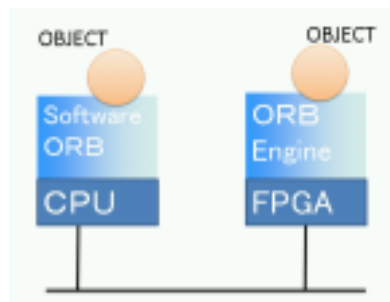


# ORB エンジン ユーザマニュアル



独立行政法人 産業技術総合研究所



## 内容

ORB エンジンとは.....	3
開発の目的.....	3
対応プラットフォーム .....	4
対応トランスポート(データ通信路) .....	4
配布物内容(ディレクトリ解説).....	5
機能(ランタイム).....	6
CORBA サーバント.....	6
機能(ツール類) .....	6
IDL コンパイラ (idlOrbe).....	6
IOR ジェネレータ (iorOrbe) .....	6
通信路ブリッジ (bridgeOrbe).....	7
CAN 用通信路ブリッジ(TCPtoCAN_Bridge) .....	8
性能.....	8
謝辞.....	9
更新履歴.....	10
連絡先.....	10

## ORB エンジンとは

ORB エンジンとは、ORB(Object Request Broker)の処理の一部(プロトコル変換部分)をハードウェア化して CPU の負荷を削減することにより CPU の動作周波数を上げることなく ORB の使用を可能にする技術・実装です。

## 開発の目的

現在、組み込み機器においては、ネットワークにつながって必要な情報を取得したり発信したりすることが高付加価値を生むための必須要件です。しかし、機器をネットワークにつなげる為のソフトウェア開発は、単体のコンピュータのソフトウェア作成に比べて原理的に困難が多いです。まず通信元と通信先の間でプロトコル(通信規約)を決め、プロトコルを処理するプログラムを実装する必要があります。更に、様々な通信障害が発生する事を想定して実装する必要があります。また、通信相手からの応答をお互いに待ちあうデッドロック状態に陥りやすく、デバッグが困難です。

以上のような並行・分散ソフトウェア特有の問題を解決する為に、企業の業務システムなどでは分散オブジェクト間通信を自動化する ORB 等のネットワークミドルウェア(通信ミドルウェア)の導入が進んでました。ORB を使用すると、ソフトウェアの開発者は実際の通信の詳細は記述せずに、ネットワーク上に分散するオブジェクト間のメソッド呼び出しを記述するだけで済みます。そのためバグの少ないシステムが開発可能となります。組み込みシステムの開発においても、近年の開発規模の増加による開発複雑性を緩和するために ORB の導入を進める動向があります。

ところが組み込み機器・ユビキタス機器の開発においては、限られた資源(CPU・メモリ)の制約条件の中で開発を行わなければならないことから、ORB を使用することが困難であることが多いです。

そのため ORB エンジンにおいては、処理をハードウェアにオフロードすることで CPU の動作周波数を上げずに ORB の導入が可能となり、高度にネットワーク通信を駆使するユビキタス機器・ロボット・産業用機器等の低消費電力化を実現します。オブジェクト間通信のハードウェア化によって、組み込み向けの低消費電力の CPU において ORB が使用可能になります。

## 対応プラットフォーム

ORB エンジン試用版は、Xilinx 社の提供する MicroBlaze ソフトコアプロセッサおよび PowerPC405 組み込みプロセッサでの動作を想定しています。以下の環境での動作を確認済みです。

- ハードウェア
  - Xilinx 社製 FPGA ボード
    - ✧ Spartan-3E スターターキット (S3E500)
    - ✧ MicroBlaze エンベデッド開発キット (S3A1800DSP)
    - ✧ ML403 プロセッサ開発ボード (V4FX12)
  - アットマークテクノ社製 FPGA ボード
    - ✧ SUZAKU (Spartan3 および Virtex2Pro, Virtex4FX)
  - x86 の PC/Linux 環境
  - Fujitsu マイコン搭載の CAN 学習ボード
  - ARM11 搭載の Armadillo 開発ボード
- 開発環境(PC 側)
  - Xilinx 社製
    - ✧ Embedded Development Kit (EDK) 9.2 もしくは 10.1
    - ✧ Integrated Software Environment (ISE) 9.2 もしくは 10.1
  - Eclipse 開発環境
    - ✧ <http://eclipse.org/>

## 対応トランスポート(データ通信路)

ほとんどの CORBA 実装は TCP/IP を通信路として使うようになっています。ORB エンジンは、TCP/IP に限らず、組み込みシステム開発でよく用いられる通信路を利用可能です。

ORB エンジンは、動作プラットフォームによりますが、以下のうちいずれか一つの通信路上で動作します。

- シリアル通信(RS232C)
  - Xilinx 社製 UartLite コア
    - ✧ 変調速度 115200 baud、データビット 8、ストップビット1、パリティ無
    - ✧ フロー制御なし
- イーサネット(UDP)

- Xilinx 社製 EmaLite コア
  - ✧ 100Base-TX
- CAN
  - 中日電子社製 UC245
  - 富士通社製 bits pot
  - Xilinx 社製 ECU コア
    - ✧ 通信速度100Kbps

既存の CORBA クライアントから ORB エンジンによるサーバントをつなげるためには、付属ツール「通信路ブリッジ (bridgeOrbe)」を使います。

※トランスポートはプラグイン・アーキテクチャにより実装されていますので、今後、要求に応じて様々なトランスポートを使用することが可能です。

## 配布物内容(ディレクトリ解説)

ソースコード配布パッケージにおけるディレクトリ構成を説明します。

- 1、UML  
UML による仕様書
- 2、arch  
各アーキテクチャ(動作プラットフォーム)に対応する為のソースファイル置場
- 3、bin  
コンパイル済み実行バイナリの置場
- 4、idl  
テスト・サンプルで使用する IDL ファイルの置場
- 5、idlc  
IDL コンパイラのソース
- 6、src  
ORBE コア部分のソースディレクトリ
- 7、test  
テストプログラムの置場

## 機能(ランタイム)

### CORBA サーバント

- CORBA メッセージを解釈し、適切にアプリケーション・コードを呼び出し、応答の CORBA メッセージを返す。
- CORBA メッセージは GIOP1.2 をサポート。
- 詳細
  - ◇ ライブラリとして提供 (libOrbe-\*.a)
  - ◇ 実装言語:C 言語
  - ◇ FPGA 上の MicroBlaze もしくは PowerPC プロセッサで動作。
  - ◇ メモリフットプリント最大 32KB 程度(コンフィギュレーションにより変化)

## 機能(ツール類)

### IDL コンパイラ (idlcOrbe)

IDL ファイルをコンパイルし、サーバントに必要なソースコードを生成するツールです。

#### 使用方法

\$ idlcOrbe IDLfile
---------------------

IDLfile: IDL を定義したファイルを指定します。

#### 使用例)

\$ idlcOrbe LED.idl
---------------------

※ #include などのディレクティブに対応するためには、事前にプリプロセッサで処理してください。

例)

\$ cpp originalLED.idl
------------------------

### IOR ジェネレータ (iorOrbe)

ORB エンジン用の IOR ファイルを生成するツールです。

## 使用方法

```
$ iorOrbe interfaceName ipAddress port
```

interfaceName: IDL で定義したインターフェース名を指定します。

ipAddress : bridge を起動しているマシンの IP アドレスを指定します。

port: bridge のリッスンポート番号を指定します。

## 使用例)

```
$ iorOrbe LED 127.0.0.1 5656
```

IDL の定義で module を使用している場合は、次のように interfaceName を”/“で区切って module 名とインターフェース名を指定します。

```
$ iorOrbe jp/go/aist/LED 127.0.0.1 5656
```

## 通信路ブリッジ (bridgeOrbe)

- (概要)JavaIDL や omniORB などで作成したクライアントプログラムからのデータを、シリアルポートや UDP/IP を通して ORB エンジンと通信するためのツールです。ほとんどの CORBA 実装は TCP/IP を通信路として使うようになっています。このツールは、既存の CORBA クライアントから ORB エンジンによるサーバントをつなげるために使います。
- (注意点)
- (詳細)
  - ☆ シリアルポート、および UDP/IP の通信路の選択は、bridge の第1引数で指定します。

## ◆シリアルポートの場合

### 使用方法

```
$ bridgeOrbe TCPToUART COMport listenport
```

COMport: シリアルポートの通信ポート名を指定します。

listenport: bridge のリッスンポート番号を指定します。

## 使用例)

```
$ bridgeOrbe TCPtoUART COM1 5656
```

#### ◆UDP/IP の場合

##### 使用方法

```
$ bridgeOrbe TCPtoUDP listenport targethost targetport
```

listenport: bridge のリッスンポート番号を指定します。

targethost: ORB エンジンの IP アドレスを指定します。

targetport: ORB エンジンのリッスンポート番号を指定します。

##### 使用例)

```
$ bridgeOrbe TCPtoUDP 5656 192.168.1.10 6767
```

#### CAN 用通信路ブリッジ(TCPtoCAN\_Bridge)

- (概要)JavaIDL や omniORB などで作成したクライアントプログラムからのデータを、CAN を通して ORB エンジンと通信するためのツールです。このツールは、既存の CORBA クライアントから ORB エンジンによるサーバントをつなげるために使います。
- (注意点)
- (詳細)

```
$ TCPtoCAN_Bridge.exe Baudrate [-d]
```

Baudrate: CAN の通信速度(Kbps)を指定します。

-d: これを指定すると CAN を通して送受信されるデータのダンプを出力します。

##### 使用例)

```
$ TCPtoCAN_Bridge.exe 100
```

#### 性能

##### ● 参考データ

- ORBE 処理時間測定
  - ◇ WireShark により、リクエスト UDP パケットが発信されてから、返事の UDP パケットが帰ってくるまでの時間を測定
    - MicroBlaze on SPARTAN-3E



- 0.36ms (平均値)
- 条件:システムクロック 50MHz
- PowerPC405 on ML403
  - 0.36ms (平均値)
  - 条件:PPC クロック 300MHz、バスクロック 100MHz、キャッシュ有効

#### 謝辞

ORB エンジンの研究開発は、平成 19 年度産業技術研究助成事業(NEDO)による援助を受けて行われたものです

## 更新履歴

日付	バージョン	コメント	作成者
2008/08/20	1.0	● ORB エンジン試用版リリース	大川猛 小川秀樹 久保田貴也 河並苗穂子
2008/10/10	1.0.1	● EDK10.1 対応 FPGA プロジェクトの追加 ● 配布物ツリーの修正	大川猛 小川秀樹 久保田貴也 河並苗穂子
2009/4/1	1.0.2	● ツールの解説追加 ● 配布物ツリーの修正	大川猛 戸田賢二 小川秀樹 久保田貴也
2009/4/13	1.0.3	● 著作権表記の修正	大川猛

## 連絡先

産業技術総合研究所 情報技術研究部門 戸田 賢二、大川 猛

住所: 〒305-8568 茨城県つくば市梅園 1-1-1 中央第 2 2-1E 311 室

電話番号: 029-861-5875 FAX: 029-861-5909

電子メール: orbe@m.aist.go.jp