OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

TÜBİTAK
MARMARA SCIENTIFIC AND INDUSTRIAL RESEARCH CENTER

DEPARTMENT OF CHEMICAL ENGINEERING
COMPUTER CENTER

SOFTWARE DEVELOPMENT AND HARDWARE MODIFICATION FOR TÜGSAŞ - KÜTAHYA PLANT

PART II FINAL REPORT ON SOFTWARE DEVELOPMENT

December 1990

Reporters : Ersan KALAFATOĞLU, Nuran ÖRS, Tulin GÖZMEN

Project Leader : Assoc. Prof. Dr. Ersan KALAFATOĞLU

Project Team :

Assoc. Prof. Dr. Ersan KALAFATOĞLU
Nuran ÖRS
Tülin GÖZMEN
Yusuf İZGİ

This report has been prepared within the scope of "Software Development and Hardware Modification" project for TÜGSAŞ Kütahya, supported by UNIDO.

TÜBİTAK
MARMARA SCIENTIFIC AND INDUSTRIAL RESEARCH CENTER
DEPARTMENT OF CHEMICAL ENGINEERING
COMPUTER CENTER

SOFTWARE DEVELOPMENT
AND HARDWARE MODIFICATION
FOR TÜGSAŞ - KÜTAHYA PLANT

FINAL REPORT

December 1990

Reporters : Ersan KALAFATOĞLU, Nuran ÖRS, Tülin GÖZMEN

Project Leader : Assoc.Prof.Dr. Ersan KALAFATOĞLU

Project Team :

Assoc.Prof.Dr. Ersan KALAFATOĞLU
Nuran ÖRS
Tülin GÖZMEN
Yusuf İZGİ

This report has been prepared within the scope of "Software Development and Hardware Modification" project for TÜGSAŞ - Kütahya, supported by UNIDO.

TABLE OF CONTENTS

ABSTRACT

In the scope of this study, softvares vhich vill be run on the existing and modified simulator hardvare in TÜGSAŞ-Kütahya plant have been developed.

Vorks performed include the design of general structure of the programs, adding special commands to the Fortran Library, developing specific editor program for process flovsheet displays, and the dynamic simulation of unit operations involved in dilute nitric acid and ammonium nitrate production processes for steady state running and start-up. Necessary subroutines to read/vrite data from/to the devices of the student's console such as indicators, controllers, recorders, etc. have also been developed.

All the prepared softvares have beer tested on the simulator in Kütahya by using a nev PC vhich vas bought for this purpose. Operation Manuals for both processes have also been prepared.

I. INTRODUCTION

The aim of the project was to develop softwares for dilute nitric acid and ammonium nitrate production processes which will be run on the modified simulator hardware in TÜGSAŞ-Kütahya plant.

Simulator will support training of personnel who will take part in the operation of ammonium nitrate and nitric acid plants. Training will cover basic instrumentation and process concepts, operating the plants at steady state and start-up and solution of process problems introduced intentionally.

In the Department of Chemical Engineering of TÜBİTAK - Marmara Research Center (MRC), the development of the softwares for dilute nitric acid and ammonium nitrate plants have been accomplished. To prepare these softwares by utilizing modular programming approach, the rather complex production processes were broken down into individual modules. These modules were then interconnected into a general system. In the general specific main routines for each process these subprograms are called in an appropriate order.

During the development of subprograms to communicate with the student's console MRC - Computer Center and TÜBİTAK - Ankara Electronic Research and Development Institute (TAEGAE) have assisted the MRC - Chemical Engineering Department.

The colored flowsheet displays of both processes have also been accomplished by using a specific editor program and additional commands added to the Fortran Library by MRC-Computer Center.

II. WORK PERFORMED

II.1. WORK PERFORMED BY COMPUTER CENTER OF MRC

In writing simulation programs some additional commands that do not exist in standart Fortran Library were needed especially for colored displays of process units. Therefore the following commands were written by using the Intel 8088 Assembly Programming Language and added to the Fortran Library by the MRC Computer Center.

II.1.1. Commands Added to the Fortran Library

Command : **BEEP**
Syntax : **CALL BEEP**
Action : Generates a sound.

Command : **BLOAD**
Syntax : **CALL BLOAD(string)**
Action : Prints a screen file with file name equal to <string> argument on the screen. The file has to be created by the ED program as described below.

Command : **BSAVE**
Syntax : **CALL BSAVE(string)**
Action : Saves current screen display as a file with name equal to the <string> argument.

Command : **CLS**
Syntax : **CALL CLS**
Action : Clears the screen.

Command : **COLOR**
Syntax : **CALL COLOR(integer1,integer2)**
Action : Changes the current color on the screen as determined by <integer1> and <integer2> variables. <integer1> is the foreground color and <integer2> the background color. If <integer2> is greater than 7 blinking occurs on the screen.

Command : **CURDI**
Syntax : **CALL CURDI**
Action : Disables cursor on the screen.

Command : **CUREN**
Syntax : **CALL CUREN**
Action : Enables cursor on the screen.

Command : **INKEY**
Syntax : **CALL INKEY(string)**
Action : The command reads each key typed and returns the typed character to the <string> argument. If no key is pressed then the string statement is equal to the null character (ASCII 0). The command does not wait for pressing any key.

Remark : If an <integer> variable is used instead of the <string> variable then this statement returns the ASCII number of the typed key.

Command : **INP**
Syntax : **CALL INP(locat,value)**
Action : Returns to the <value> argument within an integer in the range of 0 to 255 which is read from a given I/O port location.
Remark : The <locat> argument is a I/O port number which must be in the range of 0 to 65535.

Command : **KEY**
Syntax : **CALL KEY(integer)**
Action : The command waits for pressing to a key and returns the integer number to the <integer> argument. If the typed key is the function-key on the keypad then the <integer> argument is equal to the key-number which is greater than 255. Normally the <integer> argument is equal to the ASCII number of the typed key.

Command : **LOCATE**
Syntax : **CALL LOCATE(integer1,integer2)**
Action : Sets the cursor position as determined by the integer variables. The <integer1> is the row number (1-25) and <integer2> is the column number (1-80) of the cursor location.

Command : **OUTPO**
Syntax : **CALL OUTPO(locat,integer)**
Action : The OUTP statement is used to write a one-byte (8-bits) binary value into a given I/O port location.
Remark : The <locat> is an integer argument which must equal a value in the range of 0 to 65535. The <integer> argument has a value between 0 and 255.

Command : **REKEYA**
Syntax : **CALL REKEYA(string,length,2,status)**
Action : The command reads each key typed and returns the typed characters to the <string> argument and makes <status> true if the enter key is depressed at the end. <length> is equal to the length of the expected <string> argument. The command does not wait for pressing any key. The number of calls of the command is not limited at any time.

Command : **REKEYI**
Syntax : **CALL REKEYI(integer,length,1,status)**
Action : The command reads each key typed and returns the typed characters to the <integer> argument and makes <status> true if the enter key is depressed at the end. <length> is equal to the number of digits of the expected <integer> argument. The

command does not wait for pressing any key. The number of calls of the command is not limited at any time.

Command : **REKEYR**

Syntax : **CALL REKEYR(real,length,0,status)**

Action : The command reads each key typed and returns the typed characters to the <real> argument and makes <status> true if the enter key is depressed at the end. <length> is equal to the number of digits of the expected <real> argument. The command does not wait for pressing any key. The number of calls of the command is not limited at any time.

Remark : The <real> argument is a single precision float number of REAL*4 format.

Command : **RTIMER**

Syntax : **CALL RTIMER(integer,status)**

Action : Checks the timer whose number is equal to the <integer> argument. If the time elapsed is greater then the endpoint <status> argument is set to true and the timer is restarted.

Command : **STIMER**

Syntax : **CALL STIMER(integer1,integer2,integer3,integer4)**

Action : Starts a timer whose number is equal to the <integer1> argument. <integer2>, <integer3>, and <integer4> arguments are time intervals in H,M,S form which indicate a time endpoint. The number true and the timer is restarted.

Command : **VAL**

Syntax : **CALL VAL(string,real)**

Action : Returns the numeric value representing the data in the <string> argument to the <real> argument. If the first character of the string is not a minus sign (-) or a digit the value returned is zero. String conversion is finished when the end of the string or any non-digit character is found (except decimal point).

Remark : The <real> argument must be of REAL*4 format.

The string variables in the following commands must be of CHARACTER*80 format.

Command : **CHAIN**

Syntax : **CALL CHAIN(string1,string2)**

Action : This command appends the content of <string2> argument to the end of <string1>.

Command : **LEFT**

Syntax : **CALL LEFT(string1,string2,integer)**

Action : The command returns a sub-string taken from the leftmost end of the <string2> argument to the

<stringl>. The lenght of the sub-string is defined by the <integer> argument which can be in the range of 0 to 80. If the value of the <integer> argument is zero then the null-string is returned.

Command : **LEN**
Syntax : **CALL LEN(string,integer)**
Action : This command returns the number of characters in the <string> argument to the integer variable.

Command : **MID**
Syntax : **CALL MID(stringl,string2,integerl,integer2)**
Action : The MID function returns a sub-string which is taken from a larger <string2> argument to the <stringl>. The starting position of the sub-string is defined by <integerl> and the length of the sub-string by <integer2>.
Remark : The <integerl> and <integer2> arguments can have values ranging from 0 to 80.

Command : **PRSTR**
Syntax : **CALL PRSTR(string)**
Action : Writes the content of the <string> argument on the screen.

Command : **RIGHT**
Syntax : **CALL RIGHT(stringl,string2,integer)**
Action : The command returns a sub-string taken from the rightmost end of the <string2> argument to the <stringl>. The lenght of the sub-string is defined by the <integer> argument which can be in the range of 0 to 80. If the value of the <integer> argument is zero then the null-string is returned.

II.1.2. Specific Editor Program For Process Flowsheet Displays

To display flowsheets, process units, and various other information on the process, a special editor program (ED) was necessary. This program was written using the Intel 8088 Assembly Programming Language.

The ED offers features for the users to correct drawing errors easily and to make large revisions without redrawing entirely. It produces clean, precise final drawings with the ASCII character set and also saves the colored screen form. The ED allows to use a work-sheet with dimensions 23x80 rowxcolumn, respectively. The 1st and 25th rows of the screen are reserved for the ED. The first row indicates the foreground-color, background-color, current cursor location, available graphic-characters and two status-locations. The 25th row indicates functions of the keys which can be used in the program.

The functions of the keys used by the ED are as follows :

KEY DESCRIPTION

F1 Terminates the ED. If you press the F1 key then the following message appears on the 25th row :

ARE YOU SURE Y/N

The ED returns to the DOS if your answer is Y, if not it returns to the current work-sheet.

F2 Shows alternative help line on the 25th row.

F3 Loads a file from the disk. If F3 key is pressed following message appears on the 25th row :

ENTER FILE NAME LOAD

If the file name is equal to the null-character, ED returns back to the current work-sheet. Otherwise it loads a file from the disk according to the given file name. The wait prompt on the first line blinks during loading.

F4 Saves the work-sheet on the disk. The following message appears on the 25th row after when F4 is entered :

ENTER FILE NAME SAVE

The ED returns to the work-sheet if the file name is equal to the null-character. Otherwise the current work-sheet is saved under the given file name. The wait prompt blinks during saving.

F5 Increments the ASCII number and displays the corresponding character which appears on the first line.

F6 Decrements the ASCII number and displays the corresponding character which appears on the first line.

F7 ON/OFF repeat-mode. If the repeat_mode is ON then the last character typed by the user is repeated on the screen by holding down the direction keys according to the key's direction.

F8 Clears the work-sheet. If F8 key is pressed then the work-sheet is filled by the space characters. Foreground and background colors of the space characters is determined by the current attributes.

F9 Increases the foreground color number.

F10 Increases the background color number.

ESC Copies the character which appears on the first line to the current location of the cursor on the work-sheet.

Page-up Moves the cursor to the top line.

Page-down Moves the cursor to the bottom line.

Home Moves the cursor to the beginning of the line.

End Moves the cursor to the end of the line.

II.2. WORK PERFORMED BY THE CHEMICAL ENGINEERING DEPARTMENT OF MRC

Simulation may be defined as the use of a mathematical model to generate a description of the state of a system. Simulation provides a good insight into the behaviour of the actual system. This insight provides a convenient, inexpensive, and safe method of gaining the understanding of a real process without actually experimenting on an operating process plant. In order to achieve this goal, the simulation must reflect the actual process behaviour and must be fast enough to be run on a PC.

In this work, a modular design concept has been used, where complex process configurations are broken down into individual modules and through a linking mechanism these individual modules are interconnected into a total system. Steady state simulation of chemical processes chosen involves combining the building blocks of modules consisting of various unit operations and processes with the various streams. Where the process involves recycle streams the problem of solving mass and energy balances is an iterative procedure. Since the steady state alone is not always sufficient to understand the process behaviour the dynamic analysis which covers the system responses to the changes in operating conditions has also been included.

The bases for mathematical models are the fundemental physical and chemical laws, such as the laws of conservation of mass, energy, and momentum. In the study of the dynamics these are used in their general form with time derivatives included. Obviously an extremely rigorous model that includes every phenomenon down to microscopic detail would be so complex that it would take a long time to develop and might be impossible to solve. Thus, as many simplifying assumptions as are reasonable have been made. It has been made sure that the number of variables equals the number of equations namely the degree of freedom of the system was zero.

The validity of the models have been also checked by using real plant and literature data.

II.2.1. Program Language and Construction

All simulation programs were written in FORTRAN 77. The FORTRAN compiler used was of Microsoft version 5.0

In writing programs, the subroutines dedicated to a particular computing task have been used. The initialization, data input and output, calculations, and various controls have been kept separately in different subroutines. Each module which simulates a particular unit operation is called as a subroutine from a main process subroutine. All of the special task subroutines such as display of an unit operation, changing into change mode, etc. and process subroutine are interconnected by a main routine.

Once the data initialization and steady state solutions were completed, the dynamic behaviour of the system is calculated by performing a number of numerical integrations of differential equations derived for the system under consideration.

Experience with a wide range of typical chemical engineering problems has shown that the fourth-order Runge-Kutta method is quite adequate for any situation and that in many cases the second- and first- order methods will provide reosanable accuracy coupled with greater efficiency [1-3]. In this work second-order method has been used for ammonium nitrate process, but since the computation slows down with the increase of integration order only first-order method could be used for nitric acid process. Their accuracies were seen to be acceptable by comparing with the actual plant data.

At first actual plant flowsheets of dilute nitric acid and ammonium nitrate production processes have been simplified as shown in Figures 2 and 3. Each stream and unit were numbered and these numbers were used in building the stream and unit matrices. As mentioned in the preliminary reports, stream matrix S(I,J) involves the following properties (I being the stream number) [4,5]:

J = 1	Mass flow rate	[kg/hour]
J = 2	Pressure	[Pascal]
J = 3	Temperature	[°K]
J = 4	Unit enthalpy	[kJ/kg]
J = 5	Density	[kg/m3]
J = 6	Molecular weight	[kg/mole(kg)]
J = 7	Volumetric flow rate	[m3/hour]
J = 8	Volumetric flow rate	[Nm3/hour]

J = 9	Phase of stream (vapor = 1; liquid = 2)	
J = 10	Free (used for various purposes)	
J = 11	Weight ratio of component 1	[-]
J = 12	Weight ratio of component 2	[-]
J = 13	Weight ratio of component 3	[-]
J = 14	Weight ratio of component 4	[-]
J = 15	Weight ratio of component 5	[-]
J = 16	Weight ratio of component 6	[-]
J = 17	Weight ratio of component 7	[-]
J = 18	Weight ratio of component 8	[-]
J = 19	Free (used for various purposes)	
J = 20	Free (used for various purposes)	
J = 21	Volume ratio of component 1	[-]
J = 22	Volume ratio of component 2	[-]
J = 23	Volume ratio of component 3	[-]
J = 24	Volume ratio of component 4	[-]
J = 25	Volume ratio of component 5	[-]
J = 26	Volume ratio of component 6	[-]
J = 27	Volume ratio of component 7	[-]
J = 28	Volume ratio of component 8	[-]
J = 29	Free (used for various purposes)	
J = 30	Stream number	

Because the ammonium nitrate process is basically a process in solution the volume ratios of the components (J = 21-28) are not computed.

On the other hand unit matrix UN(I,J) includes basic unit characteristics such as heat transfer area, catalyst activity, numbers of the entering and outgoing streams of the unit, liquid level in the unit, holdup of the unit, etc. Mathematical models of each unit were established by using fundamental physical and chemical laws and respective time derivatives were included to describe the dynamic behavior of the unit. Since the real system is too complex for a reasonable and feasible simulation, simplification of the actual system and necessary assumptions have been made by consulting the plant engineers of TÜGSAŞ-Kütahya plant. Validity of the models developed have been checked by using actual plant and literature data [6-17].

Although the controllers on the student's console will control the simulation parameters it was requested by TÜGSAŞ that the controllers are also simulated so that in case of malfunction of one of the controllers on the student's console a simulated controller will start operating and the whole system will not break down.

There are special task subroutines which are called from any unit subroutines. These can be grouped as follows:

A. Integration subroutines

SIFIR	initializes the integration
INTI	Integrates the independent variable (time)
INT	Integrates the dependent variables

B. Mathematical subroutines

YAKIN	Converges by partial substitution
INPOL1	Interpolates one dimensional data
INPOL2	Interpolates two dimensional data
POL3	Calculates roots of a third order polynom

C. Physical property subroutines

HTEXCH	Adds enthalpy to a stream
ENTHV	Calculates enthalpy of a gaseous stream
ENTHVK	Calculates enthalpy of absorption gases
ENTHL	Calculates enthalpy of a liquid stream
SICAK	Calculates the temperature of a stream
PRES	Calculates pressure drop in heat exchanger
OD	Calculates various properties of a stream

D. Heat exchanger subroutines

THEG	Countercurrent heat exchanger (gas-gas)
PMHEGS	Pressurized heat exchanger (gas-liquid)
PHEG	Parallel heat exchanger (gas-gas)
MHEGS	Countercurrent heat exchanger (gas-liquid)
PTHEG	Pressurized heat exchanger (gas-gas)
TDHES	Countercurrent heat exchanger (liq.-liq.)
TCSHES	Heat exchanger (steady state)
PTCHES	Pressurized heat exchanger (steady state)
EJ	Ejector
DEPO	Simulates a tank with heat exchanger

E. Stream operation subroutines

SUM	Calculates the sum of two streams
SPLIT	Splits a stream in a given ratio
SUBTR	Calculates the difference of two streams
HLDP	Calculates changes in the holdup
HLDPK	Calculates changes in the column holdup
AGHA	Converts weight ratio to volume ratio
HAAG	Converts volumetric ratio to weight ratio
REACT	Calculates changes due to a reaction

PUMP	Calculates a stream leaving a pump
DEBI	Calculates the flowrate of compressor air
MA	Calculates the molecular weight

F. Controller subroutines

KONTROL	Simulates a general controller
CONTR2	Simulates a PID controller
TFN2	Solves transfer functions of controllers

G. Data input/output subroutines

VERI	Reads original plant data from disk
OKU	Reads data from ports
YAZ	Writes data to ports
PRNTF	Writes selected process data to hard disk
FILTRE	Filters consol noise

H. Screen display subroutines

PRSCR	Displays plant data of controllers
DSCR	Displays plant data of equipments
DEGIS	Makes changes for a particular equipment or a stream

II.2.2. Program Modes

- RUN MODE

When the main program is started a welcome menu appears on the screen. Plant time (not the actual time) which is used in the calculations is displayed on the upper right hand corner in HH:MM:SS format. On the upper left hand corner **KOMUT** denoted place expects an input of the selected equipments designation. The other options, **X**, **D**, or **B** can also be entered at the same location. Here **X** exits to the DOS environment, **D** calls the routine **DEGIS** where changes can be made in process variables (CHANGE MODE), and **B** returns to the original steady state process data and sets time to 00:00:00. When the equipment designation in capital letters is typed into the defined location and the enter key is depressed than the particular equipment and its data will be displayed on the screen. Screen displays and communications are in Turkish language as requested by TÜGSAŞ.

- CHANGE MODE

To make modifications in the process variables **D** is entered by typing **D** at the upper left corner and pressing the enter key. The location of the data which can be changed will start blinking. The new data will be entered again at the upper left hand location. When the enter key is pressed the new data will appear in the diagram in a different color and the next data change location if present will start blinking. This will continue until the last change has been made. When the last data change is entered the system asks whether the changes made are O.K. or not. If the changes are not proper, **0** is entered so that new changes can be made. If 1 is entered CHANGE MODE is exited and simulation starts to use the new data and during the above procedures simulation continues to run with the old data.

II.3. SIMULATION OF THE PROCESSES

The directory tree of the computer hard disk for the simulation programs is shown in Figure 1.

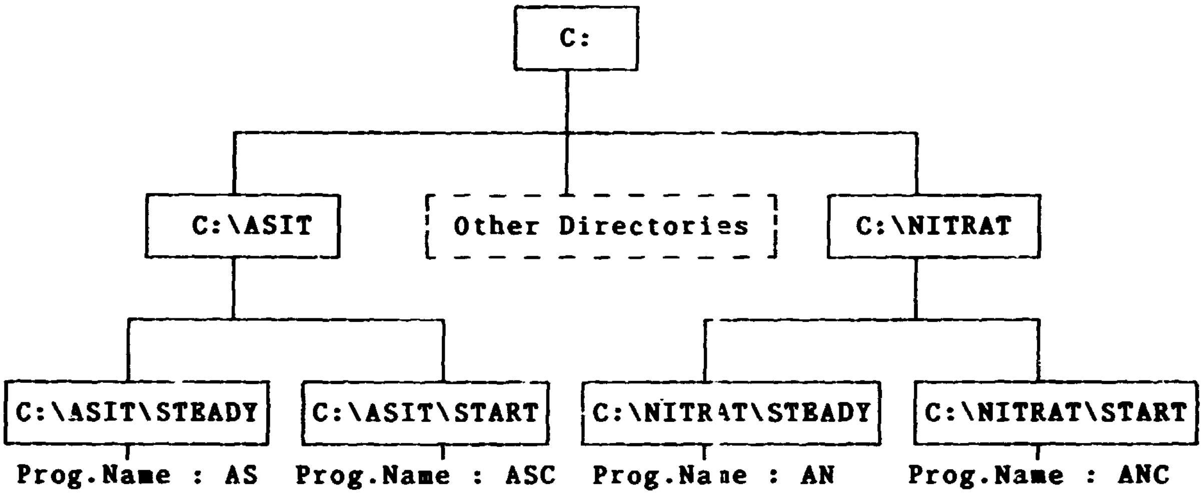

Figure 1. Directory tree of the computer for the simulation programs

The startup procedures for each process is simulated separately, and student's console initial settings are quite different from the steady state settings. Because extra valves and pumps are required for the startup operation and because there is a limited number of valves and pumps on the console, different assignments had to be used for the steady state and startup operations .

Both process simulations are supplied with separate manuals which contain necessary information on the initial settings of the console, on the chemical process, and on the operation of the programs.

II.3.1. Nitric Acid Production Process

Main routine **AS** calls **ASIT** to perform calculations. If a particular equipment or controller is selected **DSCR** or **PRSCR** is called, respectively, to display a generalized picture of the item and the related process data (RUN MODE). If it is intended to change process or unit variables **DEGIS** is called (CHANGE MODE). While changes are being made simulation is continuing with the old data until all the changed data are entered into the simulation.

ASIT first writes and reads data from and to the ports to communicate with the student's console, then calls the 22 units in an appropriate order. For all the units with the exception of a few heat exchangers special subroutines have been developed. The displays of these units are listed below :

EA3001, GB3001, EA3005, EA3101, FP3001, FK3002, BH3001, BB3101, FB3102, EA3201, EA3203, EA3205, EA3207, EA3301, DK3301TA, DK3301TE, DK3303TA, DK3303TE, DB3301, EA3311, EA3314, FB3301.

Also each of the 10 controllers are displayed when any of the following commands are entered :

LIC3003, PIC3009, FRC3009, LIC3107, LIC3301, LIC3303, LIC3306, LIC3308, FIC3301, PIC3106.

Besides of the units' and controllers' displays following general purpose displays help the user to make changes of the process parameters or follow the process or the student's console :

MENU, KOMUT1, KOMUT2, A1, A2, SU, KONSOL, ALARM, ARIZA, DEGER, UYARI, DURMA, YAZICI1, YAZICI2, LISTE.

The selected process units and controllers which were simulated for nitric acid production process are listed in Tables 1 and 2 respectively.

Table 1. Selected unit operations for nitric acid process

Unit No.	Designation	Name of the equipment
I	EA3001	Ammonia evaporator
II	GB3001	Air compressor
III	EA3005	Ammonia preheater
IV	EA3101	Boiler water preheater
V	FP3001	Ammonia - air mixer
VI	FF3002	Ammonia - air mixture filter
VII	BH3001	La-Mont reactor
VIII	BB3101	La-Mont superheater
IX	BB3101	La-Mont evaporator
X	BB3101	La-Mont preheater
XI	FB3102	Steam drum
XII	EA3201	Tail gas preheater
XIII	EA3203	Boiler water preheater I
XIV	EA3205	Condenser I
XV	EA3207	Condenser II
XVI	EA3301	Acid coolers
XVII	DK3301	Oxidation/absorption column
XVIII	DK3303	Absorption column
XIX	FB3301	Feed water tank
XX	EA3311	Feed water cooler I
XXI	EA3314	Feed water cooler II
XXII	DB3301	Degasser

Table 2. Selected controllers for nitric acid production

Cont. No.	Designation	Function
I	LIC3003	Ammonia evaporator level control
II	PIC3009	Ammonia evaporator pressure control
III	FrC3009	Ammonia/air ratio control
IV	LIC3107	Steam drum level control
V	LIC3301	Oxid./absorp. column level control
VI	LIC3303	Absorption column level control
VII	LIC3306	Degasser level control
VIII	LIC3308	Feed water tank level control
IX	PIC3106	Superheated steam pressure control
X	FIC3301	Feed water flow rate control

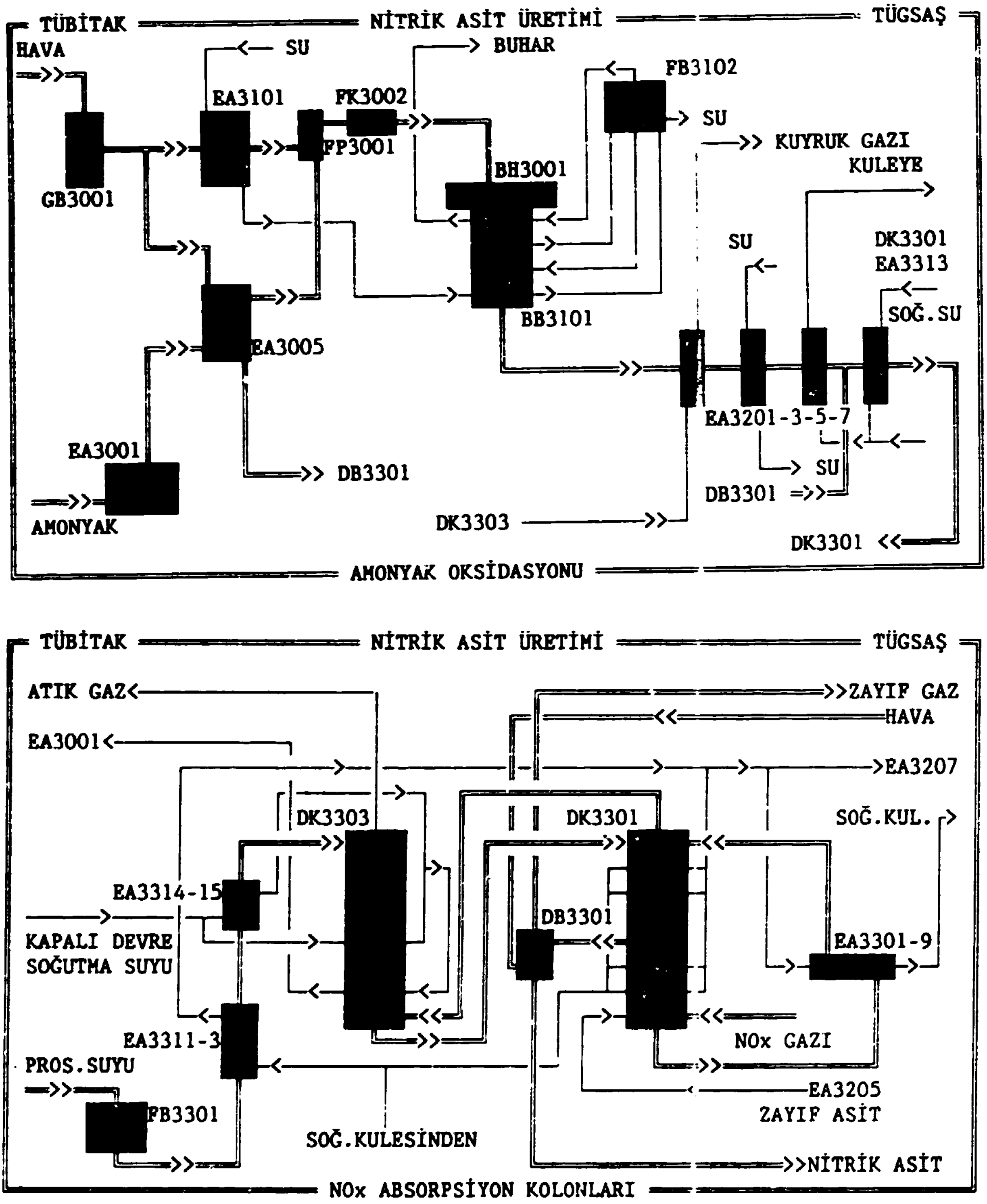

Figure 2. Simplified flowsheet of nitric acid process

II.3.2. Ammonium Nitrate Production Process

Main routine AL calls NIT to perform calculations. If a particular equipment or controller is selected DSCR or PRSCR is called, respectively, to display a generalized picture of the item and the related process data (RUN MODE). If it is intended to change process or unit variables DEGIS is called (CHANGE MODE) as in the other process. While changes are being made simulation is continuing with the old data until all the changed data are entered into the simulation.

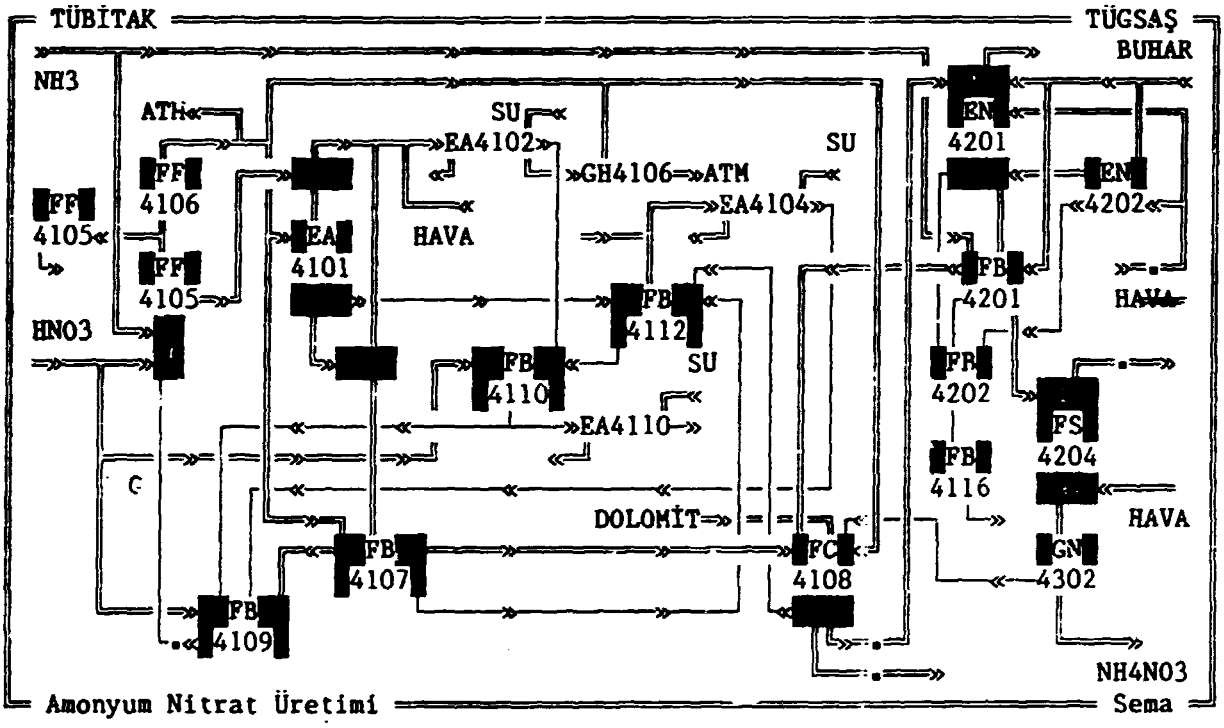

Figure 3. Simplified flowsheet of ammonium nitrate process

NIT first writes and reads data from and to the ports to communicate with the student's console, then calls the 21 units in an appropriate order. For all the units with the exception of a few tanks special subroutines have been developed. The displays of these units are listed below :

FF4105, FF4106, EA4103, EA4101, EA4102, GH4106, FB4110, FB4107, FB4109, FB4112, EA4104, FC4108, EN4201, FB4201, EN4202,FB4202, FB4116, FS4204, GN4302

Also each of the 10 controllers are displayed when any of the following commands are entered :

LRC4151, FRQC4152, PHRC4151, TRC4151, PRC4151, PRC4155, PRC4154, FRQC4201, TRC4201, LRC4155

Besides of the units' and controllers' displays following general purpose displays help the user to make changes of the process parameters or follow the process or the student's console :

MENU, KOMUT1, KOMUT2, SEMA, KONSOL, ALARM, ARIZA, DEGER, UYARI, DURMA, YAZICI1, YAZICI2, LISTE

The selected process units and controllers which were simulated for ammonium nitrate production process are listed in Tables 3 and 4 respectively.

Table 3. Selected unit operations for ammonium nitrate process

Unit No.	Designation	Name of the equipment
I	**FF4105**	Neutralizer
II	**FF4106**	Steam Droplet Trap
III	**EA4103**	Basic Steam Condenser
IV		Flash Drum
V	**EA4101**	Primer Evaporator
VI		Separator
VII	**EA4102**	Steam Condenser
VIII	**GH4106**	Ejector
IX	**FB4110**	Condensate Tank
X	**FB4107**	Barometric Solution Tank
XI	**FB4109**	Cold Solution Tank
XII	**FB4112**	Basic Steam Condens. Tank
XIV	**EA4104**	Basic Steam Condenser
XV	**FC4108**	Homogenizer
XVI	**EN4201**	Luwa-Kaltenbach Evaporator
XVII	**FB4201**	Equalizer Vessel
XVIII	**EN4202**	Air Heater
XIX	**FB4202**	Condensate Tank
XX	**FB4116**	Condensate Tank
XXI	**FS4204**	Prilling Tower
XXII	**GN4302**	Screen

Table 4. Selected controllers for ammonium nitrate production

Cont. No.	Designation	Function
I	**LRC4151**	Level control of neutralizer
II	**FRQC4152**	Acid flow control of neutralizer
III	**pHRC4151**	pH control of neutralizer
IV	**TRC4151**	Temperature control of neutralizer
V	**PRC4151**	Pressure control of neutralizer
VI	**PRC4155**	Pressure control of primer evaporator
VII	**PRC4154**	Pressure control of primer evaporator
VIII	**TRC4201**	Temperature control of Luwa
IX	**PIC4201**	Pressure control of Luwa
X	**LRC4155**	Level control of homogenizer

III. CONCLUSION AND RECOMMENDATIONS

The dynamic simulation of nitric acid and ammonium nitrate production processes are completed. The softwares are being run on the present student's console in Kütahya by TÜGSAŞ personnel. A period of three months will be allowed for the TÜGSAŞ personnel to request some last modifications which will be incorporated into the softwares.

By accomplishing this project the Department of Chemical Engineering of MRC has gained a good experience on simulation of chemical processes. This accumulation of knowledge can be increased in future and result in similar projects. This is very important for improving Turkish software technology. Therefore with a possible financial assistance of UNIDO and related companies to accomplish other similar projects will be of great help to the Turkish industry.

REFERENCES

[1] FRANKS, R.G.E.
Modelling and Simulation in Chemical Engineering
Wiley-Interscience, 1972

[2] LUYBEN, W.L.
Process Modelling, Simulation, and Control for Chemical Engineers
McGraw-Hill, 1973

[3] RAMAN, R.
Chemical Process Computations
Elsevier Appl. Sci. Publ., 1985

[4] SOFTWARE DEVELOPMENT AND HARDWARE MODIFICATION FOR KÜTAHYA-TÜGSAŞ
Preliminary Report
TÜBİTAK-MRC, 1989

[5] SOFTWARE DEVELOPMENT AND HARDWARE MODIFICATION FOR KÜTAHYA-TÜGSAŞ
Second Term Report
TÜBİTAK-MRC, 1990

[6] SHERWOOD, T.K. - PIGFORD, R.L.
Absorption and Extraction
McGraw-Hill, 1952

[7] SHERWOOD, T.K. - PIGFORD, R.L. - WILKE, C.R.
Mass Transfer
McGraw-Hill, 1975

[8] MATASA, C. - TONCA, E.
Basic Nitrogen Oxides
Chem. Publ. Co. Inc., 1973

[9] GMELINS HANDBUCH DER ANORGANISCHEN CHEMIE
Stickstoff, Lieferung 2
Verlag Chemie GmbH, 1935

[10] GMELINS HANDBUCH DER ANORGANISCHEN CHEMIE
Stickstoff, Lieferung 4
Verlag Chemie GmbH, 1936

[11] GMELINS HANDBUCH DER ANORGANISCHEN CHEMIE
Ammonium, Lieferung 1
Verlag Chemie GmbH, 1936

[12] ULLMANN'S ENCYCLOPEDIA OF INDUSTRIAL CHEMISTRY
Volume A2, pp. 243-248
Verlagsgesellscaft mbH, 1985

[13] ENCYCLOPEDIA OF CHEMICAL TECHNOLOGY
Vol.2, pp. 470-536
John-Wiley and Sons, 1978

[14] ENCYCLOPEDIA OF CHEMICAL TECHNOLOGY
Vol.15, pp. 853-870
John-Wiley and Sons, 1981

[15] CHEMICAL ENGINEERS' HANDBOOK
5. edition
McGraw-Hill, 1973

[16] SCHMIDT, E.
Properties of Water and Steam in SI Units
3. edition
Springer Verlag, 1982

[17] VDI - WÄRMEATLAS
VDI-Verlag GmbH, 1984