ソース解析

株式会社リーシス

- 概要
  - 既存にあるプログラムのリメークを検討したらそのプログラムの作成に携わった技術者が居ない
  - ドキュメントが紛失してしまいプログラム内部の仕様が判らないなどで困ったこと
  - OSなどのバージョンアップ等でそれまで動作していたプログラムが正常に動作しなくなってしまった

- 弊社ではソースコードが存在していればソースコードを解析しドキュメントを起こし直したり、潜在しているバグを再発見して最適化を図る等のサービスを実施しています

- 端末側、特定用途向けの制御系ソフト開発経験が豊富です
  通信制御や画像処理等にも広く展開を考えました

- 特定用途向け制御ソフトの経験が豊富
- DOS、Windows初期版を利用の実績
- ハードウェアの置換え提案も含めたシステム解析

ご相談もご対応致します。

対応CPU	
Renesas	H8
	SH2
	SH4
	RX210
Zilog	Z80
Intel	8085
	8051
NEC	V25
	V53
Motorola	68020
oki	ARM7
NXP	Cortex-M3
Analog Devices	Blackfin

対応OS
MS-DOS
Windows95
Windows98
Windows 2000
Windows xp
Windows vista
Windows 7
Windows 8
NORTi

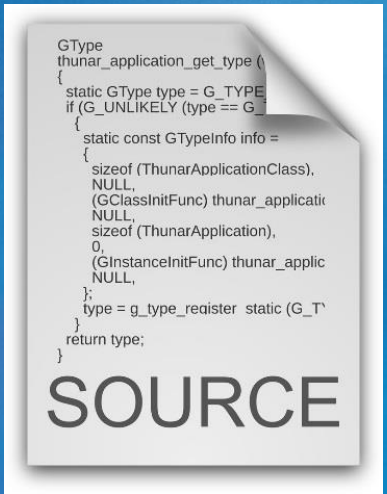
対応バス
USB
SCSI
RS232C
RS422
RS485
Ether Net

解析手順

1. お客様からプログラム動作のレクチャーを受け大まかな機能を把握します。

2. ソースコードからプログラムの流れを理解し機能ブロック図を作成

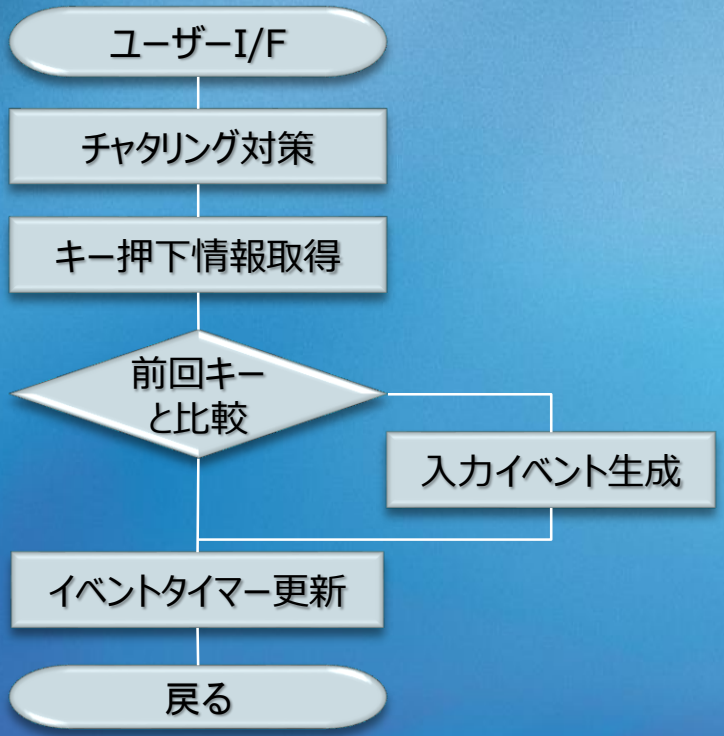
3. 機能ブロック単位で機能解析シートに登録



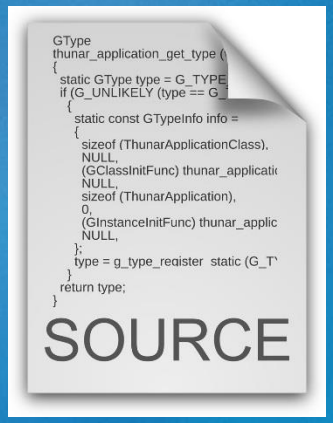
機能		起動	内容	イベント
ユーザI/F	キー入力処理	イベント タイマー	チャタリング対策	
			キー押下情報取得	
			前回キーと比較	キー入力
			イベントタイマーの更新	
	表示処理	表示更新 イベント	ディスプレー表示を更新	
			表示イベントクリア	
画像入力処理	画像補正	VSync割込	温度情報取得	
			温度補正テーブル切り替え	
	転送処理	HSync割込	ラインメモリからRAMに画像データを転送	
画像出力処理	主走査制御	HSync割込	ポリゴンモータPLL監視	PLLロック
			主走査カウンター設定	
	副走査制御	VSync割込	パルスモーターコントローラ設定	
			副走査カウンター設定	
			エンコードパルスの監視	
			リミッターSWの監視	暴走検知

4．機能ブロック単位でフローチャートを作成

機能		起動	内容	イベント
ユーザI/F	キー入力処理	イベントタイマー	チャタリング対策	
			キー押下情報取得	
			前回キーと比較	キー入力
			イベントタイマーの更新	

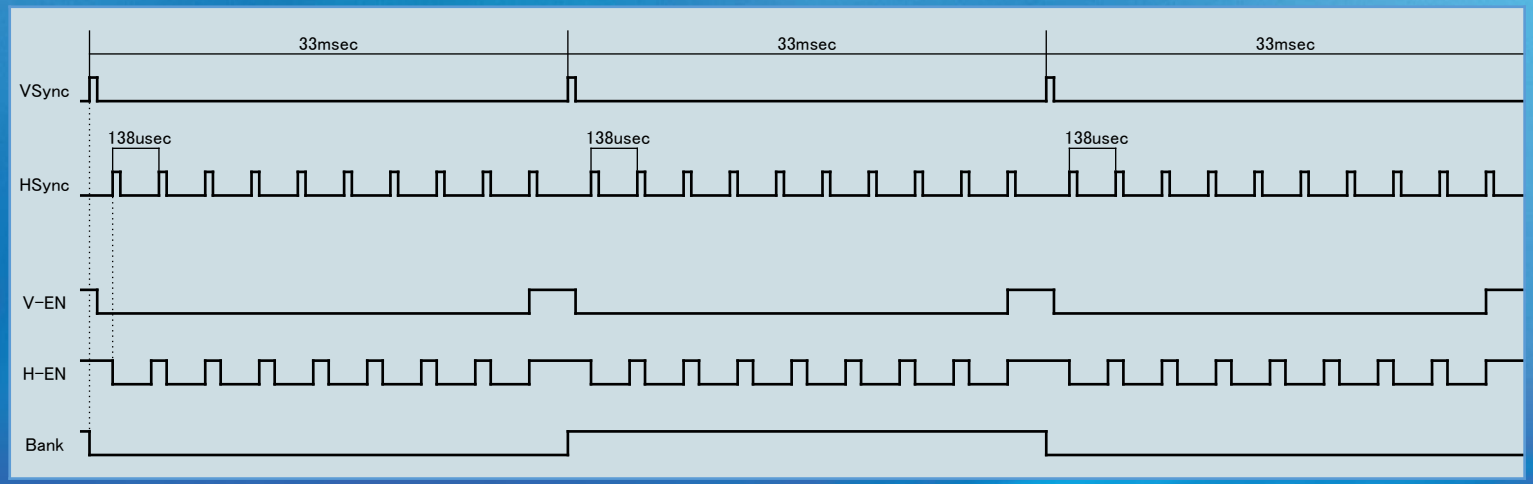


5. 通信については通信部のソースを解析後プロトコルアナライザーを使用して実際の通信を確認する

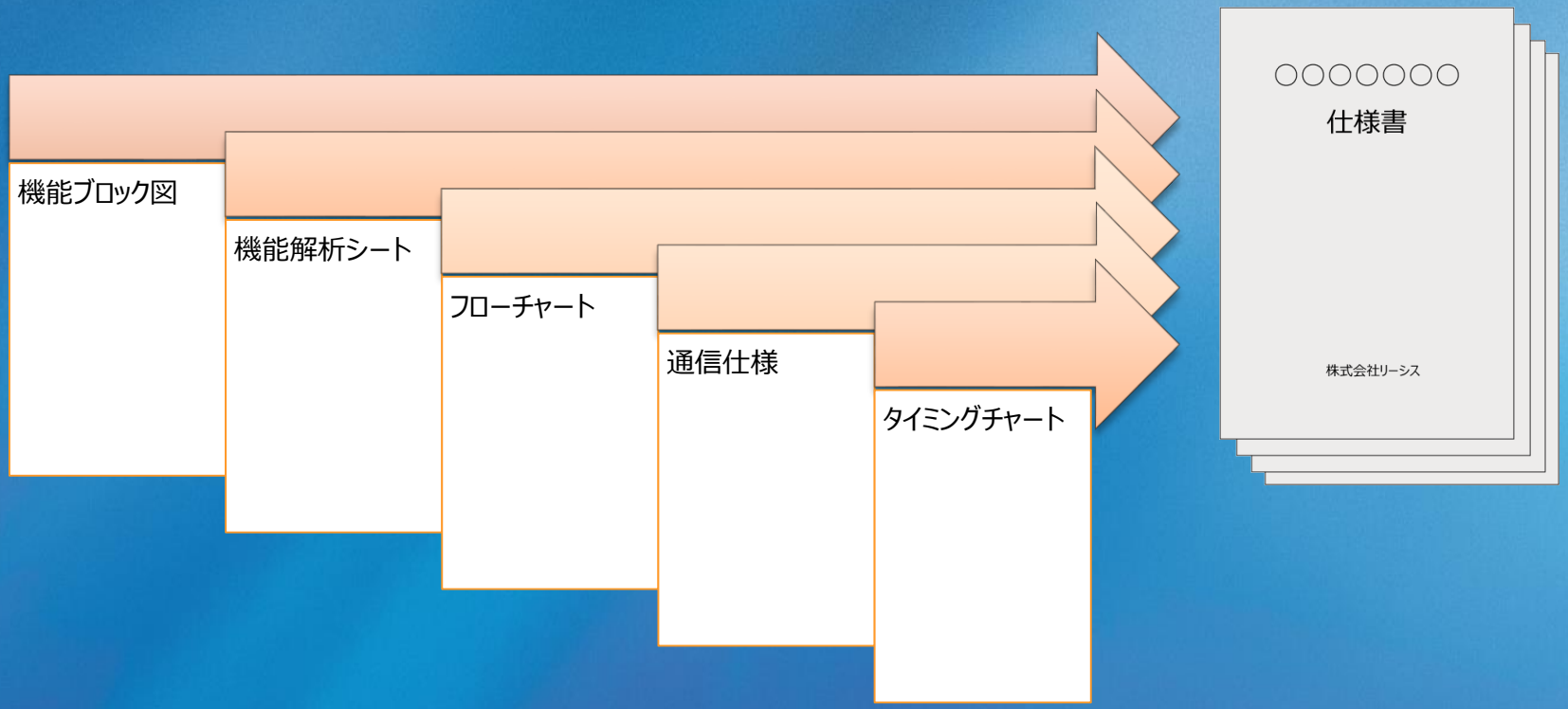


1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte	10byte	11byte	12byte	13byte
STX	command	size	Data0	Data1	Data2	Data3	Data4	Data5	Data6	Data7	BCC	ETX
0x02	0xA0	0x08	0x80	0x40	0x00	0x10	0x01	0x01	0x00	0x00	0xXX	0x03
STX	command	size	Data0	Data1	Data2	Data3	BCC	ETX				
0x02	0xA1	0x04	0x00	0x01	0x02	0x03	0xXX	0x03				

6. 入出力制御は計測器を使用して入出力のタイミングチャートを作成



7．最終的に解析結果をまとめて提出ドキュメントの作成



- 実績
  - 事例1(部分的な解析)　検査装置
    - ユーザーで不具合がある場所を、ある程度特定できており、ピンポイントで該当部分を調査
    - OS入れ替えにより発生したアプリケーション不具合
    - 原因調査、対応策の提示
    - 24h(3日)
  - 事例2(部分的な解析)　デジタルカメラ調整治具
    - カメラモデルチェンジによる治具アプリケーションの解析
    - 再構築（対応OSの追加）
    - 96h(12日)

- 参考例
  - 事例3(全体的な解析) スキャナー
    - 既存のシステムを把握し新機能を追加したファームウェアを開発
    - 既存システムの改修も含む
    - 480ｈ(3か月)