CLUE リーダコンポーネントマニュアル

産業技術総合研究所
知能システム研究部門
空間機能研究グループ
大原　賢一
E-Mail: k-oohara(at)aist.go.jp

目次

1　コンポーネント概要

ロボットが作業をするためには知識と作業対象の位置・姿勢情報が不可欠である．多くの場合，これらの双方を同一の媒体で求めるのではなく，たとえば知識の場合は RFID，位置・姿勢についてはステレオカメラから得られる三次元情報を用いたモデルベースのマッチングなどによって獲得される．

RFID は無線通信で情報が得られるため，知識保存媒体としては有効な機能を提供している．しかし，無線通信による位置計測はさまざまな問題を抱えており，位置・姿勢までの獲得は困難である．

一方，ビジョンシステムでモデルベースト手法を人間が生活するような環境で適用する場合，すべての物体のモデルデータを用意することは困難であることから，認識以前の部分で問題を抱える．

こうした問題に対して，われわれは以下のような機能を持つマーカを CLUE（Coded Landmark for Ubiquitous Environment）として提案している．

- 操作対象に関連する知識の提供.
- 操作対象までの距離や姿勢の推定するための情報.
- 通常は不可視であり、特殊な処理で可視化

本コンポーネントはこうした CLUE の一例として，QR コードを取り上げ，その中に保存される情報と，QR コードの見た目の情報を獲得，提供することを目的とする．

本コンポーネントでは，CLUE の読み込み装置つまり CLUE リーダをデンソーウェーブ社製固定式二次元コードスキャナ QD25 において実現している．本コンポーネントの提供する機能を以下に示す．

- QR コードの読み込みとその表示
- QR コード 4 角のデータと QR コードの中身の情報の獲得とそれぞれの表示

2　システム構成

2.1　ハードウェア構成

本コンポーネントを動作させるためのシステムのハードウェア構成を以下に示す.

本コンポーネントはデンソーウェーブ社製固定式二次元コードスキャナである QD25 において取得される QR コードのデータを出力する．このデータのやり取りは，制御用コンピュータと RS232C 接続において行われる.

近年では，RS232C ポートを持つ PC も減ってきていることから，本マニュアルでは，RS232C によるコンポーネントの利用方法と，USB-シリアルコンバータによる利用の双方を考えるものとする（コンポーネントポート指定に影響するため).

なお，本マニュアルにおいて利用するコンポーネントは CPU として，VIA Esther processor の 1.5GHz, memory 1GByte を搭載した PC において作成・動作している．なお，USB シリアルコンバータは，Linux との親和性のよいチップを搭載している US232R-10(Compass Lab にて 3570 円で購入可能:2007/12/18 現在)を用いている.

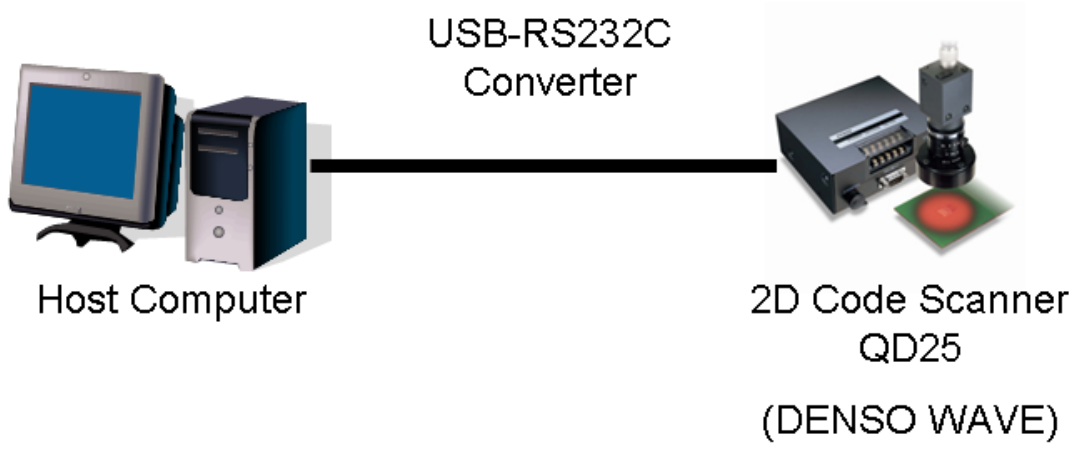

Fig.1 System Layout for CLUE Reader Component

2.2 ソフトウェア開発環境

本コンポーネントは，Linux 上での動作させることを想定している． しかし，異なる OS 上での動作も想定にいれ，OS 固有の部分（シリアル通信部分）と QD25 に依存する部分は分離されている．そのため，後述する SerialCom クラスのインタフェースと同様のインタフェースを Windows 上で実装するだけで，Windows 上での利用も期待できる（ただし，未検証）．

本コンポーネントを開発・動作検証を行った環境を以下に示す．

OS(Distribution)	Vine Linux 4.1
Kernel	2.6.16-0vl76.3
GCC	gcc-3.3.6-0vl7
RT Middleware Version	0.4.1
Programming Language	C++

3　コンポーネント構成

本コンポーネント群は以下のように構成される．

メインコンポーネント

- ClueReader コンポーネント

オプションコンポーネント

- ClueAnalyzer コンポーネント（QR コードの画像中での位置と QR コードの中身の情報を分離して出力）
- QRCodeViewer コンポーネント（QR コードの中身の情報の提示）
- PositionViewer コンポーネント（QR コードの四隅の情報を提示）

それぞれのコンポーネントの位置づけについて次に示す．

3.1　ClueReaderコンポーネント

本コンポーネントの主な機能は以下の通りである．

- QD25 の制御
- CLUE から得られる情報の出力

本コンポーネントは Fig.3-1 に示すような構成となっており，大きく RS232C シリアル通信部分と QD25 の制御コマンド API 部分から構成される．RS232C シリアル通信は，実装形態が OS に依存する．本コンポーネントは Linux 上での実装となっているが，同様のインタフェースを持たせれば，Windows 上での利用も可能である．（各クラスの詳細については添付資料のクラスリファレンスを参照のこと）

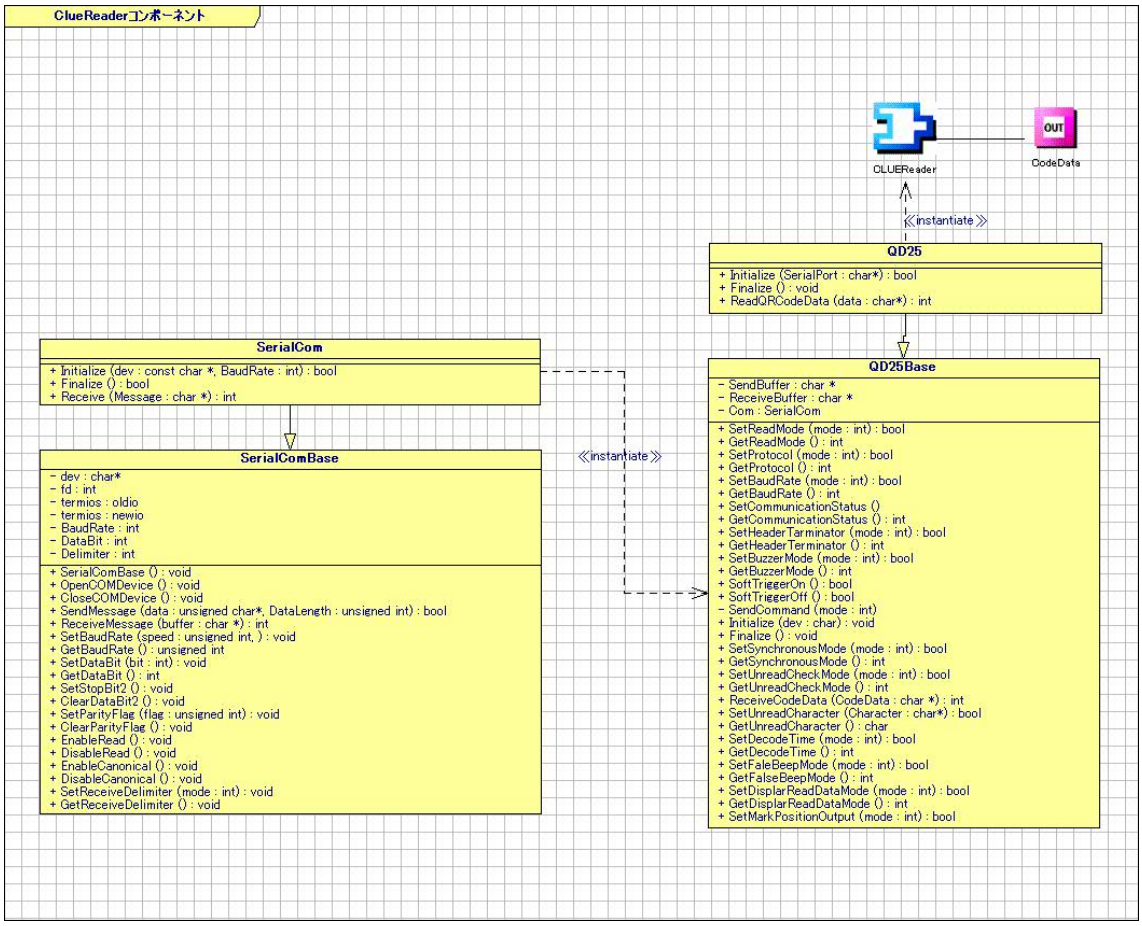

Fig.3-1 ClueReader コンポーネントの構成

本コンポーネントでは，QD25 クラスの設定を変更することにより，OutPort から QR コードに保存された情報を出力することも，加えて QR コードの 4 角のデータを返すことも可能である．モードの設定方法については，次章のコンポーネントの利用方法の項で説明を行う．本コンポーネントでは，CLUE リーダコンポーネントであることから，デフォルトの Configuration 設定では，出力は QR コードの 4 角のデータ（QR コードの画像中の位置）と QR コードの中に保存された情報の双方を出力するものとする．

コンポーネントのデータポートの詳細情報について Tbl.3-1 に示す．

Tbl.3-1.　ClueReader コンポーネントのデータポートの設定

データポート	ポートネーム	型	機能
OutPort	CodeData	TimedCharSeq	ClueReader からの出力 QR コードの情報のみ or QR コードの情報＋QR コードの 4 角の座標

注意点

ClueReader コンポーネントの後ろには, ClueAnalyzer コンポーネントも QRCodeViewer コンポーネントも双方の接続が可能であるが，不適切な動作モードで接続した場合の動作保証はできない．そこで，モードに適したコンポーネントの Inport に接続するように注意すること．

3.2　ClueAnalyzerコンポーネント

ClueReader コンポーネントからの出力は，以下のようになっている．

「QR コードの四隅の情報」＋「QR コードの中身の情報」

これらの情報は一続きのデータとしてくることから，この二つの情報を分離し，個々のデータの利用を可能とするようなインタフェースを有する．

コンポーネントのインタフェースとしては，ClueReader コンポーネントからの出力を受け付けるインポート 1 つと，QR コードの中身の情報の出力と，QR コードの位置情報を出力するアウトポートを 2 つ有するコンポーネントとなっている．本コンポーネントの詳細情報を Fig.3-3，Tbl.3-2 に示す．

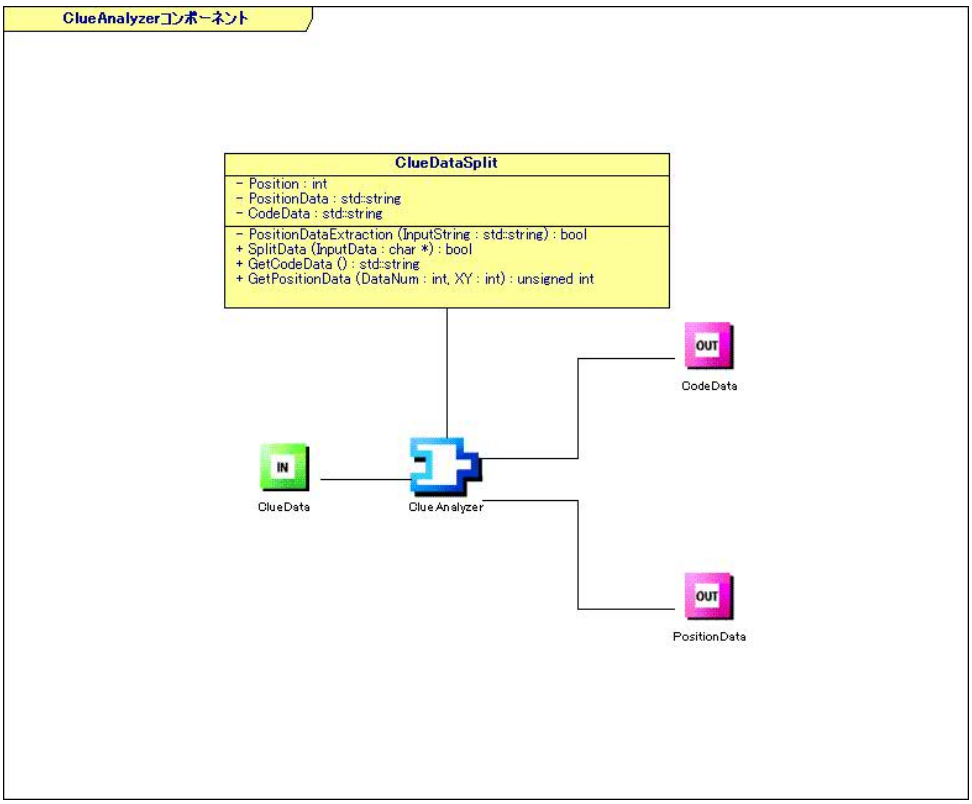

Fig.3-3.　ClueAnalyzer コンポーネントの構成

Tbl.3-2 ClueAnalyzer コンポーネントのデータポートの設定

データポート	ポートネーム	型	機能
InPort	ClueData	TimedCharSeq	ClueReader コンポーネントの OutPort と接続
OutPort	CodeData	TimedCharSeq	QR コードの中身の情報を出力
OutPort	PositionData	TimedShortSeq	QR コードの 4 角の座標を連続で出力

PositionData から出力するデータの並びは Fig.3-4 のようになっており， A 点 X 座標→A 点 Y 座標→B 点 X 座標→B 点 Y 座標→C 点 X 座標→C 点 Y 座標→D 点 X 座標→D 点 Y 座標の順番で出力する．

Fig.3-4. QR コードと出力データの対応付け

注意点

コードの中身を解析できた場合でも，四角のデータが正常に読み取れて射ないとき，その点の座標値は **999**として出力される．これはエラーコードを意味しているため，データ利用時に適宜適した処理をする必要がある．

3.3 QRCodeViewerコンポーネント

QR コードに保存された情報を提示するコンポーネント．ClueAnalyzer からの QR コードの情報を保存も可能であるが，ClueReader コンポーネントの設定時に QR コードの四角の情報を出力しないように設定した場合は, 本コンポーネントと直接接続することで, QR コードの中身を見ることができる. 双方に対して，共通のインタフェースとして，インポートをひとつ有する．本コンポーネントデザインを Fig.3-4 に，データポートの詳細を Tbl.3-3 に示す.

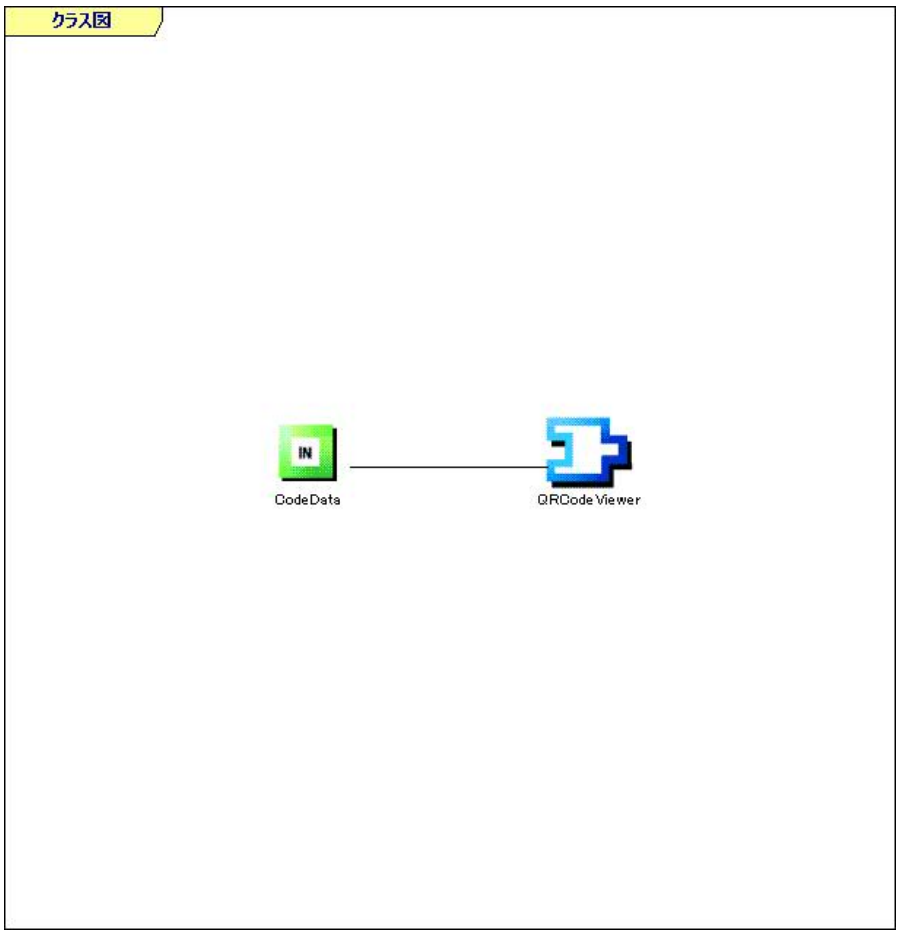

Fig.3-4 QRCodeViewer コンポーネントの構成

Tbl.3-3 QRCodeViewer コンポーネントのデータポート情報

データポート	ポートネーム	型	機能
InPort	CodeData	TimedCharSeq	QR コード情報の入力

3.4 PositionViewerコンポーネント

QD25 から出力される位置情報を表示するコンポーネント．本コンポーネントは ClueAnalyer から出力されるデータを利用することから，ClueAnalyzer の位置情報出力にあわせたインタフェースを有する．（実際に位置・姿勢情報を利用するためには，この四角のデータからカメラパラメータに基づいた位置・姿勢情報を導出する必要がある．）

QR コードの 4 角のデータは画像座標上での座標出力であり，この座標値から実世界座標系に直すためには，実世界での対応点の情報が不可欠であり，そのための情報として,QR コードの大きさを Configuration に設定できるようになっている．

本コンポーネントの構成を Fig.3-5 に，ポートの詳細情報を Tbl.3.4 に示す．

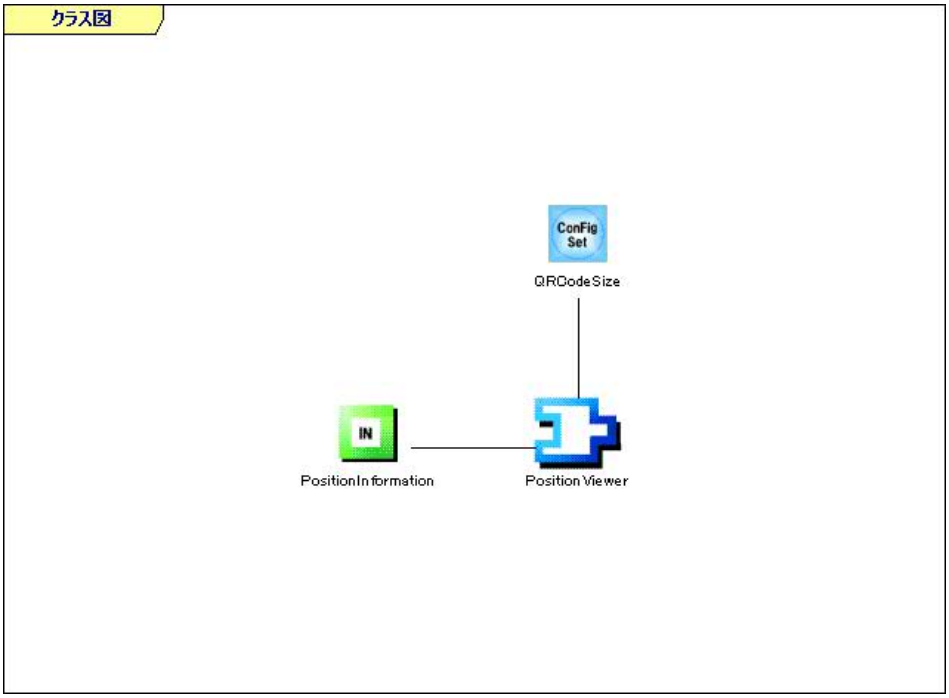

Fig.3-5. PositionViewer コンポーネントの構成

Tbl.3-4 PositionViewer コンポーネントのデータポート情報

データポート	ポートネーム	型	機能
InPort	CodeData	TimedShortSeq	QR コードの 4 角の情報の入力 4 点の X,Y データ計 8 個のデータが入力

また，Configuration の設定情報について Tbl.3-5 に示す．Configuration において複数の大きさの QR コードのサイズを設定可能である．

Tbl.3-5 PositionViewer コンポーネントの Configuration 情報

ConfigurationSet	
Default	0
Mark1	15
MARK2	20
MARK3	50

ここでは，任意のサイズの QR コードを利用できるようにするために，Configuration 設定の保存を別ファイルで行っている．変更方法の詳細はコンポーネントの次章で説明する．なお，事前に 15,20,50[mm]については，設定されている．

注意

ここでいうマークのサイズとは Fig.3-6 に示すマークの角と角の間の距離について示している．QR コード作成ソフトでマークのサイズを設定する場合, QR コードの周りの白いふちの大きさも含めてサイズを決定している場合が多いので，注意すること．

Fig.3-6　マークサイズの既定の仕方

4　コンポーネントの利用方法

4.1　シリアルポートの確認

コンポーネントの動作を確認する前に, QD25 と PC との結線を確認する. このとき, USB-シリアルコンバータによる接続であるか，RS232C ストレートケーブルによる結線であるかによって，動作方法が異なることから，きちんと確認を行う．動作しているシリアルポートを調べるためには

```
$dmesg | grep “tty”
```

とすればよい．このようにすると USB シリアルコンバータを用いている場合も含めて，以下のようにシリアルデバイスの一覧が出てくる．

```
serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
serial8250: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
00:09: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
00:0a: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
usb 2-1: FTDI USB Serial Device_converter now attached to ttyUSB0
```

この例では，USB シリアルコンバータを用いて QD25 との接続を行おうとしていることから，シリアル通信ポートの指定は“ttyUSB0”となる．

4.2　ソースファイルのコンパイル

はじめに本コンポーネント群が格納された圧縮ファイルの解凍を行う．

```
$tar xzvf ClueReaderComponent.tar.gz
```

解凍後「ClueReaderComponent」ディレクトリが生成される．本コンポーネントのソースファイルはすべてこの中に格納されている．このディレクトリを<CR_HOME>とする．

解凍後のディレクトリ構成は以下のようになる．各コンポーネントのディレクトリには，以下のファイルが存在する．

- コンポーネントソースファイル
- rtc.conf
- run.sh(コンポーネント起動スクリプト)

Configuration を持つ PositionViewer だけは上記に加えて，以下のファイルを有する．

- QRCodeSize.conf

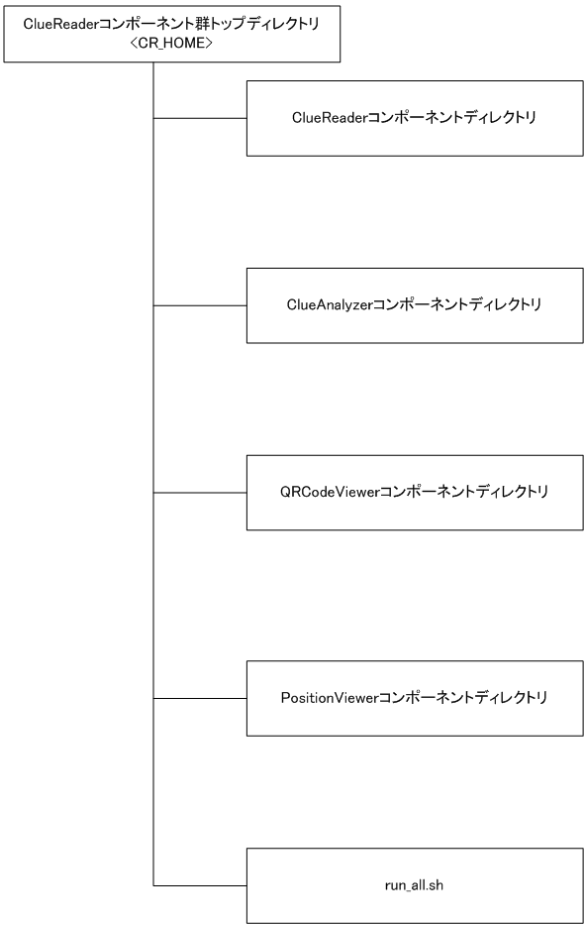

Fig.4-1 コンポーネント群ディレクトリ構成

配布媒体では，個々のコンポーネントはコンパイルされていない状態なので，まず個々のコンポーネントのコンパイルを行う．

4.2.1 ClueReaderコンポーネントのコンパイル

ClueReader コンポーネントをコンパイルする前に，ソースに対して，シリアルポートの設定を書き込む．初期状態では，“/dev/ttyUSB0”となっているため，もし同様の環境の場合は，この設定をする必要はない．

シリアルポートの設定は，”QD25.h”の 14 行目において

```
#define SERIAL_PORT        “/dev/ttyUSB0”
```

と書かれている部分を適宜，自身の仕様にあった形で書き換える必要がある．

また，初期設定では，QR コードの情報出力と，QR コードの 4 角の出力の両方を行うように設定されているが，もし，QR コードの情報だけをコンポーネントから出力させたい場合は，”QD25.h”の 25 行目を以下のように変更する．

```
//////////////////////////////////////////////////////////////////
//Output courner position mode
//////////////////////////////////////////////////////////////////
//#define OUT
```

設定を書き込んだ後で，ClueReader コンポーネントからコンパイルを行う．

```
$cd ClueReader
$make –f Makefile.ClueReader
```

正常にコンパイルが終了すれば ClueReader コンポーネントのコンパイルは終了である．

コンポーネント起動スクリプトは初期状態で実行権限がないので，実行できるように権限を変更する．

```
$chmod 777 run.sh
```

なお，この実行スクリプトの権限変更はすべてのコンポーネントにおいて同様に行う必要がある．

4.2.2 ClueAnalyzerコンポーネントのコンパイル

ClueReader と同様の方法でコンパイル可能である．この段階で ClueReader コンポーネントのディレクトリにいると想定する．

```
$cd ../
$cd ClueAnalyzer
$make –f Makefile.ClueAnalyzer
```

ここでも，エラーが出ずに正常にコンパイルできれば終了である．

4.2.3 QRCodeViewerコンポーネントのコンパイル

4.1.2 を終了した直後の場合，以下のように QRCodeViewer コンポーネントをコンパイルする．

```
$cd ../
$cd QRCodeViewer
$make –f Makefile.QRCodeViewer
```

エラーが出ずに正常にコンパイルできれば完了である．

4.2.4 PositionViewerコンポーネントのコンパイル

ここまでと同様に以下のようにすることで，コンパイルができる．

```
$cd ../
$cd PositionViewer
$make –f Makefile.PositionViewer
```

ここまですべてのコンパイルが正常に終了すれば，コンポーネントを利用可能となる．

4.3　コンポーネントの動作テスト

コンパイルが終了したら，本コンポーネント群の動作テストを行う．なお，ここではClueReader コンポーネントの出力を初期状態から変更せず，以下の出力になっていることを前提に説明する．

「QR コードの 4 角の座標」＋「QR コードの情報」

まず，すべてのコンポーネントを実行可能とする<CR_HOME>/run_all.sh に実行権限を持つように設定する．

```
cd <CR_HOME>
$chmod 777 run_all.sh
```

シリアル通信デバイスを制御するため，コンポーネント実行時にはスーパーユーザになる必要がある．

```
$su
```

スーパーユーザになったら，<CR_HOME>において，コンポーネント実行スクリプトを起動する．

```
#./run_all.sh
```

実行スクリプトを起動すると，各コンポーネントの処理内容表示用のウィンドウが生成される．

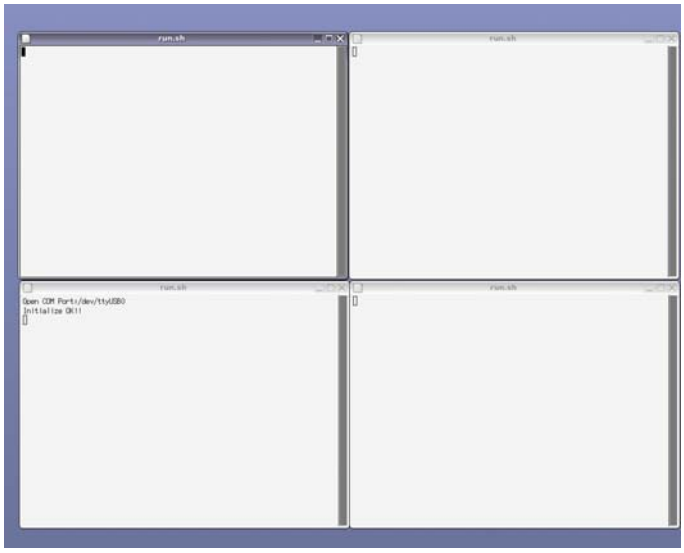

Fig.4-2 コンポーネント用のウィンドウが開いた様子

本コンポーネント群のコンポーネントはすべてで 4 つあることから，すべてで 4 つのウィンドウが生成される．正常に 4 つのウィンドウが表示されたれ，各ウィンドウ，特に ClueReader コンポーネント用のウィンドウにおいて

```
Initialize OK!
```

と出力されていれば，コンポーネントは正常に起動していることになる．もしこのウィンドウでエラーが出力されている場合の一番の要因として，スーパーユーザにならずにコンポーネントを実行していることが考えられるので，もしエラーが出ているならば，スーパーユーザであることを確認していただきたい．また以下のように出力されて止まってしまう場合，ハードウェアの設定が間違っているか，ハードウェアの結線ができていないことが予想されるので，確認していただきたい．

```
Open COM Port:/dev/ttyUSB0
```

つづいて，rtc-link 上でコンポーネントの結線を行う．Fig.4-2 のようにコンポーネントを結線する．

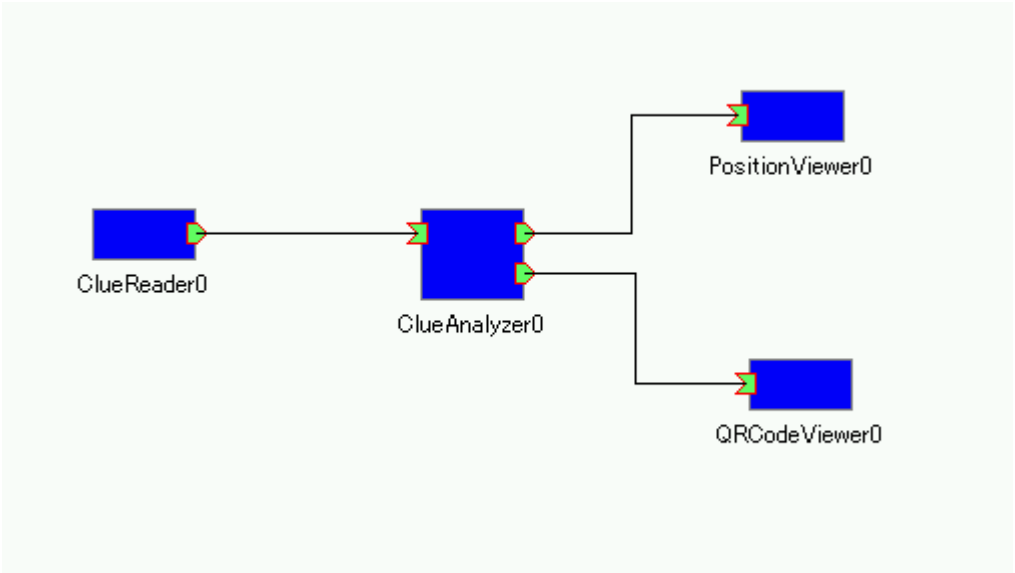

Fig.4-3　コンポーネントの結線

ClueReader コンポーネント以外のコンポーネントはデータ更新時にのみデータを取得する仕組みになっているため，すべてのコンポーネントを同時に起動可能である．そこで，rtc-link 上で，「All-Activate」を選択し，すべてのコンポーネントをアクティブにする．

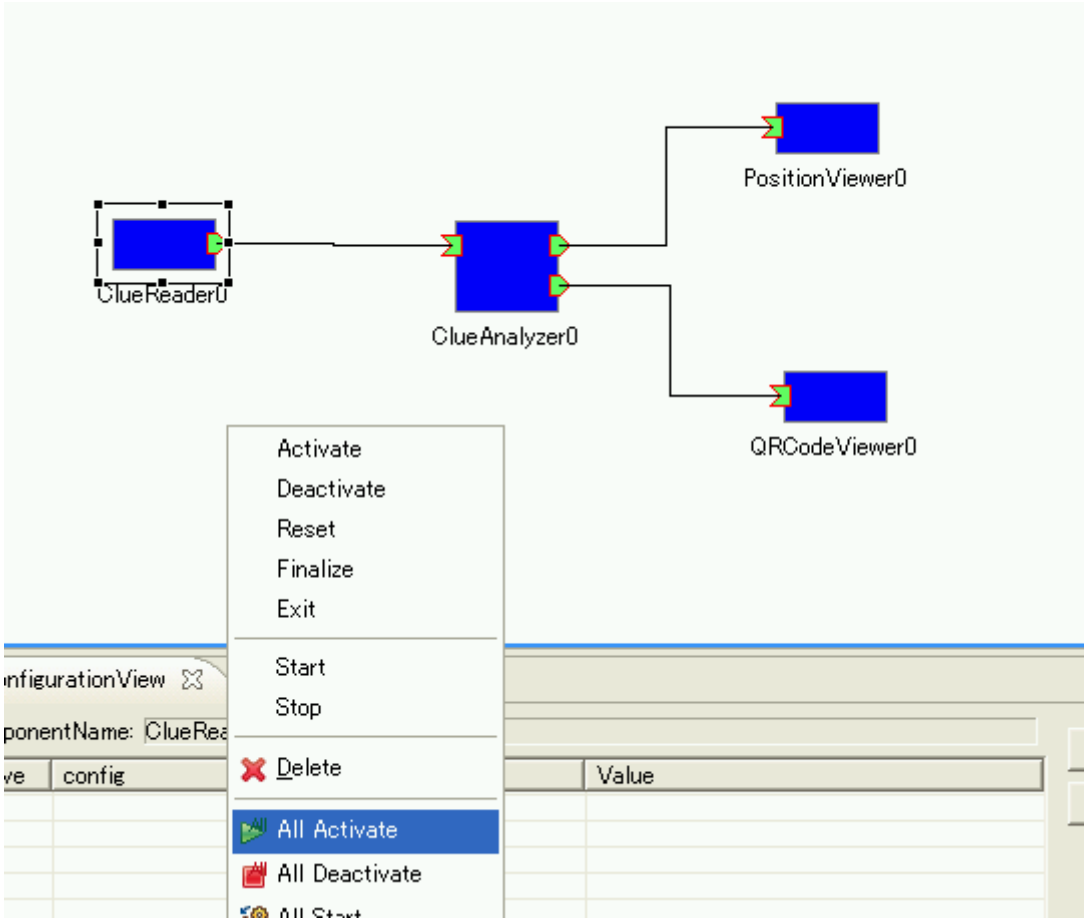

Fig.4-4 コンポーネントの Activation

コンポーネントを Activate 状態にし，QR コードリーダのカメラの視野内に QR コードが発見されると，ここのコンポーネントのウィンドウには以下のように表示される．

```
Receive Byte:55
Clue Code Length:55 Clue Data:[(149,123)(354,125)(352,329)(147,327)]1234567890ABCDEF

SoftTrigger ON
SoftTrigger OFF
Receive Byte:55
Clue Code Length:55 Clue Data:[(149,123)(354,125)(352,329)(147,327)]1234567890ABCDEF

SoftTrigger ON
SoftTrigger OFF
Receive Byte:55
Clue Code Length:55 Clue Data:[(150,123)(354,125)(353,329)(148,327)]1234567890ABCDEF

SoftTrigger ON
SoftTrigger OFF
Receive Byte:55
Clue Code Length:55 Clue Data:[(150,123)(354,125)(351,330)(148,327)]1234567890ABCDEF

SoftTrigger ON
SoftTrigger OFF
Receive Byte:55
Clue Code Length:55 Clue Data:[(150,123)(354,125)(351,329)(148,327)]1234567890ABCDEF

SoftTrigger ON
SoftTrigger OFF
Receive Byte:55
Clue Code Length:55 Clue Data:[(150,123)(354,125)(351,329)(148,327)]1234567890ABCDEF
```

Fig.4-5　ClueReader コンポーネントのウィンドウ出力

```
CodeData Before:[(149,122)(354,125)(352,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF

CodeData Before:[(150,123)(354,125)(351,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF

CodeData Before:[(149,123)(354,125)(352,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF

CodeData Before:[(149,123)(353,125)(352,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF

CodeData Before:[(149,122)(354,125)(353,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF

CodeData Before:[(149,122)(354,125)(352,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF

CodeData Before:[(149,122)(354,125)(352,329)(147,327)]1234567890ABCDEF

CodeData After:1234567890ABCDEF
```

Fig.4-6　ClueAnalyzer コンポーネントのウィンドウ出力

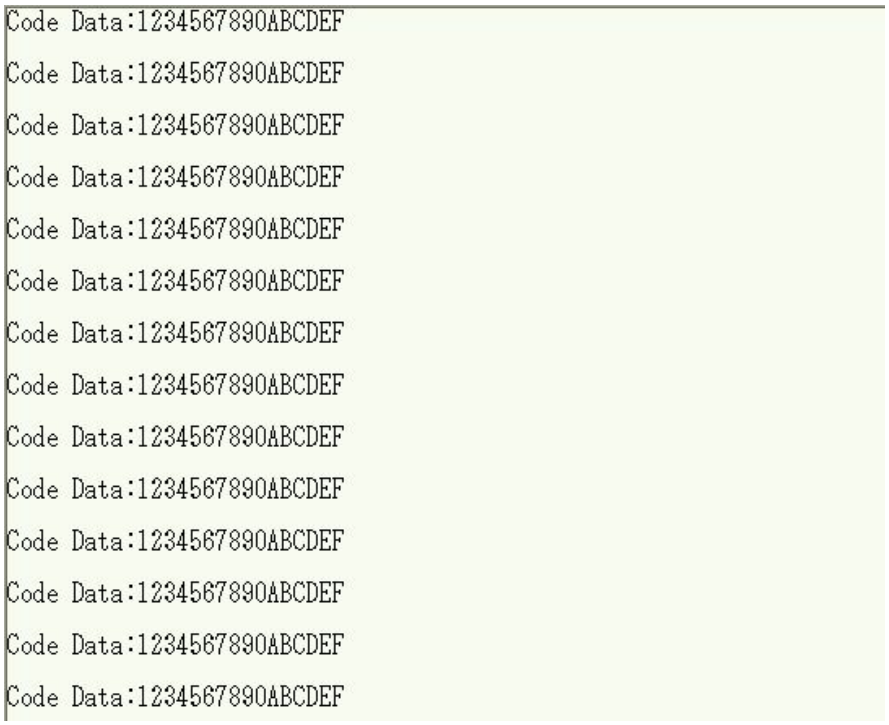

Fig.4-7　QRCodeViewer コンポーネントのウィンドウ出力

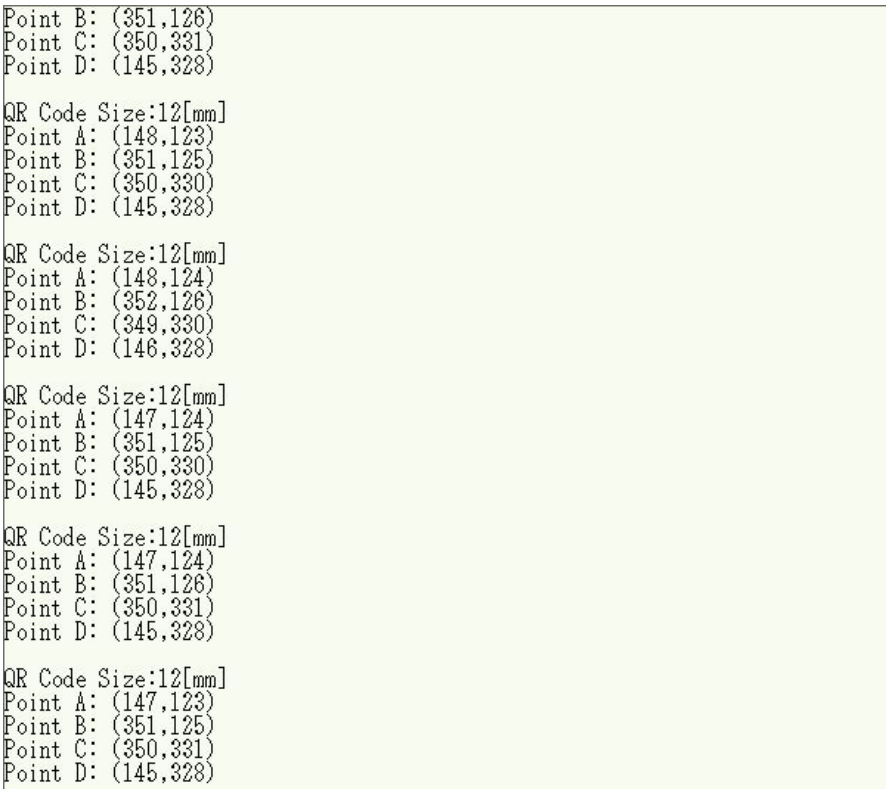

Fig.4-8　PositionViewer コンポーネントのウィンドウ出力

表示されているウィンドウにあるように，QRCodeViewer，PositionViewer コンポーネン

トのそれぞれに目的のデータが送信されている様子が見える．実運用を行うときには，QRCodeViewer の部分であればデータベースへの接続など，PositionViewer コンポーネントの部分では，カメラパラメータを用いることで，QR コードの 4 角の座標値から，実世界座標系への変換を行うことが可能である．

以上が本コンポーネントの機能である．

4.4 通常のQRコードリーダコンポーネントとしての利用

前節では，ロボット用途として想定される機能を持つコンポーネントの利用方法を示した．ここでは，ClueReader コンポーネントのモードを切り替えることで通常の QR コードリーダとしての利用方法について示す. 4.2.1 に示したように,「Output courner position mode」の部分をコメントアウトすることで, ClueReader からの出力が QR コードの情報だけになる．（モード変更時は，一旦すべてのファイルを clean してから再コンパイル行ってください.）この場合，結線するコンポーネントは QRCodeViewer を接続する．

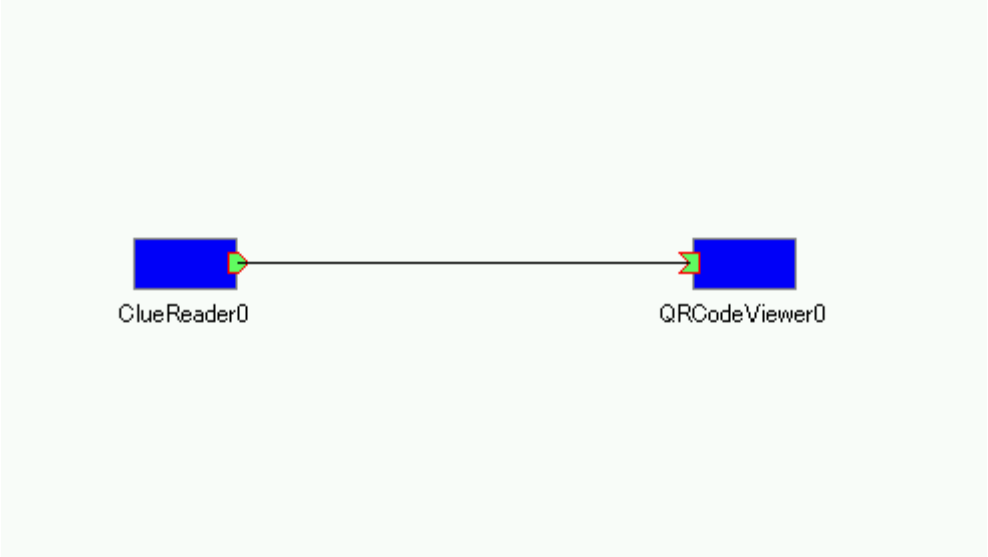

Fig.4-9 QR コードリーダ読み込みコンポーネントとしての結線

5　まとめ

本マニュアルでは，デンソーウェーブ社製 QD25 から得られる QR コードの情報と，QR コードの 4 角の情報を利用可能にするコンポーネント群の利用方法について示した．十分な検証を行っているが，もし不具合があった場合，RT ミドルウェアのメーリングリストに投稿していただきたい．

6　謝辞

本コンポーネントは，文部科学省の平成 18 年度科学技術振興調整費による「科学技術連携施策群の効果的・効率的な推進 環境と作業構造のユニバーサルデザイン」における研究成果物である．

改訂履歴

Ver.1.0：ClueReader コンポーネント全体の使用方法について記述