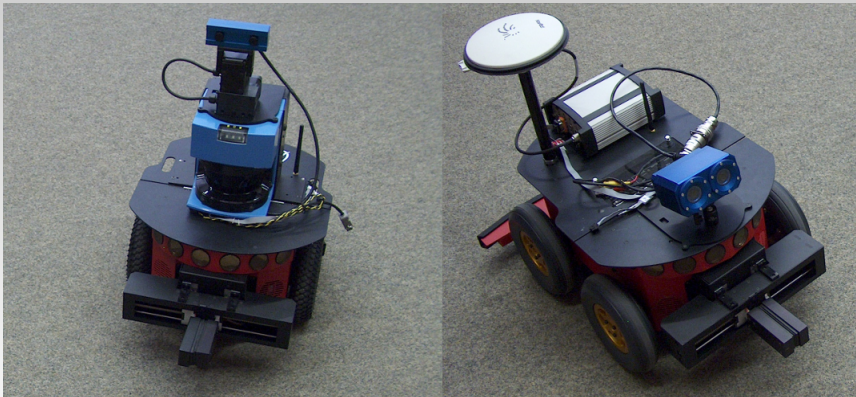


Wprowadzenie do mobilnych robotów

Adam Krechowicz

9 stycznia 2018

Adept MobileRobots ©

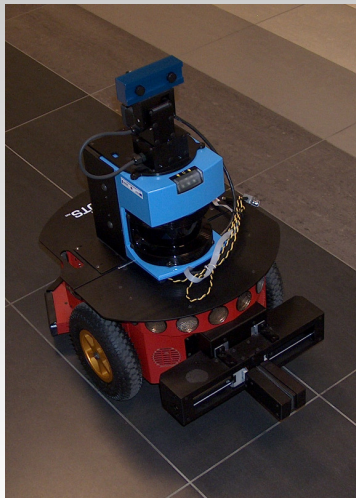


Pioneer

- Mobilne roboty
- wykorzystywane głównie w pracach badawczych
- napęd kołowy
- sterowane przez wbudowany mikrokontroler
- wyposażone w komputer pokładowy
- poruszające się samodzielnie

Pioneer 3dx

- Konstrukcja trójkołowa
- Zastosowanie w budynkach
- Waży 9 kg
- Prędkość maksymalna 1.4 m/s
- Może unieść 23 kg



Pioneer 3at

- Napęd na cztery koła
- Stosowany na dowolnym terenie
- Waży 14 kg
- Prędkość maksymalna 0.7 m/s
- Może unieść 12 kg



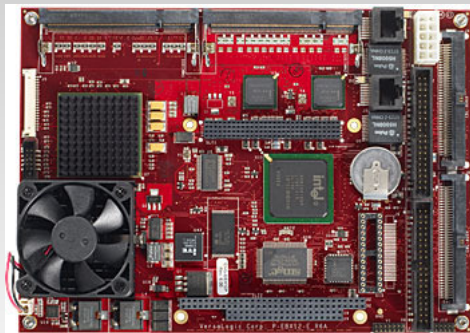
Wyposażenie

- zderzaki
- sonary
- czujnik laserowy
- odbiornik gps
- chwytaki
- kamery
- żyroskopy



Komputer pokładowy

- Pentium M 1.8GHz
- 2GB RAM
- Dysk 80GB



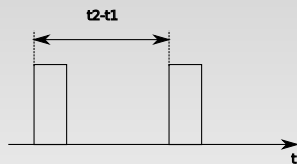
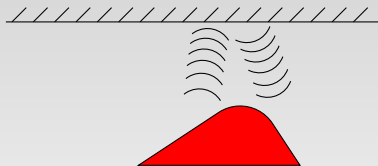
Komunikacja z robotem

- Robot
- Mikrokontroler
- RS232
- Komputer pokładowy
- WiFi
- Komputer sterujący

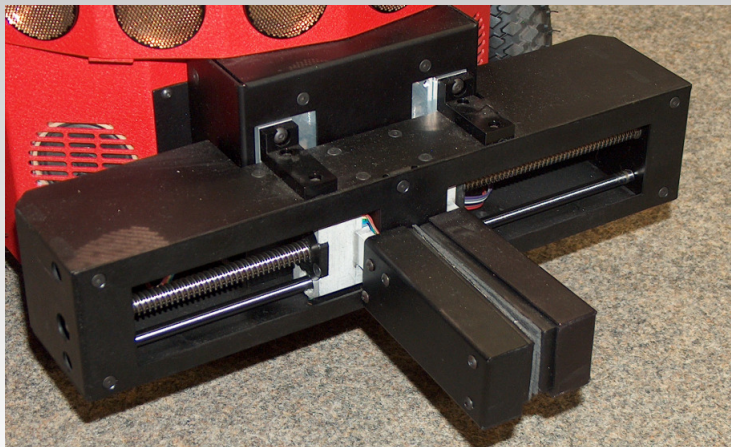
Sonary



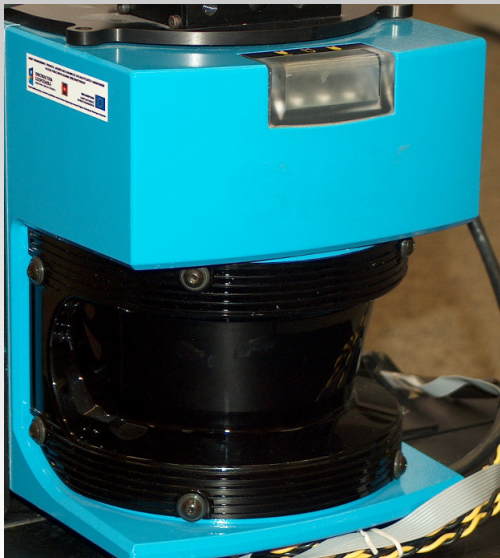
Sonar



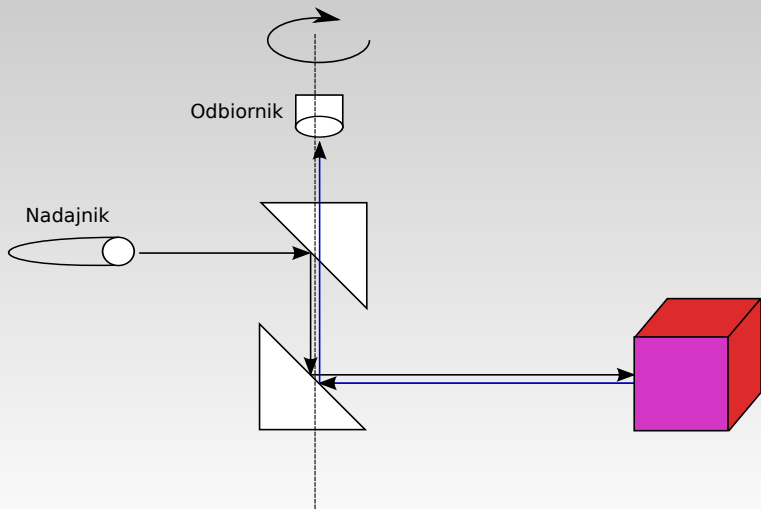
Chwytniki



Dalmierz laserowy



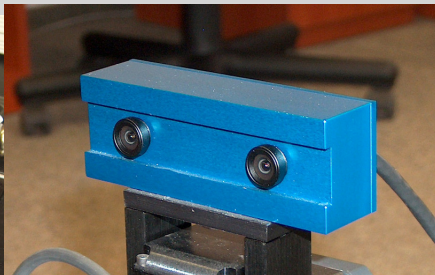
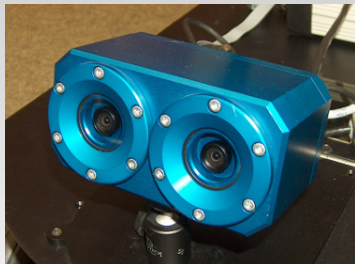
Dalmierz laserowy



Odbiornik GPS



Kamery



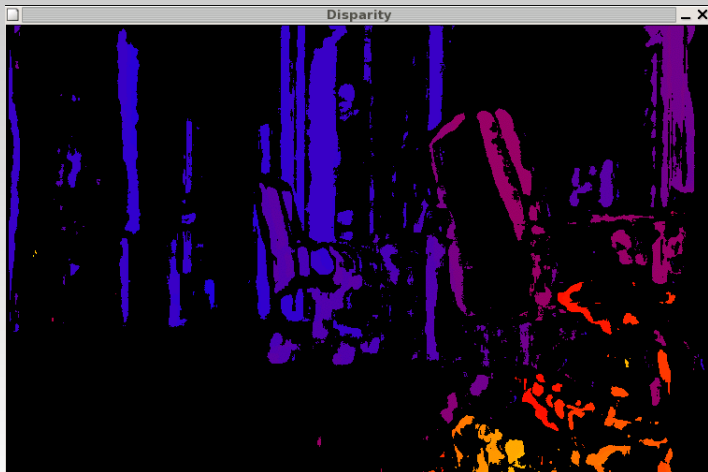
Obraz z kamery



Obraz z kamery



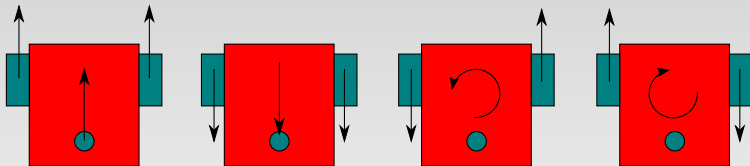
Obraz z kamery



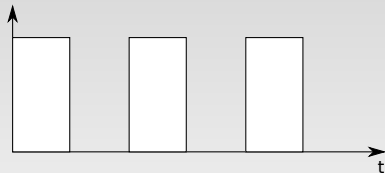
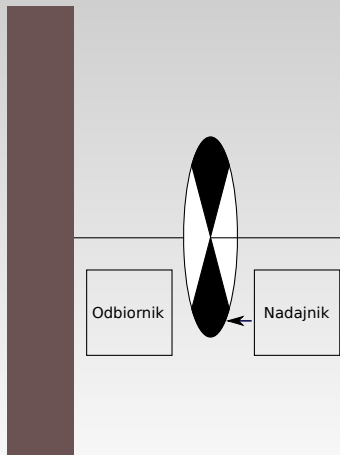
Obraz z kamery



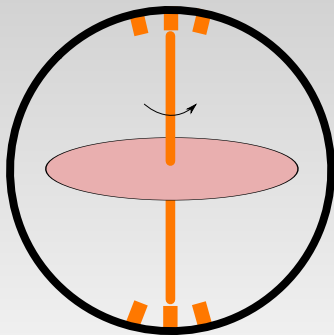
Sterowanie dyferencyjne



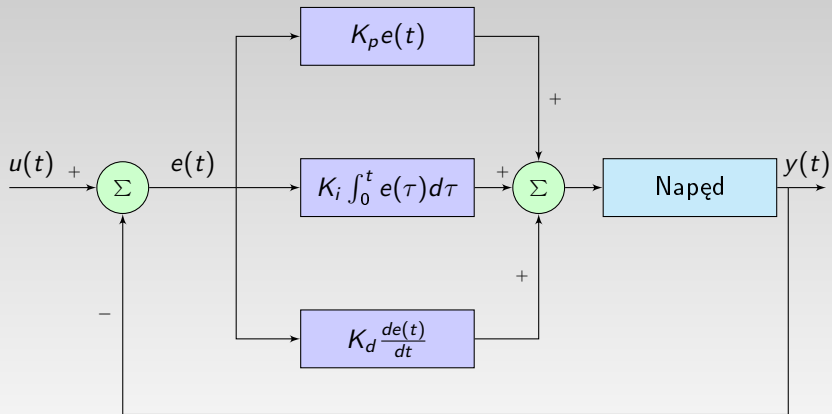
Odometria



Jazda po prostej



Sterownik PID



Oprogramowanie mobilnych robotów

- Robot jest serwerem
- Programy klienckie sterują robotem
- sterowanie ręczne
- teleoperacja
- praca autonomiczna
- praca autonomiczna + zdalna kontrola

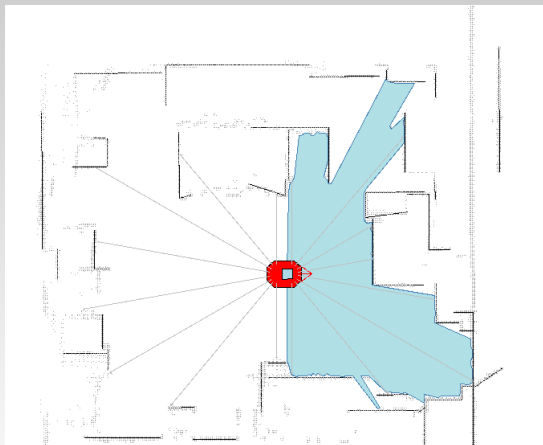
ARCOS

- Advanced Robotics Control and Operations
- firmware robota
- steruje najważniejszymi funkcjami robota
 - odometria
 - PID
 - reagowanie na zablokowania (Stall)
 - komunikacja

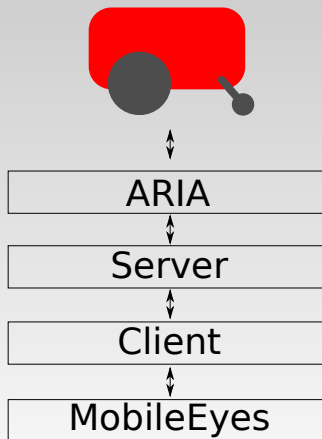
Aria

- Advanced Robot Interface for Applications
- wysokopoziomowe programowanie robota
- C++, Java, Python
 - sterowanie
 - odczytywanie danych z czujników
 - budowanie skomplikowanych zachowań

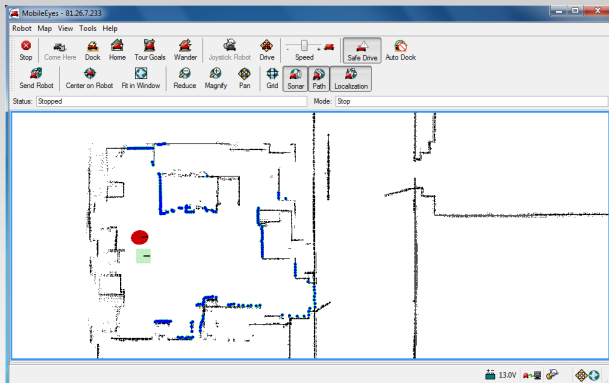
MobileSim



ArNetworking



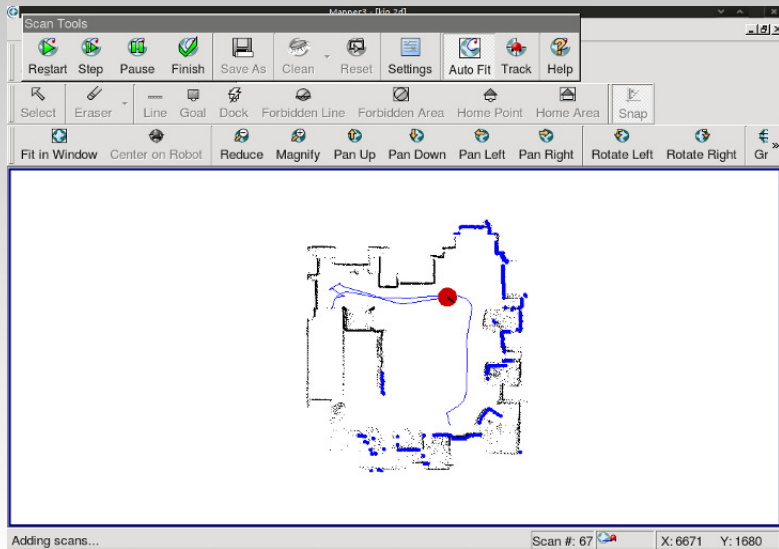
MobileEyes



ARNL

- Advanced Robotics Navigation and Localization
- Planowanie tras
- Nawigacja robota
- Wykorzystuje mapy
- Wykorzystuje dalmierz laserowy lub sonary

Mapper3



Kody źródłowe

```
/*MobileRobots Advanced Robotics Interface for Applications (ARIA)  
Copyright (C) 2004, 2005 ActivMedia Robotics LLC  
Copyright (C) 2006, 2007, 2008, 2009 MobileRobots Inc.
```

```
This program is free software; you can redistribute it and/or modify  
it under the terms of the GNU General Public License as published by  
the Free Software Foundation; either version 2 of the License, or  
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

```
If you wish to redistribute ARIA under different terms, contact  
MobileRobots for information about a commercial version of ARIA at  
robots@mobilerobots.com or  
MobileRobots Inc, 10 Columbia Drive, Amherst, NH 03031; 800-639-9481*/
```


Przykładowy program – połączenie

```
#include "Aria.h"
int main(int argc, char** argv){
    Aria::init();
    ArArgumentParser parser(&argc, argv);
    parser.loadDefaultArguments();
    ArRobot robot;
    ArRobotConnector robotConnector(&parser, &robot);
    if (!robotConnector.connectRobot(&robot)){
        ArLog::log(ArLog::Terse, "Error, could not connect to robot.\n");
        robotConnector.logOptions();
        Aria::exit(1);
    }
    robot.runAsync(true);
}
```

Przykładowy program – klasa ArRobot

```
robot.lock();  
ArLog::log(ArLog::Normal, "simpleConnect: Pose=(%.2f,%.2f,%.2f),  
  Trans. Vel=%.2f, Battery=%.2fV",  
  robot.getX(), robot.getY(), robot.getTh(),  
  robot.getVel(), robot.getBatteryVoltage());  
robot.unlock();
```

Przykładowy program – poruszanie

```
robot.lock();  
robot.enableMotors();  
robot.setVel(250);  
robot.unlock();  
ArUtil::sleep(5000);
```

```
robot.lock();  
robot.stop();  
robot.unlock();  
ArUtil::sleep(1000);
```

```
robot.lock();  
robot.setRotVel(10);  
robot.unlock();
```

Przykładowy program – akcje

```
ArActionStallRecover recover;  
ArActionBumpers bumpers;  
ArActionAvoidFront avoidFrontNear("Avoid Front Near", 225, 0);  
ArActionAvoidFront avoidFrontFar;  
ArActionConstantVelocity constantVelocity("Constant Velocity", 400);  
  
robot.addAction(&recover, 100);  
robot.addAction(&bumpers, 75);  
robot.addAction(&avoidFrontNear, 50);  
robot.addAction(&avoidFrontFar, 49);  
robot.addAction(&constantVelocity, 25);
```

Przykładowy program – własna akcja

```
class ActionGo : public ArAction
{
public:
    ActionGo();
    virtual ~ActionGo(void) {};
    virtual ArActionDesired *fire(ArActionDesired currentDesired);
    virtual void setRobot(ArRobot *robot);
protected:
    ArRangeDevice *mySonar;
    ArActionDesired myDesired;
};
```

Przykładowy program – własna akcja

```
ActionGo::ActionGo() : ArAction("Go")
{
    mySonar = NULL;
}

void ActionGo::setRobot(ArRobot *robot)
{
    ArAction::setRobot(robot);
    mySonar = robot->findRangeDevice("sonar");
    if (robot == NULL)
    {
        ArLog::log(ArLog::Terse, "actionExample: ActionGo: Warning: I found no sonar, ←
deactivating.");
        deactivate();
    }
}
```

Przykładowy program – własna akcja

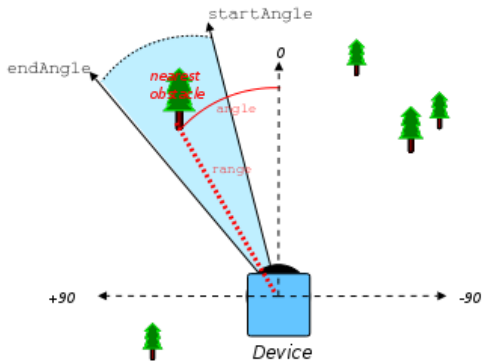
```
ArActionDesired *ActionGo::fire(ArActionDesired currentDesired)
{
    myDesired.reset();
    myDesired.setVel(100);
    return &myDesired;
}
```

Przykładowy program – chwytek

```
ArRobot robot;  
ArGripper gripper(&robot);  
  
gripper->liftUp();  
gripper->liftDown();  
gripper->gripClose();  
gripper->gripOpen();
```


Przykładowy program – RangeDevices

```
double angle;  
double range = laser.currentReadingPolar(startAngle, endAngle, &angle);
```



Aria Reference

Przykładowy program – kamera

```
////////////////////////////////////  
// Copyright (c) 2005 Focus Robotics. All rights reserved.  
//  
// Created by      : Jason Peck  
//  
// This program is free software; you can redistribute it and/or modify it  
// under the terms of the GNU General Public License as published by the  
// Free Software Foundation; either version 2 of the License, or (at your  
// option) any later version.  
//  
// This program is distributed in the hope that it will be useful, but  
// WITHOUT ANY WARRANTY; without even the implied warranty of  
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
// General Public License for more details.  
//  
////////////////////////////////////
```

Przykładowy program – kamera

```
#include "cv.h"
#include "highgui.h"
#include "frcam.h"

int main( int argc, char** argv )
{
    FrChannel *channel[2];
    IplImage *frame[2];
    char wndname [2][50];

    int i;

    for(i = 0; i < 2; i++) {
        channel[i] = frOpenChannel(i);
        if (!channel[i]) {
            printf("demo: frOpen failed for device%d\n",i);
            exit(1);
        }
    }
}
```

Przykładowy program – kamera

```
if (frSetChannelInput(channel[0], FR_CHAN_INPUT_TYPE_DISP)) {  
    printf("demo: set INPUT_TYPE failed for device 0\n");  
    exit(1);  
}  
if (frSetChannelInput(channel[1], FR_CHAN_INPUT_TYPE_RT)) {  
    printf("demo: set INPUT_TYPE failed for device 1\n");  
    exit(1);  
}  
  
sprintf(wndname[0], "Disparity");  
sprintf(wndname[1], "Right Calib");  
cvNamedWindow(wndname[0], CV_WINDOW_AUTOSIZE);  
cvMoveWindow(wndname[0], 0, 0);  
cvNamedWindow(wndname[1], CV_WINDOW_AUTOSIZE);  
cvMoveWindow(wndname[1], 0, 520);
```

Przykładowy program – kamera

```
for (i = 0; i < 2; i++) {
    if(frGrabFrame(channel[i]))
        goto out;
}
for (;;) {
    for (i = 0; i < 2; i++) {
        frame[i] = frRetrieveFrame(channel[i]);
        if (NULL == frame)
            goto out;
    }
    cvShowImage(wndname[0], frame[0]);
    cvShowImage(wndname[1], frame[1]);
    for (i = 0; i < 2; i++) {
        if(frGrabFrame(channel[i]))
            goto out;
    }
    if(-1 == frCheckKeypress(stat, cvWaitKey(2)))
        break;
}
```

Przykładowy program – kamera

```
out:
  frCloseStatusTool(stat);
  for (i = 0; i < 2; i++) {
    frCloseChannel(channel[i]);
  }
}
```

Strona MobileRobots

<http://robots.mobilerobots.com>



Serwer Cyklon



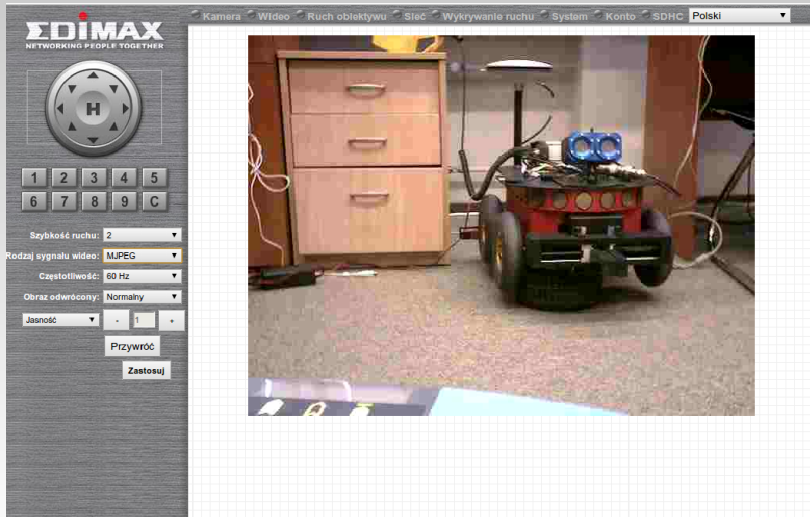
Kinect



Kamery



Kamery



The screenshot displays a web-based camera control interface for an EDIMAX camera. The interface is divided into two main sections: a control panel on the left and a video feed on the right.

Control Panel (Left):

- EDIMAX NETWORKING PEOPLE TOGETHER** logo at the top.
- A central navigation pad with a central 'H' button and four directional arrows.
- A numeric keypad with buttons for 1, 2, 3, 4, 5, 6, 7, 8, 9, and a 'C' button.
- Settings for:
 - Szybkość ruchu: 2
 - Redzaj sygnału wideo: MJPEG
 - Częstotliwość: 60 Hz
 - Obraz odwrócony: Normalny
 - Jasność: 1
- Buttons for 'Przywróć' (Reset) and 'Zastosuj' (Apply).

Video Feed (Right):

- Navigation tabs: Kamera, Wideo, Ruch obiektywu, Sieć, Wykrywanie ruchu, System, Konto, SDHC, Polski.
- A live video stream showing a red robot with blue sensors on a carpeted floor next to a wooden cabinet.

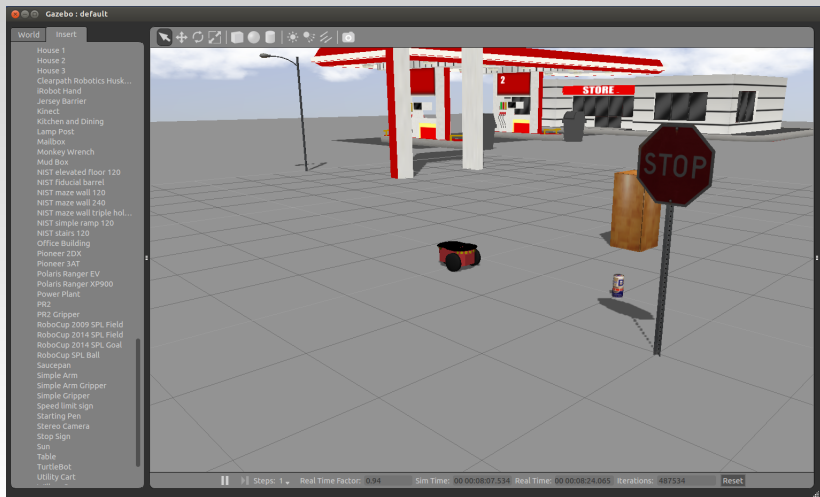
Robot Operating System

<http://www.ros.org>



Gazebo

<http://gazebosim.org/>



KONIEC