災害被災地探査ロボットの研究

＜大震災を忘れない　－　本校課題研究の紹介＞

荒川　昇
長野県岩村田高等学校　電気科･電気情報科

１．はじめに

平成 23 年３月１１日に「東日本大震災」が発生し、東北地方を中心に過大な損害を与え、多くの被災者と犠牲者が出た。マスコミでその惨状をみて、工業高校に在籍している私達にも何かできないかと考え、「災害救援ロボット」をテーマに研究することを考えた。

本研究は、「災害時に役に立つロボットとは何か」を考え、被災地を探索するセンサ搭載ロボットシステムを研究製作したのでその様子を報告するものである。

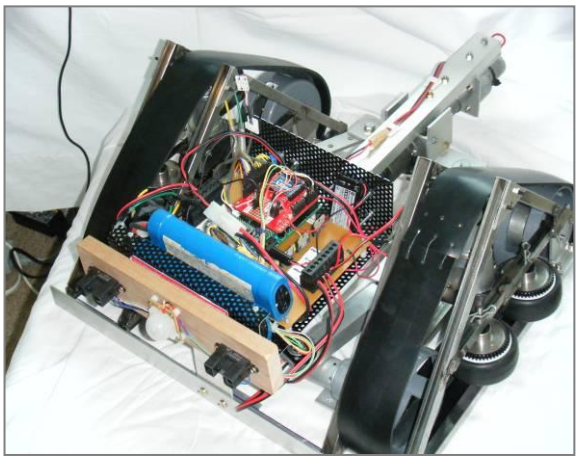


図１　製作した被災地探査ロボット

２．本研究におけるロボット仕様の検討

災害救助ロボットとは地震や水害等の災害が起こった際に、被災した人間を救助することを目的として作られたロボットである。現在開発が進められているものの多くは要救助者の探索を目的としており、瓦礫や建物内に入れるような特殊な移動機構やセンサが搭載されている。主な種類としては歩行型、キャタピラ型、またこれらの複合型等がある。しかし、それ以外にも種類はたくさんあり、開発者によって違いがある。

本研究において、まず、災害支援に必要なロボットの機能について考えた。

・人を救助する機能
・人に代わって物を運ぶ機能
・瓦礫（ガレキ）の中から怪我をしている人をセンサで見つける機能
・荒れ地を自在に進むことのできる仕組み

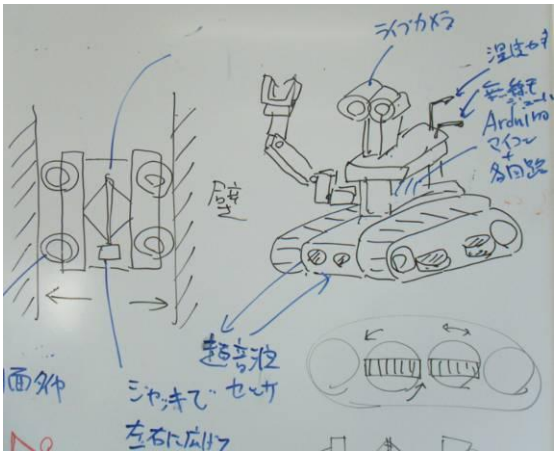


図２　ホワイトボードに描いたスケッチ

とにかく考えて行くうちに、このテーマがとても難しく感じた。それでも、生徒の知識で出来ることを検討した。その結果、色々なセンサを搭載して、悪環境の中でも動くことのできる**「被災地探査ロボット」**として、以下の仕様でこの研究に着手することにした。

① 自由に動けるように、ロボットは電波による遠隔操縦とする。このときパソコンをコントローラとして使用する。同時にロボット搭載センサの状態をパソコンで受信表示する。

② 荒れ地を走行する機構として、 本体の移動にクローラー（キャタピラ）を使用する。
加えて、本体の前に補助アームも設置する。

③ 生きている人を見つける機構として、
「赤外線人体センサ」を使用する。

④ 障害物や距離を感知する機能として
「赤外線距離センサ」を、
周囲の温度を感知する機能として，「温度センサ」を取り付ける。

⑤ 狭くなっている壁と壁の間を通り抜ける機構として、本体幅変形機構や横側タイヤを設置する。

図３に具体化した構想図を示す。

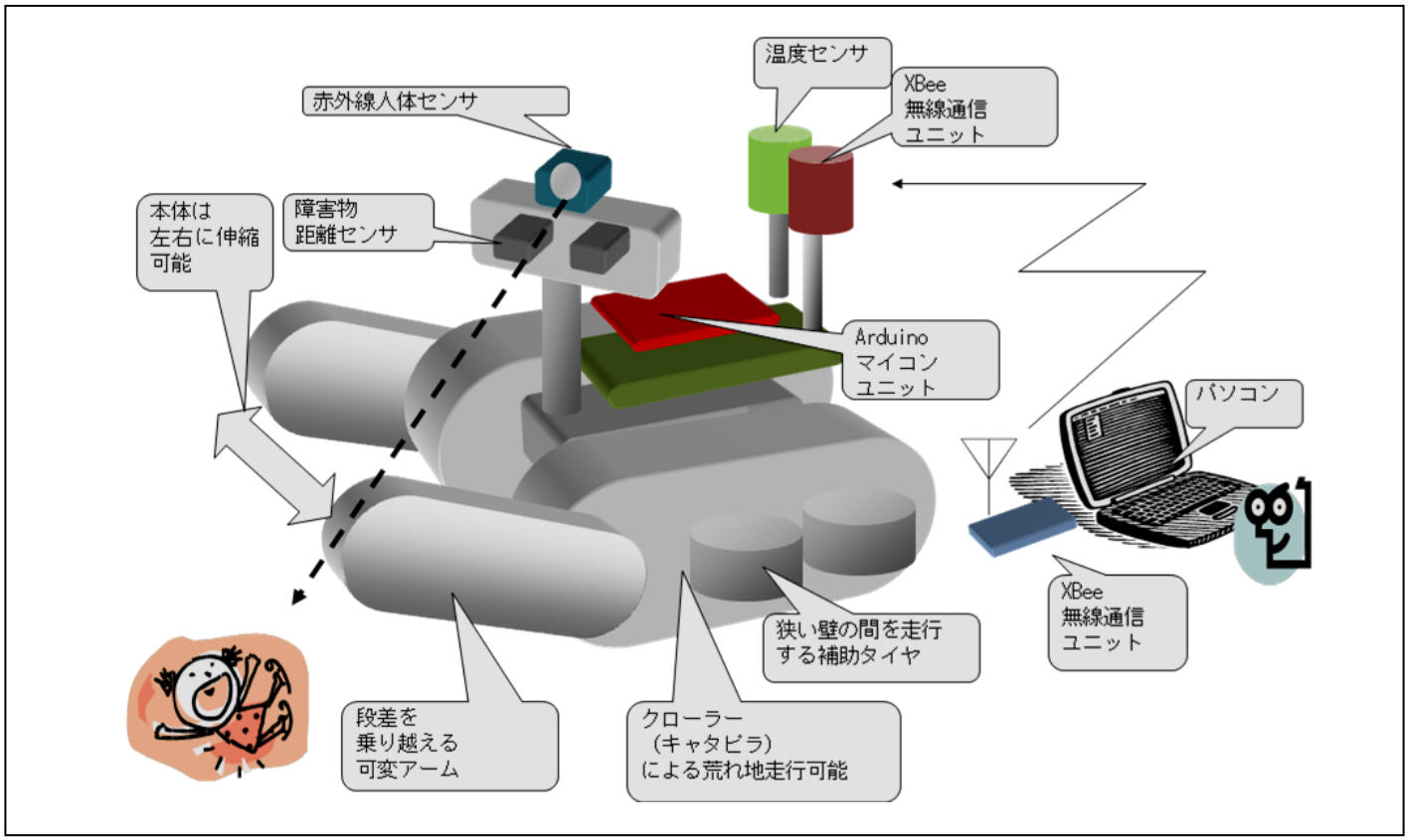


図３　本研究における被災地探査ロボットの構想図

３．課題研究としての着手

本研究を、平成 23 年度工業系学科三年生の課題研究テーマとして提示したところ、電気科３名および電子機械科２名の生徒が賛同し、取り組む事になった。それぞれの学科で学習してきたことを活かして、「回路班」と「メカ班」に分担し、ロボットの電子回路系統と機械系統の製作に着手した（図４）。

担当職員は、筆者の他に機械系に精通している職員をもう１人配置した（本校電子機械科・井出史憲先生）。

岩村田高校の工業系の３学科（機械・電子機械・電気）は、毎週同じ時間帯に課題研究を行っているため、違う学科のメンバーで協力し合いながら同じテーマの研究に取り組むことが可能である。

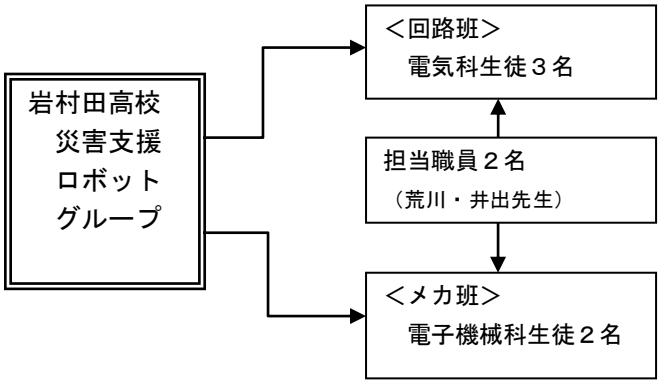


図４　本研究グループの構成

４．ロボット回路系統の製作（回路班）

本研究グループの回路・プログラム班では、以下の内容で研究を進めた。

・マイコンボードの製作と制御プログラミング学習
・ロボットの駆動回路（モータ回路）の製作
・センサ回路の製作（距離、温度、人体各種センサ）
・テスト用ロボットの製作
・制御ソフトウェアの開発
　（送信側：パソコン　受信側：マイコンボード）

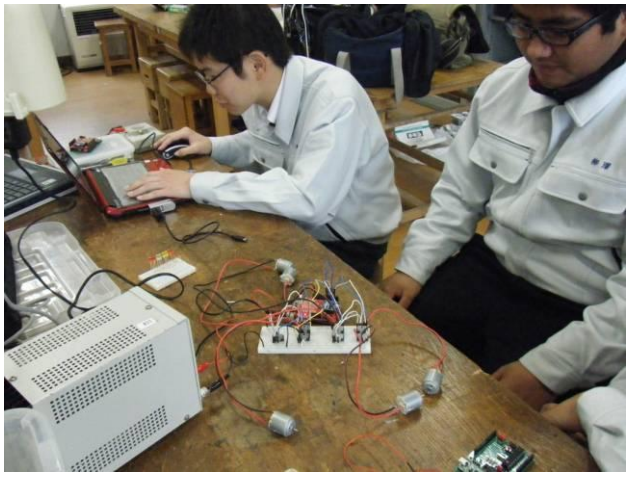


図５　研究の様子（回路の試作）

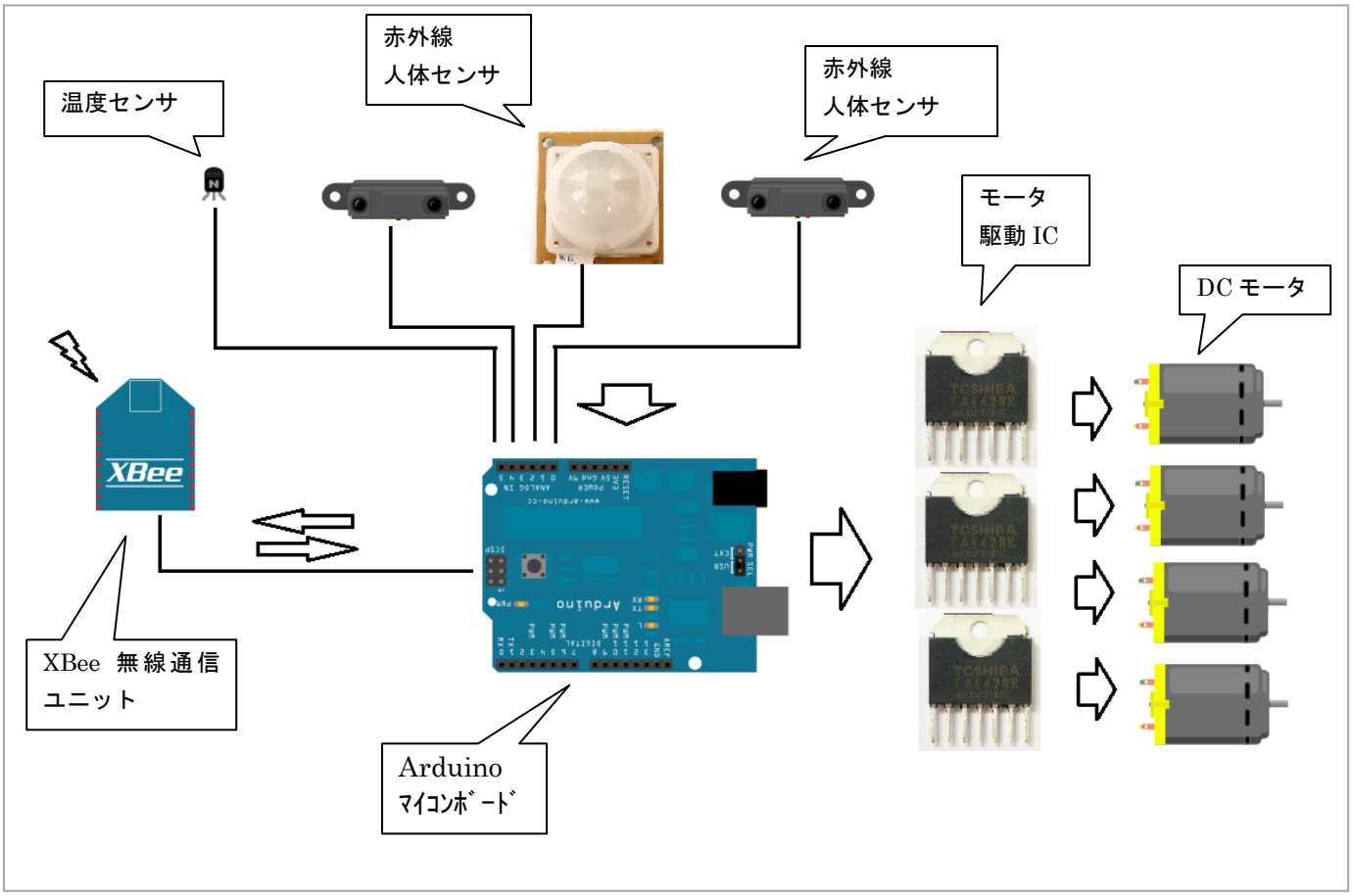


図６　Arduino マイコンを中心にしたロボット回路系統ブロック図

図６は、ロボットシステムの回路構成を示したものである。「Arduino(アルドゥイーノ)」と呼ばれるマイコンボードを中心に、モータ駆動回路、各種センサ、無線通信ユニットから構成されている。

４．１　Arduino マイコンボードについて

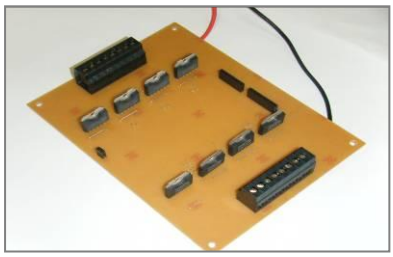
Arduino は、現在世界中で普及しているマイコンボードである。C 言語に似たプログラミング言語を用いて簡単に制御プログラムを作成することができる。本研究でこのマイコンボードを採用したのは、回路構成・関連ソフトウェアがオープンアーキテクチャ（著作権フリー）であり、またモータ、センサ、無線通信など多種多様の制御事例文献が多いためである。

今回は、Adruino 互換マイコンボードとして秋月電子社製の組立キットを使用した（図７）。

図７　Adruino 互換マイコンボード

４．２　モータ駆動回路について

回路は専用の IC「TOSHIBA　TA8428K」を使用して設計製作した。図８にモータ駆動回路と周辺回路を接続した様子を示す。回路基板は、パソコンで設計し（基板設計ソフト「PCBE」使用)、基板加工機で作成した。最大８個のモータを制御することができる。



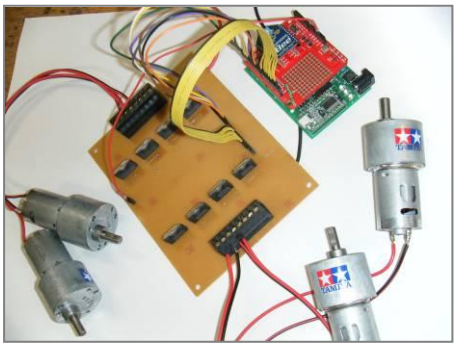


図８　モータ駆動回路

４．３　各種センサについて

赤外線距離センサは、シャープ社製測距モジュール「GP2Y0A02YK」を使用した。0.2mから 1.5mの範囲でこのセンサから放った赤外線を物体が反射し、再びこのセンサに戻ることで距離を算出することができる。

図９　赤外線距離センサ

温度検出にはＩＣ温度センサＬＭ６１ＢＩＺを使用した。－30℃から 100℃の範囲を測定することができる。１℃当たり１０mV の電圧が出力されるので、その電圧を Arduino マイコンボードのアナログポートで読み取り温度を測定する。

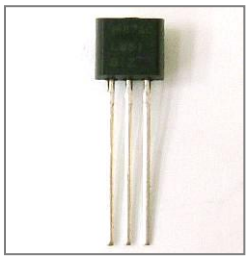


図 10　温度センサ「LM61BIZ」

赤外線人体センサとして「SE-10」と呼ばれる焦電型赤外線センサモジュールを用いた。約２ｍまでの人体が発する波長の赤外線を感知することができる。

すなわち、このセンサを用いて、がれきの中から生きている人を発見することが出来ると考えた。

図 11　赤外線人体センサ「SE-10」

４．４　無線通信ユニット「XBee」について

ロボット本体と、パソコンによるコントローラ間の無線通信を行うために「XBee」と呼ばれる無線通信ユニットを使用した。

Arduino マイコンボードは、パソコンの USB 端子と接続することで有線式シリアル通信を行うことが出来る。XBee を使用するとシリアル通信を無線化することができ、ロボットの無線遠隔操作ができるようになると考えた。図 12 は、XBee ユニットを搭載した Arduino 用無線通信回路（XBee シールド）である。また、図 13 はマイコンに通信回路を搭載し、パソコン側のユニットと通信している様子である。

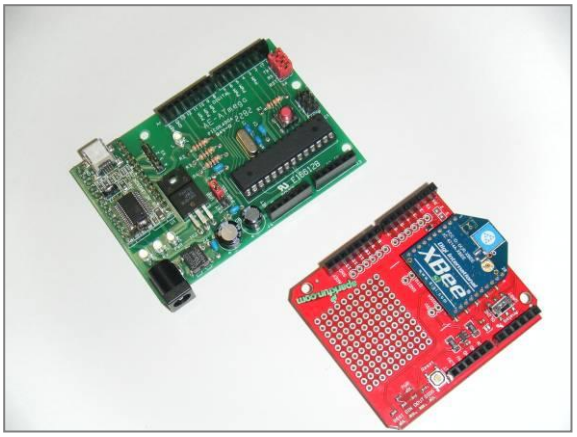


図 12　XBee ユニットを搭載した無線通信回路（右側）

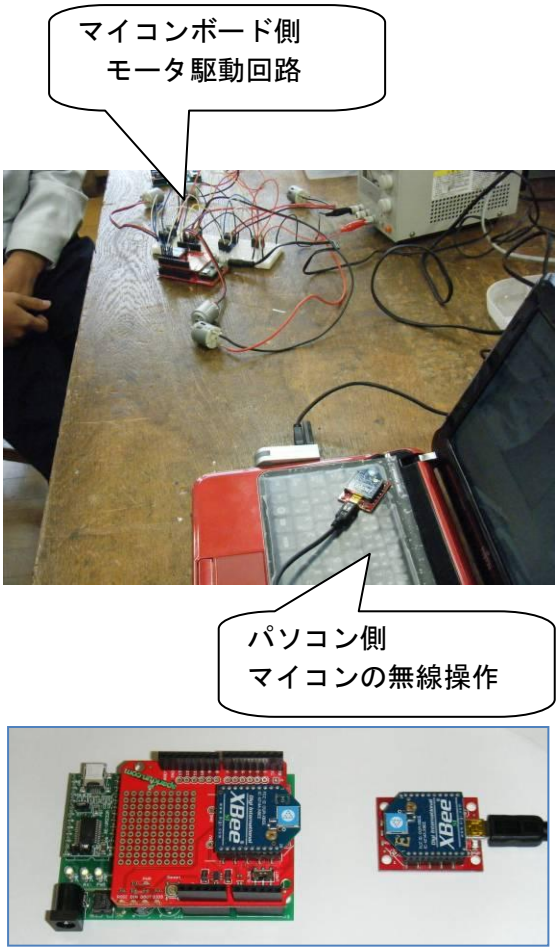


図 13　XBee による無線シリアル通信の様子

４．５　テスト用ロボットの製作

メカ班によるロボット本体製作が完成するまでの間、製作した回路やマイコンボード、制御プログラム等の動作確認をするために、テスト走行用ロボットを製作した。

図 14 にその様子を示す。簡単な構成ながら前記各種センサおよび回路を搭載し、無線操縦のできる探査センサロボットを実現している。

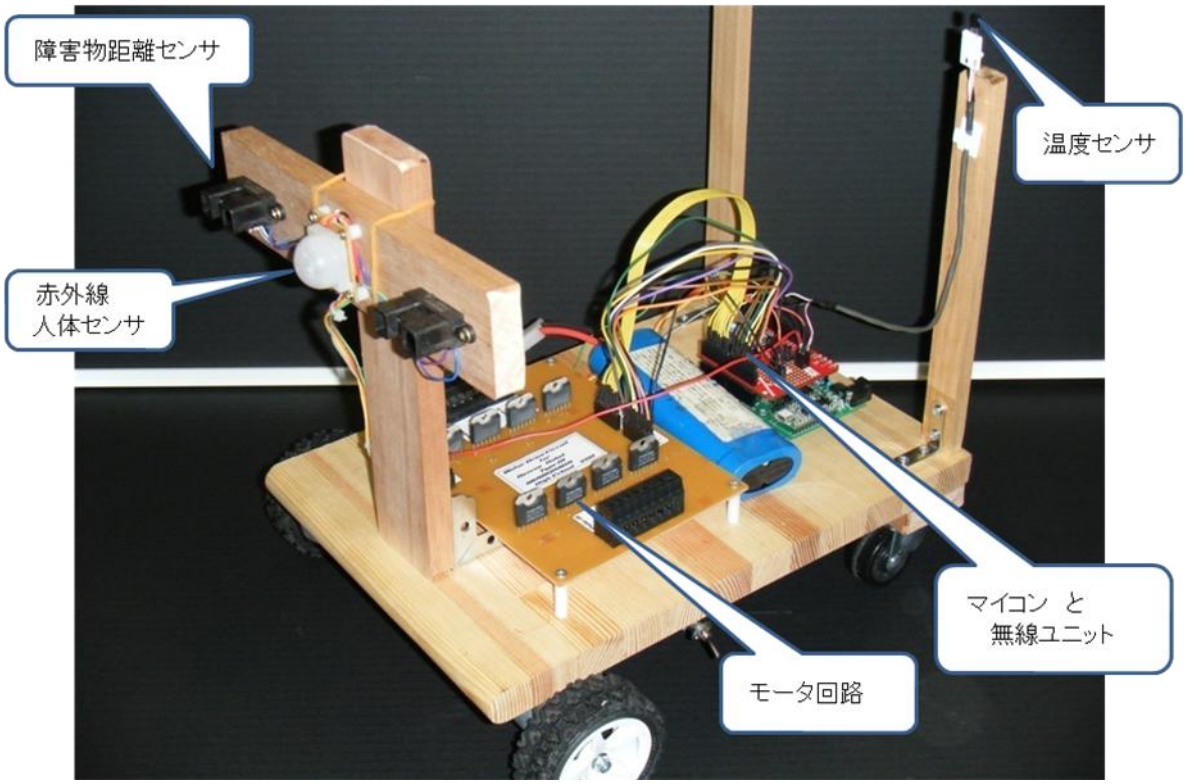


図 14　回路系統・プログラムテスト用ロボットの様子

４．６　　通信・制御プログラムについて

以上の回路のセンサデータの通信やモータ制御のために、マイコンボード側とパソコン側の双方でプログラムを作成した。

本研究では、　パソコン側のソフトを「Visual C#」で、 マイコン側のソフトを「Arduino－IDE」と呼ばれる開発ソフトを使用して作成した。

その結果、図 15 のように、 パソコンの画面をマウスで操作して操縦するシステムが完成した。

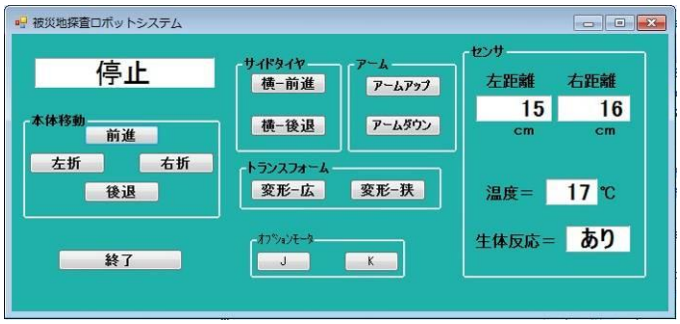


図 15　Visual C#で作成したパソコン側ロボット操作画面

図 16 は、パソコンからロボットを遠隔操作する仕組みを表したものである。

送信機のパソコンでマウス操作をして、「前進」というボタンを押すと、パソコンからは「F」という文字データがマイコンに向かって送信される。

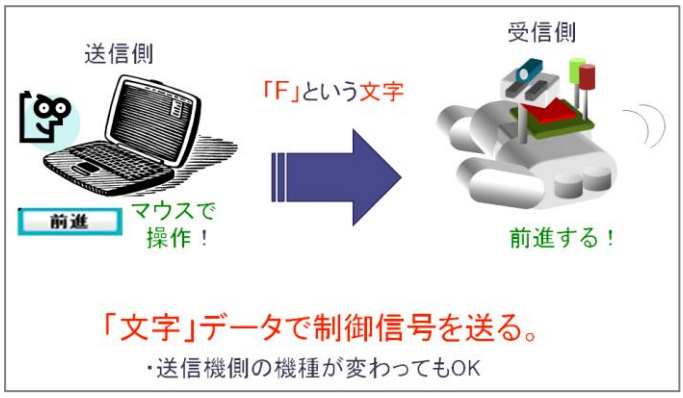


図 16　　パソコンによるロボット遠隔操作のしくみ

ロボットに搭載されたマイコン（受信側）では、この「F」の文字データの受信を受けて ロボット本体を前進させるようにモータ駆動回路を制御する。このように簡単な方法で制御信号を送ることができるため、スマートフォンなどパソコン以外の端末で動かすことも可能だと考えられる。

以上の方法で、テスト用ロボットを遠隔制御することに成功した。さらに、ロボット側から各センサのデータをパソコン側に送信し、画面上に表示させることにも成功した（図 15）。

５．ロボット機構部の製作（メカ班）

５．１　設計

メカ班では、構想を基にして設計ソフト「CADSUPER FX」を使用し設計を開始した。

まず、どのような形にするかを考え、壁隙間走行用のモータを取り付けるため本体の形は三角形にし（図17)、本体を可変式にするためにはパンダジャッキの機構を使用することにした（図 18)。

設計にはとても苦戦し、大変多くの時間を費やしてしまった。

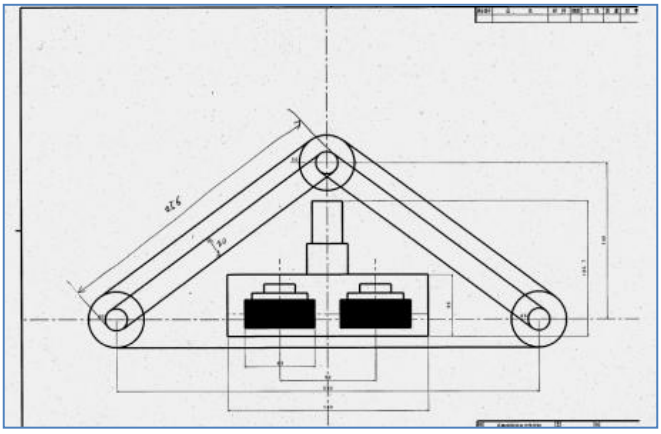


図 17　災害救援ロボット機構部の設計①（側面図）

５．２　製作

本体の設計のあと、材料の加工やフレームの製作を行った。 フレーム材料は 11㎜ の角材厚さ１㎜ を選定し、強度と軽量化の両立 を考え、接合はねじ止めではなく、半自動溶接機による溶接加工を行った（図 19)。

図 19　半自動溶接機による溶接加工

悪環境のなかでも進むことができることを可能とするため、駆動をタイヤではなくベルトにした(図 21)。

また、側溝のような両側を狭い壁で挟まれている場所を通り抜けたり登れるようにするためにパンダジャッキの原理を使い車体の幅を変えることができるようにした（図 18 中央部、図 20)。

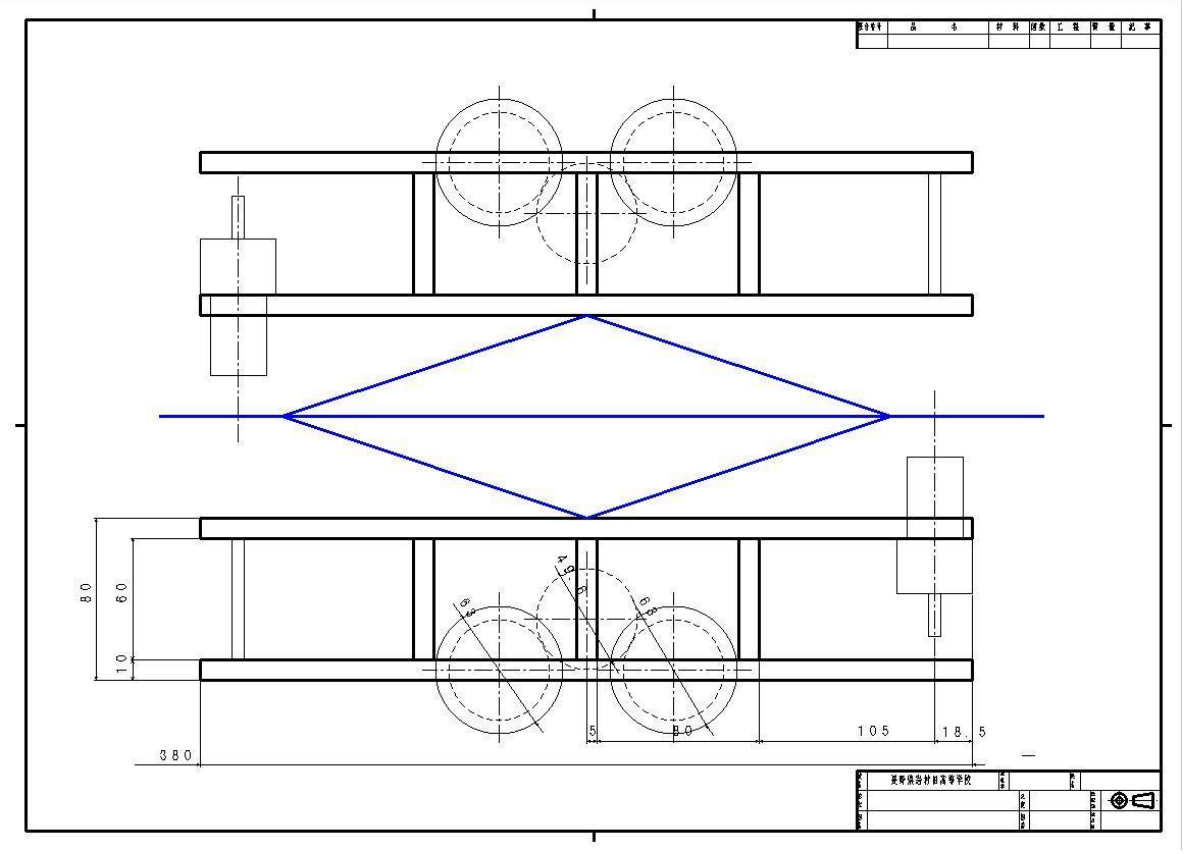


図 18　災害救援ロボット機構部の設計②（平面図）

図 20　パンダジャッキ用大型ウォームギヤ

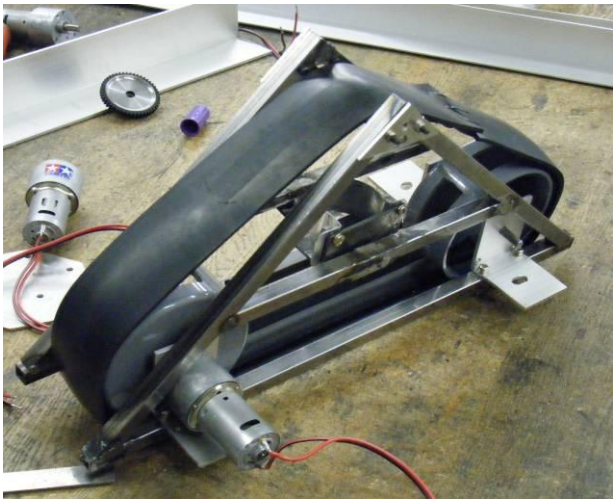


図 21　ベルト形クローラーの取付け

壁を登っている状態で進むために地上走行用のクローラーとは別に、クローラーの間に本体と平行な向きに補助サイドタイヤを取り付け、壁を登っていけるようにした。図 22 のようにフレームを三角形にすることにより、本体の強度を高めることができる。

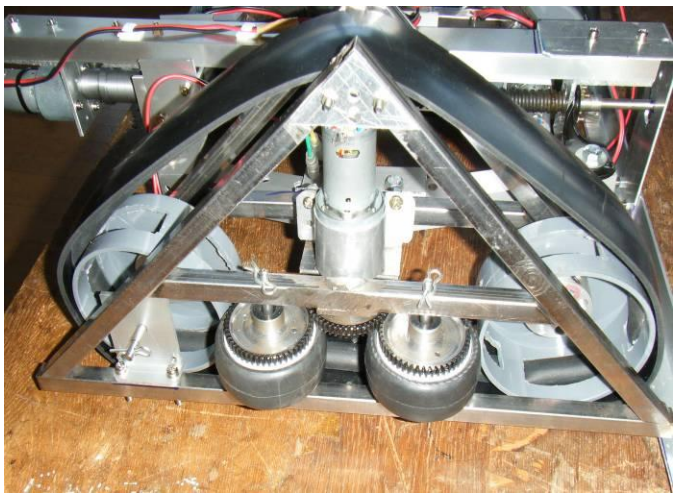
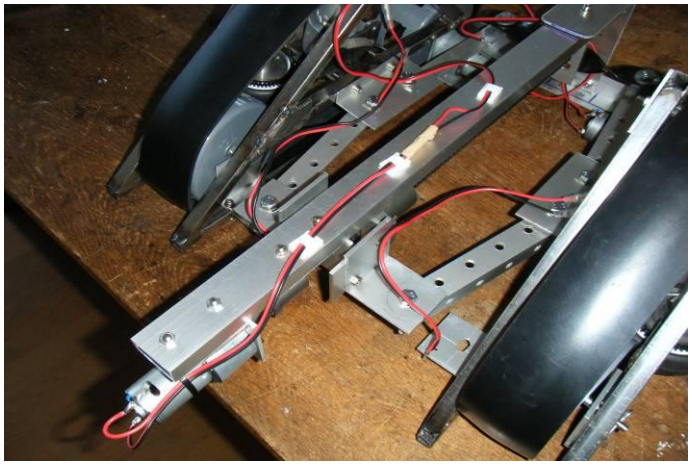


図 22　三角形フレームに、クローラーと
サイドタイヤを装着した様子

a）　ロボット機構部本体概観



b）本体中央部横幅変形機構（パンダジャッキ）

図 23　できあがった災害救援ロボット本体機構部

６．回路系統のパッケージ化（回路班）

ロボット機構部の完成に伴い、機構部に搭載する回路系統をひとまとめにし、装着しやすい形状にした。その様子を図 24 に示す。

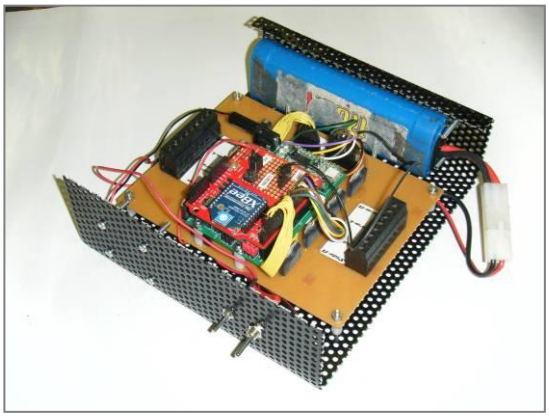
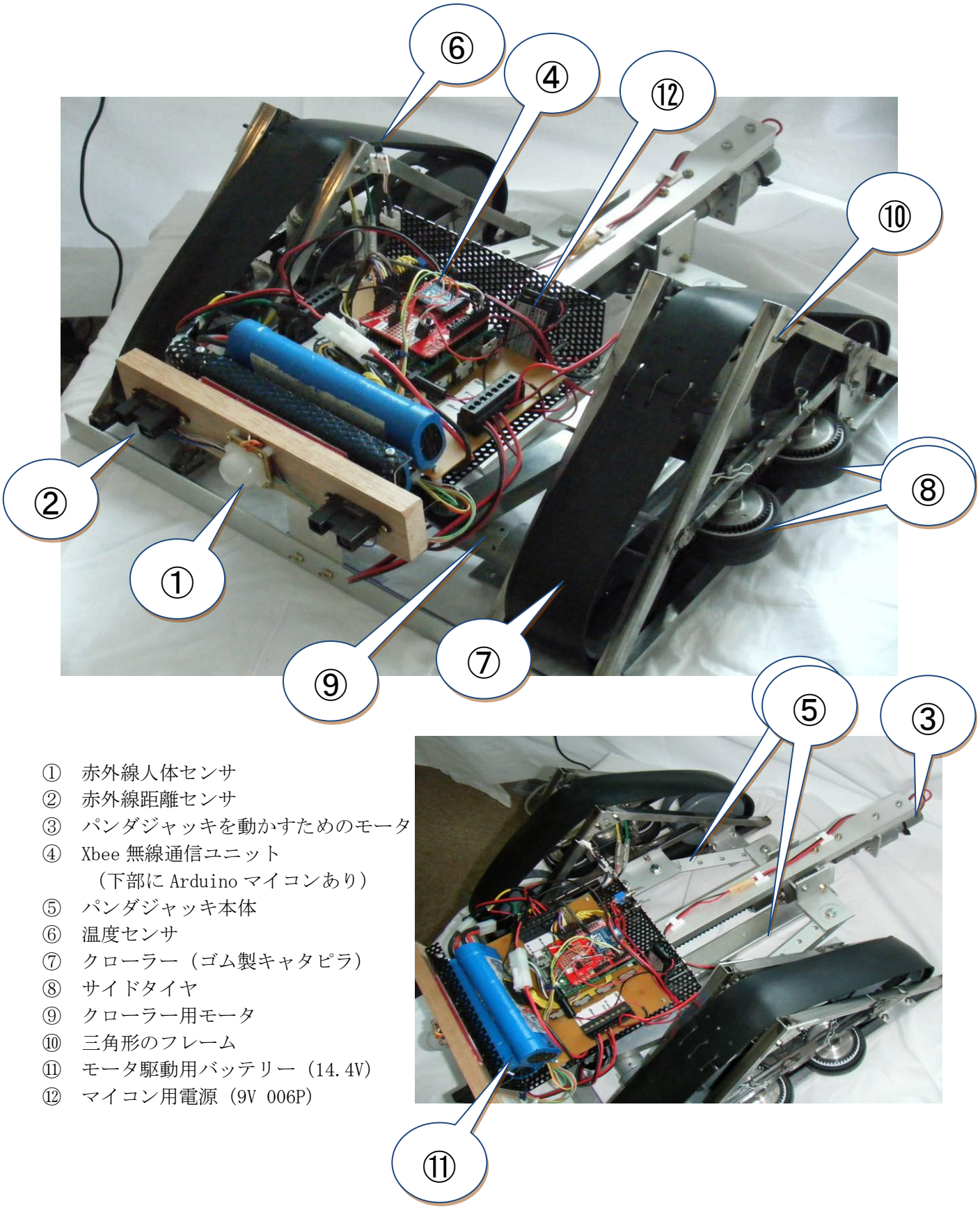


図 24　パッケージ化したロボット回路部

7．組み上げたロボットシステム本体

以上、回路系統とロボット機構部をまとめた
ロボットシステム本体の様子を図 25 に示す。



① 赤外線人体センサ
② 赤外線距離センサ
③ パンダジャッキを動かすためのモータ
④ Xbee 無線通信ユニット
（下部に Arduino マイコンあり）
⑤ パンダジャッキ本体
⑥ 温度センサ
⑦ クローラー（ゴム製キャタピラ）
⑧ サイドタイヤ
⑨ クローラー用モータ
⑩ 三角形のフレーム
⑪ モータ駆動用バッテリー（14.4V）
⑫ マイコン用電源（9V 006P）

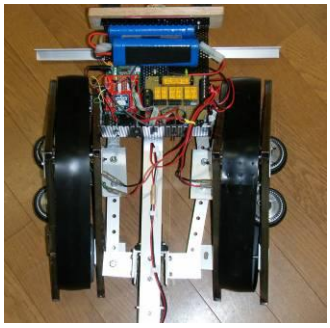
図 25　完成した災害救援ロボットシステム本体

８.製作したロボットシステムの試用と改良

まず、回路班の製作したロボット電気系統は、テスト用ロボットに搭載することで、パソコンによる遠隔操作およびセンサデータの受信を確認することができた（図 26)。

図 26 パソコンからテストロボットを操作する様子

しかしながら、メカ班の製作したロボット機構部に回路を搭載したとき、うまく動作しない結果となった。各機構部に設置されたモータにかなりの負荷がかかるため、製作したモータ駆動回路が対応できないことが分かった。後に電磁リレーによるモータ駆動回路を製作したものに変更することで、本体の起動に成功した。図 27 は、本体幅変形機構を動かしている様子である。



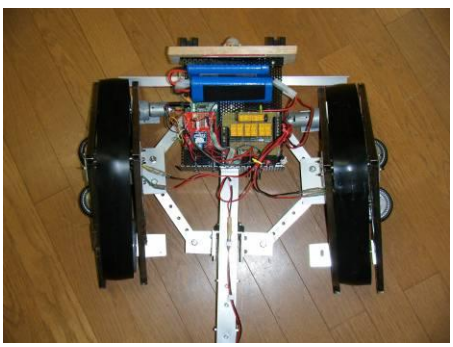


図 27　本体横幅変形機構の様子
（モータ駆動回路を変更した結果、起動に成功した。）

９．まとめ

以上、災害救援ロボットを研究し、被災地探査ロボットとしてシステムを製作することが出来た。

テスト用ロボットでは、パソコンによる遠隔操作およびセンサデータの受信の成功など、当初の目的を満足できる結果であった。そして、「本体横幅形状変形機構」の実現など、奇想天外な生徒の発想を形にすることが出来た。課題研究終了時には間に合わなかったメカ班製作のロボット本体の起動は、モータ駆動回路を電磁リレーのものに変更することで実現することができた。さらに、以下のように災害救援ロボットとしての機能を拡充し、色々な課題に挑戦したいと感じた。

- 機構部の充実
- ライブカメラの搭載
- GPS 機能による現在位置の検出
- ロボットアームの搭載
- 各種センサの増設（ガイガーカウンタなど）
- マイコンとセンサによる自動操縦
- 各種小型携帯端末（iPhone ,Android など）による遠隔操縦およびデータ収集
- レスキューロボットコンテストの出場

「H23.3.11 東日本大震災」をきっかけに始めた本研究テーマであるが、その内容は大変難しいものであり苦労することが多かった。

現在のところ、製作したロボットは、すぐ被災地の人たちの役に立つものではない。しかしながら、取り組んだ生徒たちは、

「何もないところから苦労して
アイディアを形にすることが出来た」

という点で大変満足したようであった。

今回担当した生徒たちが、本研究での経験や 3.11 震災に対する気持ちを卒業後も大事にしてくれることを願い、これからも

「人々の助けになる、もの作り」

の指導を続けたいと感じた。

10. 謝辞

本研究を進めるにあたり、その機会を与えていただいた、井出 史憲先生はじめとする岩村田高等学校の先生方に感謝いたします。また、研究主旨に賛同して参加してくれた平成 23 年度課題研究グループの諸君に感謝いたします。（葦澤 大君、飯田 健吾君、柳澤 力君、須江 規真君、友野 昭紀君）

そして、本研究は平成 23 年度長野県産業教育振興会の特別研究助成を受けて行うことができました。この場を借りて関係者の方々に御礼申し上げます。

11. 参考文献

１）田原 淳一郎

「Arduino で始める電子工作

―8bit マイコンを活用するオープンプロジェクト Arduino の世界」

カットシステム　2010 年

２）Massimo Banzi 著　船田 巧 訳

「Arduino をはじめよう (Make:PROJECTS)」

オライリージャパン　2009 年

３)「建築発明工作ゼミ」より「Arduino」

http://kousaku-kousaku.blogspot.com/2008/07/arduino.html

４）「Visual C#の覚え書き」より「文字列のシリアル通信」

http://mtmkhome.com/weblog/blog01/2011/07/19/

５）「Visual C# 2008 の超初歩的メモ 」

http://mtmkhome.net/mtmk_vcs.php/

６)「フルブリッジドライバーＴＡ８４２８Ｋ」

「シャープ測距モジュールＧＰ２Ｙ０Ａ０２ＹＫ」

「焦電型赤外線センサモジュール　SE-10」

「AKI-AVR マイコンボード」

秋月電子通商　http://akizukidenshi.com/

７）葦澤　他

「災害救援ロボットの研究」　岩村田高校

平成 23 年度特別生徒研究報告書　　長野県産業教育振興会

８）荒川

「災害被災地探査ロボットの研究」

長野県工業教育研究会　平成 24 年度電気科研究協議会報告

(2012.9.25)

長野県岩村田高等学校(普通科･機械科･電子機械科･電気科)

http://www.nagano-c.ed.jp/ganko/

〒385-0022　長野県佐久市岩村田 1248-1

Tel. (0267)-67-2439　Fax. (0267)-66-1450

※工業系学科は H27 より機械システム科・電気情報科として

新校**「長野県佐久平総合技術高等学校」**へ移転

http://www.nagano-c.ed.jp/saku-kiu/

＜付録＞ ロボット制御プログラム

１．ロボット本体側（Arduino マイコン）

```
//
災害救援ロボット
ロボット側プログラム（Arduino）
   「xbee_ar_rx.pde」（Arduino-IDE にて作成）
//

int val;//受信データ用の変数を用意

//データ送信用の変数
int Ir1pin=1;//赤外線距離センサ１　アナログポート設定
int Ir1value=0;
int Ir2pin=2;//赤外線距離センサ２　アナログポート設定
int Ir2value=0;
int TempPin=3;//温度センサ　アナログポート設定
int st,s1,s2;//５０回データ収集時の合計
int temp,tempd;

int i;
int motion,m;//人体センサ

void setup(){
  //シリアル通信開始 9600bps
  Serial.begin(9600);
  //2～13 ピンをデジタル出力(ﾓｰﾀ出力)に設定
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
}
void loop(){
 //モータ駆動信号受信処理
 //int i;
  if(Serial.available()>0){//　パソコンから文字データが来たら
    val=Serial.read();        //　データを読み込み、val へ代入
  }

  if(val=='F'){                          //文字データが「F」の場合
    digitalWrite(2,HIGH);                 //本体前進
    digitalWrite(3,LOW);
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);

  }
  else if(val=='B'){                     //「B」の場合
    digitalWrite(2,LOW);                  //本体後退
    digitalWrite(3,HIGH);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
  }
   else if(val=='L'){                    //「L」の場合
    digitalWrite(2,HIGH);                 //本体左折
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
  }
   else if(val=='R'){                    //「R」の場合
   digitalWrite(2,LOW);                   //本体右折
   digitalWrite(3,HIGH);
   digitalWrite(4,HIGH);
   digitalWrite(5,LOW);
  }
   else if(val=='X'){                    //「X」の場合
   digitalWrite(10,HIGH);                 //横タイヤ前進
   digitalWrite(11,LOW);
  }
   else if(val=='Y'){                    //「Y」の場合
   digitalWrite(10,LOW);                  //横タイヤ後退
   digitalWrite(11,HIGH);
  }
   else if(val=='U'){                    //「U」の場合
   digitalWrite(12,HIGH);                 //アームアップ
   digitalWrite(13,LOW);
  }
   else if(val=='D'){                    //「D」の場合
   digitalWrite(12,LOW);                  //アームダウン
   digitalWrite(13,HIGH);
  }
   else if(val=='W'){                    //「W」の場合
   digitalWrite(6,HIGH);                  //変形-広げる
   digitalWrite(7,LOW);
  }
  else if(val=='N'){                     //「N」の場合
   digitalWrite(6,LOW);                   //変形-狭まる
   digitalWrite(7,HIGH);
  }
    else if(val=='J'){                   //「J」の場合，オプション
   digitalWrite(8,HIGH);                  //　（予備モータ）正転
   digitalWrite(9,LOW);
  }
  else if(val=='K'){                     //「K」の場合
   digitalWrite(8,LOW);                   //　予備モータ逆転
   digitalWrite(9,HIGH);
  }

 //受信文字データが「S」の場合
   ロボット動作停止
    および　センサデータの読込＆送信
  else if(val=='S'){
    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
    digitalWrite(10,LOW);
    digitalWrite(11,LOW);
    digitalWrite(12,LOW);
    digitalWrite(13,LOW);
  // センサ読み取り処理
    st=0,s1=0,s2=0;
    //センサ値 50 回読み取り
    for(i=1;i<=50;i++){
       tempd=analogRead(TempPin)*(5000.0/1024)/10-60;
     //温度センサ-読み取り
       Ir1value=analogRead(Ir1pin);//距離センサ右-読み取り
       Ir2value=analogRead(Ir2pin);//距離センサ左-読み取り
       //センサ値合計計算
       st=st+tempd;
       s1=s1+Ir1value;
       s2=s2+Ir2value;
      }

    //生体センサ読み取り
    m=analogRead(5);
    if( m==0){
      motion='A'; //反応したら，文字データ「A」を送信
    }
    else{
      motion='N';
    }
  // データ平均値計算および補正
    int range1=(6787.0/(s1/50-3.0))-4.0; //右データ
    int range2=(6787.0/(s2/50-3.0))-4.0; //左データ
    temp=st/50;//温度データ
       //センサデータの送信（センサ値を文字として送信）
         Serial.println(temp);
         Serial.println(range1);
         Serial.println(range2);
         Serial.println(motion);
       //delay(2000);
//
  }
}
```

２．パソコン側

```
/
//災害救援ロボット・パソコン側プログラム
//　（Visual C#　2008 Express Edition　使用）
//
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;


namespace xbee_pc_tx
{
    public partial class Form1 : Form
    {
        int sw1 = 1;
        int sw2 = 1;
       int flg1=0;
       int flg2=0;
       int i;
        // 通信開始処理
        public Form1()
        {
            InitializeComponent();
            serialPort1.PortName = "COM10";
              //XBee 接続のパソコンシリアルポート番号
            serialPort1.BaudRate = 9600; //通信速度　9600bps
            serialPort1.DataBits = 8;
            serialPort1.StopBits = StopBits.One;
            serialPort1.Parity = Parity.None; //
            label1.Text = "Start!!"
            // 通信成功（通信ポートを開く）
            try
            {
                serialPort1.Open();
                label1.Text = "PortOpen!!";
            }

            // 通信失敗（XBee が未装着）
            catch
            {
            MessageBox.Show("XBee が接続されておりません");
                label1.Text = "NG!!";
            }
          }

       // 終了ボタン処理
        private void button4_Click(object sender, EventArgs e)
        {
            if (serialPort1.IsOpen == true)
            {
                serialPort1.Close();　//通信ポート閉じる
            }
            Close(); //プログラム終了
        }
      //　パソコンからロボットへ各制御ボタン押下処理
            （制御信号文字の送信処理）
         private void button1_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("L");
            label1.Text = "左折";
        }

        private void button2_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("R");
            label1.Text = "右折";
        }

        private void button1_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";
        }

        private void button2_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";

        }

        private void button3_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("F");
            label1.Text = "前進";
        }

        private void button3_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";
        }

        private void button5_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("B");
            label1.Text = "後退";
        }

        private void button5_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";
        }

        private void button6_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("X");
            label1.Text = "横-前進";
        }

        private void button6_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";
        }

        private void button7_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("Y");
            label1.Text = "横-後退";
        }

        private void button7_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";
        }

        private void button8_MouseDown
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("U");
            label1.Text = "ｱｰﾑ・ｱｯﾌﾟ";
        }

        private void button8_MouseUp
                (object sender, MouseEventArgs e)
        {
            serialPort1.WriteLine("S");
            label1.Text = "停止";
        }
```

```
    private void button9_MouseDown
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("D");
        label1.Text = "ｱｰﾑ・ﾀﾞｳﾝ";
    }

    private void button9_MouseUp
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("S");
        label1.Text = "停止";
    }

    private void button10_MouseDown
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("W");
        label1.Text = "変形-広";
    }

    private void button10_MouseUp
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("S");
        label1.Text = "停止";
    }

    private void button11_MouseDown
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("N");
        label1.Text = "変形-狭";
    }

    private void button11_MouseUp
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("S");
        label1.Text = "停止";
    }
    private void button12_MouseDown
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("J");
        label1.Text = "ｵﾌﾟｼｮﾝ J";
    }

    private void button12_MouseUp
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("S");
        label1.Text = "停止";
    }
    private void button13_MouseDown
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("K");
        label1.Text = "ｵﾌﾟｼｮﾝ K";
    }

    private void button13_MouseUp
            (object sender, MouseEventArgs e)
    {
        serialPort1.WriteLine("S");
        label1.Text = "停止";
    }



//  ロボットからパソコンへ　センサデータ受信処理
    温度・右距離・左距離・生体反応の順に文字列として
    データが送られてくるのでパソコン側でこの順番に
    読み取り，RcvData1～RcvData4 に格納する。

    // ﾃﾞﾘｹﾞｰﾄ追加宣言
    private delegate void AddRecievedDataDelegate
    (string RcvData);

    // 受信割込
    private void serialPort1_DataReceived
       (object sender, SerialDataReceivedEventArgs e)
    {
        String RcvData1, RcvData2, RcvData3, RcvData4;
        try
        {
            serialPort1.NewLine = "¥r¥n";
            RcvData1 = serialPort1.ReadLine();
            // 温度データ受信

            serialPort1.NewLine = "¥r¥n";
            RcvData2 = serialPort1.ReadLine();
            // 右距離データ受信

            serialPort1.NewLine = "¥r¥n";
            RcvData3 = serialPort1.ReadLine();
            // 左距離データ受信

            serialPort1.NewLine = "¥r¥n";
            RcvData4 = serialPort1.ReadLine();
            // 生体反応データ受信
        }
        catch (Exception ex)
        {
            RcvData1 = ex.Message;
            RcvData2 = ex.Message;
            RcvData3 = ex.Message;
            RcvData4 = ex.Message;
        }

    AddRecievedDataDelegate add1
     = new AddRecievedDataDelegate(addRcvData1);
            label1.Invoke(add1, RcvData1 + "¥r¥n");
    AddRecievedDataDelegate add2
     = new AddRecievedDataDelegate(addRcvData2);
            label2.Invoke(add2, RcvData2 + "¥r¥n");
    AddRecievedDataDelegate add3
     = new AddRecievedDataDelegate(addRcvData3);
            label3.Invoke(add3, RcvData3 + "¥r¥n");
    AddRecievedDataDelegate add4
     = new AddRecievedDataDelegate(addRcvData4);
            label3.Invoke(add4, RcvData4 + "¥r¥n");
      }

      //温度センサ表示
      private void addRcvData1(string RcvData1)
      {
          label2.Text = Convert.ToString(RcvData1);
      }

      //右距離センサ表示
      private void addRcvData2(string RcvData2)
      {
          label3.Text = Convert.ToString(RcvData2);
      }

      //左距離センサ表示
      private void addRcvData3(string RcvData3)
      {
          label4.Text = Convert.ToString(RcvData3);
      }


    // 生体反応受信および表示

      private void addRcvData4(string RcvData4)
      {
      //ロボットから「A」(ASCII 文字コード 65)という
      文字データを受けた時
          if (RcvData4 == "65¥r¥n")                {
              RcvData4 = "あり"; //「生体反応あり」
          }
          else
          {
              RcvData4 = "なし";
          }
          label5.Text = Convert.ToString(RcvData4);
      }
  }
}
```