New Perspectives in Science Education

International Conference
NEW PERSPECTIVES in SCIENCE EDUCATION

Interdisciplinary Science Project on Student Learning: Synchronization Problems in Computer Science

Haklin Kimm
Computer Science
East Stroudsburg University, USA

Florence, Italy

March 17-18, 2016

Outline

- Introduction: Interdisciplinary project
- Distributed Robotics Systems over CAN
- Synchronization Problem: Queen and Messengers
- Lessons learned
- Conclusions

Introduction

- Project Initiation
  - Project goals: what to achieve (wish -> can)
    - The students' backgrounds are from Computer science, Physics, Mathematics, Engineering Physics;
    - Study more on what we can do;
      - Survey studies related to Robotics and Controller Area Network;
      - Collect distinctive features that others can not do well;
    - Team meetings to revamp the goals
      - Discuss several project goals – feasible to work;
      - Result in the goals that the team can achieve;

Introduction

- Project Team Build
  - Team and project goal
    - A goal for a team – Distributed Robotics on CAN
    - Cooperation of two groups for complex goals
      - One for Robotics control and programming
      - The other for network-related setup and programming
    - Each group is with an informal student project manager;
      - At this time CPSC graduate students were selected;

Distributed Robotics over CAN

- Distributed Robotics Environment
  - Work in harmony with
    - Sensors, Actuators, Computers, Users
  - Multiple robots operate synchronously or asynchronously;
  - Tasks are scheduled to perform trying to achieve efficiency in time usage and resource share.

Distributed Robotics over CAN

- Network Consideration for networked or distributed robots
  - Bus access type
  - Bandwidth
  - Control options
  - Number of components over the bus
  - Power level and others

Distributed Robotics over CAN

- In event-triggered systems computers, sensors and actuators are in demand driven.
  - The computers process upon receiving sensor signals, where a sensor with a changing value immediately sends a message to the computer.
  - The actuators behave upon signals from the computers.
- In time-triggered systems,
  - Sensors are polled regularly to provide any changes or signals;
  - The actuators are working periodically even without signals from computers.

CAN

- CAN (Controller Area Network)
  - Initiated in1981 and became ISO in 1994;
  - Message frames can be transmitted at the speed of up to 1 Mbps;
  - 500 Kbps is for engine control, ABS and cruise control;
  - 125 bps is for comfort electronics;
  - CAN system supports CSMA (Carrier Sense Multiple Access);

CAN

- What is CAN?
  - Controller Area Network
  - Widely used in automotive and manufacturing systems
  - Spec'd in 1986 by Bosch, Mercedes and Intel
  - A protocol
- Where is CAN used?
  - Automotive and manufacturing systems
  - Possible space technology applications
    - ie) CubeSAT

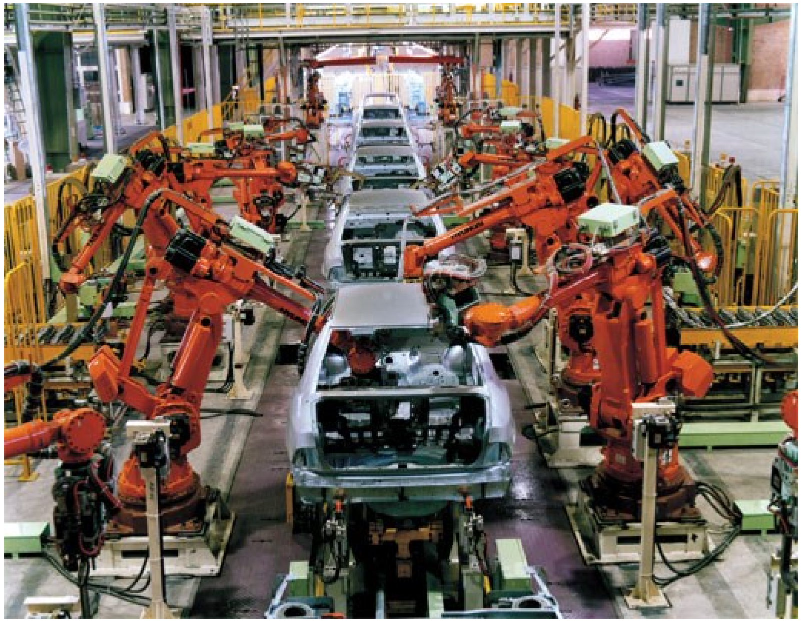
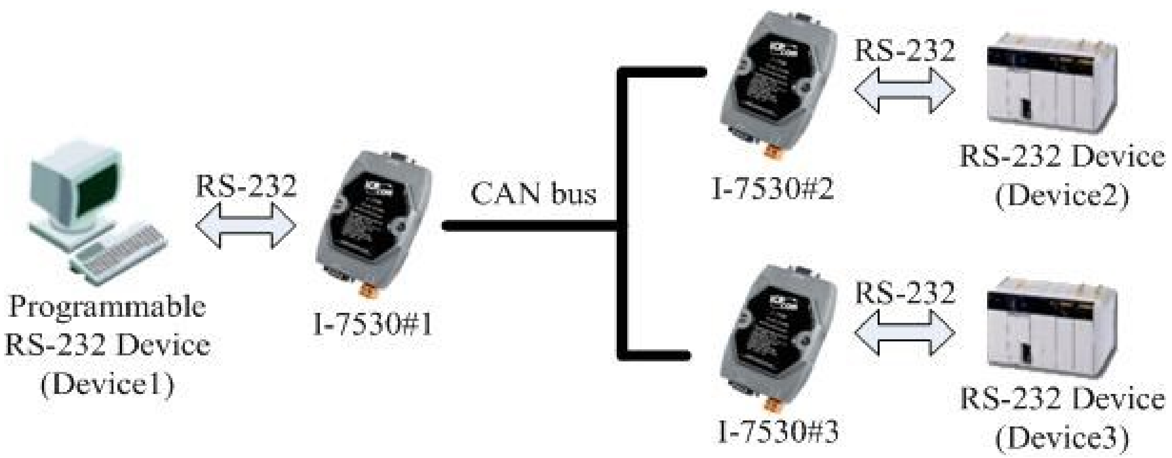


Photo: Industrial Export Development Co.

CAN

- How does **CAN** work?
  - Simple, two-wire bus cable
  - Non-destructive bus arbitration
- Why is *CAN* used
  - Highly suitable to real time systems
  - Low cost
  - High reliability

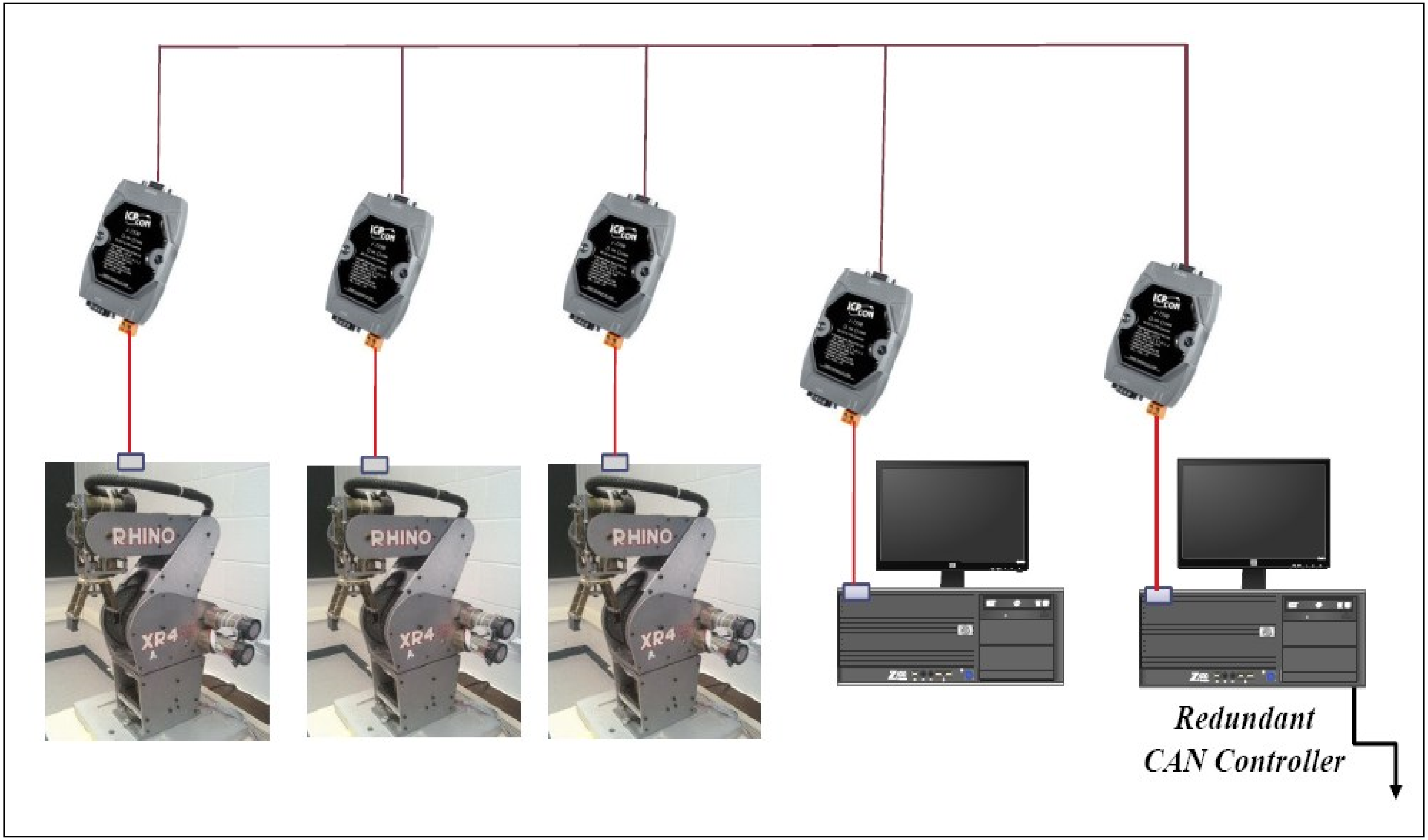


CAN Protocol

- Specific frame format
  - SOF
  - Arbitration (ID) field
  - Data, CRC, EOF etc
- Non-destructive arbitration
- Wired-and logic
  - Dominant bits override recessive bits
  - Equivalent to a logical AND

Field	SOF	Arbitration		Control			Data	CRC		ACK		EOF
	SOF	IDENTIFIER	RTR	IDE	r0	DLC	Data	C_S	C_D	A_S	A_d	EOF
Length	1	11	1	1	1	4	0...64	15	1	1	1	7
Value	0	0...2031	0	0	0	0...8	x	X	1		1	127

Distributed Robotics over CAN



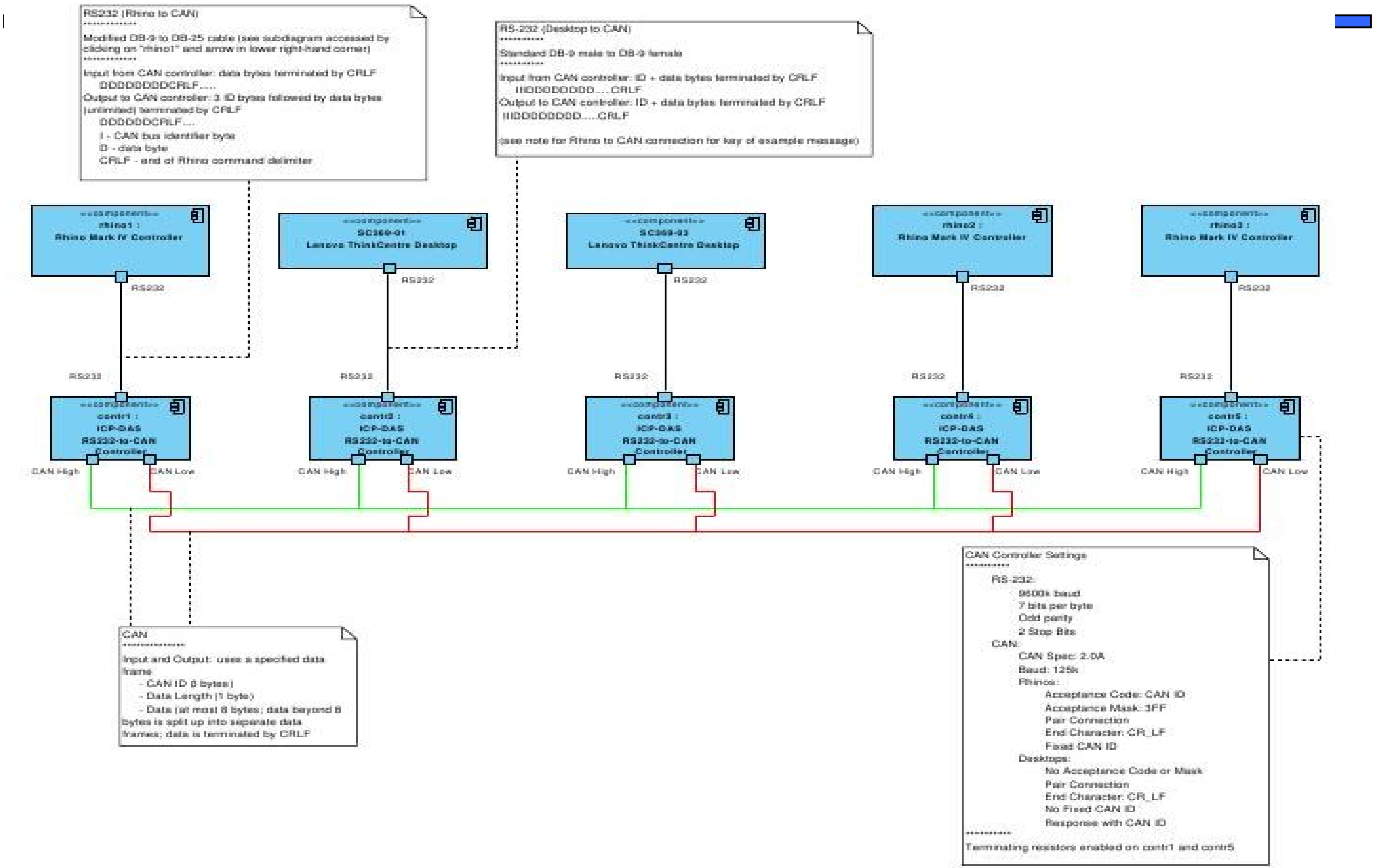
Distributed Robotics Testbed

- Prototype and testing environment
  - *CAN* network connecting 5 nodes
    - 2 control workstations and 3 Rhino XR4 robotic arms
  - 5 CAN converters at each node
    - Performs CAN to RS232 conversion, if necessary
  - Goals
    - Research platform representative of the technology in the field
    - Allows us to develop components and algorithms to support this technology

Software Environment

- Main controller's code handles
  - Serial port initiation
  - Rhino robot initiation
  - Macro job launching
  - Interactive control via direct command
  - Inverse kinematics calculation and control
  - Monitoring and status querying

UML Diagram for Distributed Robotics over CAN



Software Environment

- C and C++ code to handle centralized and distributed control;
  - Synchronization problem among CAN nodes;
- Speaks ASCII to RS232-to-CAN converters;
  - RS232-to-CAN converters translate ASCII characters to Binary data;
- Distributed controller's program
  - Handles locally-connected robot I/O
  - Accepts defined commands to initiate tasks
  - Simulates a microcontroller and movable camera module

Software & Development

- Two "macro" jobs defined in main controller code
  - Synchronous task
  - Cooperative task
- Challenges and solutions
  - Serial communication routines solved the problems seen in other terminal applications;
  - Synchronization problems occurred while implementing tasks over distributed robots over CAN;
  - Software I/O is threaded, allowing concurrent send/receive;

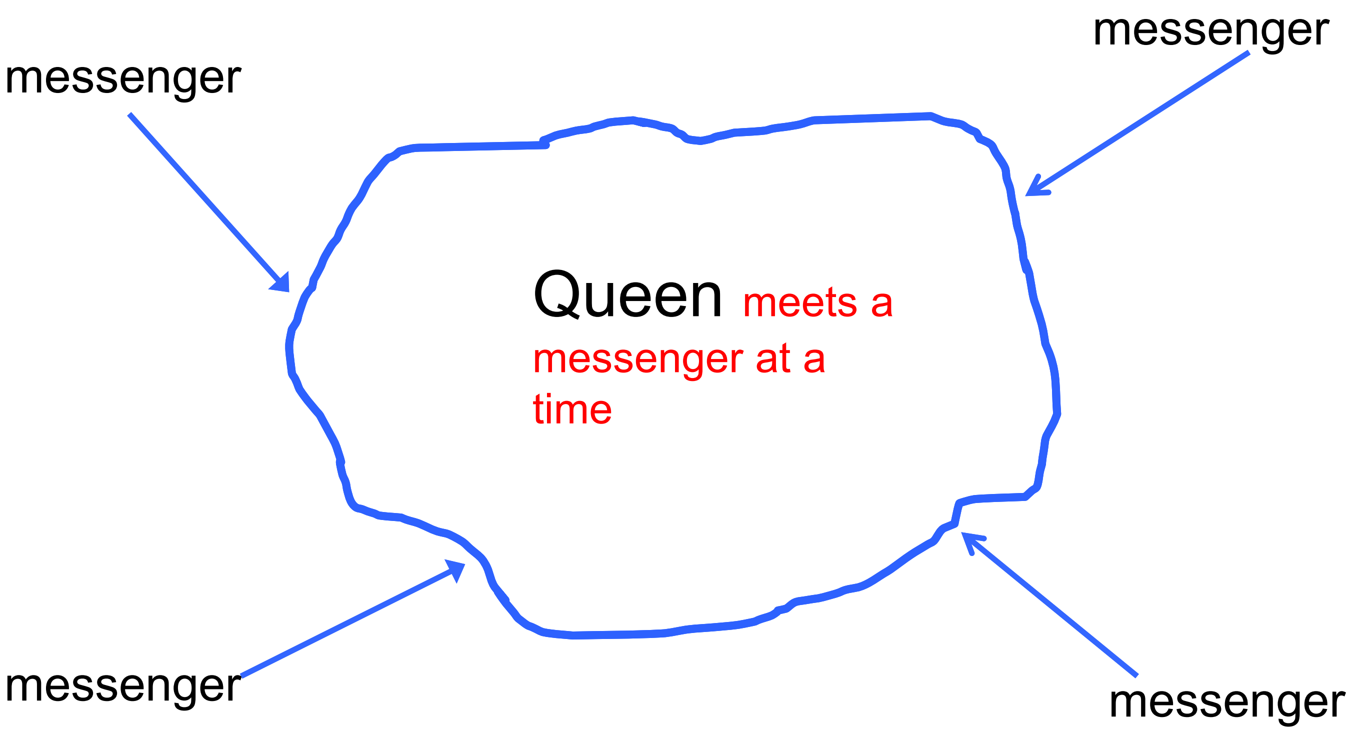
Syncrhonization Problem

- Over the CAN, the distributed robots had to be restarted many times because of synchronization problems;
- In order to resolve the problems that we ran into, we needed to know whether the bus access type, bandwidth, control option, timing and other factors were suitable enough for the robots' synchronous or asynchronous programs;
- The team failed to develop the safe programs that would resolve the critical section problems;

Syncrhonization Problem

- It is somewhat difficult for students to develop the proper synchronization programs for the project;
- The practical situation of distributed robots over CAN has been remapped to a new and easy-to-understand synchronization problem, “Queen and Messengers” ;
- The messengers are represented as robots while the queen is considered a host computer;
  - the queen’s antechamber allows only one messenger who can send a message to the queen;

Syncrhonization Problem: Queen and Messengers



Messengers arrive asynchronously

Demonstration

- Synchronized move
  - Three robots perform same moves via control from main controller
- Cooperative move
  - Robots move independently to complete a more complex task
- Kinematics calculations and movement

Cooperative Task Demo

- Arms work independently
- Central command from controller workstation
- Distributed control realized by CAN converters
- Implements serial & CAN I/O functionality
- Implements robot control class

Lessons learned to Share

- Starting the Interdisciplinary Project
  - **How** to recruit students
  - **Where to** go to recruit them.
    - At ESU, there are certain classes related to the CubeSat projects: Operating Systems, Networking, Software Engineering, Computer Architecture and Organization
    - Most students in the team were strongly encouraged to take a Physics class – Basic Electronics.
      - Extra help from other senior members in the team.
  - **How many** students would be good enough for making the project progress.

Lessons learned to Share

- Accomplishments
  - Implementation of Distributed Robotics over CAN bus
  - C/C++ libraries and classes
    - Distributed Robotic control and I/O
    - Synchronous Serial and CAN I/O and message handling
    - Implementation of kinematics for robotic control
- Discoveries
  - Serial I/O problems over CAN converters
  - Hardware communication “disagreements” and workarounds
  - Synchronization problem: “Queen and Messengers”

Conclusions

- The interdisciplinary projects enhances the students' learning on the synchronization problems.
- The the synchronization problems of the project had beeen applied to a compuer science class successfully.
- Most students in the operaring systems class were able to complete the synchronization programs of the practical Queen and Messengers.

Conclusions

- It is a clear indicator of showing how much successful when thearitical computer sicence concepts such as synchronization, scheduling, deadlock problems were taught with more practical examples;
- Later it is expected for us collecting and analyzing the students programs upon synchrinization problems using classiscal problems and the developed problems from the interdisciplinary science project: Distributed Robots over Controller Area Network.

Questions