Turun yliopisto
University of Turku

Advanced Sensor Networking

Ethiopia Nigussie
Communication Systems
Department of Future Technologies

Course Information

- Teaching language: English
- Advanced level course, 5 ECTS credits
- Instructor and mentors in UTU:
  - Ethiopia Nigussie
    - email: ethiopia.nigussie@utu.fi , Office: 454C
  - Nanda Thanigaivelan
    - nanda.k.thanigaivelan@utu.fi, 454B
  - Jawad Yasin
    - janaya@utu.fi,
- Course focuses on project and team work
  - Not a lecture course!
- Web site:
  - http://runestone.it.uu.se/

About the Course

- The course is run co-operatively in two universities simultaneously:
  - Uppsala University, Sweden (GMT+1)
    - University lecturer Arnold Pears
  - University of Turku, Finland (GMT+2)
    - Adj. Prof. Ethiopia Nigussie
- Each university has its own local practices for the course
- For participants, the course is known as ”project Runestone”

Uppsala

Turku

Suomi
Finland
Sverige
Sweden
Norge
Norway
Россия
Russia
GEOGRAPHY
Polska
Poland
Deutschland
Germany
Україна
Ukraine
Kazakhstan
Mongolia
ance
Italia
Italy
Türkiye
Turkey
대한민국
S Korea
日本
Japan
中国
China
Iraq
Iran
Afghanistan
Pakistan
India
geria
Libya
مصر
Egypt
Saudi
Arabia
ประเทศไทย
Thailand
Niger
Chad
Sudan

About the Course

- Spring term duration, breaks, and periods are different in each of the participating universities
  - All teams should be composed within next few weeks (will be done by the instructors)
  - After this, teams start working and present their progress about weekly to instructor
  - The goal is to have all final presentations by the end of May

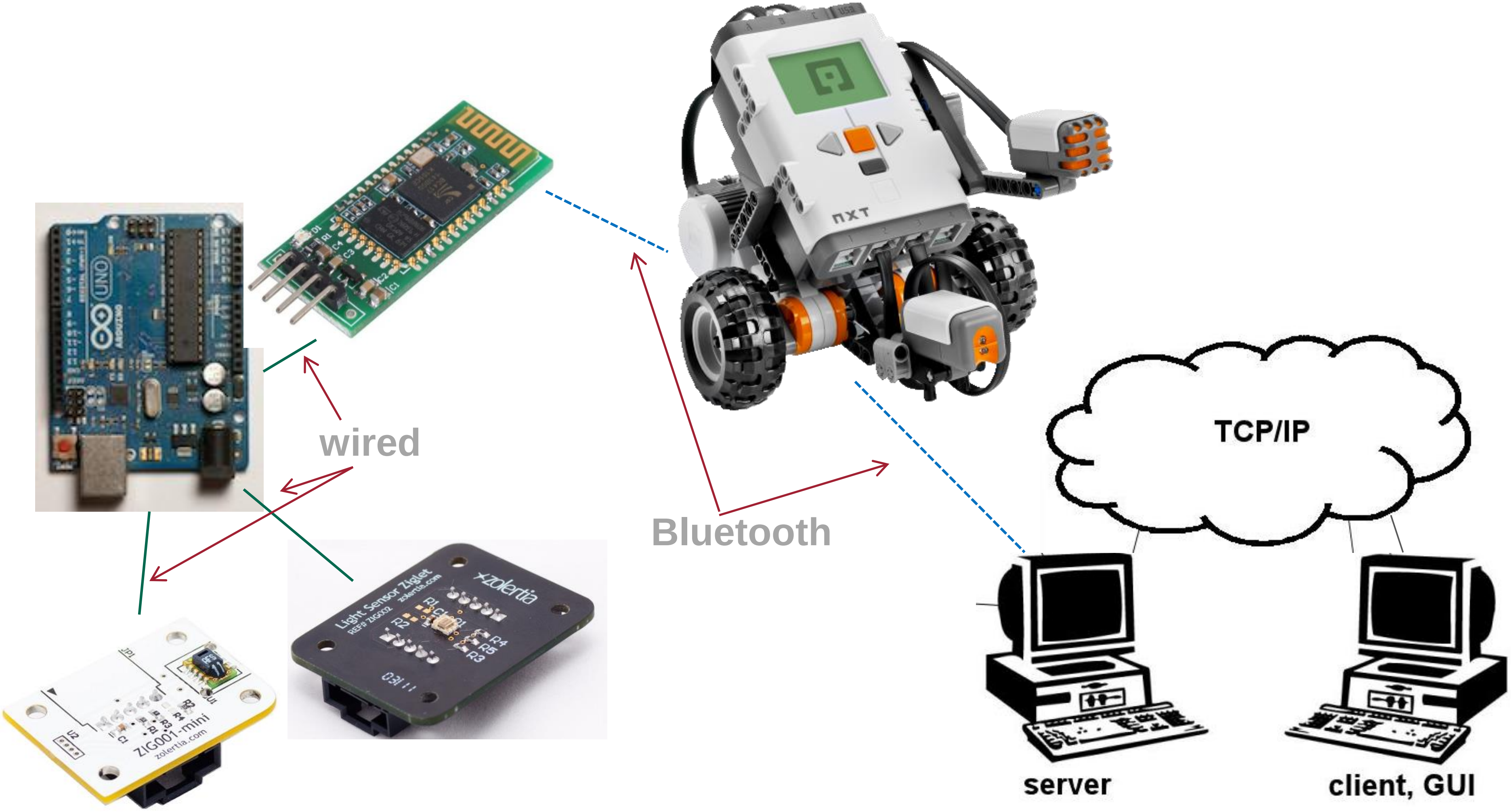
Student Teams

- Teams are composed of students from different universities
  - The team needs to choose a team leader
  - The team decides which software to use for its internal meetings, and for coding
  - Also need to take into account time zones
  - The team reports to its instructor in Milestone meetings using a communication tool determined by the instructor (probably Skype)
  - Milestone meetings occur about weekly, depending on the schedules of team members and the instructor

Problem to be Solved

- Assuming that a smart truck is unloading food/medicine or other stuffs to a large storage facility
- It has to unload the stuffs in a place where certain temperature and lighting conditions are met
- While navigating the storage facility it has to avoid obstacles and collisions
- The users should be able to see the route the truck takes remotely for checking purpose

Problem to be Solved



wired

Bluetooth

TCP/IP

server

client, GUI

Problem to be Solved

- The LEGO robot act as a smart truck which gets temperature and lighting information from the Arduino board through Bluetooth communication
- Temperature and light sensors are connected to the Arduino board
- The route taken by the robot can be visualized by the remote user from PC or android phone

Technical Task

- Design a distributed system which can perform the smart unloading and check the unloading route over the internet
  - Robot navigates its route autonomously by taking the required information from the Arduino board
  - Server communicates with one or more robots to provide real-time route information and command relay between a selected robot and a remote Control Client

System Requirements

- Teams should develop a system that meets the basic specification stated on the previous slides
- Robustness and ease of installation of the final system should be addressed during development

Course Aims

- Integrate existing knowledge and skills
- Build additional competence in
  - Project management
  - Virtual development teamwork
  - Cultural issues in teamwork
- Experience a full development cycle from conception to delivery for a small scale system
- Provide opportunities to learn professional skills with close mentorship in educational setting
  - Team management
  - Development timelines

Project Method

- Project structure
  - A large part of the project concerns requirements analysis as well as research and design of the system
  - There is no single correct answer, each team is expected to develop a requirements specification and design in collaboration with the other members of the team and the team mentor
  - Creativity and capacity to identify and synthesize relevant knowledge and apply it to the problem are vital aspects of obtaining a pass grade
- Teamwork
  - Each development team consists of students from Turku and Uppsala
  - Each team will have regular online meetings with a designated mentor at one of the Universities participating in the project

Development Server

- Redmine project management system
- Administered in Uppsala
- http://runestone.it.uu.se/
  - Course information
  - Communication forums
  - Team websites
  - Project calendar
  - Project planning and management tools
  - SVN repository

Deliverables (more info on the Wiki)

- Analysis and Design document
  - Specify how you will solve the problem
  - Provide a schedule/plan for the project: what you will do at each milestone
  - Allocate work: which team member works on which subsystem (e.g. video feed, robot software, Arduino programming, TCP/IP software, final presentation, ...)
- Milestone reports and presentations (4)
  - Four milestones
  - Report your progress to the instructor
- Final report and presentation, demo
  - Present the results of your team’s work, and reflect the results to what you planned in the analysis and design document

Milestone Meetings

- The team assigns one team member as the responsible person for preparing each milestone presentation and report
  - For each milestone, a written report must be posted onto the team’s own page on the Wiki system BEFORE the milestone meeting
  - The team needs to prepare for the meeting:
    - First, present your status on the things required for the milestone meeting to the instructor
    - Second, inform the instructor of any relevant on-going things in the team or the team’s project
    - If you need more information on something, send your questions in a clearly formatted email to your instructor at least 24h before the milestone meeting
  - Each meeting and report is graded!
- The milestone meetings last about 30 minutes, often less

Milestone 1

- About 1 week after team configurations have been announced:
  - Internal team building (leader selection, tool selection, getting to know team members)
  - Create a web page (onto the Wiki system) for the team and a profile page for each team member
  - Prepare CVS/SVN for team use
  - Start working on the team's Analysis and design document
  - Milestone meeting: present web page, present profile pages, present status of design document

Milestone 2

- About a week after Milestone 1
  - Complete the analysis and design document
  - Present your analysis and design document to the instructor
  - Instructor may request some modifications to the document
  - You will have a little time (probably less than a week) to make modifications to the document, after which it is graded

Milestones 3 and 4

- Progess reports based on design document
  - Milestone 3 about 2 weeks after MS2
  - Milestone 4 about 2 weeks after MS3
  - After milestone 3 you should be at a stage where you start integrating the subsystems (each team member's individual contributions) into a working system
  - At about milestone 4 you should already be also working on your final presentation
- Final presentation about 2 weeks after MS4

Grading

- The project and ultimately the course is examined by continuous assessment
- To pass the course you need to compile a personal online "portfolio" on the project management Redmine site
- Your portfolio should provide convincing evidence that you have
  - Made a continuous contribution to achieving the project learning goals, and
  - Participated actively in team activities and meetings, and
  - Contributed productively to compiling knowledge for the use of the team, and writing reports and making online, and face to face presentations

Grading

- Maximum score: 100 points
    - Teamwork, communication, project management **50** points
        - 5 points for presentation, 5 points for project management and documentation available at each milestone meeting
        - Analysis and Design Document max 10 points
            - Analysis and design document also affects milestone performance indirectly in terms of project management (for example, being a lot behind in schedule as determined in the team’s design document shows bad management)
    - Individual performance and contribution max **10** points
        - Assessed by team members, max 5 points
        - Assessed by team’s instructor, max 5 points
    - Technical Achievement max **40** points
        - Technical Innovation 20 points based on how well the team succeeds in their selected areas of excellence
        - Final presentation 20 points
- To pass: 80 points

The Robot Kits

- We have Mindstorms NXT Bluetooth robot kits
- The robot kit your team uses depends on the available hardware at each location
- The Robot kits are located in 454D
  - Take the kits from 454D and return it when you leave
  - You can not take the robots home
- Use the computer room 126A/B or K127 for carrying out the project tasks
  - There is also a dedicated PC for this course, with server capabilities, USB and Bluetooth
    - Teams may install software they need for this course
    - The server features MAY NOT be used for anything else but testing your robot implementations! Any server software MUST BE shut down after tests are finished.
    - Instructor provides you with server port numbers once you need them
- Each group borrow nodes, sensors and accessories from the instructor

Programming the Robot

- Each kind of kit comes with its own robot software development environment
  - This software is generally too simple for the application

- Java-enabled firmware also available
  - For example, LeJOS: http://www.lejos.org/nxj.php
  - LeJOS allows you to program the robot in Java
  - Tutorial: http://www.lejos.org/nxt/nxj/tutorial/index.htm

- The group can decide the programming tool

Code Storing

- The course web site in Sweden will also host an SVN service for storing and versioning your code and sharing it with team members

- Each team member IS REQUIRED TO use SVN for storing team code!
  - Other team members can modify files
  - Instructors can monitor the teams’ progress
  - Files are available from any computer

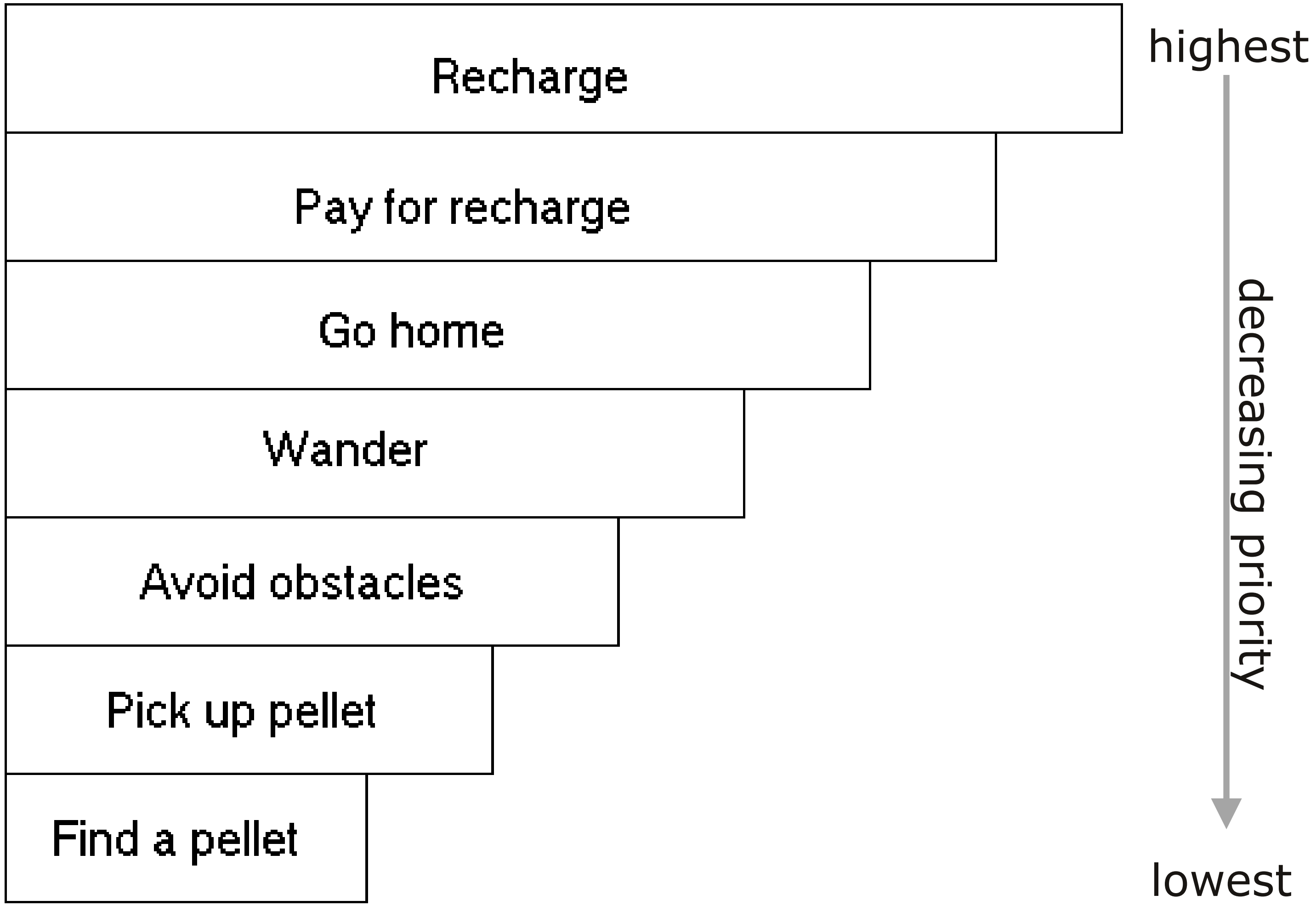
Ultrabrief Introduction to Robotic

- Traditional (old-fashioned) robotics
  - Model the world
  - Use robot’s sensors to figure out where in the world it is, and act accordingly
  - Problem: what if the world changes
- Modern behavior-based robotics
  - The robot does not need to know ”anything” of the world, so the world need not be modified
  - The robot only receives information from its sensors and reacts with pre-programmed actions and behaviors
  - Some behaviors have a higher priority than others subsumption architecture
  - Makes emergent behavior possible (Robot does ”more” than it is programmed to do as a combination of its simple reaction behaviors)

Example Behavior-based Robot

- Robot specification in subsumption architecture
  - No prior knowledge of the world is provided
  - Mission: explore the world, stay alive
    - Staying alive: must return to home base every 60 sec for recharging, must pay 1 pellet for it
    - In order to pay pellets for recharging, the robot must search and collect pellets as it explores
    - Robot may not touch boundaries of its environment
  - Turtle geometry: two individually controlled wheels
  - Two laser proximity sensors, 10% error margin
  - One homing beacon, provides an angle reading for the location of the home base, 10 % error margin
  - Recharging station detector: indicates, when the robot has reached the recharging station
  - A clock for counting down the 60 seconds: return home when no more than 35% of time left

Example of subsumption



Recharge

Pay for recharge

Go home

Wander

Avoid obstacles

Pick up pellet

Find a pellet

highest

decreasing priority

lowest

Team Work

- Managing a project is about more than the technical content

- Getting a team to work well involves:
  - Clear communication
  - Commitment (sometimes called "buy in")
  - Flexible and accountable process

- Some desirable practices include:
  - Don't let conflict hide in the background
  - Face and address passivity
  - Make objectives explicit and documentable