RS232 communication module in MLD and applications

Borisav Jovanovic

Introduction

- RS232 communication module
- Validation of communication module
- Hardware In the Loop Simulation
- System with sensors and pumps

RS232 Communication module

Module **comRS232,** which sends and receives string data packets over RS232 was implemented in ML Designer. To accomplish the task, the DE domain was chosen.

comRS232 provides C++ routines for opening the RS232 port, setting its properties (baud rate, parity), sending and receiving data .

Using **comRS232**, two computers with MLD programs running, can communicate in both directions. Another more important application is Hardware-In-The-Loop (HILS) simulation.

Input and output ports: **OpenPortTrigger**, **ReadTrigger**, **psOutput**, **Display** and **psResponse**.

Any float type particle that comes on input port **OpenPort Trigger** will open the serial port and therefore enable communication.

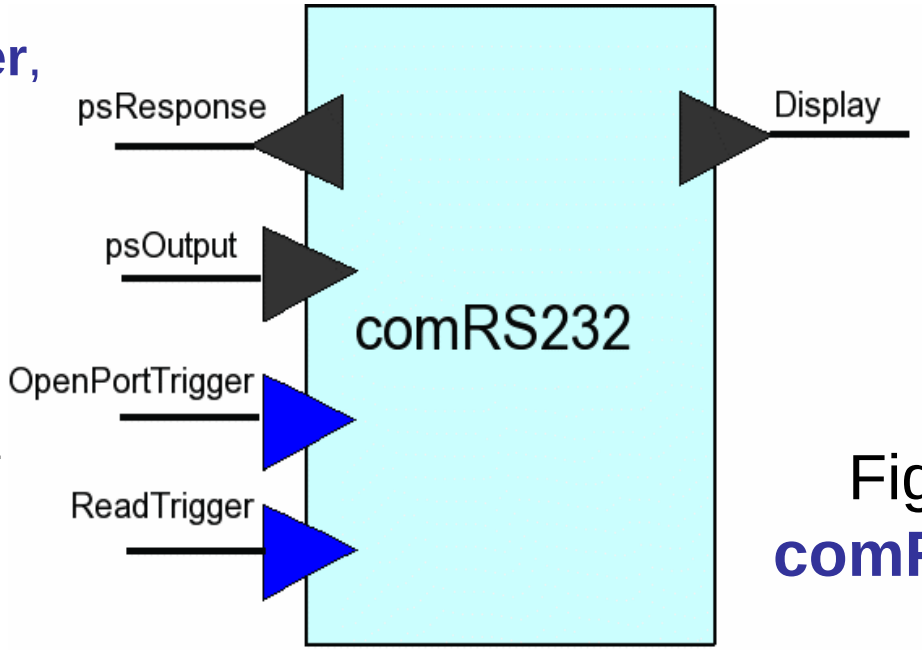

Fig.1: **comRS232**

Reading operation: **ReadTrigger** puts **comRS232** into ***reading mode*** which is active until a character (or array of characters) is read. After, received characters appears in form of string on **psResponse** output.

Writing operation: data on input **psOutput** will be transferred via RS232.

Output port **Display** is used for ***displaying parameters***: what port is open, what are the speed rate, number of bits in data packets, parity and other RS232 parameters. Also, **Display** shows characters that have been received or sent.

Validation of RS232 Communication module

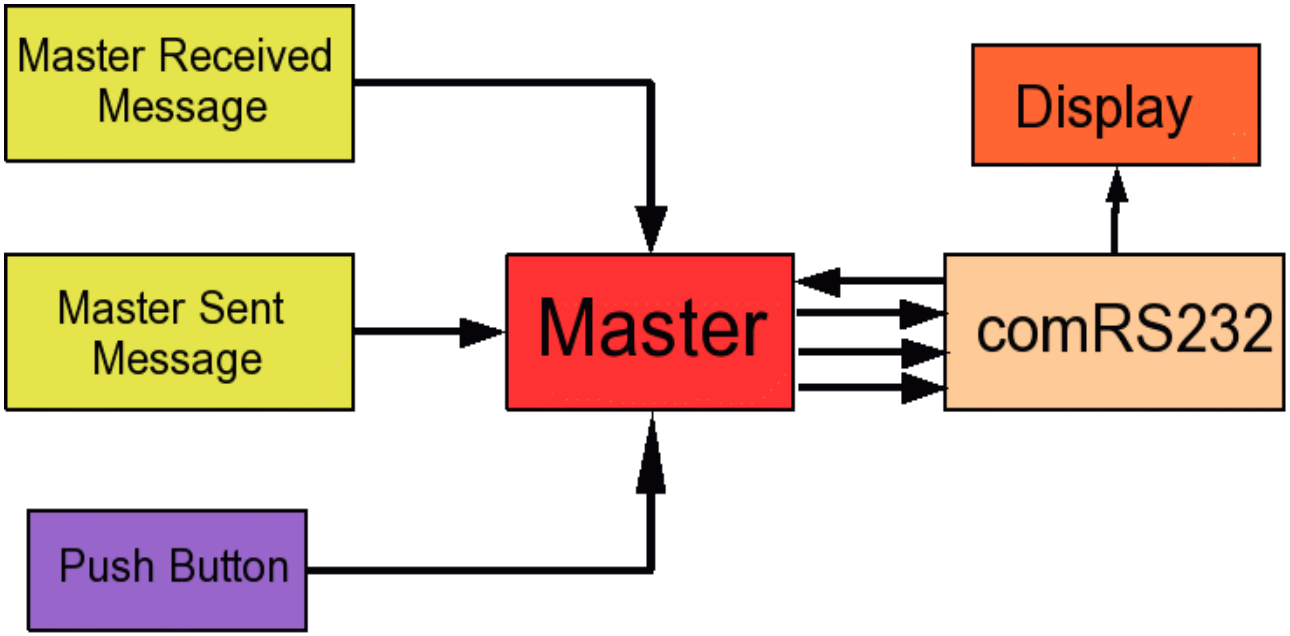

Fig.2: Master system running on one computer

Validation of RS232 Communication module

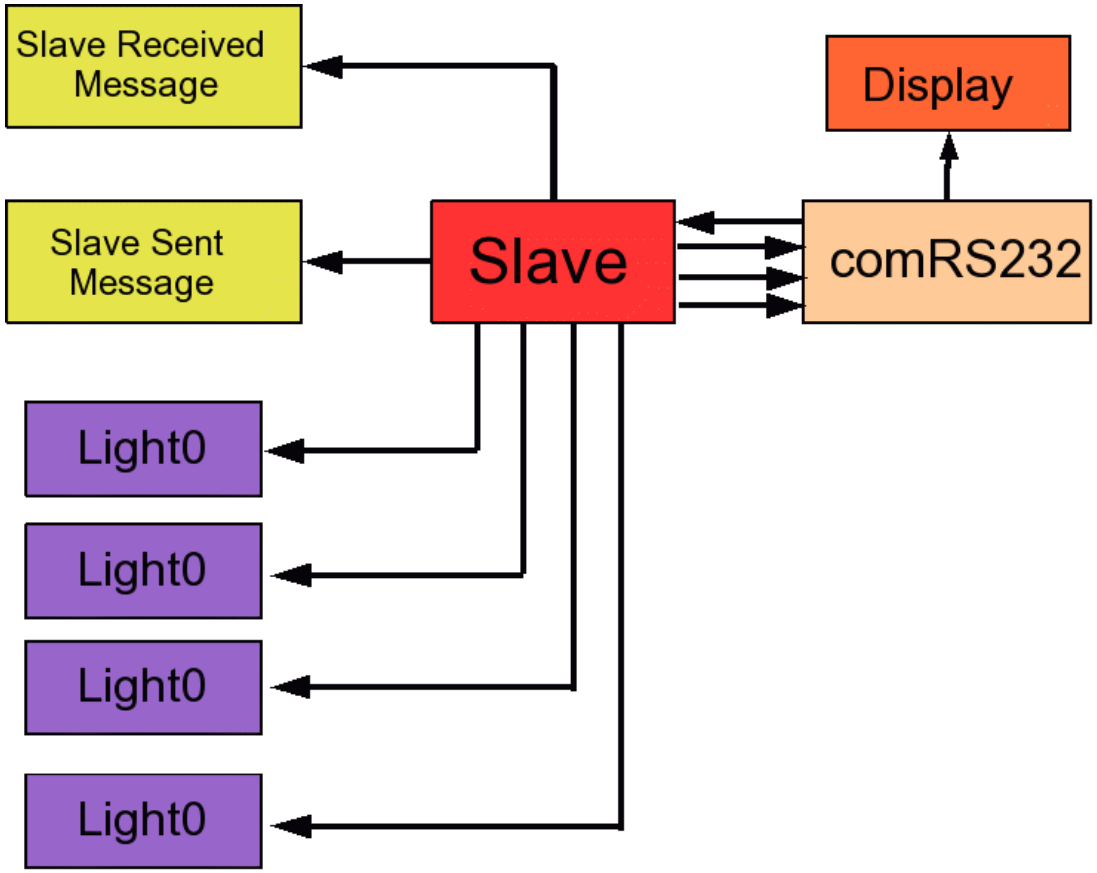

Fig.3: Slave system running on the other computer

Hardware In the Loop Simulation

- Systems can be rapidly developed using graphics based software
- For most real systems, there are characteristics that are unknown or too complex to model by pure simulation.
- Good system engineering practice would begin with a pure simulation and as components become better defined (with the aid of simulation), they can be fabricated and replaced in the control loop. This is called Hardware In the Loop Simulation (HILS).
- HILS approach eliminates expensive and lengthy iterations in machining and fabrication of parts, and speeds development towards a more efficient design;

System with sensors and pumps

It consists of some pool filled with some fluid, two fluid pumps (P0 and P1), four sensors (S0, S1, S2, and S3) and control subsystem (consisting of ML Designer program running on PC and PIC microcontroller).

Finite State Machine was left implemented in MLD. Sensors and actuators models, modeled previously in high-level simulation, were replaced by real hardware.

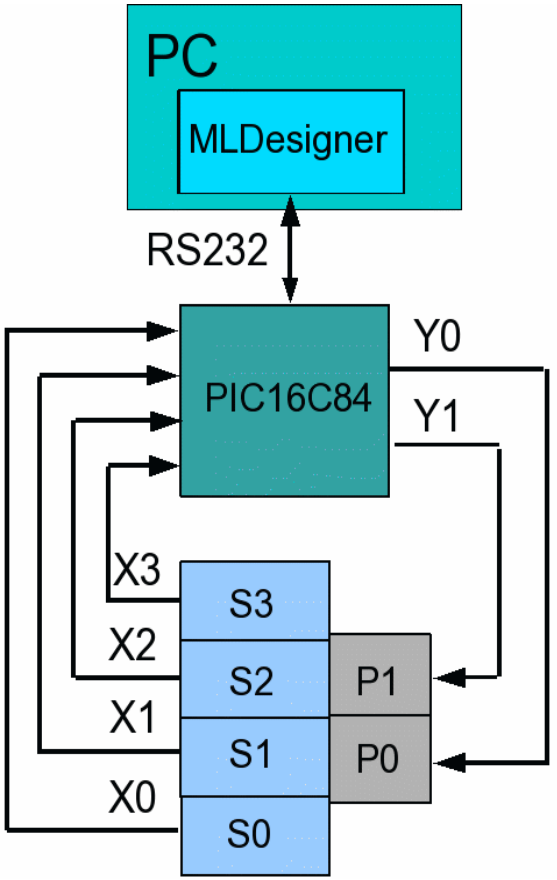

Fig.4: Block diagram of a HILS system

System with sensors and pumps

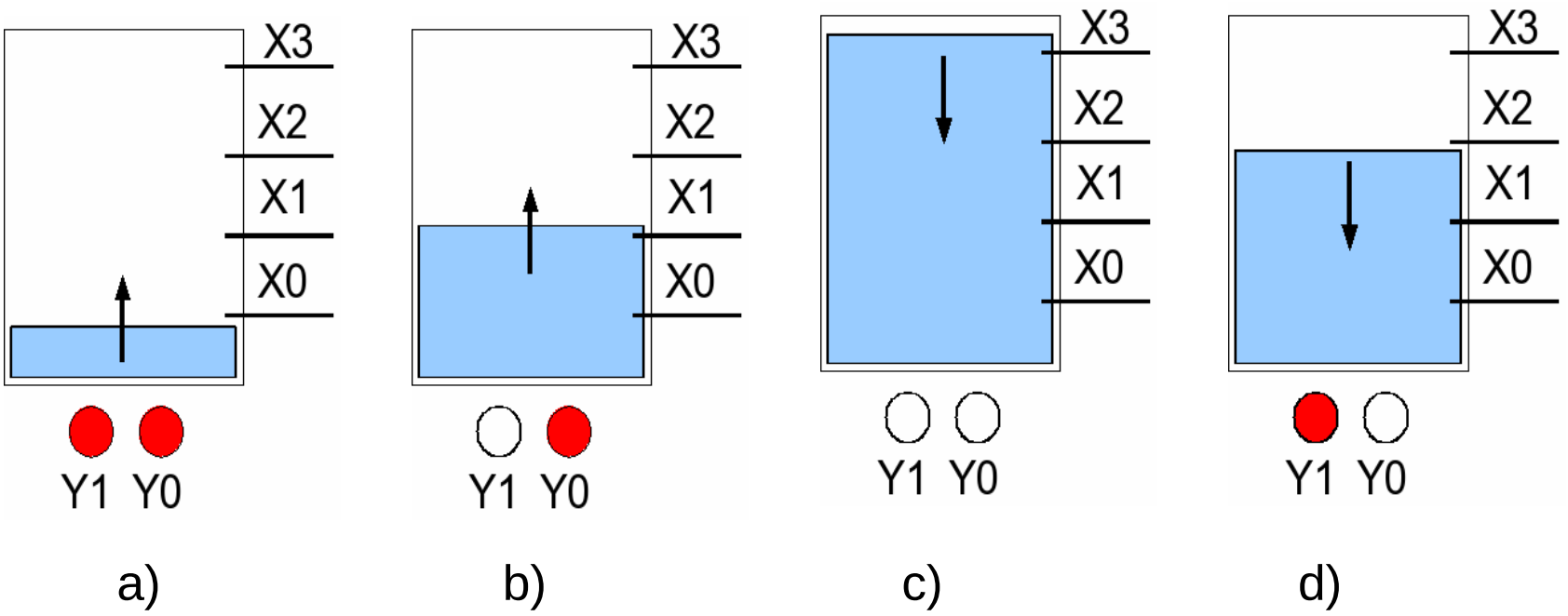

Figure 5: Pump's functioning

System with sensors and pumps

System functioning :
PIC controller periodically senses state of sensors X0 to X3. After, it sends sensor’s information to PC using its RS232 routines implemented in Assembler.

Fig.6: Pool with pumps

FSM in MLD receives sensor's information and gives the appropriate respond on its outputs. After that, this respond, in form of a command, is sent from MLD to PIC controller which turns on or off the pumps.

System with sensors and pumps

Modules: **comRS232**, **Slave**, **Control unit** and several text displays **Slave Received Message**, **Slave Sent Message**, and **Display**.

The functioning: **Slave** periodically receives from **comRS232** 8-bit sensors status information. When received it is shown on **ReceivedMessage** display.

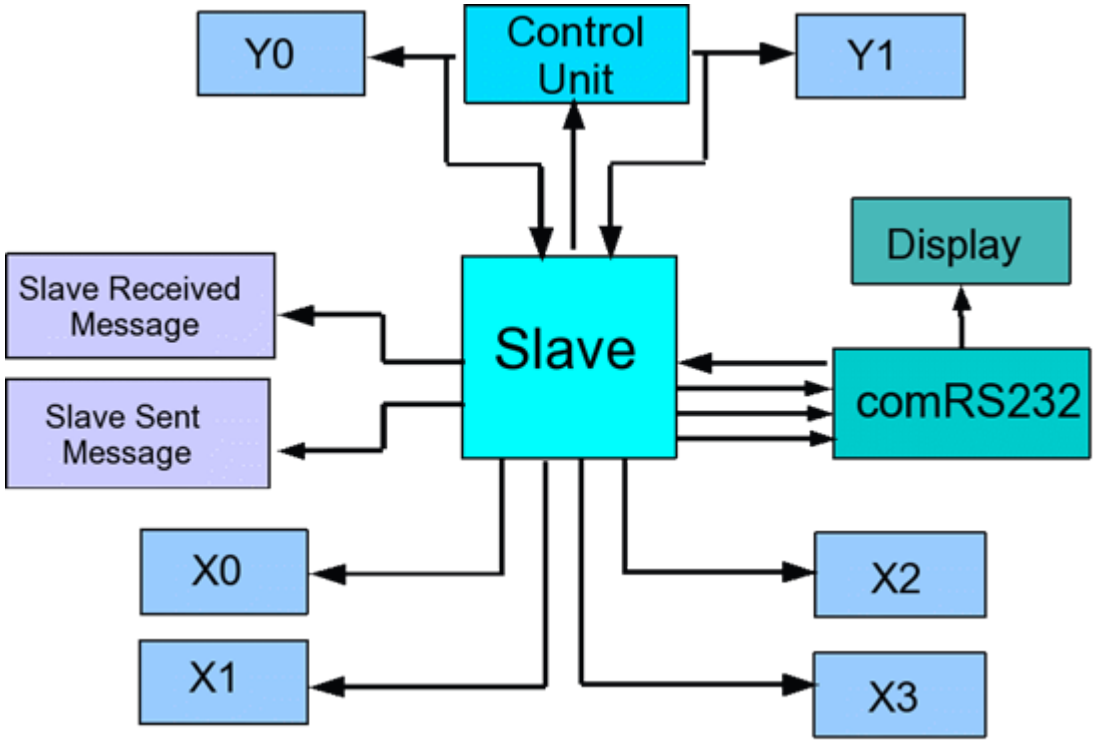

Fig.7. Part implemented in MLD

After, sensor status is given to the **Control Unit** which calculates new pump's inputs. After, **Slave** module sends new pump's states command to the **comRS232** and over RS232 to PIC controller.

System with sensors and pumps

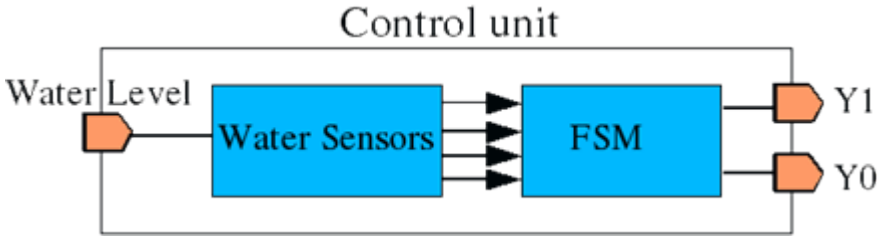

Fig.8. Control Unit in MLD

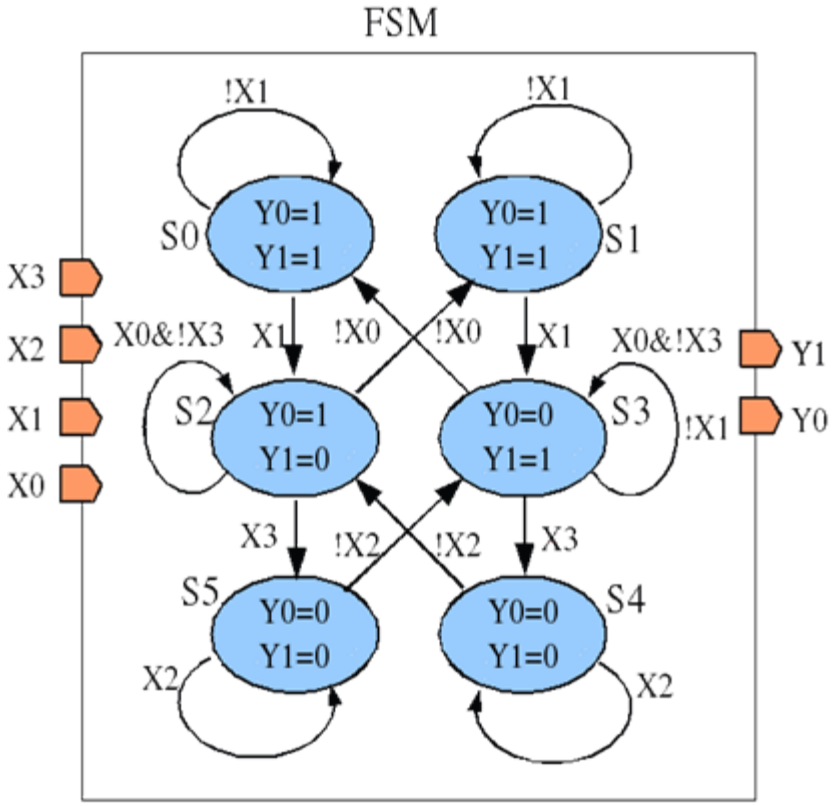

Fig.9. Finite State Machine implemented in MLD

System with sensors and pumps

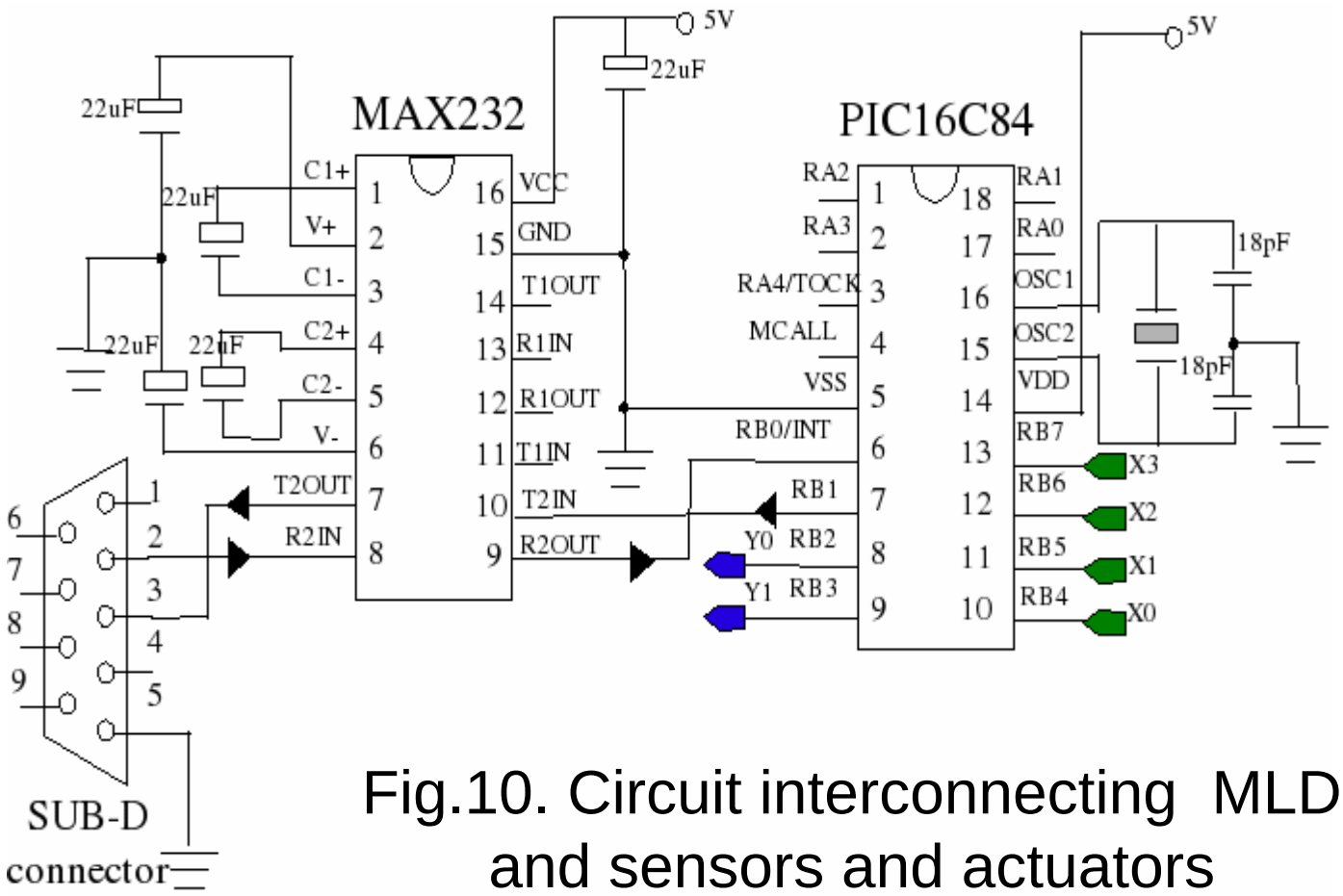

Fig.10. Circuit interconnecting MLD and sensors and actuators

System with sensors and pumps

PIC16C84 periodically senses (over RB4 to RB7) logical levels on sensor's outputs (X0 to X3). Since fluid level doesn't change abruptly, this sensor acquisition period is chosen to be 25ms.

Then, PIC16C84 has to create one byte of information and send it over RB1 output pin and after, RS232 serial cable to ML Designer.

Based on received sensor status, after every period of 25ms, ML Designer program calculates new pump states, which then has to be coded into appropriate byte of data and sent back to PIC16C84.

Whenever is a new command sent to PIC16C84, the microcontroller has to receive it and turn on or off the pumps according to its value.