

平成 24 年度卒業論文

クラウドネットワークロボット
「MYCAL」の開発

Development of cloud network robot
"MYCAL"

函館工業高等専門学校 情報工学科 5年

東海林研究室 奥山純弥 中島真太郎

目次

1. 序論	
1.1 背景	1
1.2 英文アブストラクト	1
2. 「MYCAL」制御機構について	
2.1 開発環境	2
2.2 仕様	5
3. 「MYCAL」の動作機構について	
3.1 仕様	7
3.1 素材	8
4. プログラムについて	
4.1 開発環境	10
4.2 仕様	10
4.3 「MYCAL」との接続	10
5. 動作実験	
5.1 実験概要	12
5.2 結果	12
5.3 問題点	12
6. まとめと今後の展望	
6.1 まとめ	13
6.2 活用方法の模索	13
参考文献	14
付録	
付録1 ロボット制御プログラム	16
付録2 制御情報送信プログラム	17

1 章 序論

1.1 背景

近年、産業用ロボット以外にも様々なロボットが人間の社会に進出してきている。さらに人間社会との共生に関する研究も盛んになっている。例えば、函館高専とはこだて未来大学は、「IKABO」[1]と呼ばれているロボットを用いた広報活動をしている。ここで「IKABO」はPCによって制御されるダンスロボットである。しかし「IKABO」を制作するためには、多くの人員と大きな資金を要した。更に「IKABO」はスタンドアローン[2]なシステムであり、ネットワークには接続されていない。そこで我々は安価でクラウドネットワーク[3]に接続可能なイカ型ダンスロボット「MYCAL」を開発した。

1.2 英文アブストラクト

Recently, various robots have come into the human society as well as industrial robots. For instance, Hakodate Future University and Hakodate National College of Technology have been conducting the advertisement using a robot called "IKABO", which is the dancing robot controlled by a PC. However, Creating "IKABO" required many staffs and great expense. Moreover, "IKABO" is the stand-alone system and isn't connected to the network. Therefore, we have developed a cheap and network-connected robot called "MYCAL". "MYCAL" is connected to the cloud network using Bluetooth, so we can easily control "MYCAL" through the network.

Key words: Bluetooth, Cloud network, Regional vitalization

2章 「MYCAL」の制御機構について

2.1 開発環境

開発用 PC の OS として Ubuntu 12.04、開発ツールとして MPLAB X[4]と Hi-Tech C を用いた。PIC ライターには PICKIT3[5]を使用した。

PIC に PIC16F648A[6][7]を使用し、モータドライバに TA7291P[8]、Bluetooth モジュールに送受信モジュールに 80FG990 で拡張した RBT-001[9]を使用した。

MPLAB X は、マイクロチップ社製マイクロコントローラおよびデジタルシグナルコントローラ向けのアプリケーション開発用ソフトウェアプログラムである。このソフトウェアは統合開発環境と呼ばれ、組み込みマイクロコントローラのコード開発向けに、1 つに統合された環境を提供するものである。以下にプロジェクトを作成して PIC に書きこむまでの使用方法について順を追って記述する。

(1)プロジェクトを作成する。

- ① 「Create New Project」を選択し[次へ(N)>]をクリックする。
- ② 「Device:」で PIC16F648A を選択し[次へ(N)>]をクリックする。
- ③そのまま[次へ(N)>]をクリックする。
- ④ 「Select Tool」で「Hardware Tool」のところで「PICKIT3」を選択して[次へ(N)>]をクリックする。
- ⑤コンパイラの選択で「HI-TECH PICC」を選択して[次へ(N)>]をクリックする。
- ⑥ 「Project Name:」でプロジェクト名を入力して「Finish」をクリックする。

(2)プログラムソースを作成する。

- ①ツールバーの「New File」をクリックする。
- ②エディター画面にソースを書き込む。
- ③ツールバーの「Save File」をクリックし、ファイル名を入力してソースを保存する。

(3)コンパイルをする。

- ①ツールバーの「Run」→「Build Project」をクリックしてコンパイルをする。
- ②下にコンパイル結果のウインドウ画面が出てくるので、エラーが出ていたらソースを訂正し再びコンパイルをする。

(5)PIC に書き込む。

- ①PIC ライターをストレートケーブルで接続して PIC をセットする。
- ②ツールバーの「Make and Program Device(ソースファイル名)」をクリック。
- ③画面下のウインドウに出力結果が表示され、エラーがあったなら配線をチェック。
エラーがないなら PIC にプログラムが書き込まれたことを意味する。

また RBT-001 とは、29×29(mm)という小さい基板内にアンテナを搭載した Bluetooth 通信規格の組込用モジュールである。本体にアンテナを搭載しており通信距離は約 30m まで可能で、RS232C 通信を仮想化して Bluetooth 機器間で無線通信を行うことができる。UART 側のシリアル通信速度は最大 921.6kbps である。また、5V の電源電圧で RBT-001 用の 3V の電源を作って供給するために 80FG990 で拡張している。RBT-001 のロジック信号の電圧レベルは 3V なので 5V の回路とはそのまま直結できないが、80FG990 によって 5V を 3V の信号レベルに電圧調整できる。80FG990 で拡張した RBT-001 の外観とピン配置を図 2-1、表 2-1 に示す。

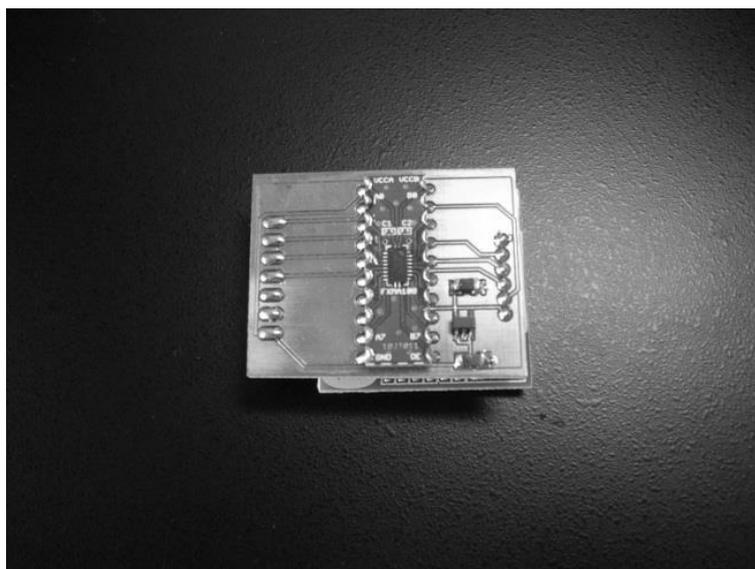


図 2-1. 80FG990 で拡張した RBT-001 の外観

表 2-1. RBT-001 のピン配列

ピン番号	名称	方向	
1	TX	0	URAT - 送信データ
2	RX	1	URAT - 受信データ
3	CTS	1	URAT - 送信可能(アクティブ Low)
4	RTS	0	URAT - 受信可能(アクティブ Low)
5			何も接続しない
6	GND	-	電源 GND
7	Vcc	1	電源電圧+5 ~ +6

2.2 仕様

PICがPCからBluetoothを介して送られてきた制御情報に従いTA7291Pを通じてモータを作動させる。PCからの制御情報の受信にはRBT-001を用いることによりシリアル通信が可能となっている。全体的な制御のフローチャートを図2-2、回路図を図2-3に示す。これらをブレッドボード上に配置・配線する。ロボット制御プログラムのソースコードは付録1に示す。

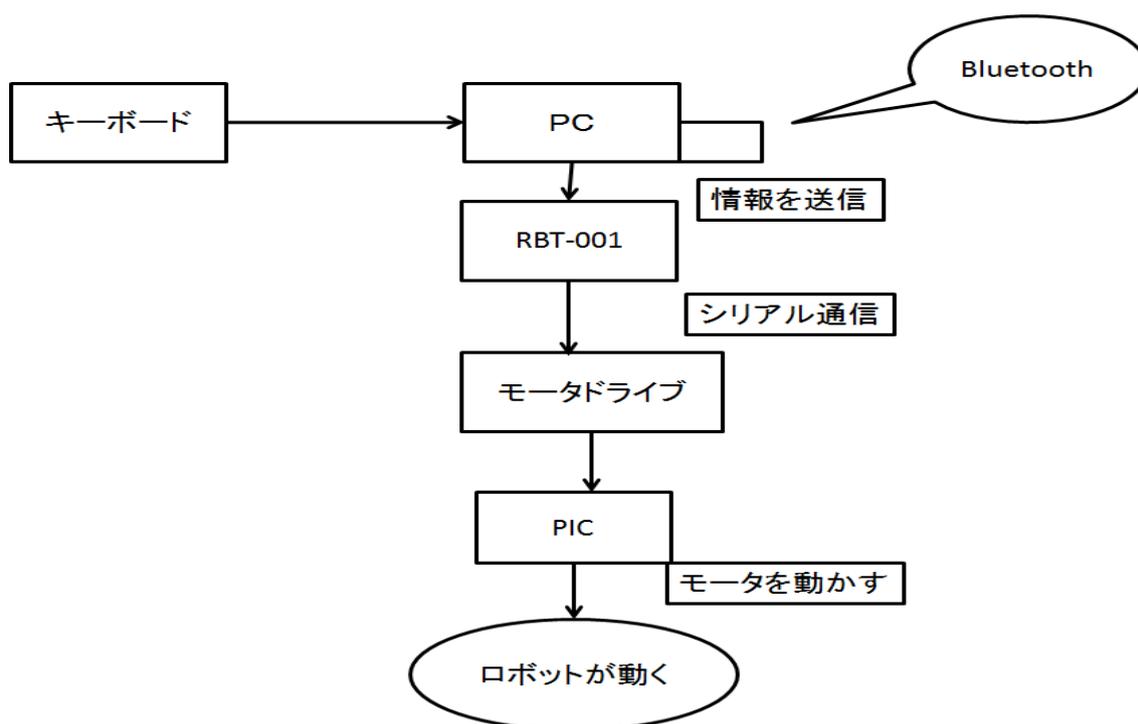


図 2-2. フローチャート

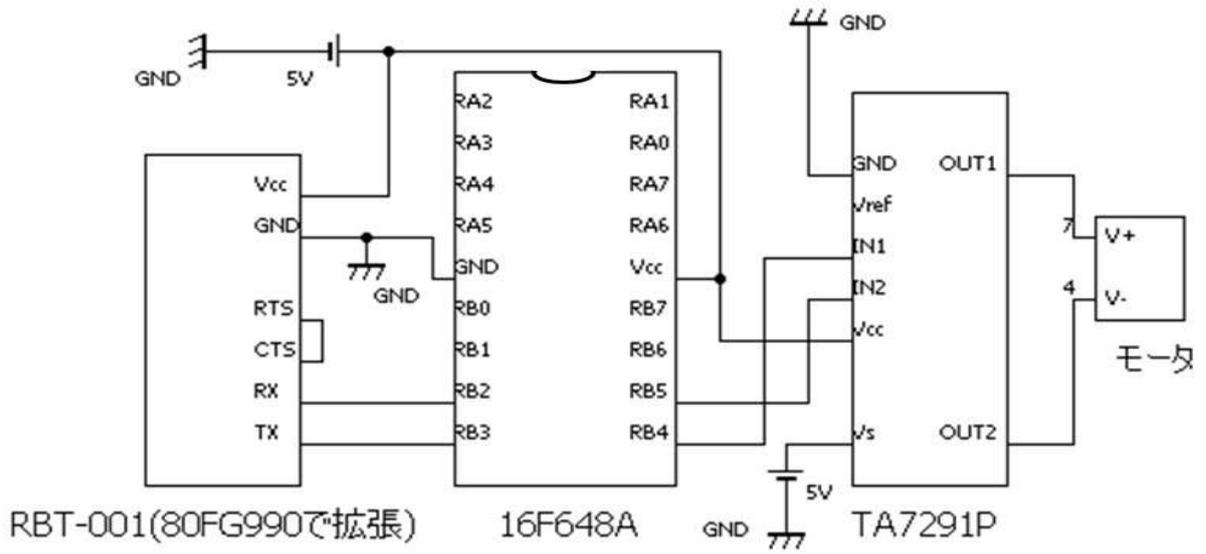


図 2-3. ロボット制御の回路図

3章 「MYCAL」の動作機構について

3.1 仕様

「MYCAL」はクラウドネットワーク[5]上の動作情報をもとにダンスのような動きをするロボットである。「MYCAL」は足を6本持っていて、クラウドネットワーク上のデータに応じて、まるでダンスを踊っているように動き出す。「MYCAL」の外装の設計図を図3-1、外観を図3-2に示す。

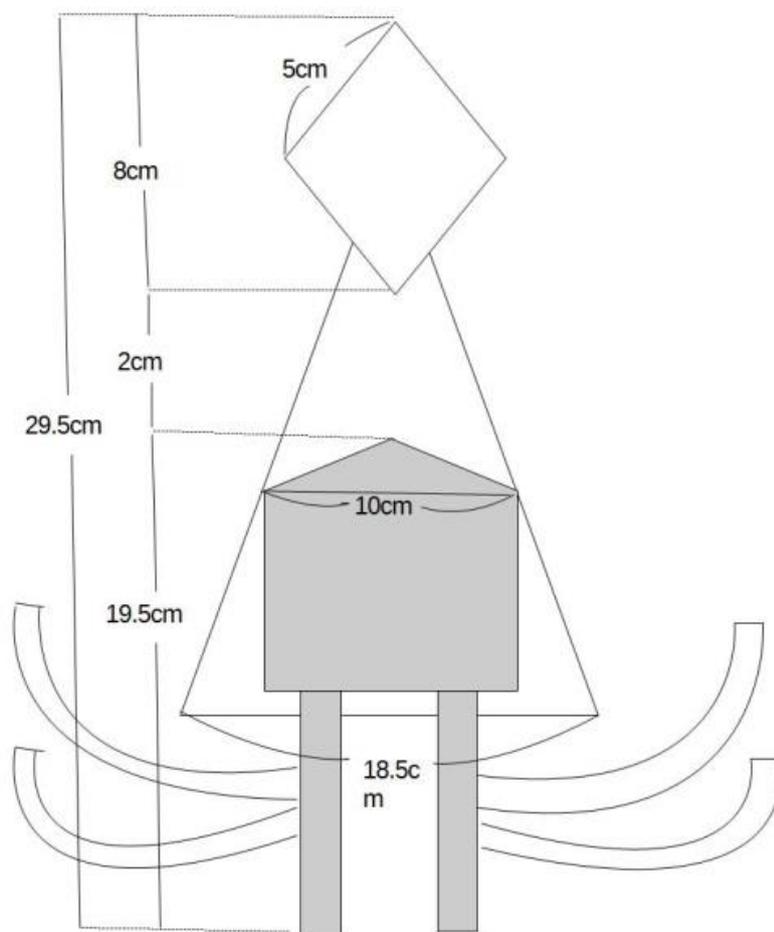


図 3-1. 「MYCAL」の外装設計図

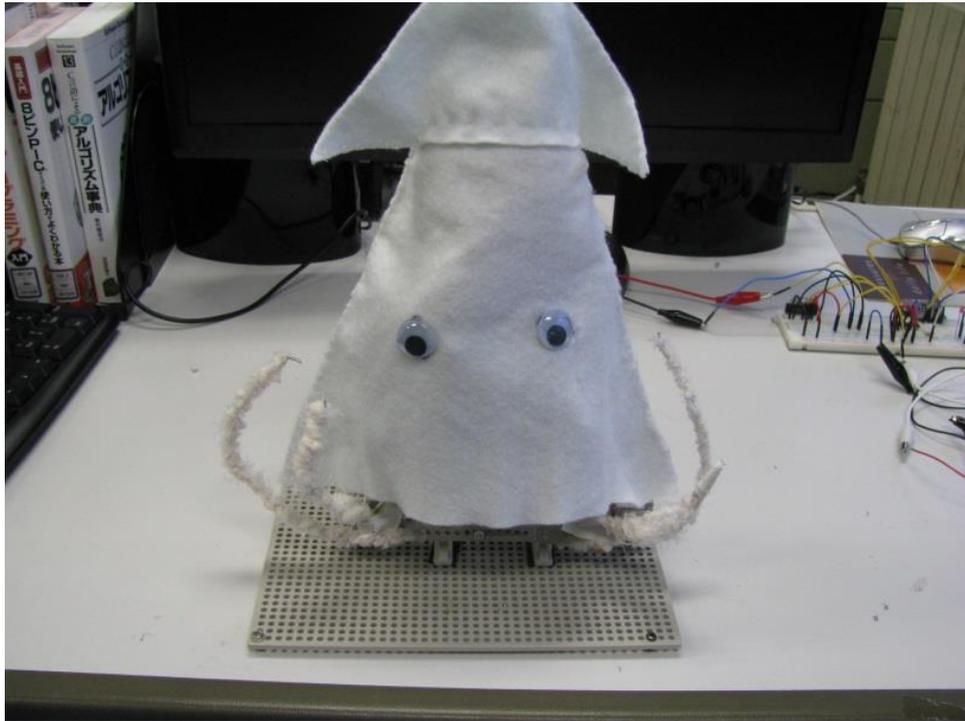


図 3-2. 「MYCAL」の外観

3.2 素材

「MYCAL」の外装の素材には布を使用している。足の素材には針金を使用している。この針金をリンクアームとギアボックス[10]を用いて動作させる。リンクアームとギアボックスを用いてクランク機構[11]を構成する。クランク設計図を図 3-3、クランク外観を図 3-4に示す。

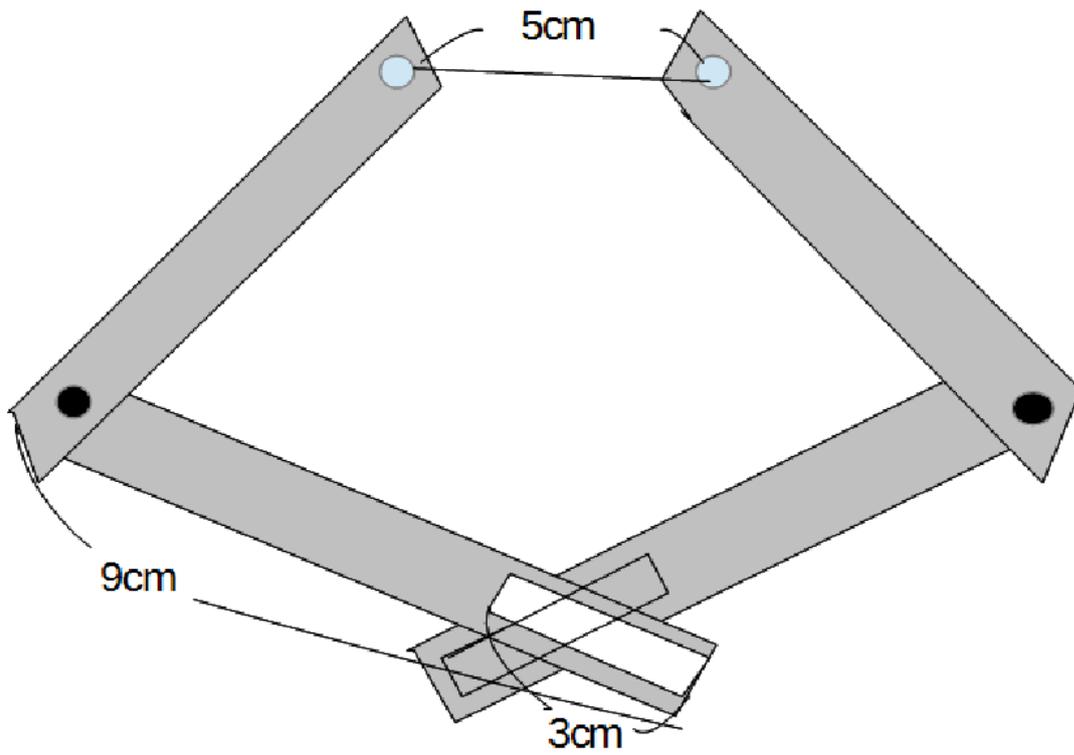


図 3-3. クランク設計図

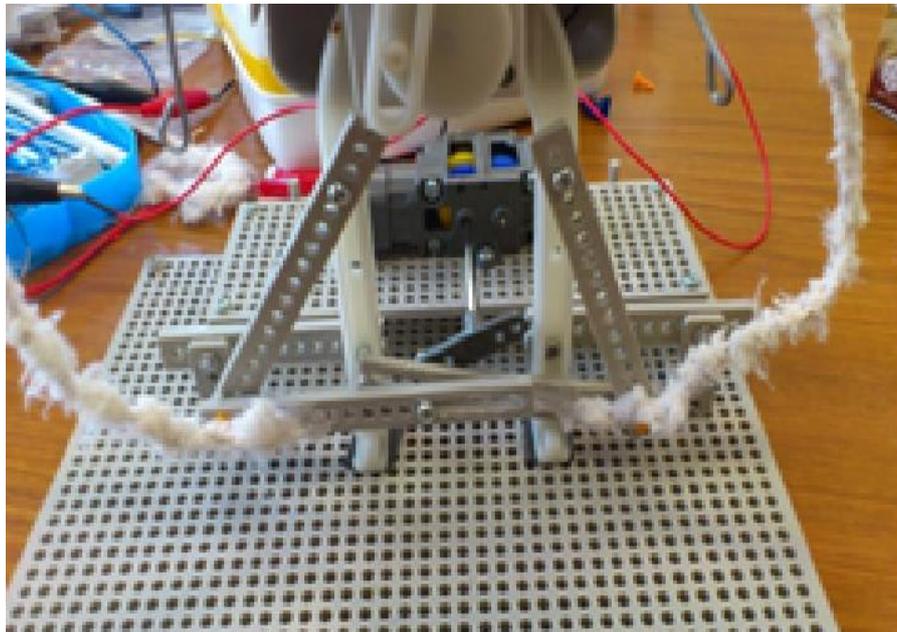


図 3-4. クランクの外観

4 章 送信側のプログラムについて

4.1 開発環境

制御情報送信プログラムの開発環境として OS に Ubuntu Linux 12.04、開発言語 に C 言語を使用した。また Bluetooth 通信用のライブラリは BlueZ[12][13]を使用している。

4.2 仕様

- (1) RFCOMM ポートを使用し仮想シリアルポートを作成する。
- (2) PC のキーボードからデータを入力する。
- (3) PC が Bluetooth を介して BT モジュールに制御情報を送信する。
- (4) (2)~(3)を繰り返す。

4.3 「MYCAL」との接続

- (1) PC の Bluetooth デバイスと RBT-001 を起動する。
 - ①PC の USB デバイスに Bluetooth デバイスを接続する。今回は BT-MicroEDR2X を使用した。
 - ②「MYCAL」の制御回路に電源を接続し RBT-001 の電源が入ったことを確認する (RBT-001 の LED が緑に光る)。
- (2) ペアリングを行う。
 - ①Bluetooth の設定ダイアログにある「新規デバイスの追加」をクリックする。
 - ②ウィザードが出てくるので「進む(F)」をクリックする。
 - ③しばらくするとデバイスが一覧に出てくるので「EasyBT」を選択する。
 - ④ペアリングが成功すると「設定」ダイアログにデバイス名とコンセントマークが表示される。
- (3) 仮想シリアルポートを作成する。
 - ①端末を起動する。
 - ②“\$ hcitool scan” で接続先のデバイスとアドレスを確認する。
 - ③“\$ sudo rfcomm connect 0 アドレス” で RBT-001 と接続する。接続に成功すると “dev/rfcomm0 to アドレス on channel 1” と表示される(図 4-1)。

```
matsui@matsui-H61H2-M5: ~
matsui@matsui-H61H2-M5:~$ hcitool scan
Scanning ...
    00:17:A0:01:6D:FF      EasyBT
matsui@matsui-H61H2-M5:~$ sudo rfcomm connect 0 00:17:A0:01:6D:FF
[sudo] password for matsui:
Connected /dev/rfcomm0 to 00:17:A0:01:6D:FF on channel 1
Press CTRL-C for hangup
█
```

図 4-1. 仮想シリアルポートの作成

5 章 動作実験

5.1 実験概要

ここで実際にロボットの動作実験を行う。ロボットの動作確認は表 5-1 に沿って行った。

表 5-1 テスト項目
実験項目

- (1) PC と RBT-001 が正常にペアリングできているか。
- (2) Bluetooth で正常にデータを送信できているか。
- (3) 送られてきたデータを正常に受信できているか。
- (4) 受信データをもとに正常に動作をしているか。

5.2 結果

全てのテスト項目を満たした。

5.3 問題点

実験を通じて次の様な問題点が浮かび上がった。まず、安価なモータを使用しているため動作が必要以上に速くなってしまった。また、足の本数を増やすとモータに負荷がかかり動かなくなってしまうため、動く足の本数を 4 本しか付けることが出来なかった。

6章 まとめと今後の展望

6.1 まとめ

本研究では安価でクラウドネットワークに接続可能なイカ型ダンスロボット「MYCAL」を開発した。なお、本研究は平成 25 年 1 月 26 日に仙台高専で開催された「第 18 回高専シンポジウム」[14]において発表された。

6.2 活用方法の模索

「MYCAL」はまだクラウドネットワークに接続されていないため、クラウドネットワークに接続した際の活用方法を模索する必要がある。活用方法の一つとして函館の地域活性化が挙げられる。例えば PR 動画を作成して YouTube などの無料投稿動画サイトにアップロードする、GPS と連携する等が挙げられる。

参考文献

[1] IKABO とは

<http://www1.atwiki.jp/ikabo-wiki/pages/15.html>

[2] スタンドアローンとは

<http://ja.wikipedia.org/wiki/%E3%82%B9%E3%82%BF%E3%83%B3%E3%83%89%E3%82%A2%E3%83%AD%E3%83%BC%E3%83%B3>

[3] クラウドコンピューティングとは

<http://ja.wikipedia.org/wiki/%E3%82%AF%E3%83%A9%E3%82%A6%E3%83%89%E3%82%B3%E3%83%B3%E3%83%94%E3%83%A5%E3%83%BC%E3%83%86%E3%82%A3%E3%83%B3%E3%82%B0>

[4] MPLAB IDE

www.microchip.com/

[5] PICKIT 3

<http://www.ne.jp/asahi/air/variable/picmel/integration/write/pickit3/index.html>

[6] 8ピン PIC マイコンの使い方がよくわかる本, 後閑哲也, 株式会社技術評論(2010).

[7] C言語による PIC プログラミング入門, 後閑哲也, 株式会社具術表鵜論社出版(2002).

[8] モータドライバ IC を使う

http://www9.plala.or.jp/fsson/NewHP_elc/H8/H8_53mterDrv.html

[9] RBT-001

http://park11.wakwak.com/~microtechnica/cgi-bin/goodslist.cgi?mode=view_detail&genre_id=00000028&goods_id=00000001

[10] 株式会社タミヤ

<http://www.tamiya.com/japan/index.htm>

[11] クランク機構とは

<http://kotobank.jp/word/%E3%82%AF%E3%83%A9%E3%83%B3%E3%82%AF%E6%A9>

%9F%E6%A7%8B

[12] BlueZ

<http://www.bluez.org/>

[13] Bluetooth programing in C with BlueZ

<http://people.csail.mit.edu/albert/bluez-intro/c404.html>

[14] 第 18 回仙台高専シンポジウム

<http://kosensymposium18.ccn.sendai-nct.ac.jp/>

付録 1. ロボット制御プログラム

```
#include <pic.h>
#include <htc.h>
#include <stdio.h>
//-----
// void  putchar(char c)と char getch(void)は HI-TECH C のサンプルコードからコピーする。
// サンプルコード: /usr/hitec/picc/9.83/samples/serial/bitbang/serial.c
//-----
__CONFIG(FOSC_INTOSCIO & WDTE_OFF & PWRTE_OFF & MCLRE_OFF &
BOREN_OFF & LVP_OFF & CPD_OFF & CP_OFF);

void main() {
    INTCON = 0;
    INIT_PORT;
    TxData = 1;
    RxData = 1;

    while (1) {

        RB5 = 0;
        RB4 = 0;
        __delay_ms(1000);

        while (RxData);
        putchar('a');

        RB5 = 1;
        RB4 = 0;
        __delay_ms(1000);

        while(RxData);
        putchar('a');
    }
}
```

付録 2. 制御情報送信プログラム

```
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <termios.h>
#include <fcntl.h>
#include <unistd.h>

int main()
{
    struct termios oldtio, newtio;           //シリアル通信を行う
    int fd;                                  //シリアル通信のファイルポインタ

    const char dev[] = "/dev/rfcomm0";      //シリアルデバイス
    fd = open( dev, O_RDWR );               //シリアルデバイスオープン

    if( fd < 0 ){
        perror( dev );
        exit( -1 );
    }
    tcgetattr( fd, &oldtio );               //古いシリアル設定を退避
    newtio = oldtio;                         //新しい設定に古い設定をコピー
    newtio.c_cflag = B9600 | CS8 | CLOCAL | CREAD;

    while (1){
        char buf[256];
        int read_size;

        unsigned char input;
        input = getchar();
        printf("write %c¥n",input);
```

```

write(fd,&input,1);
input = '¥n';
write(fd,&input,1);

read_size = read(fd, buf, sizeof(buf));
printf("read: %d byte¥n",read_size);
buf[read_size]='¥0';

if ( read_size > 0 ){
    printf("%s¥n",buf);
} else if ( read_size == 0 ){
    break;
}
}

tcsetattr(fd, TCSANOW, &oldtio); //古いシリアル設定を復帰
close(fd);

return 0;
}

```