Praktikum: 4

Telebot system environment

Lecturers

Manfred Grove
Houxiang Zhang

TAMS, Department of Informatics
University of Hamburg, Germany

@Tams group

Content of today’ s lecture

- Program introduction
- Telebot program environment
  - Overview
  - Example
- Your task

Content of today’ s lecture

- Program introduction
- Telebot program environment
  - Overview
  - Example
- Your task

C++ Introduction

- Structure of a program
- Variables, data types
- Constants
- Operators

- You can get more help in
  - http://www.cplusplus.com/doc/tutorial/
- More online documents.
  - http://tams-www.informatik.uni-hamburg.de/people/hzhang/projects/TelerobotDocument/index.htm

Structure of a program

```
#include <iostream>
 using namespace std;

int main ()
{ cout << "Hello World!" << endl;
  return 0;
}
```

```
Hello World!
```

Please use **g++** compiler.

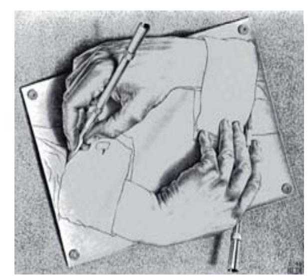
Variables & data types

- Global variables
- Local variables
- Constants
- Operators

```
// operating with variables
#include <iostream>
using namespace std;
int main ()
 {
// declaring variables:
int a, b;
int result;
// process:
a = 5;
b = 2;
a = a + 1;
 result = a - b;
// print out the result:
cout << result << endl;
 // terminate the program:
return 0;
}
```


Control Structures

- **If** and **else**
- Iteration structures (Loops)
  - **while** loop
  - **do-while** loop
  - **for** loop
- The selective structure: **switch**



Functions

- Defining functions and methods
- Calling functions and methods

```
// function example
 #include <iostream>
using namespace std;
int addition (int a, int b)
  {
 int r; r=a+b; return (r);
  }

int main ()
 { int z;
   z = addition (5,3);
   cout << "The result is " << z << endl;
  return 0;
 }
```

The result is 8

Technical Aspects of Multimodal System
Dept. Informatics, Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

Object-oriented Programming (OOP)

- Class
- Object

```
#include <iostream>
using namespace std;
class  CLrect {
float   x,y;
public:
       CLrect(void);
       CLrect(float, float);
  void  set_values(float, float);
  float area() { return(x*y);};
  };

CLrect::CLrect(void)// Konstruktor 1
{}                              // besser z.B.: x= 0.0; y= 0.0;

CLrect::CLrect(float a, float b)
                                // Konstruktor 2
{ x= a; y= b; }

void CLrect::set_values(float a, float b)
  { x= a; y= b; }
```

```
int main()
{ CLrect R;
  CLrect *R2 = new CLrect(3.0, 8.0);
  R.set_values(3.0, 4.0);
  cout << "Fläche: " << R.area() << endl;
  cout << "Fläche2= " << R2->area() <<endl;
  return 0; }
```

Fläche: 12
Fläche2:24

Technical Aspects of Multimodal System
Dept. Informatics, Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

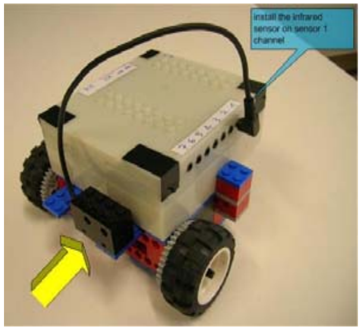
Content of today's lecture

- Program introduction
- Telebot program environment
  - Overview
  - Example
- Your task

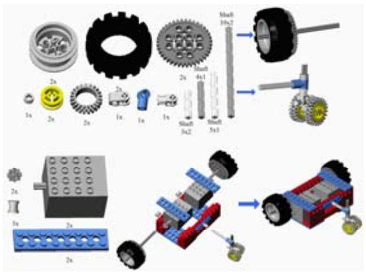
Technical Aspects of Multimodal System
Dept. Informatics, Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

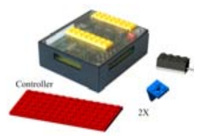
How to program the Telebot?

- First build your telebot
- Task description:
  - If there is an object in front of the telebot, it moves. If not, it stops. The Robot works without connecting to the GUI.



Building the mechanical system





Programming the robot

```
#include "Telerobot.h"                    //import the Telerobot head file
 int main(int argc, char *argv[ ])
{
        Telerobot *t=new Telerobot;   //create a telerobot object
        t->robotConnect();             // connect the telerobot and your PC
        t->initial();                  //initialization the telerobot
        t->ask();                      //gets the sensor feedback
        while (!t->Stop_flag)          //check if mission shoud be canceled
          { if (!t->Pause_flag)          //check if mission should be paused
               { if (t->getSensor1()) t->moveForward(90);
                                       //if sensor1 is true robot moves forward with 90% speed
                 else t->stop();       //else robot stops
               };
             t->checkGuiCommand();
          };
        t->missionOver();              //mission over, buzzer pipes.
        t->robotDisconnect();          //disconnect the telerobot
}
```


Compiling the program

- Compile your code with **g++**
  - **g++ -L. –o destination_file_name code_file_name.cpp -lTelerobotLibrary**
  - or
  - **g++ -L. –o destination_file_name code_file_name.cpp Telerobot.cpp**
- The API library can be found on the TAMS web page.
  - **http://tams-www.informatik.uni-hamburg.de/people/hzhang/projects/TelerobotDocument/index-Dateien/Page751.htm**

Correct use of sensors in your program

- Task: Move forward until an obstacle is found

Wrong !!

```
…..
t->moveForward(50);
while (!getSensor1())
      ;
 t->stop();
 ………
```

Correct

```
…..
t->moveForward(50);
while (!getSensor1())
      t->ask();
//
// or   t->moveForward(50);
//
 t->stop();
 ………
```

Using analogue sensors (1)

- Sensorchannels 6 and 7 can supply analogue values, when you connect one of the optical sensors to them (8 Bit)
- For every method using digital sensors there is a corresponding method using analogue sensors on channel 6 and 7
- For example:

Digital: ask() Analogue: ask_Asensor()

moveRight(50) moveRight_Asensor(50)

Using analogue sensors (2)

- You can get the analogue values by calling the methods

getAnalogueSensor6()

and/or

getAnalogueSensor7()

Both return an (unsigned) char (1 byte)

Writing standalone programs

- You don't need MCGUI to run a program using the Telebot.
- If you are writing a standalone program you should

! Delete all calls to

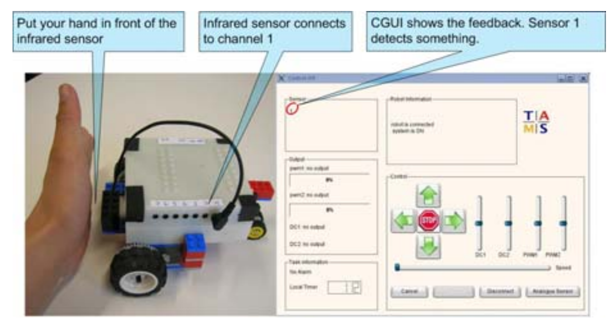
checkGuiCommand() (important !)

! After creating the telebot-object call the method

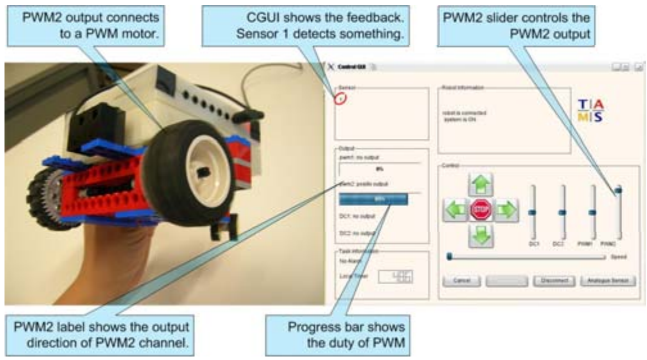
setmyoutput(true)

Testing your soft code with the Telebot

- First check your hardware with CGUI application

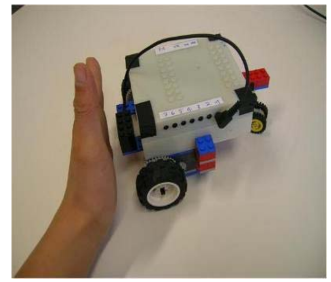


Testing your soft code with the Telebot



Testing your soft code with the Telebot

- Execute your program on a Linux console



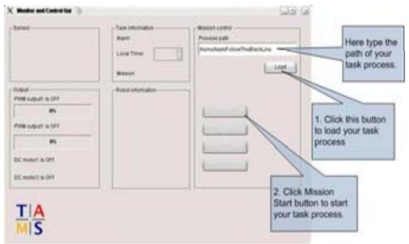
Content of today’s lecture

- Program introduction
- Telebot program environment
  - Overview
  - Example
- Your task

Your task (after a 15-min-break)

- Following a line

Execute your code on a Linux console

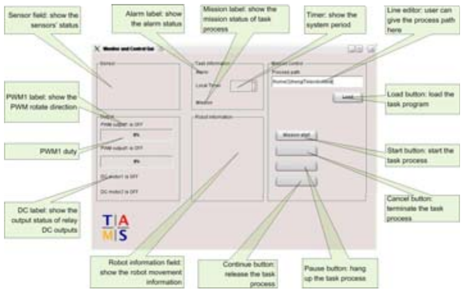


Execute your code on the MCGUI(Monitor and Control GUI)

Technical Aspects of Multimodal System
Dept. Informatics, Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

How to use MCGUI?

- Type in the execute code path
- Load
- Start the mission
- Terminate the mission

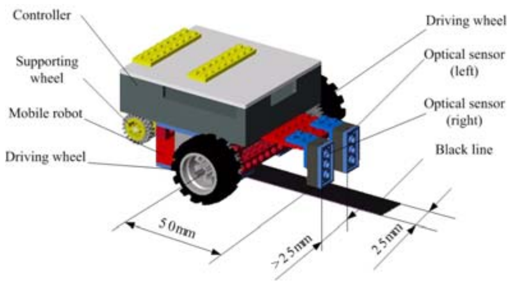


Dr. ZHANG, Houxiang hzhang@informatik.uni-hamburg.de
Institute TAMS **Technical Aspects of Multimodal Systems** http://tams-www.informatik.uni-hamburg.de/hzhang

Technical Aspects of Multimodal System
Dept. Informatics, Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

Implementation

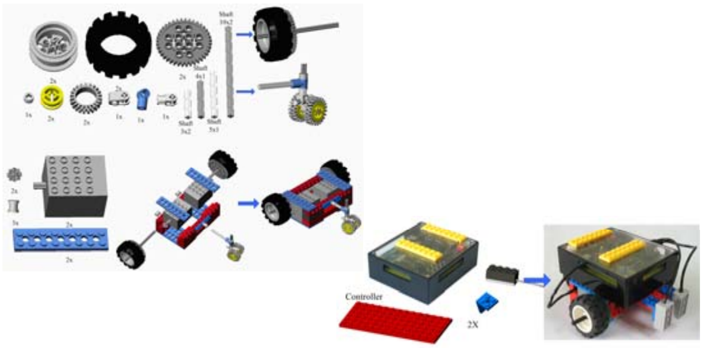
- Building the mechanical system
- Programming
- Testing



Dr. ZHANG, Houxiang hzhang@informatik.uni-hamburg.de
Institute TAMS **Technical Aspects of Multimodal Systems** http://tams-www.informatik.uni-hamburg.de/hzhang

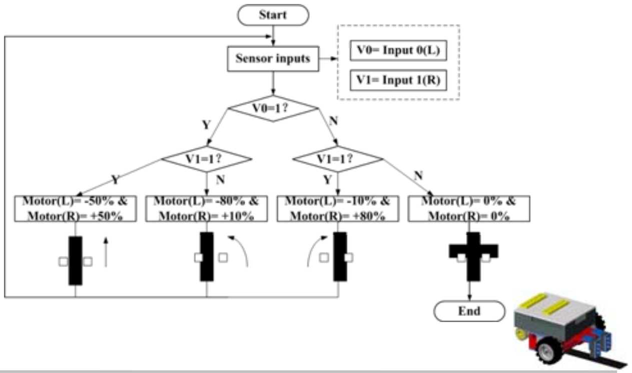
Technical Aspects of Multimodal System
Dept. Informatics, Faculty of Mathematics, Informatics and Natural Sciences
University of Hamburg

Building the mechanical system



Dr. ZHANG, Houxiang hzhang@informatik.uni-hamburg.de
Institute TAMS **Technical Aspects of Multimodal Systems** http://tams-www.informatik.uni-hamburg.de/hzhang

Programming the robot



Do it on your own…

Praktikum: 5 & 6

Telebot sensors and actuators

Lecturers

Houxiang Zhang
Manfred Grove

TAMS, Department of Informatics
University of Hamburg, Germany

@Tams group

Thanks for your attention!

Any questions?