

The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Electrical & Computer Engineering



Study Programme	: Bachelor of Technology Honours in Engineering
Name of the Examination	: Final Examination
Course Code	: EEX4436
Title	: Microprocessors and Interfacing
Academic Year	: 2024/25
Date	: 25 th January 2026
Time	: 1330-1630hrs

General Instructions

1. Read all instructions carefully before answering the questions.
2. This question paper consists of **Four (4)** questions in **SECTION A** [70 Marks] and **two (2)** questions [30 Marks] in **SECTION B** in **Four (4)** pages.
3. Answer **ALL Questions**.
4. The answer to each question should commence on a new page.
5. This is a Closed Book Test (CBT).
6. Answers should be in clear handwriting.
7. Do not use Red colour pens.

Special Instructions

- When you write any Assembly Language Program (ALP), you need to provide appropriate comments where necessary. **Full marks will only be given to correct programs with comments.**
 - Refer **datasheet of the 8051 microcontrollers (given separately)** when you answer the questions in this paper. **Do not attach it to the answer script.**
 - State your assumptions (if any) clearly.
-

Section A

Answer all questions [70 Marks]

Automated Tire Storage System (ATSS)

The Automated Tire Storage System (ATSS) is designed to organise and retrieve motor bicycle tyres from a vertical stacking facility without manual labour. The storage area consists of a (3×3) grid where tyres are stacked up to 10 levels high. The system utilises a 3-axis gantry robot to locate, grip, and transport tyres to a delivery conveyor.

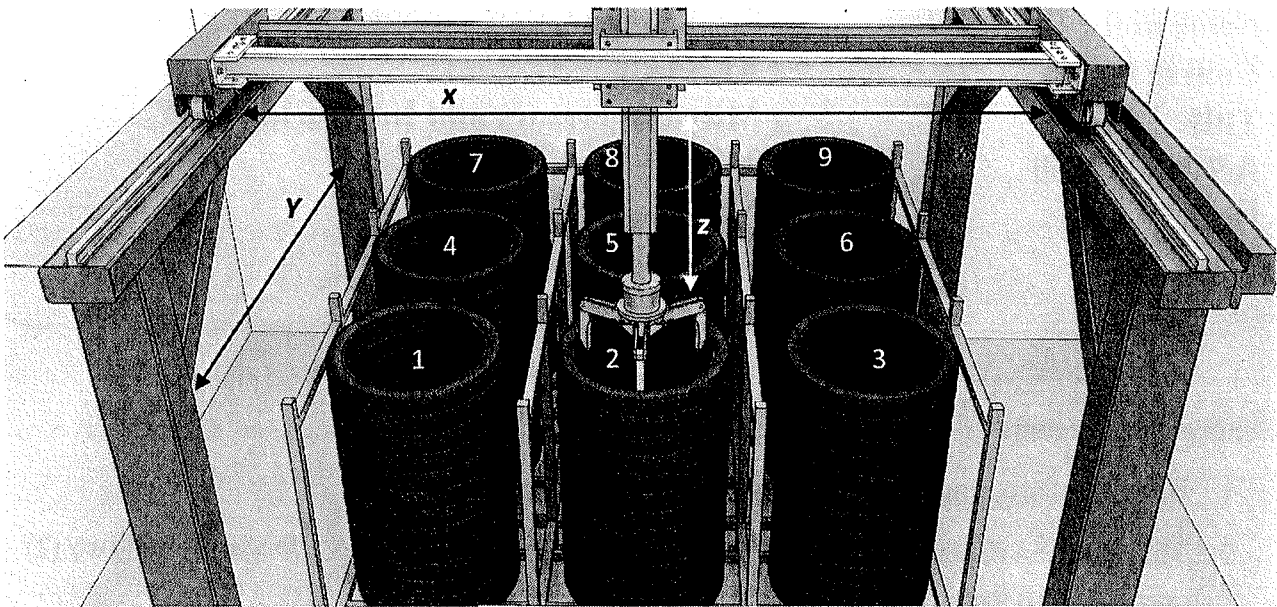


Figure 1(a): 3D view of the ATSS System

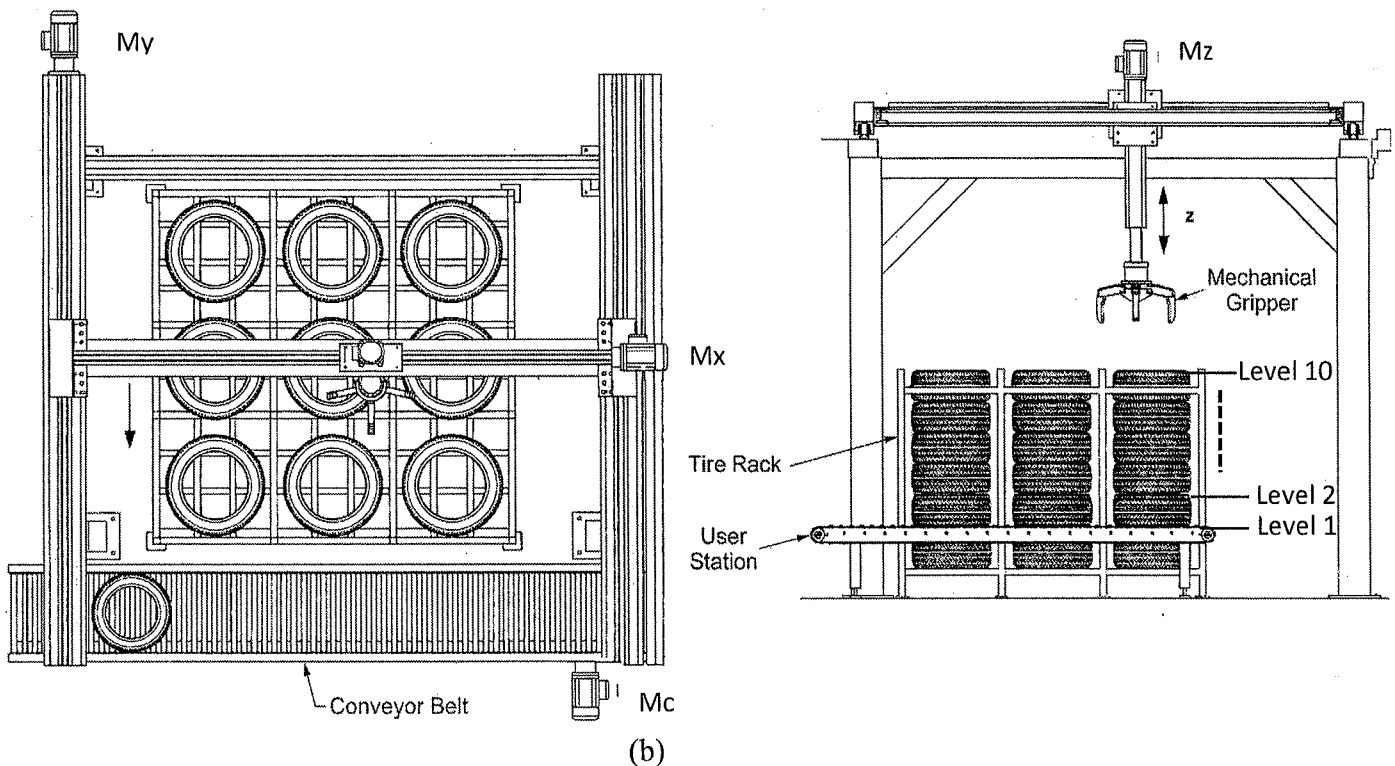


Figure 1(b): Top and Side View of the ATSS System

Figure 1(a) and 1(b) shows the major components of the ATSS. The system operates in a Cartesian (X, Y, Z) coordinate system, and in this task, we consider only the tyre retrieving mechanism.

X-Axis (Horizontal Main Rail) is driven by the Stepper Motor **M_x**. Y-Axis (Cross Rail) is driven by the Motor **M_y**. To ensure accurate alignment with the tyre columns, discrete sensors are fixed along the Y-axis frame to detect the specific "Stop Positions" for each of the 3 rows.

Z-Axis (Vertical Arm) is driven by a heavy-duty stepper Motor **M_z** to lift and lower the arm. The robot utilises a Pneumatic Gripper (controlled by a Solenoid Valve). This 3-arm gripper is designed to enter the center of the tyre and expand outward to grip the tyre rim securely from the inside.

A conveyor belt, driven by the DC Motor **M_c**, transports the tyre to the user station. The Motor **M_c** operates (at a constant velocity with 50% PWM) *only* when a tyre has been successfully deposited onto the belt.

The main interface includes a 4x3 Matrix Keypad used to enter the "Tire Location ID" (1 through 9). As Status Indicators, A Green Light (System Ready) and a Red Light (Error/Emergency) are available. There are 3 control buttons for ON, STOP, and M (Manual Override). In "Manual Override" mode (M), the operator can bypass the automated sequence and control the gantry axes directly using a tethered handheld controller module.

Operational Sequence

1. In Idle State, the green light is ON. The robot waits at the "Home Position" (Location ID 5) and Gripper(Z) at level 10 as the default position.
2. The operator enters a specific Tire Location ID.
3. The Gantry moves (X and Y axes) to the selected grid coordinates.
4. The robot must automatically detect the current height of the tyre stack (which varies from 0 to 10 tyres) to determine the correct stopping point for the Z-axis.
5. The Z-axis lowers to the detected level. The gripper expands to hold the tyre. The robot lifts the tyre, moves to the conveyor drop-off zone, lowers to the lowest Z-level, and releases the tyre.

An emergency state is triggered if the operator presses the "STOP" button or if the system detects a critical fault. In an emergency, all motors (M_x, M_y, M_z, M_c) immediately cut power and engage brakes to freeze the robot in its current position. The pneumatic gripper maintains its pressure state. If it is holding a tyre, it must not release it. This prevents the heavy load from falling and causing injury or damage. The Red Light turns ON, and an audible alarm will ON alert nearby personnel. The system remains frozen until the operator acknowledges the fault. To reset, the operator must switch to Manual Mode (M) and use the handheld controller to manually move the robot to a safe position or complete the task before restarting the automated cycle.

State all the assumptions clearly (if any) when answering questions.

[Q1] For the proposed mechanism, identify suitable sensors and actuators to detect the following and draw simple block/interfacing diagrams for each to explain your methodology.

- (i) To identify the stop positions along the Y-Axis. [03 Marks]
- (ii) To identify a tyre that is on the conveyor belt. [04 Marks]
- (iii) To identify the level Z-Axis, which should move to grab a tyre. [04 Marks]
- (iv) By giving reasons, identify motor types for Motor **My**. [02 Marks]

[Q2]

- (i) Analyse the given specifications/parameters related to the operations/processes of the ATSS and then prepare a Table of resources/features and conditions/constraints required to design the ATSS. (Table should contain, Processes, Sub-Operations, Parameters, Conditions, Resources, Feature Required of 8051 microcontrollers, and Comments). [03 Marks]
- (ii) Draw the external view of the system, i.e., a diagram that shows the inputs/sensors and the outputs/actuators of the system. [08 Marks]
- (iii) Identify the subunits/sub-modules of the system and draw the interconnected block diagram of the system using the central controller and other required interfacing devices. [08 Marks]
- (iv) Draw an Interfacing diagram to show the 8051 port connections. [08 Marks]

[Q3] Draw flowchart(s) to represent the algorithm of the ATSS operations **only to fetch two tyres from Tire Location ID 4**. Assume initially the system is fully loaded with tyres. [15 Marks]

[Q4] Write assembly language programs (ALP) to perform the above task (i.e., Q3). Clearly show the assembly language routines with comments corresponding to the flowcharts in Q3 above. [15 Marks]

Section B

Answer all questions [30 Marks]

[Q5]

1) The 8051 microcontroller has limited internal RAM. You are required to transfer a block of data to an external memory chip.

(i) Explain the difference between the instructions MOV A,@R0 and MOVX A,@DPTR. [02 Marks]

(ii) Write an Assembly Language Program (ALP) to transfer 10 bytes of data currently stored in Internal RAM, starting at location 40H, to External RAM, starting at location 2000H. [05 Marks]

2) A student needs to generate a 2 kHz Square Wave on Pin P2.5 to drive a buzzer. The microcontroller uses an 11.0592 MHz crystal.

(i) Calculate the exact value required for the TH0 and TL0 registers to achieve this frequency. Assume Timer 0 is used in Mode 1 (16-bit timer). [03 Marks]

(ii) Write the ALP to configure the TMOD register and generate this square wave continuously using the values calculated above. [05 Marks]

[Q6]

You are asked to design a remote-control interface using an 8051 microcontroller. The system receives commands from a PC via the Serial Port. Crystal Frequency is 11.0592 MHz, and Baud Rate is 4800 bps. The system must use Serial Interrupts to process incoming data so the main program can perform other tasks uninterrupted. Assume the SMOD bit in the PCON register is 0.

(i) Calculate the value that must be loaded into Timer 1 (TH1) to establish a baud rate of 4800 bps. Show your calculation steps clearly. [03 Marks]

(ii) Draw a flowchart for the Serial Interrupt Service Routine (ISR) only. The ISR must perform the following logic whenever a character is received.

- Read the received character from SBUF.
- Check: Is the character equal to '#'?
 - YES - Toggle the LED connected to P1.0.
 - NO - Transmit (Echo) the received character back to the PC.
- Clear the Interrupt flag.
- Return from Interrupt.

[06 Marks]

(iii) Write the complete ALP.

(Your code must include initialise TMOD, SCON, TH1, and enable the Serial Interrupt (ES) and Global Interrupt (EA). Then, enter a loop that does nothing (simulate background tasks). Implement the logic designed in part (ii) at the correct Vector Address (0023H) [06 Marks]

END