# Control systems, smart sensors, controller, elements in a control loop, Java Card and Security

NDERIM ZEQIRI and ARTAN LUMA
Communication Sciences and Technologies
SEE University
SEE University, Ilindenska bb. CST, Office 305.12, 305.01, 1200, Tetovo
MACEDONIA
n.zeqiri@seeu.edu.mk; a.luma@seeu.edu.mk; www.seeu.edu.mk

*Abstract*: - An automatic control system is a preset closed-loop control system that requires no operator action. This assumes the process remains in the normal range for the control system. Elements of automatic control. Finding a transfer function for block Diagram of CNC machine-tool control system. The control system design will attempt to evaluate the system response by determining a mathematical model for the system. We have described the generalized schematic block-diagram for a closed loop, or feedback control system. Feedback loop is the signal path from the output back to the input to correct for any variation between the output levels from the set level. Programming serial and parallel port. The broad usage of Java language and Java Cards in various different fields has made them quite applicable and implementable in different applications and systems.

*Key-words*: Automation, Control, System, Serial port, Parallel port, Java, Security

## 1 Introduction

One of the fundamental aims has been to work from basic principles and to present control theory in a way that is easily understood and applied. However, it is recognized that powerful automation control method and software packages exits to aid control system design. However, all systems have certain things in common. They all for example, require inputs and outputs to be specified. In control engineering, the way in which the system outputs respond in changes to the system inputs (i.e. the system response) is very important. The control system design will attempt to evaluate the system response by determining a mathematical model for the system. We have described the generalized schematic block-diagram for a closed loop, or feedback control system. The controller (or computer), and device lie along the forward path, and the tachogenerator and shaft encoder are in the feedback path. The measured value of device output is compared at the summing point with desired value. The difference or error is fed to the controller which generates a control signal to drive the device until its output equals desired value. Such an arrangement is sometimes called an error-actuated system.

Instrumentation and process control involve a wide range of technologies and sciences, and they are used in an unprecedented number of applications.

Examples range from the control of heating, cooling, and hot water systems in homes and offices to chemical and automotive instrumentation and process control.

In industrial control a wide number of variables, from temperature, flow, and pressure to time and distance, can be sensed simultaneously. All of these can be interdependent variables in a single process requiring complex microprocessor systems for total control. Due to the rapid advances in technology, instruments in use today may be obsolete tomorrow, as new and more efficient measurement techniques are constantly being introduced. These changes are being driven by the need for higher accuracy, quality, precision, and performance. To measure parameters accurately, techniques have been developed that were thought impossible only a few years ago.

Process control is the automatic control of an output variable by sensing the amplitude of the output parameter from the process and comparing it

to the desired or set level and feeding an error signal back to control an input variable.

For computer communication and device control needed-interface between computer and device. So, to have access in the device, we must create interface between computer and device. Also, we use serial port, or parallel port and we have write program to realize this interconnection [1], [2].

## 2 Control systems

System mean different things to different people and include purely physical systems such as table of Computer Numerically Controlled (CNC) machine tool or alternatively the procedures necessary for the purchase of raw materials together with the control of inventory in a Material Requirements Planning (MRP) system [1]. It necessary to clearly define the boundary of a system, together with inputs and outputs that cross that boundary. Fig. 1.
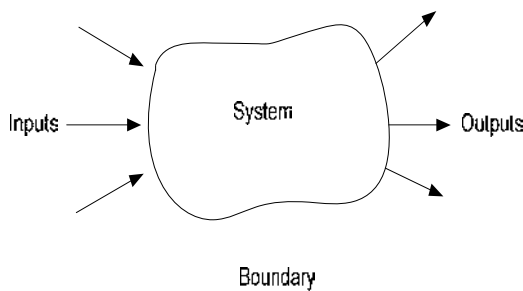


Fig. 1 The concept of a system

### 2.1 Process Control

A temperature sensor attached to the outlet pipe senses the temperature of the water flowing. As the demand for hot water increases or decreases, a change in the water temperature is sensed and converted to an electrical signal, amplified, and sent to a controller that evaluates the signal and sends a correction signal to an actuator. The actuator adjusts the flow of steam to the heat exchanger to keep the temperature of the water at its predetermined value.
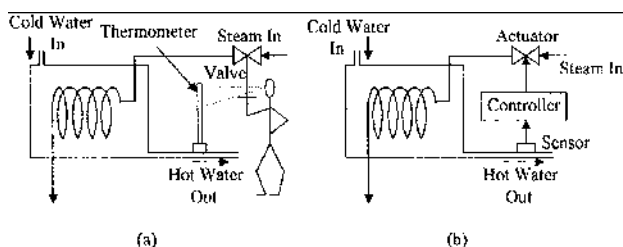


Fig. 2 Process controls (*a*) shows the manual control of a simple heat exchanger process loop and (*b*) automatic control of a heat exchanger process loop.
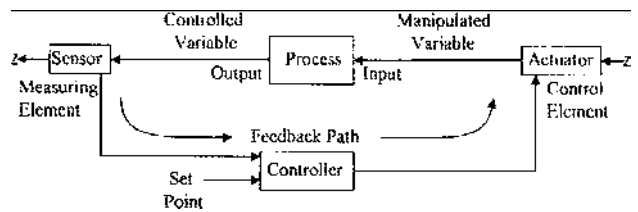


Fig. 3 Block diagram of a process control loop.

### 2.2 Elements in a Control Loop

Figure 4 breaks down the individual elements of the blocks in a process-control loop. The measuring element consists of a sensor, a transducer, and a transmitter with its own regulated power supply. The control element has an actuator, a power control circuit, and its own power supply. The controller has a processor with a memory and a summing circuit to
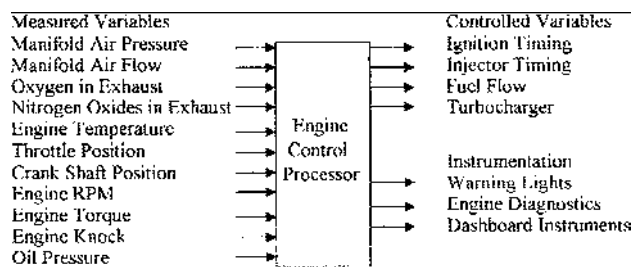


Fig. 4 Automotive engine showing some of the measured and controlled variables.

compare the set point to the sensed signal so that it can generate an error signal. The processor then uses the error signal to generate a correction signal to control the actuator and the input variable.

*Feedback loop* is the signal path from the output back to the input to correct for any variation between the output levels from the set level. In other words, the output of a process is being continually monitored, the error between the set point and the output parameter is determined, and a correction signal is then sent back to one of the process inputs to correct for changes in the measured output parameter [7], [14].

*Controlled or measured variable* is the monitored output variable from a process. The value of the monitored output parameter is normally held within tight given limits.

*Manipulated variable* is the input variable or parameter to a process that is varied by a control

signal from the processor to an actuator. By changing the input variable the value of the measured variable can be controlled.

*Set point* is the desired value of the output parameter or variable being monitored by a sensor. Any deviation from this value will generate an error signal.

*Instrument* is the name of any of the various device types for indicating or measuring physical quantities or conditions, performance, position, direction, and the like.

*Sensors* are devices that can detect physical variables, such as temperature, light intensity, or motion, and have the ability to give a measurable output that varies in relation to the amplitude of the physical variable. The human body has sensors in the fingers that can detect surface roughness, temperature, and force.
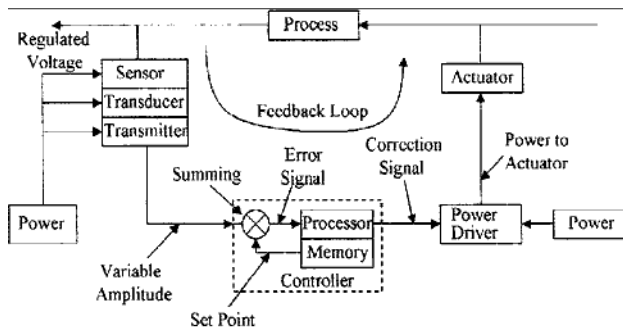


Fig. 5 Block diagram of the elements that make up the feedback path in a process-control loop.

In other sensors such as a diaphragm pressure sensor, a strain transducer may be required to convert the deformation of the diaphragm into an electrical or pneumatic signal before it can be measured.

*Transducers* are devices that can change one form of energy to another, e.g., a resistance thermometer converts temperature into electrical resistance, or a thermocouple converts temperature into voltage. Both of these devices give an output that is proportional to the temperature. Many transducers are grouped under the heading of sensors.

*Converters* are devices that are used to change the format of a signal without changing the energy form, i.e., a change from a voltage to a current signal.

*Actuators* are devices that are used to control an input variable in response to a signal from a controller. Atypical actuator will be a flow-control valve that can control the rate of flow of a fluid in

proportion to the amplitude of an electrical signal from the controller. Other types of actuators are magnetic relays that turn electrical power on and off. Examples are actuators that control power to the fans and compressor in an air-conditioning system in response to signals from the room temperature sensors.

*Controllers* are devices that monitor signals from transducers and take the necessary action to keep the process within specified limits according to a predefined program by activating and controlling the necessary actuators.

*Programmable logic controllers* (PLC) are used in process-control applications, and are microprocessor-based systems. PLCs, which have the ability to use analog or digital input information and output analog or digital control signals, can communicate globally with other controllers, are easily programmed on line or off line, and supply an unprecedented amount of data and information to the operator. Ladder networks are normally used to program the controllers.

*An error signal* is the difference between the set point and the amplitude of the measured variable.

*A correction signal* is the signal used to control power to the actuator to set the level of the input variable.

*Transmitters* are devices used to amplify and format signals so that they are suitable for transmission over long distances with zero or minimal loss of information. The transmitted signal can be in one of the several formats, i.e., pneumatic, digital, analog voltage, analog current, or as a radio frequency (RF) modulated signal. Digital transmission is preferred in newer systems because the controller is a digital system, and as analog signals can be accurately digitized, digital signals can be transmitted without loss of information. The controller compares the amplitude of the signal from the sensor to a predetermined set point [12], [13].
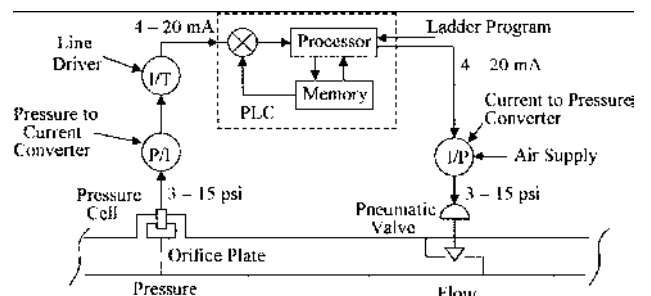


Fig.6 Process control with a flow regulator

## 2.3 Smart sensors

Smart sensor is a name given to the integration of the sensor with an ADC, processor, and DAC for actuator control and the like; such a setup for furnace temperature control is shown in Figure 7.
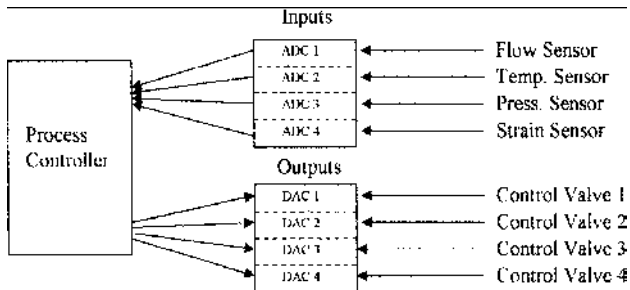


Fig. 7 Process system with individual inputs and outputs for each variable.

The electronics in the smart sensor contains all the circuits necessary to interface to the sensor, amplify the signal, apply proportional, integral, and derivative (PID) control, sense temperature to correct for temperature variations in the process if required, correct for sensor nonlinearity, the ADC to convert the signal into a digital format for the internal processor, and the DAC to convert the signal back into an analog format for actuator control. The processor has a serial digital bus interface for interfacing via the field bus to a central computer. This enables the processor in the smart sensor to receive update information on set points, gain, operating mode, and so on, and to send status information to the central computer [2], [8].
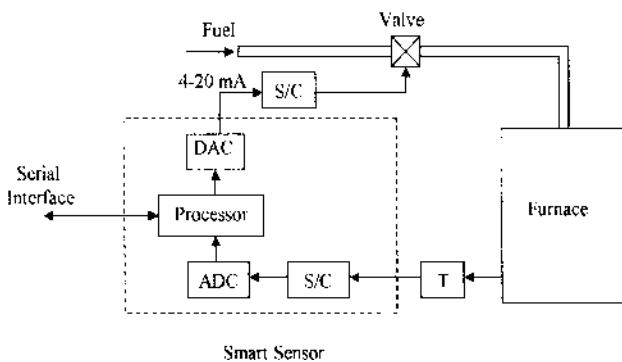


Fig. 8 Smart sensor block diagram.

## 2.4 Controller

Due to the complexity and large number of variables in many process control systems, microprocessor based Programmable Logic Controllers (PLC) are used for decision making. The PLC can be configured to receive a small number of inputs (both analog and digital) and control a small number of outputs or the system can be expanded with plug-in modules to receive a large number of signals and simultaneously control a large number of actuators, displays, or other types of devices. In very complex systems, PLCs have the ability to communicate with each other on a global basis and to send operational data to and be controlled from a central computer terminal. Figure 9 shows a typical controller setup, monitoring a single variable. The output from the sensor is conditioned and transmitted to the input module of the controller; if the signal is an analog signal it is converted to a digital signal and compared to a reference signal stored in the computer. A decision can then be made and the appropriate control signal sent via the output module to the actuator [2], [9].
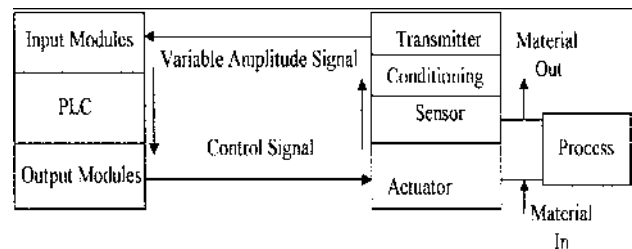


Fig. 9 Block diagram of a control loop.
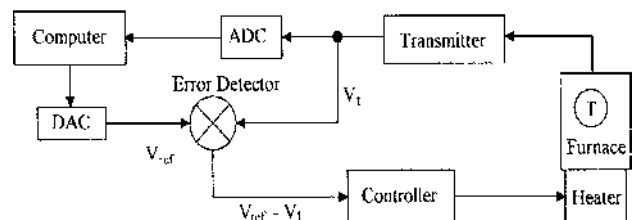


Fig. 10 Supervisory control using an analog control loop.

The internal control function of a furnace using an analog loop is shown in Fig. 10. The temperature of the furnace is transmitted to the computer in the PLC as an analog signal, where it is converted to a digital signal and recorded in the computer memory. The voltage reference signal Vref is converted to an analog signal and the two signals are subtracted as shown in an analog error differencing circuit. The amplified difference signal is then fed via a control circuit to the furnace heater, making the control loop analog.

## 3 Automatic Control Systems

An automatic control system is a preset closed-loop control system that requires no operator action.

This assumes the process remains in the normal range for the control system. An automatic control system has two process variables associated with it: a controlled variable and a manipulated variable.

A *controlled variable* is the process variable that is maintained at a specified value or within a specified range. In the previous example, the storage tank level is the controlled variable.

A *manipulated variable* is the process variable that is acted on by the control system to maintain the controlled variable at the specified value or within the specified range. In any automatic control system, the four basic functions that occur are:

- Measurement
- Comparison
- Computation
- Correction

## 3.1 Elements of Automatic Control

The three functional elements needed to perform the functions of an automatic control system are:

- A measurement element
- An error detection element
- A final control element

## 3.2 Computer Numerically Controlled (CNC) machine tool

Many systems operate under computer control, and Fig. 11 shows an example of CNC machine tool control system.

Information relating to the shape of the work-piece and hence the motion of the machine table is stored in a computer program. This is relayed in digital format, in a sequential form to the controller and is compared with digital feedback signal from the shaft encoder to generate a digital error signal. This is converted to an analogue control signal which, when amplified, drives a d.c servomotor Connected to the output shaft of the servomotor is a lead-screw to which is attached the machine table, the shaft encoder and a tachogenerator [3], [7], [11].

The purpose, of this latter device, which produces an analogue signal proportional to velocity, is to form an inner, or minor control loop in order to dampen, or stabilize the response of the system.

The block diagram for the CNC machine tool control is shown in Fig. 12.
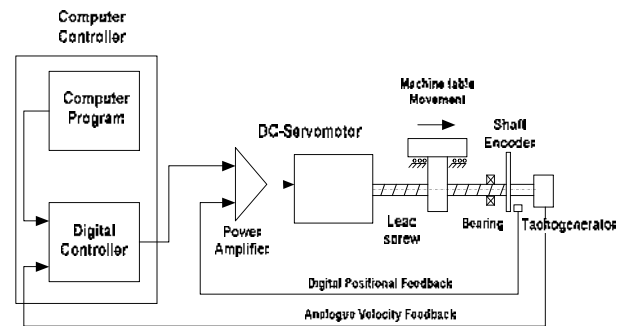


Fig. 11 Computer numerically controlled machine tool
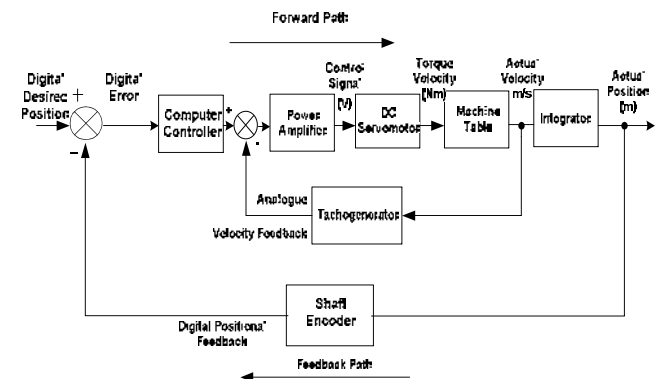


Fig. 12 Block Diagram of CNC machine-tool control system

A transfer function is a Laplace transform of a differential equation with zero initial conditions. It is very easy way to transform from the time to the $s$ domain and a powerful tool for the control engineer.

In order to compute the time response of dynamic system, it is necessary to solve the differential equations (system mathematical model) for given inputs. There are a number of analytical and numerical techniques available to do this, but the one favored by control engineers is the use of the Laplace transform. This technique transforms the problem from the time (or $t$) domain to the Laplace (or $s$ domain). The advantage in doing this is that complex time domain differential equations become relatively simple $s$ domain algebraic equations. When a suitable solution is arrived at, it is inverse transformed back to the time domain.

The Laplace transform of a function of time $f(t)$ is given by the integral:

$$L[f(t)] = \int_0^\infty f(t)e^{-st}\,dt = F(s) \qquad (1)$$

Where $s$ is a complex variable $\delta \pm j\omega$ and is called Laplace operator.

From Fig. 10:
- $X_i(s)$ - is the Laplace transform of the input function (Digital desired position).

- $X_o(s)$ - is the Laplace transform of the output function (Actual position), or system response, etc.

- $S_{PA}(s)$ -is transfer function for Power amplifier block.

- $S_{DCS}(s)$ -is transfer function for DC servomotor block.

- $S_{MT}(s)$ -is transfer function for Machine Table block.

- $S_{INT}(s)$ -is transfer function for Integrator block.

- $H_T(s)$ -is transfer function for Tachogenerator block.

- $S_{CC}(s)$ -is transfer function for Computer or Controller block.

- $H_{SE}(s)$ -is transfer function for Shaft Encoder block.

$$DigitalError = X_i(s) \pm DigitalPositionalFeedback \quad (2)$$

$$DigitalPositionalFeedback = H_{ShaftEncoder}(s) \cdot X_o(s) \quad (3)$$

$$VelocityFeedback = H_{Tachogenerator}(s) \cdot ActualVelocity \quad (4)$$

…..

Minor control loop is:

$$S_1(s) = \frac{S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s)}{1 \pm S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot H_T} \; , so \quad (5)$$

$$S_2(s) = \frac{S_{CC}(s) \cdot S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot S_{INT}(s)}{1 \pm S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot H_T} \quad (6)$$

$$S(s) = \frac{X_o(s)}{X_i(s)} \quad (7)$$

$$S(s) = \frac{\dfrac{S_{CC}(s) \cdot S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot S_{INT}(s)}{1 \pm S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot H_T}}{1 \pm \dfrac{S_{CC}(s) \cdot S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot S_{INT}(s) \cdot S_{SE}(s)}{1 \pm S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot H_T}}$$
…(8)

$$S_A = \frac{S_{CC}(s) \cdot S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot S_{INT}(s)}{1 \pm S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot H_T} \quad (9)$$

$$S_B = \frac{S_{CC}(s) \cdot S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot S_{INT}(s) \cdot S_{SE}(s)}{1 \pm S_{PA}(s) \cdot S_{DCS}(s) \cdot S_{MT}(s) \cdot H_T} \quad (10)$$

Finally $S(s)$ is:

$$S(s) = \frac{S_A}{1 \pm S_B} \quad (11)$$

$S(s)$ -is transfer function for block Diagram of CNC machine-tool control system.

# 4 Using Java for serial communication

The Java programming language has gained rapid usage in the last few years. Inevitably, people ask for being able to program serial interfaces in Java. However, due to the platform-independence of Java, serial programming in Java requires a standardized API with platform-specific implementations

## 4.1 Finding the desired serial Port
The first three things to do when programming serial lines with JavaComm are typically to enumerate all serial ports (port identifiers) available to JavaComm, to select the desired port identifier

from the available ones, and to acquire the port via the port identifier.

Enumerating and selecting the desired port identifier is typically done in one loop [5].

```
import javax.comm.*;

import java.util.*;
...
//

// Platform specific port name, here a Unix name

//

// NOTE: On at least one Unix JavaComm
implementation JavaComm

//      enumerates the ports as "COM1" ... "COMx",
too, and not

//      by their Unix device names "/dev/tty...".

//      Yet another good reason to not hard-code the
wanted

//      port, but instead make it user configurable.

//
String wantedPortName = "/dev/ttya";
//

// Get an enumeration of all ports known to
JavaComm

//
Enumeration portIdentifiers =
CommPortIdentifier.getPortIdentifiers();
//

// Check each port identifier if

//   (a) it indicates a serial (not a parallel) port, and

//   (b) matches the desired name.
```

```
//
CommPortIdentifier portId = null;  // will be set if
port found
while (portIdentifiers.hasMoreElements())

{
   CommPortIdentifier pid = (CommPortIdentifier)
portIdentifiers.nextElement();
   if(pid.getPortType() ==
CommPortIdentifier.PORT_SERIAL &&
     pid.getName().equals(wantedPortName))

   {

      portId = pid;
      break;

   }

}

if(portId == null)

{
   System.err.println("Could not find serial port " +
wantedPortName);
   System.exit(1);
}

//
// Use port identifier for acquiring the port
//
...
```

## 4.2 Parallel communication between computer and output device

When we have parallel communication between computer and output device, or logical circuits, the interface is parallel port. So, the program code to control any interface hardware, output device, is writhed with C++, as you can see in the below example:

```
// Final Project
// Program developed by N.Z and A.L
#include <iostream>
```

```
#include<windows.h>;
using namespace std;
int main()
{
int pom,n, h, x;
unsigned long bi;
int bit[8],a[8];
cout<<endl;
cout<<" Creation of the vector T from vector  A
where T[j] = 2^A[i], i={1,2,3,4,5,6,7)";
cout<<endl;
cout<<"----------------------------------------------------
-------------------------"<<endl;
cout<<endl;
cout<<" recycle = ";
cin>>n;
cout<<endl;
cout<<" Delay time (second) "<<endl;
cout<<" vector recycling = ";
cin>>x;
h=x*1000;
for(int j=0; j<=7; j++)
{
pom = 1;
for (int i=1; i <=j; i++)
{
        pom = pom*2;
}
a[j]=pom;
}
cout<<"----------------------------------------------------
-------------------------"<<endl;
cout<<endl;
cout<<" recycled vector T "<<n<<" multiple ";
cout<<endl;
cout<<"delay time "<<x<<" second ";
cout<<endl;
cout<<" decade form is: "<<endl;
cout<<endl;
for (int k=0; k<n; k++){
for(int j=1; j<=7; j++)
{cout <<a[j]<<"\t"; Sleep(h);
```

```
} cout<<endl;}
cout<<endl;
cout<<endl;
cout<<endl;
cout<<" binary number: "<<endl;
cout<<endl;
for(int m=0; m<n; m++){
for(int j=1; j<=7; j++)
{
        int bi=a[j];

{ if(bi >= 128)
        {
        bi = bi - 128;
        bit[0]=1;
        cout<<bit[0];
}
else{
    bit[0]=0;
        cout<<bit[0];}
        if(bi >= 64)
        {
                bi = bi - 64;
                bit[1]=1;
                cout<<bit[1];
        }
        else {
    bit[1]=0;
        cout<<bit[1];}
 if (bi >= 32)
{
        bi = bi - 32;
        bit[2]=1;
        cout<<bit[2];
}
else {
   bit[2]=0;
        cout<<bit[2];}
if (bi >= 16)
{
        bi = bi - 16;
        bit[3]=1;
```

```
                cout<<bit[3];
        }
        else {
    bit[3]=0;
                cout<<bit[3];}
        if (bi >= 8)
        {
                bi= bi - 8;
                bit[4]=1;
                cout<<bit[4];
        }
        else {
    bit[4]=0;
cout<<bit[4];}
if (bi >= 4)
{
        bi = bi - 4;
        bit[5]=1;
cout<<bit[5];}
else {
    bit[5]=0;
cout<<bit[5];}
if (bi >= 2)
{
        bi = bi - 2;
        bit[6]=1;
        cout<<bit[6];
}
else {
    bit[6]=0;
        cout<<bit[6];}
 if (bi == 1)
{
        bit[7]=1;
        cout<<bit[7];
 }
else {
    bit[7]=0;
cout<<bit[7];
}
}
cout<<" ";Sleep(h);}cout<<endl;}
```

```
cout<<"---------------------------------------------------
-------------------------"<<endl;
cin.get();
cin.get();
return 0;
}
```

### 4.3 Java and technology

The proposed name for this package is "*java. Industrial*". The target platform includes desktop, server, personal, embedded, card. This API is targeted for development of large scale *Java Distributed Control Technology*, and it is designed to be scaleable, robust, and reliable *Industrial Control* solutions utilizing Java technology.

The industrial control and automation solutions will utilize reliable Internet infrastructure as the principle mechanism to enable distributed control technology, where Java is already the number one preferred platform for the internet. [4], [6].

## 5 Conclusion

Industrial automation is defined as the control of an industrial process (manufacturing, production, and so on) by automatic rather than manual and *Control System Engineering* is at the heart and core of this. The purpose of automation is to increase productivity. Productivity is the ratio of physical output to the physical input and it is a reference to productivity, which is real output per hour of work. The industrial control and automation solutions will utilize reliable Internet infrastructure as the principle mechanism to enable distributed control technology, where Java is already the number one preferred platform for the internet, and Java Card is the backup for data, security or remote access, for some industrial automation context.

*References:*
   [1] Pipeline System Automation and Control, By Mike S. Yoon, C. Bruce Warren, and Steve Adam, ISBN #: 0791802639, Published: 2007, Product Type: Print-Book ,No. of pages: 450
   [2] Intelligent Support Systems Technology - by Vijayan Sugumaran - 260 pages,

Published 2002 Idea Group Inc (IGI), ISBN 1931777004.

[3] Batch Control Systems - Design, Application, and Implementation, 2nd Edition, William M. Hawkins and Thomas Fisher, Product ISBN/ID: 978-1-55617-967-9

[4] Human Machine Interface Design for Process Control Applications, Jean-Yves Fiset , Product ISBN/ID: 978-1-934394-35-9

[5] Java I/O, 2nd Edition, Copyright 2006 Elliotte Rusty Harold, O'Reilly, ISBN number is 0-596-52750-0

[6] Java Security - by Harpreet Sethi – 411 pages, Published 2002 Thomson Course Technology, ISBN 1931841853.

[7] Industrial Automation and Process Control (Hardcover), by Jon Stenerson-420 pages, Publisher: Prentice Hall (September 27, 2002), ISBN-10: 0130330302, ISBN-13: 978-0130330307

[8] Design of Feedback Control Systems (Oxford Series in Electrical and Computer Engineering) by Raymond T. Stefani, Bahram Shahian, Clement J. Savant, and Gene H. Hostetter (Hardcover - Aug 30, 2001)- 848 pages, Publisher: Oxford University Press, USA; 4 edition (August 30, 2001), ISBN-10: 0195142497, ISBN-13: 978-0195142495

[9] Sensor Technology Handbook Hardcover, by Jon S. Wilson: 704 pages Publisher: Newnes; Har/Cdr edition (December 8, 2004) Language: English ISBN-10: 0750677295, ISBN-13:978- 0750677295.

[10] Sensors for Measurement and Control, by Peter Elgar, Paperback: 168 pages, Publisher: Prentice Hall (19 Jan 1998) Language English, ISBN-10: 0582357004, ISBN-13: 978-0582357006

[11] 6th WSEAS International Conference on CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL & SIGNAL PROCESSING (CSECS'07)
Cairo, Egypt, December 29-31, 2007

[12] 6th WSEAS International Conference on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS (CIMMACS '07) Tenerife, Canary Islands, Spain, December 14-16, 2007

[13] 5th WSEAS International Conference on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS (CIMMACS '06) Venice, Italy, November 20-22, 2006

[14] 4th WSEAS Conference on: ELECTRONICS, CONTROL & SIGNAL PROCESSING (ICECS '05) Miami, Florida, USA, November 17-19, 2005