載した基板です。

NUCLEOのUSART2_TX(PA2)とUSART2_RX(PA3)は工場出荷時の状態でST-LNK用のCPU(U2)に接続されています。 Mini USB コネクタ CN1 を使用して VCP(仮想 COM ポート)通信を行うためです。

ST-LINK としてデバッグを実行しながら同時に VCP(仮想 COM ポート)通信を行うことができるようになっています。

Mini USB コネクタ CN1 を使用して VCP(仮想 COM ポート)通信を行うにためには、STMicroelectronics 社が提供する ドライバ ST-LNK/V2-1 をインストールする必要があります。

PC アプリケーションの Access_SerialPort を使用して UART(非同期シリアル通信)により、ADC 入力値の表示を行うことができます。

試用版の開発ツール Atollic TrueSTUDIO for ARM Lite で作成したプロジェクトです。 ビルド可能なプログラムのコードサイズが 32Kbyte 以内の制限があります。

プログラムの開始番地は 0x08000000 です。デバッグが可能です。

目次

1. ADC 入力について	
1.1. 入力電圧	
1.2. ADC 入力に使用する信号	
2. 使用する信号	4
2.1. UART 信号	4
2.2. VCP(仮想 COM ポート)通信に使用する場合	4
2.3. CN10に接続して使用する場合	5
1) RS232Cドライバ	5
2)RS232C ケーブル接続図	6
3) UART 通信確認のための接続例	6
3. プログラム実行時の動作	7
4. ADC 入力のデータ確認	7
4.1. アプリケーション Access_SeriaPortの起動画面	7
4.2. COM ポートの選択と通信速度の設定	
1) COM ポートの選択	
2) 通信速度の設定	
4.3. ADC 入力コマンドの送信	
1) コマンド "ADC Input CHO" 送信時の動作	
2) コマンド "ADC Input ALL" 送信時の動作	
3) コマンド "ADC Input STOP" 送信時の動作	
5. プロジェクトの構成	
5.1. プロジェクトF401N_ADC_U2の起動画面	
5.2. 追加したソース・フォルダとファイル	
6. 主なモジュールの説明	14
6.1. ソース・フォルダ src 内のファイル	14
6. 2. HandleADC	
1) ADC の初期化	
2) ADC 入力開始	
 ADC データの移動平均 	
4) 移動平均について	
6. 3. HandleCLK	
6. 4. HandleGPIO	
6. 5. Handle TIM	
6. 6. HandleUART	
6.7. UserPrograms	21
6.8. Communicate_UART	

1. ADC 入力について

1.1. 入力電圧

分解能 12bit の A/D コンバータ入力です。

ADC 入力電圧 と ディジタル値 との対応は理論上、 入力電圧 0V:0 入力電圧 +3.3V:4095

です。

実際はオフセットと傾きがあり、ぴったりこの数値にはなりません。通常、ソフトまたはハードで補正を行います。

1.2. ADC 入力に使用する信号

NUCLEO=F401REの ADC 入力に使用する信号は以下の ADC_CH0 ~ ADC_CH5 の6本です。

ADC 信号表

信号名	CPU 機能名	CPU 信号名	NUCLEO=F401RE
			コネクタピン番号
ADC_CH0	ADC1_IN10	PC0	CN7-38
ADC_CH1	ADC1_IN11	PC1	CN7-36
ADC_CH2	ADC1_IN12	PC2	CN7-35
ADC_CH3	ADC1_IN13	PC3	CN7-37
ADC_CH4	ADC1_IN14	PC4	CN10-34
ADC_CH5	ADC1_IN15	PC5	CN10-6
AGND	VSSA	VSSA	CN10-32

2. 使用する信号

2.1. UART 信号

UART 通信に使用する信号と接続相手との接続は以下の通りです。

U5のUSART2_TX(PA2)とUSART2_RX(PA3)はそれぞれ solder bridge を介してU2 または CN10 に接続されます。

UART 信号表

番号	CPU 機能名 U5	CPU 信号名	U2 との接続	基板コネクタピン番号	方向	接続相手の 信 号 名
1	USART2 TX	PA2	SB13-PA3	SB63-CN10-35	>	RxD
2	USART2_RX	PA3	SB14-PA2	SB62-CN10-37	<	TxD
3	GND	GND		CN10-20	< >	GND

2.2. VCP(仮想 COM ポート)通信に使用する場合

工場出荷時の状態では U5 の PA2 と PA3 は U2 側に接続されています。 PC と Mini USB コネクタ CN1 とを接続して VCP(仮想 COM ポート)通信を行うことができます。

Mini USB コネクタ CN1 を ST-LINK として使用し、デバッグを行いながら同時に VCP(仮想 COM ポート)通信を行うことが できます。(STMicroelectronics 社が提供するドライバ ST-LINK/V2-1 のインストールが必要です。)

アプリケーション Access_SerialPort を使用して、VCP(仮想 COM ポート)通信により PC から送信する ADC コマンドを 受信して、コマンドに従って ADC 入力値を送信します。

PCとNUCLEO_F401REとの接続は以下のようになります。





2.3. CN10に接続して使用する場合

Solder bridge の SB13 と SB14 を解放して、SB63 と SB62 を接続状態にして CN10-35(PA2)と CN10-37(PA3)に 接続します。

CMOS レベルどうしの信号を直接接続して通信を行うことができます。

UART 信号に RS232C ドライバを接続して RS232C レベルの信号どうしで接続すれば RS232C 通信を行うことが できます。

1) RS232Cドライバ

RS232Cドライバの例を以下に示します。



RS232C ストレートケーブル接続の場合の DSUB-9S への接続例を以下に示します。 この例では DSUB-9S には TxD, RxD, GND の3本の信号以外は接続されません。

番号	RS232C	方向	ドライバ側	ストレートケーブル側
	ドライバ		D-SUB 9S	D-SUB 9P
1	RS2_TxD	>	2	2 RxD
2	RS2_RxD	<	3	3 TxD
3	GND	< >	5	5 GND



図2.3.2)

3) UART 通信確認のための接続例

アプリケーション Access_SerialPort を使用して、RS232C 通信により PC から送信する ADC コマンドを 受信して、コマンドに従って ADC 入力値を送信します。

動作確認を行うための接続は以下のようになります。この例はRS232CレベルでUART 信号を接続しています。



図2.3.3)

3. プログラム実行時の動作

1) プログラムを実行すると基板上のLED LD2(緑)が 1 秒点灯、2 秒消灯 で点滅します。

2) UART2 のチャンネルでデータ受信待ちを行います。

3) 有効なコマンドを受信した場合、以下のように動作します。

番号	受信コマンド	動作	対応する ADC 入力
1	ADC Input STOP	ADC データ送信停止	
2	ADC Input CH0	ADC CHO データ送信	PC0 : ADC1_IN10
3	ADC Input CH1	ADC CH1 データ送信	PC1 : ADC1_N11
4	ADC Input CH2	ADC CH2 データ送信	PC2 : ADC1_N12
5	ADC Input CH3	ADC CH3 データ送信	PC3 : ADC1_N13
6	ADC Input CH4	ADC CH4 データ送信	PC4 : ADC1_IN14
7	ADC Input CH5	ADC CH5 データ送信	PC5 : ADC1_N15
8	ADC Input ALL	ADC 全CHデータ送信	PC0 – PC5 : ADC1_IN10 – ADC1_IN15

4. ADC 入力のデータ確認

アプリケーション Access_SerialPort を使用して、UART(非同期シリアル通信)により PC からコマンド文字列を送信して NUCLEO-F401RE に入力される ADC データを取得します。

4.1. アプリケーション Access_SerialPort の起動画面

まず、PC と NUCLEO-F401RE とを USB ケーブルまたは RS232C インターフェースで接続してください。 アプリケーション Access_SeriaPort を起動すると以下のダイアログが表示されます。

P Access SerialPort	
通信	
COM選択 通信速度: 9600 ▼	
COM検索	
送信データ: ASCII文字のみ送信可能です。(最大256byte)	送1言
受信データ表示選択 受信データ表示選択 の ASCII の HEX	
COM(シリアルポート)を検索しました。	

USB ケーブルまたは RS232C ケーブルを接続している場合、"COM(シリアルポート)を検索しました。"と表示されます。

4.2. COM ポートの選択と通信速度の設定

1) COM ポートの選択

COM 選択の ComboBox で COM を選択します。

Page Access SerialPort	They are a second state of the local division of the local divisio	
通信 COM選択 COM選択 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM12 COM2 COM2 COM2 COM2 COM2 COM2 COM2 COM	通信速度: 9600 - を信可能です。(最大256byte)	送信
受信データ: COM (シリアルボート)を検:	受信データ表示選択	



確認のメッセージが表示されるので、よい場合は [はい(Y)] ボタンをクリックします。

次ページに続く

e Access SerialPort	_ D X
通信	
COM選択 COM12 ▼ 通信速度: 9600 ▼ COM検索	
送信データ: ASCII文字のみ送信可能です。(最大256byte)	送信
受信データ表示選択 ● ASCII ◎ HEX	
受167~90: COM(シリアルボート)を検索しました。 COM12に設定しました。	

この例では、"COM12に設定しました。" とメッセージが表示されています。

2) 通信速度の設定

プロジェクトサンプル TrST_F401N_ADC_U2 の通信速度が 115200bps なので通信速度を 115200bps に設定します。

通信速度の ComboBox で 115200 を選択します。

Access SerialPort	
通信 COM選択 COM12 ● 通信速度: 9600 ● COM検索 送信データ: ASCII文字のみ送信可能です 9600 ● 19200 38400 57600 ● 115200 ● 9600 ● 115200 ● 115200 ●	送信 表示選択 〇 HEX
COM(シリアルボート)を検索しました。 COM12に設定しました。	



確認のメッセージが表示されますので [はい(Y)] ボタンをクリックしてください。

Access SerialPort	_
通信	
COM選択 COM12 ▼ 通信速度: 115200 ▼ COM検索	
送信データ: ASCII文字のみ送信可能です。(最大256byte)	送信
●受信データ表示選択 ● ASCII ● HEX	
受信データ: COM(シリアルポート)を検索しました。 COM12に設定しました。	
115200bpslに設定しました。	

受信データの欄に "115200bps に設定しました。" とメッセージが表示されます。

4.3. ADC 入力コマンドの送信

ADC 入力コマンドは以下の通りです。

番号	受信コマンド	動作	対応する ADC 入力
1	ADC Input STOP	ADC データ送信停止	
2	ADC Input CH0	ADC CHO データ送信	PC0 : ADC1_IN10
3	ADC Input CH1	ADC CH1 データ送信	PC1 : ADC1_N11
4	ADC Input CH2	ADC CH2 データ送信	PC2 : ADC1_N12
5	ADC Input CH3	ADC CH3 データ送信	PC3 : ADC1_N13
6	ADC Input CH4	ADC CH4 データ送信	PC4 : ADC1_N14
7	ADC Input CH5	ADC CH5 データ送信	PC5 : ADC1_N15
8	ADC Input ALL	ADC 全 CH データ送信	PC0 – PC5 : ADC1_IN10 – ADC1_IN15

1) コマンド "ADC Input CHO" 送信時の動作

コマンド "ADC Input CHO" を送信した場合の動作を以下に示します。

Para Access SerialPort	
通信 COM選択 COM12 ▼ 通信速度: 115200	
送信データ: ASCII文字のみ送信可能です。(最大2 ADC Input CH0	256byte) 送信
· 受信データ:	受信データ表示選択 ● ASCII ◎ HEX
COM(シリアルホート)を検索しました。 COM12に設定しました。 115200bpsに設定しました。 ADC Input CHO OK CHO : 1407	
CHO : 1408 CHO : 1406 CHO : 1410	

CPU 基板は 0.5 秒に一回 ADC CHO の入力値を送信します。

次ページに続く

2) コマンド "ADC Input ALL" 送信時の動作

コマンド "ADC Input ALL"を送信した場合の動作を以下に示します。

Access SerialPort	
通信	
COM選択 COM12 ▼ 通信速度: 115200 ▼	
	送信
送信ナーダ: ASUII文子のの送信可能に9。(取入250byte)	
ADC Input ALL	
─受信データ表示選択─	
CHO: 1408, CH1: 1339, CH2: 1270, CH3: 1244, CH4: 1211,	CH5: 1183 🔺
CHO: 1408, CH1: 1357, CH2: 1281, CH3: 1254, CH4: 1247,	CH5: 1243
CHU: 1411, CHI: 1354, CHZ: 1283, CH3: 1256, CH4: 1246, CUO: 1400, CU1: 1347, CU2: 1270, CU3: 1251, CU4: 1234	CH5: 1237
CHU: 1409, CHI: 1347, CH2: 1279, CH3: 1251, CH4: 1254, CHU: 1409, CH1: 1341, CH2: 1279, CH3: 1251, CH4: 1229	CH5: 1210 CH5: 1205
CHO: 1400, CH1: 1327, CH2: 1264, CH3: 1241, CH4: 1223, CHO: 1407, CH1: 1327, CH2: 1264, CH3: 1241, CH4: 1194.	CH5: 1162
CHO: 1409, CH1: 1331, CH2: 1262, CH3: 1238, CH4: 1197,	CH5: 1161
CHO: 1409, CH1: 1339, CH2: 1271, CH3: 1247, CH4: 1210,	CH5: 1172
CHO: 1409, CH1: 1342, CH2: 1273, CH3: 1250, CH4: 1222,	CH5: 1191
CHO: 1407, CH1: 1360, CH2: 1286, CH3: 1256, CH4: 1245,	CH5: 1238
CHO: 1407, CH1: 1352, CH2: 1282, CH3: 1254, CH4: 1247,	CH5: 1238
CHU: 1408, CHI: 1350, CH2: 1279, CH3: 1257, CH4: 1239,	CH5: 1219
CHU: 1408, CHI: 1344, CHZ: 1270, CH3: 1201, CH4: 1230, CUO: 1400, CU1: 1997, CU9: 1989, CU9: 1990, CU4: 1100	CHD: 1207 CH5: 1187
CHO: 1409, CH1: 1323, CH2: 1258, CH3: 1238, CH4: 1188,	CH5: 1153
CHO: 1407, CH1: 1337, CH2: 1272, CH3: 1247, CH4: 1213.	CH5: 1177
CHO: 1407, CH1: 1342, CH2: 1273, CH3: 1250, CH4: 1222,	СН5: 1193
CHO: 1404, CH1: 1361, CH2: 1285, CH3: 1259, CH4: 1256,	CH5: 1255 📉

CPU 基板は 0.5 秒に一回 ADC CH0 ~ ADC CH5 の 6CH の入力データを送信します。

コマンド "ADC Input STOP" を送信した場合の動作を以下に示します。

通信 COM選択 COM12 ● 通信速度: 115200 ● COM検索 送信データ: ASCII文字のみ送信可能です。(最大256byte) 送信データ: ASCII ● HEX Ø信データ: Ø信データ: Ø信データ: CH0: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1408, CH1: 1349, CH2: 1278, CH3: 1251, CH4: 1228, CH5: 1208 CH0: 1408, CH1: 1349, CH2: 1277, CH3: 1251, CH4: 1259, CH5: 1255 CH0: 1409, CH1: 1340, CH2: 1277, CH3: 1253, CH4: 1259, CH5: 1255 CH0: 1407, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1239, CH5: 1220 CH0: 1405, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1342, CH2: 1277, CH3: 1253, CH4: 1218, CH5: 1184 CH0: 1408, CH1: 1315, CH2: 1273, CH3: 1253, CH4: 1218, CH5: 1184 CH0: 1408, CH1: 1315, CH2: 1273, CH3: 1253, CH4: 1218, CH5: 1184 CH0: 1408, CH1: 1357, CH2: 1274, CH3: 1253, CH4: 1225, CH5: 1186 CH0: 1408, CH1: 1357, CH2: 1274, CH3: 1253, CH4: 1225, CH5: 1186 CH0: 1408, CH1: 1357, CH2: 1274, CH3: 1253, CH4: 1225, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1274, CH3: 1254, CH4: 1239, CH5: 1224 CH0: 1410, CH1: 1347, CH2: 1279, CH3: 1254, CH4: 1239, CH5: 1224 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1254, CH4: 1219, CH5: 1182 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1254, CH4: 1219, CH5: 1180 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1254, CH4: 1219, CH5: 1181 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1254, CH4: 1219, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1215, CH5: 1181 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1215, CH5: 1181	Access SerialPort			
COM選択 COM12 ● 通信速度: 115200 ● COM検索 送信ず - タ: ASCII文字のみ送信可能です。(最大256byte) ADC Input STOP 愛信ず - 夕: CH0: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1408, CH1: 1349, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1406, CH1: 1349, CH2: 1278, CH3: 1251, CH4: 1228, CH5: 1208 CH0: 1409, CH1: 1349, CH2: 1277, CH3: 1259, CH4: 1259, CH5: 1255 CH0: 1407, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1342, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1184 CH0: 1408, CH1: 1315, CH2: 1277, CH3: 1253, CH4: 1218, CH5: 1194 CH0: 1408, CH1: 1315, CH2: 1273, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1337, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1337, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1337, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1342, CH2: 1274, CH3: 1253, CH4: 1225, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1254, CH4: 1219, CH5: 1240 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1249, CH5: 1247 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1255, CH4: 1212, CH5: 1180 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1254, CH4: 1212, CH5: 1180 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1407, CH1: 1340, CH2: 1274, CH3: 1250, CH4:	通信			
送信データ: ASCII文字のみ送信可能です。(最大256byte) ADC Input STOP 受信データ表示選択 ● ASCII ● HEX CH0: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1408, CH1: 1349, CH2: 1278, CH3: 1251, CH4: 1215, CH5: 1208 CH0: 1409, CH1: 1349, CH2: 1277, CH3: 1251, CH4: 1228, CH5: 1208 CH0: 1409, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1408, CH1: 1315, CH2: 1273, CH3: 1253, CH4: 1218, CH5: 1194 CH0: 1408, CH1: 1315, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1357, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1362, CH2: 1274, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1408, CH1: 1357, CH2: 1278, CH3: 1253, CH4: 1239, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1279, CH3: 1253, CH4: 1239, CH5: 1240 CH0: 1400, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1340, CH2: 1277, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1409, CH1: 1357, CH2: 1274, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1409, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1249, CH5: 1247 CH0: 1409, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1249, CH5: 1247 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1180 CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1181 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1257, CH4: 1241, CH5: 1237 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1257, CH4: 1241, CH5: 1237 CH0: 1409, CH1: 1359, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1277	COM選択 COM12	▼ 通信速度: 1152	00 🔻	
ADC Input STOP 受信データ: ● ASCII ● HEX CH0: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 ● ASCII ● HEX CH0: 1408, CH1: 1349, CH2: 1277, CH3: 1251, CH4: 1215, CH5: 1208 ● CH0: 1409, CH1: 1349, CH2: 1278, CH3: 1259, CH4: 1228, CH5: 1208 CH0: 1409, CH1: 1360, CH2: 1287, CH3: 1253, CH4: 1237, CH5: 1220 ● CH0: 1405, CH1: 1342, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1342, CH2: 1277, CH3: 1253, CH4: 1218, CH5: 1194 ● CH0: 1408, CH1: 1315, CH2: 1248, CH3: 1220, CH4: 1170, CH5: 1145 CH0: 1408, CH1: 1315, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 ● CH0: 1408, CH1: 1342, CH2: 1274, CH3: 1253, CH4: 1253, CH5: 1247 CH0: 1408, CH1: 1342, CH2: 1274, CH3: 1259, CH4: 1253, CH5: 1247 ● CH0: 1408, CH1: 1347, CH2: 1274, CH3: 1259, CH4: 1253, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1259, CH4: 1249, CH5: 1240 ● CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1239, CH5: 1247 CH0: 1409, CH1: 1357, CH2: 1284, CH3: 1259, CH4: 1212, CH5: 1180 ● CH0: 1407, CH1: 1347, CH2: 1272, CH3: 1253, CH4: 1239, CH5: 1227 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1217, CH5: 1183 ● CH0: 1408, CH1: 1336, CH2: 1272, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1408, CH1: 1336, CH2: 1272, CH3: 1250, CH4: 1217, CH5: 1183 ● CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1244, CH5: 1227 CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1244, CH5: 1227 ● CH0: 1409, CH1: 1359, CH2:	送信データ: ASCII文 [:]	━━ 字のみ送信可能です。(最ス	大256byte)	送信
受信データ表示選択 受信データ。ASCII 一 HEX CH0: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1406, CH1: 1349, CH2: 1278, CH3: 1251, CH4: 1228, CH5: 1208 CH0: 1409, CH1: 1360, CH2: 1287, CH3: 1259, CH4: 1259, CH5: 1255 CH0: 1407, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1342, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1408, CH1: 1315, CH2: 1272, CH3: 1244, CH4: 1218, CH5: 1194 CH0: 1408, CH1: 1315, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1194 CH0: 1408, CH1: 1337, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1387, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1362, CH2: 1274, CH3: 1253, CH4: 1253, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1259, CH4: 1253, CH5: 1247 CH0: 1408, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1239, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1212, CH5: 1180 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1212, CH5: 1180 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1217, CH5: 1183 CH0: 1408, CH1: 1336, CH2: 1272, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1284, CH3: 1250, CH4: 1244, CH5: 1227 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1244, CH5: 1227 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1214, CH5: 1227 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1250, CH4: 1244, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1274, CH3: 1255, CH4: 1244, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1287, CH3: 1255, CH4: 1244, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227 CH0: 1409, CH1: 1354, CH2: 1281, CH	ADC Input STOP			
受信データ: ● ASCII ● HEX CH0: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1406, CH1: 1349, CH2: 1278, CH3: 1251, CH4: 1228, CH5: 1208 CH0: 1409, CH1: 1360, CH2: 1287, CH3: 1259, CH4: 1259, CH5: 1255 CH0: 1407, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1342, CH2: 1272, CH3: 1244, CH4: 1218, CH5: 1194 CH0: 1408, CH1: 1315, CH2: 1274, CH3: 1220, CH4: 1170, CH5: 1145 CH0: 1408, CH1: 1337, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1357, CH2: 1274, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1362, CH2: 1274, CH3: 1253, CH4: 1225, CH5: 1186 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1259, CH4: 1253, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1253, CH4: 1239, CH5: 1240 CH0: 1409, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1341, CH2: 1279, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1274, CH3: 1253, CH4: 1249, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1212, CH5: 1180 CH0: 1407, CH1: 1340, CH2: 1272, CH3: 1253, CH4: 1249, CH5: 1222 CH0: 1407, CH1: 1340, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1408, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1181 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1257, CH4: 1245, CH5: 1237 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1257, CH4: 1245, CH5: 1237 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227			受信データ表示選択一	
CHO: 1408, CH1: 1334, CH2: 1271, CH3: 1251, CH4: 1215, CH5: 1180 CH0: 1406, CH1: 1349, CH2: 1278, CH3: 1251, CH4: 1228, CH5: 1208 CH0: 1409, CH1: 1360, CH2: 1287, CH3: 1259, CH4: 1259, CH5: 1255 CH0: 1407, CH1: 1345, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1405, CH1: 1342, CH2: 1277, CH3: 1253, CH4: 1237, CH5: 1220 CH0: 1408, CH1: 1315, CH2: 1273, CH3: 1220, CH4: 1218, CH5: 1194 CH0: 1408, CH1: 1357, CH2: 1273, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1410, CH1: 1342, CH2: 1273, CH3: 1253, CH4: 1218, CH5: 1182 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1253, CH4: 1225, CH5: 1196 CH0: 1408, CH1: 1362, CH2: 1274, CH3: 1253, CH4: 1249, CH5: 1247 CH0: 1408, CH1: 1357, CH2: 1284, CH3: 1254, CH4: 1249, CH5: 1240 CH0: 1400, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1239, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1212, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1212, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1279, CH3: 1253, CH4: 1212, CH5: 1240 CH0: 1407, CH1: 1347, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1408, CH1: 1336, CH2: 1272, CH3: 1250, CH4: 1217, CH5: 1183 CH0: 1409, CH1: 1339, CH2: 1274, CH3: 1250, CH4: 1217, CH5: 1181 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1257, CH4: 1245, CH5: 1237 CH0: 1409, CH1: 1359, CH2: 1274, CH3: 1257, CH4: 1245, CH5: 1237 CH0: 1409, CH1: 1354, CH2: 1281, CH3: 1255, CH4: 1241, CH5: 1227	受信型→勾:		🧿 ASCII 🔘 HEX	
	CHO: 1408, CH1: 13 CHO: 1408, CH1: 13 CHO: 1406, CH1: 13 CHO: 1409, CH1: 13 CHO: 1407, CH1: 13 CHO: 1407, CH1: 13 CHO: 1408, CH1: 13 CHO: 1407, CH1: 13 CHO: 1407, CH1: 13 CHO: 1407, CH1: 13 CHO: 1409, CH	334, CH2: 1271, CH3: 349, CH2: 1278, CH3: 360, CH2: 1287, CH3: 345, CH2: 1277, CH3: 345, CH2: 1277, CH3: 345, CH2: 1277, CH3: 345, CH2: 1277, CH3: 342, CH2: 1273, CH3: 342, CH2: 1274, CH3: 362, CH2: 1284, CH3: 367, CH2: 1284, CH3: 362, CH2: 1274, CH3: 367, CH2: 1274, CH3: 367, CH2: 1272, CH3: 367, CH2: 1272, CH3: 367, CH2: 1272, CH3: 3747, CH2: 1272, CH3: 347, CH2: 1272, CH3: 321, CH2: 1272, CH3: 336, CH2: 1272, CH3: 337, CH2: 1274, CH3: 338, CH2: 1287, CH3: 359, CH2: 1287, CH3: 354, CH2: 1281, CH3:	1251, CH4: 1215, 1251, CH4: 1228, 1259, CH4: 1259, 1253, CH4: 1237, 1244, CH4: 1218, 1220, CH4: 1218, 1253, CH4: 1218, 1253, CH4: 1225, 1254, CH4: 1225, 1254, CH4: 1229, 1253, CH4: 1239, 1253, CH4: 1212, 1238, CH4: 1212, 1238, CH4: 1217, 1250, CH4: 1217, 1250, CH4: 1241,	CH5: 1180 CH5: 1208 CH5: 1255 CH5: 1220 CH5: 1194 CH5: 1194 CH5: 1182 CH5: 1247 CH5: 1247 CH5: 1240 CH5: 1222 CH5: 1180 CH5: 1180 CH5: 1181 CH5: 1237 CH5: 1227

CPU 基板は ADCデータの送信を停止します。

³⁾ コマンド "ADC Input STOP" 送信時の動作

- 5. プロジェクトの構成
 - 5.1. プロジェクト F401N_ADC_U2 の起動画面

TrueSTUDIO で作成したプロジェクトF401N_ADC_U2 を開いた状態を以下に示します。 左側のプロジェクト・エクスプローラーのF401N_ADC_U2 を展開した状態です。

	* * * * * 2 00 *	□ • 8 0•				クイック・アクセン 自風C や
コジェクト・エクスプローラー 🛛 🕒 😫 🎽 🗖 🗆	🖻 main.c 😫				• •	きアウト_ 23 日テンプ」 ® Make _ 「
F401N_ADC_U2	5 **				*	B 12 R x • W
き バイナリー	6 ** Abstract	main function.				stm32f4xx.h
g Includes	7 **	2			-	Communicate_UART.F
CommonModules	8 ** Functions :	. main				HandleADC.h
Communicate_UART	10 ** Environment :	Atollic TrueSTUDIO(R)				U HandleCLK.h
A Handles	11 **	STMicroelectronics STM32F4xx Stan	dard Peripherals Lib	srary		HandleGPIO.h
B Libraries	12 **		- W			HandleTIM.h
UserPrograms	13 ** Distribution:	of any kind	out any warranty			UserPrograms.h
A src	15 **	of any kind.				main(void) : int
Debug	16 ** (c)Copyright	Atollic AB.				
E401N ADC 1/2 off bunch	17 ** You may use t	this file as-is or modify it accord	ing to the needs of	your		
a realized for the	18 project. This	; file may only be built (assembled	or compiled and lin	(ked)		
stm3214_flash.id	20 ** with other to	ools than Atollic TrueSTUDIO(R) is	not permitted.	ner		
	21 **	NIS LIGH HEALES IT BESTINGEN,	not permanents			
	22 *************	*****************************	*****************	******		
	23 */					
	24					
	25				-	
	27 // TrST_F401N_ADO	2_02				
	28 //					
	29 // ADC人77hB企UART	2で送信する。				
	30 // ST-I TNKØMini	USB:コネクタCN1を使用してVCP(仮想COMポート)	通信が可能です。			
	32 //	use and a set of the s	BUBD ¹ - Jans C. P. a			
	33 // V001 : 2014.10	3.06 : Revive				
	34 //				*	
					*	
	▶問題 2 ④タスク □ コン	ソール 目 プロパティ 🛷 検索				7 0
	0項目			a construction of	T towned	1.1 Present/Merlin Preside Andread Activity
	紀道			09-X	7.2	ロケーション
	*					
			1917			

- 5.2. 追加したソース・フォルダとファイル 追加したソース・フォルダとファイルについて簡単に説明します。
 - CommonModules (ソース・フォルダ)
 共通に使用するモジュールを記述してあります。
 時間待ち、文字列操作 などの処理を記述しています。
 - 2) Communicate_UART (ソース・フォルダ)
 - a) Communicate_UART .h Communicate_UART .c (ファイル)
 接続相手との通信処理を記述しています。
 データを受信して、ADC コマンドならコマンドに従って ADC 入力データを送信します。
 - Handles (ソース・フォルダ)
 Peripheral の設定などを行っています。
 - a) HandleCLK.h HandleCLK.c (ファイル) 内部クロック HSI を使用するための設定を記述しています。 HSI(周波数 16MHz)を入力して PLL により 84MHz にしてシステムクロック SYSCLK として使用します。
 - b) HandleGPIO.h HandleGPIO.c (ファイル) GPIO 入出力の初期設定を記述しています。

- c) HandleTIM.h HandleTIM.c (ファイル) タイマ割り込みを使用するために、タイマの初期設定を記述しています。 1mSec ごとにタイマ割り込みが発生するように設定しています。
- d) HandleUART.h HandleUART.c (ファイル) UARTの初期化とUART 送受信の処理を記述しています。
- UserPrograms (ソース・フォルダ)
 LED のための処理を記述しています。
- a) UserPrograms.h UserPrograms.c (ファイル) Status LED : LD2(緑) に使用している GPIO の初期設定と点滅処理を記述しています。
- 6. 主なモジュールの説明
- 6.1. ソース・フォルダ src 内のファイル ソース・フォルダ src 内のファイルでプログラムを追加した主なファイルについて簡単に説明します。
 - 1) main.c
 - a) main 関数 プログラムはここから開始します。主に初期化処理関数を呼び出しています。

int main(void)

b)システムクロックの設定

システムクロック SYSCLK の設定を行います。 内部クロック HSI を入力に選択し、PLL を使用して 84MHz に設定します。

//-----// HSIを選択して、PLL ClockをSystem Clockとして使用する。 : SYSCLK = 84MHz //-----SetHSICLK84MHz();

c)周辺クロックの初期化

//----// 周辺クロックの初期化
//----void RCC_Configuration(void);

d) GPIO の初期化

//-----// GPIO初期化 //----void Init_GPIOs(void); e) Status LED ポートの初期化

//------// Status LEDポート初期化 : LD2(緑) //------InitializePortStatusLED();

f) UART の初期化

//----// UART2通信パラメータ初期化:通信速度 9600bps
//----InitializeCommunicate_UART2((uint32_t)9600);

g) ADC の初期化

//
// ADC1のWork初期化
//
InitializeWork_ADC1();

h) TIM11の初期化

//			
// TIM11初期化			
//			
<pre>InitializeTIMxx(TIM11, RCC_APB2Periph_TIM11, GLB_uint16_vTIM11_CCR1);</pre>			
<pre>EnableIrqTIMxx(TIM11, RCC_APB2Periph_TIM11, TIM1_TRG_COM_TIM11_IRQn);</pre>			
//	TIM11 I	Interrupt	ON
//			

- stm32f4xx_it.h stm32f4xx_it.c このファイルに割り込み処理を記述します。 本プロジェクトサンプルでは ADC 割り込み と TIM11 のタイマ割り込み処理 および UART2 の割り込み処理を 記述しています。
- 6.2. HandleADC
 - 1) ADC の初期化 ADC の初期化を行います。

//------// ADC1の初期化 //-----void InitializeADC1(void); 2) ADC 入力開始

指定されたA/D CHの入力を開始します。

//-----// ADCx INx入力開始 //-----/* ADC1を選択可能 */ //-----//引数 : // ADC_TypeDef *ADCx : where x can be 1, 2 or 3 to select the ADC peripheral. // uint16_t uint16_INx : ADCx_INx選択 11 0 : PA0 : ADC1 IN0 11 1 : PA1 : ADC1 IN1 11 2 : PA2 : ADC1_IN2 // 3 : PA3 : ADC1 IN3 11 4 : PA4 : ADC1_IN4 11 5 : PA5 : ADC1 IN5 11 6 : PA6 : ADC1 IN6 // 7 : PA7 : ADC1_IN7 11 8 : PB0 : ADC1 IN8 11 9 : PB1 : ADC1 IN9 11 10 : PC0 : ADC1 IN10 11 11 : PC1 : ADC1_IN11 11 12 : PC2 : ADC1_IN12 11 13 : PC3 : ADC1_IN13 // 14 : PC4 : ADC1 IN14 11 15 : PC5 : ADC1 IN15 // uint8_t uint8_ADC_SampleTime : The sample time value to be set for the selected channel. ADC SampleTime 3Cycles: Sample time equal to 3 cycles 11 ADC_SampleTime_15Cycles: Sample time equal to 15 cycles 11 ADC SampleTime 28Cycles: Sample time equal to 28 cycles 11 ADC_SampleTime_56Cycles: Sample time equal to 56 cycles 11 ADC_SampleTime_84Cycles: Sample time equal to 84 cycles 11

// ADC_SampleTime_112Cycles: Sample time equal to 112 cycles

// ADC_SampleTime_144Cycles: Sample time equal to 144 cycles // ADC_SampleTime_480Cycles: Sample time equal to 480 cycles

void Start_ADCx_INx(ADC_TypeDef *ADCx, uint16_t uint16_INx, uint8_t uint8_ADC_SampleTime);

3) ADC データの移動平均

ADC 入力データの各 CH の移動平均を行います。

//	
// ADC入力データの移動平均処理	
//	

void MovingAverageADC(void);

4) 移動平均について

データ8個の移動平均について説明します。

データ8個の平均を新しいデータが入力される毎に1データずつ、ずらして平均していきます。 平均するデータが移動していくので移動平均と呼びます。以下にその様子を示します。



//-----// A/D入力データの移動平均処理

//-----

void MovingAverageADC(void);

の処理では、移動平均の個数を 32 個に設定しています。 ヘッダファイル HandleADC.h のなかで **#define** defNumbersAverage 32 を定義しています。 この定義を変更すると、移動平均の個数を変更できます。

6. 3. HandleCLK

動作クロックに内部クロック HSI: 16MHz を選択し、PLL により 84MHz にして使用します。

//-----// HSIを選択して、PLL ClockをSystem Clockとして使用する。 : SYSCLK = 84MHz //-----

void SetHSICLK84MHz(void);

6.4. HandleGPIO

GPIO を初期化します。

1) 低消費電力モード時の GPIO 初期化 最初は、使用しない GPIO ピンをアナログ入力モードに初期化します。

//-----

// 低消費電力モード時のGPIO初期化

//-----

void InitializeGPIOs_LowPower(void);

2) GPIO の初期化

必要な GPIO の初期化を行います。

//-----// GPIO初期化 //-----

void Init_GPIOs(void);

6.5. HandleTIM

1) TIM11の初期化

タイマ割り込みのために TM11 を初期化してインターバルをセットします。 1mSec ごとに割り込みがかかるように設定しています。

以下の関数の引数にTIM11用のパラメータを指定してTIM11を初期化します。

//-----// TIMxx初期化 //-----//引数 : // TIM_TypeDef *TIMxx : TIM選択 // uint32_t RCC_APB1Periph_TIMxx : specifies the APB1 peripheral to gates its clock. // uint16_t uint16_TIMxx_CCR1 : TiMxx CH1のインターバル //-----

void InitializeTIMxx(TIM_TypeDef *TIMxx, uint32_t RCC_APB1Periph_TIMxx, uint16_t uint16_TIMxx_CCR1);

2) タイマ割り込み許可 以下の関数の引数に希望する TMxx 用のパラメータを指定して割り込みを許可します。

6. 6. HandleUART

1) UART の初期化

//----// UART2初期化
//----//引数 :
// uint32_t uint32_BaudRate : 通信速度 bps
//-----void InitializeUART2(uint32_t uint32_BaudRate);

UART 送信
 送信 Buffer に格納されたデータを指定データ数送信します。

//-----// UART2 送信処理 : 送信Bufferに送信データがセットされた状態でCallされる。 //-----//引数 : // uint16_t uint16_SendLength : 送信データ数

//戻り値 :
// 0:送信終了
// 1:エラー
//------

void SendUART2(uint16_t uint16_SendLength);

3) UART 受信

受信待ちを行いデータを受信したら、受信 Buffer に格納します。

int16_t ReceiveUART2(uint8_t *puint8_ReceiveBuffer);

6.7. UserPrograms

UserPrograms.hにはLEDに使用するGPIOに対する定義を記述してあります。

以下に、UaerPrograms.cに記述している関数の説明を記します。

LED に使用する GPIO の初期化(共通処理)
 GPIO番号とピン番号を指定して 1/0を初期化します。

```
//-----
// LEDポート初期化
//-----
//引数 :
// GPIO_TypeDef *GPIOx : GPIOポート指定
// uint16_t GPIO_Pin_x : GPIOピン指定
//------
```

void InitializePortLED(GPI0_TypeDef *GPI0x, uint16_t GPI0_Pin_x);

2) LED の点滅処理(共通処理) GPIO番号とピン番号などを指定して 希望のLEDの点滅処理を行います。

//-----// LED点滅: 点灯/消灯 切り替え //-----// 点灯/消灯 を切り替えると同時に 点灯時間/消灯時間 をセットする。 //-----//引数 : // GPIO_TypeDef *GPIOx : GPIOポート指定 // uint16_t GPI0_Pin_x : GPI0ピン指定 // int16_t *pint16_0n0ff : ON/OFF状態 11 0 : OFF 11 1 : ON // uint16_t *puint16_Timer : 点灯時間/消灯時間をセットする変数のポインタ // uint16_t uint16_TimeON : 点灯時間 // uint16 t uint16 TimeOFF : 消灯時間 //----void BlinkLED(GPI0_TypeDef *GPI0x, uint16_t GPI0_Pin_x, int16_t *pint16_OnOff, uint16_t *puint16_Timer, uint16_t uint16_TimeON, uint16_t uint16_TimeOFF);

3) LED に使用する 1/0 の初期化

a) StatusLED:LD2(緑)

void InitializePortStatusLED(void);

4) StatusLED: LD2(緑)の点滅

//-----// Status LED点滅: LD2(緑): 点灯/消灯 切り替え
//-----// TIMx割り込み内でGLB_uint16_BlinkTimerStatusLEDをデクリメントする。
// GLB_uint16_BlinkTimerStatusLEDがのになった時、呼び出される。
//-----// 点灯/消灯 を切り替えると同時に 点灯時間/消灯時間 をセットする。
//-----//引数 :
// uint16_t uint16_TimeON : 点灯時間
// uint16_t uint16_TimeOFF : 消灯時間
//-----void BlinkStatusLED(uint16_t uint16_TimeON, uint16_t uint16_TimeOFF);

6.8. Communicate_UART

1) UART の初期化

//----// UART2通信パラメータ初期化
//----//引数 :
// uint32_t uint32_BaudRate : 通信速度
//------

void InitializeCommunicate_UART2(uint32_t uint32_BaudRate);

2) UART 通信処理

UART の受信待ちを行い、受信データを判定して ADC データ送信状態フラグをセットします。

int16_t Communicate_UART2(void);

3) 受信コマンドの実行

//-----//受信コマンド判定 および 実行 //-----//引数 // uint16_t uint16_ReceiveLength : 受信データ数 // uint8_t *puint8_ReceiveData : 受信データが格納されたBufferのポインタ // uint8_t *puint8_SendData : 応答送信データを格納するBufferのポインタ

//戻り値: 応答送信データ数

//-----

4) ADC入力コマンド処理

//-----// ADC入力コマンド処理 //-----// Command : ADC Input [Parameter] 11 // [Parameter] : // STOP: ADC入力停止 CH0 : ADC1_IN10 11 CH1 : ADC1 IN11 11 11 CH2 : ADC1 IN12 CH3 : ADC1_IN13 // CH4 : ADC1 IN14 11 CH5 : ADC1 IN15 11 11 ALL : ADC1_IN10 - ADC1_IN15 //-----//引数 : // uint16 t uint16 DataLength : パラメータのデータ数 // uint8 t *puint8 Parameter : ADC入力パラメータのポインタ "CHO" ~ "CH5": 単独CHのADC入力開始 11 "ALL": 6CH全てのADC入力開始 11 "STOP": ADC入力の停止 // // uint8 t *puint8 SendData : 応答データを格納するBufferのポインタ //戻り値 : 応答送信データ数 //----uint16_t ComActInputADC(uint16_t uint16_DataLength, uint8_t *puint8_Parameter, uint8_t *puint8_SendData); // Command : "ADC Input [Parameter]" の [Parameter] の部分を判定して、ADC データの送信を決定する。 //-----// [Parameter] を判定してADCデータ送信状態フラグをセットする。 //-----//volatile int16 t GLB int16 vCommunicateADC = -1; // A/Dデータ送信状態フラグ //-----[Parameter] 11 -1 : ADC送信停止 "STOP" 11 0 : ADC CH0データ送信 "CH0" 11 **1 : ADC CH1**データ送信 "CH1" 2 : ADC CH2データ送信 "CH2" // 3 : ADC CH3データ送信 11 "CH3" 4 : ADC CH4データ送信 // "CH4" 5 : ADC CH5データ送信 11 "CH5" 6 : ADC 全CHデータ送信 "ALL" 11 //-----

5) ADC データ送信

ADC データ送信状態フラグ GLB_int16_vCommunicateADC を参照して、内容に従って送信処理を行います。

//
// ADCデータ送信
//
void SendDataADC(void);

ADC データ送信状態フラグ GLB_int16_vCommunicateADC の値に対する動作は以下の通りです。

//	
//	-1:送信停止
//	0 : ADC CHOデータ送信 : ADC1_IN10
//	1 : ADC CH1データ送信 : ADC1_IN11
//	2 : ADC CH2データ送信 : ADC1_IN12
//	3 : ADC CH3データ送信 : ADC1_IN13
//	4 : ADC CH4データ送信 : ADC1_IN14
//	5 : ADC CH5データ送信 : ADC1_IN15
//	6 : ADC 全CHデータ送信 : ADC1_IN10 - ADC1_IN15
//	

有限会社りばいぶ

電子工作のための「飛石伝ひ」

改訂履歴

- V001 2014/10/06 初版
- V002 2014/10/21 誤記訂正
- V003 2014/11/03 誤記訂正