University of Mosul
College of Engineering

Computer Engineering Laboratories

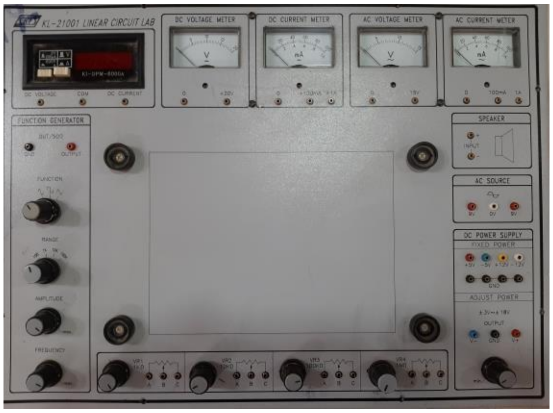
Electrical circuits' laboratory

The Electrical Laboratory is one of the laboratories of the Computer Engineering Department. Laboratory experiments are given to first-year students in order to enable them to become familiar with laboratory tools and equipment through which students will be able to apply and conduct experiments related to electrical circuits and their two types, direct current (DC) circuits and alternating current (AC) circuits. The main purpose of working in the laboratory is to achieve theories in a practical way in order to help the student understand those theories while adding technical and practical experience to him so that he can maintain and design electrical and electronic circuits.

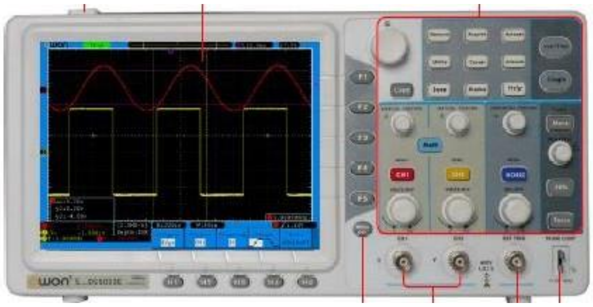
Laboratory equipment:

Millimeter: A multi-use device that can measure potential difference,current, resistance, and capacitance (and in some types of frequency).

Linear Circuit Lab.



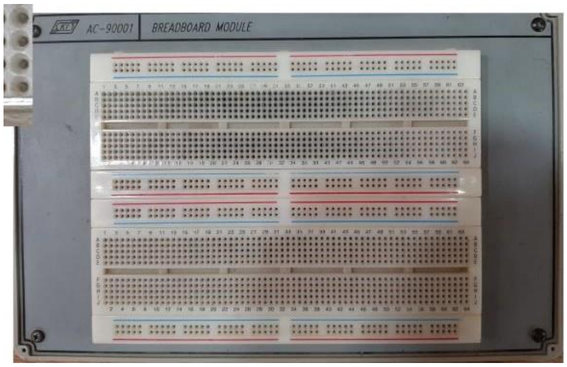
OSCILLOSCOPE: The OSCILLOSCOPE is one of the important devices used in the laboratory, as it is used to display the results of measurements as real-time plots on a screen divided like a graph paper in which the vertical axis is for voltage and the horizontal for time.



Work panel (breadboard) : The work panel consists of (640 points* 2) with every five vertical points connected to the electricalconnection. For example, points 1E, 1D, 1C,

1B, 1A are considered one point, and points 1J, 1I, 1H, 1G, 1F are considered one point

and separated from the previous ones.



Laboratory experiments:

Experiment 1: Basic information: Identifying laboratory equipment. Identifying the values of resistors by their colors and connecting rudimentary electrical circuits

Experiment 2: Ohm's law and the physical properties of conductors: Proving Ohm's law in practice when the physical properties of conductors are proven, and studying Ohm's law when the physical properties are not proven.

Experiment 3: Wave plotter (OSCILLOSCOPE): Learn about the work and uses of the OSCILLOSCOPE.

Experiment 4: Alternating current circuits and measuring the phase difference angle: Identifying the relationship between current and voltage in alternating current circuits and measuring the phase difference angle between current and voltage using Lesser shapes.

Experiment 5: Karchouf’s law: Proof of Karchouf’s law in practice in direct current circuits

Experiment 6: Phase representation of voltages and currents in alternating current circuits: Learn how to represent alternating variables with vectors and find the phase angle from the phase diagram.

Experiment 7: Superposition Theory: Proving the superposition theory practically in direct current circuits

Experiment 8: Thevenin’s Theorem: Investigation of Misch and Nodel’s analyzes in direct current circuits. Verifying Thevenin’s theorem practically in direct current circuits and proving that the greatest power transfer occurs when the load resistance is equal to the equivalent resistance value of the network.

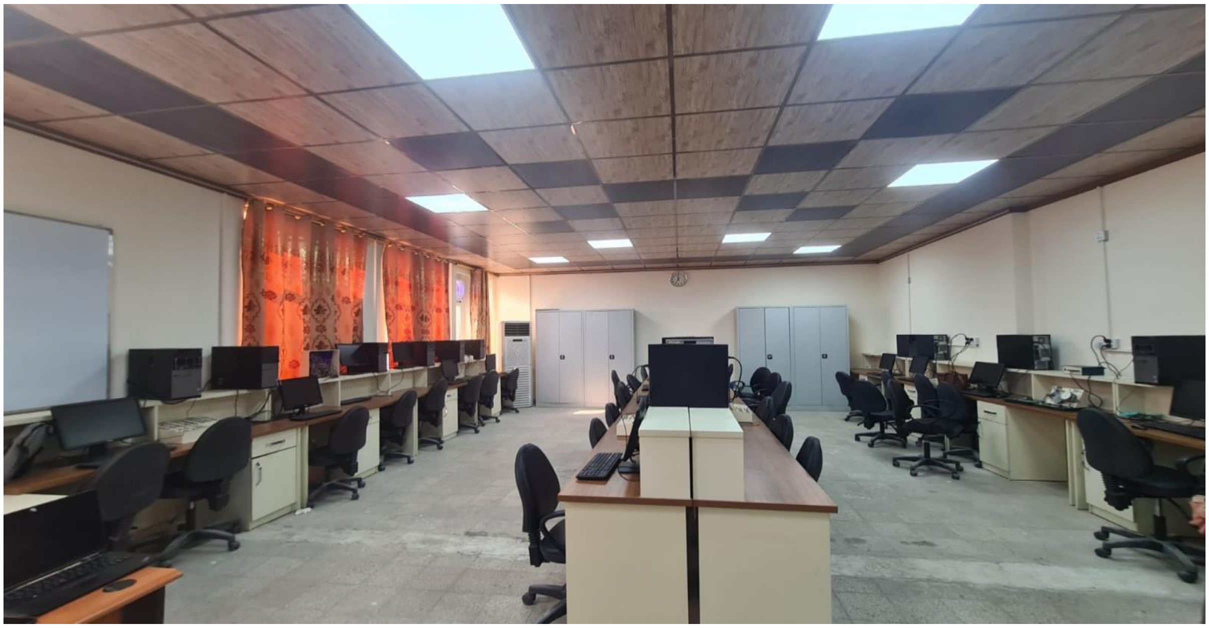
Experiment 9: Properties of the diode: Identify the properties of the P-N junction diode.

Experiment 10: p–n junction applications: Identify some p–n junction applications.

Experiment 11: The transient state in RL and RC circuits. Experiment 12: Transient state in RLC circuits.

Digital System Fundamentals Lab.

This laboratory is considered one of the basic laboratories for first stage students, as the laboratory represents the practical application of the theoretical study of the subject of digital and logical circuits, which the student studies during the first study in the Computer Engineering Department.



Laboratory objectives:

The basic objective of this Lab is to give an introduction to digital logic design with an emphasis on practical design techniques and hardware circuit implementation. Topics include:

1. Helping students understand the operation of basic logic gates, Boolean algebra theorems and its expression simplification, theory of Boolean logic functions, mapping techniques and logic function minimization through theoretical analysis.
2. Design of combinational and interactive digital circuits such as digital magnitude comparators, binary decoder and encoder, adder and subtractor logic circuits using separated components such as half-adder and full-adder.
3. Introducing the students to designing digital circuits using schematic capture, how they work, and logic simulation is included, the students also build circuits.

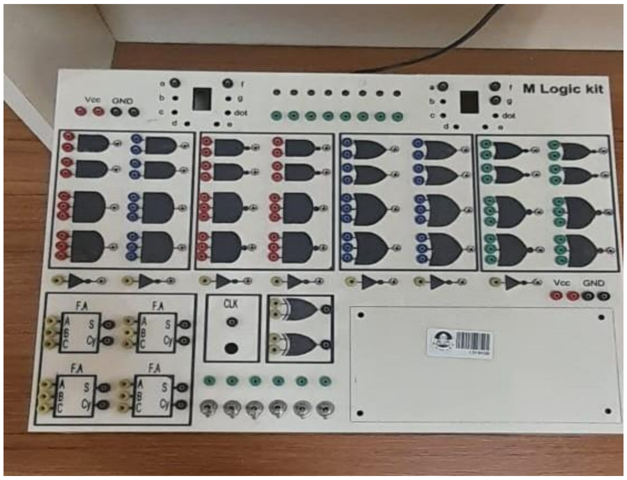
Experiments table

Week	Material Covered
Week 1	Experiment (1): Understanding the Operation of Basic Logic Gates
Week 2	Experiment (2): Boolean Algebra Laws
Week 3	Experiment (3): Boolean Expression Simplification
Week 4	Experiment (4): Functional Minimization using Karnaugh Map
Week 5	Experiment (5): Design of Digital Comparator Circuits
Week 6	Experiment (6): Adder and Subtractor Circuits using Half-Adder and Full-Adder
Week 7	Experiment (7): Design of Arithmetic Logic Unit and Implementation of Ripple Carry Adder Circuits

Laboratory Equipment:

Digital Logic Kit:

Digital logic kits are used to understand and experiment with fundamental principles of digital circuits. These kits typically include components like logic gates, flip-flops, and other building blocks. Students use them to create and analyze circuits, gaining hands-on experience in designing and troubleshooting digital systems.



An overview of how to use digital logic kits:

- **Understanding Components**: Begin by familiarizing the students with the components in the kit, such as AND, OR, and NOT gates, as well as flip-flops and other basic elements.
- **Circuit Design**: Use the components to design various digital circuits. This can include combinational circuits and sequential circuits .
- **Breadboarding:** Implementation of circuit designs on a breadboard. This involves physically connecting the components using jumper wires .
- **Testing and Verification:** Power up the circuit and observe the output. Verify that it behaves as expected based on the design. Debug any issues that arise.
- **Experimentation:** Modify the circuits to perform different tasks or achieve specific outcomes. This hands-on experimentation helps reinforce theoretical concepts.
- **Timing Analysis:** Explore the timing characteristics of circuits, especially in sequential circuits where the timing of signals is crucial.
- **Documentation:** Record your circuit designs, observations, and results. This documentation is crucial for understanding your work and for later reference.
- **Collaboration:** Often, lab work involves collaboration. Discuss your findings and solutions with peers to enhance learning.

Programmable Logic Design Lab

This laboratory is considered one of the important laboratories for Second stage students, as the laboratory instruct the students the basic principles of modern digital systems and programmable logic design, which the student studies during the second study in the Computer Engineering Department.

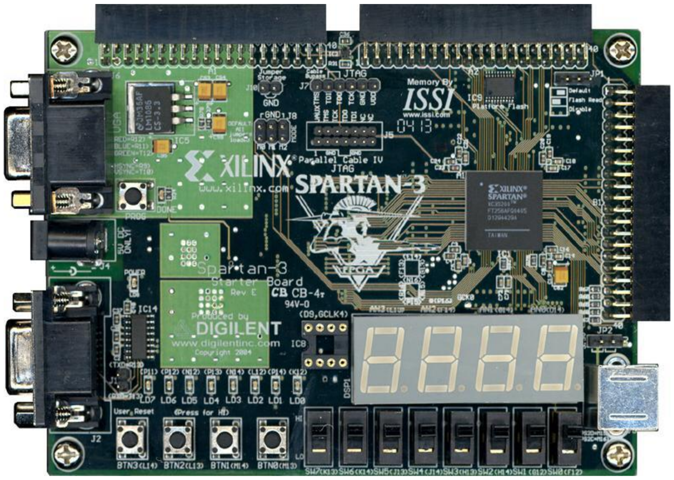
Laboratory objectives:

The basic objective of this Lab is to give an introduction to modern digital systems and programmable logic design with an emphasis on its implementation using hardware description language. Topics include:

1. Design and analysis of clocked sequential digital circuits such as latches, flip-flops, shift registers, counters, and pattern detector circuits.
2. Introduction to basic combinational components modeling, simulation, synthesis (with Xilinx, Altera, or Intel FPGAs)design and analysis tools.
3. Using very high-speed circuit hardware description language (VHDL) and Field programmable gate array (FPGA) design techniques to model basic sequential components and state-machines.
4. Gain proficiency in writing basic structural HDL code and the syntax, structure, and data types used in HDLs.

Delivery Plan (Weekly Lab. Syllabus)	
Week	Material Covered
Week 1	Experiment (1): Synthesis and Understanding the Operation of Latch, D Flip-Flop, J-K Flip-Flop, and T Flip-Flop
Week 2	Experiment (2): Design and Understanding the Operation of Shift Registers and Synchronous Counter Circuits
Week 3	Experiment (3): Using (Xilinx, Altera, or Intel FPGAs) Simulator as a Modeling Design, Synthesis, and Analysis Tool
Week 4	Experiment (4): Modeling Basic Combinational Components: Logic Gates, Multiplexers in VHDL
Week 5	Experiment (5): Modeling Basic Combinational Components: Decoders and Encoders in VHDL
Week 6	Experiment (6): Structural VHDL
Week 7	Experiment (7): Modeling Basic Sequential Components: Latches, Flip-Flops, Shift Registers, Counters in VHDL
Week 8	Experiment (8): VHDL State Machine

Laboratory Equipment:
Spartan-3 FPGA (Field-Programmable Gate Array) :



Spartan-3 FPGAs are valuable tools in a digital logic lab for hands-on learning and experimentation with digital circuit design, offering flexibility and configurability in implementing various logic circuits.

Tasks in a digital logic lab with Spartan-3 FPGAs may include designing and implementing combinational and sequential logic circuits, such as adders, multiplexers, and flip-flops. You can program the FPGA to behave like specific digital components and observe their functionality on the hardware.

Additionally, the Spartan-3 FPGA family supports the use of Hardware Description Languages (HDLs) like Verilog or VHDL. This enables you to describe your digital circuit at a higher level, making the design process more efficient.

Programming using C++ Language LAB

Course Description and Objectives:

This course introduces the students to C++ programming language, which is a starting level for getting into programming. It starts from programming basics and gives a holistic view of the C++ Programming language, detailing all the aspects of the C++ language from data types, to operators and expressions, to if statements, further to loops, arrays, strings. The course also provides hands-on training to help you write and test your coding skill, and prepare you for real-life application.

Course Outcomes:

At the end of this lab session, the students will develop a solid understanding of "Programming using C++" focusing on the fundamental concepts and skills necessary to develop engineering-oriented software using the C++ programming language. By achieving these outcomes, students will gain a solid understanding of C++ programming principles and be able to apply them effectively in practical programming scenarios.

List of Experiments:

- codeblock IDE (download, Installing, use)
- A simple C++ program using basic program construction (Identifiers, comments, variables, Assignment statements, output statement(cout)).
- A simple C++ program using input and output statements (cin, cout).
- Arithmetic and logical expression: Arithmetic operators, logical operators, relational operators.
- programs using selection statement (if-statement)
- programs using selection statements (if-statement and switch statement)
- programs using loop statements (for statement)
- programs using loop statements (while and do statements)
- Functions (call by value)
- Functions (call by reference)
- One dimensional array
- Two dimensional array
- Structures and Structure type functions

DATA STRUCTURES LAB

Course Description and Objectives:

The course is designed to develop skills to design and analyze simple linear and nonlinear data structures. It strengthen the ability to the students to identify and apply the suitable data structure for the given real world problem. It enables them to gain knowledge in practical applications of data structures.

Course Outcomes:

At the end of this lab session, the student will

- Be able to design and analyze the time and space efficiency of the data structure
- Be capable to identity the appropriate data structure for given problem
- Have practical knowledge on the applications of data structures

List of Experiments:

1. Design and Implement Simple Data Types and Abstract Data Types.
2. Design and Implement Stack general concepts and application.
3. Design and Implement Queues and Circular Queue.
4. Design and Implement Linked list.
5. Design and Implement Linked stack List.
6. Design and Implement Linked Queues List.
7. Design and Implement Linked Ring List.
8. Design and Implement Recursion and Trees (Binary Search Tree).
9. Implement Sorting algorithms and Searching algorithms.
10. Implement Hash tables.
11. Implement Files.

Microprocessor Laboratory

1) The microprocessors laboratory is considered a complement to the theoretical material given in the second stage and consists of two parts, namely, Microprocessors 1 in the first semester and Microprocessors 2 in the second semester.

2) Laboratory experiments rely on simulation programs specialized in likening the work of microprocessors to its two parts, which are the software entity and the hardware entity.

3) All laboratory experiments are carried out on the personal computers (laptops) provided by the laboratory, with one laptop for each student, to give each student the opportunity to learn and rely on himself, with the possibility of cooperating with his colleagues within the laboratory.

4) Software simulation systems consist of several programs operating within the (MSDOS) environment using the application (DOSBOX) and a number of specialized programs such as MASM, LINK, 8086 Emulator, Debug, and others, which enable the student to write programs in assembly language, detect textual errors, modify the programs, then convert them into executable programs, run them, debug then test, and execute them step by step to detect programming errors, learn from them, and correct them.

5) While the hardware simulators consist of the design suite Proteus 8 Professional which allows the student to design microprocessor and the interface circuits, operate them, write, and execute programs, and use virtual measuring devices and input and output tools to represent and test the operation of any designed circuit during experiments.

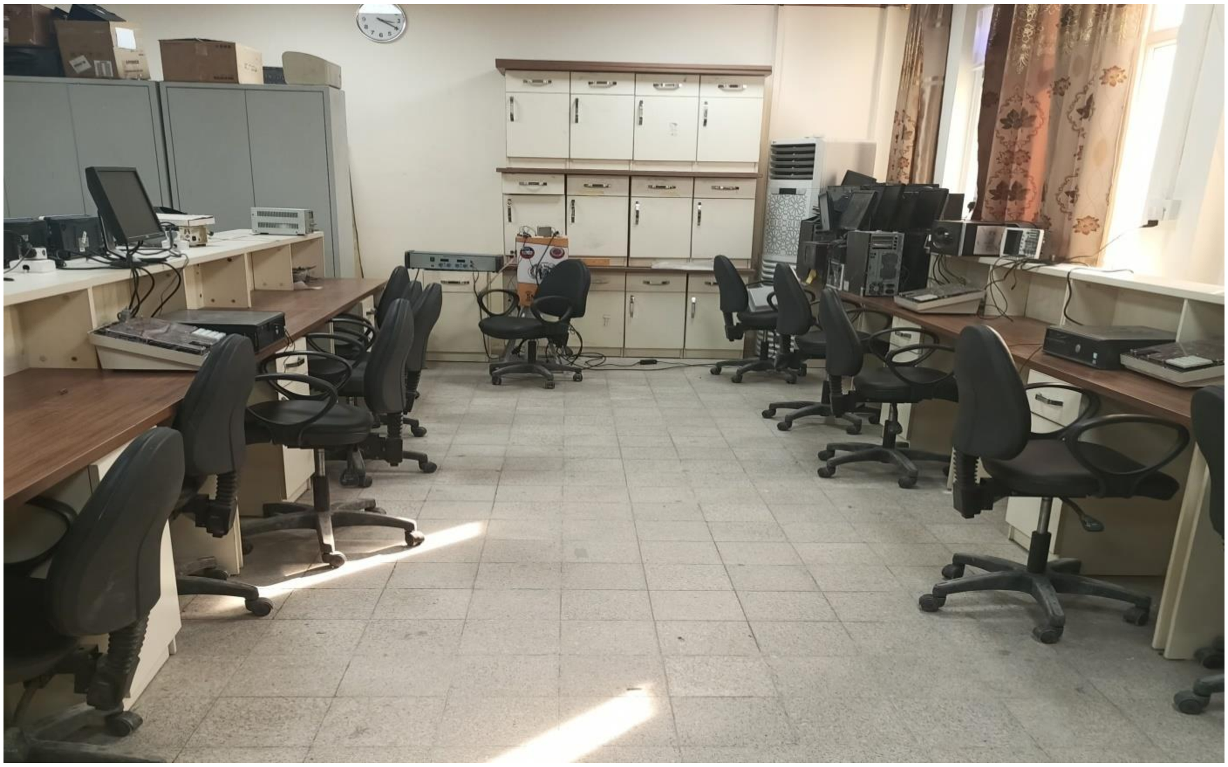
Table of experiments

	Experiment Title
1.	DEBUG Commands
2.	Data transfer instructions
3.	Data transfer instructions using MASM & LINKER
4.	Logical group and shift/rotate group
5.	Play with VRAM
6.	Play with VRAM using string instructions
7.	BIOS and DOS services
8.	Keyboard interface via BIOS services
9.	Mouse interface via DOS services
10.	Time and Date via DOS services
11.	Introduction to Proteus
12.	The microprocessor signals
13.	Address DE multiplexing
14.	Data buffering
15.	Control signals
16.	The address decoding (1)
17.	The address decoding (2)
18.	The memory interface (1)
19.	The memory interface (2)
20.	The I/O interface (1)
21.	The I/O interface (2)
22.	The I/O & memory interface (1)
23.	The I/O & memory interface (2)

Computer Interfacing lab.

In the realm of computer engineering and digital electronics, the Intel 8086 microprocessor stands as a pioneering entity, heralding a new era of computing capabilities. Paired with an array of supporting chips, the 8086 microprocessor forms the backbone of numerous computing systems, laying the foundation for modern computing architectures. In this laboratory experiment, we embark on a journey to unravel the intricacies of the 8086 microprocessor and its accompanying chips, delving into its architecture, instruction set, and interfacing principles.

Practical experiments are given at a rate of 1 experiment per a week. About 13 experiments were prepared, all of which were conducted practically in the laboratory. Benefiting from this laboratory include students in the third stages.



Objectives:

The primary objective of this laboratory experiment is to provide students with hands-on experience and in-depth understanding of the Intel 8086 microprocessor and its supporting chips. Through practical experimentation and analysis, participants will explore various aspects of the 8086 microprocessor, including its architecture, instruction set architecture (ISA), and input/output (I/O) operations. Additionally, students will gain insights into the role and functionality of supporting chips such as peripheral interface adapters PPI, Counter\ Timer, Interrupt controller chips, USART and DMA controller. The main unit that used for executing experiments is MTS-86C kit.

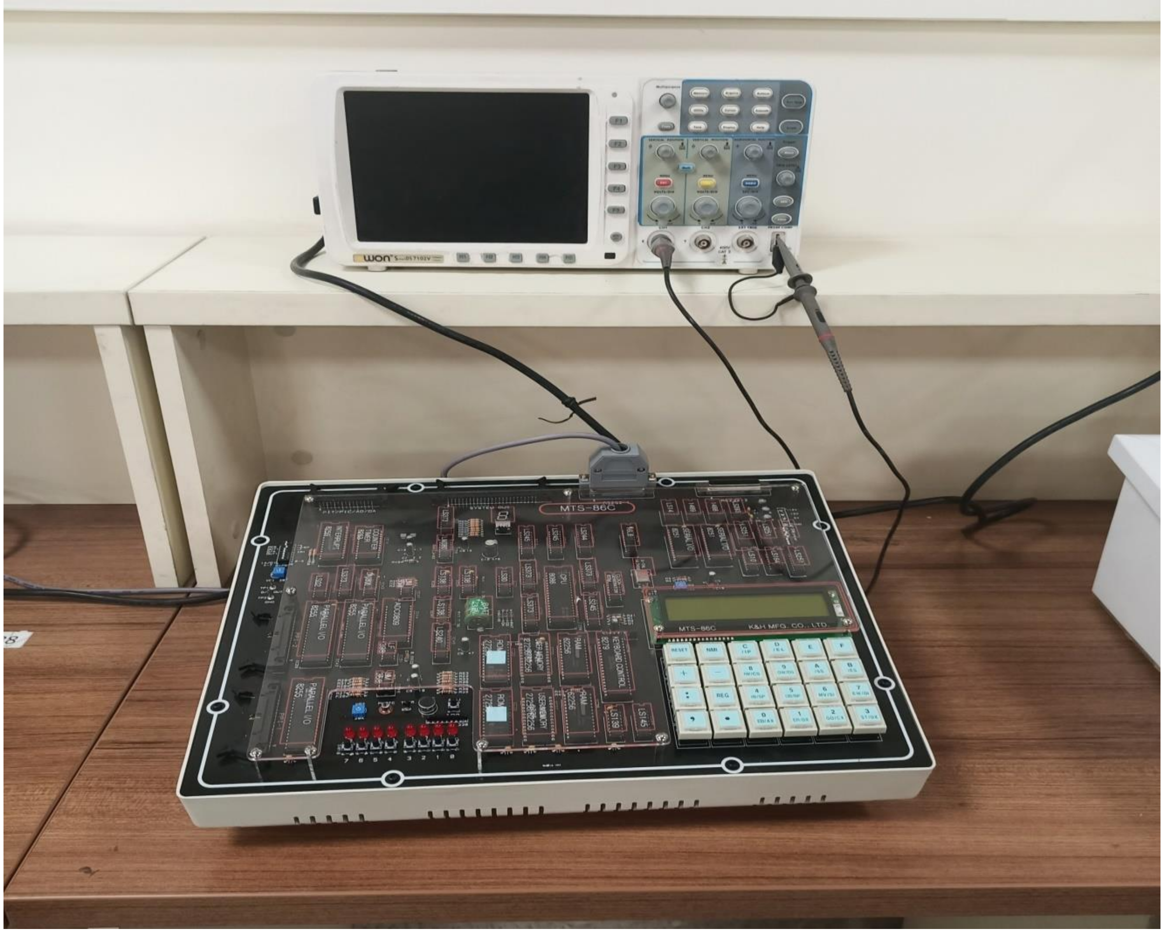


Table of experiments

No.	Experiment name
1	Introduction to MTS-86C Kit
2	The7-Segment Display and 74LS373 (part1)
3	8255 Programmable Peripheral interface Modes of operation (part1)
4	8255 Programmable Peripheral interface Modes of operation (part2)
5	8253 Programmable Interval Timer (PIT) Modes of Operation
6	Sound Generator using 8253PIT
7	8279 The Programmable Keyboard / Display
8	Digital to Analog Convertor DAC 0808
9	Analog to Digital Convertor 0809
10	Digital Sound Recorder
11	USART.1 8251 Universal Synchronous Asynchronous Receiver Transmitter USART (Asynchronous Mode)
12	USART.2 8251 Universal Synchronous Asynchronous Receiver Transmitter USART (Synchronous Mode)
13	8259 The Programmable Interrupt Controller PIC
14	Review
15	Exam

Embedded Systems lab.

Embedded systems form the cornerstone of modern technology, seamlessly integrating hardware and software to perform dedicated functions within various applications. Among the myriad platforms available for embedded development, Arduino has emerged as a popular choice due to its user-friendly interface, extensive community support, and versatility. In this laboratory experience, we delve into the realm of embedded systems using Arduino, aiming to provide students with practical insights into embedded hardware design, programming, and interface

Practical experiments are given at a rate of 1 experiment per a week. About 12 experiments were prepared, all of which were conducted practically in the laboratory. Benefiting from this laboratory include students in the third stages.

Objectives:

The primary objective of this laboratory is to familiarize students with the fundamental concepts of embedded systems and Arduino-based development. Through hands-on experimentation and project-based learning, participants will gain proficiency in designing, programming, and interfacing embedded systems using Arduino microcontrollers. The laboratory exercises will cover topics ranging from basic input/output (I/O) operations to advanced serial and parallel interfacing, A\D and Interrupt interfacing, enabling students to apply their knowledge in diverse embedded applications.

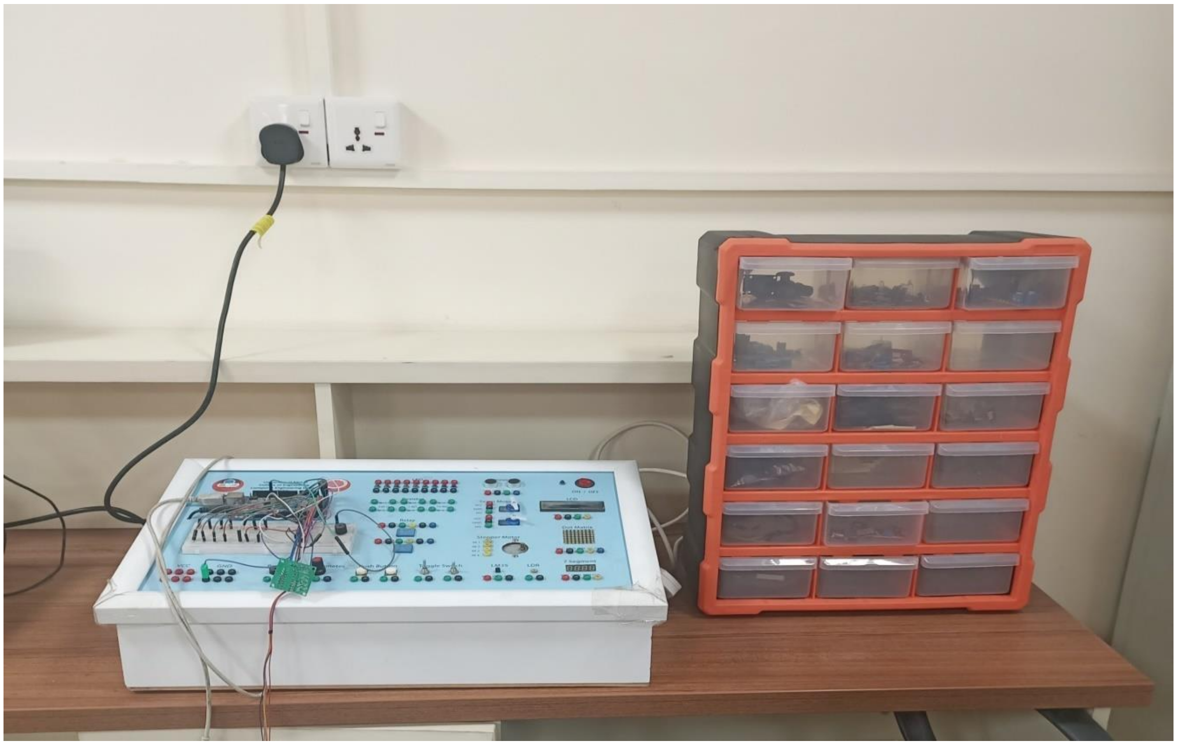


Table of experiments:

No.	Experiment name
1	Introduction To Arduino Microcontroller Chip
2	Install and configure lab software (Programming Arduino Chip)
3	Basic I/O programming
4	Traffic light system using Arduino
5	I/O LEDs and Buzzer
6	Analogue to digital convertor
7	Servo motor and stepper motor
8	Arduino with 7_ Segments
9	Up And Down Counter On The 7-Segment
10	Arduino with Keypad
11	Using DotMatrix with Arduino
12	Arduino wih LCD
13	Review
14	Term exam

Operating System with UNIX

Introduction: Third stage students study the operating systems course, which is given in two parts: operating systems 1 and operating systems 2, both practical and theoretical. The laboratory includes training students on the mechanism of dealing with more than one operating system within one computer. The following is the laboratory curriculum for the operating systems subject , in its first and second sections, for the third stage.

Section One: Operating System 1

Experiment 1: Training students on installing and dealing with the Linux system using the Command Line Interface. Within an Oracle Virtual Box environment. (Bash Experiment 2, 3: Identifying the basic instructions in the Linux system Command Line).

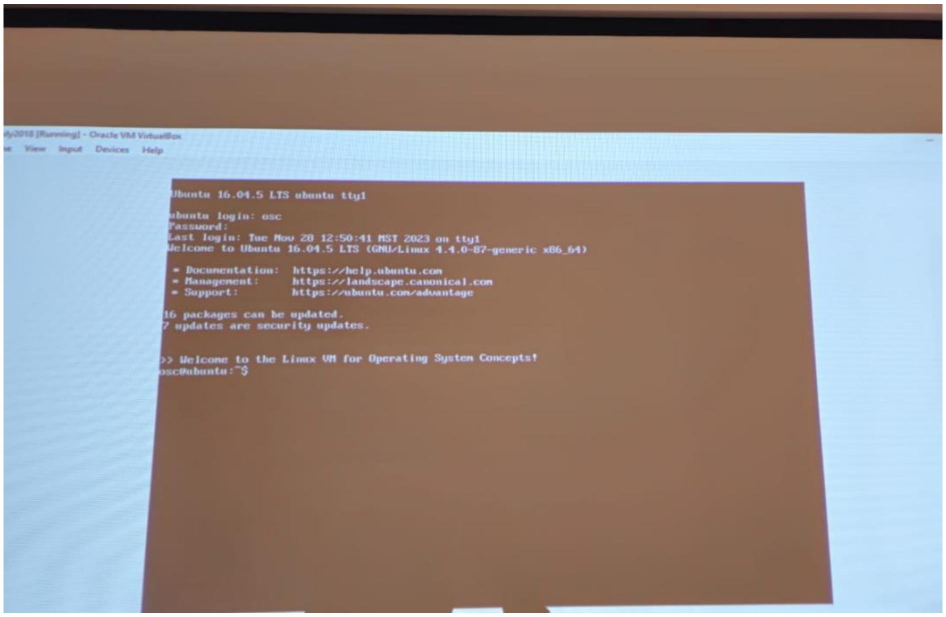
Experiment 4: Management System

- Create a file
- Copy one file to another
- Linking a file
- Delete a file.

Experiment 5: Creating Process
Experiment 6: Producer – Consumer Problem.
Experiment 7: Parent process–Child relationship process.
Experiment 8: CPU scheduling algorithms.

Section Two: Operating System 2

Experiment 1: Create Thread & Multithreading
Experiment 2: Synchronization thread with mutex.
Experiment 3: Memory Management.
Experiment 4: Logical and Physical Translation.
Experiment 5: sharing Memory between processes.
Experiment 6: Implementation of share a variable.
Experiment 7: Overall Review



```
Ubuntu 16.04.5 LTS ubuntu tty1

ubuntu login: osc
Password:
Last login: Tue Nov 28 12:50:41 MST 2023 on tty1
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

16 packages can be updated.
7 updates are security updates.


>> Welcome to the Linux VM for Operating System Concepts!
osc@ubuntu:~$
```

Control Systems Lab

A Control Lab serves as a dynamic learning environment for students pursuing studies in control systems engineering. This lab aims to bridge the gap between theoretical knowledge and practical applications through a series of hands-on experiments. Covering a diverse range of topics, the lab equips students with essential skills and insights into control theory and its implementation. The following experiments provide a structured curriculum that encompasses MATLAB, LabVIEW, analog and digital control systems, as well as programmable logic controllers (PLCs).

This comprehensive set of experiments provides students with a holistic understanding of control systems, combining theoretical knowledge with practical skills, and real-world applications using hardware components like PLCs and Arduino.

	(Control systems Lab. Experiments)
k	**Material Covered**
	Introduction to Matlab Familiarization with MATLAB, an essential tool for control system analysis and design.
	Graphical User Interface (GUI) using MATLAB Building on MATLAB basics, students learn to create user-friendly interfaces for control applications.
	Control Basics and Block reduction using MATLAB. Understanding fundamental control concepts and applying MATLAB for block reduction techniques.
	Transfer Function and Converting Between System Representations Exploring transfer functions and converting between different system representations.
	Laplace Transform Using MATLAB program.

	Exploring transfer functions and converting between different system representations.
	Step response and steady state error. Applying Laplace transforms in MATLAB to analyze step response and study steady-state error.
	Analogue Computer .Hands-on exploration of analogue computers and their relevance in simulating control system dynamics.
	LabVIEW basics Introduction to the LabVIEW programming environment and its basic functionalities.
	LabVIEW Control Design Model Applying control design principles using LabVIEW, with an emphasis on simulation and modeling.
	Design of Root Locus Understanding and applying the root locus method for control system design.
	Root Locus compensators Designing compensators using the root locus method to achieve desired control system performance.
	Time response analysis with LabVIEW Analyzing and visualizing time response characteristics using LabVIEW.
	Bode plot Understanding frequency response through Bode plot analysis.
	PID Controllers Design Designing Proportional-Integral-Derivative (PID) controllers using the Ziegler–Nichols tuning method.

	PLC Programming Introduction to programmable logic controllers and hands-on experience in programming.
	PLC applications Real-world applications of PLCs in control scenarios, emphasizing industrial use cases.
	Design with LabVIEW environment Advanced LabVIEW applications in control system design.
	LabVIEW LINX connection (Arduino and Raspberry PI) Interfacing LabVIEW with external devices like Arduino and Raspberry Pi using LabVIEW LINX.
	Motor Control using Arduino and LabVIEW Hands-on experience in controlling motors using Arduino and LabVIEW.
	Digital Control Design Introduction to digital control systems and design principles.
	Stepper motor DAQ LabVIEW Utilizing LabVIEW for Data Acquisition (DAQ) with stepper motors.
	DC motor control PID Implementing PID control for DC motor systems.

laboratory tools instruments

1. Analogue Servo unit 110-33 , Mechanical unit control and instrumentation 100-33.

2. Data Aacquisition Cards.

3. Raspberry Pi 4 and 3 versions. Arduino and sensors set.

4. LOGO 12\24 REC PLC.