INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4.SMR. 403/25

FIFTH COLLEGE ON MICROPROCESSORS: TECHNOLOGY AND APPLICATIONS IN PHYSICS

2 - 27 October 1989

Introduction to Projects

G. BARREIRA
Laboratory of Instrumentation and Particles, Lisbon, Portugal

SQUARE WAVE GENERATOR	- BANDARA
REACTION TIMER	- KARNAD
CLOCK FUNCTIONS	- IMTIAZ
DIGITAL VOLTMETER	- MARIO TRUJILLO

LOGIDULES

DIGITAL OSCILLOSCOPE	- MANUEL GONÇALVES
RC METER	- MANUEL GONÇALVES
DATA ACQUISITION SYSTEM	- XIONG

SOFTWARE

PAL PROGRAMMING	- GAO
ROSY TO ROSY COMMUN.	- WU
FILE HANDLING	- JOHN TRUJILLO

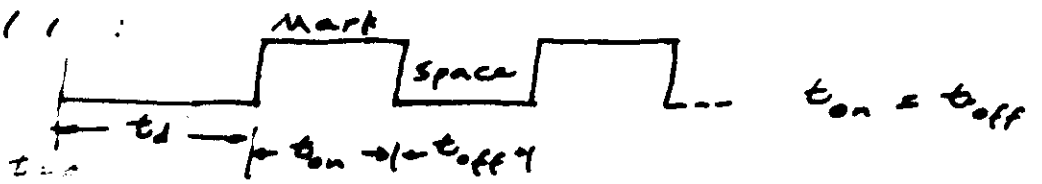
RECOVERING FROM THE EXERCISES

PROJECT : SQUARE WAVE GENERATOR

OBJECT : To use the ICTP board to generate a square wave signal

PROCEDURE : The project may be carried out at 3 Levels.

i. Level 1 :



Use the ICTP board with the ROSY station to generate a square wave signal as above. The half period (t_{on}) is to be entered from the keyboard. The signal starts after an initial delay t_d, after a push button is pressed. Timing is to be derived from the board's 100 Hz clock. The signal may be output using one (or more) lines of the PIA B-side.

ii. Level 2 :

Same as above but the signal will have different mark : space ratios. The periods t_{on} and t_{off} should be entered using push buttons on the ICTP board, and displayed on the board's display.

iii Level 3 : Wave form in i or ii above to be chosen initially using a switch. Programming to be done in a position independant manner, so that the program may be loaded into the boards ROM, enabling the board to run program in stand alone mode, using the kernel software.

Provide a 'written' report clearly defining the objective and how the program is to be used. An assembly listing of program should be provided.

Project Title: REACTION TIMER

Project Definition:

This project attempts to measure the reaction time of an individual in the following manner:

* The user presses a **START** button (time $t = \emptyset$)
* After a random time delay from this instant, an audible/visible **PROMPT** is given to the user (time $t = t_1$)
* As soon as the prompt is received, the user is expected to press the **RESPONSE** button. (time $t = t_2$)
* The duration of time between **PROMPT** and **RESPONSE** is termed as **REACTION TIME** and is displayed. ($t_r = t_2 - t_1$)

Project Methodology:

1. Precise definition of the problem with State Diagrams and state description.
2. Flow charts and coding (using structured constructs) in three phases:
   - 2.1 1st Phase: Random delay is from a look-up-table.
   - 2.2 2nd Phase: Include a random number generator.
   - 2.3 3rd Phase: Convert the programs into a position independent one - suitable for PROMing.
3. Document a SOLUTION to this project.

Project Schedule:

	MON	TUE	WED	THU.	FRI
Prob. definition, State Diagrams	*				
Flow charts	*	•			
1st Phase coding & testing.	*	*	•	•	•
2nd Phase coding & testing.		*	*	•	•
3rd Phase coding & testing.			*	*	•
Documenting the SOLUTION					*

TITLE : TO IMPLEMENT CLOCK FUNCTION ON ICTP BOARD.

BASIC FUNCTIONS:

- TIME DISPLAY

  HOURS AND MINUTES IN 12 HR. OR 24 HOUR FORMAT.

- TIME - SET

ALARM FUNCTION

- ALARM TIME DISPLAY

  HOURS AND MINUTES IN 12 HR. OR 24 HR. FORMAT.

- ALARM TIME SET

- ALARM ON/OFF

CLOCK FUNCTIONS

ADDITIONAL FUNCTIONS:

- DISPLAY OF SECONDS
- DISPLAY OF DATE
- MORE THAN ONE ALARM
- STOP WATCH
- TIMER

THE ABOVE FUNCTIONS ARE TO BE IMPLEMENTED USING THE DISPLAY AND TWO TOGGLE SWITCHES AND TWO PUSH BUTTONS AND HEXA DECIMAL SWITCH ON ICTP BOARD.

VOLTMETERS

USING THE V/F CONVERTER ON THE COLOMBO BOARD

LEVEL 1

DISPLAY READING FROM VFC USING THE KERNEL (UPGRADE OF EX 14)

LEVEL 2

SELF CALIBRATING VOLTMETER

- CALIBRATION MODE
  - CONNECT REFERENCE VOLTAGE
  - PUSH READY BUTTON
- USE MODE

LEVEL 3

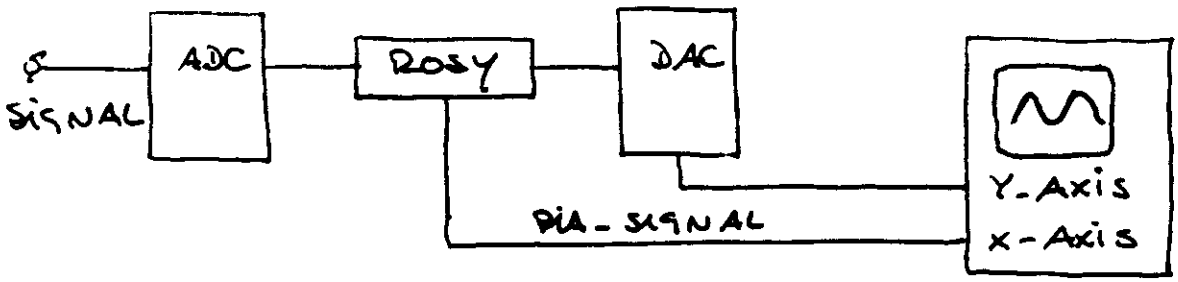
READ AND STORE A slowly VARYING VOLTAGE OVER A PERIOD OF TIME.

THE INTERVAL BETWEEN READINGS SHOULD BE SET WITH SWITCHES OR PUSHBUTTONS AT BEGINING.

IT SHOULD ALLOW STOPPING TO LOOK AT THE STORED VALUES (AND ALSO STOP WHEN IT USES UP ALL THE AVAILABLE SPACE FOR STORING READINGS.)
IT SHOULD DISPLAY RECORDED VALUES AND STEP NUMBER

AIM of THIS PROJECT:

DATA_ACQUISITION



SCOPE X-Y MODE

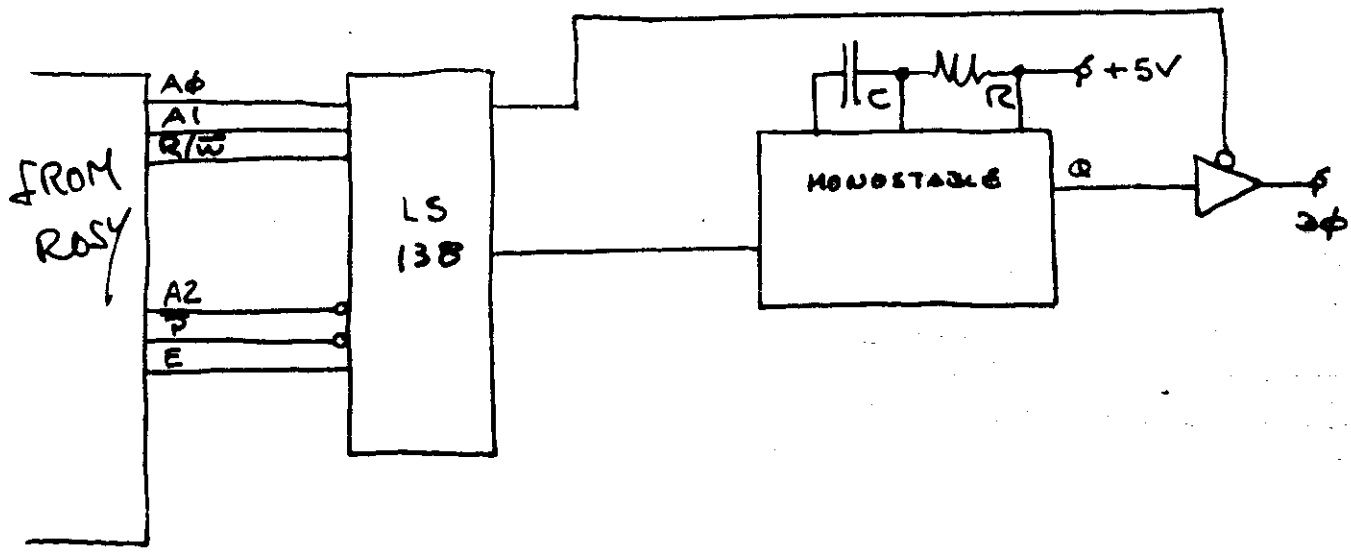
1- USE A BUFFER CONTAINNING THE Y VALUES OF A SINUSOIDAL FUNCTION.
USE A DAC TO DRIVE THE Y AXIS.
USE THE PIA TO GENERATE A TRIGGER_PULSE.

2- USE 2 DAC'S TO DRIVE BOTH X AND Y AXIS, WITH SINUSOIDAL FUNCTIONS TO GENERATE LISSAJOUS FIGURES.

3- USE AN ADC TO SAMPLE A SIGNAL, STORE SAMPLED VALUES IN A BUFFER.
UNDER USER REQUEST START THE DISPLAY OF THE BUFFER.

AIMS OF THIS PROJECT:

1- TO CONSTRUCT A CAPACITANCE (OR RESISTANCE) MEASURING SETUP USING LOGIC GATES.
2- PRESENT THE AVERAGE OF 16 DIFFERENT MEASUREMENTS OF THE SAMPLE ON THE SCREEN.



Data Acquisition System

1. Implement a Successive Approximation ADC by using logic ICs and associated software
2. Make use of a potentiometer to emulate some kind of sensors.
3. Collect data into data buffer during a definite period.
   Sampling rate and acquisition period can be entered from keyboard.
4. Process the data according to participants' own ideas, e.g. displaying max. & min. value, data conversion, plotting, graphing, setting alarm, etc.

Goals: 1. deepen the understanding of PAL;

2. get practical experience on PAL's design and programming.

Facilities:

1. An UNIVERSAL programmer controlled by IBM-PC/XT, it can be used to

   Load fuse map from disk,
   Save fuse map to disk,
   Edit fuse map,
   Blank check,
   Programming,
   Verify,
   Security fuse blow,
   etc.

2. A software package, in which

   CUPL — for writing & compiling logic descriptions.
   CSIM — for simulating your design.

3. Some PAL chips & breadboards for testing.

Procedures:

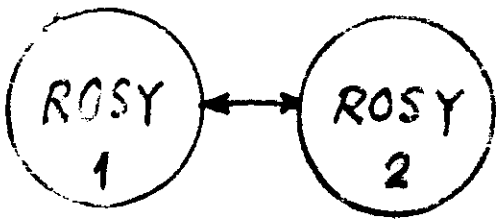
1. Learn how to use the programmer and CUPL / CSIM.
2. Have your own design.
3. Do programming and simulating.
4. Do testing.

Limitations:

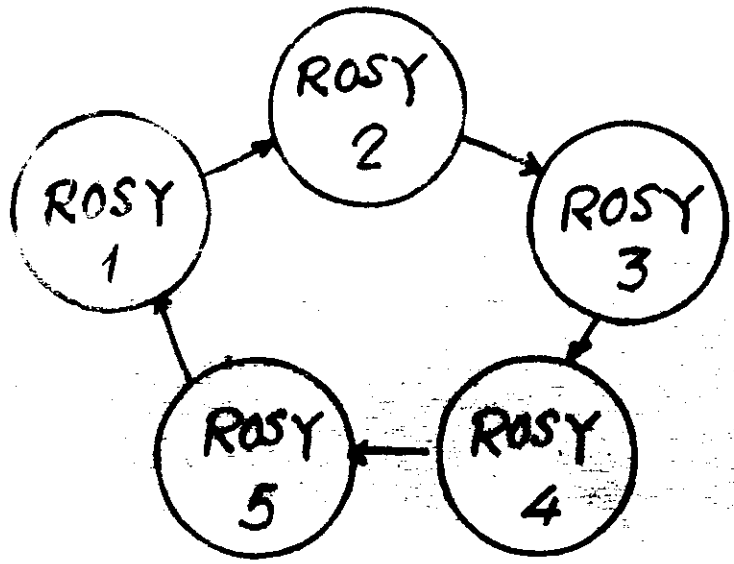
1. Only the PALs we have or can have can be used in your design.
2. Design should be simple enough to be tested easily.

PROJECT:

ROSY TO ROSY Communication using RS-232 Link



BETWEEN TWO ROSYs



IN A RING

Requirements:

1. Try two stations
2. Host port of Rosy (secondary output device) is used.
3. Message are packets with sync. flag and destination address as header and EOT as end of message.
4. Reception on interrupt (done by Rosy) and transmission on polling.

Level 2: Transmission:

Sync. Flag	Source Address	Destination addr.	USER data	Terminator

- Message typed from keyboard with destination address in line buffer is transmitted use. MON call to input messages.
- Handle only one message at a time.
- Use flags KB-buffer full, transmit ready and transmit over etc.

Reception:

- Identify the sync. flag and start receiving in receiver buffer.
- Identify the destination address and display the message on the screen if the message is addressed to the station itself.
- Use flags like Receiver-buffer full, display-buffer full and display over etc.

Level 3.

sync. flag.	Source addres.	Destination address	Message type	sequence number	Message length	User data	check SUM

- Extend the header with message type, message length and sequence number.
- Include checksum for error detection, use ROSY routines.
- Send acknowledgement (ACK) for packet received correctly.
- Send negative-acknowledgement (NACK) for packet received incorrectly.

PROJECT

Title: File handling With FLEX

Specification: Operating System.

Description: Implement new FLEX commands

Difficulty: 3 out of 3

Remarks: For those Who choose this project, an introductory lecture on file handling in FLEX Will be arranged.

All information necessary for this project is contained in the FLEX manual.

1. — Study the LIST command given as an example in the Flex Manual and implement some extension to the LIST command such as:
   - listing of specified range of lines
   - title and numbering of pages
   - etc

2. — Implement an extension of the DELETE command or create an ERASE command which accept 'DUMMY' characters, so that one may delete with a single command all files with the same name and different extensions or all files with different names and the same extension.

```
ERASE , PROG.*.1

ERASE , *.BAK.1
```

A more general form would be: ERASE EX*.SYM.1

which would erase all files with names starting with

EX... and extension .SYM on disk drive 1

The ERASE *.*.1 erases all the files on the disk

[...] Implement an UNDELETE Command to recover a file which was just deleted with the FLEX DELETE command.