## OpenCR 1.0

## アプリケーション開発マニュアル

# 【サンプル】



## はじめに

OpenCRはロボットアームなど計算量の多いターゲットと連携して開発する のにむいた試験開発用途のマイコンボードえす。実績の多いArduinoプラット フォームを用いていますので、先行する研究資源を活用することができます。さ らに、演算装置や電源や入出力まわりは競合のArduinoシリーズボードと比較し て強化されていますので、他では実現が難しいような重めなアプリケーション の開発にも対応できるポテンシャルをもっています。

この小冊子はまずOpenCRの性能について紹介し、続いてロボットアーム Open Manipulator XとArduinoエコシステムを連動したロボットアームアプリ ケーション開発していく構成になっています。



Оре	nCR の用意	1
	OpenCRとは2	
	性能諸元4	
	ボード配置と使用上の特性6	
	OpenCR の環境構築8	) 1
	PC Downloader アプリケーション11	
Ardı	uino IDE 開発環境の構築	-14
	Arduinoとは15	•
	Arduino IDEの環境導入(Linux)16	
Ardı	uino IDEでのアプリケーション開発	-25
	サンプル1:基本アプリとバイナリ出力26	
	サンプル2:Arduinoの基本と拡張シールドの活用31	
	サンプル3:ロボットアームを操作する36	
	この章で利用したArduino関数リスト40	)

## OpenCR の用意

OpenCRはロボット開発に適したマイコンボードです。開発プラットフォーム は幅広い採用実績をほこるArduno開発環境採用しています。Arduinoは学術・ 研究分野のみならず近年では産業界での導入もすすでおり、今後も大きく普及 が進んでいくものと考えられています。

この章ではOpenCRをつかう下準備として、その性能と特徴をご紹介し、その後、環境構築手順をご説明します。

#### OpenCRとは



OpenCRは中小規模のロボット開発に適したインターフェイス/電源まわりの 実装を含んだマイコンボードです。多数の採用実績を誇るArduinoプラット フォームに則った開発を行う事ができますので、先行研究の資産を活用した り、引き継ぎが容易であるなど試作開発用途に最適な特長を備えています。

#### OpenCR自体の開発情報も公開

元となったArduinoエコシステムと同様、OpenCRも開発情報がオープンソー スになっており、設計情報はインターネット上から自由に取得することがで き、実力に自信のあるユーザーは独自の拡張をすることも可能となっています。 他のArduinoシリーズと比較するとOpenCRはArduino系のなかでも特に強力 なチップセットであるSTM32系を採用しています。今回のようなロボットアーム 系プロジェクトの開発では順運動学や逆運動学などの軌跡計算、さらには各種 センサーとの連携など計算資源が多く必要となる局面がすくなくありません。 STM32系の強力な計算能力をもつOpenCRであれば他のArduino系ボードでは実 現が難しいような計算もこなすことができます。

#### Arduino UNO系規格にあったシールド対応

Arduinoシリーズの醍醐味といえば豊富な先行開発事例の活用にありますが、 OpenCRもそのメリットを最大限に活かすことができます。シールド(拡張ボー ド)の中でも特に普及率が高いUNO系のソケットレイアウトになっていますの で、データロギングに有効なSDカードシールドなど有力なシールドの開発資源 を活用できるという特性も引き続き備えています。

#### ロボット開発に適した豊富なインターフェイス

UNO規格ソケットの横には数多くのインターフェイスが備わっています。ッ メインターゲットであるOpen Manipulator Xに合わせた、TTL/RS485のコネク タは各3ポート用意されています。加えて、UART系が2系統、JTAG、CAN、 GPIOのポートも備わっています。多くのボードではこれらはシールドで拡張す るような機能ですが、標準で備わっておりかつ他のポートを占有しないため空 いたポートはセンサーやロギングなどの機能拡張にまわすことができます。 I

マイクロコントローラ	STM32F746ZGT6 / 32-bit ARM Cortex®-M7 with FPU (216MHz, 462DMIPS) Reference Manual, Datasheet		
内蔵センサー	Gyroscope 3Axis, Accelerometer 3Axis, Magnetometer 3Axis (MPU9250)		
デジタル I/O	32 pins (L 14, R 18) *Arduino connectivity 5Pin OLLO x 4 GPIO x 18 pins PWM x 6 I2C x 1 SPI x 1		
アナログ入力	ADC Channels (Max 12bit) x 6		
通信ポート	USB x 1 (Micro-B USB connector/USB 2.0/ Host/Peripheral/OTG) TTL x 3 (B3B-EH-A / Dynamixel) RS485 x 3 (B4B-EH-A / Dynamixel) UART x 2 (20010WS-04) CAN x 1 (20010WS-04)		
外部UI (LED、ボタン)	LD2 (red/green) : USB communication User LED x 4 : LD3 (red), LD4 (green), LD5 (blue) User button x 2 Power LED : LD1 (red, 3.3 V power on) Reset button x 1 (for power reset of board) Power on/off switch x 1		

入力電源	5 V (USB VBUS), 7-24 V (Battery or SMPS) Default battery : LI-PO 11.1V 1,800mAh 19.98Wh Default SMPS : 12V 4.5A External battery Port for RTC (Real Time Clock) (Molex 53047-0210)		
必要電力	*12V max 4.5A(SMW250-02) *5V max 4A(5267-02A), 3.3V@800mA(20010WS-02)		
サイズ	105(W) X 75(D) mm		
重量	60g		

Arduinoの標準的駆動電源は5V電源ですが標準の12Vアダプタ 注: ーから電源供給される場合、そちらを優先します。

<u></u> .	12Vアダプターと外部電池を両方接続している場合、電池を簡
注:	易なUPS(無停電電源)として機能させることができます。

#### ボード配置と使用上の特性

#### アナログ/デジタル

OpenCRはArduinoシールドの中でも広く普及しているArduino Uno系のピン マップと互換性のあるコネクタ構造をもっています。 ピン0〜21はArduino Uno と同じピンであり、それぞれのピンはデジタル場合を除いて5V5mAまでドライ ブ可能です(全体では120mA)。

#### 動作LED

OpenCRは4つのLEDを備えています。いわゆる"Lチカ"のような簡易な表示器 としての利用から、パラメーター表示まで幅広い利用が可能です。これらLEDは Arduinoピン22-25にマッピングされますので、アプリケーションを開発する際 はオーバーラップがないよう注意してください。



#### **GPIO**

OpenCRに18ピンの共通GPIO拡張コネクタがあります。アナログ/デジタル と同様、こちらGPIO系もArduinoのGPIOピンにマップされます。対応関係は下 表のとおりです。



### Arduino IDEでのアプリケーション開発

いよいよこの章から、具体的アプリケーションの開発手法をご紹介していき ます。まずはあらかじめ構築されたスケッチ(Arduioではソースコード類をそ う呼称します)からビルドする方法と、それを実際にダウンロードする方法を ご紹介します。Arduinoの醍醐味である数多くの拡張ボードと連携したサンプル を作りながら、開発手法を身につけていただきます。sサンプルコードの中にあ るサーボモーターのユーティリティもあわせてご紹介します。

#### サンプル1:基本アプリとバイナリ出力

それでは早速、最初のアプリケーションを導入してみましょう。まずは、 ハードウェアマニュアルでも紹介した、Processing製GUIとの標準連携ツールを 対象としてとして作業をすすめます。

[ファイル]->[スケッチ例]

->[OpenManipulator]->[example]->[Chain]->[open\_manipulator\_chain]の順で選択しアプリケーションを読み込みます。



無事読み込まれたら、下図のような画面がでます。今回のアプリケーション はロボットアームを操作するための複数のファイルで構成されていますが、基本 的に編集するのは最初に表示される[open\_manipulator\_chain]のみです。他の ファイルはProcessingのGUIやコントローラーから操作コマンドをうけとるもの です。より高度なアプリケーションが必要な場合に調整を検討してください。



27

#### ターゲットボード設定

つぎにターゲットボードの設定を行います。[ツール]=>[ボード]を選択し、対象を[OpenCR]にします。(下図左)

#### 通信ポート設定

スケッチのアップロードし、シリアル通信をするため、通信ポート設定をす る必要があります。[ツール]=>[ポート]を選択し、対象を/dev/ttyACM0 または COM1と、OpenCRを接続したポートに指定してください。(下図右)



すべての設定を確認したら、上のメニューバーにある「➡」のアイコンをク リックしてください。コンパイルの後、OpenCRへのアップロードされます。コ ンパイルには若干の時間がかかります。無事にコンパイルが完了すると、以下の ようにコンパイル結果が表示されます。ここでエラーが発生した場合は、ログに 表記された適切な対処をしてください。

emx_with_touch   Arduino 1.8.9					
	2			Ð	
omx_with_touch lnltDemo();	demo.h	processing.h	remote_controller.h		
<pre>pinMode(TouchPin }</pre>	1, INPUT); /	//タッチセンリー	を有効にします。	0	
void loop()					
		0			
ボードへの書き込みが	完了しました。				
グローバル変数は41700 openor_ld ver 1.0.2 openor_ld_main >> File name : /var/fo file size : 255 KB Open port UK Clear Buffer Start Clear Buffer End >> Board Name : OpenCF Board Ver : 0x1702 Board Rev : 0x0000 >> flash_enase : 0 : 1 flash_write : 0 : 1 CRC OK 19C3F0A 19C3 [OK] Download	0/(4 F Ø)RAM9 2 olders/02/5 20800 20800 20800 20000 L.054000 se 1.369000 se 3F04 0.0080	を使用しています。 46lp7s57l93kzy c c 00 sec	dhjZdtyzr0000gn/T/a	rduino_l	
-					

60

/dev/cu.usbmodem14121@OpenCR Board

コンパイルできる状態であれば、バイナリファイルをローカルに保存するこ とも可能です。前章で紹介したダウンローダーなどのツールを使えば、現場での 書き換えが用意に行えます。

バイナリ出力をするには、「メニュー」=>「スケッチ」=>「コンパイルした バイナリを出力」で可能です。保存先はスケッチとおなじディレクトリで、元 ファイルが「ownled.ino」とき、出力ファイル名称は

「ownled.ino.OpenCR.bin」となります。

open_manipulator_chain   Arduino 1.8.9 🛛 😑 🖨 🙆
ファイル 編集 スケッチ ツール ヘルプ
◆●●● 検証・コンパイル Ctr.+R マイコンボードに書き込む Cul+J open_mani 書込装置を使って書き込む Ctr.+Shil +J ンバイルしたパイナリを出力 Ctrl+Alt+S * Copy right スケッチのフォルダを表示 Ctrl K * Licensed * you may ni * You may
* You may optain a copy of the License at
<pre>* http://www.apacho.org/licenses/LICENSE-2.6 * * Jrless required by applicable law or agreed to in we * distributed under the License is distributed on an * WITFOUT WARRANTIES OR CONDITIONS OF ANY KIND, either * See the License for the specific language governing * limitations under the License. ************************************</pre>
/* Authors: Darby Lim, Tye-Jong KIM, Eyan Shim, Yong-L (()))
/dewittyACM0@0penCR Board_OpenCR Boot opper

#### サンプル2:Arduinoの基本と拡張シールドの活用

この節では一歩進んでArduinoをつかったアプリケーションをつくってみま す。まず手始めにOpenCRをつかって、外部センサーをトリガーにイベントを八 発生させてみましょう。

#### Arduinoの基本構造

何も実行をしない、最も基本的なArduinoスケッチは以下のようになります。 処理を記述する部分は大きく分けて「初期化部分」「ループ部分」にわかれてい ます。ライブラリや定数を設定する場合はこの二つの外に記述します。(後述)



#### 初期化部分(setup)

起動後、最初に実行される処理内容です。変数や周辺機器、I.Oの初期化を担う部分です。

#### ループ部分(loop)

Setup完了後に実行されます。名称の通り、内容が繰り返し実行されるパート です。末尾までいくと、loop関数の先頭まで飛び、同じ処理を繰り返します。

#### 拡張ボードの用意

概要でご紹介しますた通り、Arduinoの規格に則った設計のボードはシールド という拡張ボードにて機能を増やすことができます。Arduino公式はもとより、 サードパーティ製も多く存在しています。

今回は手頃な価格と使用感で研究開発での採用実績が多い、Groveシリーズを 使った開発を行います。具体的な構成はOpenCRにGroveBaseシールドを接続 し、そのD8ポートにGroveタッチセンサーを接続します。

左がArduino UNO規格のソケットに挿入する「Grove Base シールド」、右が、Grove タッチセンサーです。(※本セットには付属していません。別途 WEBショップ等でご購入ください)





OpenCRはArduino規格に則った設計がされていますので、シールドをそのま ま利用するこができます。下図のようにピン配置ソケットに差し込むと対応し たピンが。タッチセンサーはD8に接続しています。

